The Computational Neuroscience of Head Direction Cells

Daniel Walters

Submitted for the Degree of DPhil in Experimental Psychology

The University of Oxford

Trinity Term 2011

Acknowledgements

There are many people without whose advice, support and encouragement this thesis could not have been written.

Firstly, I would like to offer a huge thank you to my supervisor, Dr. Simon Stringer, for his help, expert advice, patience, and also for being a genuine friend.

I would also like to thank Dr. Mark Buckley. From the OFTNAI lab I would like to thank Ben, Jamie, Erica, and Bedeho. Thanks, guys, for making the lab such an enjoyable place to be.

A big thank you is also due to Paul, Ruth, and Jon. You are very close friends who have played a big role in ensuring that it isn't all about the work.

From back home (Sheffield), I would like to thank Ian and Danny. Two long-time friends of mine who have ensured that there have been plenty of fun times when taking a break from the research.

Last, but of course not least, I would like to thank my family for their constant encouragement, love, and latterly, financial support. So, in no particular order, a massive thank you to Rog, Gen, Beth, Betty, Jim, Chris, Mike, Sophia, Sean, Anthony, Amelia, Daisy, Matt, Julie, and Flo. I couldn't have done it without you!

Short Abstract

The Computational Neuroscience of Head Direction Cells
Daniel Walters
The Queen's College
Submitted for the Degree of DPhil in Experimental Psychology
Trinity Term 2011

Head direction cells signal the orientation of the head in the horizontal plane. This thesis shows how some of the known head direction cell response properties might develop through learning. The research methodology employed is the computer simulation of neural network models of head direction cells that self-organize through learning. The preferred firing directions of head direction cells will change in response to the manipulation of distal visual cues, but not in response to the manipulation of proximal visual cues. Simulation results are presented of neural network models that learn to form separate representations of distal and proximal visual cues that are presented simultaneously as visual input to the network. These results demonstrate the computation required for a subpopulation of head direction cells to learn to preferentially respond to distal visual cues. Within a population of head direction cells, the angular distance between the preferred firing directions of any two cells is maintained across different environments. It is shown how a neural network model can learn to maintain the angular distance between the learned preferred firing directions of head direction cells across two different visual training environments. A population of head direction cells can update the population representation of the current head direction, in the absence of visual input, using internal idiothetic (self-generated) motion signals alone. This is called the path integration of head direction. It is important that the head direction cell system updates its internal representation of head direction at the same speed as the animal is rotating its head. Neural network models are simulated that learn to perform the path integration of head direction, using solely idiothetic signals, at the same speed as the head is rotating.

Long Abstract

The Computational Neuroscience of Head Direction Cells
Daniel Walters
The Queen's College
Submitted for the Degree of DPhil in Experimental Psychology
Trinity Term 2011

Within the rat brain there are classes of cells that signal spatial information. Place cells signal the current location of the rat within its environment. Entorhinal cortex grid cells also signal the location of the rat within its environment. The receptive field of each grid cell responds to multiple locations of the rat within an environment such that the receptive field of a particular grid cell tessellates in a hexagonal grid-like pattern. Head direction cells signal the current orientation of the head in the horizontal plane.

Individual head direction cells exhibit stereotypical, approximately Gaussian, response profiles. Each head direction cell has a single response profile that is centred upon a single preferred head direction. Within a population of head direction cells, a single packet of neural activity will reflect the head direction of the rat. In effect, head direction cells provide a neural compass for the rat.

This thesis explores the computational neuroscience of head direction cells. That is, how the known properties of head direction cell responses may develop through learning, for both individual head direction cells and populations of head direction cells, to produce the computations required for a rat to navigate through its environment in the presence of visual input, and in the absence of visual input.

The research methodology employed is the computer simulation of firing-rate based neural network

models of head direction cells that self-organize through learning.

Within a particular environment, the preferred firing direction of an individual head direction cell can change if the visual cues in the environment are rotated. Previous research has shown that the preferred firing direction of each head direction cell is influenced more by the rotation of distal visual cues, relatively far away from the rat, than by the rotation of proximal visual cues, which are relatively close to the rat.

This differential influence of distal visual cues upon head direction cell preferred firing directions, compared to proximal visual cues, may be explained by the fact that, as the rat rotates, the bearings to the distal visual cues directly reflect the head direction of the rat across different locations within the environment. In contrast, the bearings to the proximal visual cues vary independently of the head direction of the rat across different locations. This can allow a competitive learning mechanism, within a feedforward neural network architecture, to form separate output representations of the distal and proximal visual objects. Cells that respond to the distal visual cues then respond like head direction cells to visual input from the environment.

This thesis begins with the development of a neural network model comprising a single layer of input cells, connected through feedforward synapses to a single layer of output cells, which can, through competitive Hebbian learning, form separate representations of distal and proximal visual cues that are presented simultaneously to the network as input.

The network is first simulated with orthogonal input representations. Within the layer of input cells, separate, non-overlapping, subsets of the input cells are set to respond to either input from the distal visual cue space, or input from a proximal object space. The computational problem that this model must learn to solve is to form separate output layer cell representations of the distal and proximal cue spaces. That is, an individual output layer cell must learn to respond exclusively to a narrow region of either the space of distal visual cues or a proximal object space, but not both.

The simulation results show that, under the correct conditions, the neural network model can learn to discriminate between distal and proximal visual cues that are presented to the network simultaneously. In particular, a rat must move through its environment in order that a population of output cells may learn to form separate representations of the distal and proximal visual cues.

These results demonstrate the computation that is required in order for individual head direction cells to be preferentially influenced by distal visual cues rather than proximal visual cues.

The network is then simulated with distributed input representations. Subsets of input layer cells are permitted to overlap such that the subsets share a number of common input cells. This model must learn to solve a harder computational problem than the model simulated with orthogonal input representations. The distributed nature of the input representation means that an individual input layer cell may signal visual input from the space of distal objects, then visual input from a proximal object, at different model updates. Consequently, the model must learn to form separate output layer representations of the distal and proximal object spaces when individual input layer cells can signal the

presence of more than one object space.

In simulation, this neural network model can, under the correct conditions, learn separate representations of the distal and proximal object spaces. Careful replication of the experiments conducted for the model with orthogonal input representations demonstrates that there is a general principle that is robust with respect to the nature of the input representation: the statistical properties of distal and proximal visual cues can help a competitive learning mechanism to form separate representations of these cue spaces.

A property of a population of head direction cells is that the angular distance between the preferred firing directions of any two head direction cells is maintained across different environments. This is called an isomapping of head direction cell preferred firing directions.

A neural network model consisting of a single layer of input cells connected, through feedforward synapses, to a single layer of output cells, which is, in turn connected to itself through excitatory recurrent collateral synapses, can learn an isomapping of output layer cell preferred firing directions across two visual training environments. The model learns the correct behaviour because the excitatory recurrent collateral synapses help to shape the activity profile in the output layer such that cells which fire together in the first environment will tend to fire together in the second environment.

The results of simulating this model show that it is possible to learn an isomapping of output layer cell preferred firing directions using visual input alone. The nature of the mapping between output layer

cell preferred firing directions is dependent upon the strength of the excitatory recurrent collateral synaptic input. Strong recurrent collateral synaptic input will produce an isomapping of output layer cell preferred firing directions across two training environments. Weak recurrent collateral synaptic input will, instead, produce a place cell-like complex remapping of the output layer cell preferred firing directions across two training environments.

The models that can learn to form separate output layer representations of distal and proximal object spaces, and the model that can learn an isomapping of output layer cell preferred firing directions across two training environments, show how some of the known properties of head direction cells can develop through learning based on visual input alone.

This thesis also examines how the head direction cell system may learn to update its internal representation of head direction using internal idiothetic (self-motion) signals in the absence of visual input. This is called the path integration of head direction.

An important computational problem regarding the path integration of head direction is how the head direction cell system learns to update its internal representation of head direction at the same speed as the rat is rotating its head. No previous research has addressed how a self-organizing neural network model can solve this problem in a biologically plausible manner.

A continuous attractor neural network of head direction cells that is reciprocally linked to layer of combination cells can learn to update a packet of head direction cell activity, during testing in the

absence of visual input, at the speed that was imposed during training in the presence of visual input. The key to learning to solve this problem is to introduce time delays into the model so that associations can be learned between head direction cells occurring at different instances in time. These learned associations enable the network to replay the correct sequence of head directions at the correct speed in the absence of visual input, i.e. the network learns to perform path integration of head direction.

The network is first simulated with axonal conduction delays incorporated into the synapses between the head direction cell continuous attractor network and the layer of combination cells. The axonal conduction delays introduce a time delay into the model because it takes a small amount of time for the signal from a presynaptic neuron to travel down the axon and reach the postsynaptic neuron.

In simulation, this model can learn to perform path integration of head direction, in the absence of visual input, at approximately the speed imposed during training in the presence of visual input. This is the first model to show how the head direction cell system can learn, in a biologically plausible manner, to perform path integration of head direction at the correct speed.

A second neural network model employs a different mechanism for generating time delays. In this model there are no axonal conduction delays. Instead, a large value for the neuronal time constant of a postsynaptic cell ensures that the firing profile of the postysnaptic cell will be delayed in time from the firing profile of the presynaptic cell that is driving the postsynaptic cell. This effective time delay allows associations to be learned between head directions occurring at different instances in time.

The simulation results show that this model can learn to perform path integration of head direction, in the absence of visual input, at approximately the speed imposed during training in the presence of visual input. Careful replication of the experiments conducted for the model incorporating axonal conduction delays illustrates that the principle of using a time delay as a mechanism for learning the correct speed of path integration is a general principle. Any biologically plausible mechanism that can produce consistent effective time delays could be incorporated into this type of model.

The two models that learn to perform path integration of head direction at the speed imposed during training demonstrate how a population of head direction cells can function in the absence of visual input.

This thesis thus addresses how individual head direction cells develop their firing properties through training in the presence of visual input, and how a population of head direction cells can act collectively to learn a population behaviour, the path integration of head direction, in the absence of visual input.

The two strands to the research presented in this thesis provide the foundation upon which a unified model of the head direction cell system can be built. This unified model will learn, in a biologically plausible manner, to produce both the known response properties of individual head direction cells in response to visual input and the known response properties of populations of head direction cells in the absence of visual input. The model will be based upon the known physiology of areas of the rat brain that contain head direction cells, and also upon the properties of the head direction cells within

these different areas. The unified model will allow a close correspondence between the computational modelling of the head direction cell system and the results of single-cell recording experiments performed upon the head direction cell system.

Contents

Chapter 1

The Visual Response Properties of Head Direction Cells

This chapter provides a literature review of research on the visual response properties of head direction cells, and develops the argument that these visual response properties can be produced through a competitive learning mechanism that acts upon visual processing.

1.1 Head Direction Cell Response Properties

Head direction cells signal the orientation (direction) of the head in the horizontal plane (Ranck, 1985; Taube et al., 1990a). Each head direction cell responds maximally to a single preferred head direction of the animal. The response profile of a head direction cell is Gaussian or triangular, and is centred on the preferred head direction of the cell in question. The width of an individual head direction cell tuning curve varies depending upon where in the rat brain the head direction cell is located. The tuning curve width is the range of head directions, $0^{o} - 360^{o}$, over which an individual head direction cell exhibits a non-zero firing rate. In general, head direction cell tuning curve widths lie in the approxi-

mate range of $40^{\circ} - 110^{\circ}$ (Sharp, 2005) with, for instance, head direction cells in the rat retrosplenial cortex possessing an average tuning curve width of 44.6° (Cho and Sharp, 2001) , and head direction cells in the rat dorsal tegmental nucleus possessing an average tuning curve width of 109.4° (Sharp et al., 2001). The firing rate of a head direction cell decreases in a symmetric manner from the centre of the tuning curve, with zero, or near-zero, firing rates outside the width of the tuning curve. This behaviour is independent of the tuning curve width.

The preferred firing directions of head direction cells are not sensitive to geomagnetism (Taube et al., 1990b). The preferred firing direction of an individual head direction cell is independent of the current location of the animal, and is independent of the current bearing of the head relative to the body of the animal (Taube et al., 1990a,b). In a population of head direction cells, the preferred head directions of the individual cells are uniformly distributed such that the population of cells collectively represents allocentric head directions from 0°-360°. In effect, a population of head direction cells provides a "neural compass" for the animal.

Head direction cells have been discovered in the postsubiculum (Ranck, 1985; Taube et al., 1990a), anterodorsal thalamic nucleus (Taube, 1995), lateral mammillary nuclei (Stackman and Taube, 1998), and retrosplenial cortex (Chen et al., 1994; Cho and Sharp, 2001) of the rat brain. Head direction cells have also been found in the following areas of the rat brain: the dorsal tegmental nucleus (Sharp et al., 2001), the dorsal striatum (Wiener, 1993; Mizumori et al., 2000), the lateral dorsal thalamus (Mizumori and Williams, 1993), the medial precentral cortex (Mizumori et al., 2005), the entorhinal cortex (Sargolini et al., 2006), and region CA1 of the hippocampus (Leutgeb et al., 2000). Within the

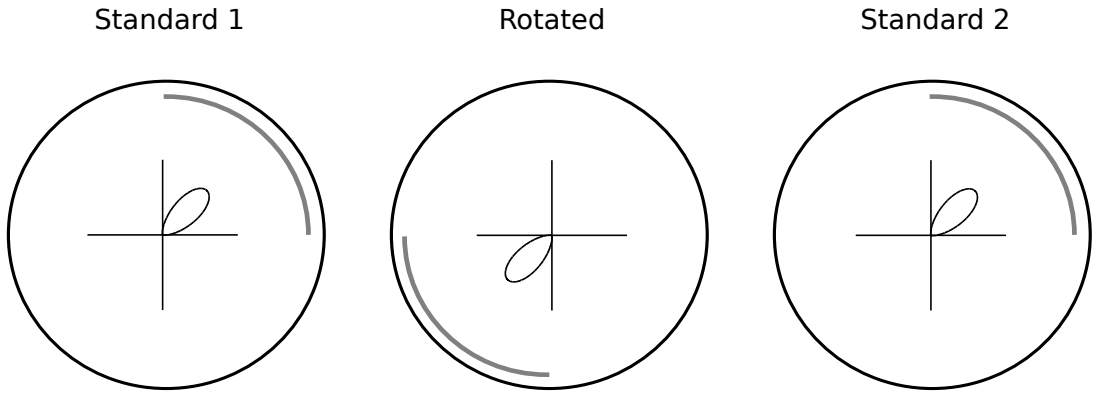

Figure 1.1: A schematic representation of the paradigmatic head direction cell visual cue control experiment (see the text for full details). The figure presents a top-down view of the cylinder into which the rat is placed. The grey arc represents the cue card. The cue card is attached to the wall of the cylinder, but is shown a small distance away from the wall for clarity. *Standard 1*: A head direction cell response is recorded as the rat explores the cylinder. This is shown in polar coordinates in the centre of the cylinder. The response profiles in the figure are abstract representations of head direction cell behaviour, and do not display data recorded from real head direction cells. *Rotated*: The rat is removed from the cylinder and the cue card is rotated; in this case through 180^{o}. The rat is placed back into the cylinder and permitted to explore as a recording of the head direction cell is taken. As can be seen, the preferred head direction of the cell has rotated by an amount approximately equal to the rotation of the cue card. *Standard 2*: The rat is again removed from the cylinder and the cue card rotated back to its original position. The rat is reintroduced to the cylinder and the head direction cell response is recorded. The preferred head direction of the cell rotates, matching the cue card, back to its original preferred head direction exhibited in Standard 1.

rat brain individual head direction cells share the same general response characteristics, regardless of their exact anatomical location (Taube, 1998, 2007; Sharp, 2005).

The majority of the head direction cell research has focused upon the rat brain. It should be noted, however, that head direction cells have been recorded from other species, including primates (Robertson et al., 1999), chinchillas (Muir and Taube, 2002), and mice (Khabbaz et al., 2000). This thesis will focus upon the research conducted upon head direction cells in rats. In principle, the results reported in this thesis are applicable to head direction cells independent of the species.

1.1.1 Visual Cue Control of Head Direction Cell Responses

Single-cell recording studies have shown that the preferred head direction of an individual head direction cell can be influenced by visual cues (Taube et al., 1990b; Goodridge and Taube, 1995; Zugaro et al., 2000). In the paradigmatic experiment, a rat is placed into a cylinder approximately 75cm in diameter and 50cm in height. A single, visually distinctive, cue card occupying approximately 100^{o} of arc is attached to the cylinder wall. This is shown schematically in Figure 1.1. The rat is allowed to explore the interior of the cylinder as the responses of a single head direction cell are recorded (some studies have recorded from multiple head direction cells simultaneously). The rat is then removed from the cylinder, the floor paper replaced, and the cue card rotated by a predetermined amount. The rat is often disoriented to prevent any tracking of current orientation by an internal sense of direction. The rat is re-introduced to the cylinder and further recording of the head direction cell takes place. The rat is then removed from the cylinder, the floor paper again replaced, and the cue card rotated back to its original position (again, the rat is often disoriented in between recording sessions). The rat is re-introduced to the cylinder and a final head direction cell recording session takes place.

The head direction cell responses are shown in polar coordinates in Figure 1.1. These plots are abstract representations of head direction cell behaviour and do not display data recorded from real head direction cells. As the cue card is rotated out-of-sight of the rat, the preferred head direction of an individual head direction cell rotates by an amount approximately equal to the rotation of the cue card. Thus, under the correct environmental conditions, the location of the cue card directly influences the preferred head directions of individual head direction cells.

The influence that visual cues can have upon head direction cell preferred firing directions is computationally very important when it is considered that visual cues can provide a directional fix, and thus calibration, to the head direction cell system. This calibration allows the current internal representation of head direction to be corrected so as to match the true current head direction of the rat. Without the ability to reset the head direction cell system, errors will accumulate to produce a large discrepancy between the current internal representation of head direction and the true head direction of the rat. A pertinent example of error accumulating in the head direction cell system is provided by data recorded from head direction cells in the postsubiculum and anterodorsal nucleus of blindfolded rats. At the end of an 8 minute blindfolded recording session, the preferred firing directions of head direction cells were shown to have drifted when compared to the preferred firing directions of the same cells at the start of the recording session (Goodridge et al., 1998). An analysis of the head direction cell preferred firing directions in the blindfolded recording session, compared to the same head direction cell preferred firing directions in a sighted recording session with visual input available to the rat, revealed that the preferred firing directions of the head direction cells drifted an average of 65.6° between the sighted and the blindfolded recording sessions. The head direction cell preferred firing directions did not drift during a recording session with visual input available to the rat. Thus, visual input calibrates the head direction cell system and promotes stability of the individual head direction cell preferred firing directions.

The association between a visual cue and the preferred firing directions of head direction cells can be established very quickly. Goodridge et al. (1998) demonstrated that an 8 minute exposure to a novel cue card is sufficient for the cue card to establish an influence over the preferred firing direc-

tions of head direction cells, such that a 90^{o} rotation of the cue card produces an average error of 19.2^{o} in the rotation of the preferred firing directions. Further to this point, Zugaro et al. (2003) report that visual cues can very quickly reset the head direction cell system. Using the standard experimental paradigm described above, but with the rat allowed to remain in the cylinder in the dark, i.e. with the lights off, as the cue card is rotated, they found that head direction cells updated their preferred firing directions as quickly as 80ms after a perceived change in the visual scene.

Thus, visual cues can exert an influence upon the preferred firing directions of head direction cells. This influence can be established within a matter of minutes and, once established, the resetting of the head direction cell system by visual cues occurs within tens of milliseconds after exposure to the visual cues.

1.1.2 The Differential Effect of Proximal vs. Distal Visual Cues

Proximal visual cues are defined as those cues that are relatively close, in space, to the observer. Conversely, distal visual cues are defined as those cues that are spatially relatively farther away from the observer. The influence that each class of visual cue has upon the processing occurring in putative neural navigation systems, and thus the behaviour of the animal, can change according to the current environmental conditions or information processing requirements.

There is much empirical behavioural evidence to suggest that when distal visual cues are placed in conflict with either proximal visual cues or internal mental representations of location, such that uncertainty about the current location increases, then rodents will preferentially use distal visual cues to successfully complete navigational tasks (Suzuki et al., 1980; Morris, 1981; Téroni et al., 1987;

Etienne et al., 1993; Alyan and Jander, 1994).

The relative salience of proximal and distal visual cues affects the macroscopic behaviour of a neural navigation system, and can also act at a more microscopic level by altering the response profiles of individual spatially-responsive neurons. For instance, Cressant et al. (1997) report that rotating a stimulus set of visual cues relative to the observer, a rat, will only produce an approximately equal rotation of the receptive-field orientation of place cells if a visual cue that is perceived as distal, relative to the observer, is included in the stimulus set of visual cues. The rotation of a stimulus set comprising solely proximal visual cues, as perceived by the observer, does not produce an approximately equal rotation of place cell receptive-fields.

Similarly, distal visual cues can exert a stronger influence upon the preferred firing directions of individual head direction cells than proximal visual cues. Zugaro et al. (2001) devised an experimental setup where the visual cues take the form of objects placed at the periphery of a raised platform. To begin, a cylinder encloses the platform such that a rat placed inside the cylinder cannot see anything beyond the cylinder walls. When the objects are rotated out-of-sight of the rat, the preferred firing directions of all of the 30 individual head direction cells recorded from the anterodorsal thalamic nucleus rotate by an approximately equal amount upon re-entry of the rat into the testing environment. When the cylinder is removed so that the rat can see more of the laboratory, then rotation of the set of objects does not produce rotation of the preferred firing directions of the head direction cells upon re-entry of the rat into the testing environment. The relative distance from the object set to the rat is the same regardless of whether a cylinder encloses the raised platform or not. The alteration in the

preferred firing directions of the head direction cells is occasioned solely by a change in the perceived distance of the objects relative to the rat, i.e. distal or proximal. All other properties of the set of objects remain constant across the two experimental conditions.

The Zugaro et al. (2001) study illustrates that a visual cue only has to vary in one dimension, the distance from the rat, to alter its efficacy in controlling the preferred firing directions of head direction cells. An important computational question is how the head direction cell system, or a pre-processing system afferent to the head direction cell system, can distinguish between proximal and distal visual cues on the basis of perceived distance alone.

Zugaro et al. (2004) conducted a study to investigate this question. Head direction cells were recorded from the anterodorsal thalamic nucleus. During the first part of the experiment a rat is allowed to explore an elevated platform in a cylindrical enclosure. There are two visual cue cards present in the recording environment: a small card in the foreground, and a large card in the background. The cue cards are proportionally dimensioned and subtend identical angles from the centre of the platform. Thus, the two cue cards appear equally salient in every respect apart from their relative distances to the platform. After the initial exposure to the recording environment, the rat is removed from the experimental setup and disoriented while the cue cards are rotated 90^{o} in opposite directions to one another. The rat is then re-introduced to the platform through forward motion in the hands of the experimenter. The head of the rat faces the midpoint between the two cue cards. This method of re-introducing the rat to the recording environment exposes the rat to both cue cards simultaneously and provides dynamic visual information about the cue cards during the period that the rat is in motion.

In 30 out of 53 (56.6%) trials, the preferred firing directions of the recorded head direction cells rotated to match the rotation of the distal background cue. Zugaro et al. also re-introduced the rat to the platform, after rotation of the cue cards, under stroboscopic lighting in an attempt to disturb the correct visual processing of image velocity. Under these conditions, it was found that only 17 out of 51 (33.3%) of the recorded head direction cells rotated their preferred firing directions to match the rotation of the distal cue card. Furthermore, under conditions of stroboscopic lighting it was found that the preferred firing directions of head direction cells were as equally likely to follow the distal cue card as the proximal cue card or a mixture of the cue cards.

A separate study by Yoganarasimha et al. (2006) also confirmed that, under conditions of cue conflict, head direction cells will preferentially anchor to distal visual cues rather than proximal visual cues. The experimental apparatus comprises a circular track divided into four segments, with each segment covered in a different coloured textured material. The track provides the proximal visual cues. Beyond the track, three objects are stood on the floor, and there are also three objects attached to a curtain completely surrounding the track. These six objects provide the distal visual cues. The experimental protocol consists of the rat exploring the track as head direction cells are recorded from the anterodorsal thalamic nucleus. The rat is then removed from the apparatus and the distal and proximal visual cues are rotated by a predefined amount in opposite directions to one another. The rat is re-introduced to the apparatus and further recording takes place. The rat is then removed from the apparatus for a second time and the distal and proximal visual cues are returned to their original positions. A further repeat of this protocol, with the angular distance of rotation of the visual cues being different, leads to

completion of the experimental task. The preferred firing directions of the head direction cells were shown to follow the rotation of the distal cues in 94% of the trials. (As opposed to the preferred firing directions of the head direction cells following the rotation of the proximal cues, a mixture of the rotation of the distal and proximal cues, or none of the cues.)

The preferred firing directions of individual head direction cells are influenced by the current location of visual cues. Moreover, the head direction system as a whole is able to discriminate between proximal and distal visual cues. There is strong evidence to suggest that the head direction cell system is able to perform this discrimination when the proximal and distal visual cues differ only in one dimension: their perceived relative distance to the observer. It may be the case that dynamic visual information is important if a system is to discriminate between distal and proximal visual cues on the basis of relative distance alone. This information may take the form of dynamic motion parallax, or optic field flow. What is evident is that the influence of visual cues upon the preferred firing directions of head directions, while expressed fairly quickly upon entry of the rat into a novel environment, does increase the greater the amount of time the rat spends in its current environment (Goodridge et al., 1998, see Section 1.1.1). Thus, the head direction cell system learns to be influenced by visual cues. Moreover, a necessary part of this learning is developing the ability to discriminate between proximal and distal visual cues.

1.2 The Self-Organization of Head Direction Cell Visual Responses

Within the literature on head direction cells there are no extant computational models, simulated or theoretical, that explicitly address the self-organization of head direction cell visual response prop-

erties. Specifically, none of the existing head direction cell models propose a theory of how head direction cells may come to be preferentially stimulated by distal visual cues, rather than proximal visual cues. Yet, there is now a large body of empirical data concerning the interactions between visual cues and head direction cell response profiles; particularly, the developmental stages of the rat at which these interactions begin to be expressed. These data are reviewed in the following sections.

1.2.1 The Development of Head Direction Cell Response Properties

Martin and Berthoz (2002) performed single-cell recordings in young rats. They found that around approximately postnatal day 30, cells in the cingulum begin to exhibit stable, adult-like head direction cell firing profiles, i.e. a single preferred directional response that is maintained across repeated visits to the same environment.

This result is confirmed by two recent studies of the development of spatial representation in young rats. Langston et al. (2010) discovered cells in the pre- and parasubiculum of young rats that showed adult-like head direction cell firing profiles at postnatal days 15 and 16. This result is especially interesting given that the eyelids of young rats do not generally unseal until approximately postnatal day 14 to 15. Thus, the stereotypical head direction cell firing profile appears in cells in the young rat after apparently very little visual experience of the world. The existence of cells in the young rat that exhibit adult-like head direction tuning is corroborated by Wills et al. (2010), who found adult firing profiles in rat medial entorhinal cortex head direction cells at postnatal day 16.

The data from the three studies suggest that either head direction cells have a genetic predisposition to express a tuning to a particular orientation of the head, or the head direction cell system is

capable of very fast learning that establishes stereotypical head direction cell response profiles in the naive cells.

It is not yet known, however, just how adult-like these head direction cells are. Although the head direction cells identified in the studies cited above undoubtedly exhibit adult-like firing profiles, insofar as the cells express a single preferred firing direction that is maintained across multiple visits to the same environment, these putative head direction cells were not tested for the presence of canonical head direction cell properties such as the influence of visual cues over the preferred firing direction. That is, the experimental protocol did not include tests such as the paradigmatic cue card rotation test described in Section 1.1.1. Thus, it remains an open question whether head direction cells in young rats exhibit a full range of adult-like properties almost immediately that the rat begins its visual experience of the world, or whether the development of mature stereotypical head direction cell response profiles occurs in a graded, more continuous manner.

A study by Save et al. (1998) examined the development of spatially-responsive neurons in rats deprived of visual input. In this study, hippocampal place cells were recorded in rats that were blinded before their eyes unsealed (the rats never had any visual experience of their world). Save et al. discovered that blind rats developed place cells which, across the majority of their firing properties, were indistinguishable from the place cells that develop in sighted rats. These stereotypical responses included the place cells in the blind rats rotating their firing fields to match the rotation of a set of cues in the testing environment, as is the case for place cells in sighted rats (Muller and Kubie, 1987). Although the study of Save et al. focused entirely on place cells, and did not address the development

of head direction cells, there are valuable inferences regarding the emergence of head direction cell responses that can be drawn from these results.

Tellingly, the place cells of the blind rats do not exhibit place fields immediately upon entry into the environment, as is the case for sighted rats. Rather, the place fields only develop after the blind rat has made physical contact with the cues in the environment. After physical contact has been made with the cues, the blind rats exhibit place cells with place fields that are evenly distributed across the environment. That is, some of the place cells develop place fields with x-y coordinates a distance away from the coordinates of any physical cue. Moreover, upon re-entry into the testing environment, and subsequent physical contact with the local cues, the place fields of the place cells of the blind rats re-establish themselves in the same locations, i.e. there is stability of the place fields across testing sessions.

In order to establish place fields that are evenly distributed across a given environment, and furthermore to re-establish these same place fields in the same locations upon re-entry into the environment, the blind rat must possess the ability to continuously track its current location. This ability is known as path integration, and will be discussed in detail in Chapter 5 of this thesis. Briefly, successful path integration of current location requires information about the current directional heading. For instance, knowing that a destination location is 10 metres North of the current location is only useful information if the current directional heading is also known. The current directional heading is exactly the type of information that is provided by the head direction cell system. Thus, the blind rats in the study of Save et al. (1998) were able to develop place cells with place fields distributed across

the environment, which in turn requires a functional head direction cell system. This result suggests that head direction cells can develop mature, stereotypical response properties with little, or no, visual experience. This conclusion concurs with the data from the studies of Langston et al. (2010) and Wills et al. (2010).

On the other hand, there is evidence to suggest that the full range of head direction cell response properties develops in a more graded and continual process. Schenk (1985) determined that young, sighted, rats 35 days postnatal could employ proximal visual cues to locate a platform in a water maze, but could not successfully employ distal visual cues to achieve the same result. A separate study by Rudy et al. (1987) showed that rats at 25 days postnatal exhibited minimal signs of employing distal cues to solve the water maze. This is in stark contrast to results from mature rats who preferentially use distal visual cues, rather than proximal visual cues, to locate the hidden platform (Morris, 1981). Further to this point, the studies discussed in Section 1.1.2 (e.g. Zugaro et al., 2001, 2004; Yoganarasimha et al., 2006) argue that adult rats will anchor the head direction cell system to cues that are perceived as distal relative to the rat. These results imply two important points: that rats are capable of discriminating between distal and proximal visual cues; and that this discriminative ability appears to be a learned ability.

It is possible that the failure of young rats to employ distal visual cues may be more the result of an immature visual system than incomplete spatial learning. Rudy et al. (1987) tested this particular hypothesis using a Morris water-maze. A single cup was suspended above the water-maze. In some of the experimental conditions, the cup was directly above the hidden platform and thus served as a

proximal (in relation to the hidden platform) visual cue to the platform location. In other experimental conditions, the cup was suspended above a quadrant of the water-maze adjacent to the hidden platform, and the cup served as a distal visual cue to the platform location. It was discovered that rats at least 18 days postnatal could use the cup to guide themselves to the hidden platform when the cup served as a proximal cue to the platform location, and these rats could find the hidden platform faster than a control group in which the was no regular relationship between the location of the cup and the location of the hidden platform. Rudy et al. further discovered that it is only when rats reach at least 22 days postnatal that they can successfully use the cup as a distal cue to guide them to the hidden platform.

Thus, the younger rats were capable of using the cup as a proximal cue to the hidden platform location. Yet, it is only when the rats are a few days older that they are capable of using the same cup as a distal visual cue to location of the hidden platform. This result suggests that the failure of younger rats to use the cup as a distal visual cue is not the result of an immature visual system, because the younger rats could easily use the same cup as a proximal cue to the location of the hidden platform, i.e. the younger rats could see the cup perfectly well. Rather, it appears that the ability to use the cup as a distal visual cue to the location of the hidden platform does not develop until the rat is a few days older. Consequently, the ability to discriminate between proximal and distal visual cues, and to preferentially anchor the head direction system to distal visual cues, appears to be a learned ability.

There are therefore two potentially conflicting theories concerning the development of head direction cell response profiles: that head direction cells can develop adult-like firing properties in the

absence of visual experience; and that head direction cells require visual experience to exhibit the full range of adult response properties.

A resolution of this theoretical conflict may only be possible with further single-cell recording research. In particular, it would be highly informative to test rats at the same postnatal age as in the studies of Langston et al. (2010) and Wills et al. (2010), and record from head direction cells as the rats undergo both the paradigmatic cue card rotation experiment (e.g. Taube et al., 1990b), and the tests of the relative influence of distal and proximal visual cues upon the preferred firing directions of the head direction cells (Zugaro et al., 2001, 2004; Yoganarasimha et al., 2006). If the data from the young rats replicate the results from these studies then there would be strong reason to believe that the visual response properties of head direction cells develop very quickly postnatal, without visual experience. Otherwise, it would appear that young rats need visual experience within their environment in order for the mature visual response properties of head direction cells to develop fully.

At present, in the absence of further empirical research, there is good reason to favour the theory that the full range of head direction cell visual response properties develops through visually-guided learning. The empirical data reviewed in the next section suggest the stereotypical visual responses of head direction cells, even in adult rats, only develop after statistically regular visual experience within an environment.

1.2.2 The Visual Response Properties of Mature Head Direction Cells

Knierim et al. (1995) trained rats to explore an experimental setup very similar to the one depicted in Figure 1.1. The rats were split into two groups: one group was disoriented before each training session

in the cylinder in order to stop a path integration mechanism from providing a stable directional frame of reference; the second group was not disoriented between training sessions in the cylinder. During testing, the conditions were identical for both groups: the rats were disoriented and then placed into the cylinder for a head-direction cell recording session. Between some of the recording sessions, the visually distinctive cue card on the cylinder wall was rotated through a pre-determined angle. Upon re-introduction to the cylinder, all rats were tested to see whether the rotation of the cue card would result in the paradigmatic corresponding shift in the preferred firing directions of the head direction cells (Taube et al., 1990b). It was found that the head direction cells of rats that had undergone disorientation during training were far less likely to follow the cue card when it was rotated during the testing sessions than the head direction cells of rats who had not undergone any disorientation during training.

From this result, Knierim et al. (1995) hypothesized that only those visual cues that appear stable and coherent, i.e. statistically regular, to the rat would develop any influence upon the preferred firing directions of the individual head direction cells. In other words, associations are learned between entities that regularly co-occur. The more frequent the co-occurrence, the stronger the learned association. When the rat is allowed to maintain a stable directional frame of reference between experimental sessions, a strong association can be learned between a visual cue card and the internal sense of direction. This association will allow rotation of the cue card to update the internal sense of direction. Head direction cells will therefore rotate their preferred firing directions to match the rotation of the cue card. When the rat is disoriented between recording sessions, it is not possible to maintain a stable directional frame of reference and the resulting learned associations between the cue card and the internal sense of direction are very weak, if at all existent. This explains the differential amount

of influence that the cue card has on the preferred firing directions of head direction cells in rats in the two experimental conditions: disoriented vs. non-disoriented. If the visual response properties of head direction cells are not learned, i.e. are somehow genetically specified, then there should be no difference between the experimental conditions in the study of Knierim et al. (1995).

Further support for the theory that head direction cells only develop their full response properties in the presence of visual experience comes from a study by Goodridge et al. (1998). In this study, the rat was presented with a novel visual cue that was then rotated after the rat had a pre-determined amount of experience with the cue. It was found that the longer the experience a rat has with the visual cue, the more control that cue exerts over the preferred firing directions of individual head direction cells. In particular, an exposure time of eight minutes was found to be sufficient for the novel visual cue to develop reliable control of the head direction cell preferred firing directions (nearly all of the head direction cells tested rotated their preferred firing directions to march the rotation of the cue card). This is contrasted with an exposure time of a single minute, after which approximately only half of the recorded head direction cells rotated their preferred firing directions to match the rotation of the visual cue card. Again, if the visual response properties of head direction cells are not learned, then the amount of temporal exposure a rat has to a visual cue card should not have any relationship with the degree of control that the cue card has over the preferred firing directions of individual head direction cells.

The studies of Knierim et al. (1995) and Goodridge et al. (1998) suggest that the visual response properties of mature head direction cells still undergo a continual process of learning and refinement.

This hypothesis is in contrast to the hypothesis that head direction cell visual response properties are genetically pre-specified and therefore not learned. Further to this point, the ability to discriminate between distal and proximal visual cues must also be a learned ability. Firstly, it is hard to see how such a discriminative ability could be genetically pre-specified, and particularly how the genetic pre-specification hypothesis could account for the result of the study by Zugaro et al. (2001), where a change in the environmental conditions, rather than a change in the relative distance of the visual cues to the observer, alters whether the cues are perceived as distal or proximal. Second, the learning hypothesis *can* account for the ability to discriminate between distal and proximal visual cues. As will be discussed in the next section, distal visual cues, compared to proximal visual cues, present different degrees of statistical regularity to the observer. It is the difference in the degree of this regularity that facilitates the discrimination between distal and proximal visual cues.

1.3 Competitive Learning and Visual Processing

A competitive learning mechanism categorizes input stimuli into different output categories. An example of such categorization is the separation of distal and proximal visual cues into distinct representations. In this section, the process of competitive learning is first outlined. The application of competitive learning to visual processing is then reviewed. It is argued that a competitive learning mechanism is the basis for the differential responses of head direction cells to distal and proximal visual cues.

1.3.1 The Principles of Competitive Learning

A neural network that incorporates competitive learning will self-organize in such a manner that, after training has occurred, each individual input in the set of network input stimuli will preferentially stimulate one, or a small subset, of the output cells in the network (Rumelhart and Zipser, 1985; Hertz et al., 1991; Rolls and Treves, 1998; Dayan and Abbott, 2001). After competitive learning has taken place, inputs to the network that are similar to one another will be responded to by the same output cell, or subset of output cells. Inputs to the network that are sufficiently dissimilar to one another will be responded to by different output cells, or subsets of output cells. In a purely "winner-take-all" competitive network, only one output cell will be active, after training, for each unique input. In a competitive network implementing a more graded level of competition, a subset of output cells may be active for each unique input.

The simplest architecture of a neural network that incorporates competitive Hebbian learning involves a single layer of input cells projecting, in a feedforward manner, to a single layer of output cells. The competitive learning mechanism functions as follows. Synaptic drive from the layer of input cells to the layer of output cells will, through an initially random distribution of the feedforward synaptic connection strengths from the input cells to the output cells, produce varying levels of activation in the output cells. Within the layer of output cells there is mutual lateral inhibition such that each output cell sends efferent inhibitory signals to, and receives afferent inhibitory signals from, every other output cell in the layer. The effect of this mutual lateral inhibition is that output cells that are more strongly active will more strongly inhibit other output cells in the layer. Consequently, only one, or a small subset, of the output cells will have a high enough activation level to commence firing. Only those

output cells that are firing are permitted to update the strengths of their synaptic connections with the currently firing input cells. The result of the competitive learning process is that when a previously seen input is presented to the network again, the input cells representing that particular input stimulus should preferentially stimulate the same output cells that were active the previous times that particular input was presented to the network. In other words, the output layer cells that learn to become active when a particular input is presented to the network have, during learning, competed for and won the right to represent that particular input.

An important feature of competitive learning is the ability to categorize input stimuli. Inputs that are similar to one another are highly likely to activate the same output layer cells. If these output cells proceed to win the competition to fire, they will be permitted to strengthen their synaptic connections from the similar input. Thus, similar inputs will, through competitive Hebbian learning, be represented by the same output cells. In this manner a competitive network performs categorization of the inputs it processes, with similar inputs being represented by the same output cells. Inputs that are dissimilar will be represented by different output cells. Furthermore, an output layer cell that exhibits a learned response will represent the average of the inputs that the output cell has learned to respond to. That is, the synaptic weight vector for that output layer cell will move towards the average of the inputs that the cell has learned to represent.

The competitive learning process is illustrated schematically in Figure 1.2. The left-hand plot represents the untrained state of the network. The grey circles are inputs to the network, and the black crosses represent initial values for the synaptic weight vectors of three output layer cells. The right-

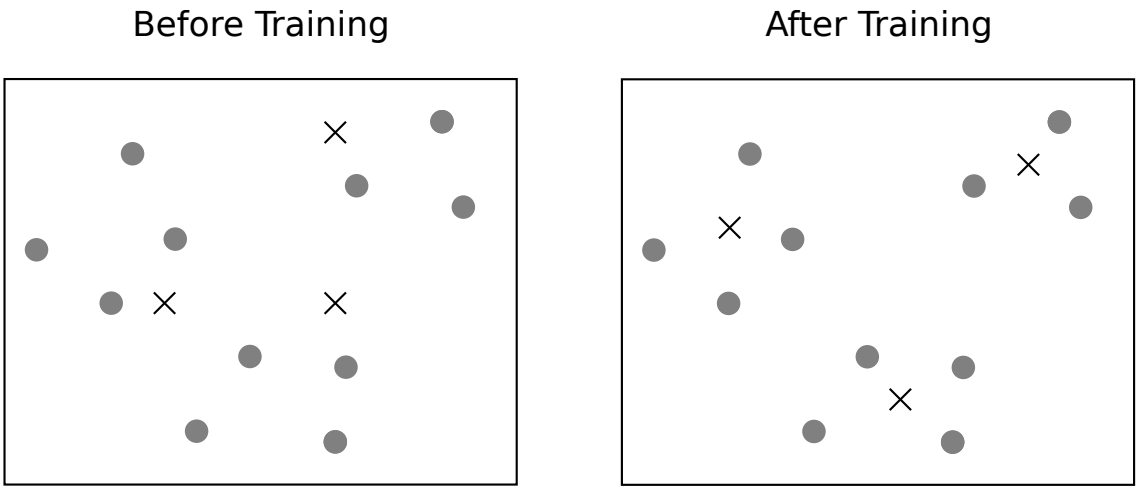

Figure 1.2: A schematic illustration of competitive learning. The left-hand plot displays the state of the network before competitive learning occurs. The grey circles represent separate inputs in the input space of the network. The black crosses represent the weight vectors of three output cells. The right-hand plot displays the states of the network after training. The effect of the competitive learning has been to shift the weight vectors of the three output cells such that each output layer cell now represents a particular cluster of inputs. The weight vector of each output cell is at the centre of the cluster of inputs that output cell represents. Figure adapted from Hertz et al. (1991).

hand plot represents the state of the network after competitive learning has occurred. Again, the grey circles represent the inputs to the network. The inputs maintain the same relationships to one another as in the untrained state of the network. The weight vectors of the three output layer cells have, through competitive learning, shifted to the centre of the three clusters of the inputs. That is, the three output layer cells have learned to each represent an individual cluster of inputs, where the within-cluster similarity of inputs is higher than the between-cluster similarity of inputs.

The number of output cells active for any given input, and the size of the cluster of inputs represented by any given output layer cell, i.e. how similar two inputs must be in order to be represented by the same output cell, are factors that are generally controlled by the amount of mutual lateral inhibition amongst the output cells. Another factor is the nature of the mechanism that is used to bound the synaptic weights of the output cells, such that no single output cell can come to dominate the

competition to fire and thus learn to represent all of the inputs.

It is not only the similarity between inputs that affects whether or not a given input will be represented by a given output cell, but also the frequency of presentation of a particular input. An input that is frequently presented to a competitive network has a high probability of being uniquely represented by an output cell, or subset of output cells, because the higher frequency of presentation affords a greater opportunity for output cells to strengthen their synapses with that particular input. Thus, two properties of any particular input that will have a big influence upon whether or not that input will come to be uniquely represented by an output cell, or subset of output cells, are the similarity of that input to any inputs previously presented to the network, and the frequency of presentation of the input. It is only in the state where distinct inputs have distinct output representations that the network can be said to have learned to discriminate between the inputs.

These general principles of competitive learning can produce a diverse set of behaviours within competitive neural networks. The architecture of the neural network, and the competitive learning algorithm itself, can be changed to produce different behaviours as required. The competitive learning paradigm is thus fairly extensible. One application of competitive learning that is pertinent to this thesis is the categorization of visual cues such that the visual cues can be discriminated based upon which category a particular cue has been placed into.

1.3.2 Applications of Competitive Learning to Visual Processing

A competitive learning paradigm has been applied to solve many problems of visual processing (Rolls and Deco, 2002). Some of these problems include learning to recognise an object invariantly: that is,

learning to recognise a particular object regardless of the current location, size, or view of that object (Wallis and Rolls, 1997). It is also possible to employ competitive learning to solve the problem of invariant object recognition under different lighting conditions (Rolls and Stringer, 2006), and to solve the problem of invariant object recognition of 3D stimuli (Stringer et al., 2006). These learning problems are relevant to the head direction cell system because the reliable control, by visual cues, of head direction cell preferred directions requires that the head direction system is capable of invariantly recognising the visual cues under a variety of views, lighting conditions, rotations etc. These previous models, however, always present the to-be-learned objects to the network one-at-a-time. An ecologically more valid training paradigm involves multiple visual cues, many of which are simultaneously present in the field-of-view of the observer at any given instant in time.

Stringer and Rolls (2008) demonstrated that a single-layer feedforward competitive network can learn representations of individual stimuli from a stimulus set when, during training, the stimuli were always presented to the network two-at-a-time. The primary result from this particular study is that, when the input stimuli are presented to the network simultaneously, the size of the input stimulus set determines the nature of the learned representation of those input stimuli. In particular, it was demonstrated that when $N < 6$, where N is the number of stimuli in the input stimulus set, then individual output cells will learn to respond to pairs of the input stimuli when those stimuli are presented to the network two-at-a-time. As the number of stimuli in the input stimulus set increases such that $N \geq 6$, then individual output cells learn to respond exclusively to just one of the input stimuli.

In a similar study, Stringer et al. (2007) showed that a single-layer feedforward competitive network

can learn representations of individual stimuli when those stimuli were always presented to the network three-at-a-time during training. When $N < 10$, where N is again the number of input stimuli in the stimulus set, the output cells of the network learned to represent combinations of three stimuli. When $N \approx 10$, the output cells represented combinations of two stimuli. For $N \approx 20$, the output cells developed representations of individual stimuli. Thus, similar to the study of Stringer and Rolls (2008), the main result of the Stringer et al. (2007) study is that as the number, N, of input stimuli in the stimulus set increases, so the nature of the learned representation in the output cells changes from learning to represent combinations of the input stimuli to representing individual input stimuli exclusively.

In the studies of Stringer and Rolls (2008), and Stringer et al. (2007), the reason for the change, as N increases, in the nature of the learned representation of the input stimuli is a consequence of the frequency of co-activation of the neurons representing the input stimuli during a training epoch. If a training epoch comprises one presentation to the network of all possible pairings of the input stimuli, then input cells that represent an individual input stimulus will be co-active whenever that stimulus is presented to the network, which is $N - 1$ times. These same cells, however, will be co-active far less frequently with cells representing a different, independent stimulus. In the case where the stimuli are presented to the network two-at-a-time, these cells will be co-active just one time per training epoch with the cells representing any other independent stimulus. As competitive networks essentially categorize the network inputs based on the correlations between inputs to the network, those input cells that are frequently co-active will be placed into one category and represented by a unique subset of the network output cells. Those cells that are not frequently co-active will be placed into different cat-

egories, with each category represented by a unique subset of the output cells. Thus, as the size of the stimulus set, N, increases, so the correlation between input cells representing different input stimuli decreases, such that the only input cells with which a particular input cell will be regularly co-active will be those input cells that represent the same individual stimulus as the input cell in question. It is this decrease in the correlation between input cells that explains why the nature of the output cell representation changes, as N increases, from learning to represent combinations of the input stimuli to developing exclusive representations of individual stimuli. The same learned behaviour is produced when the input stimuli are presented three-at-a-time.

Extending this research, Bennett and Stringer (2011) have shown that with an input stimulus set of just $N = 2$ stimuli, and with the stimuli always presented simultaneously, it is possible for a single-layer feedforward competitive network to learn separate representations of the two stimuli. The condition for separate representations to develop is that the two stimuli must move independently of one another. That is, the two stimuli must be decorrelated in some way in order for the competitive learning mechanism to form separate representations of each stimulus. One method of achieving this decorrelation is for the two stimuli to move independently, i.e. rotate in opposite directions. Bennett and Stringer showed that if the two stimuli move together, and are thus highly correlated, then the output cells in the competitive network will learn to represent a combination of the two stimuli. It is only when the stimuli rotate in different directions that the correlation between the stimuli decreases sufficiently for the competitive learning mechanism to form separate output cell representations of the two stimuli.

The three studies cited above model the development of visual processing in the primate ventral visual

stream. The computational principles demonstrated by this research are general principles, however, and are equally applicable to the visually-guided development of the rat head direction cell system.

The differential effect of distal and proximal visual cues upon head direction cell preferred firing directions is due to a decorrelation between the distal and proximal visual cues that occurs as the rat moves through its environment. This decorrelation allows a competitive learning mechanism to develop separate representations of the distal and proximal visual cues. In particular, distal visual cues will remain stable and coherent, in comparison to proximal visual cues, as the rat navigates through its environment. If the distal visual cues are far away from the rat then whenever the rat has the same particular orientation at any given location in its current environment, the rat will see the same distal cues in the same position on the retina. Because cells in the early visual processing areas of the brain have relatively small receptive field sizes, these cells, as a retinotopic map, have a relatively high spatial resolution. Thus, visual input that consistently appears in the same position on the retina will consistently activate the same cluster of input cells.

The view of the proximal cues from a particular orientation will, however, differ according to the current location of the rat. That is, whenever the rat has the same particular orientation at any given location, the egocentric bearing to a particular proximal object will be different in different locations. Thus, given a particular orientation of the rat, a proximal object will appear in different positions on the retina for different locations in the environment. Each of these different retinal positions will activate a different cluster of input cells.

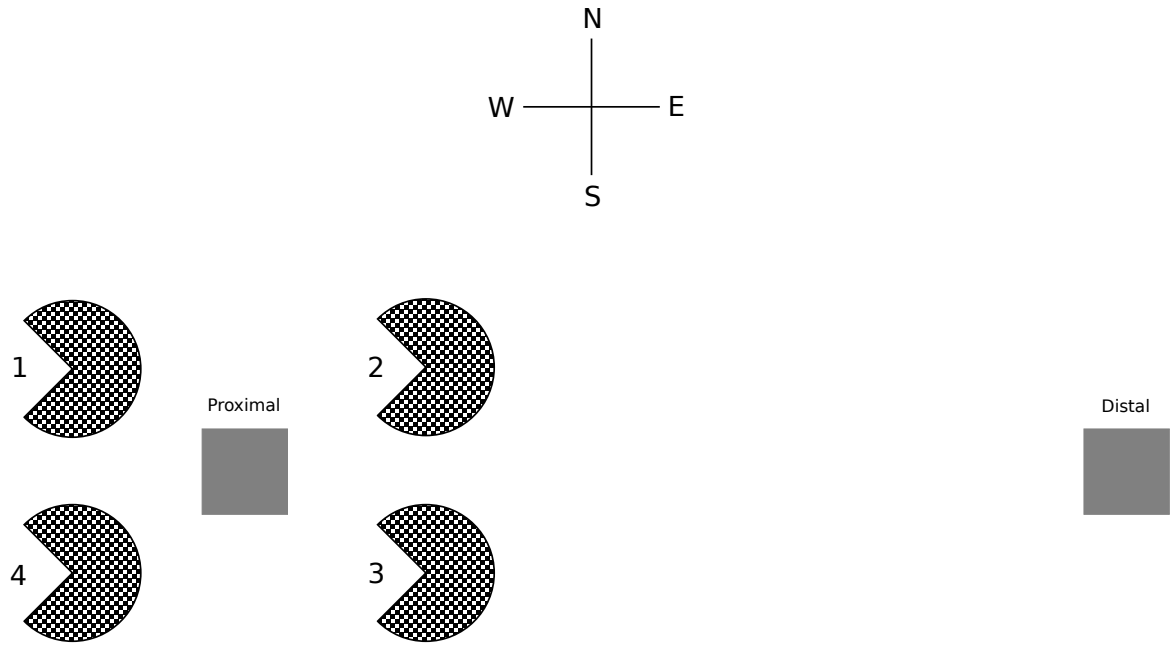

Figure 1.3: A schematic representation of the different visual input provided by distal and proximal visual cues. The figure depicts a top-down view of a training environment. The environment contains two grey cubes labelled distal and proximal. There are also four training locations in the environment, labelled $1-4$. When an observer visits each of the training locations and assumes the same orientation at each location, represented in the figure as due East, the observer will receive a particular view of the training environment on a particular part of the retina. The field-of-view at each training location is represented by the checker-board pattern. In the figure above, the view of the proximal object is different at each training location and falls on a different part of the retina, whereas the view of the distal object is the same and falls on the same part of the retina. The same view at the same position on the retina will excite the same input cells, but a different view falling on different positions on the retina will excite different input cells. This decorrelation between the currently active input cells representing the distal and proximal visual cues permits a competitive learning mechanism to form separate representations of the distal and proximal visual cues.

This phenomenon is shown schematically in Figure 1.3. The figure presents a top-down view of a training environment. There are two grey cubes labelled distal and proximal, and four training locations labelled $1-4$. When the rat has the same particular orientation at each of the training locations – in the figure this is due East – then the rat will receive a particular view of the training environment at each training location. The field-of-view of the rat at each training location is represented by the checker-board pattern. It can be seen from the figure that the rat receives approximately the same view of the distal object at each of the training locations. That is, the distal object will appear on the same part of the retina. This is not the case for the proximal object, where a different view of the proximal

object is seen on a different part of the retina at each of the training locations.

The input cells representing the particular view of the distal object at a particular position on the retina will always be co-active, therefore their firing rates will be highly correlated with one another. For each location in the environment, the input cells representing the view of the proximal object will also be co-active with one another, and thus the firing rates of these input cells will be highly correlated. There will, however, be a high decorrelation between the firing of the input cells representing the distal object at any particular position on the retina, and the input cells representing the proximal object at any other position on the retina. It is this decorrelation that permits a competitive learning mechanism to develop separate representations of the distal and proximal objects, even when they are simultaneously present in the field-of-view of the rat.

This computational principle holds true when there are multiple distal and proximal objects simultaneously present in the environment. Thus, a competitive network will be able to form unique separate representations for each of the distal and proximal objects that are present in an environment. Once the distal and proximal objects have been separated by the competitive learning process, it then becomes possible for distal visual cues to develop a strong influence over the preferred firing directions of head direction cells.

1.4 The Isomapping of Head Direction Cell Preferred Firing Directions

A well-established property of a population of head direction cells is that the angular distance between the preferred firing directions of the individual head direction cells is maintained across different en-

vironments and environmental manipulations (Taube et al., 1990b; Taube and Burton, 1995; Zugaro et al., 2004; Yoganarasimha et al., 2006). Although individual head direction cells may alter their preferred firing directions upon entry into a different environment, or after an environmental manipulation has taken place, the preferred firing directions of the population of head direction cells as a whole will remain in register with one another. Thus, if the preferred firing directions of a population of head direction cells are known, and some sort of of environmental change or manipulation occurs, then knowledge of the new preferred firing direction of just one of these head direction cells will allow for the prediction of the new preferred firing directions of all the other head direction cells within the population. This property is referred to as an isomapping of head direction cell preferred firing directions across environments and environmental manipulations.

Isomapping is shown schematically in Figure 1.4. Three polar plots display the responses of two head direction cells across two different environments. The plots are abstract representations of head direction cell behaviour, and do not display data from real head direction cells. The head direction cells change their preferred firing directions in the two different environments, and entry into a previously visited environment will recall the previous preferred firing direction in that environment. The angular distance between the preferred firing directions of the two head direction cells is, however, preserved across the two environments. In this case, the angular distance is 90^{o}. This is the isomapping of preferred firing directions across environments.

This is in stark contrast to the behaviour of the locational firing fields of a population of hippocampal place cells, which exhibit complex remappings as a result of environmental and behavioural manip-

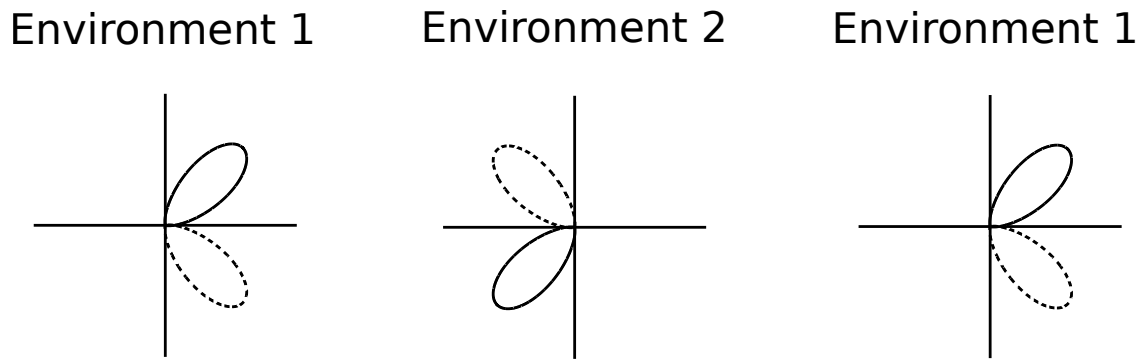

Figure 1.4: The isomapping of head direction cell preferred firing directions across environments. The polar plots are abstract representations of head direction cell behaviour and do not display real head direction cell data. The unbroken line represents the preferred firing direction of one head direction cell. The dashed line represents the preferred firing direction of a second head direction cell. As the environment changes from the first environment to the second, and back again, it can be seen that although the preferred firing directions of the head direction cells change across environments, the angular distance between the preferred firing directions is maintained across different environments. In this example, there is an angular distance of 90^{o} between the preferred firing directions of the two head direction cells across the two environments. This is referred to as the isomapping of head direction cell preferred firing directions across environments.

ulations (Muller and Kubie, 1987; Bostock et al., 1991; Markus et al., 1995). The distances between the locations of the firing fields of individual place cells will not necessarily be preserved across either changes in the environment, or the current task the rat is performing. Two place cells with firing fields located close to one another in one environmental condition may exhibit firing fields far away from one another under a different environmental condition. It is also possible that both place cells may be silent under the new condition, or one place cell may fire but the other place cell may be silent. Consequently, if the locations of the firing fields of a population of place cells are known under one environmental condition, and some sort of change to the environmental condition takes place, then knowledge of the new location of a single place cell locational firing field under the new condition would not allow one to predict the location of the firing fields of the remaining place cells in the population.

Head direction cells and place cells therefore display different mapping behaviours across differ-

ent environments. Although there is plenty of evidence to suggest that the orientation of the firing profiles of head direction cells and place cells are often tightly coupled (Knierim et al., 1995, 1998; Yoganarasimha and Knierim, 2005), it has also been demonstrated that the firing profiles of head direction cells and place cells, when recorded simultaneously, can become decoupled from one another when a set of distal visual cues and a set of proximal visual cues are rotated in opposite directions to one another. After this cue manipulation, the head direction cell preferred firing direction displayed an isomapping, yet the place cell preferred firing locations displayed a complex remapping (Yoganarasimha et al., 2006).

An interesting question to ask, therefore, is how a population of head direction cells may develop an isomapping of their preferred firing directions across different environments. One possible method of achieving the isomapping is through the addition of excitatory recurrent collateral synapses between the head direction cells. Under this neural architecture, each individual head direction cell in a population of head direction cells will send efferent synapses to, and receive afferent synapses from, every other head direction cell in the population. The hypothesis is that in the first environment experienced by the rat, individual head direction cells within a population will develop unique preferred firing directions. The presence of the recurrent collateral synapses will ensure that head direction cells with preferred firing directions close to one another, in terms of angular distance, will develop strong recurrent collateral synapses with one another. In effect, the strength of the recurrent collateral synapse between any two head direction cells is a function of the angular distance between the preferred firing directions of those two head direction cells. As the angular distance increases, so the strength of the synaptic connection decreases.

When the rat enters a new environment, the head direction cells will initially be randomly stimulated to exhibit new preferred firing directions in the new environment. The presence of the (now trained) recurrent collateral synapses will ensure that when any particular head direction cell is stimulated into firing, this cell will in turn stimulate its neighbour cells, in terms of the angular distance between preferred firing directions, into firing. Thus, head direction cells with a small angular distance between their preferred firing directions in one environment will exhibit the same angular distance between their preferred firing directions in a different environment. Consequently, an isomapping of the head direction cell preferred firing directions will develop. In other words, the excitatory recurrent collateral synapses help to shape the activity profile that develops in the layer of head direction cells for each particular orientation of the rat. This, in turn, strongly influences the learning of the feedforward synaptic weight matrix between the input cells and the head direction cells.

The excitatory recurrent collateral synapses, in conjunction with a competitive learning mechanism, ensure that there is a single persistent packet of neural activity in the head direction cell layer at each orientation of the rat. This packet of activity will be centred on a particular head direction cell representing the current head orientation of the rat, but will also include a small number of head direction cells representing head orientations close, in terms of angular distance, to the central head orientation. Only those postsynaptic head direction cells that are currently firing will be permitted to strengthen their synapses from the currently firing presynaptic input cells; the currently firing head direction cells are precisely those head direction cells that are currently contributing to the activity packet in the head direction cell layer. Thus, the firing rates of the currently firing head direction cells will be highly

correlated with one another because head orientations that are spatially close to one another tend to also temporally co-occur. Similarly, the firing rates of the currently firing input cells, representing visual input at a particular position on the retina, will be highly correlated.

The high correlation between the firing of head direction cells with preferred firing directions close to one another is reflected by strong recurrent collateral synapses between those head direction cells. The firing of one of these head direction cells will, through the recurrent collateral synapses, induce firing in the small number of head direction cells representing nearby head orientations. Consequently, head direction cells that fire together in one environment are highly likely to fire together in a second environment. Associative Hebbian learning will strengthen the synapses from the currently active input cells to the currently active head direction cells. This is how the isomapping across environments of head direction cell preferred firing directions is learned.

If a neural network does not contain excitatory recurrent collateral synapses in the head direction cell layer, that model cannot learn an isomapping of head direction cell preferred firing directions. Without the recurrent collateral synapses, there is no method by which two head direction cells that fire together can become associated with one another. Consequently, the future firing of one of these head direction cells will not induce firing in the other head direction cell. In the absence of head direction cell co-firing, it is not possible to maintain an isomapping of head direction cell preferred firing directions.

Thus, a simple single-layer feedforward competitive neural network with excitatory recurrent col-

lateral synapses amongst the output cells will be able to develop an isomapping of head direction cell preferred firing directions across different environments. It is hypothesized that the strength of these recurrent collateral synapses will influence the nature of the cell response profile mapping that occurs across environments. With a relatively strong recurrent collateral synaptic input, it is predicted that an isomapping of response profiles will occur. When the recurrent collateral synaptic input is relatively weak, it is predicted that a complex remapping of cell response profiles will occur, similar to the remapping observed in hippocampal place cells (Muller and Kubie, 1987; Bostock et al., 1991; Markus et al., 1995).

1.5 Determining the Appropriate Level of Modelling

An important factor in the design process of a computational modelling study is the level of detail at which the underlying system, in this case the brain, is modelled. A high-level abstract description of a system, if not chosen carefully, can bear little correspondence to the real-world system being modelled. Many real-world systems are inherently complex, comprising many interacting parts and processes across a range of macro- and microscopic scales. Yet, a large amount of unnecessary detail may compromise the explanatory power of any model.

Another important design factor is that all of the models presented in this thesis are self-organizing. That is, the majority of the synapses in these models are updated through learning. As the models often feature full synaptic connectivity between two layers of cells, i.e. a single postsynaptic cell receives an afferent synaptic connection from each presynaptic cell, there are many synapses to update. For instance, in a recurrently connected network, where every cell is connected to every other

cell, the number of synapses grows quadratically with the number of cells in the network. Thus, the more synapses that require updating in a model, the more computationally expensive the model becomes. Modifiable synapses are of primary importance to the models presented in this thesis, and the computational expense of these synapses therefore greatly informs the choices made concerning the appropriate level of detail at which to model head direction cells.

Hodgkin and Huxley (1952) proposed a system of nonlinear differential equations that describe the time course of the generation of an action potential in a neuron. The research of Hodgkin and Huxley has become the defining model of action potential generation. A neural network containing neurons with behaviour described by the Hodgkin-Huxley model would, however, be very computationally expensive, particularly if the model also contains self-organizing synapses. It is for this reason that the Hodgkin-Huxley equations are too detailed to be implemented for the models in this thesis.

An approach to modelling neurons that is less detailed than the Hodgkin-Huxley model is that of integrate-and-fire neurons (Lapicque, 1907; Koch, 1999; Dayan and Abbott, 2001). In an integrate-and-fire model, the emission of an action potential by a neuron is treated as a discrete event, and thus the time course of action potential generation is not modelled. This abstraction affords a reduction in the complexity of the model. The exact level of detail can vary from high-level model neurons that simply sum the action potentials they have received over a moving window of time, to more complex conductance-based model neurons that approach the Hodgkin-Huxley model in their inclusion of the effect of ion channels upon action potential generation (Koch, 1999; Dayan and Abbott, 2001). With the computational resources available to the author at the time that research was commenced for

this thesis, a neural network of integrate-and-fire neurons connected by self-organizing synapses was, however, too computationally expensive to implement in terms of the simulation run-time.

One of the most abstract representations of a neuron is the simple firing-rate based model (Rumelhart and Zipser, 1985; Rolls and Treves, 1998; Dayan and Abbott, 2001). In this type of model, neurons do not emit individual action potentials. Instead, the instantaneous average firing rate of the cell is simulated. The simplicity of a firing-rate based model leads to a reduction in the number of equations required to specify the behaviour of each individual neuron in the model (in comparison with either the Hodgkin-Huxley model, or an integrate-and-fire model). Consequently, firing-rate based models are computationally cheaper to simulate. This is particularly appealing when the neural network contains self-organizing synapses.

Further to this point, the models described in this thesis employ a competitive learning mechanism that requires many $(50 - 100)$ epochs of training in order for the synaptic weights to converge upon stable values. To keep the simulation run-time reasonable thus requires that the run-time of a single epoch is kept as short as possible.

The models described in Chapters 2, 3, and 4 therefore adopt a firing-rate based modelling approach. The firing-rate based approach allows for a detailed exploration of the computations that develop in a self-organizing neural network, whilst helping to keep the computational expense, and thus simulation run-time, of the model at an acceptable level.

1.6 Chapter Summary

This chapter reviewed the literature on head direction cell visual response properties. It was argued that the majority of the head direction cell visual responses are *learned* by the head direction cell system, and that a competitive learning mechanism provides the correct method by which these visual response properties can be learned. To summarize the main points of this chapter:

- The preferred firing directions of head direction cells can be influenced by visual cues. Rotation of a visual cue can induce rotation of the preferred firing directions by an amount approximately equal to the magnitude of the cue rotation. If the cue card is rotated back to its original position, the preferred firing directions of the head direction cells will also rotate back to their original preferred directions. The influence of visual cues is a reliable effect that has been replicated by many different research laboratories.

- Not all visual cues have the same magnitude of influence upon the preferred firing directions of head direction cells. In particular, distal visual cues, which are relatively far away from the observer, tend to have a stronger influence over preferred firing directions than proximal visual cues, which are relatively close to the observer.

- There is some debate about the degree to which head direction cell responses are learned. There is compelling evidence that head direction cells exhibit adult-like responses very quickly after the rat open its eyes. There is, however, conflicting evidence that young rats do not display the full range of properties associated with head direction cells until further learning takes place. There is also compelling evidence that mature head direction cells still exhibit learning in novel environments. Further experimental work is required to resolve this debate, but it is highly

likely that head direction cell responses are, for the most part, learned responses.

- A plausible candidate for a learning mechanism that would produce the differential influence of distal and proximal cues upon head direction cells is competitive learning. Hebbian associative competitive learning is an unsupervised learning procedure through which inputs can be categorized on the basis of similarities and differences. Previous research has demonstrated that competitive learning can enable a network to form separate representations of visual cues that are presented simultaneously to the network, as long as the visual cues vary independently.

- Proximal and distal visual cues present different patterns of visual input at different positions on the retina. When a rat has the same orientation at different training locations in an environment, the visual input from distal visual cues will appear at the same position on the retina and will thus excite the same input cells. Proximal visual input will, however, appear at different positions on the retina and will therefore excite different input cells. There is thus a statistical decoupling between the distal visual input and the proximal visual input. A competitive learning mechanism can exploit this decoupling to from separate representations of distal and proximal visual cues.

- The angular distance between the preferred firing directions of any two head direction cells is maintained across different environments. This is called an isomapping of head direction cell preferred directions. It is proposed that a competitive network incorporating excitatory recurrent collateral synapses will produce isomapping in a head direction cell model.

- The appropriate level of model detail was discussed. It was recognised that while it is important to incorporate detail into the model, practical considerations such as simulation run-time are

also important factors in the model design process. A rate-coded modelling paradigm was selected as appropriate for the models in this thesis.

The visual response properties of head direction cells, in particular the control of head direction cell preferred firing directions by distal visual cues, are thus argued to be learned by the head direction cell system. In the coming chapters of this thesis, models that can learn to discriminate between distal and proximal visual cues are proposed in detail, and the results of simulating these models are presented and analysed.

Chapter 2

Head Direction Cell Visual Responses: Orthogonal Input Representations

This chapter provides the implementation details and equations for a model that can learn to discriminate between distal and proximal visual cues presented simultaneously to the model as input. The results of simulating the model, in a controlled manner, with a variety of model and training environment configurations, are given and discussed. These different configurations were carefully chosen to illustrate the operational principles of the model. This model is representative of the type of computation that is required in order to learn to discriminate between distal and proximal visual cues. As such, this model is presented as the minimal model sufficient to produce this learned behaviour. Future additions to the model are also considered.

2.1 Model Architecture

The general architecture of the model is displayed in Figure 2.1a. There is a single layer of input cells and a single layer of output cells. The input cell layer is connected to the output cell layer by

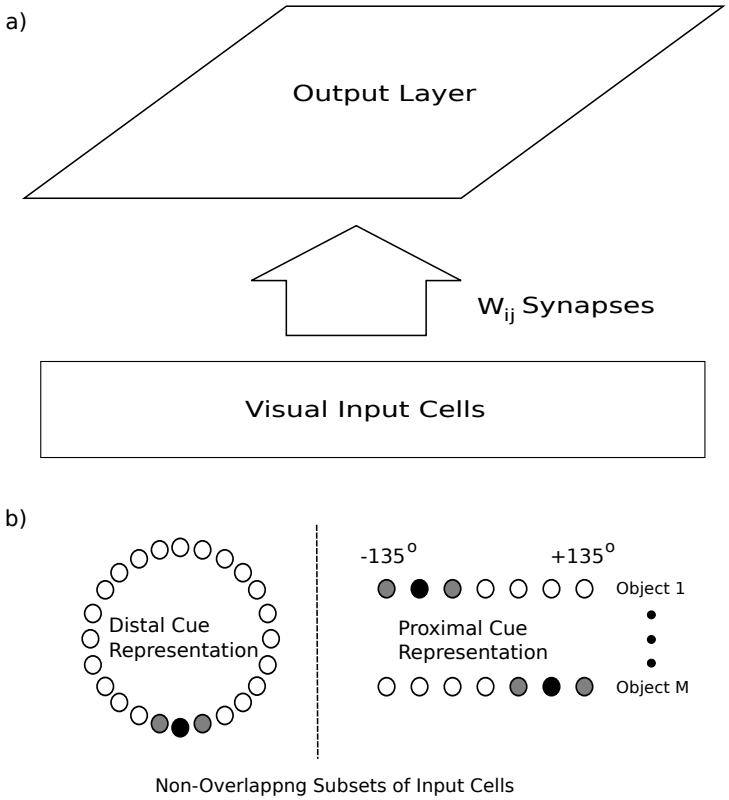

Figure 2.1: a) Neural network architecture of the simulated agent for the model of the development of head direction cell visual response properties with orthogonal input representations. The network comprises a layer of input cells that have feedforward synaptic connections to a single layer of output cells. The synaptic connections will self-organize during competitive learning as the agent explores its visual training environment. Under certain training conditions, the output layer cells are able to develop separate representations of the distal and proximal visual cues presented to the network as input. b) With non-overlapping, orthogonal subsets of input layer cells, there is one subset of input layer cells representing the distal visual cues, and M subsets representing each of the M proximal objects in the training environment. None of the input layer cells may contribute to more than one subset. The grey and black shading represents a Gaussian activity profile that specifies either a particular orientation of the agent, for those input cells assigned to represent the space of distal objects; or a particular egocentric head-centred bearing from the agent to one of the proximal objects, for those input layer cells assigned to represent that particular proximal object. The black shading represents a high firing rate, i.e. the centre of the Gaussian activity packet, and the grey shading represents a lower firing rate.

feedforward synaptic connections, where w_{ij} denotes the strength of the synaptic weight from presynaptic input layer cell j to postsynaptic output layer cell i. The synaptic weights w_{ij} will self-organize during competitive learning as the agent explores its visual training environment. During the training phase, the network receives visual input from a mixture of distal and proximal visual cues. Under the correct training conditions, the competitive learning mechanism will produce some output layer

cells that exhibit learned responses that are exclusive to the distal visual cues, and will produce some output layer cells that exhibit learned responses that are exclusive to the proximal visual cues.

The distal and proximal visual cues take the form of objects in the training environment. This is described in further detail in Section 2.2. The input cell layer is divided into $M+1$ subsets of N input layer cells.

One subset of input cells, shown on the left of Figure 2.1b, represents the visual input from the distal objects. The input cells in this subset form a closed, topologically organized, ring. Because the distal objects are relatively far from the simulated agent, the visual input from the distal objects depends only on the orientation, from $0^{\text{o}} - 360^{\text{o}}$, of the simulated agent within the environment. Consequently, each individual input cell in this ring responds maximally to a unique orientation of the agent within the training environment. During training, a Gaussian activity profile is imposed upon this ring of input cells, where the peak of the profile reflects the current orientation of the agent within the environment.

The remaining M subsets of input cells, shown on the right of Figure 2.1b, represent the visual input from the M proximal objects. Each of the M proximal objects is represented by a unique, topologically organized, subset of input layer cells. The simulated agent can move around, and in-between, the proximal objects, which are modelled as having the same visual appearance from any viewpoint. Given this assumption, the visual input from each proximal object depends only on the bearing from the simulated agent to the proximal object. The simulated agent is set to have the same 270^{o} field-

of-view as a rat, that is from -135° to $+135^{\circ}$ (Zugaro et al., 2004). Hence, each subset of input cells encodes the retinal position of one of the proximal objects within the 270° field-of-view of the simulated agent. Within a subset, each input cell is maximally responsive to a unique preferred retinal position of the proximal object. A Gaussian activity profile is imposed upon each subset of input cells such that the peak of the activity profile reflects the current retinal position of one of the M proximal objects.

The $M + 1$ subsets of N input layer cells are completely orthogonal and non-overlapping. This is shown in Figure 2.1b. Thus, a randomly chosen input layer cell j will be assigned to only one of the $M + 1$ subsets, and no single input layer cell subset will share any input layer cells with any other input layer cell subset. The total number of input layer cells is therefore equal to $(M + 1) \times N$.

2.2 The Training Environment

Both the distal and proximal visual cues take the form of objects in the training environment. Any particular training environment will consist of a number of training locations for the simulated agent, one or more proximal visual objects, and a continuous circular space of distal visual objects. All of the visual objects are implemented as symmetrical and will thus appear the same when viewed from any given angle. Figure 2.2 displays two example training environments. In Figure 2.2a there are four training locations, one centrally-located proximal object, and a circular space of distal objects. Figure 2.2b is similar, but instead contains four proximal objects. The circular space of distal objects is assumed to be at an infinite distance from the training locations and the proximal objects (although this is not metrically represented in Figure 2.2).

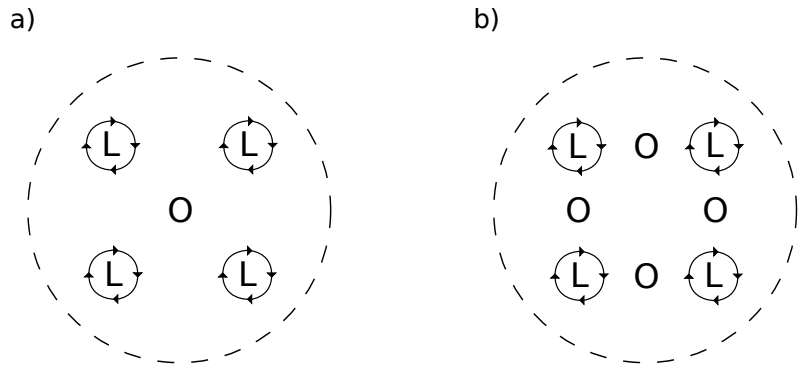

Figure 2.2: a) An example of the visual training environment. **L** = training location for the simulated agent. **O** = proximal object. The dashed line represents a continuous circular space of distal objects. In this example there are four training locations, a single centrally-located proximal object, and the space of distal objects. During a training epoch, the agent visits all the training locations, but each individual training location once only. At each training location the agent rotates through 360^{o} in single degree increments, and at each increment the current head-centred bearing from the agent to the proximal object is calculated. The head-centred bearing is then used to set the firing rates of the input layer cells that represent the proximal object. The distal object space is considered to be at an infinite distance from the training locations. Thus, the firing rates of the input layer cells representing the space of distal objects are set to match the current orientation of the agent regardless of the current training location. That is, the space of distal objects is assumed to be so far away that the agent will receive the same pattern of visual input whenever the agent has the same orientation regardless of the current training location. b) An example training environment with four training locations, four proximal objects, and one space of distal objects.

During a single epoch of training, the simulated agent visits every training location in the training environment once. At each training location, the agent rotates through 360^{o} in single degree increments. At each increment, the firing rates of those input layer cells assigned to represent the space of distal objects are set to match the current orientation of the agent, and the firing rates of those input layer cells assigned to represent a particular proximal object are set to match the current head-centred bearing of that proximal object.

For any given orientation, θ, of the agent at any given training location in the environment, the presynaptic input layer cells that are currently active propagate their firing through the w_{ij} synapses to the postsynaptic output layer cells. The firing rates of the output layer cells are calculated according to a

linear threshold function with the threshold iteratively adapted to maintain a desired level of sparseness of firing a (see Equation 2.5 later). The model described in this chapter is a rate-based model and thus simulates the instantaneous average firing rates of the cells, rather than the emission of action potentials by individual cells.

As each presynaptic input layer cell is fully connected through the w_{ij} synapses to every postsynaptic output layer cell, each individual postsynaptic output layer cell will initially receive varying levels of stimulation from *all* the currently active presynaptic input layer cells at any given training update. Thus, the computational problem is that the visual signals from the distal and proximal objects are completely combined in the untrained output layer cells. The theoretical challenge is therefore to explain how, as the agent visits each training location during learning, the network can self-organize its feedforward connection weights w_{ij} so that some of the output layer cells will exhibit learned responses that are exclusive to the distal visual objects, and some of the output layer cells will exhibit learned responses that are exclusive to the proximal visual objects. This problem is non-trivial because, as will be seen in Section 2.5, only under certain training conditions is the model able to develop learned responses in the output layer cells that can separate the distal and proximal visual objects presented to the network as input.

2.3 Model Operation

Building upon the argument developed in Chapter 1, this section will explore in detail how the model learns to discriminate between distal and proximal visual objects.

2.3.1 The Learning Paradigm

The model operates on the principle of unsupervised competitive learning, which is a learning paradigm that enables a neural network to perform categorization of the stimuli presented to it as input (Hertz et al., 1991). At any given update of the model during the training phase, only a small number of postsynaptic output layer cells are allowed to fire and therefore strengthen their w_{ij} synapses with the currently active presynaptic input layer cells through associative Hebbian learning. The number of output layer cells that are allowed to fire at any given model update depends on the threshold of the activation function for the output layer cells, which is iteratively adapted to maintain a pre-specified level of sparseness of firing a. This simulates the effect of inhibitory interneurons in the output layer of the model.

At any particular update of the model, the postsynaptic output layer cells that win the competition and are consequently allowed to fire will be those cells that receive the most stimulation, through the w_{ij} synapses, from the currently active presynaptic input layer cells. The active output layer cells will then have each of their w_{ij} synaptic weights strengthened in proportion to the product of the firing rates of the postsynaptic output layer cell i and the presynaptic input layer cell j. Thus, presynaptic input layer cells that are consistently co-active with a postsynaptic output layer cell during training will develop strong synaptic connections with that postsynaptic output layer cell. It is the relative strengths of the w_{ij} synapses between the presynaptic input layer cells and a particular postsynaptic output layer cell that determine what stimuli that postsynaptic output layer cell will show a learned response to. For instance, if, after training, an output layer cell has strong w_{ij} synapses from the input layer cells assigned to the subset representing the space of distal objects, but very weak w_{ij} synapses

from the input layer cells assigned to all other subsets, then the output layer cell will show a learned response to the space of distal objects, but not to any of the proximal objects.

During learning, the weight vector for each postsynaptic output layer cell is normalized after each learning update. This has the effect of limiting the size of the weight vectors of the output layer cells, which prevents any particular output layer cell from dominating the competition and becoming the only output layer cell that is allowed to fire. Consistent and strong stimulation of a postsynaptic output layer cell by a subset of presynaptic input layer cells will lead to those input layer cells receiving an increasingly larger share of the distribution of the w_{ij} synaptic strengths between the presynaptic input layer cells and the particular postsynaptic output layer cell. Under the correct training conditions, the normalization procedure will help some of the output layer cells to develop learned responses that are exclusive to either the space of distal objects or a proximal object, but not both. In the brain, this type of normalization could be achieved by synaptic weight decay (Oja, 1982; Rolls and Treves, 1998), or conservation of the total synaptic weights through a mixture of Long-Term Potentiation (LTP) and Long-Term Depression (LTD) (Royer and Paré, 2003).

2.3.2 Competitive Learning and the Statistics of the Environment

The statistics of the interaction between the agent and its training environment will produce output layer cells that learn to discriminate between the distal and proximal object spaces. That is, output layer cells that learn to respond exclusively to either the space of distal objects or one of the proximal objects, but not both.

As an example, consider the training environment displayed in Figure 2.2a, containing four train-

ing locations, the space of distal objects, but only a single, centrally-located, proximal object. When the simulated agent visits each training location during a training epoch, the agent will rotate through 360^{o} at each location.

Now consider the firing rates of the subset of input layer cells assigned to represent the proximal object. When the agent rotates at each of the four training locations, there will be a point at each of the training locations where the agent has the same allocentric orientation or head direction θ. When the agent has this particular head direction θ, the head-centred bearing to the proximal object will be different at each of the four training locations. Consequently, for a given head direction θ, the visual input from the proximal object will fall on a different portion of the retina at each training location. This means that within the subset of input layer cells representing the proximal object, there will be a different cluster of input layer cells active when the agent has a particular head direction θ at each of the four training locations.

In contrast, the space of distal objects is assumed to be at an infinite distance from the agent. Thus, when the agent has the same particular head direction θ at each training location, the agent will, in each case, receive the same particular global pattern of visual input from the space of distal objects. Within the subset of input layer cells that are responsive to the orientation of the agent to the space of distal objects, the same cluster of input layer cells will be active when the agent has a particular head direction θ at each of the four training locations.

The key principle here is that, for each particular head direction θ across the four different train-

ing locations, the same small cluster of input cells representing the current orientation to the space of distal objects will be active at each training location. Yet, this consistently active cluster of input cells will, for the same particular head direction θ at each training location, be co-active with a different small cluster of input cells representing the current head-centred bearing from the agent to the proximal object. Thus, across the four training locations there is a statistical decoupling of the input layer cell representations of the distal and proximal objects. This level of statistical decoupling between the visual input from the space of distal objects and the visual input from the proximal object plays a key role in how the output layer of the network is able to learn to form separate representations of the distal and proximal objects. Moreover, as the number of training locations in the training environment increases, this effective statistical decoupling is enhanced. This enhancement helps to increase the discrimination between the distal and proximal objects in the output layer after training of the network.

The result of the statistical decoupling between the distal and proximal object spaces is that, after competitive learning, a subpopulation of the output layer cells will have learned to respond exclusively to different narrow regions of the space of distal objects. The population response of this subpopulation of output cells will reflect the current head direction of the agent, and the individual output layer cells within the population will behave like head direction cells.

2.3.3 Statistics of the Environment with Multiple Proximal Objects

In a natural environment there are likely to be multiple proximal objects as shown in Figure 2.2b. As the number of proximal objects present in the environment increases, it will become more difficult for the competitive network architecture to develop separate representations of all of the proximal objects

as well as the space of distal objects. Thus, the computational problem that the model must learn to solve becomes harder as the number of proximal objects increases.

2.3.4 Anatomical Considerations

As an anatomical consideration, it is highly likely that the type of computation required to form separate representations of distal and proximal visual cues takes place in the visual processing stream that inputs to the head direction cell system. It is also highly likely that the computation will occur at a place in the visual processing stream where the information has already undergone a significant amount of processing (in order for the computation to be able to form separate representations of whole objects).

Bassett and Taube (2005) propose that visual information reaches the head direction cell system by a path of either visual cortex $\rightarrow$ postsubiculum, or visual cortex $\rightarrow$ retrosplenial cortex $\rightarrow$ postsubiculum. It thus seems plausible that the processing proposed in the previous sections would take place in one of these two processing streams. No direct mapping is proposed, however, between the model outlined in the previous sections and a specific region or regions of the brain in one or both of these processing streams. Instead, the computation carried out by the model described in this chapter should be taken as representative of the computation required to form separate learned representations of distal and proximal visual cues. Moreover, the computation is simple enough that it can be carried out by any single set of synapses in the visual processing stream that forms an input to the head direction cell system.

2.4 Model Equations and Implementation

A simplifying assumption is that the eyes of the simulated agent do not move in the eye-sockets: this produces a constant mapping from the retinal visual space of the agent to the head-centred visual space of the agent.

When the network is initialized, 200 input layer cells are assigned to the input layer cell subset representing the space of distal objects, and 200 input layer cells are assigned to each of the M input layer cell subsets representing the M proximal objects in the training environment.

As the simulated agent travels its environment and rotates through 360° at each training location, the firing rate r_i of each input layer cell is set, for the current orientation θ of the agent, to a Gaussian response profile as follows:

$$r_i = \lambda e^{-s_i^2/2\sigma^2} \tag{2.1}$$

where σ is the standard deviation of the Gaussian response, and λ is a scaling factor.

For each input layer cell i that is set to respond to visual input from one of the proximal objects, s_i is defined to be $|x_i - x|$. The current head-centred bearing from the agent to the particular proximal object the input layer cell responds to is given by x, and x_i is the preferred head-centred bearing from the agent to the proximal object for that particular input layer cell i.

For an input layer cell i that is set to respond to visual input from the space of distal objects, s_i is

defined to be

$$s_i = \text{MIN}(|x_i - x|, 360 - |x_i - x|) \tag{2.2}$$

where x is the current orientation of the agent with respect to the space of distal objects, and x_i is the preferred orientation of the agent to the space of distal objects for that particular input layer cell i. It is assumed that at least one distal cue is always visible to the agent as it rotates through 360^{o}. Equation (2.2) ensures that, for those input layer cells set to respond to the space of distal objects, the input layer cell preferred orientations are continuous through 360^{o}, i.e. there is a wrap-around in the input layer cell preferred orientations.

After the input layer cell firing rates r_i have been set, these firing rates are then propagated along the w_{ij} synapses to the output layer cells. The activation level h_i of each postsynaptic output layer cell i is calculated according to

$$h_i = \sum_j w_{ij} r_j \tag{2.3}$$

where r_j is the firing rate of the presynaptic input layer cell j, and w_{ij} is the synaptic weight between the presynaptic input layer cell j and the postsynaptic output layer cell i. The firing rate r_i of each output layer cell i is then calculated as a threshold linear function of the activation level h_i of that cell

$$r_i = f(h_i) \tag{2.4}$$

where the threshold of the function is iteratively adjusted until either the desired sparseness of firing

a in the output layer is achieved, or a maximum number of iterations (300) is reached. The sparseness of firing a of the population of output layer cells is given by

$$a = \frac{(\sum_{i=1}^{T} r_i/T)^2}{\sum_{i=1}^{T} r_i^2/T} \tag{2.5}$$

where r_i is the firing rate of the i-th output layer cell, and T is the total number of output layer cells (Rolls and Treves, 1990, 1998).

The sparseness of firing a governs the level of activity within the output cell layer. For binarized firing rates, the sparseness is the proportion of output layer cells that are active at any given moment. Adjusting the threshold of the function to obtain a desired sparseness of firing promotes competitive learning in the postsynaptic output layer cells, as only those cells with a firing rate above the threshold are allowed to "win" the competition to fire and thus update their w_{ij} synaptic weights with the currently active presynaptic input layer cells. In the brain, this type of competition could by implemented by inhibitory interneurons in the output layer cells.

The synaptic weights w_{ij} between the presynaptic input layer cells and the postsynaptic output layer cells are then updated according to the following Hebbian associative learning rule

$$\delta w_{ij} = k r_i r_j \tag{2.6}$$

where δw_{ij} is the change in strength between the postsynaptic output layer cell i and the presynaptic input layer cell j, r_i is the firing rate of output layer cell i, r_j is the firing rate of input layer cell j, and

k is constant expressing the rate at which the w_{ij} synapses are updated (the learning rate).

The competitive learning taking place in the layer of output cells presents the possibility of the w_{ij} synapses growing unbounded, which would lead to the same few postsynaptic output layer cells always winning the competition to fire and thus updating their w_{ij} synapses with the presynaptic input layer cells. The result of this would be that proper categorization of the input stimuli would fail to occur and the competitive network would cease to function. In order to prevent this, the w_{ij} synapses are bounded by normalization to ensure that, after normalization, for each postsynaptic output layer cell i

$$\sqrt{\sum_{j}(w_{ij})^2} = 1 \tag{2.7}$$

where the sum is over all the presynaptic input layer cells j. The process of weight normalization may be achieved in the brain by synaptic weight decay (Oja, 1982; Rolls and Treves, 1998), or conservation of the total synaptic weights through a mixture of LTP and LTD (Stent, 1973; Royer and Paré, 2003).

When the above calculations have been completed for every orientation of the agent at every training location in the training environment, then one epoch of training has taken place. In total, there are 100 epochs in the training phase to allow the competitive learning mechanism time to converge upon a solution.

The simulation then enters the testing phase to determine if the output layer cells have learned to

respond to the space of distal objects, a proximal object, some mixture of the distal and proximal objects, or a mixture of proximal objects (if there is more than one proximal object in the training environment).

During testing, the firing rate responses of the output layer cells are computed and recorded as the network is tested separately on visual input from either the space of distal objects, or one of the M proximal objects. The network is first presented with visual input from the space of distal objects as the agent rotates through 360^{o}. For each orientation of the agent, the firing rates of the input layer cells assigned to represent the space of distal objects are set as described above. For each orientation, the firing rates of all of the output layer cells are computed and recorded. This process is then repeated for each of the M proximal objects in the training environment. Each proximal object is moved through 270^{o} on the retina, and the firing rates of the input cells representing that proximal object are set as described above. The firing rates of the output layer cells are computed and recorded for each position of the proximal object on the retina.

After the model has been probed for the responses of the output layer cells to the distal and proximal object spaces, the testing phase is complete.

2.5 Results

This section presents the results of investigating the effect that controlled alterations of the model parameters, and the training environment, have upon the learning ability of the model.

Table 2.1: Network parameters for the simulations reported in Chapter 2

Network Parameters	
No. Input Cells Responsive to Space of Distal Objects	200
No. Input Cells Responsive to Proximal Objects	200 $\times$ (No. Proximal Objects)
No. Output Layer Cells	900
No. w_{ij} Synapses onto each Output Layer Cell	Total No. Input Cells
a (Sparseness of Output Layer Cell Firing)	0.1
Training Parameters	
No. Training Epochs	100
k (Learning Rate)	0.001
λ (Scaling of Input Cell Gaussian Response Profiles)	10.0
σ (S.D. of Input Cell Gaussian Response Profiles)	5.0^{o}

All values are constant across all experiments reported in this chapter except where noted in the text.

2.5.1 Experiment 2.1: Demonstration of Core Model Performance

The simulations comprising this experiment were conducted to demonstrate the main operational principles of the model with orthogonal input representations.

The training environment consisted of a single, centrally-placed proximal object, ten training locations placed in a ring around the proximal object, and a circular space of distal objects at an infinite distance from the training locations. The network was simulated with the parameters given in Table 2.1, in accordance with the training and testing protocol described in Section 2.4. The simulation was repeated five times with identical model parameters in each case, but with different random synaptic weight initializations. The repeated simulations were conducted to check that the model learned behaviour is not highly dependent upon a particular initial synaptic weight configuration.

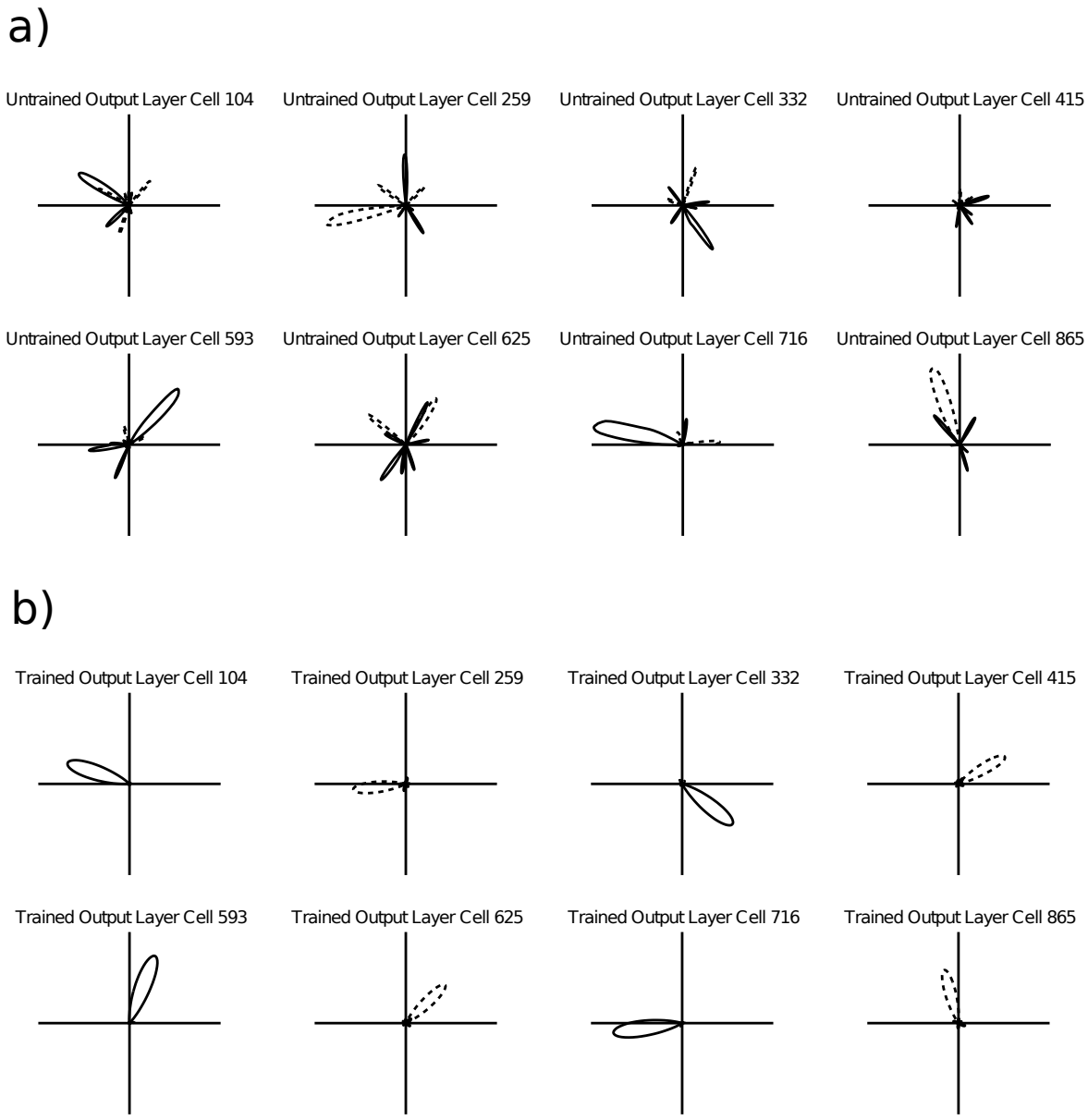

Figure 2.3: The responses of the output layer cells in Experiment 2.1. The responses are presented in polar plot form. a) The untrained responses of eight output layer cells after a testing phase, but no learning phase, has taken place. The unbroken line represents the cell response to the space of distal objects, which is plotted over the interval $0^{o} - 360^{o}$. The dashed line represents the cell response to the proximal object, which is plotted over the $0^{o} - 270^{o}$ field-of-view of the simulated agent. It is clear from the plots that the untrained output layer cells respond to both the distal and proximal object spaces and do so in a multi-modal fashion, i.e. the response curves show multiple peaks. b) The trained responses of the same eight output layer cells shown above. The cell responses are now exclusive to either the space of distal objects, or the proximal object, but not both. The response curves also exhibit a single prominent peak, in contrast to the multi-modal response of the untrained cells. It is clear from comparing the the untrained and trained cell responses that the training regime has a strong effect upon the naive cell responses.

Figure 2.3 displays the responses of eight of the output layer cells when probed with the distal and proximal objects during the testing phase. The cell responses are given in polar plot form. The top eight plots display a selection of the untrained responses of the output layer cells when the network underwent the testing phase without any prior learning taking place. Thus, the responses exhibited by the output layer cells are completely naive. It is clear that the untrained output layer cells respond to both the space of distal objects and the proximal object when probed with these objects during testing. The untrained responses are also multi-modal as the individual response curves exhibit several peaks rather than a single peak.

In comparison, the bottom eight plots of Figure 2.3 display the *learned* responses of the same eight output layer cells. Each response curve exhibits a single peak indicating a defined learned response to a particular view of either the space of distal objects or the proximal object. Each output layer cell displayed has thus learned to represent the space of distal objects, or the proximal object, but not both. Comparing the untrained and trained response curves of each output layer cell reveals that the training regime has a definite influence upon the nature of the cell response, i.e. the individual output layer cells in the untrained network cannot discriminate between the distal and proximal object spaces. Moreover, ten training locations are sufficient for the model to form separate representations of the space of distal objects and a single proximal object.

To further quantify the model performance, the learned responses of the output layer cells were classified by response type. An individual output layer cell can be classified into one of four mutually exclusive groups: responsive to the space of distal objects; responsive to the proximal object; re-

sponsive to both object spaces; or does not respond after training. The criterion for inclusion in the distal-responsive group is that the maximum firing rate of the cell must be above a threshold of 0.03, and the maximum firing rate of the cell in response to the proximal object must be $\leq 20\%$ of the maximum firing rate in response to the space of distal objects. To be included in the proximal-responsive group, an output layer cell must exhibit a maximum firing rate above the threshold of 0.03 and the maximum firing rate of the cell in response to the space of distal objects must be $\leq 20\%$ of the maximum firing rate of that cell in response to the proximal object. Inclusion in the no response group was determined by cells that had a maximum firing rate ≤ 0.03 in response to both the space of distal objects and the proximal object. Any output layer cell that did not meet the three previous criteria was classified as exhibiting a learned response to both the distal and proximal object spaces. The criteria of 0.03 and $\leq 20\%$ were set by a visual inspection and comparison of the output layer cell response curves with the number of cells classified into each mutually exclusive group by different values of the criteria: the aim being to arrive at values for the two criteria which produce a classification of the output layer cells that agrees closely with the classification of the output layer cells by visual inspection alone.

The average number of cells and standard deviations for each response classification, across the five simulation runs with different random synaptic weight initializations, are displayed in Table 2.2. The data shown are the result of applying the testing phase to the model both before and after the training phase had taken place. The data are also displayed as a bar chart in Figure 2.4.

For the untrained, naive, model an average of 32.0 (3.56%) out of 900 output layer cells were classi-

Table 2.2: Responses of the output layer cells in Experiment 2.1 both pre and post-training

Experiment 2.1 Output Layer Cell Responses				
	Untrained		**Trained**	
	Cells	**S.D.**	**Cells**	**S.D.**
No. Cells Responsive to Space of Distal Objects	32.0 (3.56%)	9.3	387.4 (43.04%)	3.0
No. Cells Responsive to Proximal Object	10.6 (1.18%)	1.5	512.6 (56.96%)	3.0
No. Cells Responsive to Both Object Spaces	857.4 (95.27%)	10.1	0	0
No. Cells Responsive to Neither Object Space	0	0	0	0

The response criteria and their values are described in the text (Section 2.5.1). The results shown in the table are the average of five simulation runs in both the untrained and trained model states. Each of the simulation runs was conducted with identical model parameters, but different random synaptic weight initializations. S.D. = standard deviation. It is clear from the table that, in the untrained case, the output layer cells mainly exhibit responses to both the distal and proximal object spaces. After training, the output layer cells show a learned response to either the space of distal objects, or the proximal object, but not both.

fied as exhibiting an exclusive response to the space of distal objects. $10.6\ (1.18\%)$ of the cells were classified as showing an exclusive response to the proximal object. The majority, $857.4\ (95.27\%)$, of the untrained output layer cells were classified as responding to both the distal and proximal object spaces.

After the model was trained, an average of $387.4\ (43.04\%)$ out of 900 output layer cells were classified as exhibiting a learned response that is exclusive to the space of distal objects, and an average of $512.6\ (56.96\%)$ of the output layer cells were classified as showing a learned response that is exclusive to the single proximal object. No output layer cells were classified as responding to both object spaces, and no output layer cells were classified as showing no response.

Thus, it is clear from Figures 2.3, 2.4, and Table 2.2 that the model can produce output layer cells

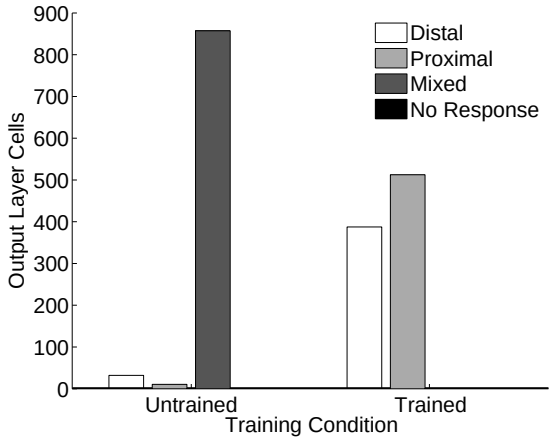

Figure 2.4: Bar chart of the output layer cell responses in Experiment 2.1. The data displayed are the average response types of the output layer cells across five simulations conducted with identical model parameters but different random synaptic weight initializations, as reported in Table 2.2. It is clear that, in the untrained case, the majority of the output layer cells respond to a mixture of the distal and proximal object spaces. After the model has been trained, the output layer cells exhibit a response to either the space of distal objects, or the proximal object, but not both.

that have learned to distinguish between the distal and proximal object spaces in the training environment. That is, after training the model, there will be individual output layer cells that show a learned response that is exclusive to one of the distal or proximal object spaces, but not both. The standard deviations shown in Table 2.2 are small enough in all cases to conclude that there is little variability in the performance of the model when different random synaptic weight initializations are used. Thus, the learning phase has a genuine effect upon the model and the model itself is robust to variations in the initial synaptic weight configurations.

Because the space of distal objects is at an infinite distance from the training locations, whenever the agent has a particular head direction θ at each of the ten training locations then the orientation of the agent to the space of distal objects will be the same. The proximal object, however, will lie on a different head-centred bearing to the agent for the same particular head direction θ at each of the training locations. Thus, the subset of 200 input layer cells that represent the space of distal objects

will receive the same pattern of global visual input from the distal object space, which is represented as aGaussian activity peak at the same position on the ring of input cells whenever the agent has a particular head direction θ. Consequently, the same small cluster of these input cells will be active when the agent has the head direction θ. Within the subset of 200 input layer cells representing the proximal object, however, the visual input from the proximal object will occur at a different retinal position whenever the agent has the same particular head direction θ at each of the ten training locations. This results in a different small cluster of the proximal-object-responsive input layer cells being active for the same particular head direction θ across the different training locations.

The result is that a statistical decoupling is produced between the input layer cell representations of the distal and proximal object spaces. The output layer cells receive these representations from the input layer cells through the w_{ij} synapses, and the statistical decoupling is sufficient for some of the output layer cells to develop learned responses, through associative competitive learning, that are exclusive to either the space of distal objects, or the proximal object space, but not both.

The results reported here demonstrate how a neural network can learn to form separate representations of distal and proximal visual cues presented to it as input. Moreover, it has been shown that a network can achieve this separation in an unsupervised manner, i.e. there is no teacher specifying how to categorize the inputs, or which category to place an input into. It has also been shown that a network can achieve this unsupervised categorization in a single layer of synapses: in this case the w_{ij} synapses. This is the core result that demonstrates the type of computation required in order for head direction cells to be preferentially stimulated by distal visual cues rather than proximal visual cues.

2.5.2 Experiment 2.2: The Effect of Varying the Number of Training Locations

This experiment was conducted to investigate the effect that varying the number of training locations in the training environment has upon the performance of the model. It is hypothesized that, with a single proximal object in the training environment, increasing the number of training locations that the agent visits will increase the ability of the model to produce output layer cells that have learned to discriminate between the distal and proximal objects spaces. It is also hypothesized that there will be a minimum number of training locations that the agent must visit in order that the model can learn to discriminate between the distal and proximal object spaces.

For this experiment four simulations were conducted. In each simulation, the training environment consisted of a single centrally-placed proximal object and a continuous circular space of distal objects at an infinite distance from the proximal object as in Experiment 2.1 (Section 2.5.1). For all simulations the network was simulated with the parameter values given in Table 2.1. The simulations differed in one parameter; the number of training locations in the training environment, which was varied as follows: 1, 2, 5, 10.

Table 2.3 displays the output layer cells classified by learned response type. The data are also displayed in Figure 2.5. With a small number, 1 or 2, of training locations in the environment, the majority of the output layer cells, 768.2 (85.36%) and 900.0 (100%) respectively, exhibit a learned response to both the distal and proximal object spaces. With only a small number of training locations, the statistical decoupling between the distal and proximal object spaces is not high enough for individual output layer cells to form separate representations of the distal and proximal object spaces.

Table 2.3: Learned Responses of the Output Layer Cells in Experiment 2.2

Experiment 2.2 Output Layer Cell Learned Responses								
	No. Training Locations							
	1		2		5		10	
	Cells	S.D.	Cells	S.D.	Cells	S.D.	Cells	S.D.
Distal Responsive	131.8 (14.64%)	2.4	0	0	379.6 (42.18%)	2.9	387.4 (43.04%)	3.0
Proximal Responsive	0	0	0	0	327.2 (36.36%)	22.8	512.6 (56.96%)	3.0
Mixed Response	768.2 (85.36%)	2.4	900.0 (100%)	0	193.2 (21.47%)	25.2	0	0
No Response	0	0	0	0	0	0	0	0

The response criteria are the same as used for Experiment 2.1 (Section 2.5.1). The results shown for each different number of training locations are the average of five simulations conducted with identical model parameters, but different random synaptic weight initializations. S.D. = standard deviation. It is clear that as the number of training locations in the training environment increases, the nature of the output layer cell learned representation changes. With a small number of training locations, (1 or 2), the network is unable to form separate representations of the distal and proximal object spaces (N.B. With 1 training location, a number of output layer cells develop learned responses that are exclusive to the space of distal objects. This is an artefact of the experimental setup, and a full explanation of why this anomaly occurs is given in the text). It is only as the number of training locations increases to be, in the current experiment, ≥ 5 that the network is able to form separate representations of both of the distal and proximal object spaces, which the network does by developing an exclusively local, grandmother-cell-like representation in the output layer cells.

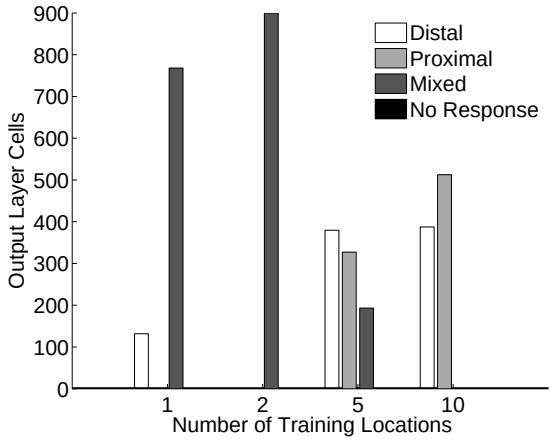

Figure 2.5: Bar chart of the output layer cell learned responses in Experiment 2.2. The chart displays, for each different number of training locations, the average response types of the output layer cells recorded across five simulations conducted with identical model parameters but with different random synaptic weight initializations, as reported in Table 2.3. As the number of training locations increases, the output layer cells change from learning to represent a mixture of the distal and proximal object spaces to learning to represent one of the distal or proximal object spaces, but not both.

Instead, there is a high degree of statistical coupling between the distal and proximal object spaces. This results in individual output layer cells exhibiting mixed learned responses to both the distal and proximal object spaces.

For the simulations with one training location in the environment, an average of 131.8 (14.64%) of the output layer cells were classified as having developed an exclusive response to the space of distal objects. This is due to the agent having the same 270^{o} field-of-view as the rat. Thus, when the agent rotates through 360^{o} at the single training location, there is a 90^{o} interval during which the proximal object is not be visible to the agent. Consequently, during this 90^{o} interval the postsynaptic output layer cells continue to receive stimulation from the presynaptic input layer cells representing the space of distal objects, but do not receive any stimulation from the presynaptic input layer cells representing the single proximal object. This is why an average of 131.8 of the output layer cells were able to develop learned responses that are exclusive to the space of distal objects when there is only a single

training location in the training environment.

This result is clearly an artefact of the experimental setup. When the number of training locations is increased to two, all 900 of the output layer cells learn to respond to both the distal and proximal object spaces. It is not until the number of training locations is further increased that individual output layer cells can learn to discriminate between the distal and proximal object spaces.

When the training environment contains five training locations, 379.6 (42.18%) of the output layer cells exhibited a learned response that is exclusive to the space of distal objects; 327.2 (36.36%) of the output layer cells were classified as showing a learned response that is exclusive to the proximal object; and 193.2 (21.47%) of the output layer cells demonstrated a learned response to both of the distal and proximal object spaces.

Thus, with a single proximal object in the training environment, only a small increase in the number of training locations is required in order for the network to form separate learned representations of the distal and proximal object spaces. This is a direct result of the fact that the level of statistical decoupling between the distal and proximal object spaces is directly proportional to the number of training locations in the environment (although the number of training locations is only one factor in producing a statistical decoupling between the object spaces).

In the simulation conducted with ten training locations in the training environment, none of the output layer cells exhibited a learned response to both of the distal and proximal object spaces. Consequently,

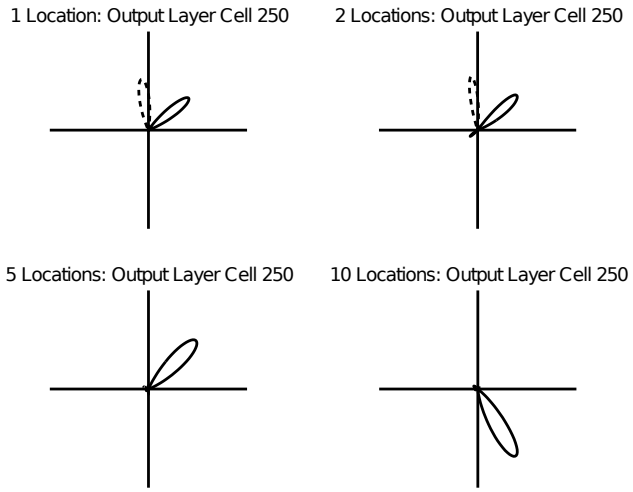

Figure 2.6: The learned responses of the output layer cells in Experiment 2.2. All four polar plots represent data recorded from the same output layer cell in a model simulated with the same initial synaptic weight initializations, but with a different number of training locations in the training environment in each of the four cases: top left, one location; top right, two locations; bottom left, five locations; bottom right, ten locations. The output layer cell response to the space of distal objects is represented by the unbroken line. The cell response to the proximal object is represented by the dashed line. It is clear from the plots that as the number of training locations in the training environment increases, so the network is better able to form separate learned representations of the distal and proximal object spaces. With a small number of training locations, (1 or 2), the network cannot form separate representations of the distal and proximal object spaces. The result is that individual output layer cells learn to respond to both of the distal and proximal object spaces. With a higher number of training locations, (5 or 10), the network is able to form separate representations of the two object spaces, and does so by employing an exclusively local, grandmother-cell-like representation in the output layer cells. This is shown in the bottom two plots, where the output layer cells have learned to respond exclusively to the space of distal objects, and not to the proximal object.

the output layer cells can learn to form separate representations of both of the distal and proximal object spaces. The output layer cells employ an exclusively local, grandmother-cell-like representation where a single output layer cell can uniquely identify either the current orientation of the agent to the space of distal objects (head direction), or the current head-centred bearing from the agent to the proximal object, but not both.

Figure 2.6 displays the learned response profiles of the output layer cells when probed with the distal and proximal objects during the testing phase. Each one of the four polar plots displays the recorded response of the same output layer cell recorded from a simulation conducted with the same synaptic

weight initializations, but with a different number of training locations in the training environment in each case. From the plots it is clear that as the number of training locations in the training environment increases, so the network is better able to separate out the distal and proximal object spaces. With the experimental setup reported here, when the number of training locations is ≥ 5, the network employs a local, grandmother-cell-like representation where individual output layer cells respond exclusively to either a small part of the space of distal objects, or the proximal object space, but not both.

The nature of the representation that develops in the output layer cells is a direct result of an interaction between the number of training locations in the training environment and the distal and proximal object spaces. The same small cluster of input layer cells signalling visual input from the distal object space will be constantly active for a particular head direction θ across all training locations. But, a different cluster of input layer cells, signalling visual input from the proximal object at a particular retinal position, will be active for the same head direction θ across all training locations. Thus, with N training locations in the training environment, a particular cluster of input layer cells representing the orientation of the agent to the space of distal objects will only be co-active 1 in N times with any particular cluster of input layer cells representing the current head-centred bearing of the proximal object to the agent.

Therefore, when there is a single training location in the training environment, a packet of activity in the input layer cells representing the current orientation of the agent to the space of distal objects will always be paired with the same corresponding packet of activity representing the current head-centred bearing of the proximal object to the agent. Thus, the associative competitive learning

mechanism employed by the network will not be able to form separate representations of the distal and proximal object spaces. This is shown in Table 2.3, Figure 2.5, and in the top-left plot of Figure 2.6.

As the number of training locations increases, so the statistical decoupling between the distal and proximal object spaces also increases. This is due to the fact that an activity packet in the subset of input layer cells representing the space of distal objects is paired very infrequently, 1 in $N_{\mathrm{LOCATIONS}}$, with any other activity packet in the proximal-responsive input layer cells. This is why, in the current experiment, as few as five training locations are required in order for the output layer cells to begin to form separate representations of either the space of distal objects or the proximal object space, but not both.

It is interesting to note that, in the case of a single proximal object in the training environment, a relatively small number of training locations is required to produce a high enough statistical decoupling between the distal and proximal object spaces such that local, grandmother-cell-like representations develop in the output layer cells whereby the network can form separate representations of the distal and proximal object spaces. In a natural environment, with multiple proximal objects present, there would not be discrete training locations as in the simulations reported here, but rather a continuum of training locations as the animal travels through the environment. This continuum of training locations should provide a high enough level of statistical decoupling between the distal and proximal object spaces such that the neural network described here should be able to form separate representations of the distal and proximal visual cues presented to it as input.

A prediction that can be made, based upon the results of this experiment, is that an animal must be able to move through its environment in order to learn to discriminate successfully between distal and proximal visual objects. Thus, if an infant animal is restricted in its movements, corresponding to the simulations with 1 or 2 training locations in this current experiment, then the inability to discriminate between distal and proximal visual cues will mean that head direction cells will show no learned preference for stimulation by rotation of distal visual cues rather than stimulation by rotation of proximal visual cues. In principle, this prediction should be fairly straightforward to test empirically.

2.5.3 Experiment 2.3: The Sparseness of Firing within the Output Layer Cells

The purpose of this experiment was to explore the effect that the sparseness of firing within the output layer cells has upon the performance of the model. When the output layer cells are approximately binarized, i.e. 0/1, then the sparseness value, a, as defined by Equation (2.5) in this chapter, can be interpreted as the approximate proportion of output layer cells that are allowed to fire at any given update of the model. The sparseness value effectively determines the level of competition that occurs between the output layer cells, which, in turn, can determine the nature of the output layer cell learned responses.

This experiment explored six different values of the sparseness a. In all the simulations conducted, the training environment was identical to the one used in Experiment 2.1, consisting of ten training locations placed in a ring around a centrally-located proximal object, and a circular space of distal objects at an infinite distance from the training locations. The network was simulated in all cases with the parameter values given in Table 2.1. The only difference across the simulations was the value of

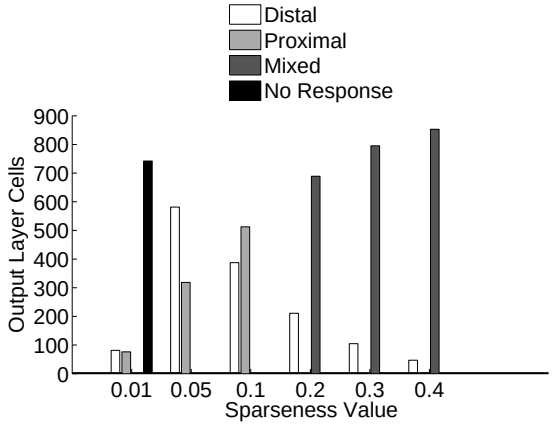

Figure 2.7: The learned responses of the output layer cells in Experiment 2.3. For each different value of sparseness a, the bar chart displays the average number of output layer cells per response type across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The data are also reported in Table 2.4. As the sparseness value a increases, the nature of the learned representation in the output layer cells changes from a local grandmother-cell-like representation where an individual output layer cell represents one of either the distal or proximal object spaces, to a distributed representation where individual output layer cells represent a mixture of the distal and proximal object spaces.

the sparseness, a, which was varied as follows: 0.01, 0.05, 0.1, 0.2, 0.3, 0.4.

Table 2.4 and Figure 2.7 display the results of classifying the output layer cells by learned response type for each of the six different values of a. Of note is the increase in the number of output layer cells that show learned responses to both the distal and proximal object spaces as the sparseness value a becomes larger, i.e. as the number of output layer cells allowed to fire at any given update increases. An increase in a also produces a decrease in the number of output layer cells that do not exhibit a response to either of the distal or proximal object spaces. The optimal sparseness for producing the largest number of output layer cells that respond exclusively to the distal object space is, under the current experimental setup, in the interval $[0.05, 0.1]$.

The representation of the input stimuli in the output layer cells changes, as the value of a increases,

Table 2.4: The learned responses of the output layer cells in Experiment 2.3

Experiment 2.3 Output Layer Cell Learned Responses												
	Sparsness a											
	0.01		**0.05**		**0.1**		**0.2**		**0.3**		**0.4**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	81.4 (9.04%)	5.2	581.4 (64.60%)	3.8	387.4 (43.04%)	3.0	211.0 (23.44%)	3.2	104.8 (11.64%)	1.6	47.0 (5.22%)	3.6
Proximal	76.2 (8.47%)	10.3	318.6 (35.40%)	3.8	512.6 (56.96%)	3.0	0	0	0	0	0	0
Mixed Response	0.2 (0.02%)	0.4	0	0	0	0	689.0 (76.56%)	3.2	795.2 (88.36%)	1.6	853.0 (94.78%)	3.6
No Response	742.2 (82.47%)	12.8	0	0	0	0	0	0	0	0	0	0

The responses are classified according to the same criteria described in Experiment 2.1 (Section 2.5.1). For each value of sparseness, a, the results shown are the average of the results from five different simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. As the sparseness value increases, so the nature of the learned representation in the output layer cells changes from a local, grandmother-cell-like representation with low values of a, to a distributed representation with high values of a. That is, as the sparseness value increases, the number of output layer cells that learn to respond to both of the distal and proximal object spaces also increases, and the number of output layer cells that learn to respond exclusively to one of the distal or proximal object spaces decreases.

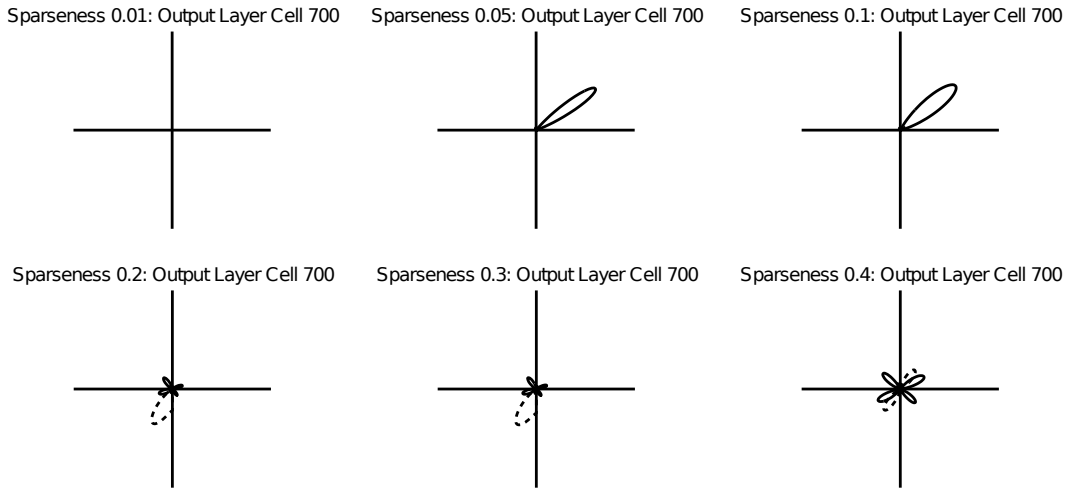

Figure 2.8: Learned responses of the output layer cells in Experiment 2.3. Each polar plot shows the response profile from the same output layer cell taken from one of six simulations conducted with the same synaptic weight initializations, but with a different value for the sparseness a in each simulation. Top row, left to right, a value = 0.01, 0.05, 0.1. Bottom row, left to right, a value = 0.2, 0.3, 0.4. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. As the value of a increases, so the number of output layer cells that respond to both the distal object space and the proximal object increases. The firing profiles also tend to become broader and weaker as a increases, and, with high values of a such as 0.3 and 0.4, multiple peaks are present within the output layer cell response curves to both the distal and proximal object spaces. The plots clearly show that as the sparseness value a increases, so the nature of the learned representation in the output layer cells changes from a local, grandmother-cell-like representation, where individual output layer cells respond to a narrow region of either the distal or proximal object space, but not both, into a distributed representation where individual output layer cells respond to multiple narrow regions of both object spaces.

from a local, grandmother-cell-like encoding where the majority of the output layer cells learn to respond exclusively to one of the distal or proximal object spaces, but not both, to a distributed representation where the majority of the output layer cells learn to respond to both of the distal and proximal object spaces.

Figure 2.8 displays the learned response profiles of the output layer cells when probed with the distal and proximal objects during the testing phase. Each one of the six plots represents the same output layer cell recorded during one of six simulations conducted with the same synaptic weight initializations, but with a different value of the sparseness a for each simulation.

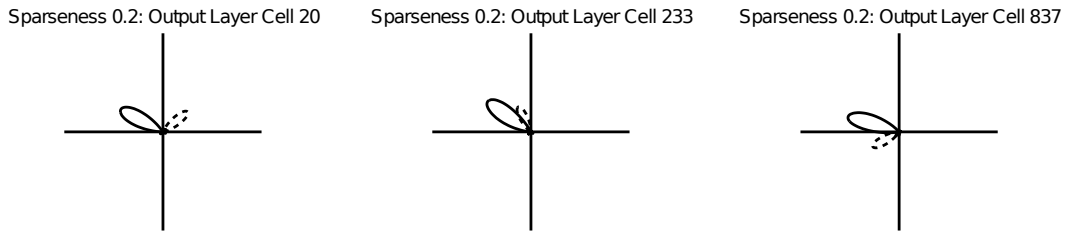

Figure 2.9: The learned responses of three output layer cells from a simulation conducted with a sparseness value, a, of 0.2. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. The three output layer cells respond maximally to the same particular orientation θ of the agent to the space of distal objects. However, the three cells respond maximally to the proximal object when it lies on different bearings to the agent. Thus, there is no fixed relationship between the response of an output layer cell to the space of distal objects and the response of the same cell to the proximal object. This implies that there is likely to be a unique pattern of firing across the whole population of cells in the output layer for each orientation of the agent, θ, to the space of distal objects. There will also be a unique pattern of firing for each head-centred bearing from the proximal object to the agent. It is from this unique pattern of firing across the output layer cells that the orientation of the agent or the position of the proximal object can be determined.

With a low value of a, 0.01, there will only be approximately 1% of the output layer cells firing at any given update of the model. The majority, 742.2 (82.47%), of the output layer cells in the simulations conducted with this sparseness value exhibited no learned response, as is shown in Table 2.4, Figure 2.7, and the top-left plot of Figure 2.8.

When the value of a increases to either 0.05 or 0.1, there will be more of the output layer cells allowed to fire at any given update of the model. All of the output layer cells in the simulations conducted with these two values of a show a learned response that is exclusive to either the space of distal objects, or the proximal object, but not both. This is shown in Table 2.4, Figure 2.7, and the top-middle and top-right plots of Figure 2.8.

For the simulations conducted with an a value of 0.2, the majority, 689.0 (76.56%), of the output layer

cells learned to respond to both the distal and proximal object spaces. Furthermore, as can be seen in the bottom-left plot of Figure 2.8, the learned response curves to both the distal and proximal object spaces are single-peaked in both cases. Because these output layer cells have learned to respond to both the distal and proximal object spaces, the firing of a single one of these cells cannot, by itself, discriminate between the two spaces.

But, as long as each of the output layer cells learns to respond to random parts of the distal and proximal object spaces then the entire population of output layer cells can still discriminate between the two spaces using a distributed population encoding. That is, each part of the distal and proximal object spaces will be represented by a unique pattern of firing across the whole of the output layer. This effect is demonstrated in Figure 2.9, which shows the learned responses of three output layer cells. In each of the three plots, the response to the space of distal objects is the same in that the three output layer cells respond maximally to the same particular orientation, θ, of the agent to the space of distal objects. However, the responses of the cells to the proximal object are different in each plot, i.e. each of the three output layer cells responds maximally to a different head-centred bearing from the proximal object to the agent. Thus, there is no regular relationship between the responses of the output layer cells to the space of distal objects, and the responses of the output layer cells to the proximal object. That is, if, for a particular output layer cell, the response curve to the space of distal objects is known, then the response curve to the proximal object cannot be predicted from this information. This means that there will be a unique pattern of firing across the whole population of output layer cells for each orientation, θ, of the agent to the space of distal objects. There will also be a unique pattern of firing for each head-centred bearing from the proximal object to the agent. Consequently, the head

direction θ of the agent can still be uniquely identified from the firing of the population of output layer cells if the population firing rate vector for the output layer cells points in a unique direction for each unique orientation, θ, of the agent (Dayan and Abbott, 2001).

With a sparseness value, a, of 0.2 a number of the output layer cells, 211.0 (23.44%), were classified as exhibiting a learned response that is exclusive to the space of distal objects. Thus, almost a quarter of the output layer cells developed a local, grandmother-cell-like representation.

With higher values of a such as 0.3 and 0.4, the majority of the output layer cells exhibit a learned response to a mixture of the distal and proximal object spaces. In comparison to an a value of 0.1, however, the output layer cell response curves that develop are multi-peaked for each of the distal and proximal object spaces. This can be seen in the bottom-middle and bottom-right plots of Figure 2.8. With a multi-peaked response profile it is harder to determine the exact current orientation, θ, of the agent to the space of distal objects, or the current head-centred bearing from the proximal object to the agent.

The reason that the nature of the output layer cell representation changes as the value of a increases is that the value of a determines the nature of the competition amongst the output layer cells. Relatively small values of a, such as 0.05 and 0.1, lead to only a small fraction of the total output layer cells being permitted to fire at any model update. This is because a small value of a effectively corresponds to a large amount of inter-cell inhibition amongst the output layer cells, which would be provided by inhibitory interneurons in the brain. The effect of this inhibition is to promote specificity in the learned

responses of the output layer cells. Individual output layer cells will only be permitted to fire for a small fraction of the time that the animal spends exploring its environment, which will lead to those individual output layer cells strengthening their w_{ij} synapses with a relatively small number of the input layer cells. This will produce a local, grandmother-cell-like representation in the output layer cells.

Larger values of a, such as 0.3 or 0.4, effectively correspond to a reduced amount of inter-cell inhibition amongst the output layer cells. Consequently, individual output layer cells will be permitted to fire for a relatively larger amount of the total time the animal spends exploring its environment (in comparison to the case with an a value of, say, 0.05). The output layer cells will now have the opportunity to strengthen their w_{ij} synapses with a relatively large number of the input layer cells. This will produce a distributed representation in the output layer cells.

Therefore, as the value of the sparseness a increases, so the nature of the learned output layer representation changes from a local, grandmother-cell-like representation to a distributed representation. Although, with a distributed representation, the population of output layer cells can discriminate between the distal and proximal object spaces, the original modelling hypothesis was that *individual* output layer cells would learn to discriminate between the distal and proximal object spaces. Thus, as the value of a increases, the model cannot learn the correct type of output representation.

2.5.4 Experiment 2.4: The Number of Proximal Objects and Model Performance

The simulations in this experiment were conducted to investigate the effect that increasing the number of proximal objects in the training environment has upon model performance. In the experiments reported so far in this chapter, there was one circular space of distal objects and one proximal object

in the training environment. This setup was intentionally chosen for purposes of control, as this arrangement produced one of the simplest training environments with which the model could function correctly. Natural environments, however, are highly likely to contain many proximal objects. Thus, in the simulations for this experiment, the number of proximal objects in the training environment was increased.

Four different simulations were conducted with parameter values given in Table 2.1. The only difference between the simulations was the number of proximal objects in the training environment, which was varied as follows: 2, 3, 4, 5.

In Table 2.5 the output layer cells are classified by learned response type for each of the four simulations conducted. The classification criteria are similar to those described in Experiment 2.1 (Section 2.5.1). To accommodate the multiple proximal objects in the environment, an output layer cell is now classified as responding exclusively to the space of distal objects if the maximum firing rate of the cell is above a threshold of 0.03, and the maximum firing rate of the cell in response to all of the proximal objects must be $\leq 20\%$ of the maximum firing rate of the cell in response to the space of distal objects.

Similarly, for an output layer cell to be classified as exhibiting an exclusive learned response to one of the proximal objects, the cell must have a maximum firing rate above the threshold of 0.03, and must have a maximum firing in response to the space of distal objects and the remaining proximal objects that is $\leq 20\%$ of the maximum firing rate of that cell in response to the proximal object.

Table 2.5: The learned responses of the output layer cells in Experiment 2.4

Experiment 2.4 Output Layer Cell Learned Responses								
	Number of Proximal Objects							
	2		3		4		5	
	Cells	S.D.	Cells	S.D.	Cells	S.D.	Cells	S.D.
Distal	190.4 (21.16%)	3.9	11.8 (1.31%)	1.3	1.0 (0.11%)	0	1.2 (0.13%)	0.8
Proximal	120.8 (13.42%)	2.9	23.4 (2.60%)	11.4	4.2 (0.47%)	0.8	0	0
Mixed Response	588.8 (65.42%)	5.9	864.8 (96.09%)	12.1	894.8 (99.42%)	0.8	898.8 (99.87%)	0
No Response	0	0	0	0	0	0	0	0

The responses of the output layer cells are classified according to the criteria described in the text. Four different training environments were employed, with each training environment containing a different number of proximal objects. For each number of proximal objects the results shown are the average across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. As the number of proximal objects increases, so the learned representation in the output layer cells becomes almost exclusively a distributed representation.

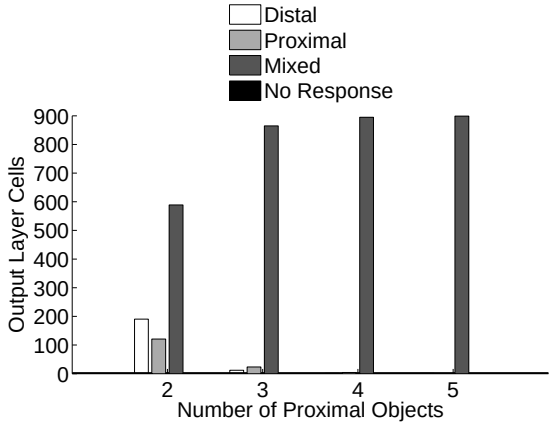

Figure 2.10: The learned responses of the output layer cells in Experiment 2.4. The bar chart displays the average number of output layer cells per response type across the five simulations conducted with identical model parameters but different random synaptic weight initializations for each different number of proximal objects in the training environment. The data are also given in Table 2.5. Increasing the number of proximal objects in the training environment results in almost all of the output layer cells learning a distributed representation.

The criteria for an output layer cell to be classified as either showing no response, or showing a response to some mixture of the space of distal objects and the proximal object spaces, are identical to the criteria given in Experiment 2.1. The average number of output layer cells per response type are also displayed in Figure 2.10.

When there are two proximal objects in the training environment, the majority, 588.8 (65.42%), of the output layer cells were classified as exhibiting a learned response to a mixture of the distal and proximal object spaces. A number of the output layer cells were classified as having developed a learned response exclusively to the space of distal objects, 190.4 (21.16%), or exclusively to one of the proximal object spaces, 120.8 (13.42%), but not both. This is contrasted with the results from Experiment 2.1 where all of the output layer cells were classified as showing a learned response that was exclusive to either the space of distal objects or the proximal object, but not both.

With the current experimental setup, increasing the number of proximal objects in the training environment results in a monotonic increase in the number of output layer cells that exhibit a learned response to a mixture of the space of distal objects and one or more of the proximal object spaces. This is particularly noticeable when comparing the results from the simulations conducted with two and three proximal objects in the training environment. In these cases, the number of output layer cells showing a mixed learned response increases from 588.8 (65.42%) to 864.8 (96.09%) respectively. With five proximal objects in the training environment, nearly all of the output layer cells, 898.8 (99.87%), exhibit a learned response to a mixture of the distal and proximal object spaces.

It is clear that, with all other environmental and model parameters held constant, as the number of proximal objects in the training environment increases from 1 to 5 so the representation of the input stimuli in the output layer cells changes. With 1 proximal object, the output layer cells learn an exclusively local, grandmother-cell-like representation where individual output layer cells respond exclusively to a narrow region of either the space of distal objects or one of the proximal object spaces, but no more than one object space. With ≥ 2 proximal objects in the environment, the output layer cells develop a distributed representation where individual output layer cells learn to respond to multiple narrow regions of both the distal and one, or more, of the proximal object spaces.

Although not shown, an examination of the learned response profiles of the output layer cells for simulations conducted with ≥ 3 proximal objects in the training environment revealed a heterogeneous distribution of learned response types in the output layer cells. That is, some of the output layer cells exhibited a learned response to multiple proximal objects; or a learned response to the space of

distal objects and one of the proximal objects; or a learned response to the space of distal objects and multiple proximal objects.

An important question is why the model performance deteriorates as the number of proximal objects in the training environment increases. As discussed in Section 1.3.2 of Chapter 1, the results of studies by Stringer et al. (2007) and Stringer and Rolls (2008) show that as the complexity of the training environment increases, so the model is better able to learn to discriminate between the input stimuli. Indeed, a minimum level of complexity in the training environment is required in order for the model to successfully discriminate between the input stimuli. On the basis of these results, a prediction regarding the current model is that increasing the number of proximal objects in the training environment should, if anything, improve the ability of the output layer cells to discriminate between the distal and proximal object spaces. In the simulation results reported in this Chapter, improved discrimination is not the case: the model performance deteriorates as the number of proximal objects in the training environment is increased.

It is unclear exactly why a greater number of proximal objects leads to poorer model performance. It should be noted that the previous simulation results of Stringer et al. (2007) and Stringer and Rolls (2008) were achieved with models that were presented with the input stimuli in all possible combinations either two-at-a-time or three-at-a-time. In the current model, all proximal objects are presented simultaneously to the model together with the space of distal objects, rather than in combinations. Thus, the training protocol in the simulations of Stringer et al. (2007) and Stringer and Rolls (2008) may have produced a greater statistical decoupling between the input stimuli than is the case with the

training protocol reported in this chapter. Because the amount of statistical decoupling between input stimuli is a very important factor in determining whether or not a competitive learning mechanism can form separate output representations of input stimuli that are seen together, the current model, with potentially less statistical decoupling between the distal and proximal object spaces, may have a more difficult computational problem to solve than the previous models of Stringer et al. (2007) and Stringer and Rolls (2008).

In principle, learning to discriminate between a space of distal objects and multiple proximal objects may still be a matter of achieving the correct amount of statistical decoupling between those object spaces. Computationally, this could amount to something as simple as discovering the correct training protocol. For instance, if the agent visits a continuum of training locations, rather than the discrete training locations employed in the simulations reported in this Chapter, then it may be possible to produce a local representation in the output layer cells, when there are multiple proximal objects in the training environment, because the larger number of training locations would promote a higher degree of statistical decoupling between the distal and proximal object spaces. This is a natural extension to the results on the number of training locations given in Experiment 2.2 of this chapter, and would correspond to the animal following a more ecologically valid path through its environment. However, it must be noted that it is still unclear exactly why increasing the number of proximal objects in the training environment produced such as degradation in the ability of the current model to discriminate between the distal and proximal object spaces.

Although individual output layer cells cannot discriminate between object spaces when there are

multiple proximal objects in the training environment, when the output layer cells exhibit a learned response to a mixture of the distal and proximal objects, the current orientation, θ, of the agent to the space of distal objects can still be determined as long as there is a unique pattern of firing in the population of output layer cells for each unique orientation of the agent (see the explanation given in the discussion of Experiment 2.3 (Section 2.5.3)). The egocentric bearing to each of the proximal objects can also be determined in the same way from a distributed representation. But the modelling hypothesis is that, with multiple proximal objects in the training environment, *individual* output layer cells will learn to respond exclusively to either the space of distal objects, or one of the proximal object spaces, but not more than one space per output cell. Thus, with multiple proximal objects in the training environment, the current model cannot learn the correct output layer representation.

2.5.5 Experiment 2.5: The Model Performs Well with Less Training

This experiment was conducted to explore the effect that the amount of training has upon the performance of the model. All previous experiments reported in this chapter have been simulated with the model undergoing 100 training epochs in the training environment. This number of training epochs was set to allow time for the feedforward synaptic weights to converge to a stable state, given the competitive learning that takes place in the feedforward synapses between the input cells and the output layer cells. It is of interest whether or not the model can learn to form separate representations of distal and proximal visual objects when exposed to substantially less training epochs, as the results of this experiment will be instructive concerning the speed with which an animal can learn to form separate representations of the distal and proximal visual objects in its environment. As discussed in Chapter 1 (Section 1.2), there is empirical evidence that head direction cells exhibit mature response properties early in the infancy of the rat (Langston et al., 2010; Wills et al., 2010), which suggests that

the head direction system is capable of fast learning.

Three different simulations were conducted with the parameters used in Experiment 2.1 and given in Table 2.1. The only difference between the simulations was the number of training epochs undergone by the model. The number of training epochs was varied as follows: 1, 5, 10.

Table 2.6 displays the output layer cells classified by learned response type for each of the three simulations conducted. The data are also displayed in Figure 2.11. When the model is trained with only one training epoch, the majority, 699.2 (77.69%) of the output layer cells were classified as showing a learned response to a mixture of the distal and proximal object spaces. An average of 126.6 (14.07%) of the output layer cells developed a learned response that was exclusive to the space of distal objects. An average of 72.4 (8.04%) of the output layer cells learned to respond exclusively to the proximal object. Thus, when trained with a single training epoch, the model produces a distributed representation in the output layer cells of the distal and proximal objects. This can be seen in Figure 2.11 and the left-hand plot of Figure 2.12. This is in contrast to Experiment 2.1, in which, after 100 training epochs, all of the output layer cells were classified as responding exclusively to either the space of distal objects or the space of proximal objects. Therefore, the amount of training the model receives has an influence upon the nature of the learned representation in the output layer cells, with too little training leading to an incorrect type of output representation, i.e. distributed.

Training the model with five training epochs changes the nature of the learned representation in the output layer cells. 307.6 (34.18%) of the output layer cells learned to respond exclusively to the

Table 2.6: The learned responses of the output layer cells in Experiment 2.5

Experiment 2.5 Output Layer Cell Learned Responses						
	Number of Training Epochs					
	1		5		10	
	Cells	S.D.	Cells	S.D.	Cells	S.D.
Distal	126.6 (14.07%)	7.3	307.6 (34.18%)	3.4	377.6 (41.96%)	12.6
Proximal	72.4 (8.04%)	4.3	315.8 (35.09%)	4.3	429.4 (47.71%)	12.8
Mixed Response	699.2 (77.69%)	11.7	276.6 (30.73%)	3.4	93.0 (10.33%)	2.6
No Response	1.8 (0.20%)	1.1	0	0	0	0

The output layer cell responses are classified according to the criteria described in Experiment 2.1 (Section 2.5.1). Three different simulations were conducted, each with the model trained for a different number of epochs. For each number of training epochs, the results shown are the average across five simulations conducted with identical model parameters, but with different synaptic weight initializations. S.D. = standard deviation. As the number of training epochs increases, so the nature of the output layer cell learned representation changes from a distributed representation, where individual output layer cells respond to multiple narrow regions of both of the distal and proximal object spaces, to a local representation, where individual output layer cells learn to respond exclusively to a narrow region of either the space of distal objects or the proximal object space, but not both. The model is able to approach the performance of the fully-trained model after only a few, i.e. $5 - 10$, epochs of training.

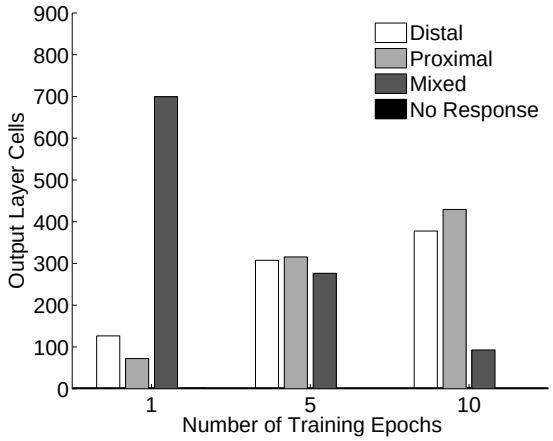

Figure 2.11: The learned responses of the output layer cells in Experiment 2.5. Displayed are the average number of output layer cells per learned response type for simulations conducted with different numbers of training epochs, as reported in Table 2.6. For each different number of training epochs, the average is calculated across five simulations conducted with identical model parameters but with different random synaptic weight initializations. As the number of training epochs is increased, the output layer cell learned representation changes from a distributed representation to a local representation.

space of distal objects. 315.8 (35.09%) of the output layer cells learned to respond exclusively the proximal object, and 276.6 (30.73%) of the output layer cells learned to respond to a mixture of the distal and proximal object spaces. After five training epochs, the output layer cells exhibit a mixture of learned representations. Approximately one third of the output layer cells learn to respond exclusively to the space of distal objects, and approximately one third of the output layer cells learn to respond exclusively to the proximal object. These output layer cells thus represent the distal and proximal object spaces in a local, grandmother-cell-like representation. The remaining third of the output layer cells represent the distal and proximal object spaces in a distributed manner. As can be seen in the middle plot of Figure 2.12, analysis of the output layer cell learned response curves for those output layer cells classified as developing a grandmother-cell-like representation revealed that the response curves were uni-modal, i.e. the output layer cells exhibit a response to a single narrow region of either the space of distal objects or the proximal object space, but not both.

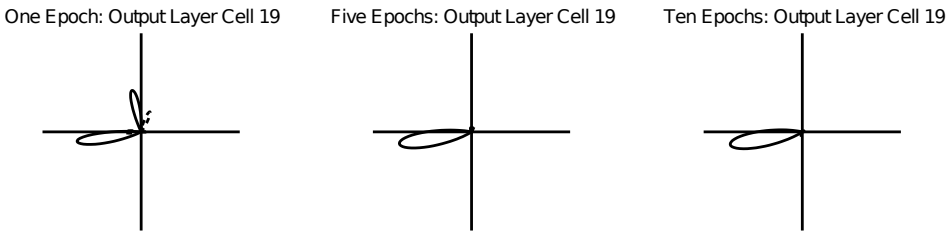

Figure 2.12: The learned responses of a single output layer cell after different amounts of training. The plots display data recorded from the same simulation at different instances in the duration of the simulation. The left-hand plot shows the output layer cell after one epoch of training. The cell exhibits a preference for a particular view of the distal object space. There is a smaller response to another view of the distal object space, as well as a small response to the proximal object space. The output layer cells have learned to produce a distributed representation of the distal and proximal object spaces. That is, the majority of the output layer cells respond to multiple regions of the distal object space and the proximal object space. The middle plot displays the same output layer cell after five epochs of training. The response of the output layer cell has become more specific in comparison to the situation after a single epoch of training. The output layer cell now exhibits only one response to the space of distal objects, although there is still a small response to the proximal object space. The right-hand plot displays the same output layer cell after ten epochs of training. The further training has increased the specificity of the output cell response. There is now a clearly defined response to one particular view of the distal object space. After ten epochs of training, the output layer cells produce a local representation of the distal and proximal object spaces. That is, the majority of the output layer cells respond to either one small region of the space of distal objects or the proximal object space, but not both.

Increasing the number of training epochs to ten further changes the nature of the output layer cell learned representation. An average of 377.6 (41.96%) of the output layer cells learned to respond exclusively to the space of distal objects. 429.4 (47.71%) of the output layer cells learned respond exclusively to the proximal object, and 93.0 (10.33%) of the output layer cells learned to respond to a mixture of the distal and proximal object spaces. The majority of the output layer cells learn to represent the distal and proximal object spaces by employing a local representation, with each output layer cell responding to a single small region of either the space of distal objects or the proximal object space, but not both. The right-hand plot of Figure 2.12 displays an output layer cell that has learned to respond to the space of distal objects in a local, grandmother-cell-like manner.

Thus, training the model with only ten training epochs produces model performance that approaches the performance of a model trained with 100 training epochs (c.f. Experiment 2.1 in this chapter).

This result has interesting implications for the speed at which the head direction cell system develops. Recent studies have shown that head direction cells exhibit adult-like firing properties almost immediately after the rat pup opens its eyes (Langston et al., 2010; Wills et al., 2010). The simulations comprising this experiment demonstrate that head direction cells can exhibit adult-like firing properties, such as differentiating between distal and proximal visual cues, with very little visual experience of the current environment. More training does improve performance, but the important point is that the general trend of how the output layer cells will respond is already evident with as little as one epoch of training in the environment.

2.5.6 Experiment 2.6: Model Performance and the Learning Rate

The simulations comprising this experiment explored the effect that different values of the learning rate, k, of the Hebbian learning rule, Equation (2.6), have upon the performance of the model. k is often set to be $\ll 1$ but, even at such a relatively small magnitude, the different values of k may have a profound effect upon the performance of the model. Three different simulations were conducted with the model parameters given in Table 2.1. The difference between the simulations was the value of the learning rate, k, which was varied as follows: 0.1, 0.01, 0.001.

Table 2.7 displays the output layer cells classified by learned response type for each of the three simulations conducted. The data are also displayed as a bar chart in Figure 2.13. When the model is simulated with a learning rate, k, of 0.1 then the majority of the output layer cells, 866.4 (96.27%), learn to respond to a mixture of the distal and proximal object spaces. A small number of the output layer cells, 25.4 (2.82%) and 8.2 (0.91%), learn to respond exclusively to the space of distal objects or the proximal object respectively. All of the output layer cells learn to respond to at least one of

Table 2.7: The learned responses of the output layer cells in Experiment 2.6

Experiment 2.6 Output Layer Cell Learned Responses						
	Learning Rate					
	0.1		0.01		0.001	
	Cells	S.D.	Cells	S.D.	Cells	S.D.
Distal	25.4 (2.82%)	7.0	92.2 (10.24%)	2.8	387.4 (43.03%)	3.0
Proximal	8.2 (0.91%)	2.6	124.8 (13.87%)	8.4	512.6 (56.96%)	3.0
Mixed Response	866.4 (96.27%)	7.8	683.0 (75.89%)	10.7	0	0
No Response	0	0	0	0	0	0

The output layer cell responses are classified according to the criteria given in Experiment 2.1 (Section 2.5.1). Three different simulations were conducted, each with a different value of the learning rate k. For each of the three values of the learning rate, the results shown are the average of the results from five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. As the value of the learning rate, k, decreases by an order of magnitude, so the nature of the output layer cell learned representation changes. That is, with a smaller value of the learning rate k, the model is better able to form separate representations of the distal and proximal visual objects in the training environment using a local representation. With a higher value of k, the model employs a distributed representation of the distal and proximal object spaces. For every value of the learning rate, all output layer cells developed a learned response to at least one of the object spaces. Decreasing the value of the learning rate, k, increases the specificity of the output layer cell learned response.

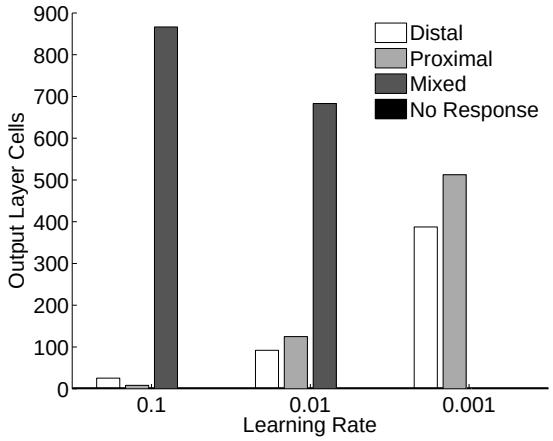

Figure 2.13: The learned responses of the output layer cells in Experiment 2.6. For each value of the learning rate k, the bar chart displays the average number of output layer cells per response type calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The data are also reported in Table 2.7. As the value of the learning rate k decreases, the output layer cells are better able to learn to discriminate between the distal and proximal object spaces by employing a local representation. In contrast, a higher value of k produces a distributed representation in the output layer cells, with the result that individual output layer cells learn to respond to a mixture of the distal and proximal object spaces.

the distal or proximal object spaces. The left-hand plot of Figure 2.14 displays an output layer cell that has learned to respond to several views of both the distal and proximal object spaces. Thus, if specificity of the output layer cell learned response is required then a learning rate, k, of 0.1 is of too large a magnitude. Instead, a particular output layer cell will with high probability, strengthen its w_{ij} synapses with a large number of the input cells representing both distal and proximal object spaces. This results in the majority of the output layer cells exhibiting a learned response to a mixture of the distal and proximal object spaces, and thus representing these object spaces in a distributed manner.

When the learning rate, k, is decreased by an order of magnitude to 0.01 then the majority of the output layer cells, 683.0 (75.89%), still learn to respond to a mixture of the distal and proximal object spaces. There is a slight increase in the number of output layer cells that learn to respond exclusively to either the space of distal objects, 92.2 (10.24%), or the proximal object, 124.8 (13.87%). All of

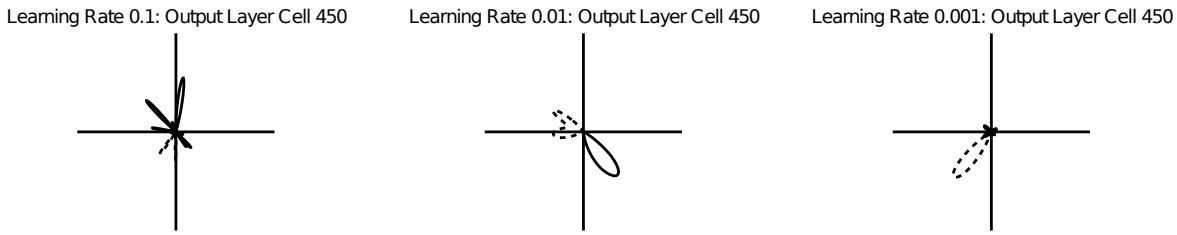

Figure 2.14: The learned responses of a single output layer cell when the model is simulated with different values of the learning rate k. The three plots display data recorded from the same output layer cell in simulations conducted with the same synaptic weight initializations. The only difference between the simulations was the value of the learning rate k. In the left-hand plot, the learning rate is 0.1. The output layer cell has learned to respond to a mixture of the distal and proximal object spaces, and the output layer cell is preferentially stimulated by multiple views of both the distal and proximal object spaces. In the middle plot, the model was simulated with a learning rate of 0.01. The output layer cell has learned to respond to both the distal and proximal object spaces. In comparison to the left-hand plot, the output layer cell in the middle plot is more specific in its learned response as it is preferentially stimulated by a single view of the both the distal and proximal object spaces. The right-hand plot displays data recorded from a simulation with a learning rate of 0.001. The output layer cell has learned to respond exclusively to a single view of the proximal object. Thus, as the value of the learning rate k is decreased, so the learned responses of the output layer cells become more specific.

the output layer cells developed a learned response to at least one of the distal or proximal object spaces. The middle plot of Figure 2.14 displays an output layer cell that has learned to respond to both the distal and proximal spaces, but the response of this cell is more specific than the response of the cell displayed in the left-hand plot of the same Figure. A learning rate k of 0.01 is thus of still too high a magnitude to promote specificity in the learned responses of the output layer cells such that the output layer cells produce a local representation of the distal and proximal object spaces, with output layer cells responding to a single region of either the space of distal objects or the proximal object space, but not both. Instead, the output layer cells represent the distal and proximal object spaces in a distributed manner.

Decreasing the learning rate k by another order of magnitude to 0.001 significantly changes the nature of the output layer cell learned representation. Now, 387.4 (43.03%) of the output layer cells exhibit

a learned response that is exclusive to the space of distal objects. 512.6 (56.96%) of the output layer cells display a learned response that is exclusive to the proximal object. None of the output layer cells learn to respond to a mixture of the distal and proximal object spaces, nor do any of the output layer cells become unresponsive through learning. The right-hand plot of Figure 2.14 displays an output layer cell that has learned to respond exclusively, and specifically, to a particular view of the proximal object. Consequently, a learning rate k of 0.001 is of a small enough magnitude that the output layer cells develop a specificity for a view of either the distal object space, or the proximal object space, but not both, in their learned responses. The output layer cells thus learn to represent the distal and proximal object spaces in a local, grandmother-cell-like manner

This experiment was conducted to explore the effect that the magnitude of the learning rate k has upon the performance of the model. In the current model, a learning rate k of a relatively high magnitude, either 0.1 or 0.01, results in the postsynaptic output layer cells becoming "greedy" and strengthening their w_{ij} synapses with a large number of the presynaptic input layer cells representing both the distal and proximal object spaces. Thus, the general trend is for the output layer cells to lack specificity in their learned responses, and a distributed representation of the distal and proximal object spaces develops in the output cell layer. A relatively small value of the learning rate k of 0.001 leads to the postsynaptic output layer cells only strengthening their synapses with a small number of the presynaptic input cells. Thus, the output layer cells develop specific responses that are exclusive to either the space of distal objects, or the proximal object, but not both. This produces a local, grandmother-cell-like representation of the distal and proximal object spaces in the output cell layer.

This result could also be evidence of a "stability-plasticity dilemma" within the current model. The stability-plasticity dilemma is a computational problem that any learning system must solve: namely, being plastic enough to incorporate new information, but remaining stable enough to not overwrite already stored information (Carpenter and Grossberg, 1987, 1988; Grossberg, 1987). The value of the learning rate k in the current model has a direct influence upon how stable or plastic the model is. A relatively high value of k, such as 0.1 or 0.01, results in too much plasticity in the system and not enough stability. Consequently, the model learns to respond to multiple object spaces, overwriting the exclusive responses to a single object space. It is only when k is set to a relatively small value of 0.01 that the system is plastic enough to incorporate new information, but stable enough to maintain exclusive responses to a single object space.

2.5.7 Experiment 2.7: Model Performance is Independent of Network Size

Competitive learning is introduced into the model by adapting the threshold of the activation-to-firing-rate transfer function so that only a certain percentage of the output layer cells are allowed to fire at any given model update. This percentage is usually approximately 10%, and the competitive learning mechanism is described in Section 2.4 and Equations (2.4) and (2.5). For a given sparseness, the percentage of output layer cells that are allowed to fire at any given model update will remain approximately constant assuming the firing rates of the output layer cells are reasonably binarized. The actual number of output layer cells firing will vary, however, with the total number of output layer cells in the current model. The number of output layer cells firing at any given update may have an effect upon the learned behaviour of the model. Thus, the simulations comprising this experiment were conducted to explore the effect that varying the number of output layer cells has upon the performance of the model.

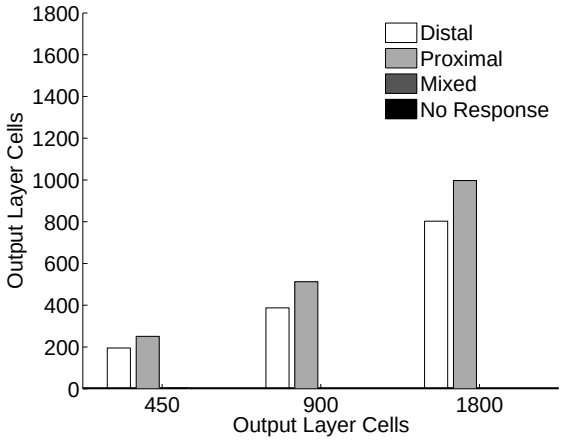

Figure 2.15: The learned responses of the output layer cells in Experiment 2.7. The bar chart displays the average number of output layer cells per learned response type calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations, for each of the three models implemented with a different number of output layer cells. The data are also given in Table 2.8. The nature of the output layer learned representation remains qualitatively the same across the three different models containing different numbers of output layer cells. That is, the output layer cells learn to respond exclusively to either the space of distal objects or the proximal object, but not both, for each of the three different numbers of output layer cells in the model.

Three different simulations were conducted in total. For each simulation, the model was simulated with the parameters given in Table 2.1. The difference between the simulations was the number of output layer cells. This was varied as follows: 450, 900, 1800.

Table 2.8 and Figure 2.15 display the learned response types of the output layer cells for each of the three simulations. With 450 output layer cells in the model, 195.2 (43.38%) of the output layer cells learned to respond exclusively to a particular view of the distal object space. 251.0 (55.78%) of the output layer cells exhibited a learned response that is exclusive to a particular view of the proximal object. A small number, 3.8 (0.84%), of the output layer cells learned to respond to a mixture of the distal and proximal object spaces. All of the output layer cells learned to respond to at least one of the distal or proximal object spaces.

Table 2.8: The learned responses of the output layer cells in Experiment 2.7

Experiment 2.7 Output Layer Cell Learned Responses						
	No. Output Layer Cells					
	450		900		1800	
	Cells	S.D.	Cells	S.D.	Cells	S.D.
Distal	195.2 (43.38%)	4.6	387.4 (43.03%)	3.0	802.6 (44.59%)	7.8
Proximal	251.0 (55.78%)	3.2	512.6 (56.96%)	3.0	997.4 (55.41%)	7.8
Mixed Response	3.8 (0.84%)	0	0	0	0	
No Response	0	0	0	0	0	0

The output layer cell responses are classified according the to the criteria given in Experiment 2.1 (Section 2.5.1). Three simulations were conducted. Each simulation was implemented with a different number of output layer cells. For each number of output layer cells, the results shown are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. It is clear from the learned responses of the output layer cells that changing the number of output layer cells in the model has no effect upon the learned behaviour of the model. For the three different simulations, the percentage of output layer cells classified as showing a particular learned response is very similar across the three simulations. Thus, the model performance is independent of the number of output layer cells in the model.

Increasing the number of output layer cells to 900 resulted in 387.4 (43.03%) of the output layer cells learning to respond exclusively to the space of distal objects. 512.6 (56.96%) of the output layer cells learned to respond exclusively to the proximal object. No output layer cells learned to respond to a mixture of the distal and proximal object spaces. Thus, all of the output layer cells learned to respond exclusively to either the distal object space, or the proximal object, but not both.

Further increasing the number of output layer cells to 1800 produced 802.6 (44.59%) output layer cells that learned to respond exclusively to the space of distal objects. 997.4 (55.41%) of the output layer cells learned to respond exclusively to the proximal object. None of the output layer cells developed a response to a mixture of the distal and proximal object spaces. All of the output layer cells learned to respond exclusively to either the space of distal objects, or the proximal object, but not both.

Importantly, the relative percentages of the output layer cells that are classified according to a particular learned response type remain approximately the same regardless of the number of output layer cells in the model. For instance, approximately $43\% - 44\%$ of the output layer cells learn to respond exclusively to the space of distal objects whether there are 450, 900, or 1800 output layer cells in the model. Thus, the learned behaviour of the model is independent of the size of the model. Consequently, the computational mechanism by which the model learns to form separate representations of distal and proximal visual objects is scalable across different network sizes.

Figure 2.16 displays the learned responses of a single output layer cell recorded from three different models, each implemented with a different number of output layer cells. The left-hand plot displays

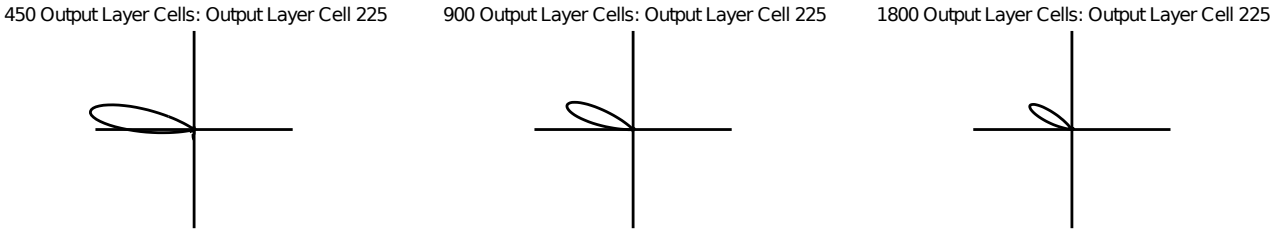

Figure 2.16: The learned responses of a single output layer cell when the model is simulated with different numbers of output layer cells. The left-hand plot displays data recorded from a simulation with 450 output layer cells. The output layer cell in the plot has learned to respond exclusively to a particular preferred view of the space of distal objects. The middle plot displays the same output layer cell as the left-hand plot, but this time recorded from a simulation conducted with 900 output layer cells. Again, the output layer cell has learned to respond exclusively to the space of distal objects, and the preferred view is approximately the same as the cell in the left-hand plot, although there is a reduction in the magnitude of the cell response. The right-hand plot displays data recorded from a simulation conducted with 1800 output layer cells. The output layer cell has also learned to respond exclusively to the space of distal objects with a preferred view that is approximately the same as the cells in the left-hand and middle plots. There is a reduction in the magnitude of the cell response in comparison to the left-hand and middle plots. Thus, the learned behaviour of the model is independent of the number of output layer cells in the model.

data from a model simulated with 450 output layer cells. The middle plot displays data from a model simulated with 900 output layer cells. The right-hand plot displays data from a model simulated with 1800 output layer cells. In all three plots, the same output layer cell develops a learned response that is exclusive to approximately the same view of the space of distal objects. The only difference between the plots is that as the number of output layer cells increases, so the magnitude of the output layer cell learned response decreases. This decrease in response magnitude does not affect the ability of the model to learn separate representations of the distal and proximal object spaces.

In summary, the learned model behaviour is independent of the number of output layer cells in the network. This is an important result, which demonstrates that the computational principle is a general one that can be implemented by a network of any size.

2.6 Discussion

This chapter provides the implementation details, equations, and simulation results for a single-layer feedforward neural network model that can learn to form separate representations of distal and proximal visual cues presented simultaneously to the network as input. The computation performed by this model is representative of the type of computation required in order for the head direction cell system to be stimulated preferentially by distal visual cues rather than proximal visual cues. In summary, the main points of this chapter are:

- The model comprises a single layer of input cells connected to a single layer of output cells in a feedforward manner by a single layer of synapses.

- The training environment consists of a circular space of distal objects, together with a number of proximal objects and a number of training locations. The number of proximal objects is altered according to the current experimental conditions. The number of training locations is also altered according to the current experimental conditions.

- The computational task that the model must learn to solve is to form separate output layer cell representations of the distal and proximal object spaces. Orthogonal, non-overlapping, subsets of input layer cells represent the current retinal position of visual input from the space of distal objects, and the current retinal position of visual input from each of the proximal objects in the current training environment. In the case of visual input from the space of distal objects, the current retinal position of that visual input corresponds to the current orientation of the agent with respect to the space of distal objects. In the case of visual input from a proximal object, the current retinal position of that visual input corresponds to the current head-centred bearing

from the agent to that proximal object.

- The statistics of the interaction between the agent and the distal and proximal object spaces permits a competitive learning mechanism to form separate output layer cell representations of the distal and proximal object spaces. When the agent maintains a particular head direction θ at each of the training locations in the training environment, then the visual input from the space of distal objects will always occur at the same retinal position. For the same behaviour of the agent, the visual input from a proximal object will, however, occur at different retinal positions. These differences in retinal position produce a statistical decoupling between the distal and proximal objects spaces, which allows the competitive learning mechanism to solve the computational task of forming separate output layer cell representations of the two distal and proximal object spaces.

- In simulation, when the agent explores a simple training environment, the model is able to learn to discriminate between the distal and proximal object spaces. The agent must visit a minimum number of discrete training locations in order for the model to solve the computational task, which corresponds to a requirement that an animal must move through its environment in order to learn to differentiate distal and proximal visual cues. There is an optimal range of the sparseness value a in order to produce output layer cells that can discriminate between the distal and proximal object spaces. This optimal range corresponds to a requirement for relatively strong inter-cell inhibition in the output cell layer.

- Increasing the number of proximal objects in the training environment changes the nature of the output layer cell learned representation. When there are $\geq$ 2 proximal objects present in the training environment, the output layer cells learn to represent the distal and proximal

object spaces in a distributed manner, where individual output layer cells develop response to multiple narrow regions of both the distal and proximal object spaces. With 1 or 2 proximal objects present in the training environment, the output layer cells learn to represent the distal and proximal object spaces in a local, grandmother-cell-like manner, where individual output layer cells learn to respond exclusively to a narrow region of either the space of distal objects or a proximal object space, but not multiple spaces. Altering the value of the learning rate k also has a similar effect. Relatively high values of k, such as 0.1 or 0.01, produce a distributed output layer cell representation. A low value of k, such as 0.001, is required to produce a local representation. Because the modelling hypothesis is that *individual* output layer cells will learn to discriminate between the distal and proximal object spaces, whenever the model learns a distributed output representation then the model has not learned the correct type of output representation.

- The model is robust to a reduction in the amount of training the model is permitted. The model is also robust to a change in the number of output layer cells in the network.

As stated above, the model presented in this chapter is representative of the type of computation necessary to discriminate between distal and proximal visual inputs. As such, the model is presented in the form of the simplest network architecture and training environment capable of producing the required output layer cell differentiation between the distal and proximal visual cues. Consequently, the output layer cells lack a few properties that are common to real head direction cells.

In previous experimental studies, distal and proximal visual cues have been placed in conflict with one another in order to determine that the head direction cell system is primarily influenced by distal

visual cues (Zugaro et al., 2004; Yoganarasimha et al., 2006). This is not the case for the testing protocol used for the model reported in this chapter. The current model cannot, without adaptation, be tested with a conflict situation between distal and proximal visual cues. This was a deliberate modelling decision, as the aim of the simulations conducted in this chapter was to demonstrate, in as simple a manner as possible, how it is the statistical decoupling between the distal and proximal visual cues that can lead a neural network model, through competitive learning, to discriminate between these cues. The relative influence of distal visual cues, compared to proximal visual cues, upon the head direction cell system is likely to be more complex, involving many factors. For instance, Zugaro et al. (2001) have shown that a change in the perceived, but not relative, distance of a visual cue to the observer can dramatically alter the influence that the visual cue has upon the preferred firing directions of individual head direction cells. In order to produce a simpler setup in which it is possible to fully elucidate how the statistical decoupling between distal and proximal visual cues enables a neural network to learn to discriminate between those cues, the decision was taken to not test the current model in a distal vs. proximal cue conflict situation as reported in the experimental literature.

Further to this point, the output layer cells in the current model will not act coherently as a population. When distal and proximal visual cues are placed in conflict with one another, real head direction cells will track either the distal visual cues or the proximal visual cues, but not a mixture of the two, i.e. the population of head direction cells will act as a coherent whole (Zugaro et al., 2004; Yoganarasimha et al., 2006). In the model presented in this chapter, a conflict between the distal and proximal object spaces will produce some output layer cells that follow the distal object space, and some output layer cells that will follow the proximal object. Without modification, the model presented in this chapter

cannot replicate the empirical data. The current model is, however, the minimal model that can successfully learn to reproduce a well-known behaviour of real head direction cells. As a minimal model, the current model provides a high degree of experimental control over its behaviour, which clarifies the analysis and explanation of that behaviour. The current model is presented as a first step towards a more complex model where the trained output layer cells more closely reproduce the known properties of head direction cells, including population coherency in the presence of a conflict between distal and proximal visual cues.

The current model does not incorporate any idiothetic input; particularly, vestibular input. The absence of vestibular input does not imply that vestibular input is unimportant to the head direction cell system. Indeed, the addition of vestibular input to the current model would aid in the forming of separate representations of the distal and proximal object spaces. This is because only the distal visual cues would maintain a stable relationship with the vestibular input. A competitive learning mechanism would take advantage of this stable relationship by favouring the strengthening of the feedforward w_{ij} synapses between the distal-responsive input cells and the output layer cells. The addition of vestibular inputs to the current model would thus "clean-up" the learned behaviour of the output layer cells insofar as significantly more of the output layer cells would learn to respond to the distal visual cues than would learn to respond to the proximal visual cues.

It is interesting that increasing the number of proximal objects in the training environment alters the nature of the output layer cell learned representation. As discussed in Section 2.5.3, the output layer cells, as a population, can still discriminate between the distal and proximal object spaces when

employing a distributed representation. It would, however, be interesting to further investigate why an increase in the number of proximal objects produces such a change in the output layer cell learned representation. Moreover, is it possible to find a model configuration where increasing the number of proximal objects produces a local, grandmother-cell-like representation in the output layer cells. One possibility is that if the agent travels a more ecologically valid continuous path through the environment, rather than visiting discrete locations, the statistical decoupling between the distal and proximal object spaces may increase to the extent that a local representation is maintained in the output layer cells. It would also be interesting to investigate the learned behaviour of the model when the agent is embedded in a 3D visual training environment. Such an environment would provide a more rich and complex input to the network, which could increase the statistical decoupling between the distal and proximal object spaces.

It can be argued that the reason the output layer cells are able to learn to discriminate between the distal and proximal object spaces is because the distal and proximal object spaces are orthogonal, and therefore the object spaces are already separated in the input cell layer. That is, one subset of input layer cells represents the current visual input from the space of distal objects, and a separate subset of input layer cells represents the current visual input from each of the proximal objects in the training environment. Under this hypothesis, the output layer cells are highly likely to learn to discriminate between the distal and proximal object spaces because the orthogonal nature of the inputs to the model biases the learning of the output layer cells and produces the outcome where the output layer cells can discriminate between the distal and proximal object spaces.. Thus, it is argued that the model does not in fact learn to discriminate between distal and proximal visual objects based solely upon a statistical

decoupling between the objects. Rather, the predetermined orthogonal nature of the model inputs favour the development of output layer cells that can discriminate between the distal and proximal object spaces.

This is not, however, the case. Firstly, there is full feedforward synaptic connectivity between the input cell layer and the output cell layer. That is, every input cell sends efferent synapses to every output layer cell, and every output layer cell receives afferent synapses from every input layer cell. Consequently, although the distal and proximal object spaces may be separated in the input cell layer, the object spaces are, in the untrained model, completely combined in the output cell layer through the full feedforward synaptic connectivity from the input cell layer.

Second, if the orthogonal nature of the inputs in any way biases the output layer cells towards learning to discriminate between the distal and proximal object spaces, then it would be expected that when the output layer cells are tested without having first undergone training there would already be a degree of discrimination present between the distal and proximal object spaces. The results of Experiment 2.1 clearly show that the output layer cells in the untrained model do not exhibit any meaningful discrimination between the distal and proximal object spaces. After training, all of the output layer cells have learned this discrimination and respond exclusively to either the space of distal objects or the proximal object, but not both.

Third, if it is the case that the orthogonal inputs are largely responsible for producing output layer cells that discriminate between the distal and proximal object spaces then an implication of this hy-

pothesis is that a model that incorporates distributed, i.e. overlapping, non-orthogonal, input layer cell representations of the distal and proximal object spaces should not be able to produce output layer cells that are capable of this discrimination. The simulation results presented in the following chapter clearly show that this is not the case, and that there is no significant qualitative difference between the learned behaviour of a model with orthogonal input layer cell representations of the distal and proximal object spaces, and a model with distributed input layer cell representations, of the distal and proximal object spaces.

In conclusion, this chapter presents a model that is the minimal architecture required in order for the output layer cells to learn to discriminate between distal and proximal visual cues presented to the model as input. A detailed exploration of the effects of altering the model and training environment parameters was conducted in a controlled manner. Under the correct conditions, the model can learn to produce output layer cells that respond exclusively to a narrow region of the space of distal objects. These cells reflect the current head direction of the agent and thus behave like head direction cells.

The model employs an orthogonal input representation of the distal and proximal visual cues in order to carefully control the nature of the input. There is strong evidence to suggest that some areas of the brain employ a distributed encoding of inputs, rather than the orthogonal encoding used in this chapter (Rolls and Treves, 1998; Dayan and Abbott, 2001). In the next chapter, a neural network model will be reported that is capable of learning to form separate representations of distal and proximal visual objects that are represented in a distributed, overlapping manner.

Chapter 3

Head Direction Cell Visual Responses: Distributed Input Representations

This chapter reports the implementation details, and simulation results, of a model that learns to discriminate between distal and proximal visual cues that are presented to the model in a distributed manner. The distributed input representation means that the current model has a harder computational task to solve than the model with orthogonal input representations that was reported in the previous chapter. Careful replication of the experiments conducted in the previous chapter demonstrates a general principle that is independent of the nature of the model input representation: the statistical decoupling between distal and proximal visual cues can be used to form separate representations of the visual cue spaces.

3.1 Model Architecture

The model contains a single layer of input cells connected by feedforward w_{ij} synapses to a single layer of output cells. These w_{ij} synapses will self-organize through competitive learning to produce

some output layer cells that learn to respond exclusively to the space of distal objects, and some output layer cells that learn to respond exclusively to a proximal object. The model architecture is very similar to the architecture of the model with orthogonal input representations depicted in Figure 2.1

The input cell layer comprises a common set of cells from which, during initialization of the network, are drawn M subsets of N cells. Each subset of input layer cells is defined as completely representing the possible visual input from either the space of distal objects, or a proximal object. Each object space will only be represented by one subset of input layer cells.

The main architectural difference between the current model and the model reported in the previous chapter is that, in the current model, the M input layer cell subsets are allowed to overlap and thus represent the distal and proximal object spaces in a distributed manner. A distributed representation is defined as the state where at least two subsets of input layer cells share a number, N_{OVERLAP}, of common input layer cells. A randomly chosen input layer cell j can therefore possibly be assigned to multiple input layer cell subsets, representing multiple object spaces.

The assignment of N input layer cells to one of the M subsets is achieved by random sampling without replacement from the common pool of input layer cells. After N input layer cells have been randomly assigned to a particular subset, the random sampling process is reset by replacing all the selected input cells into the common pool. The next N randomly selected input layer cells are then assigned to a different input layer cell subset by further random sampling without replacement. This process of random sampling without replacement within a subset, and random sampling with replace-

ment between the subsets, continues until all of the M subsets have been randomly assigned the requisite number N of input layer cells.

When there is an equal number, N_{SUBSET}, of cells randomly assigned to each of the M input layer cell subsets, then the proportion of the input cell overlap between the subsets is given by $N_{\mathrm{OVERLAP}}/N_{\mathrm{SUBSET}}$. When the overlap is non-zero, the random assignment of input layer cells to different subsets will produce a distributed representation of the distal and proximal visual objects in the input cell layer.

The visual inputs from the distal and proximal object spaces are represented by the M input layer cell subsets. All possible orientations, from $0^{\mathrm{o}} - 360^{\mathrm{o}}$, of the agent within its environment are represented by one subset. All possible head-centred bearings, -135^{o} to $+135^{\mathrm{o}}$, from the agent to each proximal object are represented by the remaining subsets. For any given input layer cell subset, a single packet of Gaussian activity within that subset will thus uniquely identify either the current orientation of the agent, or the current head-centred bearing from the agent to a proximal object, depending upon which object space the particular subset represents.

The number, 200, of input layer cells that constitute a subset is held constant across all experiments. The magnitude of the overlap between subsets is thus controlled by altering the size of the common pool of input layer cells from which individual input layer cells may be randomly assigned to a particular subset. For instance, with a pool of 2000 input layer cells from which individual input layer cells may be randomly assigned to subsets of 200 input layer cells, there would be an expected overlap,

N_{OVERLAP}, of 20 cells between two subsets. That is, the two subsets will share 10% of their randomly assigned input layer cells. The random allocation procedure is repeated with different random seeds until the desired number of overlapping cells, N_{OVERLAP}, between the subsets is achieved.

3.2 Implementation

The current model, with distributed input representations, is implemented in much the same way as the model with orthogonal input representations. This is described in Section 2.4. The main difference in implementation between the two models is how the firing rates r_j of the presynaptic input layer cells are set. In the model reported in this chapter, a presynaptic input layer cell j that has been assigned to more than one input layer cell subset will have a separate firing rate r_j in response to each of the object spaces that cell represents. Consequently, the overall firing rate r_j of that input cell is set, at any update of the model, to the maximum firing rate of that cell in response to the different object spaces represented by that cell, given the current training location and orientation of the agent.

3.3 Differences Between Models with Different Input Representations

In the model reported in Chapter 2, an individual input layer cell will only be assigned to one input layer cell subset representing one of the distal or proximal object spaces. Consequently, whenever a postsynaptic output layer cell i receives stimulation from a particular presynaptic input layer cell j, the stimulation from that input layer cell will always signal the presence of only one type of visual object space in the field-of-view of the simulated agent.

In the model reported in this chapter, whenever a postsynaptic output layer cell i receives stimula-

tion from a particular presynaptic input layer cell j that is one of the N_{OVERLAP} input layer cells that have been assigned to more than one input layer cell subset, then stimulation from that presynaptic input layer cell j will signal the presence of a different type of visual object space, distal or proximal, in the field-of-view of the agent at different updates of the model.

Both of these two models must solve the same general computational problem, which is learning to discriminate between distal and proximal objects that are simultaneously present in the field-of-view of the agent. For the model with distributed input representations, the problem is computationally more difficult because a postsynaptic output layer cell i must learn to not only discriminate between different object spaces, but also to discriminate between the occasions when a particular presynaptic input layer cell j is representing one particular object space, and the occasions when that same input layer cell is representing a different object space.

Consider the w_{ij} synapse between a particular input layer cell j, which has been assigned to represent more than one object space, and a particular postsynaptic output layer cell i. The presynaptic input layer cell j will have the same w_{ij} synapse with the postsynaptic output layer cell i no matter which object space the presynaptic input layer cell is currently representing. Consequently, any strengthening or weakening of that synapse will equally affect the presynaptic input layer cell representation of both the distal and proximal object spaces. For instance, it is not possible for the synapse to flip between a strong weight when the input layer cell represents the distal object space, and a weak weight when the input layer cell represents a proximal object. Thus, the presynaptic input layer cell j will represent the distal and proximal objects with equal strength. The distributed representation

Table 3.1: Network parameters for the simulations reported in Chapter 3

Network Parameters	
Total No. Input Cells	Varies by Experiment
No. Input Cells Assigned to Each Object Space Subset	200
No. Output Layer Cells	900
a (Sparseness of Output Layer Cell Firing)	0.1
Training Parameters	
No. Training Epochs	100
k (Learning Rate)	0.001
λ (Scaling of Input Cell Gaussian Response Profiles)	10.0
σ (Standard Deviation of Input Cell Gaussian Response Profiles)	5.0^{o}

All values are constant across all experiments reported in this chapter except where noted in the text.

of the distal and proximal objects in the input layer cells therefore adds a layer of complexity to the computational problem that must be solved.

3.4 Results

This section presents the results of exploring the effect that controlled alterations of the model parameters, and training environment, have upon the learning ability of the model.

3.4.1 Experiment 3.1: Demonstration of Core Model Performance

The simulations in this experiment were conducted to illustrate the core operational principles of the model with distributed input representations.

The training environment contained a single, centrally-located proximal object, together with ten training locations placed in a ring around the proximal object, and a continuous circular space of dis-

tal objects at an infinite distance from the training locations. The total number of input layer cells was set to 500. The proximal object space and the space of distal objects were each represented by overlapping subsets of 200 input layer cells. The two subsets will share an expected 40% of their assigned input layer cells. Simulation results are reported only from those simulations where the actual overlap between the input layer cell subsets equalled the expected overlap.

The network was simulated with the parameters given in Table 3.1, in accordance with the training and testing protocol outlined in Section 2.4. The simulation was conducted a total of five times with identical model parameters, but with different random synaptic weight initializations for each simulation.

Figure 3.1a displays the responses of eight naive, untrained output layer cells. The responses of the cells were recorded during a testing phase when no prior training phase had taken place. The plots show that the untrained output layer cells will respond to both the space of distal objects and the proximal object when probed with these object spaces during testing. The untrained responses of the output layer cells are also multi-modal: the response curves exhibit several defined peaks, rather than a single peak, in response to each object space.

The eight plots of Figure 3.1b display the learned responses of the same eight output layer cells. In comparison to the untrained responses, the majority of the output layer cells have learned to respond to either the space of distal objects or the proximal object, but not both. Moreover, the learned response curves exhibit single peaks and thus a defined response to a particular view of either the space

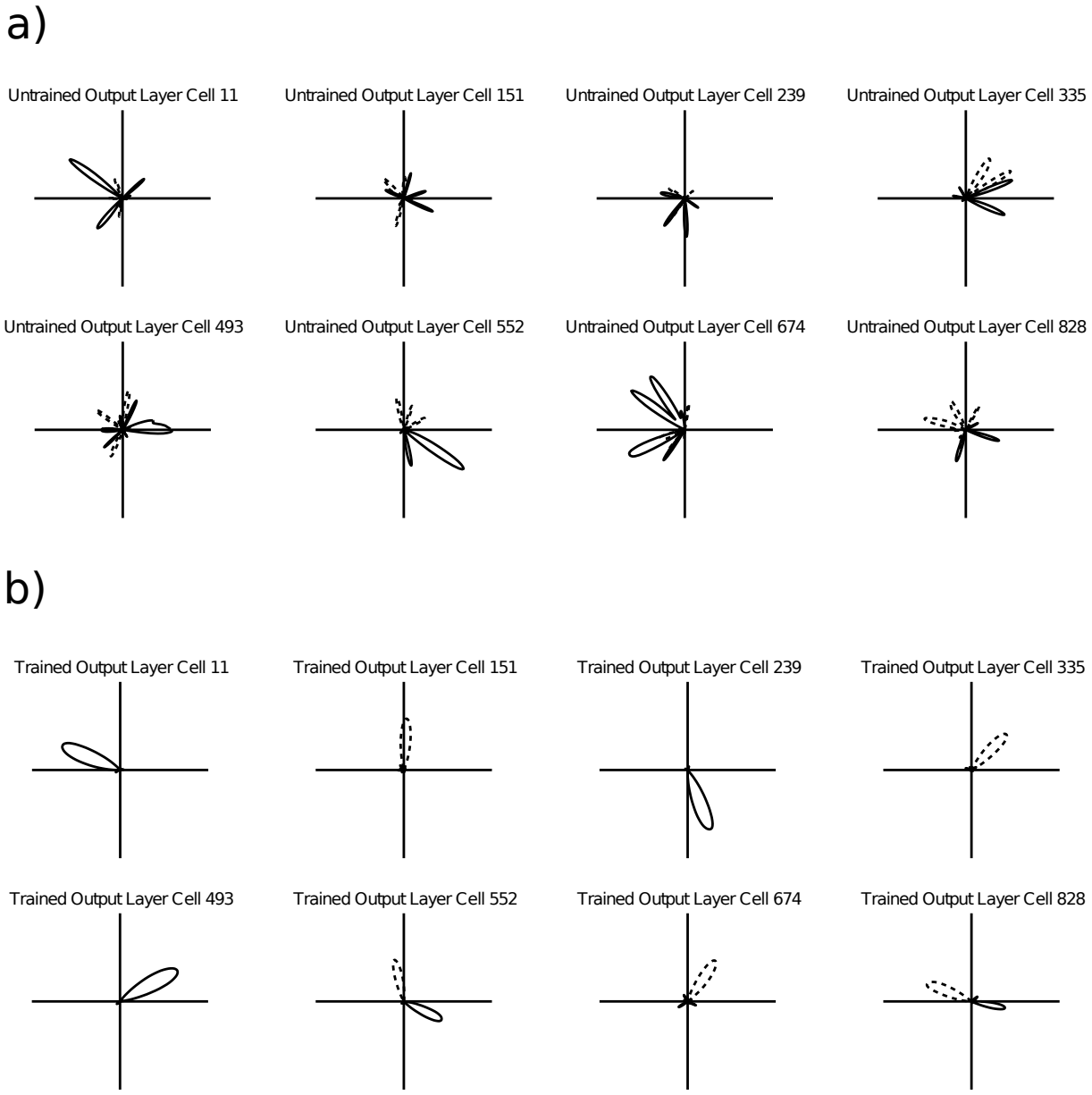

Figure 3.1: The responses of the output layer cells in Experiment 3.1 for a simulation with an overlap of 40% between the two input layer cell subsets. a) The untrained responses of eight output layer cells after a testing phase, but no learning phase, has taken place. The unbroken line represents the output layer cell response to the space of distal objects plotted over the interval $0^{o} - 360^{o}$. The dashed line represents the cell response to the proximal object plotted over the 270^{o} field-of-view of the simulated agent. It is clear that the naive output layer cells respond to both the distal and proximal object spaces. These response curves exhibit multiple peaks, i.e. are multi-modal. b) The responses of the same eight output layer cells as shown in (a), but after training has taken place. The majority of the learned cell responses are exclusive to the space of distal objects or the proximal object, but not both. A small number of output layer cells exhibit a learned response to a mixture of the distal and proximal object spaces. This is evident in the second and fourth plots in the bottom row. The learned response curves of the output layer cells exhibit a single prominent peak for each object space they respond to. This is in contrast to the untrained responses of the output layer cells. It is clear from comparing the trained and untrained cell responses that the training regime has a strong effect upon the nature of the output layer cell responses.

of distal objects or the proximal object. A small number of output layer cells do exhibit a learned response to both the distal and proximal object spaces. This is evident in the second and fourth plots of the bottom row in Figure 3.1b.

As can be seen from a visual comparison of the untrained and trained output layer cell response curves, the training phase has a direct influence upon the nature of the output layer cell responses. It is only after training that the output layer cells are able to effectively discriminate between the distal and proximal object spaces. Moreover, the majority of the output layer cells are able to achieve this discrimination when the distal and proximal object spaces are represented in an overlapping, distributed manner by the input layer cells. That is, 40% of the input layer cells were assigned to both the subset of input layer cells representing the space of distal objects, and the subset of input layer cells representing the proximal object. Furthermore, ten training locations are sufficient for the output layer cells to develop separate learned representations of the distal and proximal object spaces.

As a further measure of model performance, each individual output layer cell was classified by learned response type. An output layer cell can be classified into one of four mutually exclusive groups: responsive to the space of distal objects; responsive to the proximal object; responsive to a mixture of the distal and proximal object spaces; or does not show a response.

To be included in the distal-responsive group, an output layer cell must have a maximum firing rate > 0.075 in response to the space of distal objects. The output layer cell must also have a maximum firing rate in response to the proximal object that is $\leq 20\%$ of the maximum firing rate of that cell in

Table 3.2: The responses of the output layer cells in Experiment 3.1 both pre and post-training

Experiment 3.1 Output Layer Cell Responses				
	Untrained		**Trained**	
	Cells	**S.D.**	**Cells**	**S.D.**
No. Cells Responsive to Space of Distal Objects	7.4 (0.82%)	4.7	324.8 (36.09%)	7.4
No. Cells Responsive to Proximal Object	2.0 (0.22%)	1.0	467.8 (51.98%)	4.1
No. Cells Responsive to Both Object Spaces	890.6 (98.96%)	5.2	107.4 (11.93%)	6.6
No. Cells Responsive to Neither Object Space	0	0	0	0

The response criteria and the values used are described in the text (Section 3.4.1). The results displayed in the table are the average of five simulation runs for both the untrained and trained models. Each of the simulation runs was conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. In the untrained case, the output layer cells show a response to a mixture of the distal and proximal object spaces. After training, the majority of the output layer cells develop responses that are exclusive to either the space of distal objects or the proximal object, but not both.

response to the space of distal objects. For an output layer cell to be classified as proximal-responsive, that cell must have a maximum firing rate > 0.075 in response to the proximal object. The cell must also have a maximum firing rate in response to the space of distal objects that is $\leq 20\%$ of the maximum firing rate of the cell in response to the proximal object. The criterion for inclusion in the no response group is that the output layer cell must have a maximum firing rate ≤ 0.075 in response to both the distal and proximal object spaces. Any output layer cells that do not meet the three previous criteria are classified as showing a learned response to a mixture of the distal and proximal object spaces. The classification thresholds of 0.075 and $\leq 20\%$ were set by visual inspection of the output layer cell learned response curves and a comparison with the number of output layer cells classified into each group according to different values of the thresholds.

Table 3.2 displays the results of probing the model with the distal and proximal object spaces both be-

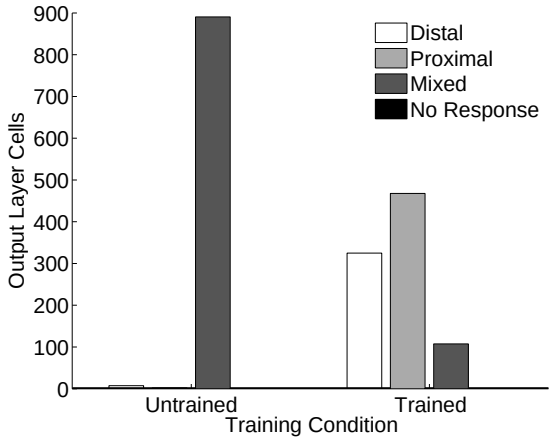

Figure 3.2: The responses of the output layer cells in Experiment 3.1. The bar chart displays the average number of output layer cells per response type calculated across five simulations with identical model parameters, but with different random synaptic weight initializations. Data are displayed for both an untrained model and a trained model. It is evident that training has an effect upon the behaviour of the model. In the untrained case, the majority of the output layer cells exhibit a response to a mixture of the distal and proximal object spaces. In the trained case, the majority of the output layer cells respond exclusively to either the space of distal objects or the proximal object, but not both. The displayed data are also reported in Table 3.2.

fore and after training has taken place. In all cases, the results shown are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The classification criteria used are those described above. Figure 3.2 displays this data in the form of a bar chart.

For the untrained, naive, model an average of 7.4 (0.82%) out of 900 output layer cells were classified as exhibiting an exclusive response to the space of distal objects. An average of 2.0 (0.22%) of the output layer cells showed an exclusive response to the proximal object. The majority, 890.6 (98.96%), of the output layer cells were classified as demonstrating a response to both the distal and proximal object spaces. None of the output layer cells were classified as unresponsive during testing.

After the training phase had taken place, an average of 324.8 (36.09%) of the output layer cells

were classified as exhibiting a learned response that is exclusive to the space of distal objects. An average of 467.8 (51.98%) of the output layer cells demonstrated an exclusive learned response to the proximal object. A small number, 107.4 (11.93%), of the output layer cells showed a learned response to a mixture of the distal and proximal object spaces. None of the trained output layer cells were classified as unresponsive during testing.

The learned responses of the trained output layer cells, as given in Figure 3.1b, Figure 3.2, and Table 3.2, clearly show that the model with distributed input representations can form separate output layer cell representations of the distal and proximal object spaces. In other words, individual output layer cells will exhibit learned responses that are exclusive to one of the distal or proximal object spaces, but not both.

The reason why the training phase has the effect it does upon the model performance was covered in Chapter 2 (Section 2.5.1) for the case of a model with orthogonal inputs. Thus, to avoid repetition, the details will not be covered here suffice to say that, when the network is trained at a large enough number of training locations (e.g. ten), a statistical decoupling occurs between the distal and proximal object spaces. This statistical decoupling is of a high enough degree that an associative competitive learning mechanism is able to learn to form separate output layer cell representations of the distal and proximal object spaces.

The ability of a neural network to discriminate between distributed, overlapping input layer cell representations of multiple object spaces is non-trivial. The w_{ij} synapse between a single input layer cell

representing more than one object space, and a single output layer cell, will have the same synaptic strength no matter which object space is currently driving the input cell to fire. This means that as the strength of the w_{ij} synapse increases or decreases it will affect the strength of the input layer cell signal for all the object spaces represented by that input layer cell. It is only if the level of statistical decoupling between the object spaces is high enough that an individual output layer cell will learn to be preferentially stimulated by input layer cells that represent either the space of distal objects or the proximal object, but not both. In this case, the computational problem of some of the input layer cells signalling the presence of both object spaces is overcome by the fact that, after training, the input layer cell signals representing the object space that the output layer cell has *not* learned to respond to will not be strong enough by themselves to stimulate the output layer cell into firing.

The results of the simulations conducted for this experiment have shown that a neural network can learn to form separate output layer cell representations of distal and proximal visual cues presented to the network as input when the distal and proximal cues are represented in a distributed, overlapping manner by the input layer cells. Qualitatively, the results of this experiment are very similar to the results of Experiment 2.1 in Chapter 2. This demonstrates that the ability of a neural network to form separate representations of the distal and proximal object spaces is not abolished by distributed input representations. The network can achieve the discrimination between the distal and proximal object spaces in an unsupervised manner, i.e. there is no teacher specifying how to categorize the inputs or which category to place an input into. It has also been shown how the network is capable of unsupervised categorization using only a single layer of synapses: in the current model specifically the w_{ij} synapses.

3.4.2 Experiment 3.2: The Effect of Varying the Input Layer Cell Subset Overlap

When there is a single proximal object in the training environment, there are only two input layer cell subsets. The simulations reported here were conducted to examine the effect upon the model performance of varying the percentage overlap of common input layer cells between the two input layer cell subsets.

Five different percentages of overlap were tested: 10%, 20%, 30%, 40%, and 50%. In each simulation, the training environment consisted of a single centrally-placed proximal object, ten training locations placed in a ring around the proximal object, and a circular space of distal objects at an infinite distance from the training locations. All remaining network parameters were set to the values given in Table 3.1.

Table 3.3 and Figure 3.3 display the average number of output layer cells classified by learned response type after training the model with different percentages of overlap between the input layer cell subsets. When there is an overlap of 10% between the input layer cell subsets, an average of 505.4 (56.16%) of the output layer cells exhibit a learned response that is exclusive to the space of distal objects. An average of 394.6 (43.84%) of the output layer cells were classified as showing a learned response that is exclusive to the single proximal object. None of the output layer cells demonstrate a learned response to both the distal and proximal object spaces. Nor are any of the output layer cells unresponsive.

When there is an overlap of 50% between the input layer cell subsets, an average of 300.4 (33.38%)

Table 3.3: The Learned Responses of the Output Layer Cells in Experiment 3.2

Experiment 3.2 Output Layer Cell Learned Responses										
	Percentage Overlap									
	10%		20%		30%		40%		50%	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	505.4 (56.16%)	3.6	438.4 (48.71%)	8.2	391.6 (43.51%)	6.5	324.8 (36.09%)	7.4	300.4 (33.38%)	3.2
Proximal	394.6 (43.84%)	3.6	461.6 (51.29%)	8.2	503.4 (55.93%)	6.5	467.8 (51.98%)	4.1	534.0 (59.33%)	8.1
Mixed Response	0	0	0	0	5.0 (0.56%)	4.6	107.4 (11.93%)	6.6	65.6 (7.29%)	6.6
No Response	0	0	0	0	0	0	0	0	0	0

The response criteria are the same as those described in Experiment 3.1 (Section 3.4.1). The results shown for each value of the percentage overlap are the average of the results from five simulations conducted with identical model parameters, but with different synaptic weight initializations. S.D. = standard deviation. As the percentage overlap between the input layer cell subsets increases, so the number of output layer cells that lean to respond to a mixture of the distal and proximal object spaces also increases. In all cases, however, the great majority of the output layer cells can still discriminate between the distal and proximal object spaces.

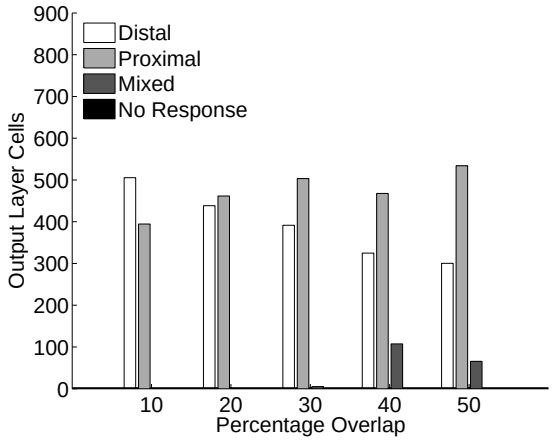

Figure 3.3: The learned responses of the output layer cells in Experiment 3.2. For each of the five different percentage overlaps between the input layer cell subsets, the bar chart displays the average number of output layer cells per response type. The averages are calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The data for the averages are given in Table 3.3. As the percentage overlap between the input layer cell subsets increases, so the majority of the output layer cells still learn to respond exclusively to either the space of distal objects or the proximal object. Increasing the percentage overlap between the input layer cell subsets does, however, produce a small increase in the number of output layer cells that learn to respond to a mixture of the distal and proximal object spaces.

of the output layer cells exhibit a learned response that is exclusive to the space of distal objects. An average of 534.0 (59.33%) of the output layer cells show a learned response to the single proximal object, and an average of 65.6 (7.29%) of the output layer cells demonstrate a learned response to both object spaces.

For the current experimental setup, varying the percentage overlap in the interval $[10\%, 50\%]$ has little effect upon the overall performance of the model. That is, the majority of the output layer cells learn to respond exclusively to a narrow region of either the space of distal objects or the proximal object, but not both, regardless of the magnitude of the overlap between the input cell subsets. As will be seen in the experiments reported below, this result generally holds true for all the simulations conducted for this chapter.

3.4.3 Experiment 3.3: The Effect of Varying the Number of Training Locations

The simulations in this experiment were conducted to examine the effect that the number of training locations in the training environment – which can be loosely correlated to the amount of movement of an agent through its environment – has upon the learned behaviour of a model with distributed input representations.

The training environment consisted of a single, centrally located proximal object together with a continuous circular space of distal objects at an infinite distance from the proximal object. The number of training locations in the training environment was varied as follows: 1, 2, 5, and 10. The network was simulated with the parameter values given in Table 3.1. For each distinct number of training locations, the percentage of overlap between the input layer cell subsets was varied in the interval $[10\%, 50\%]$ in an identical manner to Experiment 3.2.

Table 3.4 and Figure 3.4 display the distribution of the output layer cell learned responses. For each distinct number of training locations, the table displays the results for a simulation conducted with a 10% overlap between the input layer cell subsets, and for a simulation conducted with a 40% overlap between the subsets.

With only 1 or 2 training locations in the training environment, the majority of the output layer cells show a learned response to both the distal and proximal object spaces. This is true for overlaps of 10% and 40% between the input layer cell subsets.

Table 3.4: The learned responses of the output layer cells in Experiment 3.3

Experiment 3.3 Output Layer Cell Learned Responses								
	10% **Overlap** **No. Training Locations**							
	1		**2**		**5**		**10**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	146.4 (16.27%)	3.3	14.8 (1.64%)	3.7	441.6 (49.07%)	15.1	505.4 (56.16%)	3.6
Proximal	0.2 (0.02%)	0.4	0.2 (0.02%)	0.4	427.6 (47.51%)	10.4	394.6 (43.84%)	3.6
Mixed Response	640.0 (71.11%)	3.8	718.0 (79.78%)	6.2	18.8 (2.09%)	8.5	0	0
No Response	113.4 (12.60%)	5.3	167.0 (18.56%)	3.2	12.0 (1.33%)	3.9	0	0
	40% **Overlap** **No. Training Locations**							
	1		**2**		**5**		**10**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	146.6 (16.29%)	2.6	14.4 (1.6%)	3.1	315.6 (35.07%)	6.2	324.8 (36.09%)	7.4
Proximal	0.4 (0.04%)	0.9	0	0	434.2 (48.24%)	1.8	467.8 (51.98%)	4.1
Mixed Response	743.4 (82.60%)	4.6	884.2 (98.24%)	2.9	150.2 (16.69%)	5.8	107.4 (11.93%)	6.6
No Response	9.6 (1.07%)	4.5	1.4 (0.16%)	0.9	0	0	0	0

The response criteria used for the classifications are identical to those described in Experiment 3.1 (Section 3.4.1). The results shown are the average of five simulations conducted with identical model parameters, but with different synaptic weight initializations. S.D. = standard deviation. As the number of training locations increases from 1 to 10, so the nature of the learned response in the output layer cells changes. With a small number, 1 or 2, of training locations, the majority of the output layer cells learn to respond to both of the distal and proximal object spaces. With 1 training location, a small number of output layer cells respond exclusively to the space of distal objects. This result is an artefact of the experimental setup, and is explained fully in the text. When the number of training locations is ≥ 5, the majority of the output layer cells learn to respond to one of the distal or proximal object spaces, but not both. This is true for overlaps of both 10% and 40% between the input layer cell subsets.

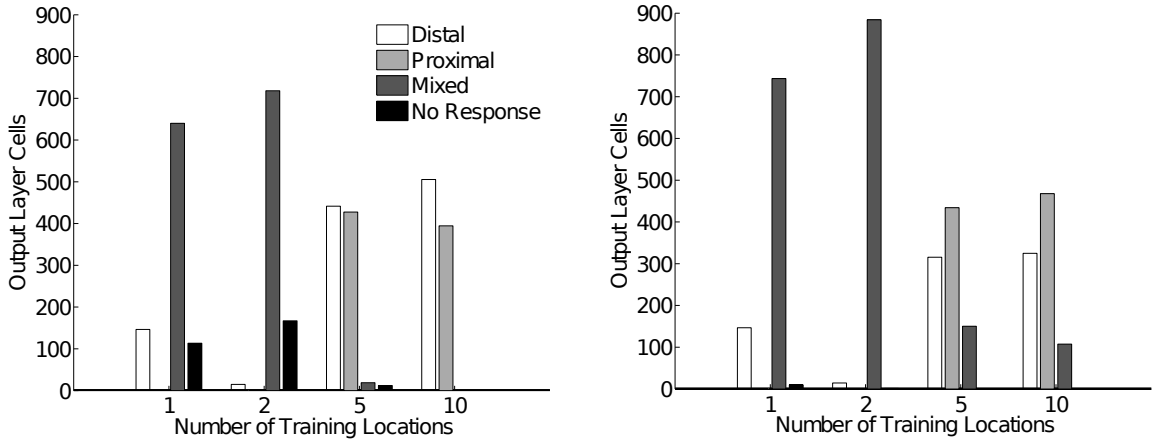

Figure 3.4: The learned responses of the output layer cells in Experiment 3.3. The data displayed are the average number of output layer cells per response type for each of the five different numbers of training locations in the training environment. The averages are calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The left-hand bar chart displays data recorded from a simulation conducted with an overlap of 10% between the input layer cell subsets. The right-hand bar chart displays data recorded from a simulation conducted with an overlap of 40% between the input layer cell subsets. As the number of training locations in the training environment increases, so the majority of the output layer cells change from learning to represent a mixture of the distal and proximal object spaces when trained at a small number of training locations, to learning to exclusively represent either the space of distal objects or the proximal object when trained at a larger number of training locations. This result holds for simulations conducted with both a 10% and a 40% overlap between the input layer cell subsets. The data are also reported in Table 3.4.

With an overlap of 10%, averages of 640.0 (71.11%) and 718.0 (79.78%) output layer cells exhibit a learned response to both of the distal and proximal object spaces when trained with 1 and 2 training locations respectively. When the overlap is 40%, averages of 743.4 (82.60%) and 884.2 (98.24%) output layer cells learn to respond to both of the distal and proximal object spaces for 1 and 2 training locations respectively.

A small number of training locations will not produce a high enough statistical decoupling between the distal and proximal object spaces for individual output layer cells to develop separate representations of these object spaces through associative competitive learning. In other words, a small number of training locations promotes a high degree of statistical coupling between the distal and proximal

object spaces. Consequently, there is a low probability of an individual output layer cell learning to discriminate between the two object spaces. This is true whether the input representation is orthogonal or distributed.

When there is one training location in the training environment, an average of 146.4 (16.27%) output layer cells for the 10% overlap, and 146.6 (16.29%) output layer cells for the 40% overlap, were classified as exhibiting a learned response that is exclusive to the space of distal objects. As described in Experiment 2.2 (Chapter 2, Section 2.5.2), this result is an artefact of the experimental setup. Because the agent has a 270^{o} field-of-view, there will be a 90^{o} interval during a single 360^{o} rotation of the agent where the proximal object will not be visible to the agent, but the space of distal objects will be visible. This produces a small statistical decoupling between the distal and proximal object spaces that is sufficient for some of the output layer cells to develop exclusive learned responses to the space of distal objects. Increasing the number of training locations to 2 removes this artefact.

With five training locations in the training environment, and an overlap of 10% between the input layer cell subsets, an average of 441.6 (49.07%) of the output layer cells exhibit a learned response that is exclusive to the space of distal objects, and an average of 427.6 (47.51%) of the output layer cells show a learned response that is exclusive to the proximal object. Under these parameters, an average of only 18.8 (2.09%) of the output layer cells demonstrate a response to both of the distal and proximal object spaces. Similar results were obtained with an overlap of 40% between the input layer cell subsets.

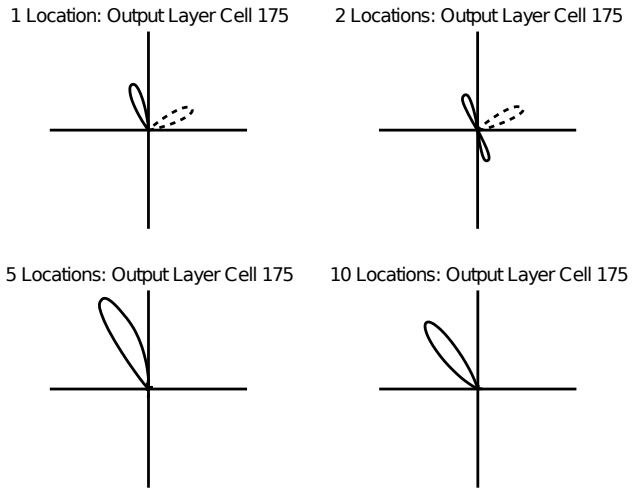

Figure 3.5: The learned response of the output layer cells in Experiment 3.3. Each one of the four plots shows the same output layer cell recorded from a simulation conducted with a constant overlap of 40% between the input layer cell subsets, and a different number of training locations in the training environment in each case. The four simulations were conducted with the same synaptic weight initializations. The training locations are varied as follows: top-left, one location; top-right; two locations; bottom-left, five locations; bottom-right, ten locations. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. As the number of training locations increases, so the nature of the learned representation in the output layer cells changes. With a small number, 1 or 2, of training locations, the majority of the output layer cells learn to respond to a mixture of the distal and proximal object spaces. This is shown in the two plots in the top row of the figure. As the number of training locations increases, output layer cells develop that have learned to respond exclusively to one of the distal or proximal object spaces using a grandmother-cell-like representation, where individual output layer cells respond to a narrow region of either the space of distal objects or the proximal object space, but not both. This is shown in the bottom two plots of the figure.

With ten training locations in the training environment, and an overlap of 10% between the input layer cell subsets, none of the output layer cells showed a learned response to a mixture of the distal and proximal object spaces. All of the output layer cells show a learned response that is exclusive to either the space of distal objects or the proximal object, but not both. With ten training locations, and an overlap of 40%, an average of 107.4 (11.93%) of the output layer cells exhibited a learned response to both of the distal and proximal object spaces.

The output layer cell learned response curves are displayed in Figure 3.5. The response curves were recorded during the testing phase when the network was probed with the distal object space followed

by the proximal object. There are four plots in total, representing the four simulations, each with a different number of training locations in the training environment. Visual comparison of the output layer cell response curves from the simulations conducted with overlaps of 10% and 40% between the input layer cell subsets revealed little qualitative difference between these response curves. Therefore, the plots in Figure 3.5 display the learned response curves of four output layer cells taken from a simulation conducted with an overlap of 40% between the input layer cell subsets.

The results of this experiment qualitatively resemble the results reported in Experiment 2.2 (Chapter 2, Section 2.5.2). With a single proximal object in the training environment, only a small increase in the number of training locations, ≥ 5, is required to produce a high enough level of statistical decoupling between the distal and proximal object spaces such that the output layer cells will learn to discriminate between the two object spaces. The current model can solve this particular learning problem despite having the harder computational task: i.e. to form separate representations of the distal and proximal object spaces when those spaces are represented in a distributed, overlapping manner in the input layer.

The nature of the learned representation that develops in the output layer cells is direcrtly influenced by an interaction between the number of training locations in the training environment and the distal and proximal object spaces. The mechanisms of this interaction were discussed in Experiment 2.2 (Chapter 2, Section 2.5.2). In summary, the level of statistical decoupling between the distal and proximal object spaces is directly proportional to the number of training locations in the training environment. This corresponds to a requirement that the agent must move through its environment in

order to produce separate representations of the distal and proximal object spaces.

In a natural environment, with multiple proximal objects present, there would not be discrete training locations as is the case for the simulations conducted in this chapter. Instead, there would be a continuum of training locations as the animal travels through its environment. This continuum of training locations will provide a high degree of statistical decoupling between the distal and proximal object spaces, such that the neural network presented here will able to discriminate between the multiple distal and proximal objects through associative competitive learning.

3.4.4 Experiment 3.4: The Sparseness of Firing within the Output Layer Cells

The sparseness value, a, as defined by Equation (2.5) in Chapter 2, effectively determines the level of competition between the output layer cells. The simulations conducted for this experiment explored the effect that the value of a has upon the performance of the model with distributed input representations.

The training environment for all simulations comprised ten training locations placed around a single centrally-located proximal object, together with a continuous circular space of distal objects located at an infinite distance from the training locations. The network was simulated with the parameter values given in Table 3.1, except for the value of the sparseness, a, which was varied as follows: 0.01, 0.05, 0.1, 0.2, 0.3, 0.4. For each distinct value of a, the network was simulated with different percentage overlaps between the input layer cell subsets. The percentage overlap was varied in the interval $[10\%, 50\%]$ in the manner described in Experiment 3.2 in this chapter.

Table 3.5: The learned responses of the output layer cells in Experiment 3.4

Experiment 3.4 Output Layer Cell Learned Responses												
	10% **Overlap** **Sparseness** a											
	0.01		**0.05**		**0.1**		**0.2**		**0.3**		**0.4**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	58.4 (6.49%)	3.4	449.0 (49.89%)	8.9	505.4 (56.16%)	3.6	186.4 (20.71%)	2.3	159.4 (17.71%)	4.4	192.0 (21.33%)	8.0
Proximal	51.0 (5.67%)	2.6	215.4 (23.93%)	6.5	394.6 (43.84%)	3.6	359.4 (39.93%)	9.1	227.8 (25.31%)	7.7	165.6 (18.40%)	9.3
Mixed Response	12.8 (1.42%)	4.1	0	0	0	0	354.2 (39.36%)	9.3	511.6 (56.84%)	9.9	488.4 (54.27%)	9.3
No Response	777.8 (86.42%)	3.4	235.6 (26.18%)	10.5	0	0	0	0	1.2 (0.13%)	1.3	54.0 (6.00%)	2.2
	40% **Overlap** **Sparseness** a											
	0.01		**0.05**		**0.1**		**0.2**		**0.3**		**0.4**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	137.8 (15.31%)	4.1	495.2 (55.02%)	9.8	324.8 (36.09%)	7.4	158.2 (17.58%)	3.2	172.8 (19.20%)	8.2	250.6 (27.84%)	9.3
Proximal	98.4 (10.93%)	4.7	398.4 (44.27%)	14.2	467.8 (51, 98%)	4.1	301.2 (33.47%)	3.4	178.6 (19.84%)	4.3	130.8 (14.53%)	3.4
Mixed Response	4.2 (0.47%)	1.1	0	0	107.4 (11.93%)	6.6	440.6 (48.96%)	5.7	540.6 (60.07%)	10.6	454.6 (50.51%)	12.0
No Response	659.6 (73.29%)	7.4	6.4 (0.7%)	4.6	0	0	0	0	8.0 (0.89%)	2.0	64.0 (7.11%)	6.0

The criteria for the response classifications are identical to those described in Experiment 3.1 (Section 3.4.1). Each result shown is the average of the results from five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. As the sparseness value, a, is increased, so the nature of the learned representation in the output layer cells changes. With a low value of a, such as 0.01, the majority of the output layer cells are unresponsive after training. With an intermediate value of a, such as 0.1, most output layer cells have learned to respond to either the distal object space or the proximal object space, but not both. With a high value of a, such as 0.4, the majority of the output layer cells learn to respond to both of the distal and proximal object spaces. This trend is common to the simulations conducted with 10% and 40% overlaps between the input layer cell subsets.

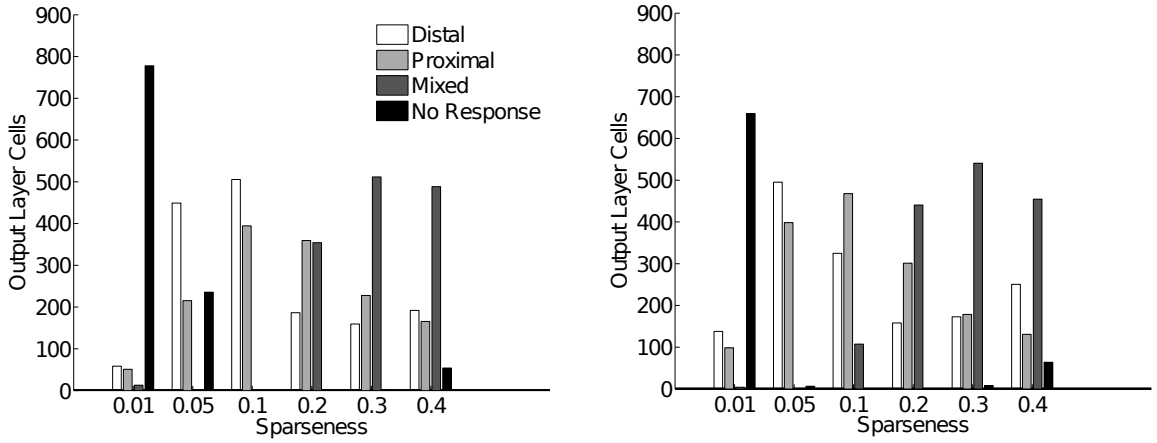

Figure 3.6: The learned responses of the output layer cells in Experiment 3.4. For each value of the sparseness a, the bar charts display the average number of output layer cells per response type. The averages are calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The left-hand bar chart displays the data recorded from a simulation conducted with an overlap of 10% between the input layer cell subsets. The right-hand bar chart displays the data recorded from a simulation conducted with an overlap of 40% between the input layer cell subsets. When the sparseness a is set to an intermediate value, the majority of the output layer cells learn to respond exclusively to either the space of distal objects or the proximal object. When a is set to a higher value, the majority of the output layer cells learn to respond to a mixture of the distal and proximal object spaces. This behaviour occurs when there are overlaps of both 10% and 40% between the input layer cell subsets. The data displayed in the bar charts are also reported in Table 3.5.

Table 3.5 and Figure 3.6 display the output layer cells classified by response type in accordance with the criteria described in Experiment 3.1 (Section 3.4.1). For each unique value of a, the table displays the results from simulations conducted with an overlap of 10% between the input layer cell subsets, and also the results from simulations conducted with an overlap of 40% between the subsets.

With an overlap of 10% between the input layer cell subsets, an increase in the sparseness a changes the nature of the learned output layer cell representation. With a low or intermediate value of the sparseness a, there is a local, grandmother-cell-like representation where individual output layer cells respond exclusively to a narrow region of either the space of distal objects or the proximal object space, but not both. With a high value of the sparseness a, there is a distributed representation where

individual output layer cells respond to multiple narrow regions of both of the distal and proximal object spaces. The number of output layer cells that are classified as unresponsive decreases as a increases.

This trend is also the case for simulations conducted with a 40% overlap between the input layer cell subsets. As the magnitude of the percentage overlap has little effect upon the performance of the model when simulated with different values of a, the following analysis applies to all models simulated with a percentage overlap in the interval $[10\%, 50\%]$. However, only simulations conducted with an overlap of 40% will be directly considered.

Each one of the six plots in Figure 3.7 displays the same output layer cell recorded from one of the six simulations conducted with a unique value of the sparseness, a, but with identical synaptic weight initializations.

When the value of a is 0.01, approximately 1% of the output layer cells will be permitted to fire at any given update of the model. Thus, in the simulations conducted with a 40% overlap between the input layer cell subsets, the majority, 659.6 (73.29%), of the output layer cells remain unresponsive to the distal and proximal object spaces after training. This is shown in Table 3.5, Figure 3.6, and the top-left plot of Figure 3.7.

Increasing the value of a to be either 0.05 or 0.1 means that approximately 5% or 10%, respectively, of the output layer cells will be permitted to fire at any given update of the model. This will

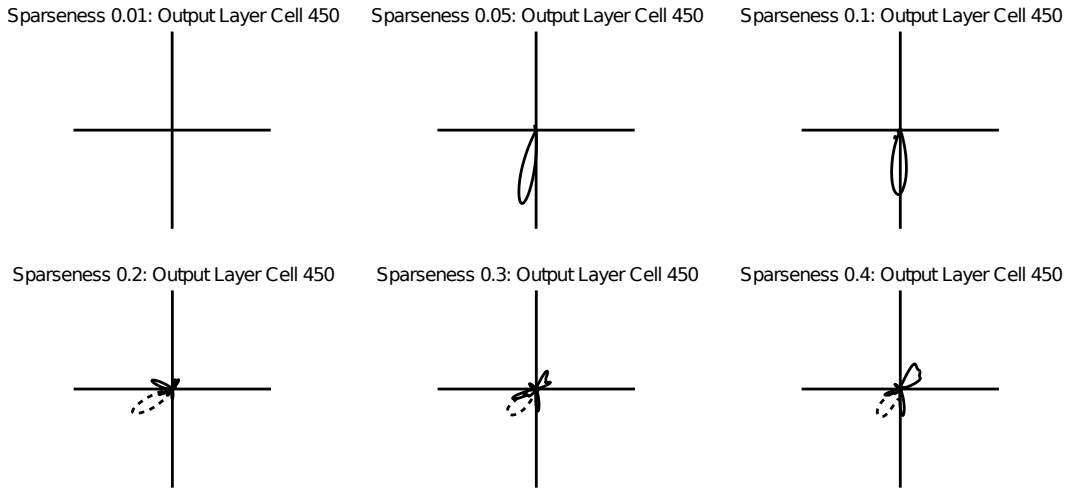

Figure 3.7: The learned responses of the output layer cells in Experiment 3.4. Each of the six plots displays an output layer cell taken from a simulation conducted with a constant overlap of 40% between the input layer cell subsets and a different value of the sparseness a. The values of a are varied as follows: top row, left to right, a = 0.01, 0.05, 0.1; bottom row, left to right, a = 0.2, 0.3, 0.4. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. The polar plots display the same output layer cell recorded from simulations with different values of a, but with identical synaptic weight initializations. As the value of a increases, the number of output layer cells that learn to respond to both of the distal and proximal object spaces increases. The firing profiles also tend to become broader and weaker as a increases. It is clear from the plots that as the sparseness value, a, increases, so the nature of the learned representation in the output layer cells changes from a local, grandmother-cell-like representation, where individual output layer cells respond exclusively to a narrow region of either the space of distal objects or the proximal object space, into a distributed representation where individual output layer cells respond to multiple regions of both of the distal and proximal object spaces.

result in more of the output layer cells developing a learned response to either the space of distal objects or the proximal, but not both. This is shown in Table 3.5, Figure 3.6, and the top-middle and top-right plots of Figure 3.7. The optimal value of a for producing cells that exhibit an exclusive, grandmother-cell-like response to the space of distal objects is thus within the interval $[0.05, 0.1]$.

When the a value is 0.2, an average of 440.6 (48.96%) of the output layer cells learn to respond to both of the distal and proximal object spaces. This is shown in Table 3.5, Figure 3.6, and the bottom-left plot of Figure 3.7. If an output layer cell is chosen at random from the sub-population of output layer cells that have learned to respond to a mixture of the object spaces, then the firing of

that single output layer cell cannot, by itself, discriminate between the distal and proximal objects. If, however, each of the output layer cells in this sub-population learns to respond maximally to different random parts of the distal and proximal object spaces, then the entire layer of output cells will still be able to discriminate between the object spaces using a distributed population coding.

The reasons why a population coding would work in this situation were discussed in Experiment 2.3 (Chapter 2, Section 2.5.3). The main details of this argument will thus be summarised here. If there is no regular relationship between the responses of the output layer cells to the space of distal objects and the responses to the proximal object, then each part of the distal and proximal object spaces will be represented by a unique pattern of firing across the whole layer of output cells. This is because if an output layer cell is chosen at random from the whole population of output layer cells, and the response curve of that cell to the space of distal objects is known, then it will not be possible to predict the response curve of that cell to the proximal object on the basis of this information. Consequently, there will be a unique pattern of firing across the whole population of output layer cells for each unique orientation of the agent to the space of distal objects, and also a unique patten of firing for each head-centred bearing from the agent to the proximal object. Thus, the orientation of the agent, or the bearing of the proximal object, can be determined by the unique pattern of firing or population code across all the output layer cells.

This is demonstrated by the two plots in Figure 3.8. The output layer cells displayed in these plots are taken from a simulation with an a value of 0.2. In each of the two plots, the output layer cells respond maximally to the same particular orientation of the agent to the space of distal objects. How-

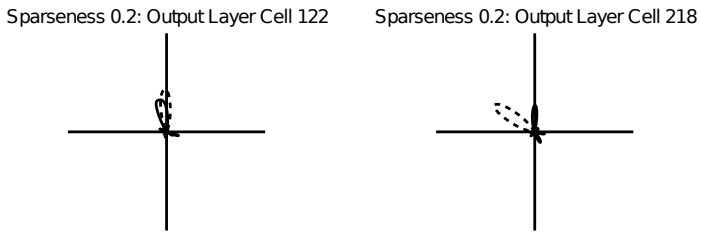

Figure 3.8: The learned responses of two output layer cells from a simulation in Experiment 3.4 conducted with a 40% overlap between the input layer cell subsets and a sparseness value of 0.2. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. The two output layer cells respond maximally to the same particular orientation of the agent with respect to the space of distal objects. However, the two cells respond maximally when the proximal object lies on different head-centred bearings from the agent. Thus, there is no fixed relationship between the response of an output layer cell to the space of distal objects and the response of the same cell to the proximal object. The implication of this is that there is likely to be a unique pattern of firing across the whole population of output layer cells for each orientation of the agent to the space of distal objects, as well as a unique pattern of firing for each head-centred bearing from the agent to the proximal object. The orientation of the agent, or the position of the proximal object, can be determined from this unique pattern of firing across the output layer cells.

ever, each of the two output layer cells responds maximally to a different head-centred bearing from the agent to the proximal object. There is thus no predictable relationship between the response of an output layer cell to the space of distal objects and the response of that same cell to the proximal object.

As the value of a increases, in the case of the current simulations to be 0.3 or 0.4, the majority of the output layer cells learn to respond to both of the distal and proximal object spaces. This is a result of a higher a value permitting a greater number of output layer cells to fire at any given update of the model and thus update their w_{ij} synapses with the input layer cells. This is shown in Table 3.5 and the bottom-middle and bottom-right plots of Figure 3.8. Because the modelling hypothesis is that individual output layer cells will learn to discriminate between the distal and proximal object spaces, increasing the value of thes sparseness a leads to the model learning an incorrect output layer representation.

The results of this experiment are qualitatively similar to the results of the same experiment performed on a model with orthogonal input representations. The sparseness of firing in the output layer affects the nature of the learned representation that develops in the output layer cells. The optimal value of the sparseness a, within the interval $[0.05, 0.1]$, is the same for the current model and for the model with orthogonal input representations. Thus, the effect that the sparseness of firing, a, has upon the learned behaviour of the model is independent of the nature of the input representation to the model.

3.4.5 Experiment 3.5: The Number of Proximal Objects and Model Performance

The simulations conducted in this experiment examined the effect upon the model performance of increasing the number of proximal objects in the training environment. The number of proximal objects was varied across the simulations as follows: 2, 3, 4, and 5. For each simulation, the percentage overlap between the input layer cell subsets was varied in the interval $[10\%, 50\%]$ as described in Experiment 3.2 (Section 3.4.2). The network parameters were set according to the values given in Table 3.4.

Table 3.6 and Figure 3.9 display the results of classifying the output layer cells by learned response. An output layer cell is classified as responding exclusively to the space of distal objects if its firing rate in response to the space of distal objects is > 0.075, and its firing rate in response to any of the proximal objects is $\leq 20\%$ of its maximum firing rate in response to the space of distal objects. An output layer cell is classified as responding exclusively to one of the proximal object spaces if its firing rate in response to that space is > 0.075, and its firing rate in response to the space of distal objects, and any of the remaining proximal object spaces, is $\leq 20\%$ of its maximum firing rate in response to

Table 3.6: The learned responses of the output layer cells in Experiment 3.5

Experiment 3.5 Output Layer Cell Learned Responses								
	10% Overlap No. Proximal Objects							
	2		3		4		5	
	Cells	S.D.	Cells	S.D.	Cells	S.D.	Cells	S.D.
Distal	234.8 (26.09%)	8.5	56.2 (6.24%)	2.6	0.2 (0.02%)	0.4	4.6 (0.51%)	1.4
Proximal	199.0 (22.11%)	11.2	50.6 (5.62%)	1.2	1.6 (0.18%)	1.0	0	0
Mixed Response	413.4 (45.93%)	16.9	782.2 (86.91%)	3.5	872.2 (96.91%)	5.1	836.8 (92.98%)	3.8
No Response	52.8 (5.87%)	3.1	11.0 (1.22%)	1.3	26.0 (2.98%)	4.6	58.6 (6.51%)	3.5
	40% Overlap No. Proximal Objects							
	2		3		4		5	
	Cells	S.D.	Cells	S.D.	Cells	S.D.	Cells	S.D.
Distal	358.8 (39.87%)	19.1	37.6 (4.18%)	4.4	0	0	0	0
Proximal	196.8 (21.97%)	1.9	67.4 (7.48%)	4.5	0	0	0	0
Mixed Response	264.2 (29.36%)	19.2	746.4 (82.93%)	10.0	900.0 (100.00%)	0	900.0 (100.00%)	0
No Response	80.2 (8.91%)	5.2	48.6 (5.40%)	2.1	0	0	0	0

The classification criteria used are described in the text. Each of the results displayed is the average of the results of five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. As the number of proximal objects in the training environment increases, so the nature of the output layer cell learned representation changes. With an overlap of 10% between the input layer cell subsets, and the number of proximal objects ≥ 4, the output layer cells almost exclusively employ a distributed representation where individual output layer cells respond to multiple regions of both of the distal and proximal object spaces. With an overlap of 40% between the input layer cell subsets, and the number of proximal objects ≥ 4, the output layer cells also exclusively employ a distributed representation.

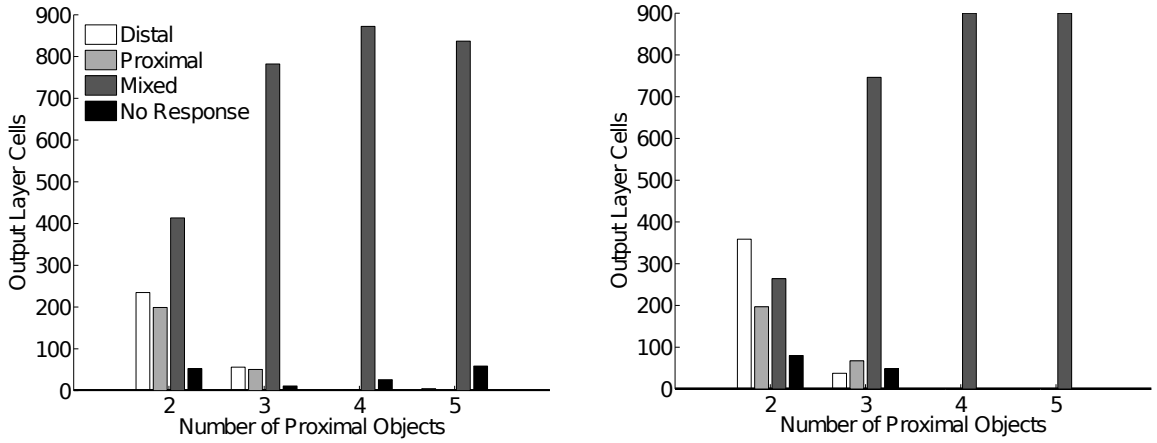

Figure 3.9: The learned responses of the output layer cells in Experiment 3.5. The average number of output layer cells per response type are calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The left-hand bar chart displays data recorded from a simulation conducted with an overlap of 10% between the input layer cell subsets. The right-hand bar chart displays data recorded from a simulation conducted with an overlap of 40% between the input layer cell subsets. As the number of proximal objects in the training environment increases, the majority of the output layer cells learn to respond to a mixture of the distal and proximal object spaces. This result holds for simulations conducted with an overlap of 10% between the input layer cell subsets, and for simulations conducted with an overlap of 40% between the input layer cell subsets. The data displayed are also reported in Table 3.6.

the proximal object space in question. The criteria for an output cell layer to be classified as either exhibiting no response or a response to a mixture of the object spaces are given in Experiment 3.1 (Section 3.4.1).

The table displays the results of varying the number of proximal objects in the training environment for a model simulated with a 10% overlap between the input layer cell subsets, and for a model simulated with a 40% overlap between the input layer cell subsets. Output layer cells that are classified as responding exclusively to one of the proximal object spaces are pooled together, rather than identifying which particular proximal object space the output layer cell responds to.

For the model simulated with a 10% overlap between the input layer cell subsets, two proximal objects

in the training environment results in 413.4 (45.93%) of the output layer cells learning to respond to a mixture of the distal and proximal object spaces. Increasing the number of proximal objects to three nearly doubles the number, 782.2 (86.91%), of output layer cells that exhibit a learned response to a mixture of the distal and proximal object spaces. Further increasing the number of proximal objects to four or five leads to $\geq 90\%$ of the output layer cells showing a learned response to a mixture of the distal and proximal object spaces.

When the model is simulated with an overlap of 40% between the input layer cell subsets, and two proximal objects in the training environment, 264.2 (29.36%) of the output layer cells learn to respond to a mixture of the distal and proximal object spaces. In a similar manner to the model with the 10% overlap, increasing the number of proximal objects from two to three leads to double the number of output layer cells, 746.4 (82.93%), learning to respond to a mixture of the object spaces. With four or five proximal objects in the training environment, all of the 900 output layer cells learn to respond to a mixture of the distal and proximal object spaces.

Thus, increasing the number of proximal objects in the training environment leads, for both the 10% and 40% overlaps, to a change in the nature of the output layer cell learned representation. With the current experimental setup, a small number of proximal objects (≤ 2) will still produce output layer cells that exhibit a learned response that is exclusive to either the space of distal objects or one of the proximal objects, but only to one object space. The output layer cells that exhibit an exclusive learned response are thus employing a local, grandmother-cell-like representation of the input stimuli where the individual output layer cell only responds to a narrow region of one object space. As the number

of proximal objects increases to be ≥ 3, the output layer cells change to employ a fully distributed representation of the input stimuli where an individual output layer cell responds to multiple regions of a mixture of the object spaces.

When there are ≥ 3 proximal objects in the training environment, an overlap of 40% between the input layer cell subsets produces more pronounced results, compared to an overlap of 10%. That is, with an overlap of 40% all of the output layer cells learn to respond to a mixture of the distal and proximal object spaces. The reason for this difference is that a greater percentage overlap between the input layer cell subsets leads to a greater chance of a randomly chosen input layer cell being assigned to several, if not all, of the possible subsets. Thus, with a larger number of proximal objects in the training environment, and a high percentage of overlap between the input layer cell subsets, the output layer cells are less likely to be able to form a grandmother-cell-like representation in the input stimuli. This can be confirmed by examination of the results in Table 3.6 and Figure 3.9.

With a higher number, ≥ 3, of proximal objects in the training environment, the model learns a distributed, incorrect, output layer representation. An increase in the statistical decoupling between the distal and proximal object spaces is required in order for the output layer to learn the correct, local, representation. When a local representation develops, individual output layer cells will respond exclusively to a narrow region of the space of distal objects, and these cells will thus behave like head direction cells.

As discussed in Section 2.5.4 of Chapter 2, a possible reason why the model performance deterio-

rates as the number of proximal objects in the training environment increases is that, with the current training protocol, increasing the number of proximal objects actually *decreases*, rather than increases, the statistical decoupling between the distal and proximal object spaces. Exactly why the model performance deteriorates as more proximal objects are added to the training environment, however, remains unclear.

3.4.6 Experiment 3.6: The Model Performance with Less Training

The simulations comprising this experiment were conducted to investigate how the model performs when exposed to less training. The model with orthogonal input representations can produce output layer cells that respond exclusively to either the space of distal objects or the space of proximal objects, but not both, after as little as 10 training epochs (Experiment 2.5, Chapter 2). In the current model, the overlapping distributed nature of the input representations may have an effect upon the amount of time it takes for the model to learn the correct mature behaviour. The network was simulated undergoing 1, 5, 10, and 25 training epochs. The network parameters are given in Table 3.4.

Table 3.7 and Figure 3.10 display the output layer cells classified by learned response type. When the model is simulated with a 10% overlap between the input cell subsets, the learned behaviour of the model approaches the behaviour of the fully mature model after only 25 training epochs. That is, 415.2 (46.13%) of the output layer cells learn to respond exclusively to the space of distal objects, and 223.2 (24.80%) of the output layer cells learn to respond exclusively to the proximal object after 25 epochs of training. The average number of output layer cells classified into each response category is approximately the same as the case of a model implemented with the same model parameters, but trained for 100 epochs (c.f. Table 3.3). When the model is trained with less than 25 training epochs,

Table 3.7: The learned responses of the output layer cells in Experiment 3.6

Experiment 3.6 Output Layer Cell Learned Responses								
	10% **Overlap** **No. Training Epochs**							
	1		**5**		**10**		**25**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	53.0 (5.89%)	9.57	138.4 (15.38%)	8.41	253.8 (28.20%)	3.49	415.2 (46.13%)	7.82
Proximal	6.6 (0.73%)	1.95	39.8 (4.42%)	6.06	104.6 (11.62%)	6.23	223.2 (24.80%)	5.12
Mixed Response	414.6 (46.07%)	5.13	446.4 (49.60%)	9.4	378.0 (42.0%)	6.04	243.2 (27.02%)	7.4
No Response	425.8 (47.31%)	3.77	275.4 (30.60%)	5.41	163.6 (18.18%)	1.82	18.4 (2.04%)	3.91
	40% **Overlap** **No. Training Epochs**							
	1		**5**		**10**		**25**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	78.4 (8.71%)	10.74	194.0 (21.56%)	3.61	261.2 (29.02%)	7.4	309.6 (34.40%)	7.92
Proximal	16.8 (1.87%)	4.6	78.8 (8.76%)	7.92	76.0 (8.44%)	7.48	63.0 (7.00%)	4.47
Mixed Response	670.8 (74.53%)	7.85	596.4 (66.27%)	9.13	562.4 (62.49%)	14.42	527.4 (58.60%)	11.19
No Response	134.0 (14.89%)	3.81	30.8 (3.42%)	2.05	0.4 (0.04%)	0.55	0	0

The classification criteria used are described in Experiment 3.1 (Section 3.4.1). Each of the results displayed is the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. As the number of training epochs is increased, so the performance of the model increases. With as little as 25 training epochs, the model produces learned behaviour similar to the case when the model is implemented with 100 training epochs (c.f. Table 3.3). In comparison to the model simulated with orthogonal inputs (Experiment 2.5, Chapter 2), the model with distributed inputs requires more training epochs to approach the behaviour of the mature model. The distributed nature of the inputs add complexity to the computational problem that the model must learn to solve, which requires further training in order for the model to learn the correct solution.

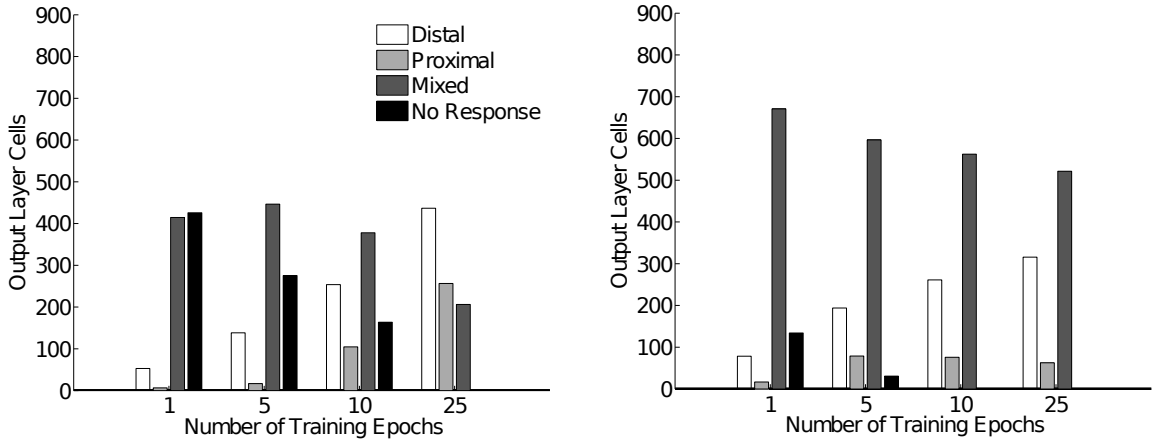

Figure 3.10: The learned responses of the output layer cells in Experiment 3.6. The bar charts display the average number of output layer cells per response type for simulations conducted with different amounts of training. In all cases, the averages are calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The left-hand bar chart displays results recorded from a model with a 10% overlap between the input layer cell subsets. The right-hand bar chart displays results recorded from a model with a 40% overlap between the input layer cell subsets. For overlaps of both 10% and 40% it is clear that the model is able to approach its mature learned behaviour after only a small amount of training. It is, however, the case that the ability of the output layer cells to discriminate between the distal and proximal object spaces continues to improve as the amount of training is increased. The data displayed in the bar charts are also reported in Table 3.7.

there is a general trend that as the number of training epochs is increased, so the number of output layer cells that learn to respond exclusively to either the space of distal objects or the proximal object increases. The number of output layer cells that learn to respond to both of the distal and proximal object spaces similarly decreases. A difference between the fully-trained model and the model with less training is that, in the latter case, a number of output layer cells do not exhibit a response until a number of training epochs have been experienced by the model. This result is also different to the results obtained in the corresponding Experiment 2.5 of Chapter 2, where the output layer cells in the model with orthogonal input representations all display some type of learned response, even after as little as 5 epochs of training.

Simulating the model with a 40% overlap between the input cell subsets produces slightly different

learned behaviour. The general trend is for there to be an increase in the number of output layer cells that learn to respond exclusively to the space of distal objects as the number of training epochs is increased. Similarly, the number of output layer cells that learn to respond to a mixture of the distal and proximal object spaces decreases as the number of training epochs decreases. After 25 epochs of training, 309.6 (34.40%) of the output layer cells respond exclusively to the space of distal objects. 527.4 (58.60%) of the output layer cells respond to a mixture of the distal and proximal object spaces. These results compare favourably to the situation where the model is trained with identical parameters, but with 100 training epochs (Table 3.3).

In comparison to the model simulated with orthogonal, non-overlapping input cell representations, the current model with distributed, overlapping input cell representations learns at a slower rate. That is, the distributed nature of the input cell subsets leads to the model requiring more training epochs before mature learned behaviour is produced. The model with distributed inputs can still produce mature learned behaviour after undergoing only 25 training epochs. This number of training epochs, however, is greater than the number of training epochs required in order for the model with orthogonal inputs to display mature behaviour. The distributed nature of the input representation has the effect that more training, compared to the model with orthogonal inputs, is required in order to produce output layer cells that have a high degree of specificity in their learned response. This result demonstrates that the model with distributed input representations has a more difficult computational task to solve than the model with orthogonal input representations.

Figure 3.11 displays the learned responses of two output layer cells recorded after 1, 10, and 25 epochs

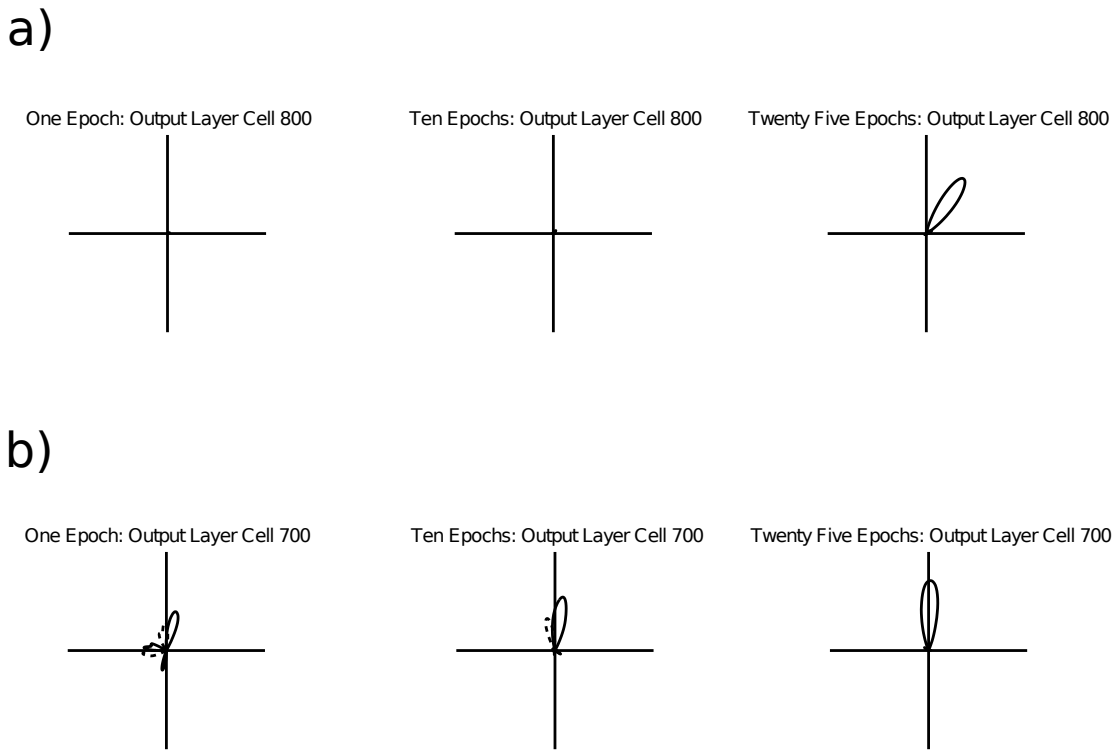

Figure 3.11: The learned responses of the output layer cells in Experiment 3.6. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. a) The responses of the same output layer cell recorded from the same simulation after the model has undergone different amounts of training with a 10% overlap between the input cell subsets. After the model has undergone 25 epochs of training, it exhibits performance that approaches the performance of the fully trained model. In other words, after 25 epochs of training the model produces output layer cells that have learned to respond exclusively to either the space of distal objects or the proximal object, but not both. Interestingly, the model requires a certain amount of training in order for the output layer cells to exhibit a response. b) The responses of the same output layer cell recorded at different stages in the training of the model with an overlap of 40% between the input cell subsets. Similar to the case with a 10% overlap, the model with a 40% overlap approaches the learned behaviour of the mature model after only 25 epochs of training.

of training. The top row shows results from a model with an overlap of 10% between the input cell subsets, and the bottom row shows results from a model with an overlap of 40% between the input cell subsets. It is clear from the plots that as the number of training epochs increases, so the learned behaviour of the model changes. With a 10% overlap between the input cell subsets, the model requires up to 25 training epochs in order that the output layer cells begin to show some sort of response. With a 40% overlap between the input cell subsets, the responses of the output layer cells change from representing a mixture of the distal and proximal object spaces, to exclusively representing one of the two

object spaces by 25 training epochs. Once 25 training epochs have been experienced, the responses of the output layer cells are qualitatively similar, regardless of the amount of overlap between the input layer cell subsets.

In summary, when the model with distributed, overlapping input cell subsets is exposed to a reduced amount of training, the model is still capable of producing learned behaviour, within 25 training epochs, that approaches the learned behaviour of the mature model. The model with distributed input representations requires more epochs of training to approach the mature output layer cell response than does the model with orthogonal input representations reported in Chapter 2. This result indicates that learning to form separate representations of distal and proximal object spaces is more difficult when those spaces are represented in a distributed, overlapping manner in the input layer. Nevertheless, the model with distributed input representations can still approach mature learned behaviour after a reduced amount of training of 25 epochs.

3.4.7 Experiment 3.7: Model Performance and the Learning Rate

The simulations that comprise this experiment were conducted to investigate the effect that different values of the learning rate k have upon the performance of the model. The value of the learning rate k was varied as follows: 0.1, 0.01, 0.001. All other network parameters were set to the value reported in Table 3.4.

Table 3.8 and Figure 3.12 display the learned responses of the output layer cells classified by learned response type. When the model is simulated with an overlap of 10% between the input cell subsets, and a learning rate k set to 0.1 or 0.01, then the majority of the output layer cells, 878.2 (97.58%)

Table 3.8: The learned responses of the output layer cells in Experiment 3.7

Experiment 3.7 Output Layer Cell Learned Responses						
	10% **Overlap** **Learning Rate**					
	0.1		**0.01**		**0.001**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	13.6 (15.1%)	1.52	92.0 (10.22%)	4.95	505.4 (56.16%)	3.6
Proximal	2.8 (0.22%)	1.48	104.4 (11.60%)	11.17	394.6 (43.84%)	3.6
Mixed Response	878.2 (97.58%)	2.49	703.6 (78.18%)	10.92	0	0
No Response	5.4 (0.60%)	2.61	0	0	0	0
	40% **Overlap** **Learning Rate**					
	0.1		**0.01**		**0.001**	
	Cells	**S.D.**	**Cells**	**S.D.**	**Cells**	**S.D.**
Distal	11.4 (1.27%)	3.58	128.0 (14.22%)	6.12	324.8 (36.09%)	7.4
Proximal	2.0 (0.22%)	0	6.4 (0.71%)	14.31	467.8 (51.98%)	4.1
Mixed Response	881.4 (97.93%)	4.51	765.6 (85.07%)	19.24	107.4 (11.93%)	6.6
No Response	5.2 (0.58%)	3.56	0	0	0	0

The response criteria are described in Experiment 3.1 (Section 3.4.1). The results shown are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S. D. = standard deviation. It is clear that, independent of the percentage overlap between the input cell subsets, the model requires a low learning rate in order to produce output layer cells that learn to respond exclusively to either the space of distal objects or the proximal object, but not both. In other words, a low learning rate promotes specificity and exclusivity in the output layer cell learned response

.

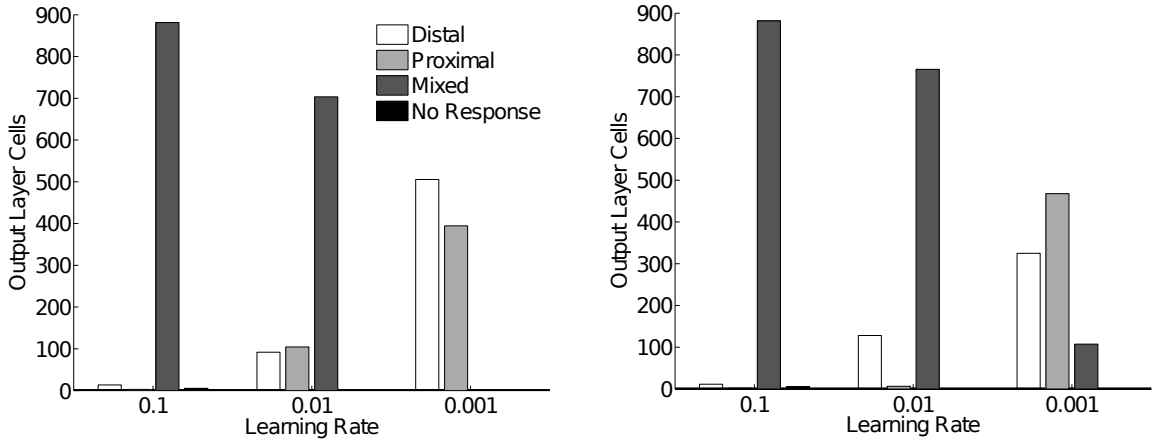

Figure 3.12: The learned responses of the output layer cells in Experiment 3.7. The average number of output layer cells per response type are displayed for simulations conducted with three different values of the learning rate k. The averages are calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The left-hand bar chart displays data recorded from a model with a 10% overlap between the input layer cell subsets. The right-hand bar chart displays data recorded from a model with a 40% overlap between the input layer cell subsets. It is only when the learning rate k is set to a low value that the majority of the output layer cells learn to respond exclusively to either the space of distal objects or the proximal object, but not both. This is true for models implemented with 10% and 40% overlaps between the input layer cell subsets. The data displayed are also reported in Table 3.8.

and 703.6 (78.18%) respectively, learn to respond to a mixture of the distal and proximal object spaces. When the value of the learning rate is decreased to 0.001, there is a noticeable reduction in the number of output layer cells that develop a mixed response to both object spaces. All output layer cells develop a learned response that is exclusive to either the space of distal objects or the proximal object, but not both. 505.4 (56.16%) of the output layer cells learn to respond exclusively to the space of distal objects. 394.6 (43.84%) of the output layer cells learn to respond exclusively to the proximal object.

When the model is simulated with an overlap of 40% between the input cell subsets, a relatively high learning rate value of 0.1 or 0.01 leads to the majority of the output layer cells, 881.4 (97.93%) and 765.6 (85.07%) respectively, learning to respond to a mixture of the distal and proximal object spaces.

Decreasing the value of the learning rate to 0.001 leads to 324.8 (36.09%) of the output layer cells learning to respond exclusively to the space of distal objects. An average of 467.8 (51.98%) of the output layer cells learn to respond exclusively to the proximal object space. A small number, 107.4 (11.93%), of the output layer cells learn to respond to a mixture of the distal and proximal object spaces. That a number of the output layer cells learn a mixed response is the primary difference between the model simulated with an overlap of 10% between the input cell subsets and the model simulated with an overlap of 40%. Otherwise, the general trend of the output layer cell representation changing from a distributed representation to a local, grandmother-cell-like representation as the value of the learning rate k decreases is the same for both percentages of overlap in the input layer.

Figure 3.13 displays the learned responses of the output layer cells. In Figure 3.13a, the three plots display the same output layer cell recorded from simulations conducted with values of the learning rate k of 0.1, 0.01, and 0.001. In each plot, the data was recorded from a simulation implemented with an overlap of 10% between the input cell subsets. All simulations were conducted with the same synaptic weight initializations. It is evident from the plots that as the value of the leaning rate k decreases, so the learned behaviour of the model changes. With a relatively high learning rate of 0.1 or 0.01, the model produces output layer cells that learn to respond to a mixture of the distal and proximal object spaces. With a learning rate of 0.001, the model produces output layer cells that learn to respond exclusively to one of the two object spaces.

The three plots in Figure 3.13b display another output layer cell recorded from simulations implemented with the same changes in the learning rate k as in Figure 3.13a. The data in all three plots

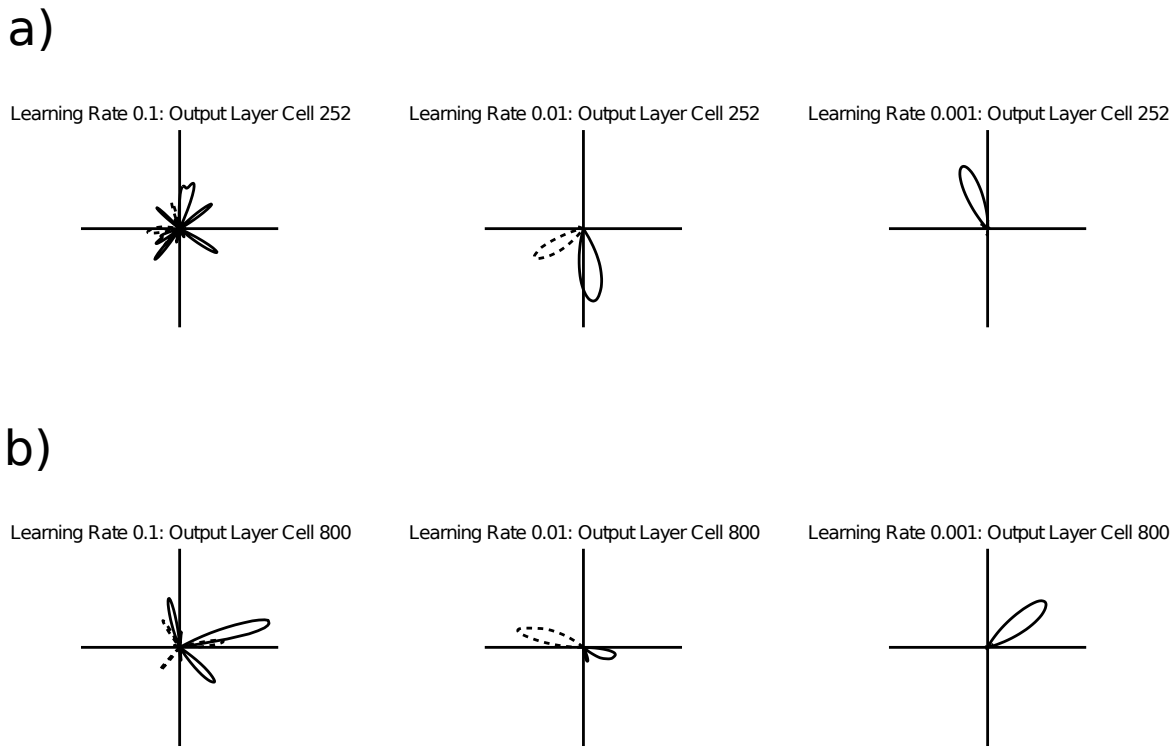

Figure 3.13: The learned responses of the output layer cells in Experiment 3.7. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. a) The responses of the same output layer cell recorded from simulations conducted with different values of the learning rate k, but with identical synaptic weight initializations. Each of three plots displays data recorded from a simulation implemented with an overlap of 10% between the input cell subsets. It is clear from the plots that as the value of the learning rate k decreases, so the learned behaviour of the output layer cells changes from representing a mixture of the distal and proximal object spaces, to exclusively representing one of the two object spaces. b) The learned responses of another output layer cell recorded from simulations conducted with different values of the learning rate k, but with identical synaptic weight initializations. In all three plots, the output layer cell was recorded from a simulation implemented with an overlap of 40% between the input cell subsets. Similar to a), decreasing the value of the learning rate k changes the learned behvaiour of the output layer cells such that the cells eventually learn to exclusively represent one of the two object spaces. This result is independent of the size of the overlap between the input cell subsets.

were recorded from simulations conducted with the same synaptic weight initializations and with a 40% overlap between input cell subsets in each case. The results are similar to Figure 3.13a: as the value of the learning rate k decreases, so the output layer cells change from learning to represent a mixture of the distal and proximal object spaces, to learning to exclusively represent one of the two object spaces. Thus, the effect of the learning rate upon the learned behaviour is independent of the percentage overlap between the input cell subsets.

In conclusion, a low value of the learning rate k is required in order for the model to produce output layer cells that learn to respond exclusively to one of the distal or proximal object spaces. When the learning rate k is relatively high, 0.1 or 0.01, the output layer cells lose the specificity of their learned responses. A relatively low value of the learning rate, 0.001, promotes the development of output layer cells that learn to respond exclusively to one of the distal or proximal object spaces, but not both. This behaviour is generally independent of the percentage overlap between the input cell subsets; although, with the larger overlap of 40%, a number of the output layer cells still learn to respond to a mixture of the object spaces. The higher magnitude of the overlap between the input cell subsets leads to a harder computational problem to be solved, which produces a lack of specificity in a small number of the output layer cells. As discussed in Section 2.5.6 of Chapter 2, the simulation results obtained when altering the value of the learning rate k may be evidence that the current model is susceptible to the stability-plasticity dilemma (Carpenter and Grossberg, 1987, 1988; Grossberg, 1987).

3.4.8 Experiment 3.8: Model Performance is Independent of Model Size

The competitive learning mechanism acts by adapting the threshold of the activation to firing-rate transfer function for the cells in the output layer. As discussed in Experiment 2.7 (Chapter 2, Section 2.5.7), the adaptive threshold transfer function has the effect that the relative percentage of output layer cells that are permitted to fire at any model update will remain approximately the same. The actual number of output layer cells firing will, however, change with the number of output layer cells in the model.

The simulations comprising this experiment were conducted to investigate the effect that different

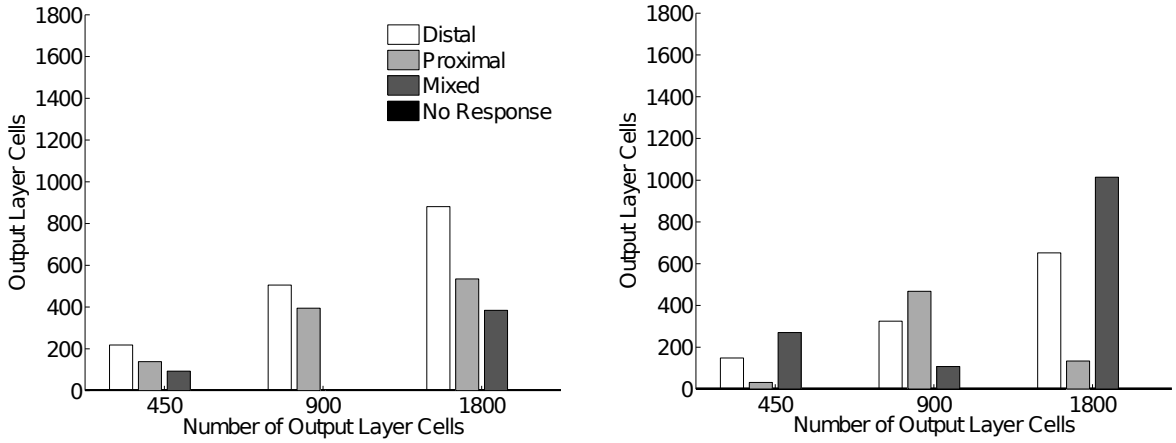

Figure 3.14: The learned responses of the output layer cells in Experiment 3.8. The bar charts display the average number of output layer cells per response type for models with implemented with different numbers of output layer cells. The averages are calculated across five simulations conducted with identical model parameters, but with different random synaptic weight initializations. The left-hand bar chart displays data recorded from a model implemented with a 10% overlap between the input layer cell subsets. The right-hand bar chart displays data recorded from a model implemented with a 40% overlap between the input layer cell subsets. It is evident that in simulation the model can learn to produce output layer cells that respond exclusively to either the space of distal objects or the proximal object across a range of different sizes of the output layer. This result holds for models implemented with a 10% and a 20% overlap between the input layer cell subsets. The data displayed in the bar charts are also reported in Table 3.9.

numbers of cells in the output layer has upon the learned behaviour of the model. Ideal model performance will be achieved if the learned behaviour of the output layer cells is independent of the number of output layer cells. The model was simulated with three different numbers of output layer cells: 450, 900, and 1800. The network parameters are given in Table 3.4.

Table 3.9 and Figure 3.14 display the learned responses of the output layer cells classified by response type. When the model is simulated with an overlap of 10% between the input cell subsets, the majority of the output layer cells, for each of the three models with different numbers of output layer cells, learn to respond exclusively to one of the distal or proximal object spaces, but not both.

Simulating the model with an overlap of 40% between the input cell subsets produces approximately

Table 3.9: The learned responses of the output layer cells in Experiment 3.8

Experiment 3.8 Output Layer Cell Learned Responses						
	10% **Overlap** **No. Output Layer Cells**					
	450 Cells	**S.D.**	**900 Cells**	**S.D.**	**1800 Cells**	**S.D.**
Distal	218.2 (48.49%)	3.11	505.4 (56.16%)	3.6	881.0 (48.94%)	6.67
Proximal	138.8 (30.84%)	3.03	394.6 (43.84%)	3.6	534.8 (29.71%)	6.14
Mixed Response	93.0 (20.67%)	3.39	0	0	384.2 (21.34%)	10.23
No Response	0	0	0	0	0	0
	40% **Overlap** **No. Output Layer Cells**					
	450 Cells	**S.D.**	**900 Cells**	**S.D.**	**1800 Cells**	**S.D.**
Distal	148.4 (32.98%)	1.82	324.8 (36.09%)	7.4	651.8 (36.21%)	8.11
Proximal	31.2 (6.93%)	3.03	467.8 (51.98%)	4.1	133.8 (7.43%)	15.21
Mixed Response	270.4 (60.09%)	1.82	107.4 (11.93%)	6.6	1014.4 (56.36%)	20.88
No Response	0	0	0	0	0	0

The response criteria are given in Experiment 3.1 (Section 3.4.1). The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations. S.D. = standard deviation. It is clear that, for both a 10% and a 40% overlap between the input cell subsets, and with different numbers of cells in the output layer, the model can produce some output layer cells that learn to respond exclusively to either the space of distal objects or the proximal object, but not both.

the same learned behaviour as in the case of a 10% overlap between the input cell subsets. That is, the majority output layer cell learned response is to exclusively represent one of the distal or proximal object spaces. An exception to this is the case where the model contains 450 output layer cells. With this set of model parameters, the majority, 270.4 (60.09%), of the output layer cells learn to respond to both of the distal and proximal object spaces. This is not the case for a model simulated with the same number of output layer cells, but with an overlap of 10% between the input cell subsets. Thus, there is an interaction between the number of output layer cells and the size of the overlap between the input cell subsets. A high percentage of overlap between the input cell subsets presents a harder computational problem to solve.

The three plots in Figure 3.15a display the same output layer cell recorded from three simulations conducted with 450, 900, and 1800 output layer cells in the model. In all cases, the model was simulated with a constant overlap of 10% between the input cell subsets. It is clear from the plots that the ability of the output layer cells to learn to respond exclusively to one of the distal or proximal object spaces is independent of the number of output layer cells in the model. The only qualitative difference between the three simulations is that increasing the number of output layer cells decreases the magnitude of the individual output layer cell responses. Similar behaviour was observed in the corresponding Experiment 2.7 of Chapter 2 (Section 2.5.7). The three plots in Figure 3.15b display similar information to 3.15a. The difference is that the data in 3.15b were recorded from simulations with a constant overlap of 40% between the input cell subsets. Similar to 3.15a, the output layer cells are able to learn to respond exclusively to one of the distal or proximal object spaces independent of the number of output layer cells in the model.

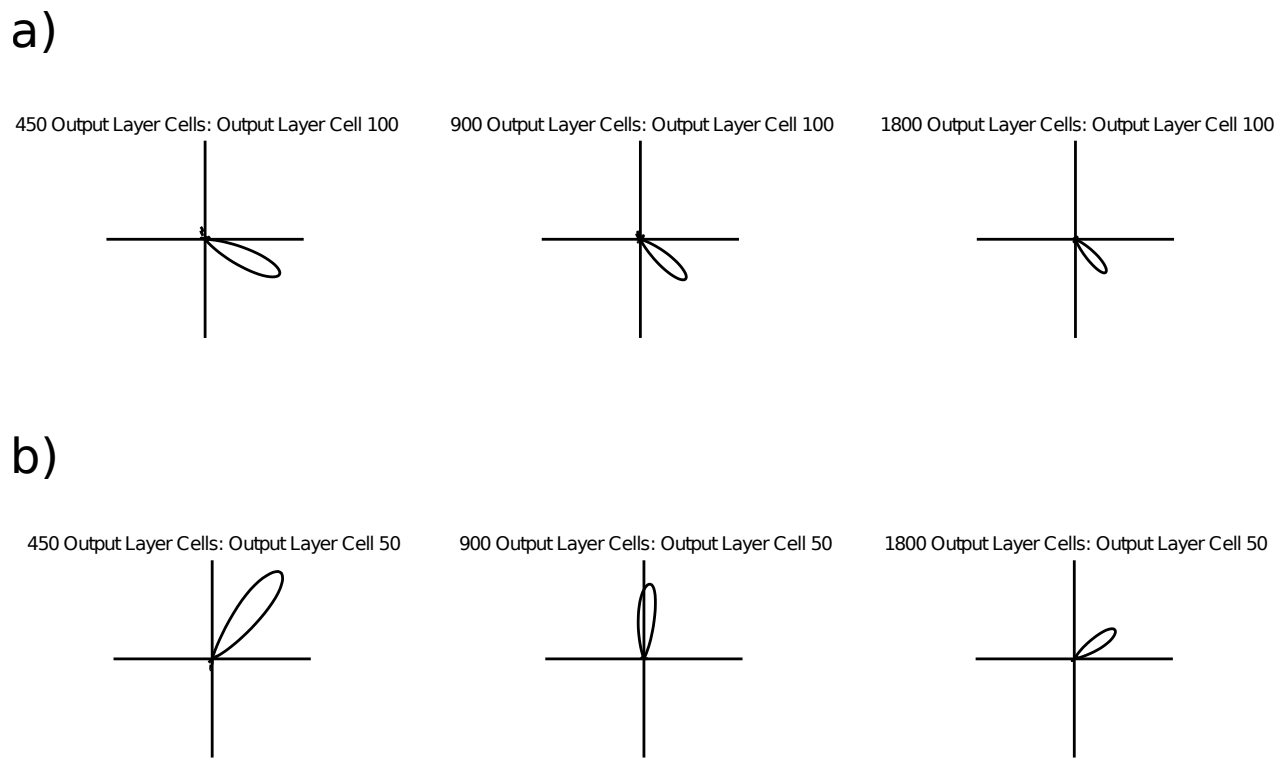

Figure 3.15: The learned responses of the output layer cells in Experiment 3.8. The response to the space of distal objects is represented by the unbroken line. The response to the proximal object is represented by the dashed line. a) The responses of the same output layer cell recorded from the three simulations conducted with different numbers of output layer cells. The data displayed in each of the three plots are taken from simulations that were conducted with a constant overlap of 10% between the input cell subsets. The model produces output layer cells that have learned to respond exclusively to one of the distal or proximal object spaces independently of the number of output layer cells in the model. b) The responses of another output layer cell recorded from three simulations conducted with different numbers of output layer cells. The data was recorded from three simulations with a constant overlap of 40% between the input cell subsets. The model can produce output layer cells that respond exclusively to one of the distal or proximal object spaces, and this behaviour is independent of the number of output layer cells in the model. Thus, this learned behaviour is also largely independent of the size of the overlap between the input cell subsets.

In summary, successful model performance is largely independent of the size of the output layer of the model. The model is able to produce output layer cells that respond exclusively to one of the distal or proximal object spaces, but not both, given different numbers of output layer cells and different percentage overlaps between the input cell subsets. This is an important result, because the data recorded during this experiment demonstrate that neural network model explored in this chapter can

be scaled up or down in size with no major effect upon the model behaviour. Thus, there is a general comutational principle, independent of the size of the network, which is that it is possible to learn to form separate representations of distal and proximal visual cues on the basis of a statistical decoupling between the visual cue spaces when the input are presented to the network in a distributed, overlapping manner. A real neural network in the brain of arbitrary size can therefore solve this particular computational problem.

3.5 Discussion

This chapter provides the implementation details and simulation results for single-layer feedforward neural network that can learn to form separate representations of distal and proximal visual cues that are presented simultaneously to the network as input. In this model, the network inputs are represented in a distributed, overlapping manner. This is the primary difference between the model reported in this chapter and the model reported in the previous chapter, which has orthogonal input representations. The main aim of the research reported in this chapter is to demonstrate that there are general computational principles that can operate with distributed input representations (this Chapter), or orthogonal input representations (previous Chapter), by which separate representations can be formed of distal and proximal visual cues. To this end, the experiments reported in the previous chapter were carefully replicated in this chapter with a model with distributed input representations. The main points of this chapter are:

- The model comprises a single layer of input cells connected to a single layer of output cells by a single layer of feedforward synapses.
- The input layer comprises a common pool of input layer cells. Subsets of the input layer cells

are drawn from this common pool of cells. Each subset represents one of the distal or proximal object spaces. It is therefore possible that a randomly chosen input layer cell will belong to more than one input layer subset. This possibility produces an overlap of input cells that are common to all the different input cell subsets. The magnitude of this overlap can be controlled by altering the total number of cells in the common pool of input layer cells.

- The model must learn to solve a harder computational task than the model reported in the previous chapter. Although the end aim of learning to form separate representations of the distal and proximal object spaces is common to both models, the current model must learn this task when an individual input layer cell can represent different object spaces at different model updates. This is not the case for the model with orthogonal input representations.

- In simulation, the model can learn to form separate representations of the distal and proximal object spaces when presented with distributed input representations. Moreover, the model can achieve this across a variety of values of the overlap between the input layer cell subsets. The agent must visit a minimum number of training locations to solve this computational task, and there is a requirement for relatively strong inter-cell inhibition amongst the output layer cells.

- Increasing the number of proximal objects in the training environment results in a change in the nature of the output layer cell representation. With a small number of proximal objects in the training environment, the output layer cells develop a local, grandmother-cell-like representation where individual output layer cells respond exclusively to a narrow region of either the space of distal objects or the proximal object space, but not both. With a larger number of proximal objects, the output layer cells develop a distributed representation where individual output layer cells respond to multiple regions of the distal and proximal object spaces. A similar effect

is produced by increasing the value of the sparseness a. It is only in the case where the model produces a local output layer representation that the model is considered to have learned the correct type of output representation.

- The value of the learning rate k has a similar effect upon the nature of the output layer cell representation. Relatively high values of k produce a distributed representation, whereas a low value of k, such as 0.001, will produce a local, grandmother-cell-like representation.

- The learned behaviour of the model is relatively robust to a reduction in the amount of training the model is exposed to.

- Altering the size of the output cell layer does not qualitatively effect the nature of the learned output cell layer responses.

Much like the model with orthogonal input representations, the current model is intended to be representative of the type of computation required to discriminate between distal and proximal visual cues. Consequently, the output layer cells lack some of the common properties of real head direction cells. These properties were discussed in Section 2.6 of Chapter 2, and so will not be repeated here.

A possible reason why the current model, incorporating distributed input representations, is able to learn to discriminate between the distal and proximal object spaces is that the distributed input representations actually facilitate learning to discriminate between the object spaces. It was argued in Section 3.3 that distributed representations of the object spaces in the input cell layer should produce a harder computational problem for the model to solve, compared to the model with orthogonal representations in the input cell layer that is reported in Chapter 2. It can be argued, however, that the

opposite is true, and the distributed representation in fact helps the model to learn to discriminate between the distal and proximal object spaces.

It can be argued that an input layer cell that belongs to more than one input layer cell subset, and thus represents more than one object, will project weaker efferent synapses to the output layer cells than an input layer cell that only belongs to one input layer cell subset. The reason for this is that the input layer cell that belongs to multiple input layer cell subsets will not have its efferent synapses to the output cell layer consistently updated, whereas the input layer cell belonging to a single input layer cell subset *will* have its efferent synapses consistently updated. The consistent weight updating, together with the synaptic weight normalization, ensures that the input layer cell that belongs to only a single input layer cell subset will have stronger efferent synapses than the input layer cell that belongs to multiple input layer cell subsets. Consequently, the input layer cells with stronger efferent synapses will play an increasingly more influential role in determining which object space an output layer cell learns to respond to. As the input layer cells with the strongest efferent synapses will be those input layer cells that only belong to a single input layer cell subset, the distributed input layer cell representation in fact facilitates, rather than hinders, the output layer cells in learning to discriminate between the distal and proximal object spaces.

This hypothesis is not correct though. Firstly, if the distributed input representations do in fact facilitate learning to discriminate between the distal and proximal object spaces, one might expect to see an improvement in model performance as the percentage overlap between the input layer cell subsets increases. However, as the results of Experiment 3.2 (Section 3.4.2) show, changing the per-

centage overlap between the input layer cell subsets has very little effect upon the qualitative and quantitative performance of the model.

Second, if the hypothesis were correct then input layer cells that belong to a single input layer cell subset should have stronger efferent synapses than input layer cells that belong to multiple input layer cell subsets. Although the results are not shown here, an inspection of the trained efferent weight vectors from the input layer cells to the output layer cells for both input layer cells belonging to a single input layer cell subset, and input layer cells belonging to multiple input layer cell subsets, revealed no subset membership dependent difference in the magnitude of the synaptic weights, i.e. whether an input layer cell belongs to one or more subsets.

Thus, the distributed input layer cell representations do not facilitate learning to discriminate between the distal and proximal object spaces. In fact, a comparison of the results obtained with orthogonal input representations (presented in Chapter 2) and the results presented in the current chapter reveals that a model with distributed input representations in general performs a little bit worse than a model with orthogonal input representations. This supports the argument given in Section 3.3 that a model with distributed input layer cell representations must solve a harder computational problem than a model with orthogonal input layer cell representations.

In conclusion, a model with distributed input representations can, under the correct conditions, learn to form separate representations of the distal and proximal visual cues presented simultaneously to the model as input. A subpopulation of the output layer cells will develop responses that are exclusive

to narrow regions of the space of distal objects. The firing of this subpopulation of output layer cells will reflect the current head direction of the agent, and the output layer cells within this subpopulation will behave like head direction cells.

Careful replication of the experiments reported in Chapter 2, that were conducted with a model containing orthogonal input representations, has shown that the principle of using the statistical decoupling between distal and proximal visual cues as an aid to competitive learning is a general principle that can operate with distributed input representations. Moreover, to return to a point discussed in Section 2.6 of the previous Chapter, the results reported in this current Chapter clearly argue that it is the statistical decoupling between the distal and proximal object spaces, and not the nature of the input cell layer representation, i.e. orthogonal or distributed, that determines whether or not the output layer cells can learn to discriminate between the distal and proximal object spaces.

Chapter 4

The Isomapping of Preferred Firing Directions Across Environments

This chapter reports the implementation details, equations, and simulation results of a neural network model that can learn to maintain the angular distance between the preferred firing directions of its output layer cells across two environments. That is, the model learns to produce an isomapping of the output layer cell preferred firing directions. The isomapping of preferred firing directions is a well-established property of real head direction cells (Taube et al., 1990b; Taube and Burton, 1995; Zugaro et al., 2004; Yoganarasimha et al., 2006), and was discussed in Section 1.4 of Chapter 1. Through a set of carefully controlled experiments, the effect of the model parameters upon the learned model behaviour is explored. Future research is also considered.

4.1 Model Architecture

The isomapping model contains a single layer of input cells, and a single layer of output cells. The input layer cells are connected to the output layer cells by a single layer of feedforward synaptc con-

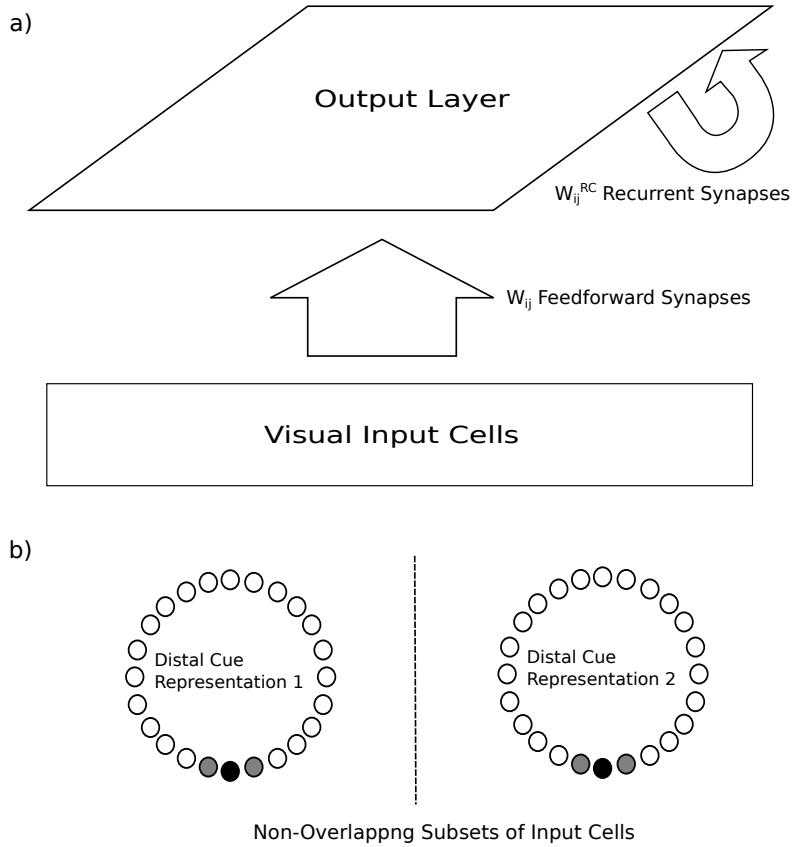

Figure 4.1: a) Neural network architecture of the simulated agent for the model of the development of the isomapping of head direction cell preferred firing directions across different environment. The model comprises a single layer of input cells that are connected, through feedforward synaptic connections w_{ij}, to a single layer of output cells. Within the output layer cells, there are excitatory recurrent collateral synapses, w_{ij}^{RC}, that connect every output layer cell to every other output layer cell. The two sets of synaptic connections self-organize through competitive learning as the agent explores different training environments. Under the correct model parameters, an isomapping of the output layer cell preferred firing directions occurs across different training environments. b) The layer of input cells consists of two orthogonal subsets of input cells. Each subset represents the input from the distal visual cue space in one of the environments. None of the input layer cells are sensitive to proximal visual cues.

nections, w_{ij}. Excitatory recurrent collateral synapses, w_{ij}^{RC}, in the output layer ensure that every output layer cell sends efferent synapses to, and receives afferent synapses from, every other output cell in the layer. The architecture of the model is displayed in Figure 4.1a.

In the input layer, the cells are divided into two orthogonal subsets, which represent visual input from

two visual environments. One subset of input cells represents all views, from $0^{o} - 360^{o}$, of the distal visual cue space in the first environment. The second subset of input cells represents all views, from $0^{o} - 360^{o}$, of the distal visual cue space in the second environment. The input layer does not contain any input cells that are sensitive to proximal visual cues. The input layer is displayed in Figure 4.1b. The grey and black shading in the subsets of input layer cells represents a Gaussian packet of activity that corresponds to the current orientation of the agent with respect to the distal visual cue space of the current environment. The black shading represents a high firing rate, and the grey shading represents a lower firing rate.

4.2 Training Environment

The training protocol incorporates two visual training environments. An example training environment is depicted in Figure 4.2. In either of the two training environments, when the agent has a particular orientation with respect to the space of distal visual cues, the agent will receive the same pattern of visual input, no matter its current location. Thus, each training environment effectively contains only a single training location. Each environment also contains a continuous circular space of distal objects placed at an infinite distance from the training location. Neither of the two training environments contains any proximal visual cues. Within a training environment, the simulated agent will rotate through 360^{o} ten times at the training location to ensure that the synaptic weights receive enough training.

The computational problem is to learn to preserve the angular distance between the preferred firing directions of any two output layer cells across the two training environments.

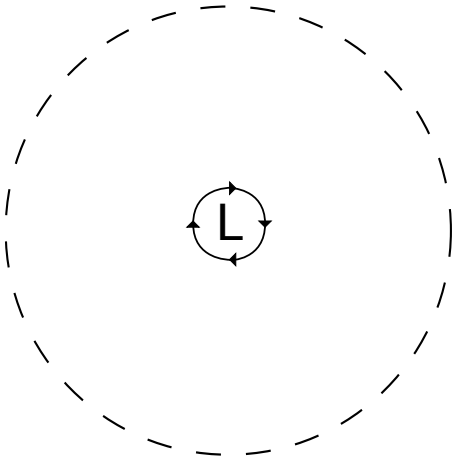

Figure 4.2: An example of the visual training environment for the isomapping model. There is a single training location in the environment, represented by **L**. At this training location, the agent will rotate through 360° in single degree increments. The dashed line represents a continuous space of distal visual cues located at an infinite distance from the training location (although the distance is depicted as finite in the figure for clarity). As the agent rotates at the training location, the current orientation of the agent, with respect to the space of distal visual cues, is calculated. Because there is effectively only a single training location in the environment, a single training epoch consists of the agent rotating ten times through 360° at the training location. This ensures that the synaptic weights receive enough training to produce meaningful output.

4.3 Model Operation

A competitive learning paradigm ensures that an individual output layer cell will learn to respond to a particular view of the distal object space in each environment. The excitatory recurrent collateral synapses ensure that the angular distance between the preferred firing directions of any two output layer cells is maintained across the two environments as follows.

Output layer cells that develop preferred firing directions that are close to one another, in terms of angular distance, in the first environment (through competitive learning in the feedforward synaptic connections w_{ij}), will also develop strong excitatory recurrent collateral synaptic connections between one another (through associative Hebbian learning in the recurrent collateral synapses w_{ij}^{RC}). Thus, the recurrent weights between the output layer cells learn to reflect the learned topological responses

of the output layer cells to the first environment.

In the second training environment, the firing of the input layer cells is conveyed through initially untrained feedforward connections, which will initially lead to a random assortment of the output layer cells firing. The strong excitatory recurrent collateral synapses developed during learning in the first environment will then help to force the firing of a cluster of output layer cells that have preferred firing directions that are close to one another in the first environment. Hebbian associative learning in the feedforward synaptic connections w_{ij} ensures that the currently firing presynaptic input layer cells are associated with the currently firing postsynaptic output layer cells. The currently firing input layer cells will represent orientations of the agent that are close to one another in the second environment, and the currently firing output layer cells will come to represent those orientations. This is how an isomapping of output layer cell preferred firing directions is learned, across two environments, by the model.

4.4 Model Equations and Implementation

In the input cell layer, there are 360 input cells in each of the two input cell subsets representing distal visual input from the two visual training environments. The two subsets are orthogonal and therefore do not share any of the input cells.

As the simulated agent rotates through 360^{o} at a training location, the firing rates r_i of the individual input layer cells representing the distal object space in the current environment are set to a Gaussian response profile according to the current orientation θ of the agent as follows

$$r_i = \lambda e^{-s_i^2/2\sigma^2} \tag{4.1}$$

where σ is the standard deviation of the Gaussian response, and λ is a scaling factor. s_i is defined as

$$s_i = \mathrm{MIN}(|x_i - x|, 360 - |x_i - x|) \tag{4.2}$$

where x is the current orientation of the agent, and x_i is the preferred orientation of the agent for input cell i. This equation ensures that the input layer cell preferred orientations are continuous through 360^{o}, i.e. there is wrap-around in the input layer cell preferred orientations.

Once the input layer cell firing rates, r_i, have been set, these firing rates are propagated along the w_{ij} feedforward synapses to the output layer cells. The activation level h_i of each postsynaptic output layer cell i is calculated according to

$$h_i = \sum_{j \in input} w_{ij} r_j + \phi^{\mathrm{RC}} \sum_{j \in output} w_{ij}^{\mathrm{RC}} r_j \tag{4.3}$$

where the first term on the right-hand side of the equation arises from the feedforward synaptic connections from the input layer. The term involves a sum over all the input cells, where r_j is the firing rate of presynaptic input cell j, and w_{ij} is the strength of the feedforward synaptic weight from presynaptic input cell j to postsynaptic output layer cell i. The second term on the right-hand side of the equation arises from the excitatory recurrent collateral connections between the output layer cells. It involves a sum over all the output layer cells, where r_j is the firing rate of output layer cell j, w_{ij}^{RC} is the recurrent collateral synaptic weight from output layer cell j to output layer cell i, and ϕ^{RC} is a

scaling parameter that controls the overall strength of the recurrent collateral synaptic input.

The firing rate r_i of each postsynaptic output layer cell i is then calculated as a threshold linear function of the activation level h_i of that cell

$$r_i = f(h_i) \tag{4.4}$$

where the threshold of the function is iteratively adjusted until either the desired sparseness of firing a in the output layer is reached, or 300 iterations are completed. The sparseness of firing a of the output layer cells is given by

$$a = \frac{(\sum_{i=1}^{T} r_i/T)^2}{\sum_{i=1}^{T} r_i^2/T} \tag{4.5}$$

where r_i is the firing rate of output layer cell i, and T is the total number of output layer cells (Rolls and Treves, 1990, 1998). Adjusting the threshold of the transfer function in Equation (4.4) has the effect of introducing competitive learning into the output layer of the model. Only the output layer cells with a firing rate above the threshold are permitted to fire, and thus update both their w_{ij} synaptic weights with the input layer cells and the w_{ij}^{RC} synaptic weights with the other output layer cells.

The feedforward w_{ij} synapses, and the recurrent w_{ij}^{RC} synapses, are then updated according to the following equation. This equation also incorporates homosynaptic long-term depression (LTD) (Rolls and Treves, 1998; Dayan and Abbott, 2001; Stent, 1973)

$$\delta w_{ij} = (1 - w_{ij})\Big[k(r_i - \langle r_i \rangle)r_j\Big] \tag{4.6}$$

where w_{ij} is the synaptic weight between the presynaptic cell j and the postsynaptic cell i, r_i is the firing rate of the postsynaptic cell i, $\langle r_i \rangle$ is the average firing rate of the postsynaptic cell i, r_j is the firing rate of the presynaptic cell j, and k is the learning rate.

Subtracting the average firing rate of postsynaptic cell i from the current firing rate of postsynaptic cell i in Equation (4.6) produces homosynaptic LTD. The possibility of LTD arises because Equation (4.6) dictates that the synaptic weight w_{ij} between a particular postsynaptic cell i and a particular presynaptic cell j is only strengthened if there is simultaneous pre- and postsynaptic cell firing, *and* the current postsynaptic cell firing rate is above the average firing rate for the postsynaptic cell in question. In the situation where there is simultaneous pre- and postsynaptic cell firing but the current postsynaptic cell firing rate does not exceed the average firing rate for that postsynaptic cell, then the synaptic weight w_{ij} will decrease and homosynaptic LTD occurs. The LTD will increase the specificity of the learned response of the postsynaptic cell i because there will no longer be potentiation of the synapse w_{ij} between a strongly firing presynaptic cell j and a weakly firing postsynaptic cell i. In other words, only those presynaptic cells that produce a strong response in the postsynaptic cell i will be permitted to strengthen their synapses with that postsynaptic cell. The effect of the $(1 - w_{ij})$ term in Equation (4.6) is to bound the synaptic weight w_{ij} such that the value of the synaptic weight can never exceed 1. Because the learning rule above now permits LTD, there is the possibility that the value of any synaptic weight w_{ij} could become negative. To prevent negative synaptic weights from developing, all synaptic weights w_{ij} are set to have a minimum value of 0.

The training commences by rotating the agent through 360^{o} in the first training environment. Ten rotations of the agent through 360^{o} constitute a single training epoch. The firing rates and synaptic weights of the isomapping model are updated according to the equations given above. To allow the output layer cell recurrent collateral synapses to be effective, the agent is simulated as remaining at each particular orientation θ during a 360^{o} rotation for a total of five updates of the cell firing rates before all the synaptic weights are updated. This protocol permits the spread of the output layer cell activations to all other output layer cells before modifying the synaptic weights by learning. After the requisite number of training epochs have been performed, usually 50, then the training in the first environment is complete.

The model is then similarly trained rotating through 360^{o} in the second visual training environment. Again, ten rotations of the agent through 360^{o} constitute a single training epoch. After the model has been trained for the requisite number of training epochs, usually 50, in the second training environment then the training phase is complete.

After the training phase, the model enters the testing phase. The model is tested by recording the responses of the output layer cells as the agent performs a single 360^{o} rotation in each training environment in turn. After the training and testing phases have been finished, the simulation is complete.

Table 4.1: Network parameters for the simulations reported in Chapter 4

Network Parameters	
No. Environments	2
No. Input Cells Responsive to each Environment	360
Total No. Input Cells	720
No. Output Layer Cells	900
No. w_{ij} Feedforward Synapses onto each Output Layer Cell	720
No. w_{ij}^{RC} Recurrent Synapses onto each Output Layer Cell	900
a (Sparseness of Output Layer Cell Firing)	0.2
Training Parameters	
No. Training Epochs in each Environment	50
Total No. Training Epochs	100
k (Learning Rate)	0.001
λ (Scaling of Input Cell Gaussian Response Profiles)	10.0
σ (S.D. of Input Cell Gaussian Response Profiles)	5.0^{o}
ϕ^{RC} (Scaling of Excitatory Recurrent Collateral Synapses)	20.0

All values are constant across all experiments reported in this chapter except where noted in the text

4.5 Results

This section presents results concerning the effect that controlled alterations of the model parameters have upon the learned behaviour of the model.

4.5.1 Experiment 4.1: Demonstration of Core Model Performance

The simulations that comprise this experiment were conducted to demonstrate the core operational principles of a neural network that self-organizes to produce an isomapping of output layer cell preferred firing directions across different environments.

The network was simulated with the parameters given in Table 4.1.[1] The firing rates of the 900 output layer cells were recorded during the testing of the model. These firing rates are displayed in Figure 4.3. The plots display the results from two simulations conducted with identical model parameters except for the value of the recurrent collateral scaling factor ϕ^{RC}.

The results displayed in Figure 4.3a are taken from a simulation conducted with a large ϕ^{RC} value of 20.0. The left and middle plots of Figure 4.3a show the firing rates of the output layer cells ordered according to either their preferred firing directions in the first training environment (left-hand plot), or their preferred firing directions in the second environment (middle plot). If the angular distance between output layer cell preferred firing directions is maintained across the two environments, each of the plots should show a localized contiguous region of activity moving continuously through both orderings of the output layer cells as the agent rotates through $0^{\mathrm{o}} - 360^{\mathrm{o}}$. It is evident from Figure 4.3a that there is an isomapping of output layer cell preferred firing directions across the two environments.

The right plot of Figure 4.3a is a scatterplot that shows the preferred firing directions of the output layer cells in the first environment plotted against the preferred firing directions of the output layer cells in the second environment. If the angular distance between output layer cell preferred firing directions is maintained across the two environments then the data points in the scatterplot should display a linear trend. The scatterplot also confirms that an isomapping of the output layer cell preferred firing directions has been learned by the model.

[1] The sparseness value a reported in Table 4.1 is 0.2. A lower sparseness of 0.1 produces an interaction between the low proportion of the output layer cells firing at any model update, and the LTD now present in the learning rule. Consequently, a number of the output layer cells do not show any learned response. To allow for a simpler analysis of the model behaviour, all results reported in this chapter are from simulations conducted with a sparseness value of 0.2.

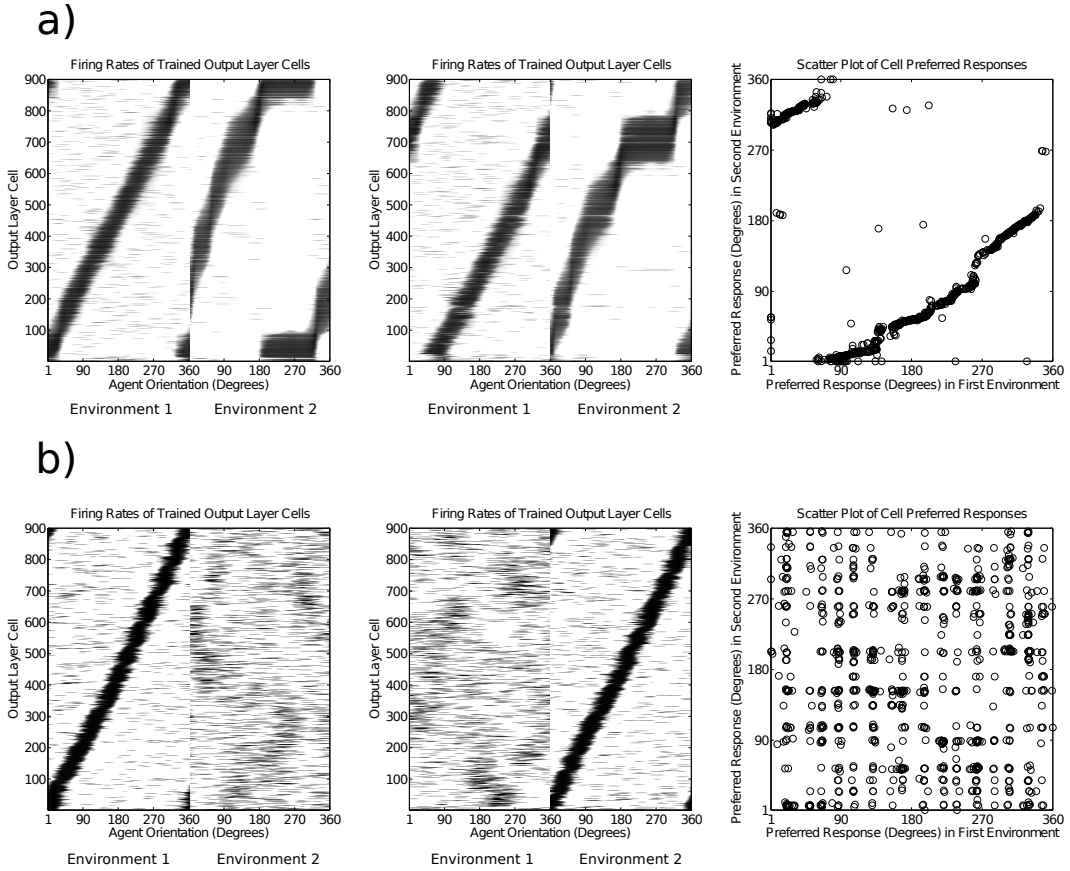

Figure 4.3: The learned responses of the output layer cells in the isomapping model. a) With a strong recurrent collateral scaling parameter ϕ^{RC} value of 20.0, an isomapping of the output layer cell preferred firing directions occurs across the two environments. The left and middle plots show the firing rates of the output layer cells as a function of the orientation of the agent within the first and second environments. (High firing is represented by black.) In the left plot, the output layer cells are ordered according to their preferred firing directions within the first environment. In the middle plot, the output layer cells are ordered according to their preferred firing directions within the second environment. The left and middle plots demonstrate that the angular distances between the preferred firing directions of the output layer cells are the same across both environments. Thus, the plots show that the network has learned an isomapping of preferred firing directions across the two environments. The right plot is a scatterplot that shows the preferred firing directions of the output layer cells in the first environment plotted against the preferred firing directions of the output layer cells in the second environment. It can be seen that there is a continuous mapping of the preferred firing directions across the two environments, showing that an isomapping has occurred. b) When the recurrent collateral scaling parameter ϕ^{RC} is reduced to a smaller value of 0.5, there is a remapping, rather than isomapping, of the output layer cell preferred firing directions in response to the two environments. All three plots have the same conventions as in (a). It is clear from the plots in (b) that the output layer cells learn a coherent mapping within each environment, but there is no isomapping of preferred firing directions across the two environments.

In Figure 4.3b, the plots display the firing rates of the output layer cells taken from a simulation conducted with a low ϕ^{RC} value of 0.5. The plots in 4.3b have the same conventions as the plots in Figure 4.3a. It is clear from the three plots that the model has learned a complex remapping of the output layer cells preferred firing directions, rather than an isomapping of the preferred firing directions. In the left and middle plots of Figure 4.3b, the output layer cells have clearly learned a coherent mapping of both of the two environments the agent was exposed to, but there is no isomapping of the output layer cell preferred firing directions across the two environments. That is, when the output layer cells are ordered according to their preferred firing directions in one of the two environments, the plot shows localized activity packets for that environment, but does not show localized activity packets for the other environment. Instead, a complex remapping of output layer cell preferred firing directions has occurred across the two environments.

The right plot of Figure 4.3b provides further evidence of a complex remapping of preferred firing directions. The preferred firing directions of the output layer cells in both environments, when plotted against one another, display in a random manner: there is no linear trend to the data points, unlike the scatterplot of Figure 4.3a. Thus, a relatively small value of the recurrent collateral scaling parameter ϕ^{RC} is not strong enough to produce an isomapping of output layer cell preferred firing directions.

From these results it is expected that a model with no recurrent collateral synapses will not be able to learn an isomapping of output layer cell preferred firing directions. The model was simulated with the recurrent collateral scaling parameter, ϕ^{RC}, set to a value of 0. The data are not presented here, but the results are qualitatively very similar to the data presented in Figure 4.3b. Thus, recurrent collat-

eral synapses are required if the model is to learn an isomapping of output layer cell preferred firing directions.

When the model is trained in the first environment, the output layer cells that develop preferred firing directions close to one another, in terms of angular distance, will also develop strong w_{ij}^{RC} synapses with one another. Upon the commencement of training in the second environment, the firing of the input layer cells signalling the current orientation of the agent, with respect to the space of distal objects, will produce firing in some of the output layer cells. If the scaling parameter of the recurrent collateral synapses, ϕ^{RC}, is set to a reasonably high value like 20.0, then the recurrent collateral synaptic input will result in a small cluster of output layer cells firing, which will have preferred firing directions in the first environment that are close to one another. Through associative Hebbian learning, the currently firing cluster of output layer cells will strengthen their w_{ij} synapses with the currently firing input layer cells. This is how the isomapping of output layer cell preferred firing directions is produced. If the scaling parameter, ϕ^{RC}, is set to a reasonably low value like 0.5, the recurrent collateral synaptic input will not be strong enough to influence the firing of the output layer cells and an isomapping of preferred firing directions will not occur.

The results of the simulations conducted in this experiment confirm two hypotheses. Firstly, the isomapping of the preferred firing directions of output layer cells across different environments can be learned on the basis of visual input alone. Second, the isomapping is dependent upon the recurrent collateral synapses, and it is the strength of the recurrent collateral synaptic input to the head direction cells that determines whether a head direction cell-like isomapping, or a place cell-like complex

remapping, occurs across different environments. It is interesting to note that a change in the value of one single model parameter, ϕ^{RC}, is sufficient to change the nature of the mapping that occurs between the preferred firing directions of the output layer cells across different environments. This result suggests that strong recurrent collateral inputs to head direction cells may be responsible for the development of the isomapping of cell preferred firing directions across different environments. In contrast, the fact that the preferred firing locations of place cells remap across different environments suggests that the recurrent connections between place cells may be relatively weaker (Muller and Kubie, 1987; Bostock et al., 1991; Markus et al., 1995).

4.5.2 Experiment 4.2: Long-Term Depression is Required for Isomapping

The purpose of this experiment was to investigate the role that long-term depression (LTD) plays in determining whether or not an isomapping of output layer cell preferred firing directions will be produced.

The model was simulated with the parameters given in Table 4.1. In this experiment, the learning rule, which governs both the feedforward synaptic weights from the input cells to the output layer cells and the excitatory recurrent collateral synapses between the output layer cells, does not implement any form of LTD. Thus, the learning rule is a straightforward Hebbian associative learning rule as follows

$$\delta w_{ij} = (1 - w_{ij}) k r_i r_j \tag{4.7}$$

where w_{ij} is the strength of the synaptic connection between presynaptic cell j and postsynaptic cell

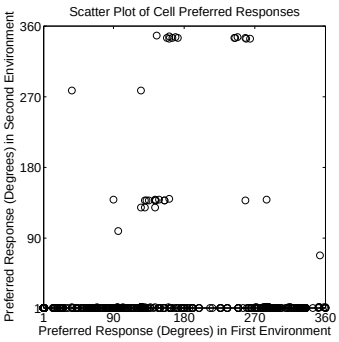

Figure 4.4: The learned responses of the output layer cells recorded from the model in Experiment 4.2. In this experiment, the model was simulated without long-term depression in the learning rule, i.e. the learning rule is a standard Hebbian learning rule given in Equation (4.7). The plot displays the preferred firing directions of the output layer cells in the first training environment plotted against the preferred firing directions of the output layer cells in the second environment. It is clear that the model has not learned an isomapping of the output layer cell preferred firing directions.

i, r_i is the firing rate of postsynaptic cell i, r_j is the firing rate of presynaptic cell j, and k is a learning rate parameter. In this learning rule, there is no subtraction of the cell average firing rate from the current firing rate of either the presynaptic or postsynaptic cell.

The results of a simulation conducted with the Hebbian associative learning rule, Equation (4.7), are displayed in Figure 4.4. The output layer cell preferred firing directions in one environment are plotted against the output layer cell preferred firing directions in the second environment. The scatterplot clearly shows that, without LTD in the learning rule, the model does not produce an isomapping of output layer cell preferred firing directions.

During training in the first environment, the recurrent collateral synapses, w_{ij}^{RC}, are strengthened between output layer cells that represent nearby orientations of the agent in the first environment. Then, during training in the second environment, the strong recurrent collateral synapses tend to enhance the firing of output layer cells that form localized clusters in the first training environment. The

homosynaptic LTD learning rule, Equation (4.6), ensures that active input cells only strengthen their feedforward synaptic connections w_{ij} with highly active output layer cells. This helps the input cells to selectively strengthen their feedforward synaptic connections to those output layer cells with firing rates enhanced by recurrent collateral synaptic connections, and weaken their feedforward connections to other less active output layer cells.

In this way, during training in the second environment, the homosynaptic LTD learning rule, Equation (4.6), helps individual input cells to selectively learn associations with output layer cells that form localized clusters in the first environment. This ensures that the feedforward synaptic connections w_{ij} from the input cells learn to respect the same topological arrangement of output layer cells across the two training environments, which produces isomapping. A learning rule with LTD, Equation (4.7), does not promote the correct specificity in the strengthening of the feedforward w_{ij} synaptic connections, and thus this learning rule does not promote isomapping.

4.5.3 Experiment 4.3: Heterosynaptic LTD Produces Isomapping

The learning rule given in Equation (4.6), employs homosynaptic LTD. That is, a strong presynaptic cell firing rate coupled with a below-average postsynaptic cell firing rate will result in a decrease in the strength of the synaptic connection between the presynaptic and postsynaptic cells. Experiment 4.3 was conducted to explore the effect of heterosynaptic LTD upon the isomapping of output layer cell preferred firing directions: heterosynaptic LTD occurs when a below-average presynaptic cell firing rate is coupled with a strong postsynaptic cell firing rate.

The model was simulated with the parameters given in Table 4.1. Homosynaptic LTD was not in-

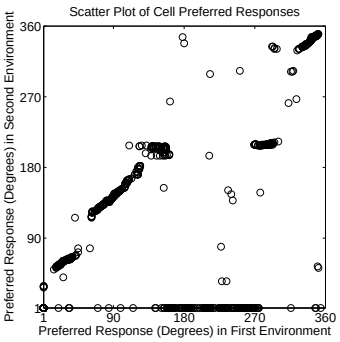

Figure 4.5: The learned responses of the output layer cells recorded from Experiment 4.3. The model was simulated with heterosynaptic LTD, Equation (4.8), but no homosynaptic LTD. The plot displays the preferred firing directions of the output layer cells in the first training environment plotted against the preferred firing directions of the output layer cells in the second training environment. The model can learn an isomapping of the output layer cell preferred firing directions across the two environments. However, the learned model performance is not as good as the case where homosynaptic LTD is incorporated in the learning rule (c.f. Experiment 4.1

corporated in the learning rule, but heterosynaptic LTD was present. The learning rule thus takes the following form

$$\delta w_{ij} = (1 - w_{ij})\Big[kr_i(r_j - \langle r_j \rangle)\Big] \tag{4.8}$$

where w_{ij} is the synaptic connection from the presynaptic cell j to the postsynaptic output cell i, r_i is the firing rate of postsynaptic cell i, r_j is the firing rate of presynaptic input cell j, $\langle r_j \rangle$ is the average firing rate of presynaptic input cell j, and k is a learning rate parameter.

The results of the simulation are displayed in Figure 4.5. The preferred firing directions of the output layer cells in one environment are plotted against the preferred firing directions of the output layer cells in the second environment. The data in the plot clearly demonstrate that an isomapping of output layer cell preferred directions can occur, but the isomapping is not as complete as the case when homosynaptic LTD is incorporated in the learning rule (c.f. Figure 4.3a).

4.5.4 Experiment 4.4: Synaptic Weight Bounding and Isomapping

Competitive neural networks usually incorporate some mechanism for bounding the synaptic weights. There are different methods of accomplishing synaptic weight bounding. A popular method is to bound the length of any given synaptic weight vector from a set of presynaptic input cells to a single postsynaptic output cell, so that the length of that vector is always constrained to be 1.0. An alternative method, employed in the experiments reported so far in this Chapter, is to limit each individual synaptic weight so that it can grow no bigger than 1.0. Both of these two methods of synaptic weight bounding are employed to ensure that the same postsynaptic cells do not always win the competition to fire when a competitive learning mechanism is employed. The aim of this experiment was to investigate whether or not synaptic weight bounding is required in order to produce an isomapping of output layer cell preferred firing directions.

The model was simulated with the parameters given in Table 4.1. The bounding term, $(1 - w_{ij})$, in the learning rule is omitted such that the learning rule is now a Hebbian associative rule incorporating homosynaptic LTD, which takes the following form

$$\delta w_{ij} = k(r_i - \langle r_i \rangle) r_j \tag{4.9}$$

where w_{ij} is the strength of the synaptic connection from presynaptic input cell j to postsynaptic output layer cell i, r_i is the firing rate of postsynaptic output layer cell i, $\langle r_i \rangle$ is the average firing rate of postsynaptic output cell i, r_j is the firing rate of presynaptic input cell j, and k is a learning rate parameter.

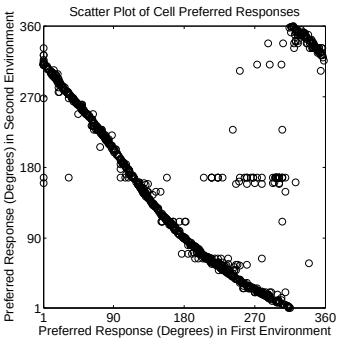

Figure 4.6: The learned responses of the output layer cells in Experiment 4.4. In this experiment, the model was simulated with Equation (4.9), which incorporates homosynaptic LTD but does not bound the synaptic weights. The plot displays the preferred firing directions of the output layer cells in the first training environment plotted against the preferred firing directions of the output layer cells in the second training environment. It is clear from the plot that the model can learn an isomapping of output layer cell preferred firing directions across training environments. The gradient of the plot is negative, but the plot is still linear, demonstrating that the absolute angular distance between the preferred firing directions of any two output layer cells is maintained across the two training environments, which is isomapping.

The results of the simulation are displayed in Figure 4.6. It is clear from the scatterplot that the model has produced an isomapping of output layer cell preferred firing directions. The gradient of the plot is negative, but the plot is linear so the absolute magnitude of the angular distance between the preferred firing directions of any two output layer cells is maintained across the two training environments; thus an isomapping is still produced. The negative gradient simply implies that two output layer cells that have preferred firing directions of say 1^{o} and 5^{o} in the first training environment will have preferred firing directions of say 315^{o} and 311^{o} in the second training environment. The absolute angular distance of 4^{o} between the preferred firing directions of the two cells is maintained across the two training environments. Consequently, it seems that an explicit synaptic weight bounding mechanism is not required in order for successful isomapping to occur.

It may, however, be the case that the presence of long term depression in the learning rule acts to implicitly constrain the values of the synaptic weights. But it would be hard to test this hypothesis

because, as the results reported in Experiment 4.2 (Section 4.5.2) show, it is not possible to achieve an isomapping of output layer cell preferred firing directions without there being some type of long term depression present in the learning rule. Therefore, dissociating whether the long term depression is able to implicitly bound the synaptic weights in the absence of explicit synaptic weight bounding may not be possible.

4.5.5 Experiment 4.5: The Learning Rate and Isomapping

The results reported so far in this chapter have demonstrated that successful isomapping is dependent upon a particular synaptic structure developing during training. The learning rate k scales the magnitude of the change to the strength of any particular synapse, and thus has a clear influence upon the nature of the synaptic structure that develops. This experiment was conducted to explore the effect that different values of the learning rate k have upon the learned behaviour of the model.

The model was simulated with the parameters given in Table 4.1. The learning rate k was varied as follows: 0.1, 0.01, 0.001.

The results of the simulations are displayed in Figure 4.7. There are three plots, and in each plot the output layer cell preferred firing directions in the first environment are plotted against the output layer cell preferred firing directions in the second environment. The left-hand plot displays the data recorded from a simulation with the learning rate k set to a value of 0.1. It is clear from the plot that the model is not able to learn an isomapping of output layer cell preferred firing directions. The middle plot displays the data recorded from a simulation with the learning rate k set to a value of 0.01. It is also clear that the model has not learned the required isomapping. The right-hand plot displays

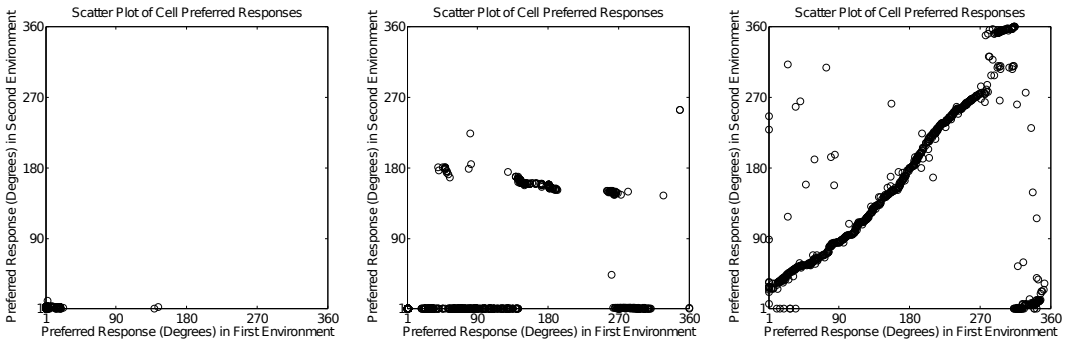

Figure 4.7: The learned responses of the output layer cells recorded from models simulated with different values of the learning rate k. All three plots display the preferred firing directions of the output layer cells in the first environment plotted against the preferred firing directions of the output layer cells in the second environment. The left-hand plot displays the data from a simulation with the learning rate k set to 0.1. It is clear that the learning rate is of too high a magnitude to produce an isomapping of output layer cell preferred firing directions. The middle plot displays the data from a simulation with the learning rate k set to a value of 0.01. It is also evident that the learning rate is of too high a magnitude to produce an isomapping. The right-hand plot displays data from a simulation with the learning rate set to 0.001. In this case, the learning rate is of the correct magnitude to produce the desired isomapping. Therefore, isomapping is dependent upon the value of the learning rate k, which must be set to a small enough value.

data recorded from a simulation with the learning rate k set to a value of 0.001. In this case, the model has successfully learned the required isomapping.

The value of the learning rate k thus has an influence upon the learned behaviour of the model. Higher values of k, such as 0.1 and 0.01, promote "greedy" postsynaptic cells that are permitted to strengthen their synapses with many of the currently active presynaptic cells. This has the effect that output layer cells with preferred firing directions close to one another in the first environment will become associated with input layer cells that are signalling orientations of the agent that are far away from one another in the second environment. Thus, the required isomapping will not develop.

It is only when k is set to a lower value, such as 0.001, that an isomapping of output layer cell preferred firing directions is produced.

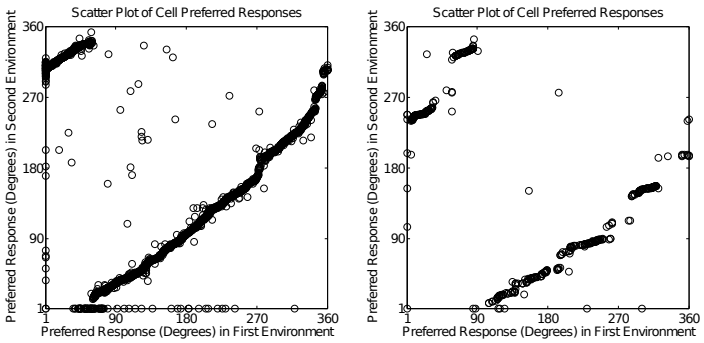

Figure 4.8: The learned responses of the output layer cells recorded from different sizes of model architecture. The left-hand plot displays the results from a simulation with 2500 output layer cells. The right-hand plot displays the results from a simulation with 400 output layer cells. The output layer cell preferred firing directions in the first environment are plotted against the output layer cell preferred firing directions in the second environment. It is clear that, for both network sizes, the model has learned an isomapping of output layer cell preferred firing directions across the two training environments. The model is capable of learning this isomapping independently of the size of the number of cells in the output layer.

4.5.6 Experiment 4.6: Isomapping is Independent of Network Size

The simulations comprising this experiment were conducted to demonstrate that the ability of the model to produce an isomapping of output layer cell preferred firing directions is independent of the number of output layer cells in the network.

The simulations were conducted with the model parameters given in Table 4.1. The only difference between the models was the number of output layer cells contained in each model. A "large" model was simulated with 2500 output layer cells. A "small" model was also simulated, containing 400 output layer cells.

The two plots in Figure 4.8 display the output layer cell preferred firing directions in the first environment plotted against the output layer cell learned preferred firing directions in the second environment. The left-hand plot displays the results recorded from the simulation of a network containing

2500 output layer cells. The right-hand plot displays the results recorded from the simulation of a network containing 400 output layer cells. It is evident from the two scatterplots that the model has, in both cases, learned an isomapping of the output layer cell learned responses profiles across the two training environments. Moreover, the phenomenon of isomapping is independent of the number of output layer cells contained in the network.

4.5.7 Experiment 4.7: Isomapping and the Amount of Training

The simulations in this experiment were conducted to show that the isomapping of output layer cell preferred firing directions is robust over a wide variation in the amount of training the model receives. The simulations were conducted with the model parameters given in Table 4.1. The simulations differed in the amount of training that the model received. A "long" simulation was conducted with the model undergoing 100 training epochs in each of the two training environments. Also, a "short" simulation was conducted, with the model receiving 25 epochs of training in each of the two training environments. This is contrasted with the "standard" simulation of 50 training epochs. The aim of this experiment was to demonstrate that the isomapping of output layer cell preferred directions will be produced over a wide range of training epochs.

The two plots in Figure 4.9 display the output layer cell preferred firing directions in one training environment plotted against the output layer cell preferred firing directions in the second environment. The left-hand plot displays data recorded from a simulation with the model permitted 100 training epochs in each training environment. The right-hand plot displays data recorded from a simulation where the model was permitted 25 epochs in each training environment. The two scatterplots clearly demonstrate that the model has learned an isomapping of the output layer cell preferred firing direc-

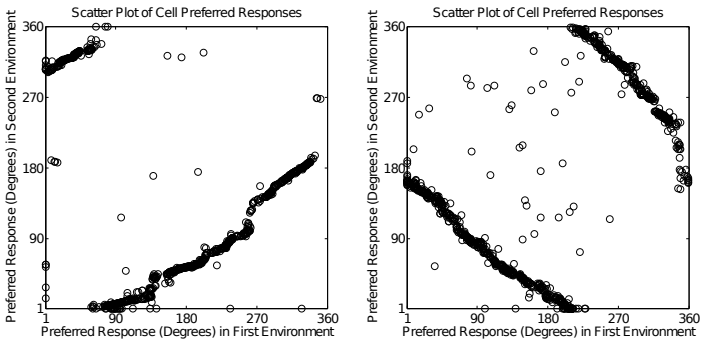

Figure 4.9: The learned responses of the output layer cells recorded from models exposed to different training durations. The left-hand plot displays data recorded from a simulation where the model was allowed 100 training epochs in each of the two training environments. The right-hand plot displays data recorded from a simulation where the model was allowed 25 training epochs to each of the two training environments. It is evident from these two scatterplots that the model is capable of learning an isomapping of the output layer cell preferred directions across two different environments over this wide variation in the amount of training received. In the right-hand plot, the model has still learned an isomapping of the output layer cell preferred firing directions: the gradient of the plot has simply changed direction. The plot is still linear, and this demonstrates that the absolute angular distance between the preferred firing directions of any two output layer cells is maintained across the two training environments.

tions across two different environments. The model is able to achieve this isomapping over a wide variation in the number of training epochs the model experiences in each training environment. Thus, the isomapping of output layer cell preferred firing directions is robust to the amount of training the model receives. It should be noted that in the simulation represented by the right-hand plot, the model has still achieved an isomapping of the output layer cell preferred firing directions. The gradient of the plot has simply changed direction. As the plot is still linear, the absolute angular distance between any two output layer cells is maintained across the two training environments.

4.6 Discussion

In this chapter, the implementation details, equations, and simulation results are provided for a neural network that can learn to produce an isomapping of output layer cell preferred firing directions across two environments. The main points of this chapter are:

- The model contains a single layer of input cells that are connected in a feedforward manner by a single layer of synapses to a layer of output cells. Within the layer of output cells, excitatory recurrent collateral synapses connect every individual output layer cell to every other output layer cell.

- A single training environment contains a single training location and a continuous circular space of distal objects located an infinite distance from the training location. The training environment does not contain any proximal objects. During training, the agent visits two training environments.

- The model must learn to maintain the angular distance between the preferred firing directions of any two output layer cells across the two environments. This task is referred to as learning an isomapping of output layer cell preferred firing directions across the two environments.

- The recurrent collateral synapses within the output layer help to produce the isomapping of output layer cell preferred firing directions. When the agent is trained in the first environment, output layer cells that develop preferred firing directions close to one another will also develop strong symmetric excitatory recurrent collateral synapses. These output layer cells will form localized clusters of activity. Upon entering the second training environment, some of the output layer cells will initially fire. Strong recurrent collateral synapses will encourage output layer cells that form a localized packet of activity in the first environment to also fire in the second environment. These output layer cells will strengthen their w_{ij} synapses with the currently active input layer cells. Thus, the currently active output layer cells will have preferred firing directions close to one another in the first environment and, because the currently active input layer cells will represent orientations of the agent that are close to one another in the second

environment, the output layer cells will also develop preferred firing directions that are close to one another in the second environment. This is how the isomapping of output layer cell preferred firing directions is achieved.

- In simulation, the model can learn an isomapping of output layer cell preferred firing directions across two training environments. The excitatory recurrent collateral synapses within the output cell layer play a crucial role in producing the isomapping. Firstly, lowering the value of the recurrent collateral scaling parameter, ϕ^{RC}, to 0.5 abolishes the isomapping and instead produces a place-cell-like complex remapping of preferred firing directions. Second, setting the value of the scaling parameter, ϕ^{RC}, to 0, which effectively eliminates the recurrent collaterals, also completely abolishes any isomapping.

- Changing the learning rule can have an effect upon the learned behaviour of the model. A learning rule that does not incorporate long-term depression fails to induce isomapping in the ouptut layer cells. If the learning rule incorporates heterosynaptic long-term depression then isomapping is still produced. Similarly, if the learning rule incorporates explicit weight normalization then isomapping is still produced. A low value of the learning rate k is necessary to produce isomapping.

- The model can produce isomapping over a wide range of the number of training epochs that the model experiences, and also independently of the number of output layer cells in the network. The ability to produce isomapping is thus reasonably robust.

In the simulations reported in this chapter, the training environments do not contain any proximal visual objects. This permits a set of carefully controlled experiments to be conducted with a minimal

model in order to explore the effect of various model parameters upon the model learned behaviour. The absence of proximal objects from the training environment thus allows a cleaner analysis of the simulation results. In principle, however, the model should still function correctly if proximal visual objects are present in the environment. A natural next step would therefore be to train the model in an environment that does contain proximal objects.

Also, the agent only visits two different training environments during the training regime. This protocol was chosen to allow for a more straightforward analysis of the simulation results. The model should still function if the agent is trained in several different training environments. The operational principles that produce the isomapping of output layer cell preferred firing directions are not dependent upon the number of training environments that the agent visits. Therefore, a further next step would be to train the agent across a larger range of different training environments. It is predicted that the model will still be able to produce the isomapping behaviour.

In conclusion, this chapter presents a neural network model that is capable of learning an isomapping of output layer cell preferred firing directions across two training environments. The isomapping is produced by the excitatory recurrent collateral synapses within the output layer. The isomapping behaviour is independent of the size of the network, and robust across a wide variation in the amount of training received, but is dependent, to a certain extent, upon the form of the learning rule. The simulation results of this neural network demonstrate how a complex behaviour of a population of head direction cells, such as the isomapping of preferred firing directions, can be learned on the basis of visual input alone.

Chapter 5

The Path Integration of Head Direction

The previous chapters of this thesis have explored how head direction cells develop their known responses to visual cues. The remainder of this thesis will present an exploration into how the head direction cell system can update itself in the absence of visual cues. This chapter provides a literature review that focuses on the path integration of head direction and previous neural network models that can perform path integration. It will be seen that none of the extant models of the path integration of head direction address how the synaptic connections can be set up by learning in a biologically plausible manner such that the internal representation of head direction is updated to match the true external head direction of the animal.

5.1 Path Integration

Path integration is a navigational process where an animal continuously integrates idiothetic (self-motion) cues to update its internal representation of its location and orientation within an environment in the absence of visual input (Mittelstaedt and Mittelstaedt, 1980, 1982; Redish, 1999). Path integration has been demonstrated in a range of animals, from arthropods (Collett and Zeil, 1998), to

mammals (Etienne et al., 1998; Etienne and Jeffery, 2004).

Successful path integration is based upon an animal possessing an internal representation, or map, of its current environment. An animal can use path integration in order to successfully return in a straight line to its approximate starting position when that starting position is out-of-sight of the current position of the animal (e.g. Wehner and Wehner, 1990; Séguinot et al., 1993). Animals can also achieve path integration in a novel environment (e.g. Etienne et al., 1998). This evidence suggests that an animal uses, and continuously updates, an internal representation of its current location and orientation within its environment.

The head direction cell system provides the orientation signal. This requires that the head direction cell system is capable of two feats of information processing:

i) When the head of the animal is stationary, the head direction cell system must maintain a stable representation of the current head direction in the absence of visual input.

ii) When the head of the animal is moving, the head direction cell system must update its representation of the current head direction to match the true head direction in the absence of visual input. In this case, the head direction cell system must update its internal representation using internal idiothetic (self-motion) signals.

When the head of the animal is stationary, the currently active head direction cell will continue to fire until the head moves to a new orientation, i.e. there is no habituation of the cell firing rate (Taube et al., 1990a). Moreover, when the head of the animal rotates, particularly in the absence of visual input, the head direction cell system will update its internal representation of head direction to track the actual

head direction (Goodridge et al., 1998). The head direction cell system satisfies these criteria because it functions as a continuous attractor neural network.

5.2 Continuous Attractor Neural Networks

A network of neurons, biological or simulated, with a recurrent collateral symmetric weight matrix between those neurons possesses a number of information processing properties. One of the most important of these properties is that the network will possess discrete attractor states (Hopfield, 1982, 1984). These attractor states, or memories, are stable states, each containing a small number of firing neurons. If the network is given input that is sufficiently close to, i.e. within the basin of attraction of, one of the attractor states, then the network will relax into that particular stable attractor state; thus a memory has been recalled. The network can only relax to one stable state at a particular moment in time. This type of network is known as a point attractor, or "Hopfield" network.

A neural network that has the same type of recurrent collateral synaptic connectivity as an attractor, but where the network is trained using a continuous input space, such as spatial input, will produce a *continuous attractor neural network* (Amari, 1977; Taylor, 1999). A continuous attractor possesses a continuum of attractor, or stable, states. Similar to a point attractor, a continuous attractor network can only relax into one particular stable state at a time. This particular stable state will be the memory that is currently recalled by the continuous attractor network. An example of a continuous attractor network that has settled into a stable state is given in Figure 5.1.

If a continuous attractor neural network is trained with visual input from distal cues representing

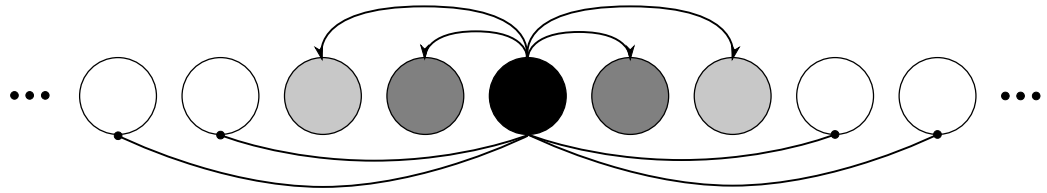

Figure 5.1: An example of a continuous attractor network. The figure displays a small portion of the cells in the network. Every cell is connected to every other cell, although this is not shown in the figure for clarity. When the correct balance is achieved between short-range excitation, represented by the arrows, and longer-range inhibition, represented by the dots, then a single persistent packet of activity will be present in the network. This activity packet occurs when the network has settled into one of its attractor states. Black represents a cell with a high firing rate, and white represents a cell that is not firing. The key property of a continuous attractor network is that, in principle, the network can stably support an activity packet at any point within a continuum of memory states.

head direction, each learned stable state of the continuous attractor will represent a particular head direction. The strength of the synaptic connection between any two neurons in this network will be proportional to the difference between the preferred firing directions of the two neurons. This is due to the associative Hebbian learning rule that updates the recurrent collateral synaptic connections between the neurons. Under this learning rule, the magnitude of the synaptic weight between the two neurons is directly proportional to the product of the firing rates of the two neurons. The firing rates of two neurons with preferred firing directions only a few degrees apart will produce a relatively large product, and thus a strong symmetric synaptic connection. The firing rates of two neurons with preferred firing directions that are the maximal distance, 180^{o}, apart will produce a relatively small product, and thus a weak symmetric synaptic connection. The properties of the synaptic weight matrix, in conjunction with the appropriate balance between excitatory and inhibitory inputs within the continuous attractor network, will ensure that the network can maintain a single persistent packet of neural activity, representing the current head direction, in the absence of any further perturbing input.

After training, the neurons in the continuous attractor network can be conceptually arranged in a ring

of head directions from $0^{\text{o}} - 360^{\text{o}}$. Neurons nearby in the ring represent nearby head directions, and are connected by strong excitatory recurrent collateral synaptic weights. Neurons far apart in the ring represent quite different head directions, and are connected by weak excitatory recurrent collateral synaptic weights. There is also global competition between all of the head direction cells in the ring, which, in the brain, would be mediated by inhibitory interneurons. The excitatory recurrent collateral synaptic weights permit the network to stably support a localized, e.g. Gaussian-shaped, packet of activity at any point in the ring. In this way, the network is able to represent a continuous space of memory states corresponding to head directions from $0^{\text{o}} - 360^{\text{o}}$.

Applying an external, asymmetric, input to the continuous attractor network will update the location of the activity packet within the network. There are different methods by which updating the packet of activity can be accomplished. Some of these methods will be discussed in the following sections of this chapter. Regardless of the exact mechanism, a network that can maintain a stable and persistent packet of neural activity in the absence of further input, but can also update the packet of neural activity in the presence of a perturbing input, has the desirable property of being an attractor-integrator network. This type of network is ideal for representing head direction.

Consider the following example. A rat has a current head direction of 40^{o}. In the absence of any further perturbing input, an appropriately trained continuous attractor network will maintain a stable representation of this directional heading. If the rat then performs a clockwise head rotation at a speed of $40^{\text{o}}/\text{s}$ for a total of two seconds, then the new head direction of the rat will 120^{o} (assuming angles increase in the clockwise direction). Further assuming that the continuous attractor network

is capable of updating to match this head rotation, the network will now maintain a stable packet of activity representing the new head direction of 120°. If the rat then performs a counter-clockwise head rotation at a speed of $15^{\circ}/\mathrm{s}$ for a total of one second, the new head direction of the rat will be 105°. The continuous attractor network will then update its packet of activity to represent the new head direction. Thus, this type of continuous attractor network is performing velocity path integration of head direction. Some of the neural architectures by which the path integration may be achieved are discussed in the next section.

5.3 Previous Models of the Path Integration of Head Direction

The first continuous attractor model of the head direction cell system was proposed by Skaggs et al. (1995).[1] In this particular model, the head direction cells are arranged in a ring topology, with cells representing nearby head directions placed close to one another in the ring. (Many of the models discussed in the following text share this topological property. In all cases, this particular arrangement of cells is a modelling convenience and should not be taken to imply that head direction cells possess a corresponding topography in the brain.) The head direction cells are connected, each cell to every other cell, by excitatory recurrent collateral synapses. There are also two populations of rotation, or turn-modulated, cells. One population of cells signals clockwise turns, and the other population of cells signals counter-clockwise turns. Both populations of rotation cells project in an offset manner to the population of head direction cells. The clockwise rotation cells also receive input from a cell signalling vestibular input resulting from a clockwise rotation. Likewise, the counter-clockwise rotation

[1] The first model of the head direction cell system was put forward by McNaughton et al. (1991). However, this model was not a continuous attractor network. Instead, the model took the form of a neural look-up table that associates the current head direction with the current head angular velocity and outputs the future head direction. As all the models of the head direction cell system discussed in this section incorporate some sort of continuous attractor network, the particular model of McNaughton et al. will not be described further.

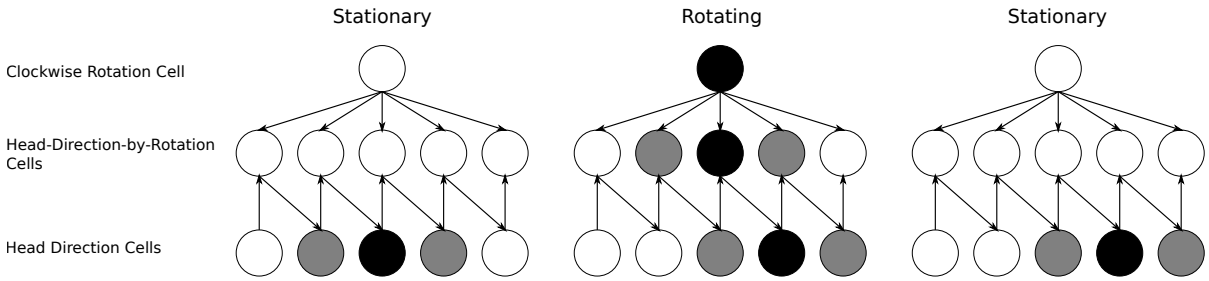

Figure 5.2: An example of the general network architecture employed by neural network models of the path integration of head direction. The model comprises a layer of head direction cells representing head directions from $0^{o} - 360^{o}$. The head direction cells are connected to one another through excitatory recurrent collateral synapses, although these are omitted from the figure for clarity. The head direction cells send excitatory synapses to a layer of head-direction-by-rotation cells, which, in turn, send excitatory synapses back to the layer of head direction cells in an offset manner. A clockwise rotation cell sends excitatory synapses to the layer of head-direction-by-rotation cells. In the left-hand plot the agent is stationary, and there is a persistent packet of neural activity in the head direction cell layer (Darker colours represent cells with higher firing rates). In the middle plot the agent is rotating in a clockwise direction. The clockwise rotation cell is now firing, which results in an activity packet emerging in the layer of head-direction-by-rotation cells. This in turn shifts the location of the activity packet, within the head direction cell layer, in the clockwise direction. In the right-hand plot the agent is stationary again. There is a persistent packet of activity in layer of head direction cells, but the location of the packet has shifted in a clockwise direction in comparison to the left-hand plot. The shift is due the offset excitatory synapses from the layer of head-direction-by-rotation cells. The model also incorporates a counter-clockwise rotation cell, and a layer of head-direction-by-rotation cells that send offset excitatory synapses to the layer of head direction cells to shift an activity packet in the head direction cell layer in the counter-clockwise direction. These cells have been omitted from the figure for clarity.

cells receive input from a cell signalling vestibular input resulting from a counter-clockwise rotation.

This neural architecture is capable of sustaining a packet of persistent neural activity representing the current head direction of the rat when the rat is stationary. Also, the network can update the packet of head direction cell activity to represent the changing head direction of the rat as its head rotates. Figure 5.2 presents an illustration of the general type of network architecture employed by the extant neural network models of the path integration of head direction.

The model of Skaggs et al. (1995) is a theoretical proposal and there are no simulation, or analytical, results. It remains an influential model, however, and most neural network models of path integration

in the head direction cell system bear some degree of resemblance to this original model; particularly in the use of a continuous attractor network. In the following text these models will be discussed.

5.3.1 Models with Predetermined Synaptic Weights

Sharp et al. (1996) present a model, with simulation results, of head direction cells in the postsubiculum and anterior thalamic nucleus of the rat brain. Similar to the model of Skaggs et al. (1995), this model contains two populations of cells, postulated to be located in the reticular thalamic nucleus, that signal head angular velocity modulated by head direction. One population of cells signals clockwise rotations of the head. The second population of cells signals counter-clockwise rotations of the head. These angular-head-velocity modulated head direction cells have offset inhibitory connections to the pure head direction cells such that, in the absence of any input signalling a rotation of the head, a persistent packet of neural activity, representing the current head direction of the rat, will be present in the population of head direction cells. When the head begins to rotate, the packet of head direction cell activity will move through the population of head direction cells to track the current head direction. The simulated neurons are implemented with biophysically accurate and detailed equations, although the authors do not incorporate the known, approximately Gaussian, tuning curve of head direction cell responses.

Zhang (1996) describes a model that is implemented as a ring of head direction cells. The behaviour of these cells is governed by leaky-integrator firing rate dynamics. Simulation and analytical results show that the model is capable of maintaining a stable and persistent packet of neural activity. The model can also update the location of this packet of neural activity in order to reflect the changing head direction of the animal. The stable packet of neural activity is achieved through the means of

a symmetric synaptic weight matrix (as discussed in Section 5.2). To shift the location of the packet of activity, an asymmetric component must be introduced into the weight matrix. A criticism of this model is that in order to switch between different model states (stable packet of activity, and shifting packet of activity) an instantaneous, online, change in the synaptic weight matrix must take place. Zhang does not address how such a change may be biologically implemented. Thus, in the absence of any hypothesis regarding such an implementation, the model remains biologically implausible.

Redish et al. (1996) also propose a model, together with simulation results, of head direction cells in the rat postsubiculum and anterior thalamic nucleus. In comparison to the models of Sharp et al. (1996) and Zhang (1996), the model of Redish et al. implements two connected continuous attractor networks: one network in the postsubiculum; and one network in the anterior thalamic nucleus. The two attractor networks are reciprocally connected, and the strength of these offset connections is modulated by the firing of cells that signal head angular velocity (either clockwise or counter-clockwise). The model of Redish et al. is simulated with integrate-and-fire neurons and is able to reproduce the known tuning curves of head direction cell responses. The model is also able to accurately integrate real rat head direction trajectories. A problem with the model, however, is the dynamic modulation of the strength of the postsubiculum to anterior thalamic nuclei excitatory connections. It is not known, and the authors do not address, how such a multiplicative effect upon synaptic strength could be achieved, particularly in an online manner.

To address the problem of rapid, online, modulation of synaptic efficacies, Goodridge and Touretzky (2000) report a model that simulates three distinct brain regions containing head direction cells:

the postsubiculum, the anterior thalamic nucleus, and the lateral mammillary nucleus. The cells in the lateral mammillary nucleus are divided into two sub-populations: cells that respond to clockwise angular head velocity, and cells that respond to counter-clockwise angular head velocity. These two sub-populations of cells project to the head direction cells in the anterior thalamic nucleus with the appropriate excitatory synaptic offsets. The neurons were simulated with continuous dynamics, and output an instantaneous firing rate instead of individual action potentials. In simulation, the model was able to replicate the deformation of anterior thalamic nucleus head direction cell responses seen under clockwise and counter-clockwise rotations of varying speeds. The model was also able to account for the anticipatory time interval in cell firing seen in the anterior thalamic nucleus head direction cells.

Xie et al. (2002) identified a problem with previous models of path integration in the head direction cell system. In order for these models to achieve accurate integration of angular head velocity information, and thus correctly update the representation of head direction, the models had to be implemented with very fast synaptic time constants (in the order of 1ms), particularly for fast rotational velocities. The model of Xie et al. (2002) takes the form of a single population of head direction cells spread across two ring networks with asymmetric connections, e.g. a head direction cell is only allowed to receive excitatory connections from neighbours to its right on the same ring, and from neighbours to its left on the opposite ring. The neurons are simulated with continuous dynamics, and the results show that the network is able to accurately integrate a wide range of angular head velocities such that the internal representation of head direction is correctly updated. Moreover, Xie et al. (2002) showed that successful integration is possible even with the relatively slow synaptic time constants associated with NMDA or GABA-B synapses. A problem, however, is that the model does not attempt

to address how the representation of head direction, within a single population of head direction cells, could take the form of a double-ring attractor network.

5.3.2 Models without Excitatory Recurrent Collateral Synapses

Research into the origin of the head direction signal, and thus the attractor-integrator required for path integration, suggests that the primary candidate is a reciprocal loop between the lateral mammillary nucleus and the dorsal tegmental nucleus of Gudden. Single-cell recording studies have revealed the existence of head direction cells in the lateral mammillary nucleus (Blair et al., 1998; Stackman and Taube, 1998) and the dorsal tegmental nucleus of Gudden (Sharp et al., 2001). Moreover, lesions of the dorsal tegmental nucleus have been shown to disrupt the directional firing of head direction cells recorded from the anterior thalamic nucleus (Bassett and Taube, 2001a). Bilateral lesions of the lateral mammillary nucleus have also been shown to disrupt directional firing in anterior thalamic nucleus head direction cells (Blair et al., 1998). As lesions of the anterior thalamic nucleus disrupt the directional firing of head direction cells in the postsubiculum, but not vice versa, (Goodridge and Taube, 1997), it would appear highly likely that the seat of the head direction signal is the lateral mammillary nucleus and the dorsal tegmental nucleus of Gudden.

The models described in the previous section all incorporate excitatory recurrent collateral synapses between the head direction cells. As discussed in Section 5.2, the purpose of these recurrent collateral synapses is to allow a persistent packet of neural activity, representing head direction, to exist in the continuous attractor network in the absence of any external perturbation of the network. It is doubtful, however, that these excitatory recurrent collateral synapses exist in the required regions of the putative head direction cell circuit. For instance, if the origin of the head direction signal is the dorsal tegmen-

tal nucleus to lateral mammilary nucleus circuit then anatomical studies have yet to find evidence of excitatory recurrent collateral synapses in the lateral mammillary nucleus (Allen and Hopkins, 1988). A continuous attractor network need not, however, be implemented with excitatory recurrent collateral synapses. Provided that there exists some mechanism for producing and maintaining a stable and persistent packet of neural activity, the network will be capable of possessing a continuum of attractor states and thus functioning as a continuous attractor network. Although excitatory recurrent collateral synapses are proposed as one implementation of the mechanism for generating persistent neural activity, the three studies that will be discussed in the following text demonstrate that it is possible to implement a fully functioning continuous attractor network using mechanisms other than excitatory recurrent collateral synapses.

Rubin et al. (2001) propose a model of head direction cells in the thalamus. Rubin et al. suggest a two-layer model of reciprocally linked thalamo-cortical relay cells and thalamic reticular cells. Neural activity is generated in, and spread through, the network by means of postinhibitory rebound in the thalamo-cortical cells. The neurons in the model are implemented as conductance-based integrate-and-fire neurons. In simulation, Rubin et al. (2001) show that their model is capable of maintaining a persistent packet of neural activity that is stable and does not drift. It is also demonstrated that the model could propagate the packet of neural activity through the network in a manner that would be appropriate for tracking the current head direction of the animal. The study does not, however, reproduce realistic head direction cell tuning curves. Nor does the study provide a description of how the correct angular head velocity signals would be input to the network in order to update the current representation of head direction. But the model of Rubin et al. (2001) does demonstrate that it

is possible to maintain and update a persistent packet of neural activity without requiring excitatory recurrent collateral synapses; thus providing an alternative implementation of a path integration of head direction mechanism.

Song and Wang (2005), based upon an earlier proposal by Song and Wang (2003), describe a model of the reciprocal loop between the lateral mammillary bodies and the dorsal tegmental nucleus. There is no anatomical evidence for excitatory recurrent collateral synapses in either of these brain regions, but there is evidence that the dorsal tegmental nucleus sends inhibitory inputs to the lateral mammillary nucleus (Shibata, 1987; Allen and Hopkins, 1989; Wirtshafter and Stratford, 1993). There is also evidence that the lateral mammillary nucleus sends excitatory inputs to the dorsal tegmental nucleus (Allen and Hopkins, 1989, 1990). The lateral mammillary nucleus and the dorsal tegmental nucleus thus appear to form a reciprocal processing loop. The neurons are implemented as conductance-based integrate-and-fire neurons. In simulation, Song and Wang show that the model is capable of maintaining a stable and persistent packet of neural activity in the absence of any rotational signal, and the model can update this packet of neural activity to track the current head direction of the animal. It is also shown that the model could accurately integrate real angular head velocity data recorded from a rat to within a small margin of error. The model is also capable of reproducing the known tuning curves of head direction cells and proposes plausible mechanisms for inputting the angular head velocity signal to the head direction cell system.

Research by Boucheny et al. (2005) examined the proposed model of Song and Wang (2003). Through simulation and analytical results, Boucheny et al. explored the behaviour and dynamics of the model

proposed by Song and Wang (2003). Like Song and Wang, Boucheny et al. showed that the model can maintain a stable and persistent packet of neural activity in the absence of rotational input; and when rotational input is applied, the packet of neural activity will move through the lateral mammillary nucleus head direction cell network to reflect the changing head direction of the animal. In addition, Boucheny et al. demonstrated that the current location of the neural activity packet is capable of rapid updating by the presence of visual cues.

5.3.3 Models with Self-Organizing Synaptic Weights

The extent of self-organization in the head direction cell system remains an open question. Although a degree of predetermined wiring may be present in the head direction cell circuit, it is unlikely that the entire system is prespecified in minute detail: the dynamics of continuous attractor networks are such that a slight imbalance between excitatory and inhibitory inputs can lead to a degradation in the ability of the network to maintain a stable and persistent packet of neural activity. It is more plausible that the head direction cell system is fine-tuned through learning instead of the entire system being genetically hard-wired to the required high degree of precision. The following text will consider models of the path integration of head direction that, to varying extents, exhibit self-organization, through learning, of their synaptic weight matrices.

In the model of Stringer et al. (2002), all synaptic weights fully self-organize. That is, none of the synaptic weight matrices are prespecified. The model is implemented as a population of head direction cells, together with clockwise and counter-clockwise rotation cells signalling rotation of the animal in the respective directions. The neurons are implemented as leaky-integrator firing rate cells. All synaptic strengths are altered through an associative Hebbian learning rule that, for each

synapse, factors in a temporal trace, i.e. memory, of both presynaptic and postsynaptic cell activity. In simulation, Stringer et al. show that the model is capable of maintaining a stable and persistent packet of neural activity in the absence of visual input. When one of the population of rotation cells commences firing, signalling either clockwise or counter-clockwise rotation, the packet of neural activity will move through the head direction cell network, reflecting the changing head direction of the animal. A limitation of the model, however, is that, given the synapses self-organize, the model does not learn to perform path integration at the correct speed, i.e. update the packet of head direction cell activity to match the true head direction of the simulated agent. Another limitation is that the model employs Sigma-Pi synaptic connections that associate together a combination of two or more presynaptic inputs with the postsynaptic firing rate. The Sigma-Pi synapses are therefore higher-order synapses. It is not known how such synapses could be implemented in the brain, and thus the model suffers from a lack of biological plausibility.

Stringer and Rolls (2006) propose a more biologically plausible version of their earlier model. In this new model, there is a population of head direction cells, a population of rotation cells (with two subpopulations signalling clockwise and counter-clockwise rotation respectively), and a population of combination cells that represent combinations of head direction and rotation. All neurons are implemented as leaky-integrator firing rate neurons. All synapses in the model fully self-organize through associative Hebbian learning. The learning rule for the synapses from the population of combination cells to the population of head direction cells incorporates a temporal trace of presynaptic combination cell activity. By using a separate population of combination cells, Stringer and Rolls obviate the need for the Sigma-Pi synapses that were necessary in the previous model. Stringer and Rolls show that

this new model is capable of maintaining a stable and persistent packet of activity in the population of head direction cells in the absence of visual input. When the rotation cells start firing, reflecting the changing head direction of the animal, the combination cells in turn commence firing and send excitatory inputs to the population of head direction cells. The memory trace in the synaptic learning rule has the effect that the current firing rate of a postsynaptic head direction cell is associated, through learning, with the firing rate of a presynaptic combination cell some time in the past. As the combination cells represent combinations of a particular head direction and a rotation, the model effectively learns an association between a current head direction, a rotation, and a head direction some time in the past. This ensures that each combination cell projects most strongly to a head direction cell with a preferred firing direction that is different to the preferred firing direction of the head direction cell that the combination cell is most strongly stimulated by. This offset in the synaptic connections between head direction cells provides the requisite asymmetry to drive a packet of neural activity through the population of head direction cells and thus reflect the changing head direction of the animal. The model of Stringer and Rolls suggests a biologically plausible method by which the path integration of head direction can be learned. The model does not, however, learn to perform path integration at the correct speed.

In the model of Hahnloser (2003), a population of head direction cells is split across a double-ring network. The head direction cells receive visual and vestibular input signalling rotation of the head. The synapses are trained by an error-correcting learning rule. Hahnloser shows that, in comparison to the naive untrained model, the model after learning is capable of maintaining a stable and persistent packet of neural activity in the absence of any input signalling rotation of the head. When such input

did signal rotation of the head, the model is capable of shifting the packet of neural activity through the population of head direction cells to reflect the changing head direction of the animal. Hahnloser also demonstrates that the trained model is capable of accurately integrating angular head velocity signals to reflect the current head direction over a wide range of angular head velocities. A criticism of this model, however, is in its use of an error-correcting learning rule. This type of learning rule compares the actual output of a cell with the desired output of that cell if the model were functioning correctly. The synaptic weights are then adjusted up or down to bring the actual output closer to the desired output on future updates of the model. This type of error-correcting learning rule requires that a teaching, or error, signal is propagated back through the network to earlier synapses. It is doubtful that such a neural architecture generally exists in the brain, with the exception of a few known regions such as the cerebellum (Rolls and Treves, 1998).

Stratton et al. (2010) propose a hybrid model of the head direction cell system. Arguing that some pre-specification of a synaptic weight matrix is likely to be found in the head direction cell circuit, Stratton et al. prespecify the synaptic weight matrix for a population of lateral mammillary nucleus head direction cells, but the weight matrix initially contains a deliberate offset. Without correction, this offset ensures that the population of head direction cells is incapable of maintaining a stable and persistent packet of neural activity, as the packet will continuously drift through the population of cells. Two populations of dorsal tegmental nucleus angular head velocity cells signal clockwise and counter-clockwise rotation of the animal respectively. These angular head velocity cells have offset connections to the head direction cells and are thus responsible for shifting the packet of neural activity through the population of head direction cells when the animal is moving its head. There is also

a population of symmetric angular head velocity cells that project to the population of head direction cells and signal the speed at which the animal is moving its head. All of the neurons are implemented as spiking, integrate-and-fire neurons. All of the modifiable synapses are trained with an associative Hebbian learning rule. In simulation, Stratton et al. show that through training, the model can correct the initial bias in the head direction cell synaptic weight matrix and thus maintain a stable and persistent packet of neural activity reflecting the current head direction in the absence of input from the angular head velocity cells. When the animal starts to rotate, and therefore the angular head velocity cells commence firing, the packet of head direction cell activity is shifted through the population of head direction cells to accurately reflect the changing head direction of the animal.

5.3.4 Summary of Previous Models

The previous models of the velocity path integration of head direction have proposed a variety of neural network architectures through which this navigational process can be realised. Each of these models demonstrates the ability to maintain a persistent and stable packet of head direction activity in the absence of any perturbing input. In the presence of an external drive, or internal synaptic weight asymmetry, the packet of head direction cell activity moves through the population of head direction cells to reflect the changing head direction of the animal.

A particular problem that none of these models address, with the exception of Hahnloser (2003), is how the network might learn to update the packet of head direction cell activity so that it moves through the head direction cell population at the same speed as the animal is moving its head. That is, the internal continuous attractor network representation of head direction must learn to accurately track the true current head direction when the animal is rotating its head. The model of Hahnloser

does accurately track the velocity of head rotation. To achieve this, however, requires some special neural architecture and an error-correcting synaptic learning rule. Neither of these two mechanisms is entirely biologically plausible. Thus, there remains a requirement for a self-organizing model of the head direction cell system that learns, in a biologically plausible manner, to update a packet of head direction cell activity that accurately reflects the true current head direction of the animal.

5.4 Biologically Plausible Timing Mechanisms

To successfully update a representation of head direction at the same speed as the animal is moving its head requires a calibrated system that can, over short time intervals, accurately reflect the true head direction of the animal. In order to achieve this calibration, the system must posses a mechanism that provides temporally accurate updates of the state of the system. In other words, some form of biologically plausible timing mechanism is required.

5.4.1 Axonal Conduction Delays

An axonal conduction delay is defined as the length of time it takes for an action potential to travel from the soma of a presynaptic neuron, along the length of the axon, to reach the synapse with a postsynaptic neuron. The length of the axon of the presynaptic neuron will remain constant, which in turn will provide a constant time interval for an action potential to travel from the presynaptic neuron to the postsynaptic neuron. This constant time interval allows associations to be learned, over the duration of the time interval, between the activity of the presynaptic neuron at time t_1 and the activity of the postsynaptic neuron at time t_2; where t_2 is $t_1 + \Delta t$, with Δt defined as the length of the axonal conduction delay for the presynaptic neuron in question. Thus, at a future time when the presynaptic

cell is active, the postsynaptic cell will, under the right conditions, become active $t_1 + \Delta t$ after the presynaptic cell, i.e. the postsynaptic cell will become active at the correct time interval after the presynaptic cell was active. Therefore, the correct temporal delay between presynaptic and postsynaptic cell activations can, after learning, be replayed on future occasions.

With regards to the head direction cell system, axonal conduction delays can be used to enable the system to learn to update the internal representation of the current head direction at the same speed as the animal is moving its head. When a presynaptic head direction cell representing a head direction θ_1 is active at time t_1, the axonal conduction delay will result in the signal from this presynaptic head direction cell reaching a postsynaptic cell Δt later. Thus, this postsynaptic cell will receive a representation of the head direction of the animal Δt in the past. During the time it takes for the action potential to travel down the axon of the presynaptic head direction cell, the true head direction of the animal will have changed (assuming the animal is moving its head). Thus, an association between a previous head direction θ_1, and a new head direction θ_2, can be learned as a result of the axonal conduction delay Δt. On a later occasion, if the animal is moving its head with the same velocity as previous occasions, then activation of the presynaptic head direction cell will result in the postsynaptic cell becoming active at the correct moment, accurately reflecting the current head direction of the animal. That axonal conduction delays may aid a neural network in accurately processing temporal information has been previously proposed (e.g. Carr and Konishi, 1988, 1990; Konishi, 1991; Carr, 1993; Gerstner et al., 1996).

5.4.2 Neuronal Time Constants

A second biologically plausible timing mechanism occurs when there are different values for the presynaptic and postsynaptic neuronal time constants. A neuronal time constant is here defined as the time constant that governs the rate of the rise and decay of the level of activation of a postsynaptic neuron, where the level of activation is driven by a linear weighted sum of the presynaptic inputs to the postsynaptic neuron in question. In the brain, there are many different time constants including membrane time constants, and active and passive time constants (Koch et al., 1996; Koch, 1999). In this thesis, a neuronal time constant, unless otherwise described, will represent a generic time constant governing the rate of the rise and decay of a particular postsynaptic cell activation level, rather than a time constant related to the specific morphology of a particular cell.

Neuronal time constants are able to provide a timing mechanism between two connected cells as follows. If the postsynaptic neuron has a large time constant, the postsynaptic neuron will become active at a time that is later than the time at which the presynaptic neuron becomes active (assuming the instantaneous transmission of an action potential from the presynaptic neuron to the postsynaptic neuron). Thus, an association can be learned between the activity of the presynaptic neuron at time t_1, and the activity of the postsynatic neuron at time t_2; where t_2, in this particular case, is defined as $t_1 + \Delta t$, with Δt defined as the time difference between the activations of the presynaptic and postsynaptic neurons. After learning has taken place, activation of the presynaptic neuron at time t_1 will result in activation of the postsynaptic neuron at time t_2. Assuming the time constants of the presynaptic and postsynaptic neurons remain constant, the postsynaptic neuron will always become active a constant time interval Δt after the presynaptic neuron has become active. This constant time

interval, Δt, will allow, after learning, for a sequence of neural activity to be replayed in the correct order and at the correct speed.

Neuronal time constants can aid a head direction cell system to accurately update its internal representation of the current head direction. Consider two linked pre- and postsynaptic head direction cells representing two different head directions, θ_1 and θ_2, respectively. If the postsynaptic head direction cell has a large time constant, then the postsynaptic head direction cell will always become active a particular time interval Δt after the presynaptic head direction cell has become active. Consequently, an association through time can be learned between the two cells representing the two different head directions θ_1 and θ_2. On a future occasion, the activation of the presynaptic head direction cell representing head direction θ_1 will result in the activation of the postsynaptic head direction cell representing head direction θ_2 at a time Δt in the future. Thus, a head direction cell network will learn to accurately update its internal representation of the head direction to match the true current head direction.

5.5 The Appropriate Level of Modelling

A detailed discussion of the different approaches to modelling a neural system can be found in Section 1.5 of this thesis. Consequently, that discussion will not be repeated here.

The models of path integration presented in the following two chapters will employ a rate-coded paradigm. The main difference between the modelling scheme of the path integration models and the modelling schemes employed in Chapters 2, 3, and 4 is that the models of path integration employ

a differential scheme for the updating of the cell activations, firing rates, and synapses. That is, the neural network models of the path integration of head direction operate in the space of continuous time. The differential scheme is employed because an important measure of the functionality of the path integration models is whether or not they can learn to perform path integration at the correct speed, which requires the dimension of time to be accurately modelled through differential equations.

5.6 Chapter Summary

This chapter reviewed the literature on the velocity path integration of head direction. No extant neural network models of the head direction cell system have addressed how this system may learn, in a biologically plausible manner, to perform the path integration of head direction at the correct speed. That is, how the head direction cell system may update an internal representation of current head direction to match the true external head direction. In summary, the main points of this chapter are:

- Path integration is a navigational process where an animal continuously updates an internal representation of its current orientation and location based upon idiothetic (self-motion) signals.
- The head direction cell system provides the orientation signal. The head direction cell system is capable of maintaining a stable and persistent packet of neural activity representing the current head direction of the animal in the absence of visual input. The head direction cell system can also update the current representation of head direction to accurately reflect the changing head orientation of the animal in the absence of visual input. In this case, the head direction cell system must use internal idiothetic (self-motion) signals to update its internal representation. Both of these properties are required in a path integration mechanism.

- In order to exhibit these two properties, the head direction cell system must incorporate some form of continuous attractor neural network architecture.

- Previous modelling studies of the path integration of head direction have assumed that the head direction cell system does incorporate a continuous attractor architecture. These models can be placed into one of three categories: models with predetermined synaptic weight matrices and therefore no learning; models that do not incorporate excitatory recurrent collateral synapses and also do not learn; and, models that self-organize, through learning, to produce the path integration of head direction.

- All of these models are capable of maintaining a stable and persistent packet of neural activity, representing the current head direction, in the absence of any perturbing input. External drive to these networks, often in the form of vestibular input, updates the internal representation of head direction to match the true external head direction. None of these models, however, address how the network may learn to update the internal representation of head direction at the correct speed in a biologically plausible manner.

- Axonal conduction delays – the time it takes for a signal to pass from the body of the presynaptic neuron down the length of its axon – are one class of natural time interval that can be used as a biologically plausible timing mechanism to facilitate the updating, at the corrects speed, of the internal representation of the current head direction. A second biologically plausible timing mechanism is to use different values of the neuronal time constants that govern the rate of the rise and decay of postsynaptic cell activation levels.

In the next two chapters of this thesis, two neural network models will be presented that can learn, using biologically plausible architectures and learning mechanisms, to perform the velocity path integration of head direction at approximately the same speed as the simulated agent is rotating its head.

Chapter 6

Path Integration of Head Direction: Axonal Conduction Delays

This chapter reports the implementation details, model equations, and simulation results for a neural network model that can learn to perform path integration of head direction at the speed of rotation imposed during training. No previous neural network model has achieved this in a biologically plausible manner. An exploration of the effect upon the learned model behaviour of controlled alterations of the model parameters is presented in detail. Future modifications of the current model are also discussed.

6.1 Model Architecture

The network architecture is presented in Figure 6.1. The model comprises two reciprocally connected layers of cells: a layer of head direction cells; and a layer of combination cells. The layer of head direction cells (with firing rate r_i^{HD} for head direction cell i) represents the current head direction of the agent, and operates as a continuous attractor network performing path integration of head direction. Within the layer of head direction cells there is a single Gaussian packet of activity. The centre of the

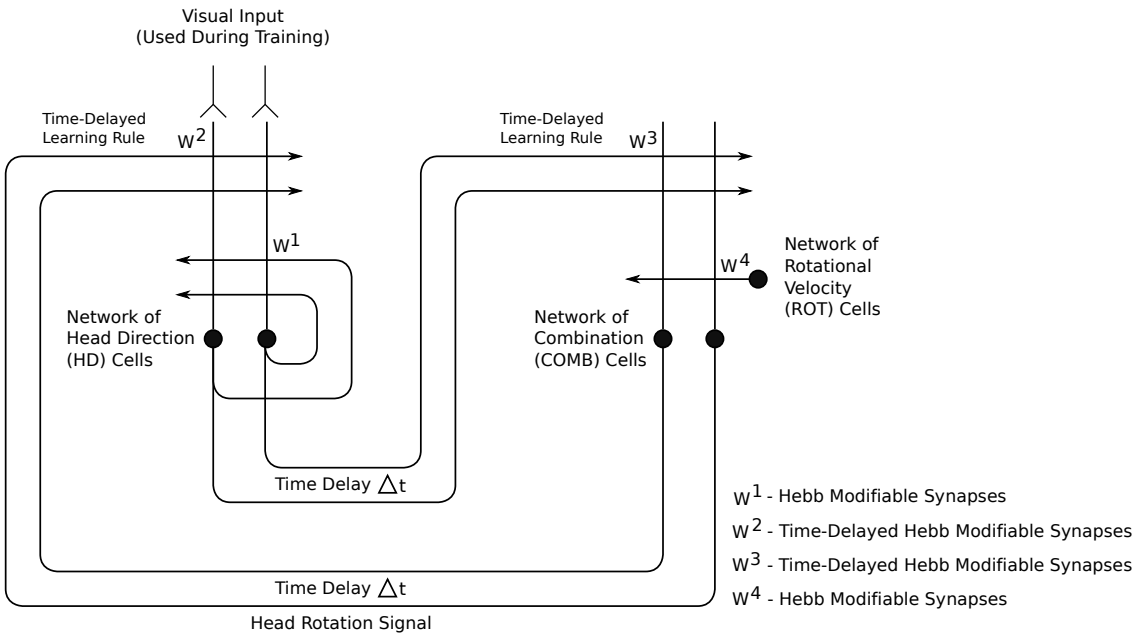

Figure 6.1: The neural network architecture for a model that learns to perform path integration of head direction at the speed experienced during training. Explicit axonal conduction delays are incorporated into the w_{ij}^2 and w_{ij}^3 synapses to provide a timing mechanism over which associations between activity packets in the head direction cell network and the combination cell network may be learned. The w_{ij}^1 recurrent synapses help to produce a continuous attractor network in the head direction cell layer, and thus support a stable and persistent packet of head direction cell activity when the agent is not rotating. The w_{ij}^4 synapses provide inputs from the layer of rotational velocity cells signalling the current direction of rotation of the agent. All of the synapses are Hebb-modifiable, with the exception of the visual input to the head direction cell layer, which is only present during the training of the model.

activity packet represents the current head direction of the simulated agent.

The layer of combination cells (with firing rate r_i^{COMB} for combination cell i) receives inputs from both the layer of head direction cells and a layer of rotational velocity cells (with separate sub-populations of cells signalling clockwise and counter-clockwise directions of rotation). The combination cells operate as a competitive network and develop their firing properties during training, with individual combination cells learning to represent a combination of a particular rotational velocity and a particular head direction. Head direction cells in the lateral mammillary nucleus exhibit modulation of their firing rate by angular head velocity, i.e. the firing rate increases as the angular

head velocity increases (Stackman and Taube, 1998). In addition, a small number of angular head velocity cells in the dorsal tegmental nucleus exhibit modulation of their firing rate by head direction, with an increase in the firing rate when the head of the rat is within a broad contiguous range of head directions (Bassett and Taube, 2001b; Sharp et al., 2001).

Half the population of rotational velocity cells signal when the agent is rotating in a clockwise direction, and half the population of rotational velocity cells signal when the agent is rotating in a counter-clockwise direction. The firing rates of the rotational velocity cells are binary such that an individual cell is either firing or not firing, i.e. the possible firing rates are either 0 or 1.

Each layer of synapses, $w_{ij}^1, \ldots, w_{ij}^4$, is Hebb-modifiable through associative learning. The exception to this is the visual input to the head direction cells. These visual inputs are only present during the training phase, and are non-modifiable so as to force the head direction cell layer to reflect the current head direction of the agent.

This network architecture is proposed as the minimal synaptic architecture required to update a packet of head direction cell activity in the dark at the same speed as was imposed during training in the light. The model is not intended to directly represent any one specific are of the brain known to contain head direction cells.

6.2 Operational Principles of the Model

The model incorporates axonal conduction time delays Δt in the propagation of signals from the layer of head direction cells to the layer of combination cells, and then back again from the combination cells to the head direction cells. The effect of the axonal conduction delays is that when a presynaptic cell is driving a postsynaptic cell, the activity profile of the postsynaptic cell will be delayed in time by an amount Δt from the activity profile of the presynaptic cell. This delay Δt will be equal to the axonal conduction delay Δt between the presynaptic and the postsynaptic cell, i.e. the length of time it takes for signal to propagate from the presynaptic cell to the postsynaptic cell.

The firing-rate-based associative Hebbian learning rule, governing the strength of the synaptic connection between the presynaptic and postsynaptic cell, also takes account of the axonal conduction delay Δt. Thus the presynaptic cell, with instantaneous firing rate r_j at time t, will, through synaptic strengthening, learn to stimulate the postsynaptic cell with instantaneous firing rate r_i at time $t + \Delta t$. In this way the model is able to learn associations between neural activity in the head direction cell and combination cell layers over specific fixed time intervals Δt, and thus learn to shift a packet of head direction cell activity at the correct speed.

To fully explore the operational principles of the model, consider a simple two-cell system in which signals take a short time interval Δt to pass from the presynaptic cell j to the postsynaptic cell i due to the natural time delay of axonal transmission. A consequence of this conduction delay is that the value of the synaptic weight w_{ij} between the two cells will depend upon the value of the presynaptic cell firing rate $r_j(t - \Delta t)$ that occurred Δt before the current postsynaptic cell firing rate $r_i(t)$. This

reflects the time Δt taken for the presynaptic signal to reach the postsynaptic cell. The incorporation of an axonal conduction delay Δt into the presynaptic cell firing rate term $r_j(t - \Delta t)$ in both the governing equation for the postsynaptic cell activation h_i

$$\tau \frac{dh_i(t)}{dt} = -h_i(t) + w_{ij}(t) r_j(t - \Delta t)$$

and the Hebbian associative learning rule for the synapse between the presynaptic and postsynaptic cells

$$\frac{dw_{ij}(t)}{dt} = k r_i(t) r_j(t - \Delta t)$$

allows the presynaptic and postsynaptic cells to learn a temporally delayed association. That is, if, during learning, the postsynaptic cell was active Δt after the presynaptic cell was active, i.e. so that the product $r_i(t) r_j(t - \Delta t)$ is large, then the presynaptic cell learns to stimulate the postsynaptic cell at a time Δt in the future. (Given that the signals still take Δt to travel from the presynaptic cell to the postsynaptic cell due to the time delay Δt in the governing equation for the postsynaptic cell activation h_i.)

In the manner described above, the correct time intervals between cell activities can be learned and replayed by cells in the model. This is essential for the model to be able to replay the temporal sequence of cell activity at the speed that was experienced during learning. The synaptic learning rule governing weight adaptation is a standard Hebbian rule normally written without the $(t - \Delta t)$ shown in the preceding equation, which merely draws attention to the fact that the presynaptic firing rate

takes time Δt to travel to the postsynaptic cell.

In the full version of the model, displayed in Figure 6.1, temporally-delayed associative learning is achieved over fixed time delays Δt for both the afferent and the efferent connections between the layer of head direction cells and the layer of combination cells. If the agent is rotating with angular velocity $\frac{\mathrm{d}\theta}{\mathrm{d}t}$, then over the time $2\Delta t$ it takes for the signal from the head direction cells to be propagated to the combination cells and back to stimulate a new subset of head direction cells, the agent will have rotated $2\Delta t\frac{\mathrm{d}\theta}{\mathrm{d}t}$. The network learns that an activity pattern in the head direction cell layer at time t_1 representing a particular head direction θ_1 should stimulate (via the layer of combination cells) a new pattern of activity in the head direction cell layer at time $t_2 = t_1 + 2\Delta t$, representing a later head direction $\theta_2 = \theta_1 + 2\Delta t\frac{\mathrm{d}\theta}{\mathrm{d}t}$. This kind of associative learning over fixed time intervals enables the model to learn the correct velocity for updating the packet of neural activity in the head direction cell layer, and thus allows the path integration of head direction to occur at the correct speed.

6.3 Model Equations and Implementation

This section first presents the governing equations of the model, then outlines the training and testing protocol, and finally discusses the numerical solution of the differential equations.

6.3.1 Governing Equations

During the training phase, each head direction cell i receives an external visual input e_i that carries information about the current head direction of the agent. When visual cues are available, these external inputs will dominate all other excitatory inputs to the head direction cell layer and will drive

the activity levels of the head direction cells. Each individual head direction cell is set to respond maximally to visual input from a unique preferred head direction of the agent with a Gaussian profile as described later in Equations (6.13) and (6.14). Hence, the effect of the visual inputs is to stimulate a single packet of neural activity in the head direction cell layer. The location of this activity packet reflects the current head direction of the agent. The Gaussian tuning of individual head direction cell response profiles in the light will lead, for any given head direction cell, to an increase in cell firing as the head direction of the agent moves towards the preferred head direction of that cell, and a decrease in cell firing as the head direction of the agent moves away from the preferred head direction of the cell.

The activation level h_i^{HD} of head direction cell i is governed by

$$\begin{aligned}\tau^{\mathrm{HD}}\frac{dh_i^{\mathrm{HD}}(t)}{dt} = &- h_i^{\mathrm{HD}}(t) + e_i(t) \\ &- \frac{1}{N^{\mathrm{HD}}}\sum_j \tilde{w}^{\mathrm{HD}} r_j^{\mathrm{HD}}(t) \\ &+ \frac{\phi_1}{C^{\mathrm{HD}\to\mathrm{HD}}}\sum_j w_{ij}^1(t) r_j^{\mathrm{HD}}(t) \\ &+ \frac{\phi_2}{C^{\mathrm{COMB}\to\mathrm{HD}}}\sum_j w_{ij}^2(t) r_j^{\mathrm{COMB}}(t-\Delta t) \\ &- I^{\mathrm{EXTERN}}\end{aligned} \tag{6.1}$$

where the activation level $h_i^{\mathrm{HD}}(t)$ is driven by the following terms. The term $\sum_j \tilde{w}^{\mathrm{HD}} r_j^{\mathrm{HD}}(t)$ represents inhibitory feedback within the head direction cell layer, where the summation is performed over all presynaptic head direction cells j; $\tilde{w}^{\mathrm{HD}}$ is a global constant describing the effect of inhibitory interneurons within the layer of head direction cells, and N^{HD} is total number of head direction cells in

the model. The term $\sum_j w_{ij}^1(t) r_j^{\mathrm{HD}}(t)$ represents excitatory feedback through the recurrent collateral synapses within the layer of head direction cells, where the summation is over those presynaptic head direction cells which have excitatory synapses onto the postsynaptic head direction cell i. Within this term, $r_j^{\mathrm{HD}}(t)$ is the firing rate of presynaptic head direction cell j at time t, and $w_{ij}^1(t)$ is the value, at time t, of the excitatory (positive) synaptic weight from presynaptic head direction cell j to postsynaptic head direction cell i.[1]

Further terms in Equation (6.1) are as follows. The term $-h_i^{\mathrm{HD}}(t)$ is a decay term such that, in the absence of further presynaptic input, the activation level of postsynaptic head direction cell i will decay to zero according to the time constant τ^{HD}. The term $e_i(t)$ represents an external visual input to postsynaptic head direction cell i at time t, which, during training, is set according to the Gaussian profile given by Equations (6.13) and (6.14). When there is no visual input the term e_i is set to zero. Thus, in the absence of visual input, the key term driving the head direction cell activation levels in Equation (6.1) is the sum of the input from the presynaptic combination cells $\sum_j w_{ij}^2(t) r_j^{\mathrm{COMB}}(t - \Delta t)$. Within this term, $r_j^{\mathrm{COMB}}(t - \Delta t)$ is the firing rate of presynaptic combination cell j at a time Δt in the past, and $w_{ij}^2(t)$ is the value, at time t, of the excitatory synaptic weight from presynaptic combination cell j to postsynaptic head direction cell i.[2] The term I^{EXTERN} is a constant that represents external feedforward inhibition to the head direction cell network: this is necessary during the learning phase to ensure that, in the presence of visual input, only a small subset of the head direction cells (those representing head directions nearby the actual head direction of the agent) are active at any one point

[1] The scaling factor $\frac{\phi_1}{C^{\mathrm{HD}\to\mathrm{HD}}}$ controls the overall strength of the recurrent inputs to the network of head direction cells, where ϕ_1 is a constant and $c^{\mathrm{HD}\to\mathrm{HD}}$ is the number of synapses onto each postsynaptic head direction cell from the presynaptic head direction cells.

[2] The scaling factor $\frac{\phi_2}{C^{\mathrm{COMB}\to\mathrm{HD}}}$ controls the overall strength of the combination cell inputs, where ϕ_2 is a constant and $C^{\mathrm{COMB}\to\mathrm{HD}}$ is the number of synapses onto each postsynaptic head direction cell from the presynaptic combination cells.

in time, i.e. the standard deviation of the head direction cell layer activity packet remains small. In the absence of external visual input, during the testing phase, the term I^{EXTERN} is set to zero.

The firing rate $r_i^{\mathrm{HD}}(t)$ of postsynaptic head direction cell i at time t is determined from the activation level $h_i^{\mathrm{HD}}(t)$ of that cell and the sigmoid activation function

$$r_i^{\mathrm{HD}}(t) = \frac{1}{1 + \mathrm{e}^{-2\beta(h_i^{\mathrm{HD}}(t) - \alpha)}} \tag{6.2}$$

where α and β are the sigmoid threshold and slope respectively. The head direction cells are implemented as firing-rate based cells, and thus the emission of individual action potentials is not simulated.

The recurrent synapses in the head direction cell layer are trained by a local associative Hebbian learning rule

$$\frac{dw_{ij}^1(t)}{dt} = k^1 r_i^{\mathrm{HD}}(t) r_j^{\mathrm{HD}}(t) \tag{6.3}$$

where k^1 is the learning rate, $r_i^{\mathrm{HD}}(t)$ is the firing rate of postsynaptic head direction cell i at time t, and $r_j^{\mathrm{HD}}(t)$ is the firing rate of presynaptic head direction cell j at time t. This learning rule increases the strength of the synapses between those head direction cells that represent nearby directions in the head-direction space, and which tend to be co-active due to broadly-tuned overlapping Gaussian receptive fields. To bound the synaptic weights, weight normalization is used. The recurrent weights $w_{ij}^1(t)$ are rescaled after every timestep of the learning phase to ensure that for each postsynaptic head direction cell i

$$\sqrt{\sum_j (w_{ij}^1(t))^2} = 1 \tag{6.4}$$

where the sum is over all the presynaptic head direction cells j. Such a process of weight normalization may be achieved in the brain by synaptic weight decay (Oja, 1982; Rolls and Treves, 1998), or conservation of the total synaptic weights through a mixture of LTP and LTD (Stent, 1973; Royer and Paré, 2003). The renormalization helps to ensure that the learning rule (6.3) is convergent in the sense that recurrent weights within the head direction cell layer settle down over time to steady values, i.e. the weights do not grow unbounded.

During the learning phase, associative synaptic modification in the recurrent synapses w_{ij}^1, in conjunction with the continuous nature of the head-direction space, allows the strength of the synapses between any two head direction cells to reflect the distance between the head directions represented by these two cells. The recurrent connectivity of the w_{ij}^1 synapses allows the head direction cell layer to operate as a continuous attractor network and support stable packets of neural activity in the absence of external visual input; thus, the agent can operate in the dark.

The learning rule to update, at time t, the $w_{ij}^2(t)$ synapse from presynaptic combination cell j to postsynaptic head direction cell i is given by

$$\frac{dw_{ij}^2(t)}{dt} = k^2 r_i^{\mathrm{HD}}(t) r_j^{\mathrm{COMB}}(t - \Delta t) \tag{6.5}$$

where k^2 is the learning rate, $r_i^{\mathrm{HD}}(t)$ is the firing rate of postsynaptic head direction cell i at time t,

and $r_j^{\mathrm{COMB}}(t - \Delta t)$ is the firing rate of presynaptic combination cell j at a time $t - \Delta t$ in the past. This learning rule forms associations betweens packets of activity occurring at the current time t in the head direction cell layer, and packets of activity occurring at time $t - \Delta t$ in the past in the combination cell layer.

In order to bound the $w_{ij}^2(t)$ synaptic weights, rescaling takes place after every timestep, as in (6.4), to ensure that for each postsynaptic head direction cell i

$$\sqrt{\sum_j (w_{ij}^2(t))^2} = 1 \tag{6.6}$$

where the sum is over all the presynaptic combination cells j.

During the testing of the model in the absence of visual input, the w_{ij}^2 synapses are responsible for stimulating head direction cells representing head directions further along in the direction the agent is currently rotating. The w_{ij}^2 synapses help to shift the location of the head direction cell layer activity packet and, thus, perform path integration of head direction.

The combination cells are driven by synaptic inputs $w_{ij}^3(t)$ from the head direction cell layer, and synaptic inputs $w_{ij}^4(t)$ from the layer of rotational velocity cells. During the training phase, the w_{ij}^3 and w_{ij}^4 synapses onto the combination cells self-organize using associative Hebbian competitive learning rules, which enables the combination cell layer to operate as a competitive network that learns to represent different combinations of a particular head direction and a rotational velocity.

The activation level of a postsynaptic combination cell i is governed by

$$\begin{aligned}\tau^{\mathrm{COMB}}\frac{dh_i^{\mathrm{COMB}}(t)}{dt} = &- h_i^{\mathrm{COMB}}(t) \\ &- \frac{1}{N^{\mathrm{COMB}}}\sum_j \tilde{w}^{\mathrm{COMB}} r_j^{\mathrm{COMB}}(t) \\ &+ \frac{\phi_3}{C^{\mathrm{HD}\to\mathrm{COMB}}}\sum_j w_{ij}^3(t) r_j^{\mathrm{HD}}(t-\Delta t) \\ &+ \frac{\phi_4}{C^{\mathrm{ROT}\to\mathrm{COMB}}}\sum_j w_{ij}^4(t) r_j^{\mathrm{ROT}}(t) \end{aligned} \tag{6.7}$$

with the terms defined as follows. The term $\sum_j \tilde{w}^{\mathrm{COMB}} r_j^{\mathrm{COMB}}(t)$ represents inhibitory feedback within the layer of combination cells, where the summation is performed over all presynaptic combination cells j; $\tilde{w}^{\mathrm{COMB}}$ is the global lateral inhibition constant describing the effect of inhibitory interneurons with the combination cell network, and N^{COMB} is the number of combination cells in the model. The term $\sum_j w_{ij}^3(t) r_j^{\mathrm{HD}}(t-\Delta t)$ is the synaptic input from the head direction cell layer, where $w_{ij}^3(t)$ is the value, at time t, of the excitatory synaptic connection from presynaptic head direction cell j to postsynaptic combination cell i, and $r_j^{\mathrm{HD}}(t-\Delta t)$ is the firing rate of presynaptic head direction cell j at a time Δt in the past.[3] The term $\sum_j w_{ij}^4(t) r_j^{\mathrm{ROT}}(t)$ is the synaptic input from the layer of rotational velocity cells, where the firing rate of presynaptic rotational velocity cell j at time t is given by $r_j^{\mathrm{ROT}}(t)$, and $w_{ij}^4(t)$ is the value, at time t, of the excitatory synaptic weight from presynaptic rotational velocity cell j to postsynaptic combination cell i.[4]

[3]The scaling factor $\frac{\phi_3}{C^{\mathrm{HD}\to\mathrm{COMB}}}$ controls the overall strength of the synaptic inputs from the head direction cell layer, where ϕ_3 is a constant and $C^{\mathrm{HD}\to\mathrm{COMB}}$ is the number of synapses onto each postsynaptic combination cell from the presynaptic head direction cells.

[4]The scaling factor $\frac{\phi_4}{C^{\mathrm{ROT}\to\mathrm{COMB}}}$ controls the overall strength of the inputs from the layer of rotational velocity cells, where ϕ_4 is a constant and $c^{\mathrm{ROT}\to\mathrm{COMB}}$ is the number of synapses onto each postsynaptic combination cell from the presynaptic rotational velocity cells.

The firing rate $r_i^{\mathrm{COMB}}(t)$ of postsynaptic combination cell i at time t is determined from the activation level of that cell, $h_i^{\mathrm{COMB}}(t)$, and the sigmoid activation function

$$r_i^{\mathrm{COMB}}(t) = \frac{1}{1 + \mathrm{e}^{-2\beta(h_i^{\mathrm{COMB}}(t) - \alpha)}} \tag{6.8}$$

where α and β are the sigmoid threshold and slope respectively. The threshold α is set to a high value to ensure that each individual postsynaptic combination cell j will function in a similar manner to a logical AND gate. That is to say that temporally conjunctive inputs from the presynaptic head direction cells and the presynaptic rotational velocity cells are required in order for the postsynaptic combination cells to fire. In this manner, after training, individual combination cells become selective to combinations of a particular head direction occurring a small time interval Δt in the past and an instantaneous rotational velocity. The combination cells are implemented as firing-rate based cells, and thus the emission of individual action potentials is not simulated.

The synaptic weight $w_{ij}^3(t)$ from presynaptic head direction cell j to postsynaptic combination cell i is updated according to the local associative Hebbian learning rule

$$\frac{dw_{ij}^3(t)}{dt} = k^3 r_i^{\mathrm{COMB}} r_j^{\mathrm{HD}}(t - \Delta t) \tag{6.9}$$

where k^3 is the learning rate, $r_i^{\mathrm{COMB}}(t)$ is the firing rate of postsynaptic combination cell i at time t, and $r_j^{\mathrm{HD}}(t - \Delta t)$ is the firing rate of presynaptic head direction cell j at a time Δt in the past. This learning rule forms associations between packets of activity occurring at the current time t in the combination cell layer, and packets of activity occurring at time $t - \Delta t$ in the past in the head direction

cell layer.

The synaptic weights w_{ij}^3 are bound by rescaling after every timestep to ensure that for each postsynaptic combination cell i

$$\sqrt{\sum_j (w_{ij}^3(t))^2} = 1 \tag{6.10}$$

where the summation is performed over all presynaptic head direction cells j.

The synaptic weight $w_{ij}^4(t)$ from a presynaptic rotational velocity cell j to a postsynaptic combinaton cell i is updated according to a local associative Hebbian learning rule

$$\frac{dw_{ij}^4(t)}{dt} = k^4 r_i^{\mathrm{COMB}}(t) r_j^{\mathrm{ROT}}(t) \tag{6.11}$$

where k^4 is the learning rate, $r_i^{\mathrm{COMB}}(t)$ is the firing rate of postsynaptic combination cell i at time t, and $r_j^{\mathrm{ROT}}(t)$ is the firing rate of presynaptic rotational velocity cell j at time t. The learning rule associates together temporally conjunctive packets of activity in the rotational velocity cell layer and the combination cell layer.

The weights w_{ij}^4 are rescaled after each timestep so that for each postsynaptic combination cell i

$$\sqrt{\sum_j (w_{ij}^4(t))^2} = 1 \tag{6.12}$$

where the sum is over all presynaptic rotational velocity cells j.

After training, activity within the combination cell layer is driven by the head direction cell layer if, and only if, the rotational velocity cells are also active. If the rotational velocity cells cease firing, i.e. the agent is stationary, then the activity in the combination cell layer decays to zero according to the time constant τ^{COMB}.

6.3.2 Training and Testing Protocol

During the training phase, the agent revolves through 360^{o} at a single point in the training environment. A full clockwise rotation of the agent through the consecutive bearings 0^{o}-360^{o}, followed by a full counter-clockwise rotation of the agent through consecutive bearings 360^{o}-0^{o}, constitutes a single epoch of training. When the agent has performed 50 training epochs then the training phase is complete. Throughout the training phase, the activation levels and firing rates of the head direction cells are simulated according to Equations (6.1) and (6.2).

When training with visual input available, the external visual inputs e_i, given in Equation (6.1), are the dominant influence upon the activation levels, and thus firing rates, of the head direction cells. The head direction cells are mapped onto a regular grid of different head directions such that each postsynaptic head direction cell i has a preferred head direction x_i of the agent at which that cell will be maximally stimulated by visual input. Thus, for every bearing of the agent during a training epoch, the visual input e_i to a particular postsynaptic head direction cell i is calculated according to the following Gaussian response profile

$$e_i^{\mathrm{HD}} = \lambda \mathrm{e}^{-(s_i^{\mathrm{HD}})^2/2(\sigma^{\mathrm{HD}})^2} \tag{6.13}$$

where s_i^{HD} is the difference between the actual head direction x of the agent and the preferred head direction x_i for postsynaptic head direction cell i; λ is a scaling factor that expresses the strength of the non-modifiable visual input synapses onto the postsynaptic head direction cells, and σ^{HD} is the standard deviation of the Gaussian response profile.

For each postsynaptic head direction cell i, the difference s_i^{HD} is given by

$$s_i^{\mathrm{HD}} = \mathrm{MIN}(|x_i - x|, 360 - |x_i - x|) \tag{6.14}$$

which has the effect of allowing wrap-around such that the Gaussian response profiles of the population of head directions are continuous through 360^{o}.

The activation levels and firing rates of the combination cells are updated at each timestep of the simulation according to Equations (6.7) and (6.8) respectively.

At the start of the training phase, the synaptic weights w_{ij}^1, w_{ij}^2, w_{ij}^3, and w_{ij}^4 are all initialized to random positive values. These weights are updated at every timestep of the training phase according to Equations (6.3), (6.5), (6.9), and (6.11) respectively, and thus allowed to self-organize.

After the completion of the training phase, the model is tested to determine if the packet of head direction cell activity can update on the basis of idiothetic rotation signals alone (the external visual inputs e_i are set to zero). During the testing phase Equations (6.1), (6.2), (6.7), and (6.8) are simulated but without any further learning in any of the synapses, i.e. Equations (6.3), (6.5), (6.9), and (6.11)

are not simulated.

At the start of the testing phase, all of the firing rates r_i^{HD}, r_i^{COMB}, and r_i^{ROT} are set to zero. The agent is oriented to an initial head direction and simulated with visual input available, but with no rotational velocity cells active, for a period of $1\mathrm{s}$. For the period that the agent maintains this head direction, the visual input term e_i for each postsynaptic head direction cell i is set to a Gaussian response profile according to Equation (6.13). The visual input is then removed by setting all e_i terms to zero, and the agent is allowed to rest at the same initial head direction for a further $1\mathrm{s}$. The purpose of the initial part of this testing is to allow the development of stable packet of activity, representing the initial head direction, within the head direction cell layer.

The agent remains stationary for a further $1\mathrm{s}$, then the firing rates r_i^{ROT} of rotational velocity cells $1 - 250$ signalling clockwise rotation are set to 1 for a period of $1\mathrm{s}$. The packet of head direction cell activity should thus move through the head direction cell continuous attractor network and track the instantaneous head direction of the agent.

The firing rates r_i^{ROT} of all the rotational velocity cells are then set to zero for $1\mathrm{s}$. When the 250 rotational velocity cells signalling clockwise rotation cease firing, the combination cells in turn will cease firing. Consequently, the packet of head direction cell activity should remain stationary within the continuous attractor network, reflecting the last-visited head direction of the agent. In this case the model has demonstrated that it is capable of maintaining a stable packet of activity representing any given head direction.

Rotational velocity cells $251 - 500$, signalling counter-clockwise rotation, then have their firing rates set to 1 for 1s. The head direction cell activity packet is then updated to reflect the counter-clockwise rotation of the agent.

The firing rates of all the rotational velocity cells are then set to zero for a further 1s to demonstrate that the continuous attractor network in the head direction cell layer can maintain a stable packet of head direction cell activity at any given head direction.

Once the training and testing phases have finished, the simulation of the model is considered complete and the model performance is analysed.

6.3.3 Numerical Solution of the Differential Equations

The differential equations (6.1), (6.3), (6.5), (6.7), (6.9), and (6.11) cannot be solved analytically. Instead, discrete approximations to the solutions of these equations must be arrived at during simulation. A Forward Euler finite difference scheme is used to approximate all the differential equations during simulation.

If the activation level $h_i(t)$ of a postsynaptic cell i at time t is given by the differential equation

$$\tau \frac{dh_i(t)}{dt} = -h_i(t) + f_i(t)$$

where the $f_i(t)$ term represents a sum of the driving presynaptic inputs to postsynaptic cell i at time

t, then the Forward Euler approximation to this differential equation is given by

$$h_i(t + \delta t) = h_i(t) - \frac{\delta t}{\tau} h_i(t) + \frac{\delta t}{\tau} f_i(t)$$

where $h_i(t + \delta t)$ is the value of h_i to be approximated at time $t + \delta t$, and δt is the timestep of the Forward Euler method. In other words, the differential equation is replaced by a difference equation where, at a given timestep, the new value of the function, $h_i(t + \delta t)$, is approximated on the basis of the currently known values $h_i(t)$ and $f_i(t)$, and the timestep of the Forward Euler method δt.

Similarly, if the value, at time t, of the synaptic weight $w_{ij}(t)$ between presynaptic cell j and postsynaptic cell i is given by the differential equation

$$\frac{dw_{ij}(t)}{dt} = k r_i(t) r_j(t)$$

then the Forward Euler approximation to this differential equation is given by

$$w_{ij}(t + \delta t) = w_{ij}(t) + \delta t \left[k r_i(t) r_j(t)\right]$$

where $w_{ij}(t + \delta t)$ is the value of the synaptic weight to be approximated at time $t + \delta t$ on the basis of the currently known values $w_{ij}(t)$, $r_i(t)$, and $r_j(t)$ together with the timestep of the Forward Euler method δt.

It is fairly straightforward to expand from these simplified example differential equations, to the application of the Forward Euler method to the full differential equations given above.

6.4 Results

This section presents the results of an exploration into the effect that controlled alterations of the model parameters have upon the learned behaviour of the model. For all the simulations reported in the following text, the timestep, δt, of the Forward Euler numerical scheme was set to 0.00005s. Simulating the model with smaller values of δt confirmed that the numerical solution had converged by $\delta t = 0.00005$s.

6.4.1 Experiment 6.1: Demonstration of Core Model Performance

The simulations comprising this experiment were conducted to explore the main operational principles of the model. In this section, the results of four main simulations are presented and analysed. The model is simulated with values of 50ms and 100ms for the axonal conduction delay, and for each delay the model is simulated rotating at velocities of 180°/s and 360°/s.

100ms Axonal Conduction Delay; 180°/s Rotational Velocity

The network parameters used during the simulations are displayed in Table 6.1. The network was simulated according to the training and testing protocol described in Section 6.3.

Figure 6.2 displays the recurent w^1 synaptic weights within the head direction cell layer after training with a Hebbian associative learning rule, Equation (6.3), and weight normalization, Equation (6.4). For all of the plots, the w^1 synaptic weight distribution is clearly symmetric about the individual presynaptic head direction where the w^1 synaptic strength is maximal. Due to the Hebbian learning rule and weight normalization, the presynaptic head direction cell j with maximal synaptic weight

Table 6.1: Network parameter values for the simulations reported in Chapter 6

Network Parameters	
No. Head Direction Cells	500
No. Combination Cells	1000
No. Rotational Velocity Cells	500
No. w^1 synapses onto each head direction cell	500
No. w^2 synapses onto each head direction cell	1000
No. w^3 synapses onto each combination cell	25
No. w^4 synapses onto each combination cell	500
$\tilde{w}^{\mathrm{HD}}$	250.0
$\tilde{w}^{\mathrm{COMB}}$	50.0
I^{EXTERN}	90.0
σ^{HD}	20.0^{o}
k^1, k^2, k^3, k^4	0.1
τ^{HD}, τ^{COMB}	1.0ms
λ	100.0
ϕ_1	3.75×10^3
ϕ_2	2.5×10^3
ϕ_3	1.25×10^3
ϕ_4	4.0×10^2
No. Training Epochs	50
δt (timestep of Forward Euler method)	0.00005s
Head Direction Cell Sigmoid Transfer Function Parameters	
α	0.0
β	0.1
Combination Cell Sigmoid Transfer Function Parameters	
α	10.0
β	0.3

All values are constant across all experiments reported in this chapter except where noted in the text.

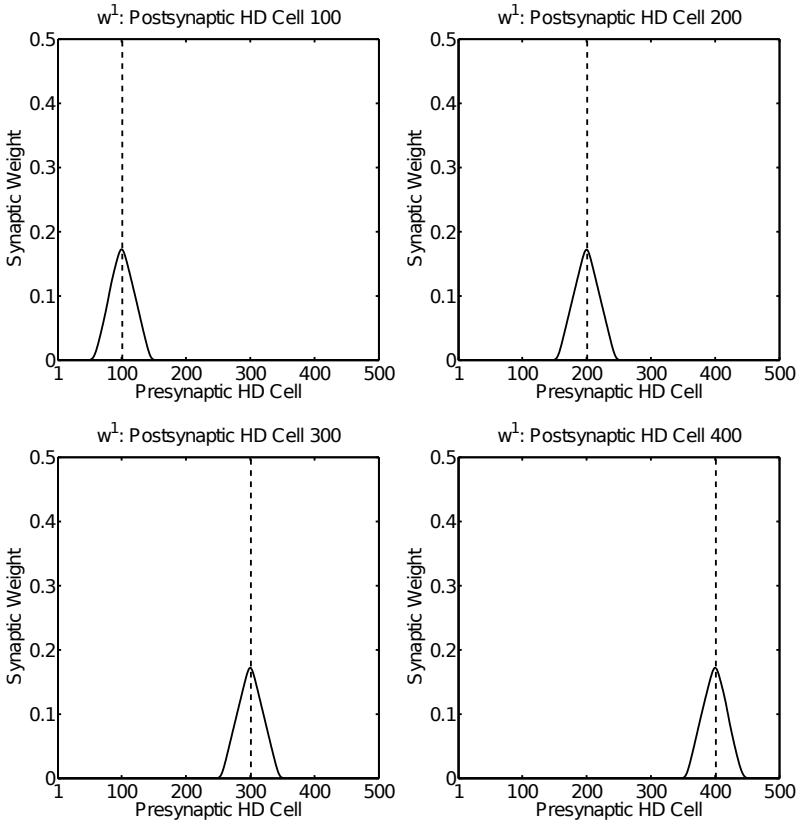

Figure 6.2: The recurrent synaptic weights w^1 within the head direction (HD) cell layer after training with a Hebbian associative learning rule and weight normalization. These synaptic weights are taken from a simulation implemented with an axonal conduction delay of 100ms, and a rotational velocity imposed during training of 180°/s. All the other network parameters are given in Table 6.1. Each of the four plots displays the learned synaptic weights to a different postsynaptic HD cell from the other 500 presynaptic HD cells in the HD cell layer. In the plots the presynaptic HD cells are arranged according to where they fire maximally in the head-direction space of the agent when visual input is available. For each plot a dashed vertical line indicates the presynaptic HD cell with which the postsynaptic HD cell has maximal w^1 synaptic weight. The Hebbian learning rule and weight normalization, Equations (6.3) and (6.4), ensure that the presynaptic head direction cell j with maximal w^1_{ij} synaptic weight to postsynaptic head direction cell i will in fact be head direction i itself. In all of the plots, the synaptic weight profile is symmetric about this presynaptic HD cell. This symmetry helps to support a stable packet of activity during testing in the absence of visual input.

w^1_{ij} to postsynaptic head direction cell i will be head direction cell i itself. That is, each postsynaptic head direction cell i has maximal w^1 synaptic strength with itself and the synaptic weight distribution tails off symmetrically. This symmetry ensures that a packet of head direction cell activity can be maintained in the head direction cell layer when the agent is stationary in the absence of visual input.

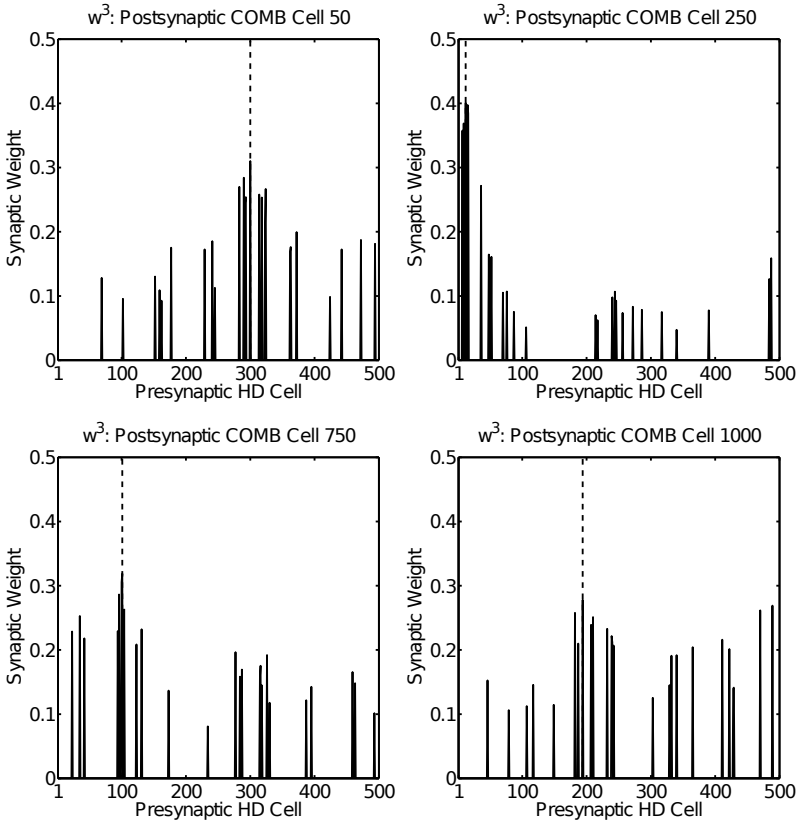

Figure 6.3: The synaptic weights w^3 from the head direction (HD) cell layer to the combination (COMB) cell layer after competitive learning with a time-delayed Hebbian associative learning rule, Equation (6.9), and weight normalization, Equation (6.10). The synaptic weights are taken from a simulation implemented with an axonal conduction delay of 100ms, and a rotational velocity imposed during training of 180°/s. All other network parameters are as given in Table 6.1. Each of the four plots shows the learned synaptic weights from the 500 presynaptic HD cells to a different postsynaptic COMB cell. The presynaptic HD cells are arranged in the plots according to where they fire maximally in the head-direction space of the agent when visual input is available. For each plot, a dashed vertical line indicates the presynaptic HD cell that has maximal w^3 synaptic strength with the postsynaptic COMB cell. Except for the effects of diluted synaptic connectivity, each of the weight profiles is centred on region of similarly tuned HD cells, with a weight distribution that is approximately symmetric about the presynaptic HD cell that has maximal synaptic strength with the postsynaptic COMB cell. Thus, the learned w^3 synaptic weights show that individual COMB cells learn to receive maximal stimulation from particular HD cells. Given that the model parameters ϕ_3, ϕ_4, and the threshold α of the COMB cell sigmoid transfer function are tuned to ensure that a strong rotational velocity cell input through the w^4 synapses is also needed in order to fire the COMB cells, these cells in fact learn to respond to combinations of a particular head direction and clockwise or counter-clockwise rotational velocity.

Figure 6.3 displays the synaptic weights w^3 from the presynaptic head direction cell layer to the postsynaptic combination cell layer after competitive learning with a time-delayed Hebbian associative learning rule, Equation (6.9), and weight normalization, Equation (6.10). Diluted synaptic connectiv-

ity, where individual postsynaptic combination cells receive synapses from only 5% of the presynaptic head direction cells, was implemented to preserve competitive learning in the combination cell layer. The diluted connectivity ensures that individual combination cells learn to respond to a combination of a particular head direction and a rotational velocity. Without diluted connectivity it is possible for individual combination cells to learn to respond to all possible head directions due to the continuity of the head-direction space, and the broadly-tuned overlapping nature of the head direction cell receptive fields (Stringer and Rolls, 2006). With the exception of diluted connectivity, each of the synaptic weight profiles in Figure 6.3 is centred on region of similarly-tuned head direction cells and is approximately symmetric about the presynaptic head direction cell from which the postsynaptic combination cell has maximal w^3 synaptic weight. The symmetric weight distribution demonstrates that individual postsynaptic combination cells have learned to be preferentially stimulated by a subset of presynaptic head direction cells representing nearby head directions. Because the model parameters ϕ_3, ϕ_4, and the threshold α of the combination cell sigmoid transfer function are tuned to ensure that a strong rotational velocity cell input through the w^4 synapses is also necessary to fire individual postsynaptic combination cells, these combination cells can be said to learn to respond to combinations of particular head directions and clockwise or counter-clockwise rotational velocity.

Figure 6.4 displays the synaptic weights w^2 from the presynaptic combination cell layer to the postsynaptic head direciton cell layer after learning with a time-delayed Hebbian associative learning rule, Equation (6.5), and weight normalization, Equation (6.6). In all of the plots, the w^2 synaptic weight profile is asymmetric about the postsynaptic head direction cell with maximal w^3 weight. This asymmetry shows that the presynaptic combination cell has learned to preferentially stimulate a localized

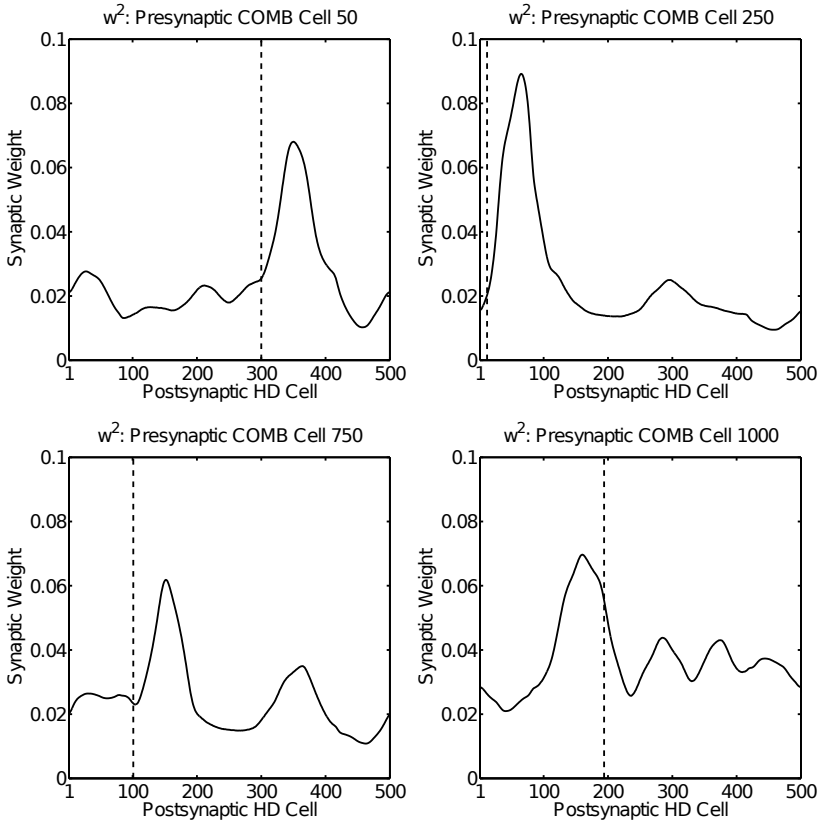

Figure 6.4: The synaptic weights w^2 from the presynaptic combination (COMB) cell layer to the postsynaptic head direction (HD) cell layer after training with a time-delayed Hebbian associative learning rule and weight normalization, Equations (6.5) and (6.6) respectively. These results are taken from a simulation implemented with an axonal conduction delay of 100ms, and a rotational velocity imposed during training of 180°/s. All of the other network parameters are given in Table 6.1. Each of the four plots shows the learned synaptic weights from a different presynaptic COMB cell to the 500 postsynaptic HD cells. The HD cells are arranged in the plots according to where they fire maximally in the head-direction space of the agent when there is visual input available. In each plot a dashed vertical line indicates the postsynaptic HD cell with which the presynaptic COMB cell has maximal w^3 weight as shown in Figure 6.3. It is clear that, for each plot above, the w^2 synaptic weight profile is asymmetric about the postsynaptic HD cell with maximal w^3 synaptic weight, indicating that the presynaptic COMB cell in the plot prefentially stimulates a postsynaptic HD cell representing a different head direction to the presynaptic HD cell from which the postsynaptic COMB cell receives maximal w^3 synaptic weight. This reflects the fact that the packet of HD cell activity will have moved through the head-direction space of the agent in the time it takes for a signal to travel from the HD cell layer through the w^3 synapses to the COMB cell layer, and back again to the HD cell layer through the w^2 synapses. Thus, the axonal conduction delays in the w^2 and w^3 synapses act, through learning, as a timing mechanism that enables the update of the packet of HD cell activity at the same speed as the agent is rotating.

cluster of postsynaptic head direction cells representing a different head direction to the presynaptic head direction cell from which the combination cell receives maximal w^3 stimulation. Thus, the asymmetry reflects the fact that, during training in the presence of visual input, the current head di-

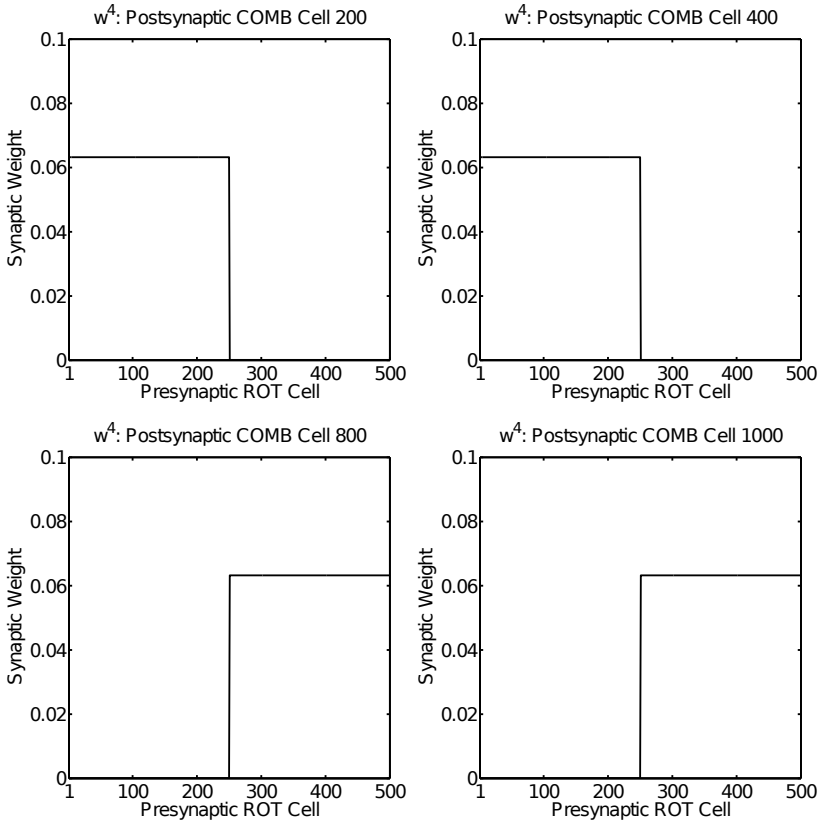

Figure 6.5: The synaptic weights w^4 from the layer of rotational velocity (ROT) cells to the layer of combination (COMB) cells after training with an associative Hebbian learning rule, Equation (6.11), and weight normalization, Equation (6.12). These results are taken from a simulation implemented with an axonal conduction delay of 100ms, and a rotational velocity imposed during training of 180°/s. All of the other network parameters are given in Table 6.1. Each of the four plots shows the learned synaptic weight distribution from the 500 presynaptic ROT cells to a different postsynaptic COMB cell. Each of the postsynaptic combination cells either develops strong w^4 connections from the presynaptic clockwise rotational velocity cells $1-250$, or develops strong w^4 connections from the presynaptic counter-clockwise rotational velocity cells $251-500$.

rection of the agent will have changed in the time it takes for a signal to travel from the head direction cell layer along the w^3 synapses to the combination cell layer, and back to the head direction cell layer through the w^2 synapses. The w^2 and w^3 axonal conduction delays therefore act as a timing mechanism that enable individual combination cells, through learning, to update the packet of head direction cell activity at the correct speed and reflect the changing head direction of the agent.

The plots in Figure 6.5 display the w^4 synaptic weights from the layer of presynaptic rotational veloc-

ity cells to the postsynaptic combination cell layer after training with an associative Hebbian learning rule, Equation (6.11), and weight normalization, Equation (6.12). Because the firing profiles of the presynaptic rotational velocity cells are binary, with each cell signalling that the agent is either rotating in a given direction or is not, the w^4 synaptic weight profiles can be described as a step function. As can be seen in the plots, each postsynaptic combination cell receives positive synaptic weights from the subset of exactly 250 presynaptic rotational velocity cells that signal either clockwise or counter-clockwise rotation. A single postsynaptic combination cell does not receive positive synaptic weights from both of the two rotational velocity cell subsets. Thus, the postsynaptic combination cells have learned to be maximally stimulated by a particular head direction and a particular rotational velocity (clockwise or counter-clockwise).

The firing rates of the head direction, combination, and rotational velocity cells are shown in the plots in Figure 6.6. The top-left plot displays the firing rates of the head direction cells during training in the presence of external visual input. During the time interval 0.0-2.25 seconds, the agent rotated in a clockwise direction. During the time interval 2.25-4.5 seconds, the agent rotated in a counter-clockwise direction. The top-right plot displays the firing rates of the head direction cells recorded during testing in the absence of visual input. During the time interval 0.0-1.0 seconds, there was no firing in the layer of rotational velocity cells (bottom-left plot) and consequently no firing in the layer of combination cells (bottom-right plot). Thus, there was a stable and persistent packet of activity in the head direction cell layer supported by the recurrent w^1 synapses. During the time interval 1.0-2.0 seconds, the 250 rotational velocity cells signalling clockwise rotation of the agent became active, which in turn stimulated an activity packet in the combination cell layer through the w^4 synapses in

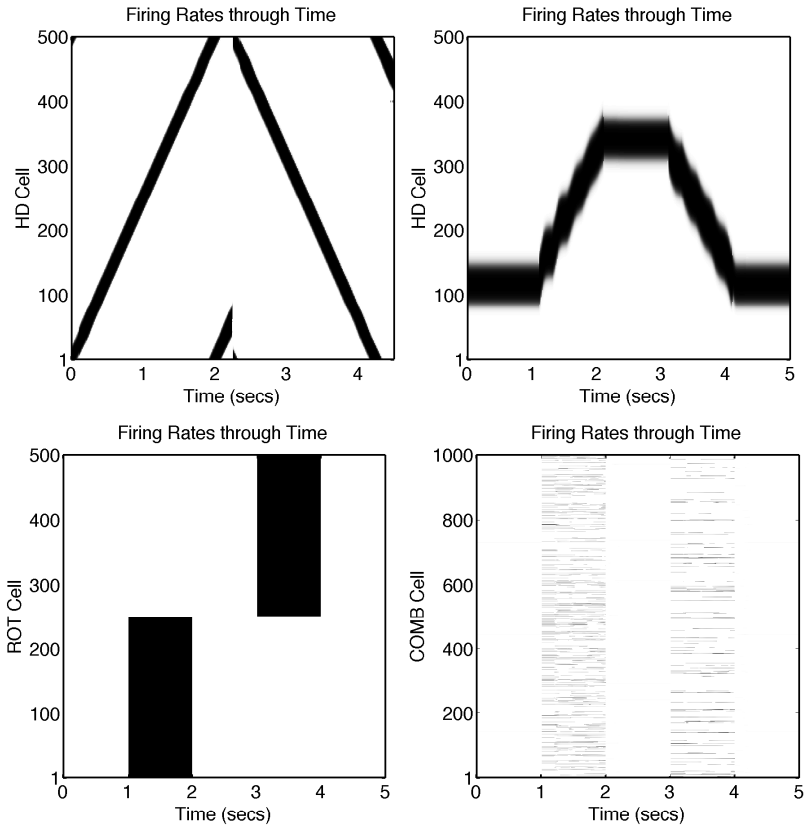

Figure 6.6: The firing rates of the head direction (HD), combination (COMB), and rotational velocity (ROT) cells during training and testing. The results are taken from a simulation implemented with an axonal conduction delay of 100ms, and a rotational velocity imposed during training of 180^{o}/s. All the other network parameters are given in Table 6.1. *Top Left* Firing rates in the layer of 500 HD cells during the 4.5 seconds of training with visual input available. 0.0-2.25 seconds: agent rotation clockwise. 2.25-4.5 seconds: agent rotating counter-clockwise. *Top Right* Firing rates in the layer of HD cells during the 5 seconds of testing in the absence of visual input. During the time interval 1.0-2.0 seconds, the clockwise rotational velocity cells are stimulated, which, in turn, activates the combination cells. The combination cells drive the packet of activity in the HD cell network in the clockwise direction. During the time interval 3.0-4.0 seconds, the counter-clockwise rotational velocity cells are stimulated. This leads to the activity packet in the HD cell network shifting in the counter-clockwise direction. At all other times the packet of HD cell activity remains stationary and persistent due to quiescence in the ROT and COMB cell firing. *Bottom Left* Firing rates in the layer of 500 ROT cells during testing in the absence of visual input. 1.0-2.0 seconds: ROT cells signalling clockwise rotation are active. 3.0-4.0 seconds: ROT cells signalling counter-clockwise rotation are active. *Bottom Right* Firing rates in the layer of 1000 COMB cells during testing in the absence of visual input. 1.0-2.0 seconds: the COMB cells become active due to the firing of the 250 ROT cells signalling clockwise rotation of the agent. 3.0-4.0: the COMB cells become active due to the firing of the 250 ROT cells signalling counter-clockwise rotation of the agent. At all other times the COMB cells are quiescent. In each of the four plots regions of high cell firing are represented by darker shading.

conjunction with the head direction cell input to the combination cell layer through the w^3 synapses.

Due to the axonal conduction delays, Δt, and the asymmetries in the learned weight profiles of the w^2

synapses, the activity packet in the combination cell layer stimulated head direction cells representing head directions further along in the clockwise direction of rotation. Thus, the head direction cell activity packet moved through the head-direction space of the agent, and the model performed path integration of head direction in the clockwise direction.

During the time interval 2.0-3.0 seconds, the rotational velocity cells, and thus the combination cells, were quiescent in their firing and a stable and persistent packet of activity remained in the head direction cell layer. In the time interval 3.0-4.0 seconds, the 250 rotational velocity cells signalling counter-clockwise rotation of the agent became active, and stimulated a packet of activity in the combination cell layer. In an identical mechanism to that for clockwise rotation, the model thus performed velocity path integration of head direction, but this time in the counter-clockwise direction of rotation. During the time interval 4.0-5.0 seconds, the rotational velocity cells and the combination cells ceased firing and there was a stable and persistent packet of activity in the head direction cell layer.

To quantify whether the model could perform path integration of head direction at the same speed during testing as was imposed during training, the speed of update of the head direction cell activity packet was recorded during the testing phase. The measurement was taken according to

$$\text{speed} = \left| \frac{p_2 - p_1}{t_2 - t_1} \right| \tag{6.15}$$

where p_1 and p_2 represent the start and end positions, respectively, of the packet of head direction cell activity. The positions are measured in degrees. t_1 and t_2 represent the time, in seconds, at which the start and end packet positions were recorded. The packet position, p_{n}, is calculated as follows

$$p_{\mathrm{n}} = \frac{\sum_i r_i \theta_i}{\sum_i r_i} \tag{6.16}$$

where r_i is the firing rate of postsynaptic head direction cell i, and θ_i is the preferred head direction for postsynaptic head direction cell i in the presence of visual input. Equation (6.16) effectively gives the centre of mass of the head direction cell activity packet.

The packet positions, p_{n}, were calculated at $t = 1.25$ and $t = 1.75$ seconds during the clockwise rotation of the agent, and at $t = 3.25$ and $t = 3.75$ seconds during the counter-clockwise rotation. The measurements at the start of each period of agent rotation were taken 0.25 seconds after the rotational velocity cells started firing in order to allow time for the signal from the rotational velocity cells to propagate through the w^4 synapses to the combination cell layer, and on to update the head direction cell layer through the w^2 synapses.

To obtain a reliable measure of the learned model performance, five simulations were conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. Table 6.2 displays the mean results of these five simulations. The average speed of clockwise rotation was calculated to be 143.91^{o}/s (S.D. = 2.8^{o}/s). The mean speed of counter-clockwise rotation was calculated to be 132.35^{o}/s (S.D. = 5.7^{o}/s). This is compared to a true speed of 180^{o}/s imposed during training in the presence of visual input. During the period of clockwise rotation, the model updated the packet of head direction cell activity at a mean speed that was 79.95% of the speed imposed during training. For the period of counter-clockwise rotation the model updated the head direction cell activity packet at a speed that was 73.53% of the speed imposed

Table 6.2: The speed of movement of the head direction cell activity packet during testing

100ms Delay; 180°/s Rotational Velocity		
	Clockwise	**Counter-Clockwise**
Mean Speed	143.91°/s	132.35°/s
Standard Deviation	2.8°/s	5.7°/s
Percentage	79.95%	73.53%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. The table displays the mean speed of the head direction cell activity packet across the five simulations, together with the standard deviation, and the mean speed as a percentage of the speed imposed during training in the presence of visual input. With an axonal conduction delay of 100ms and a rotational velocity of 180°/s, the model learns to update the packet of head direction cell activity, in the absence of visual input, at approximately the same speed as was imposed during training in the presence of visual input.

during training. The standard deviations are small enough to conclude that different random synaptic weight initializations and different random synaptic connectivities have almost no effect upon the performance of the model. It can be seen that the speeds of packet update recorded during the testing phase are approximately those speeds experienced by the model during the training phase. The model has learned to achieve this automatically without careful hand-tuning of a parameter specifically governing the speed of update, as was used for instance by Stringer and Rolls (2006). It is noted, however, that the speeds recorded during the testing phase are regularly ($\approx 20\%$) below the speed imposed during the training phase.

100ms Axonal Conduction Delay; 360°/s Rotational Velocity

These simulations were conducted to demonstrate that if the model is trained with the same parameter values, but at twice the rotational velocity, then during testing in the absence of visual input the model will still perform path integration at the speed that was imposed during training. The simulations were

conducted with the network parameters given in Table 6.1.

Figure 6.7 displays the firing rates of the head direction, combination, and rotational velocity cells recorded during training and testing. The conventions are the same as for Figure 6.6. When the model is trained at twice the rotational velocity, the same mechanism of time-delayed synaptic input, due to axonal conduction delays, provides a natural timing interval over which associations can be learned between different head directions of the agent at different points in time. Thus, the model has learned to update a packet of head direction cell activity on the basis of idiothetic input alone.

During the clockwise period of rotation, the model achieved a mean packet update speed of 288.57°/s (S.D. = 20.8°/s). This is shown in Table 6.3. In the period of counter-clockwise rotation, the model achieved a mean packet update speed of 291.02°/s (S.D. = 21.1°/s). The mean speed of clockwise rotation is 80.17% of the speed imposed during training in the presence of visual input. The mean speed of counter-clockwise rotation is 80.84% of the speed imposed during training. The packet update speeds recorded during testing in the absence of visual input are approximately the speeds imposed during training in the presence of visual input. The recorded results demonstrate that the model, implemented with the same network parameters, is capable of learning to perform path integration of head direction at two very different rotational velocities. Similar to the results when the model is trained at 180°/s, it is noted that the model performance regularly undershoots the desired speed by $\approx 20\%$.

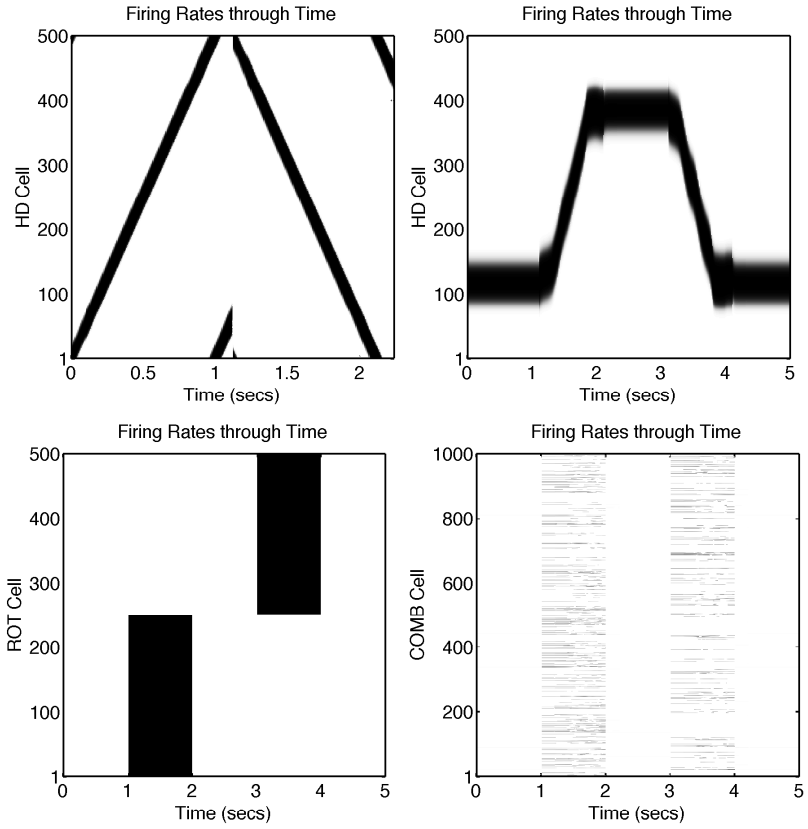

Figure 6.7: The firing rates of the head direction (HD), combination (COMB), and rotational velocity (ROT) cells recorded during training and testing of the model. These results are taken from a simulation implemented with an axonal conduction delay of 100ms, and a rotational velocity of 360°/s imposed during training in the presence of visual input. All other network parameters are given in Table 6.1. The conventions are as for Figure 6.6.

Table 6.3: The speed of movement of the head direction cell activity packet during testing

100ms Axonal Conduction Delay; 360°/s Rotational Velocity		
	Clockwise	**Counter-Clockwise**
Mean Speed	288.57°/s	291.02°/s
Standard Deviation	20.8°/s	21.1°/s
Percentage	80.17%	80.84%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With an axonal conduction delay of 100ms, and a rotational velocity of 360°/s, the model learns to update the packet of head direction activity, in the absence of visual input, at approximately the same speed as was imposed during training in the presence of visual input.

50ms Axonal Conduction Delay; 180^{o}/s Rotational Velocity

The simulations were conducted to demonstrate that the model can learn to perform path integration of head direction when implemented with a different value for the axonal conduction delays. All of the other network parameters are as given in Table 6.1.

Figure 6.8 displays the firing rates of the head direction cells during the 5 seconds of testing in the absence of visual input. The conventions are the same as for the top-right plots in Figures 6.6 and 6.7. A model implemented with an axonal conduction delay of 50ms, and trained at a rotational velocity of 180^{o}/s, can learn to perform path integration of head direction.

The speed of update of the head direction cell activity packet is displayed in Table 6.4. During the clockwise rotation in the absence of visual input, the mean speed of the activity packet update was 132.26^{o}/s (S.D. = 8.7^{o}/s). During the counter-clockwise rotation the mean speed of packet update was 110.45^{o}/s (S.D. = 8.3^{o}/s). The mean speed of clockwise rotation was calculated to be 73.48% of the rotational velocity imposed during training. The mean speed of counter-clockwise rotation is 61.38% of the speed imposed during training.

The packet update speeds recorded during testing in the absence of visual input are approximately the speeds imposed during training in the presence of visual input. Thus, two models implemented with identical model parameters and trained at the same rotational velocity, but with a different values for the axonal conduction delay Δt, can learn to perform path integration at approximately the speed experienced during training. Therefore, using a naturally occurring axonal conduction delay as

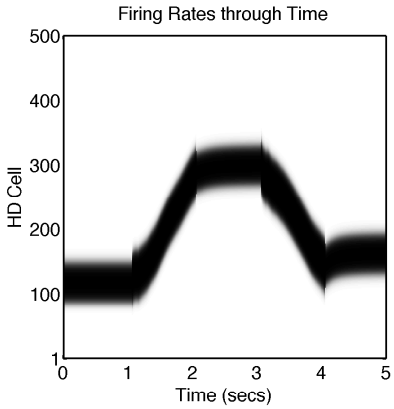

Figure 6.8: The firing rates of the head direction (HD) cells during testing in the absence of visual input. The results are taken from a simulation implemented with an axonal conduction delay of 50ms, and a rotational velocity imposed during training in the presence of visual input of 180°/s. The conventions are as for the top-right plots in Figures 6.6 and 6.7. The model has learned to perform path integration of head direction.

Table 6.4: The speed of movement of the head direction cell activity packet during testing

50ms Axonal Conduction Delay; 180°/s Rotational Velocity		
	Clockwise	**Counter-Clockwise**
Mean Speed	132.26°/s	110.45°/s
Standard Deviation	8.7°/s	8.3°/s
Percentage	73.48%	61.38%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With an axonal conduction delay of 50ms, and a rotational velocity of 180°/s, the model learns to perform path integration at approximately the speed that was imposed during training in the presence of visual input.

a timing mechanism is a general principle that is valid for a range of values of the delay Δt. It should be noted, however, that with a shorter conduction delay Δt, the recorded speeds of packet update undershoot the true rotational velocity, and do so by an amount that is larger than the recorded speeds with a conduction delay of 100ms.

50ms Axonal Conduction Delay; 360^{o}/s Rotational Velocity

Figure 6.9 shows the firing rates of the head direction cells recorded during the 5 seconds of testing in the absence of visual input. The conventions are the same as for the top-right plots of Figures 6.6 and 6.7. A model implemented with an axonal conduction delay of 50ms, and trained at a rotational velocity of 360^{o}/s, can learn to perform path integration of head direction.

For the clockwise rotation in the absence of visual input, the mean speed of the activity packet update was 233.30^{o}/s (S.D. = 6.9^{o}/s). For the counter-clockwise rotation the mean speed of update was 233.29^{o}/s (S.D. = 4.2^{o}/s). For the clockwise rotation, the mean speed was 65.36% of the speed imposed during training. For the counter-clockwise rotation, the mean speed was 64.80% of the speed imposed during training. These results are given in Table 6.5.

The packet update speeds recorded during testing in the absence of visual input are approximately the speeds imposed during training in the presence of visual input. With an axonal conduction delay of 50ms, and a rotational velocity of 360^{o}/s, the recorded speeds of packet update consistently undershoot the true rotational velocity imposed during training, and do so by an amount greater than observed in the previous simulations.

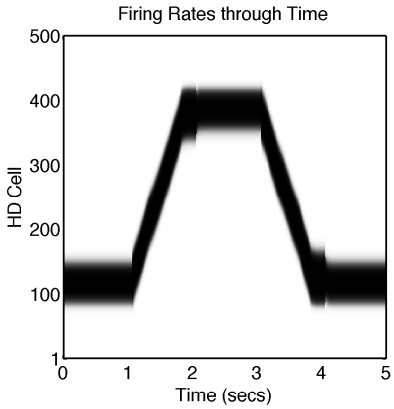

Figure 6.9: The firing rates of the head direction (HD) cells during testing in the dark. The data are recorded from a simulation implemented with an axonal conduction delay of 50ms, and a rotational velocity imposed during training of 360°/s. The model has learned to perform path integration of head direction.

Table 6.5: The speed of movement of the head direction cell activity packet during testing

50ms Axonal Conduction Delay; 360°/s Rotational Velocity		
	Clockwise	**Counter-Clockwise**
Mean Speed	233.30°/s	233.29°/s
Standard Deviation	6.9°	4.2°
Percentage	65.36%	64.80%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic weight connectivities. With an axonal conduction delay of 50ms, and a rotational velocity of 360°/s, the model learns to perform path integration of head direction at approximately the same speed that was imposed during training in the presence of visual input.

These results demonstrate that the same general model, implemented with two different values for the conduction delay Δt, and trained at two different rotational velocities, is capable of learning to perform path integration of head direction at speeds near those experienced during training. Thus, axonal conduction delays can serve as natural timing mechanism over which associations can be learned between neural activity packets occurring at different times, and the value of the delay Δt can be varied within an interval and successful performance can still be achieved.

6.4.2 Experiment 6.2: A Distribution of Axonal Conduction Delays

The simulations comprising this experiment were conducted to demonstrate that the model can perform path integration when the w^2 and w^3 synapses are implemented with a distribution of axonal conduction delays. Upon initialization of the network, the individual synapses w_{ij} in the collection of w^2 and w^3 synapses are randomly assigned a conduction delay within the interval $[1\text{ms}, 100\text{ms}]$ according to a uniform distribution over this interval. The model was trained with a rotational velocity of 360^{o}/s.

Figure 6.10 displays the firing rates of the head direction cells recorded during testing in the dark. It is clear from the plot that a model implemented with a uniform distribution of axonal conduction delays in the interval $[1\text{ms}, 100\text{ms}]$, and trained with a rotational velocity of 360^{o}/s, can learn to perform path integration of head direction.

The speed of update of the head direction cell activity is displayed in Table 6.6. For the clockwise rotation, the mean speed of the head direction cell activity packet update is 215.04^{o}/s (S.D. = 6.5^{o}/s). For the counter-clockwise rotation, the mean speed of packet update is 213.14^{o}/s (S.D. = 9.1^{o}/s).

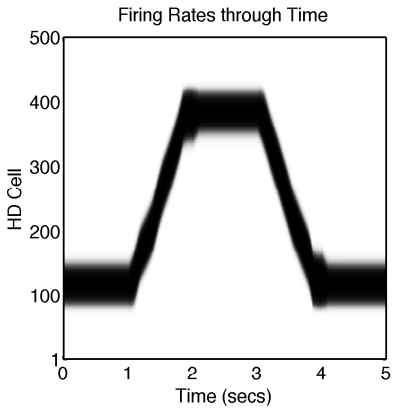

Figure 6.10: The firing rates of the head direction (HD) cells during testing in the absence of visual input in Experiment 6.2. The HD cells are recorded from a simulation implemented with a uniform distribution of axonal conduction delays for w^2 and w^3 synapses within the interval from 1ms to 100ms. The model was trained with a rotational velocity of 360^{o}/s. It is clear from the plot that the model can learn to perform path integration of head direction when simulated with a range of axonal conduction delays.

Table 6.6: The speed of movement of the head direction cell activity packet during testing

Experiment 6.2 Results		
	Clockwise	**Counter-Clockwise**
Mean Speed	215.04^{o}/s	213.14^{o}/s
Standard Deviation	6.5^{o}/s	9.1^{o}/s
Percentage	59.73%	59.20%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. When the model is implemented with a uniform distribution of delays within the interval of 1ms to 100ms, and is trained with a rotational velocity of 360^{o}/s, then the model learns to perform path integration of head direction at approximately the speed imposed during training in the presence of visual input.

When the agent rotates in the clockwise direction during testing, it achieves a mean rotational speed that is 59.73% of the speed imposed during training. For the counter-clockwise rotation, the mean speed is 59.20% of the speed imposed during training.

The model can thus learn to perform path integration of head direction at approximately the speed imposed during training when the model is implemented with a uniform distribution of axonal conduction delays. This is an important result because it demonstrates that the model has biological validity: it is much more likely that there is a distribution of axonal conduction delays in the brain, rather than a single value of the conduction delay that is implemented across all synapses. The result is also important because it shows that the model can function correctly across a wide range of axonal conduction delays, from 1ms to 100ms.

6.4.3 Experiment 6.3: Conduction Delays in One Direction

In the simulations presented so far in this chapter, the models have been implemented with axonal conduction delays in both the w^2 synapses from the presynaptic combination cells to the postsynaptic head direction cells, and in the w^3 synapses from the presynaptic head direction cells to the postsynaptic combination cells. The model should still function correctly if implemented with axonal conduction delays in only one set of the w^2 or w^3 synapses. This experiment explores the results of implementing this model architecture

Axonal Conduction Delays in the w^2 Synapses Only

In this version of the model, there are axonal conduction delays in the w^2 synapses only. Synaptic transmission through the w^3 synapses is now instantaneous with the firing of the presynaptic head

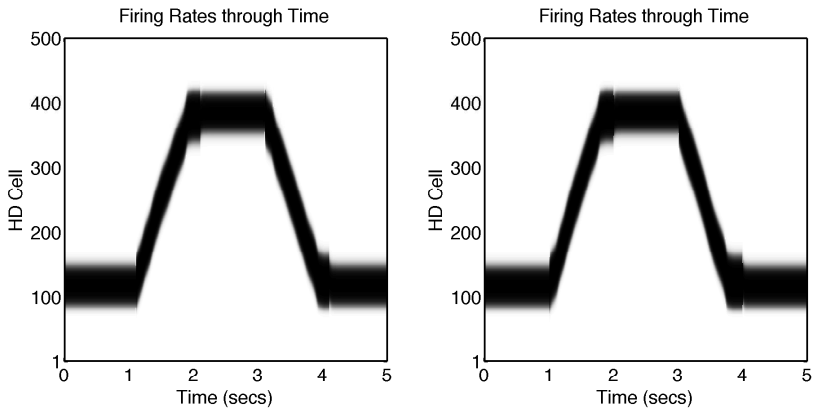

Figure 6.11: Firing rates of the head direction cells from models with axonal conduction delays in one set of the w^2 or w^3 synapses only. The left-hand plot displays the head direction cell firing rates recorded during the testing, in the absence of visual input, of a model simulated with axonal conduction delays in the w^2 synapses only. It is clear that the model can still learn to perform path integration of head direction. The right-hand plot displays the firing rates of head direction cells recorded during the testing, in the absence of visual input, of a model simulated with axonal conduction delays in the w^3 synapses only. It is also clear that the model can still learn to perform path integration of head direction.

direction cells. The model was simulated with an imposed rotational velocity of 360°/s, and an axonal conduction delay of 100ms. The network parameters are given in Table 6.1.

The left-hand plot of Figure 6.11 displays the firing rates of the head direction cells recorded during testing in the absence of visual input. It is evident that a model implemented with axonal conduction delays in the set of w^2 synapses only can learn to perform path integration of head direction when trained with a rotational velocity of 360°/s.

When rotating in a clockwise direction, the mean speed of update of the head direction cell activity packet is 227.52°/s (S.D. = 4.97°/s). The results are given in Table 6.7. When rotating in a counter-clockwise direction, the mean speed of update of the head direction cell activity packet is 227.91°/s (S.D. = 3.56°/s). The mean speed during clockwise rotation is 63.21% of the speed imposed during training in the light. For the counter-clockwise rotation, the mean speed is 63.31% of the speed

Table 6.7: The speed of movement of the head direction cell activity packet during testing

Experiment 6.3 Results: w^2 Delays Only		
	Clockwise	**Counter-Clockwise**
Mean Speed	227.52^{o}/s	227.91^{o}/s
Standard Deviation	4.97^{o}/s	3.56^{o}/s
Percentage	63.21%	63.31%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. When the model is implemented with axonal conduction delays only in the w^2 synapses, then the model can still learn to perform path integration of head direction at approximately the same speed as was experienced during training.

imposed during training.

Axonal Conduction Delays in the w^3 Synapses Only

This version of the model only has axonal conduction delays in the set of w^3 synapses. The model was simulated with the network parameters given in Table 6.1, and trained at a rotational velocity of 360^{o}/s. The axonal conduction delay was set to 100ms.

The right-hand plot of Figure 6.11 displays the firing rates of the head direction cells recorded during testing in the dark. The plot clearly shows that a model implemented with axonal conduction delays only in the w^3 synapses between the presynaptic head direction cells and the postsynaptic combination cells can still learn to perform path integration of head direction.

When rotating clockwise in the absence of visual input, the mean speed of the head direction cell activity packet is 237.23^{o}/s (S.D. = 4.84^{o}/s). For the counter-clockwise rotation, the mean speed is 247.69^{o}/s (S.D. = 28.16^{o}/s). The head direction cell activity packet updates in a clockwise direction at

Table 6.8: The speed of movement of the head direction cell activity packet during testing

Experiment 6.3 Results: w^3 Delays Only		
	Clockwise	**Counter-Clockwise**
Mean Speed	237.23^{o}/s	247.69^{o}/s
Standard Deviation	4.84^{o}/s	28.16^{o}/s
Percentage	65.90%	68.80%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. When the model is implemented with axonal conduction delays present only in the w^3 synapses then the model can still learn to perform path integration of head direction at approximately the same speed as was experienced during training.

a speed that is 65.90% of the speed that is imposed during training. When rotating counter-clockwise, the head direction cell activity packet updates at a speed that is 68.80% of the speed imposed during training. These results are given in Table 6.8.

The results of this experiment demonstrate that the model can still function correctly when implemented with axonal conduction delays in only one set of the w^2 or w^3 synapses, but not both sets. There is, however, a slight reduction in model performance in comparison to the model simulated with axonal conduction delays in both sets of synapses. The model is able to learn the correct behaviour with a single set of axonal conduction delays because those delays still provide a natural time interval over which associations between different head directions can be learned by the model. Thus, upon replay in the absence of visual input, path integration of head direction is performed.

6.4.4 Experiment 6.4: The Effect of the Learning Rate

The learning rate, k, scales the amount by which a particular synapse w_{ij} will change at any particular learning update. In the experiments reported so far in this chapter, the learning rate, k, has been set

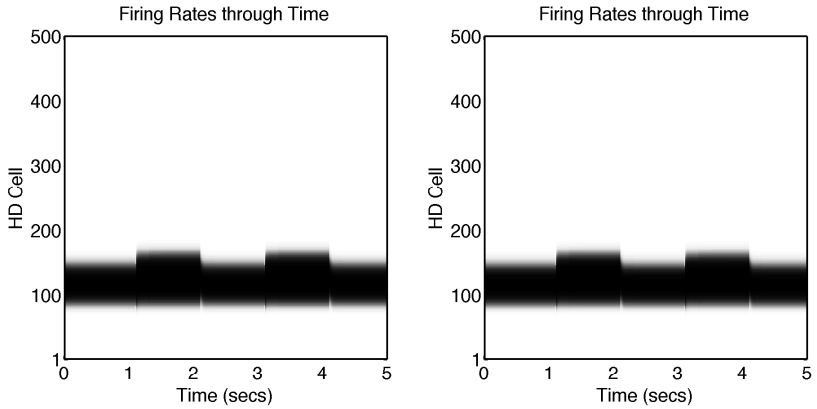

Figure 6.12: The firing rates of the head direction cells recorded during testing in the dark in Experiment 6.4. The left-hand plot displays data recorded from a model simulated with the learning rates k^2 and k^3 set to 0.01. The right-hand plot displays data from a simulation with the learnng rates k^2 and k^3 set to 0.001. It is clear from both plots that with the learning rates k^2 and k^3 set to lower values than the standard 0.1, the model cannot, in either case, learn to perform path integration of head direction.

to a constant value of 0.1. This current experiment will investigate the effect that altering the learning rate has upon the learned model performance.

Learning Rates k^2 and k^3 of 0.01

In the simulation comprising this part of the experiment, the learning rates k^2 and k^3, governing the update of the w^2 and w^3 synapses respectively, were set to a value of 0.01. The learning rates k^1 and k^4 remained set at the previous value of 0.1. The network was simulated with an imposed rotational velocity of 360°/s during training in the light, and axonal conduction delays of 100ms. The network parameters are given in Table 6.1.

The left-hand plot of Figure 6.12 displays the firing rates of the head direction cells during the testing phase in the dark. It is clear from the plot that the model cannot learn to perform path integration of head direction.

Further simulations were conducted with all of the learning rates, k^1 to k^4, set to 0.01. The results are not shown, but are qualitatively very similar to the results of the simulations with only k^2 and k^3 set to 0.01.

Learning Rates k^2 and k^3 set to 0.001

In this part of the experiment, the model is simulated with learning rates k^2 and k^3 set to 0.001. All of the other model parameters are as given in Table 6.1. The model was trained with a rotational velocity of 360^{o}/s, and the axonal conduction delays were set to 100ms.

The right-hand plot of Figure 6.12 displays the firing rates of the head direction cells recorded during the testing phase in the absence of visual input. The plot shows that the model cannot learn to perform the path integration of head direction.

The model was also simulated with all of the learning rates, k^1 to k^4, set to 0.001. The results were qualitatively very similar to results when the model is simulated with only learning rates k^2 and k^3 set to 0.001.

Both parts of this experiment have demonstrated that the model must be implemented with relatively high values of 0.1 for the learning rates k^2 and k^3. Otherwise, the model cannot learn to perform path integration of head direction. With a learning rate value of 0.01 or 0.001, the strengths of the synapses in the model do not update at a fast enough rate to produce the correct, time-delayed, associations between head directions required for successful path integration of head direction.

6.4.5 Experiment 6.5: Long-Term Depression in the Learning Rules

The simulations reported so far in this chapter have implemented standard Hebbian associative learning rules. Long-term depression (LTD) can increase the specificity of cell responses by permitting a decrease, as well as increase, in the strength of a particular synapse w_{ij}. The simulations in this experiment investigate the effect that homosynaptic LTD and heterosynaptic LTD have upon the learned model performance.

Homosynaptic LTD

Homosynaptic LTD promotes a decrease in the strength of a synapse w_{ij} when strong firing of the presynaptic cell j is temporally conjunctive with below-average firing of the postsynaptic cell i.

Equation (6.5) now becomes

$$\frac{dw_{ij}^2(t)}{dt} = k^2 \left[r_i^{\mathrm{HD}}(t) - \langle r_i^{\mathrm{HD}} \rangle \right] r_j^{\mathrm{COMB}}(t - \Delta t) \tag{6.17}$$

where k^2 is the learning rate, $r_i^{\mathrm{HD}}(t)$ is the firing rate of postsynaptic head direction cell i at time t, $\langle r_i^{\mathrm{HD}} \rangle$ is the time-average firing rate of postsynaptic head direction cell i, and $r_j^{\mathrm{COMB}}(t - \Delta t)$ is the firing rate of presynaptic combination cell j at a time $t - \Delta t$ in the past. The average firing rate $\langle r_i^{\mathrm{HD}} \rangle$ is calculated by summing the firing rates r_i of head direction cell i over successive model updates, and then dividing by the number of model updates so far.

Equation (6.9) now becomes

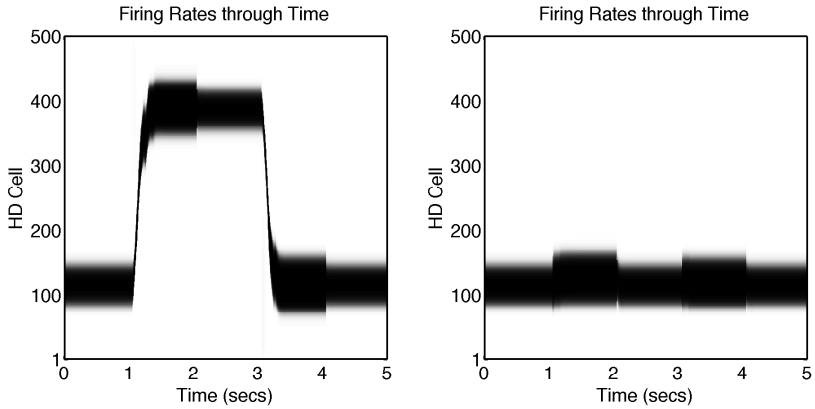

Figure 6.13: The firing rates of the head direction cells during testing of the model in Experiment 6.5. The left-hand plot displays data from the model simulated with homosynaptic LTD, Equations (7.4) and (7.5), now present in the model. It is clear that the model can still learn to perform the path integration of head direction. The right-hand plot displays data from a simulation with heterosynaptic LTD, Equations (7.6) and (7.7), in the model. The model cannot learn to perform path integration of head direction.

$$\frac{dw_{ij}^{3}(t)}{dt} = k^{3}\left[r_{i}^{\mathrm{COMB}}(t) - \langle r_{i}^{\mathrm{COMB}}\rangle\right] r_{j}^{\mathrm{HD}}(t-\Delta t) \tag{6.18}$$

where k^3 is the learning rate, $r_i^{\mathrm{COMB}}(t)$ is the firing rate of postsynaptic combination cell i at time t, $\langle r_i^{\mathrm{COMB}}\rangle$ is the time-average firing rate of postsynaptic combination cell i, and $r_j^{\mathrm{HD}}(t-\Delta t)$ is the firing rate of presynaptic head direction cell j at a time $t-\Delta t$ in the past. The average firing rate $\langle r_i^{\mathrm{COMB}}\rangle$ is calculated in the same manner as the average firing rate $\langle r_i^{\mathrm{HD}}\rangle$ described above.

Simulations were conducted with the network parameters given in Table 6.1. The network was trained at a rotational velocity of 360^{o}/s in the presence of visual input. The axonal conduction delays were set to 100ms.

The left-hand plot of Figure 6.13 displays the firing rates of the head direction cells recorded during the testing of the model in the dark. The plot clearly demonstrates that with homosynaptic LTD

Table 6.9: The speed of movement of the head direction cell activity packet during testing

Experiment 6.5 Results: Homosynaptic LTD		
	Clockwise	**Counter-Clockwise**
Mean Speed	64.22^{o}/s	31.34^{o}/s
Standard Deviation	4.19^{o}/s	6.28^{o}/s
Percentage	17.84%	8.71%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. A model simulated with Homosynaptic LTD, Equations (7.4) and (7.5), in the learning rules can learn to peform path integration of head direction although the learned performance is somewhat degraded in comparison to the model simulated without Homosynaptic LTD (Experiment 6.1).

present in the learning rules, the model can still learn to perform path integration of head direction.

The speed of update of the head direction cell activity packet is given in Table 6.9. During the period of clockwise rotation, the head direction cell activity packet updates at an average speed of 64.22^{o}/s (S.D. = 4.19^{o}/s), which is 17.84% of the speed imposed during training in the light. For the period of counter-clockwise rotation, the head direction cell activity packet updates at an average speed of 31.34^{o}/s (S.D. = 6.28^{o}/s), which is 8.71% of the speed imposed during training. Thus, the model simulated with homosynaptic LTD can learn to perform path integration, but the speed of update of the activity packet falls quite short of the target speed.

This is an interesting result. Qualitatively, as depicted in the left-hand plot of Figure 6.13, the learned behaviour of the model simulated with homosynaptic LTD is very similar to the model simulated without homosynaptic LTD (Experiment 6.1). Quantitatively, however, there appears to be a big difference in the trained performance of the two models. This may be due to the fact that the speed of update

of the head direction cell activity packet has been measured over 0.5s during the period of clockwise rotation, and 0.5s during the period of counter-clockwise rotation. In the left-hand plot of Figure 6.13, the head direction cell activity packet is stationary for most of the 0.5s recording period during both clockwise and counter-clockwise rotations. This may bias the recording of the speed of the activity packet update such that the average speed appears slower than it really is.

The model was also simulated with homosynaptic LTD in the learning rules for all of the sets of synapses, w^1 to w^4. The results are not shown, but are qualitatively very similar to the results already reported for this part of Experiment 6.5.

Heterosynaptic LTD

Heterosynaptic LTD promotes a decrease in the strength of a particular synapse w_{ij} when below-average firing of the presynaptic cell j is temporally conjunctive with strong firing of the postsynaptic cell i.

Equation (6.5) now becomes

$$\frac{dw_{ij}^2(t)}{dt} = k^2 r_i^{\mathrm{HD}}(t) \left[r_j^{\mathrm{COMB}}(t - \Delta t) - \langle r_j^{\mathrm{COMB}} \rangle \right] \tag{6.19}$$

where k^2 is the learning rate, $r_i^{\mathrm{HD}}(t)$ is the firing rate of postsynaptic head direction i at time t, $r_j^{\mathrm{COMB}}(t - \Delta t)$ is the firing rate of presynaptic combination cell j at a time $t - \Delta t$ in the past, and $\langle r_j^{\mathrm{COMB}} \rangle$ is the time-average of the firing rate of presynaptic combination cell j. The average firing rate $\langle r_j^{\mathrm{COMB}} \rangle$ is calculated by summing the firing rates r_j over successive model updates, and then

dividing by the total number of model updates so far.

Equation (6.9) now becomes

$$\frac{dw_{ij}^{3}(t)}{dt} = k^{3} r_{i}^{\mathrm{COMB}}(t) \left[r_{j}^{\mathrm{HD}}(t - \Delta t) - \langle r_{j}^{\mathrm{HD}} \rangle \right] \tag{6.20}$$

where k^3 is the learning rate, $r_i^{\mathrm{COMB}}(t)$ is the firing rate of postsynaptic combination cell i at time t, $r_j^{\mathrm{HD}}(t - \Delta t)$ is the firing rate of presynaptic head direction cell j at a time $t - \Delta t$ in the past, and $\langle r_j^{\mathrm{HD}} \rangle$ is the time-average firing rate of presynaptic head direction j, which is calculated in the same way as the average firing rate$\langle r_j^{\mathrm{COMB}} \rangle$ described above.

The model was simulated with the network parameters given in Table 6.1, and with a rotational velocity of 360^{o}/s imposed during training. The axonal conduction delays were set to 100ms.

The right-hand plot of Figure 6.13 displays the firing rates of the head direction cells during testing. The plot demonstrates that a model incorporating heterosynaptic LTD into its learning rules cannot learn to perform path integration of head direction.

Further simulations, which incorporate heterosynaptic LTD in the learning rules for each set of synapses, w^1 to w^4, produced results that are qualitatively very similar to the results from the simulations where heterosynaptic LTD is only incorporated in the learning rule for the w^2 and w^3 synapses.

In summary, homosynaptic LTD helps to increase the specificity of the learned cell responses, and

thus the model can still perform path integration of head direction when homosynaptic LTD is incorporated into the learning rules. The accuracy of the path integration, however, is much reduced, In contrast, the model with heterosynaptic LTD is not capable of learning to perform path integration of head direction.

6.4.6 Experiment 6.6: Learning during the Testing Phase

The experiments reported previously in this chapter are all implemented so that, during the testing of the model in the absence of visual input, no learning takes place. This protocol was adopted to facilitate analysis of the learned model performance: as the model synapses are not permitted to change strength during the testing phase, all observed behaviour during this phase should be the result of a combination of the learning that has taken place and the manipulation of the model and the environment during testing. In the real brain, it is doubtful that learning can be turned off and on at arbitrary moments in time. The simulations comprising this experiment were therefore conducted to examine the effect of permitting learning to occur during testing of the model.

During the testing phase of the model, all learning and weight normalization equations for the sets of synapses, w^1 to w^4, were simulated in full. The network parameters are given in Table 6.1. The axonal conduction delay was 100ms, and the rotational velocity during training was 360°/s.

Figure 6.14 displays the firing rates of the head direction cells recorded during the testing phase in the absence of visual input. It is evident that, with learning permitted during the testing phase, the model is incapable of performing path integration of head direction.

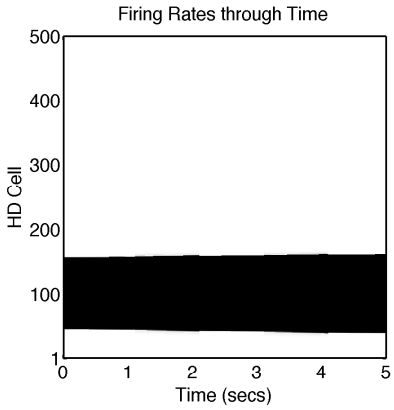

Figure 6.14: The firing rates of the head direction cells recorded during the testing phase of Experiment 6.6. When learning is permitted to occur during the testing phase, the model is incapable of performing path integration of head direction.

Permitting learning in the testing phase clearly abolishes the ability of the model to perform path integration of head direction. The training phase in this experiment was identical to the training phase in all of the simulations reported previously in this chapter. Moreover, the model was simulated with the same parameter values as the models in Experiment 6.1. Thus, the model in the current experiment will have learned the synaptic structure required to perform path integration of head direction in the absence of visual input. The further learning appears to overwrite this synaptic structure. In the real brain, it may be that the neural equivalent of the learning rate decreases through time, which would prevent previously learned associations from being overwritten.

6.5 Discussion

This chapter provides the implementation details, model equations, and simulation results for a neural network model that can learn to perform path integration of head direction at approximately the speed imposed during training. In summary, the main points of this chapter are:

- The model architecture comprises two reciprocally-linked layers of cells: the head direction

cells and the combination cells. There is also a layer of rotational velocity cells.

- The model must learn to perform path integration of head direction. The model is trained rotating at a prespecified velocity with visual input available. If learning is successful, the model will be able to update a packet of head direction cell activity during testing in the absence of visual input. The speed of update of this activity packet should match the rotational speed imposed during the training phase.

- Axonal conduction delays are incorporated into two of the sets of synapses. These conduction delays have the effect that, for a particular synapse w_{ij}, the firing of a presynaptic cell j will arrive at a postsynaptic cell i after a delay Δt. The value of this delay Δt is the length of the axonal conduction delay. The model is thus able to learn associations, through time, between a particular head direction θ_1 at time t_1, a rotational velocity, and a particular head direction θ_2 at time t_2. In the absence of visual input, the model will be able to use these learned associations to perform path integration of head direction at the speed that was experience during training.

- In simulation, the model is able to learn to perform path integration of head direction. Moreover, the model can learn path integration at approximately the same speed as was imposed during training in the presence of visual input. The model can solve this problem when implemented with different values of the axonal conduction delays, and when trained at different rotational speeds.

- Importantly, the model still learns the correct behaviour when implemented with a distribution of axonal conduction delays. Thus, the model behaviour is not dependent upon one particular value of the axonal conduction delays.

- The model also learns to perform correctly when implemented with axonal conduction delays in either the w^2 or w^3 synapses, but not both.

- The model requires that the learning rates, k^2 and k^3, be set to a relatively high value of 0.1. Otherwise, the model cannot learn to perform path integration,

- The model can still learn to perform path integration, albeit less accurately, when homosynaptic LTD is incorporated in the learning rules. When heterosynaptic LTD is incorporated, the model performance collapses and path integration cannot be performed.

- Permitting learning to occur during the testing phase has a deleterious effect upon model performance. In this case, path integration is destroyed.

The results of the experiments reported in this chapter clearly demonstrate that using axonal conduction delays as a natural timing interval provides a viable computational mechanism for the model to learn path integration of head direction at approximately the correct speed. Moreover, the model reported in this chapter is the first model to address how a neural network may learn, in a biologically plausible manner, to the path integration integration of head direction at the correct speed.

It is noted, however, that the model does not learn to perform path integration at *exactly* the speed imposed during training. Moreover, there is always a consistent undershoot in the rotational velocity learned by the model compared to the rotational velocity imposed during training. One way to increase the accuracy of the learned model performance may be to simulate the cells as integrate-and-fire cells. It has been shown that head direction cells, simulated as integrate-and-fire cells, are capable of responding to a stimulus within tens of milliseconds (Degris et al., 2004). If the current model

were to be implemented with integrate-and-fire cells, the rapid responses of the cells may increase the accuracy of the speed of path integration.

Another factor in the slower path integration speed of the current model may be the w_{ij}^{1} excitatory recurrent collateral synapses between the head direction cells. As discussed in Chapter 5, recurrent collateral synapses are traditionally employed to help form a continuous attractor network. These synapses may, however, act as a source of friction that damps the speed of update of the head direction cell activity packet. Some neural network models of the head direction cell system have shown that it is possible to produce a continuous attractor network without recurrent collateral synapses (Rubin et al., 2001; Song and Wang, 2003, 2005; Boucheny et al., 2005). It would be possible to implement the model reported in this chapter without recurrent collateral synapses, and this architectural change may improve the learned performance of the model.

Without further simulation results for comparison, it is not exactly clear why, in the testing phase of the simulation, the model always undershoots the rotational velocity imposed during training. That is, there is no immediately obvious computational reason why the model under-performs. As stated above, the absence of w^{1} excitatory recurrent collateral synapses may improve model performance, as may faster integrate-and-fire cell spiking dynamics, or a combination of both. Future simulations will help to elucidate the influence of each of these factors upon the accuracy of the path integration mechanism.

Given that the aim of the research presented in this chapter was to simulate a neural network model

that can learn to perform path integration of head direction at the correct speed, and the current model exhibits a consistent undershoot of the correct speed, a valid question is whether or not this undershoot affects the validity of the model. The undershoot of the speed of path integration, although problematic, is not a fatal flaw in the current model. Theoretically, there is nothing to indicate that the current model is computationally incapable of learning to perform path integration at the speed imposed during training. The current model is not completely accurate when it performs path integration of head direction in the absence of visual input, but it is the first model to address how this may be learned in a biologically plausible manner. Moreover, it is likely that further simulations incorporating the architectural changes and cell spiking dynamics outlined above will improve upon the accuracy of the current model. Thus, the current model is valid as a first step in addressing how a neural network model may learn to perform path integration of head direction at the correct speed.

The current model cannot, without modification, generalize across different speeds of path integration. Currently, a different synaptic structure has to be learned for each speed of rotation. There are cells within the rat brain that respond proportionally to the speed of rotation of the rat (Bassett and Taube, 2001b). Incorporating this class of cell in place of the current combination cells will allow the model to generalize across different speeds of path integration.

With typical conduction velocities of unmyelinated axons of 0.1m/s (Girard et al., 2001), the standard axonal conduction delays of 50ms and 100ms that are employed in this chapter would correspond to axonal lengths of 5mm and 10mm respectively. This implies that the current network contains axons that are too long to be biologically plausible for a model of the head direction cell system. Simulating

the model with smaller uniform conduction delays of say 10ms produced a significant degradation in the accuracy of the path integration, although as reported in Section 6.4.2 the model can still perform path integration when implemented with a distribution of conduction delays. The best simulation results were for models implemented with uniform conduction delays of either 50ms or 100ms, and so these are the results reported in this chapter. Theoretically, the model should still function correctly when implemented with any length of conduction delay, and improving the accuracy of the path integration could facilitate this. It should also be noted that a conduction delay of say 50ms does not necessarily imply a single axon of 5mm length. Head direction cells are found in a number of brain areas (Sharp, 2005), and the 50ms delay could correspond to a polysynaptic path across multiple head direction cells and through multiple head direction cell areas, which would make the 5mm axonal length a total length across this polysynaptic path. In principle, the computational mechanism of using axonal conduction delays to provide a natural timing mechanism should still function when implemented in a polysynaptic path.

In conclusion, this chapter presents a neural network model that learns to perform path integration of head direction at approximately the speed imposed during training. A controlled exploration of the effect upon model performance of altering the model parameters was carried out. Axonal conduction delays are just one possible timing mechanism over which associations between head directions can be learned through time. Another possible mechanism, which will be explored in detail in the next chapter, is to implement cells that have different values for their neuronal time constants.

Chapter 7

Path Integration of Head Direction: Neuronal Time Constants

This chapter presents the operational principles and simulation results of a second model that can learn to perform path integration of head direction at the speed of rotation imposed during training. The model uses different values of the neuronal time constants τ^{HD} and τ^{COMB} to produce effective time delays over which associations can be learned between packets of neural activity. Through careful replication of the experiments reported in Chapter 6, the similarities and differences between the current model and the model incorporating axonal conduction delays are explored.

7.1 Model Architecture

The model architecture is presented in Figure 7.1. The model contains a layer of head direction cells that is reciprocally connected to a layer of combination cells. There is also a layer of rotational velocity cells that project, in a feedforward manner, to the the layer of combination cells.

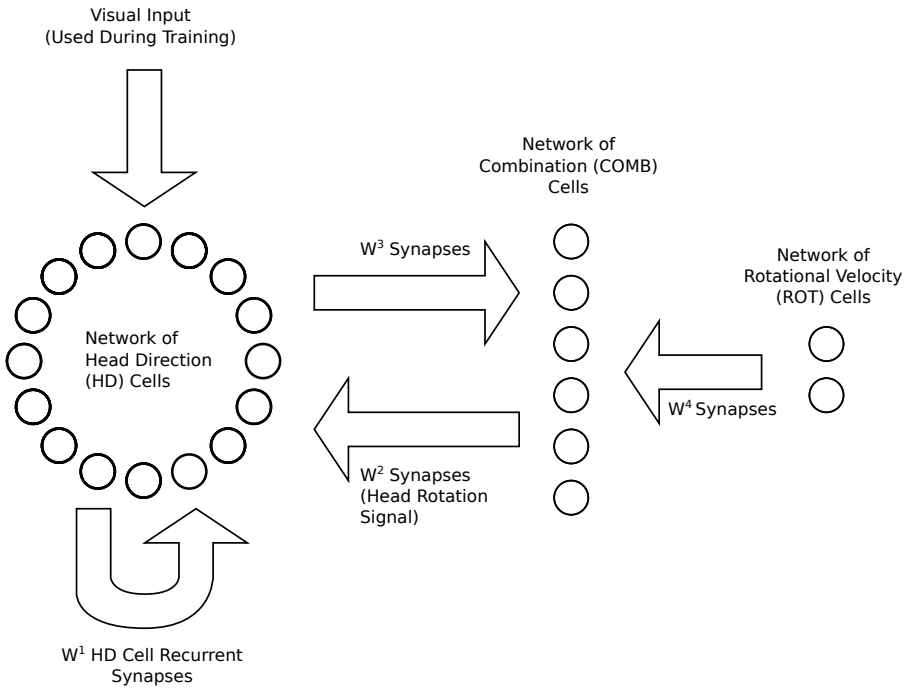

Figure 7.1: The neural network architecture for a model that learns to perform path integration of head direction at the speed imposed during training. The values of the neuronal time constants τ^{HD} and τ^{COMB} provide a timing mechanism over which associations between activity packets in the head direction cell network and the combination cell network may be learned through the w^2_{ij} and w^3_{ij} synapses. The w^1_{ij} recurrent synapses help to produce a continuous attractor network in the layer of head direction cells and thus support a stable packet of head direction cell activity when the agent is not rotating. The w^4_{ij} synapses provide inputs from the layer of rotational velocity cells signalling the current direction of rotation of the agent. All of the synapses are Hebb-modifiable except for the visual inputs to the head direction cell layer, which are only present during the training phase of the simulation.

The architecture of the current model, incorporating neuronal time constants, is very similar to the architecture of the model incorporating axonal conduction delays reported in Chapter 6 (Figure 6.1). The difference between the architectures of the two models is that the current model does not incorporate axonal conduction delays in any of the w^1 to w^4 synapses. That is, in the current model the transmission of a signal from presynaptic cell j to postsynaptic cell i is always instantaneous with the firing $r_j(t)$ of the presynaptic cell.

The model architecture is proposed as the minimal synaptic architecture that is capable of updat-

ing a packet of head direction cell activity, in the absence of visual input, at the same speed as was imposed during training in the light. The model is not intended to directly correspond to any particular brain region known to contain head direction cells.

7.2 Operational Principles of the Model

Under certain values of the presynaptic cell and postsynaptic cell neuronal time constants, when the presynaptic cell is driving the postsynaptic cell, the activity profile of the postsynaptic cell will be slightly delayed in time from the activity profile of the presynaptic cell. This produces an effective time delay over which associations between presynaptic and postsynaptic neural activity can be learned.

A useful measure to consider is the time delay between the centres of mass of the presynaptic cell and postsynaptic cell temporal activity profiles. In particular, if the time constant governing the temporal evolution of the postsynaptic activity is reasonably large, the centre of mass of the postsynaptic activity profile will be delayed by a small fixed time interval Δt after the centre of mass of the presynaptic activity profile.

Although a small delay Δt exists between the presynaptic and postsynaptic activity centres of mass, the synaptic connection from the presynaptic cell to the postsynaptic cell will still be strengthened by a firing-rate-based Hebbian associative learning rule. Hebbian learning rules in general will strengthen the connection between two cells in proportion to the product of their current instantaneous firing rates. Thus, this type of learning rule will strengthen the connection between a presynaptic cell, and a

postsynaptic cell whose activity is delayed by Δt from that of the presynaptic cell, provided that the activity profiles of the two cells still partially overlap through time. The network thus learns associations between neural activity in the head direction cell and combination cell layers over these specific fixed time intervals Δt. The result of this is that the network will learn to shift a packet of activity through the head direction cell layer at the correct speed.

7.2.1 Analysis of a Two-Cell System

To explore the operation of the model it is helpful to consider a simple two-cell system. The activation h_i of the postsynaptic cell i is governed by

$$\tau \frac{\mathrm{d}h_i(t)}{\mathrm{d}t} = -h_i(t) + w_{ij}(t) r_j(t) \tag{7.1}$$

where τ is the neuronal time constant, $r_j(t)$ is the firing rate of the presynaptic cell j at time t, and w_{ij} is the value of the synaptic weight from the presynaptic cell j to the postsynaptic cell i at time t. The firing rate of the postsynaptic cell i at time t is given by the sigmoid transfer function

$$r_i(t) = \frac{1}{1 + \mathrm{e}^{-2\beta(h_i(t) - \alpha)}} \tag{7.2}$$

where α is the sigmoid threshold and β is the slope. Although the presynaptic cell j and the postsynaptic cell i may fire together, a relatively large time constant, τ, say 50-100ms, will ensure that the centre of mass of the postsynaptic cell temporal activity profile is delayed by a small time interval Δt from the centre of mass of the presynaptic cell temporal activity profile.

During learning in the two-cell system the synaptic weight w_{ij} from the presynaptic cell j to the

postsynaptic cell i is strengthened according to the following associative Hebbian learning rule

$$\frac{dw_{ij}(t)}{dt} = kr_i(t)r_j(t) \tag{7.3}$$

where k is the learning rate, $r_i(t)$ is the firing of the postsynaptic cell at time t, and $r_j(t)$ is the firing of the presynaptic cell at time t.

This learning rule will strengthen the synaptic weight between the presynaptic and postsynaptic cells during the brief periods in which both of these cells have high firing rates. However, if the postsynaptic cell neuronal time constant is reasonably large, the simple two-cell system learns to associate a temporal activity profile in the presynaptic cell with a slightly delayed temporal activity profile in the postsynaptic cell. In other words, the centre of mass of the postsynaptic cell temporal activity profile will always be delayed Δt from the centre of mass of the presynaptic cell temporal activity profile. This will hold true both during and after learning.

A consequence of this mode of operation is that the presynaptic cell j learns to drive activity in the postsynaptic cell i with the same effective time delay Δt that was present during learning. Moreover, a longer postsynaptic time constant τ ensures that the presynaptic cell j learns to stimulate the postsynaptic cell i a greater time interval Δt after the presynaptic cell firing. It is this delay Δt between the presynaptic cell and postsynaptic cell firing that allows the correct time intervals between cell activities to be learned and relayed by the cells in the model. This is essential for the model to be able to replay the temporal sequence of cell activity at the speed that was experienced during learning.

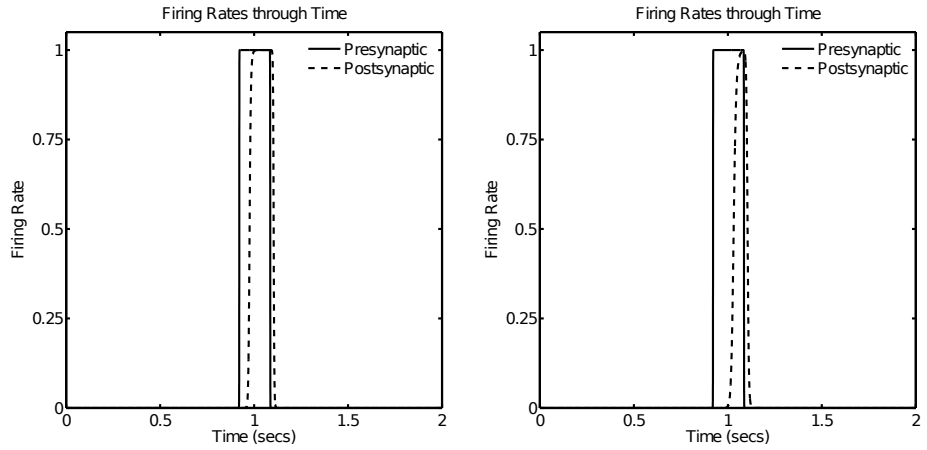

Figure 7.2: The firing rates through time of two simulations of the simple two-cell model. The presynaptic cell firing rate is indicated by the solid line, and the postsynaptic cell firing rate is indicated by the dashed line. The left-hand plot displays data recorded from a simulation in which the time constant τ of the governing equation for the postsynaptic cell activation was set to 50ms. As can be seen, there is a delay between the centres of mass of the temporal firing profiles of the presynaptic cell and the postsynaptic cell. The right-hand plot displays data recorded from a simulation in which the time constant τ of the governing equation for postsynaptic cell activation was set to 100ms. There is a clear delay between the centres of mass of the temporal firing profiles of the presynaptic and postsynaptic cells. This delay is more pronounced than in the left-hand plot, which shows that a longer time constant τ will produce a longer delay Δt between the presynaptic cell and postsynaptic cell activity profiles.

To further illustrate the effect of the neuronal time constant upon the delay Δt between the presynaptic cell and postsynaptic cell firing, the two-cell system was simulated according to the cell activation equation and sigmoid transfer function given above in this section, Equations (7.1) and (7.2) respectively, with a non-modifiable synaptic weight $w_{ij} = 1$ between the presynaptic and postsynaptic cells. Figure 7.2 displays the presynaptic and postsynaptic cell firing rates recorded through time for two simulations of the two-cell system. In the left plot the neuronal time constant τ for the postsynaptic cell was set to 50ms. There is a clear delay between the temporal firing profiles of the presynaptic cell and the postsynaptic cell. In the right plot the neuronal time constant τ was set to 100ms. The delay between the presynaptic cell and postsynaptic cell firing is more pronounced than in the left plot, which illustrates that a longer time constant τ will produce a longer delay Δt between the presynaptic and postsynaptic cell temporal activity profiles.

Table 7.1: Simulation results for the simple two-cell model

Results from the Two-Cell System	
τ (ms)	Δt (ms)
50	36.8
60	43.2
70	49.2
80	54.7
90	59.7
100	64.2

For each of the six simulations, the neuronal time constant τ was varied and the effective time delay between the centres of mass of the presynaptic cell and postsynaptic cell temporal firing profiles was calculated. An increase of 10ms in the time constant τ produces an increase of approximately 5-6ms in the time delay Δt between the centres of mass of the presynaptic cell and postsynaptic cell temporal firing profiles.

To quantify the mode of operation, the simple two-cell model was simulated six times: each unique simulation was conducted with a different value of the time constant τ. For each simulation, the firing rates of the presynaptic and postsynaptic cells were recorded through time and then the effective time delay Δt between the centres of mass of the presynaptic cell and postsynaptic cell firing profiles was calculated. Table 7.1 displays the results from the six simulations. Each increase of 10ms in the value of the time constant τ leads to an approximate increase of 5-6ms in the value of the time delay Δt. These results show that the length of the time delay Δt is approximately proportional to the value of the neuronal time constant τ. The time constant can thus serve as a reliable timing mechanism over which associations through time can be made between presynaptic cell and postsynaptic cell activities.

7.2.2 Analysis of the Two-Layer Model

The analysis of the simple two-cell system can be expanded to include the full version of the model given in Figure 7.1. When the neuronal time constant τ^{COMB} is set to a relatively large value, for instance 100ms or 150ms, and the neuronal time constant τ^{HD} is set to a relatively small value of 1ms, an effective time delay Δt is introduced into the w^3 synaptic connections from the presynaptic head direction cells to the postsynaptic combination cells. Thus, through competitive learning, individual head direction cells will learn to represent combinations of a particular rotational velocity and a particular head direction θ that, due to the effective time delay in the w^3 synapses, occurred at a time $t - \Delta t$ in the past. There is no corresponding time delay in the w^2 synapses and so, through pattern association learning, associations are made between patterns of neural activity in the combination cell layer at a time t and patterns of activity in head direction cell layer at the same time t. The time-delayed competitive learning in the w^3 synapses, together with the pattern association learning in the w^2 synapses, has the following effect.

During training, visual input drives the activity in the head direction cells such that an individual head direction cell is forced to fire when the agent is facing its preferred direction. At any moment in time t the individual combination cells will respond to an earlier pattern of head direction cell firing at time $t - \Delta t$ through the synaptic connections w^3_{ij}. At the same time, the combination cells will strengthen their w^2_{ij} connections onto the head direction cells that are currently active at time t. If the agent is rotating with velocity $\frac{\mathrm{d}\theta}{\mathrm{d}t}$ then over the time interval Δt the agent will have rotated $\Delta t \frac{\mathrm{d}\theta}{\mathrm{d}t}$. Thus, the model learns that an activity pattern in the head direction cell layer occurring at time t_1, representing a particular head direction θ_1, should stimulate (via the layer of combination cells) a new

pattern of activity in the head direction cell layer at time $t_2 = t_1 + \Delta t$, representing a later head direction $\theta_2 = \theta_1 + \Delta t \frac{\mathrm{d}\theta}{\mathrm{d}t}$. This kind of associative learning over fixed time intervals enables the model to learn the correct velocity for updating the packet of neural activity in the head direction cell layer, with the result that path integration of head direction can occur at the correct speed.

7.3 Model Implementation

The governing equations for the model incorporating neuronal time constants are the same as for the model incorporating axonal conduction delays. These equations were detailed in Section 6.3 of Chapter 6 and will not be repeated here. The training and testing protocol, and the numerical solution of the differential equations, are also as described in Section 6.3 of Chapter 6.

The main difference between the two models is that in the current model, for the majority of the experiments reported in the following text, the neuronal time constant τ^{HD} is set to a value of 1ms, the neuronal time constant τ^{COMB} is set to a value of either 100ms or 150ms. In the model incorporating axonal conduction delays, the time constants τ^{HD} and τ^{COMB} are always both set to the same small value of 1ms. Also, in the current model there are no axonal conduction delays, so the axonal time delay Δt in the governing equations is always set to 0. This control measure ensures that the path integration effects demonstrated in this chapter are due to the long neuronal time constant τ^{COMB}, rather than long axonal conduction delays.

7.4 Results

This section presents the results of a series of experiments that explored the effect that controlled alteration of the model parameters have upon the learned model performance. For all of the simulations conduced, the timestep δt of the Forward Euler numerical scheme was set to 0.00005s. Simulating the model with smaller values of δt confirmed that the numerical solution had converged by $\delta t = 0.00005$s.

7.4.1 Experiment 7.1: Demonstration of Core Model Performance

The simulations in this experiment were conducted to explore the core operational principles of the model. The results of four main simulations are presented and analysed. The model is simulated with τ^{COMB} set to 150ms and 100ms. For each value of the time constant, the model is simulated with imposed rotational velocities of 180^{o}/s and 360^{o}/s.

τ^{COMB} 150ms; 180^{o}/s Rotational Velocity

The network was simulated with the parameters given in Table 7.2. The training and testing protocol is given in Section 6.3 (Chapter 6).

Figure 7.3 shows the recurrent w^1 synaptic weights after training with a Hebbian associative learning rule and weight normalization, Equations (6.3) and (6.4) respectively. In the plots, the w^1 synaptic weight distribution is symmetric about the individual presynaptic head direction cell with which the postsynaptic cell has maximal w^1 synaptic weight. The Hebbian learning rule (6.3) ensures that the presynaptic cell j with maximal weight w^1_{ij} to postsynaptic cell i will be cell i itself. The w^1

Table 7.2: Network parameter values for the simulations reported in this Chapter 7

Network Parameters	
No. Head Direction Cells	500
No, Combination Cells	1000
No. Rotational Velocity Cells	500
No. w^1 synapses onto each head direction cell	500
No. w^2 synapses onto each head direction cell	1000
No. w^3 synapses onto each combination cell	25
No. w^4 synapses onto each combination cell	500
$\tilde{w}^{\mathrm{HD}}$	375.0
$\tilde{w}^{\mathrm{COMB}}$	50.0
I^{EXTERN}	150.0
σ^{HD}	20^{o}
k^1, k^2, k^3, k^4	0.1
λ	200.0
τ^{HD}	1.0ms
ϕ_1	3.75×10^3
ϕ_2	2.5×10^3
ϕ_3	5.0×10^3
ϕ_4	4.0×10^2
No. Training Epochs	50
δt (Timestep of Forward Euler method)	0.00005s
Head Direction Cell Sigmoid Transfer Function Parameters	
α	0.0
β	1.5
Combination Cell Sigmoid Transfer Function Parameters	
α	10.0
β	1.5

All of the values are constant across all experiments reported in this chapter except where noted in the text.

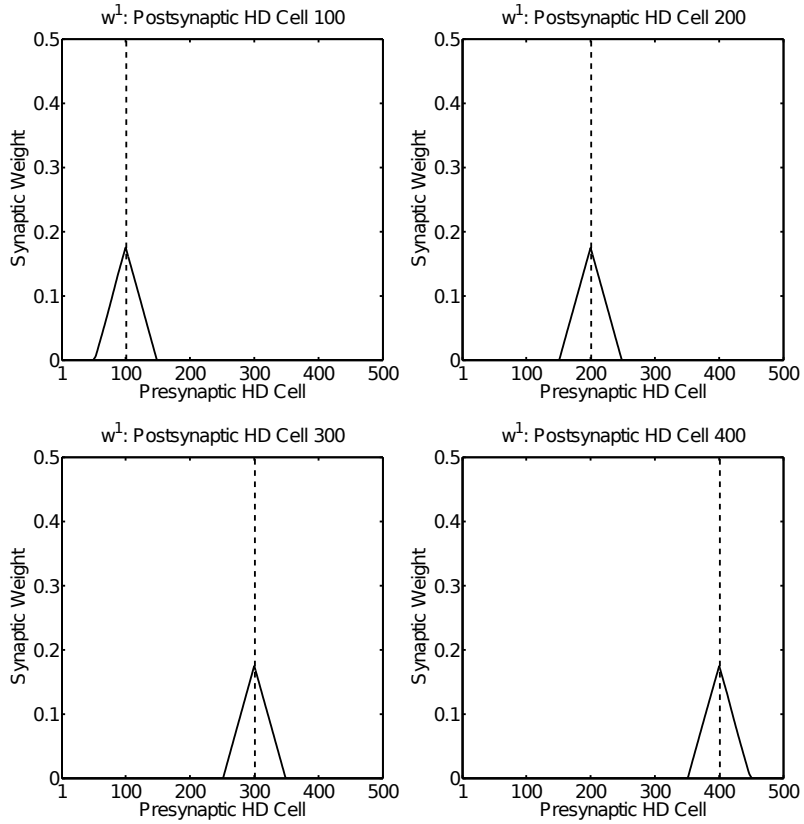

Figure 7.3: The recurrent synaptic weights w^1 within the head direction (HD) cell layer after training with a Hebbian associative learning rule, Equation (6.3), and weight normalization (6.4). The synaptic weights are recorded from a simulation implemented with a τ^{COMB} of 150ms, and an imposed rotational velocity of 180^{o}/s. All of the other network parameters are given in Table 7.2. Each of the four plots displays the learned synaptic weights to a different postsynaptic HD cell from the other 500 presynaptic HD cells in the HD cell layer. In the plots the presynaptic HD cells are arranged according to where they fire maximally in the head-direction space of the agent when visual input is available. For each plot a dashed vertical line indicates the presynaptic HD cell with which the postsynaptic HD cell has maximal w^1 synaptic weight. In all of the plots the synaptic weight profile is symmetric about the presynaptic HD cell with maximal strength of the synaptic connection to the postsynaptic HD cell. Due to the Hebbian learning, the presynaptic HD cell j with maximal w^1_{ij} to postsynaptic HD cell i will, in fact, be HD cell i itself. The symmetry of the w^1 synaptic weight distribution helps to support a stable and persistent packet of activity in the absence of visual input.

synaptic weight distribution tails off symmetrically about this cell. The symmetry of the w^1 synaptic weight distribution allows a stable and persistent packet of neural activity to be maintained in the head direction cell layer when the agent is stationary and there is no visual input.

Figure 7.4 displays the synaptic weights w^3 from the presynaptic head direction cell layer to the post-

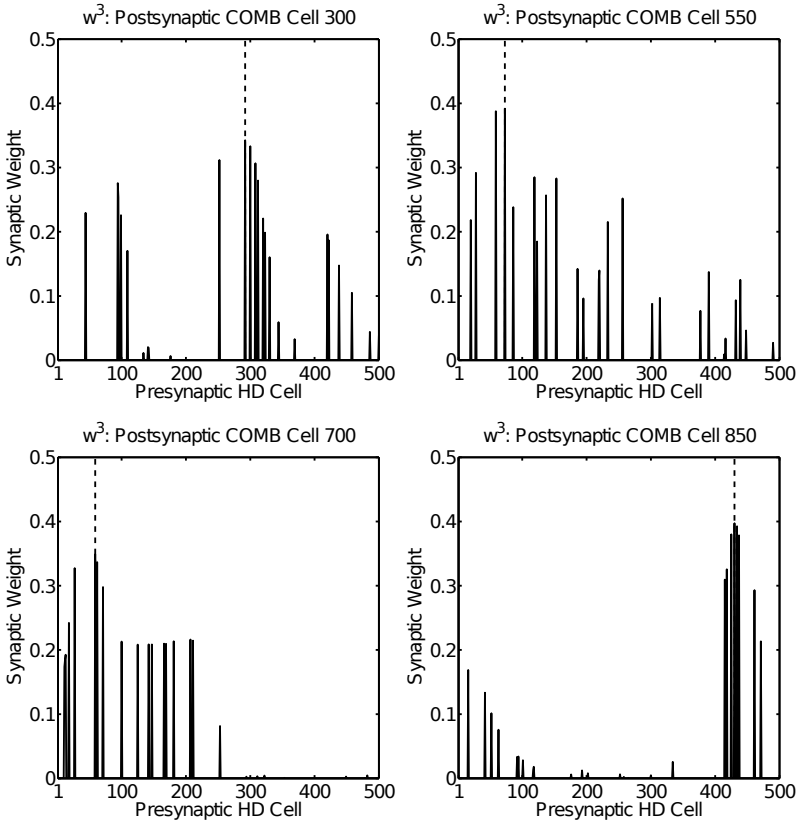

Figure 7.4: The synaptic weights w^3 from the head direction (HD) cell layer to the combination (COMB) cell layer after competitive learning with a Hebbian associative learning rule, Equation (6.9), and weight normalization (6.10). The synaptic weights are taken from a simulation implemented with a τ^{COMB} of 150ms, and a rotational velocity imposed during training of 180^{o}/s. All of the other network parameters are given in Table 7.2. Each of the four plots displays the synaptic weights from the 500 presynaptic HD cells to a different postsynaptic COMB cell. The presynaptic HD cells are arranged in the plots according to where they fire maximally in the head-direction space of the agent when visual input is available. The dashed vertical line indicates the presynaptic HD cell which has maximal w^3 synaptic strength with the postsynaptic COMB cell. With the exception of diluted synaptic connectivity, each of the weight profiles is centred on a region of similarly tuned HD cells. The w^3_{ij} weight distribution is approximately symmetric about the presynaptic HD cell that has maximal synaptic strength with the postsynaptic COMB cell. The learned w^3 synaptic weights show that individual COMB cells learn to receive maximal stimulation from a subset of HD cells representing nearby head directions. The model parameters ϕ_3, ϕ_4, and the threshold α of the COMB cell sigmoid transfer function are tuned to ensure that a strong rotational velocity cell input is also needed to stimulate the COMB cells into firing. The combination cells thus learn to respond to combinations of a particular head direction and clockwise or counter-clockwise rotational velocity.

synaptic combination cell layer after competitive learning with a Hebbian associative learning rule and weight normalization, Equations (6.9) and (6.10) respectively.. The individual plots show the learned synaptic weights from the 500 presynaptic head direction cells to a different postsynaptic combination

cell. With the exception of diluted connectivity, implemented to preserve competitive learning in the combination cell layer, the synaptic weight profiles in Figure 7.4 are centred on regions of similarly-tuned head direction cells. The synaptic weight distributions are approximately symmetric about the presynaptic head direction cell with which the postsynaptic combination cell has maximal w^3 synaptic weight. The symmetry of the weight distribution indicates that individual postsynaptic combination cells have learned to be preferentially stimulated by a small subset of presynaptic head direction cells representing nearby head directions. The combination cells have learned to respond to combinations of a particular head direction and clockwise or counter-clockwise rotational velocity.

The plots in Figure 7.5 display the synaptic weights w^2 from the presynaptic combination cell layer to the postsynaptic head direction cell layer after learning with a Hebbian associative learning rule, Equation (6.5), and weight normalization, (6.6). It is clear that the w^2 synaptic weight profile is asymmetric about the postsynaptic head direction cell with maximal w^3 synaptic weight to that combination cell. This demonstrates that the presynaptic combination cell has learned to preferentially stimulate a different localized cluster of head direction cells to the presynaptic head direction cell that preferentially stimulates the combination cell. During training in presence of visual input, the packet of head direction cell activity, representing the current head direction, will have moved through the head-direction space of the agent in the natural time delay Δt between the activity profile in the head direction cell network and the activity profile in the combination cell network. This results, at any given moment in time, in the most active combination cell learning to stimulate a different postsynaptic head direction to the presynaptic head direction it has learned to be stimulated by, reflecting the changing head direction of the agent. Therefore, the neuronal time constants τ^{COMB} and τ^{HD} act,

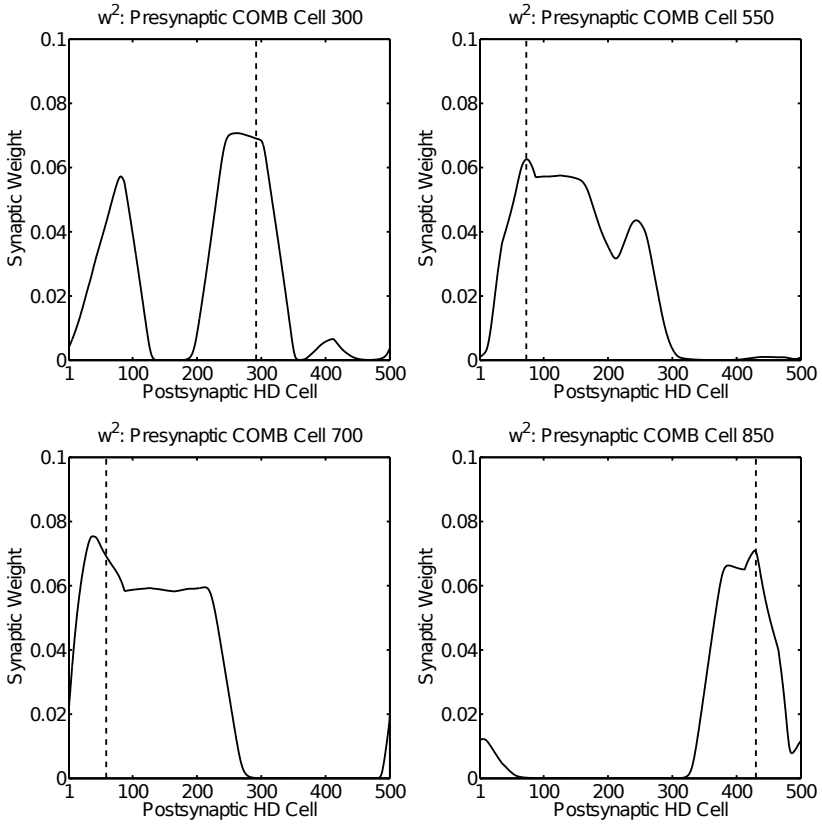

Figure 7.5: The synaptic weights w^2 from the presynaptic combination (COMB) cell layer to the postsynaptic head direction (HD) cell layer after training with a Hebbian associative learning rule and weight normalization. These results are taken from a simulation implemented with a τ^{COMB} of 150ms, and a rotational velocity imposed during training of 180°/s. The other network parameters are as given in Table 7.2. Each plot shows the learned synaptic weights from a different presynaptic COMB cell to the 500 postsynaptic HD cells. The HD cells are arranged in the plots according to where they fire maximally in the head-direction space of the agent when visual input is available. The dashed vertical line indicates the postsynaptic HD cell with which the presynaptic COMB cell has maximal w^3 synaptic weight as shown in Figure 7.4. It is evident that the w^2 weight profiles are asymmetric about the postsynaptic HD cell with maximal w^3 synaptic weight to the COMB cell. The asymmetry allows each COMB cell to preferentially stimulate a different postsynaptic HD cell from the presynaptic HD cell that preferentially stimulates it. This reflects the fact that the packet of HD cell activity will have moved through the head-direction space of the agent in the time delay Δt between the firing of the presynaptic HD cells and the resultant firing of the postsynaptic COMB cells. Thus, under the correct values for τ^{COMB} and τ^{HD}, a natural time delay is produced that acts as a timing mechanism to allow associations to be learned between packets of HD cell activity and packets of COMB cell activity that occur at different moments in time. It is these associations that allow the packet of HD cell activity to be updated at the same speed as the agent is rotating.

through learning, as a timing mechanism that is sufficient for the purpose of enabling the update of the packet of head direction cell activity at the same speed as the agent is rotating.

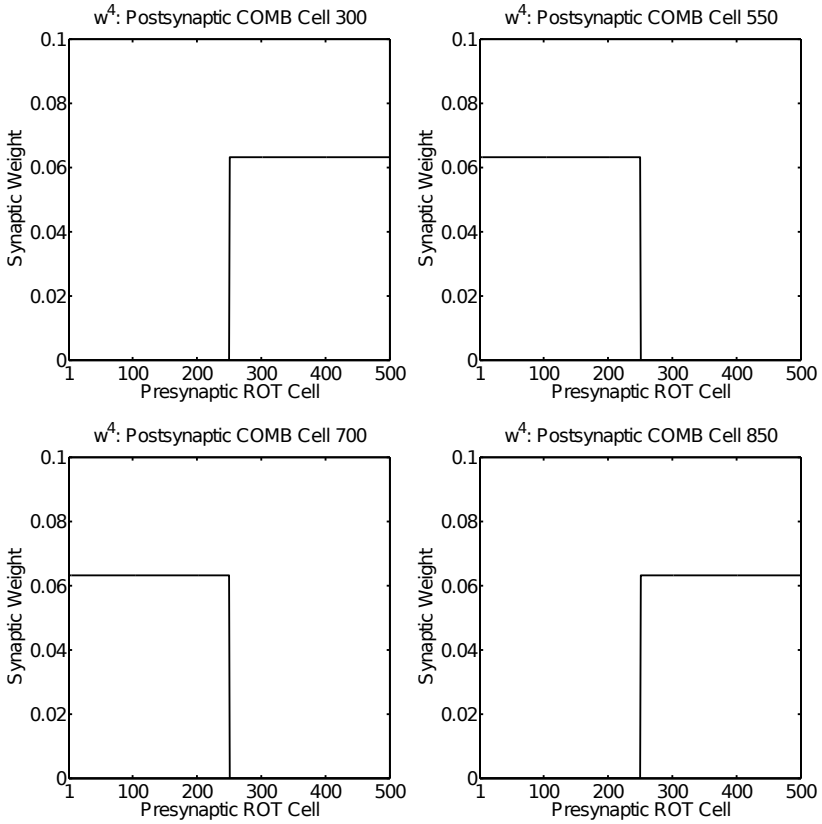

Figure 7.6: The synaptic weights w^4 from the layer of presynaptic rotational velocity (ROT) cells to the layer of postsynaptic combination (COMB) cells after learning with a Hebbian associative learning rule, Equation (6.11), and weight normalization, Equation (6.12). These synaptic weights are recorded from a simulation implemented with a τ^{COMB} of 150ms, and a rotational velocity imposed during training of 180^{o}/s. All of the other network parameters are given in Table 7.2. Each of the four plots shows the learned synaptic weights to a different postsynaptic COMB cell from the 500 presynaptic ROT cells. Each postsynaptic combination cell either develops strong w^4_{ij} connections with rotational velocity cells $1-250$ signalling clockwise rotation, or develops strong w^4_{ij} connections with rotational velocity cells $251-500$ signalling counter-clockwise rotation.

Figure 7.6 displays the w^4 synaptic weights from the layer of presynaptic rotational velocity cells to the layer of postsynaptic combination cells after training with a Hebbian associative learing rule and weight normalization, Equations (6.11) and (6.12) respectively. The binary nature of the rotational velocity cell firing rates means that the learned w^4_{ij} synaptic weight profiles are step functions. Each combination cell receives positive synaptic weights from one subset of 250 rotational velocity cells only. That is, each postsynaptic combination cell will either be preferentially stimulated by rotational velocity cells signalling clockwise rotation, or by rotational velocity cells signalling counter-clockwise

rotation, but not by both subsets. Thus, the postsynaptic combination cell have learned to be preferentially stimulated by a particular head direction occurring Δt in the past and a particular rotational velocity (either clockwise or counter-clockwise).

The firing rates of the head direction, combination, and rotational velocity cells are displayed in Figure 7.7. The top-left plot displays the firing rates of the head direction cells during the training phase of the experiment. Throughout the training, the activity in the head direction cells is driven by the presence of external visual input. In the time interval 0.0-2.25 seconds, the agent rotated in a clockwise direction, with counter-clockwise rotation occurring in the time interval $2.25 - 4.5$ seconds. The top right plot displays the firing rates of the head direction cells in the testing phase without external visual input. During the time interval 0.0-1.0 seconds there was no firing in the layer of rotational velocity cells (bottom-left plot), and a moderate amount of baseline activity in the layer of combination cells (bottom-right plot). Thus, there was a stable and persistent packet of head direction cell activity supported by the w^1 recurrent synapses (top-right plot).

During the time interval 1.0-2.0 seconds, the 250 rotational velocity cells representing clockwise rotation became active (bottom left), stimulating activity in the combination cells (bottom right) through the w^4 synapses in conjunction with the presynaptic head direction cell input through the w^3 synapses. As a result of the high value of the neuronal time constant τ^{COMB}, and the asymmetry between the w^2 and w^3 synaptic weight profiles produced during learning, the firing of the combination cells during this period of the testing phase stimulated head direction cells representing head direction cells further along in the clockwise direction of rotation. Thus, the head direction cell activity packet moved

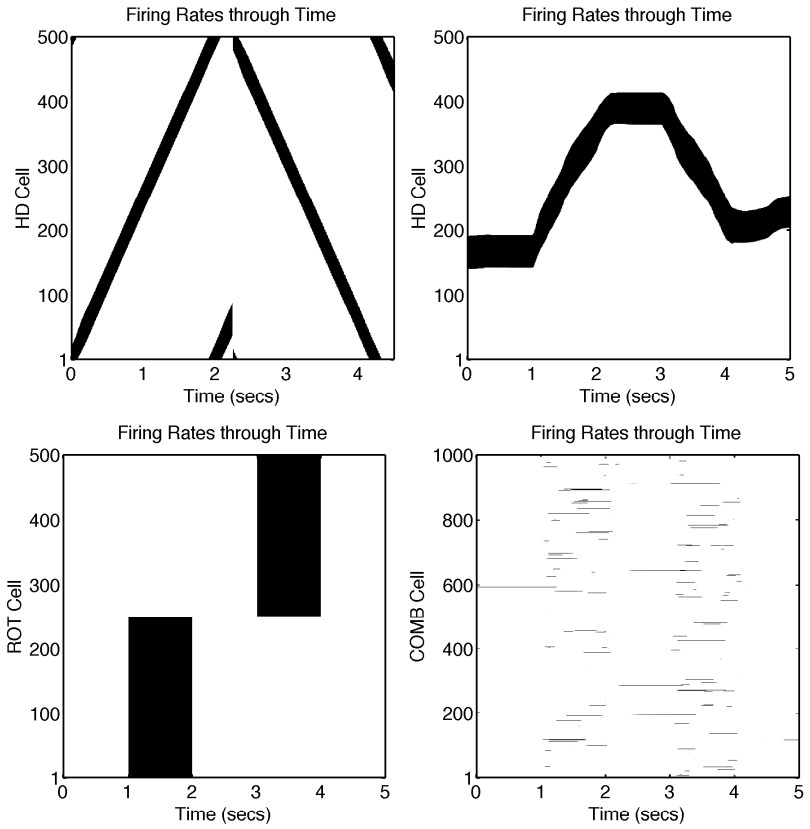

Figure 7.7: The firing rates of the head direction (HD), combination (COMB), and rotational velocity (ROT) cells during training and testing. The results are taken from a simulation implemented with a time constant τ^{COMB} of 150ms, and a rotational velocity of 180^{o}/s. All of the other network parameters are as given in Table 7.2. *Top Left* The firing rates of the 500 HD cells during the 4.5 seconds of training with the HD cells driven by visual input (0.0-2.25 seconds: agent rotating clockwise; 2.25-4.5 seconds: agent rotating counter-clockwise). *Top Right* The firing rates in the layer of HD cells during the 5 seconds of testing in the absence of visual input. During the interval 0.0-1.0 seconds there was a stable packet of activity in the HD cell layer. In the interval 1.0-2.0 seconds, the ROT cells signalling clockwise rotation were turned on, which in turn stimulated a packet of activity in the COMB cell layer that drives the packet of HD cell activity through the HD cell layer in a clockwise direction. During the interval 2.0-3.0 seconds, there was no ROT cell firing and the packet of HD cell activity remained stable and persistent within the HD cell layer. In the interval 3.0-4.0 seconds, the ROT cells signalling counter-clockwise rotation were turned on, stimulating a packet of COMB cell activity that drives the packet of HD cell activity through the HD cell layer in a counter-clockwise direction. During the interval 4.0-5.0 seconds the packet of HD cell activity again remained stable and persistent within the HD cell layer. *Bottom Left* The firing rates in the layer of 500 ROT cells during the 5 seconds of testing in the absence of visual input (1.0-2.0 seconds: ROT cells signalling clockwise rotation are active; 3.0-4.0 seconds: ROT cells signalling counter-clockwise rotation are active.) *Bottom Right* The firing rates in the layer of 1000 COMB cells during the 5 seconds of testing in the absence of visual input. In the interval 1.0-2.0 seconds, the COMB cells become active due to the firing of the 250 ROT cells signalling clockwise rotation of the agent. In the interval 3.0-4.0 seconds, the COMB cells become active due to the firing of the 250 ROT cells representing counter-clockwise rotation. At other periods of time there is a moderate amount of baseline activity in the COMB cell layer. In all of the plots regions of high firing are represented by darker shading.

through the head-direction space of the agent, and the model performed path integration of head direction in the clockwise direction. During the time interval 2.0-3.0 seconds, the rotational velocity cells were quiescent in their firing, and there was only a moderate amount of baseline activity in the layer of combination cells. Thus, the head direction cell activity packet remained stable and persistent in the head direction cell layer.

During the time interval 3.0-4.0 seconds, the 250 rotational velocity cells representing counter-clockwise rotation started firing (bottom left) and stimulated activity in the layer of combination cells (bottom right). In an identical mechanism to that for clockwise rotation, the model thus performed velocity path integration of head direction, but this time in the counter-clockwise direction of rotation. During the time interval 4.0-5.0 seconds, the rotational velocity cells ceased firings, the activity level in the combination cell layer returned to a baseline level, and again there was a stable and persistent packet of activity in the head direction cell layer.

To determine whether the model can perform path integration of head direction at the same speed during testing as was imposed during training, the speed of update of the head direction cell activity packet was recorded during the testing phase. The details of the measurement are described in Chapter 6, Equations (6.15) and (6.16) and will not be repeated here.

Measurements of the packet speed were taken for 0.5 seconds during both the clockwise (1.25-1.75 seconds) and counter-clockwise (3.25-3.75 seconds) periods of rotation during the testing phase.

Table 7.3: The speed of movement of the head direction cell activity packet during testing

τ^{COMB} 150ms; 180°/s Rotational Velocity		
	Clockwise	**Counter-Clockwise**
Mean Speed	121.48°/s	128.38°/s
Standard Deviation	24.7°/s	19.0°/s
Percentage	67.49%	71.32%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With a τ^{COMB} of 150ms, and a rotational velocity of 180°/s, the model learns to perform path integration in the absence of visual input at approximately the same speed that was imposed during training in the presence of visual input.

Five simulations were conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. The results are displayed in Table 7.3. The mean speed of packet update was 121.48°/s (S.D. = 24.7°/s) for clockwise rotation, and 128.38°/s (S.D. = 19.0°/s) for counter-clockwise rotation. This is compared to a speed of 180°/s experienced during training in the presence of visual input. For the period of clockwise rotation, the model updated the packet of head direction cell activity at a speed that is 67.49% of the speed experienced during training. For the period of counter-clockwise rotation, the recorded speed is 71.32% of the speed during training.

The model has thus learned to perform path integration of head direction at speeds that are approximately those experienced during training. Moreover, the model can self-organize to achieve this and there is no need to hand-tune a parameter governing the speed of packet update. It is noted, however, that mean speeds of packet update consistently undershoot the true speed experienced during training.

τ^{COMB} 150ms; 360°/s Rotational Velocity

The simulations in this experiment were conducted to demonstrate that if the model is trained with the same parameter set, but at twice the rotational velocity, then the model can still perform path integration of head direction at the speed experienced during training. Thus, the same model can learn to perform path integration across a range of velocities. Except for the difference in rotational velocity, all the model parameters are the same as for above (given in Table 7.2).

The plots in Figure 7.8 display the firing rates of the head direction, combination, and rotational velocity cells during the training and testing phases. The conventions are the same as for Figure 7.7 and so is the interpretation. When the same model is trained at twice the rotational velocity compared to the simulation reported above, the same mechanism of a high value for the neuronal time constant τ^{COMB} on the activations of the combination cells produces a natural time delay Δt between the activity profiles in the head direction cell layer and the activity profile in the combination cell layer. This delay enables associations to be learned between different head directions of the agent at different points in time.

To quantify the model performance, the speed of update of the head direction cell activity packet was measured from five different simulations, each conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. The results are displayed in Table 7.4.

The mean speed of rotation was 196.96^{o}/s (S.D. = 26.1^{o}/s) for the period of clockwise rotation, and

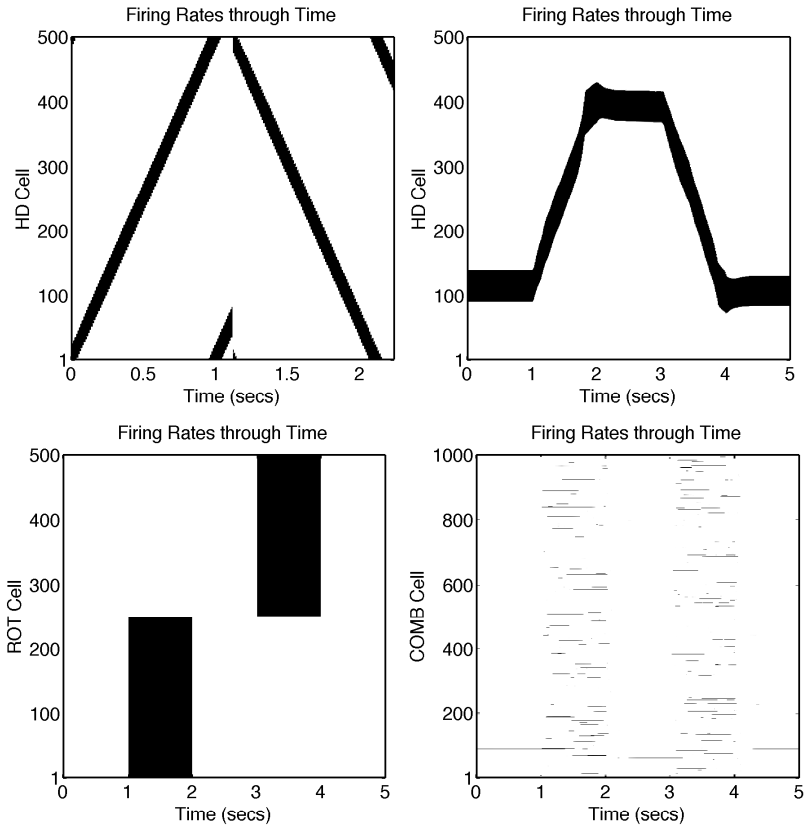

Figure 7.8: The firing rates of the head direction (HD), combination (COMB), and rotational velocity (ROT) cells during training and testing. These results are taken from a simulation implemented with a time constant τ^{COMB} of 150ms, and a rotational velocity of 360°/s. All of the other network parameters are as given in Table 7.2

Table 7.4: The speed of movement of the head direction cell activity packet during testing

τ^{COMB} 150ms; 360°/s Rotational Velocity		
	Clockwise	**Counter-Clockwise**
Mean Speed	196.96°/s	198.45°/s
Standard Deviation	26.1°/s	19.9°/s
Percentage	54.71%	55.12%

Table 7.5: The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With a time constant τ^{COMB} of 150ms, and a rotational velocity of 360°/s, the model learns to perform path integration at approximately the same speed as was imposed during training in the presence of visual input. The percentage accuracy of path integration is, however, reduced at this higher rotational velocity.

198.45°/s (S.D. = 19.9°/s) for the period of counter-clockwise rotation. This is in comparison to a true speed imposed during training of 360°/s. During the clockwise rotation, the speed is 54.71% of the speed imposed during training. During the counter-clockwise rotation the recorded speed is 55.12% of the speed imposed during training.

Thus, the speeds recorded during testing are approximately the speeds experienced by the model during training. However, there is a regular underestimation of the correct speed. Despite this, the model is capable of learning to perform path integration at two different rotational velocities. Although, the percentage accuracy of the path integration is reduced at the higher rotational velocity.

τ^{COMB} 100ms; 180°/s Rotational Velocity

These simulations were conducted to demonstrate that the model can learn to perform path integration of head direction when simulated with a different value for the neuronal time constant τ^{COMB}.

Figure 7.9 displays the firing rates of the head direction cells during the 5 seconds of testing the model in the absence of visual input. The conventions are the same as for the top-right plots of Figures 7.7 and 7.8, and demonstrate that the model has learned to perform velocity path integration of head direction.

The speed of update of the head direction cell activity packet was recorded from five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. The results are given in Table 7.6.

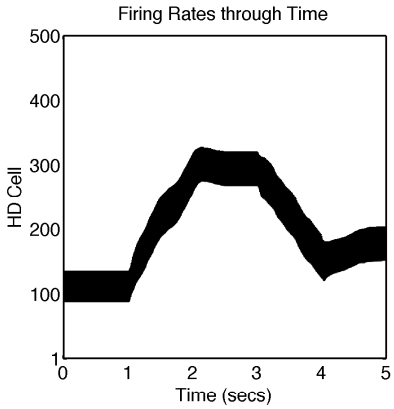

Figure 7.9: The firing rates of the head direction (HD) cells during testing in the absence of visual input. These results are taken from simulations implemented with a time constant τ^{COMB} of 100ms, and a rotational velocity during training of 180^{o}/s. The model has learned to perform path integration of head direction.

Table 7.6: The speed of movement of the head direction cell activity packet during testing

τ^{COMB} **100ms; 180°/s Rotational Velocity**		
	Clockwise	**Counter-Clockwise**
Mean Speed	113.15^{o}/s	105.85^{o}/s
Standard Deviation	5.2^{o}/s	10.7^{o}/s
Percentage	62.86%	58.81%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With a time constant τ^{COMB} of 100ms, and a rotational velocity of 180^{o}/s, the model learns to perform path integration at approximately the speed imposed during training.

During the clockwise rotation the mean speed of packet update was 113.15°/s (S.D. = 5.2°/s). For the counter-clockwise rotation, the mean speed was 105.85°/s (S.D. = 10.7°/s). This is compared to a true speed imposed during training of 180°/s. The speed during clockwise rotation is 62.86% of the speed imposed during training. During counter-clockwise rotation the speed is 58.81% of the speed imposed during training.

The speeds recording during testing are approximately the speeds experienced by the model during training. There is, however, a constant underestimation of the true speed even though the recorded speed is within the same order of magnitude as the speed imposed during training. The model can thus be implemented with a different value for the time constant τ^{COMB} and can still learn to update a packet of head direction cell activity at the speed that was experienced during training. In comparison, however, to the model trained with the neuronal time constants τ^{COMB} set to 150ms, and rotational velocity of 180°/s imposed during training in the light, Table 7.3, the model trained at the same rotational velocity but with the neuronal time constants τ^{COMB} set to 100ms has a reduced percentage accuracy of path integration.

τ^{COMB} 100ms; 360°/s Rotational Velocity

Figure 7.10 displays the firing rates of the head direction cells during the 5 seconds of testing in the absence of visual input. The conventions are the same as for the top-right plots in Figures 7.7 and 7.8 and show that the model has learned to perform path integration of head direction.

The speed of update of the head direction cell activity packet was recorded from five simulations conducted with identical model parameters, but with different random synaptic weight initializations

and different random synaptic connectivities. The results are displayed in Table 7.7.

During clockwise rotation the mean speed was 210.87^{o}/s (S.D. = 38.0^{o}/s). For the period of counter-clockwise rotation, the mean speed was 223.63^{o}/s (S.D. = 8.2^{o}/s). This is compared to a true speed of 360^{o}/s imposed during training. The clockwise rotation is 58.57% of the speed during training. The counter-clockwise rotation is 61.96% of the speed during training.

The model can thus perform velocity path integration of head direction at approximately the same speed as was experienced during training. Similar to previous experiments there is regular underestimation of the correct speed. However, the same general model implemented with different values for the time constant τ^{COMB} and trained at different rotational velocities can, in all cases, learn to perform velocity path integration of head direction at a speed that is within the same order of magnitude of the speed experienced by the model during training. The results of the model simulated with a value of 100ms for the neuronal time constants τ^{COMB}, and a rotational velocity of 360^{o}/s imposed during training, are quantitatively very similar to the results obtained from the model simulated with the same imposed rotational velocity, but with the neuronal time constants τ^{COMB} set to 150ms (Table 7.4. In fact, there is a slight increase in the accuracy of path integration when the model is simulated with the neuronal time constants atu^{COMB} set to 100ms compared to the model simulated with the neuronal time constants τ^{COMB} set to 150ms. It is interesting to note that, when simulated with a rotational velocity of 360^{o}/s, changing the value of the time constants τ^{COMB} from 150ms to 100ms has a small effect upon the percentage accuracy of the path integration. When the model is simulated with a rotational velocity of 180^{o}/s, however, changing the value of the time constants τ^{COMB} from

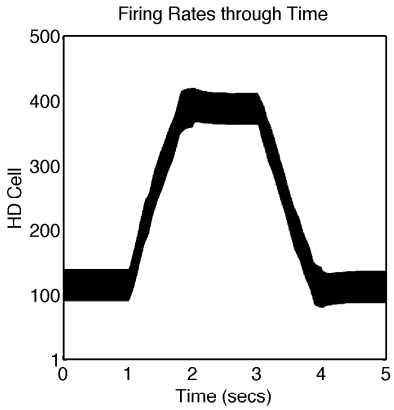

Figure 7.10: The firing rates of the head direction (HD) cells recorded during testing in the absence of visual input. The results are taken from a simulation implemented with a time constant τ^{COMB} of 100ms, and a rotational velocity during training of 360°/s. It is clear that the model has learned to perform path integration of head direction.

Table 7.7: The speed of movement of the head direction cell activity packet during testing

τ^{COMB} **100ms; 360°/s Rotational Velocity.**		
	Clockwise	**Counter-Clockwise**
Mean Speed	210.87°/s	223.63°/s
Standard Deviation	38.0°/s	8.2°/s
Percentage	58.57%	61.96%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With a time constant τ^{COMB} of 100ms, and a rotational velocity of 360°/s, the model learns to perform path integration at approximately the speed imposed during training.

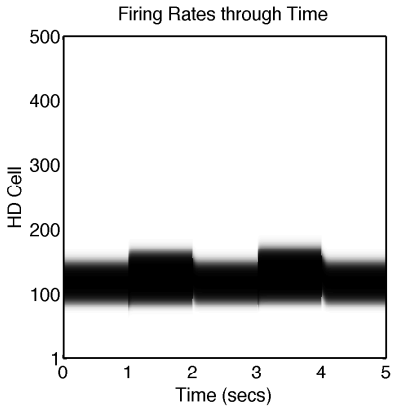

Figure 7.11: Firing rates of the head direction (HD) cells recorded during the testing phase of Experiment 7.2. It is evident that with a small value for the time constant τ^{COMB}, the model cannot learn to perform path integration of head direction.

150ms to 100ms has a large effect upon the percentage accuracy of the path integration.

7.4.2 Experiment 7.2: τ^{COMB} Must be a Large Value

This experiment was conducted to demonstrate that a large value for the time constant τ^{COMB} is required if the model is to successfully learn path integration of head direction. τ^{COMB} was set to 1.0ms, and the model was simulated with a rotational velocity of 360^{o}/s imposed during training.

Figure 7.11 displays the firing rates of the head direction cells recorded during testing in the absence of visual input. It is clear that the model cannot learn to perform path integration of head direction when the time constants τ^{HD} and τ^{COMB} are both set to small values (in this case, the same value).

When τ^{COMB} is set to a small value, the activity profile in the postsynaptic combination cell i will not be delayed through time from the activity profile of the presynaptic head direction cell j. Because there are no axonal conduction delays in the current model, signal transmission is instantaneous with

the presynaptic cell firing. This means that, in the current experiment, the current head direction θ_1 will, after signal transmission to the combination cell layer and back again to the head direction cell layer, become associated with itself. That is, a head direction cell representing head direction θ_1 does not become associated with a head direction cell representing head direction θ_2 occurring at some later time in the future. Consequently, the model will not learn to perform path integration of head direction.

The results of the current experiment also highlight the difference between the current model and the model reported in Chapter 6, incorporating axonal conduction delays. If, for the current model, the time constants τ^{HD} and τ^{COMB} are both set to small values, then some other mechanism for producing an effective time delay is required if the model is to learn to perform path integration of head direction. It is also clear that, for the model reported in Chapter 6, it is the axonal conduction delays that produce the effective time delay, and not the neuronal time constants, which were set to a small value of $1\mathrm{ms}$. Thus, the two models share some similarities but operate with completely different timing mechanisms.

7.4.3 Experiment 7.3: A Distribution of Time Constants τ^{COMB}

This experiment was conducted to demonstrate that the model can still learn to perform path integration when the model is implemented with a distribution of values for the neuronal time constant τ^{COMB}. When the network is initialized, the time constant τ^{COMB} for each combination cell is randomly assigned a value from within the interval $[1\mathrm{ms}, 150\mathrm{ms}]$ according to a uniform distribution over this interval. A rotational velocity of 360^{o}/s was imposed during training in the light. The network was simulated with the parameters given in Table 7.2.

Figure 7.12 displays the firing rates of head direction cells recorded during testing in the dark. The plot clearly shows that the model can learn to perform path integration of head direction when implemented with a wide range of values,within the same model, for the time constant τ^{COMB} .

The speed of update of the head direction cell activity packet is given in Table 7.8. During the period of clockwise rotation, the mean speed of packet update was 189.13^{o}/s (S.D. = 28.81^{o}/s). During the period of counter-clockwise rotation, the mean speed of packet update was 203.46^{o}/s (S.D. = 30.28^{o}/s). For the clockwise rotation, the packet of head direction cell activity updated at a speed that was 52.54% of the speed imposed during training. During the period of counter-clockwise rotation, the packet updated at a speed that was 56.52% of the speed imposed during training.

These results show that the model can learn to perform path integration of head direction when implemented with a distribution of values for the neuronal time constant τ^{COMB}. This is an important experiment because it shows that the idea of using the values of the neuronal time constants as an effective time delay is robust across a range of values, from 1ms to 150ms, for time constant τ^{COMB}. Moreover, the results demonstrate that the learned model behaviour is not dependent upon a particular value for the time constant τ^{COMB}. This experiment also shows that the model possesses biological validity, as it is likely that there are a range of values for the neuronal time constants in the brain.

7.4.4 Experiment 7.4: The Effect of the Learning Rate

This experiment investigated the effect that different values of the learning rate k have upon the learned model performance.

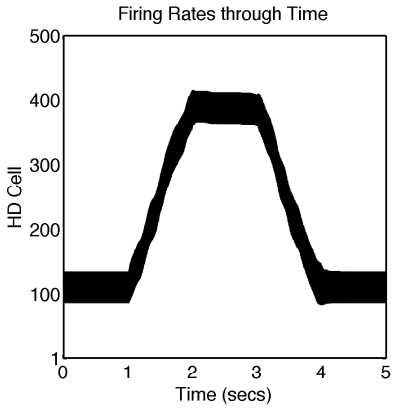

Figure 7.12: The firing rates of head direction (HD) cells during the testing phase of Experiment 7.3. The model can learn to perform path integration of head direction when implemented with a range of values for the neuronal time constant τ^{COMB}.

Table 7.8: The speed of movement of the head direction cell activity packet during testing

Experiment 7.3: $\tau^{\mathrm{COMB}} \in [1\mathrm{ms}, 100\mathrm{ms}]$; 360°/s Rotational Velocity		
	Clockwise	**Counter-Clockwise**
Mean Speed	189.13°/s	203.46°/s
Standard Deviation	28.81°/s	30.28°/s
Percentage	52.54%	56.52%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With a uniform distribution of values for the time constant $\tau^{\mathrm{COMB}} \in [1\mathrm{ms}, 100\mathrm{ms}]$, and a rotational velocity of 360°/s, the model learns to perform path integration at approximately the speed imposed during training.

Learning Rates k^2 and k^3 of 0.01

For the simulations comprising this part of the experiment, the learning rates k^2 and k^3 were set to 0.01. The learning rates k^1 and k^4 were set to 0.1. A rotational velocity of 360°/s was imposed during training in the light. The time constant τ^{COMB} was set to 100ms. All of the other network parameters were set to the values given in Table 7.2.

The left-hand plot of Figure 7.13 displays the firing rates of the head direction cells during testing in the dark. The model has clearly learned to perform path integration of head direction.

During the period of clockwise rotation, the mean speed of packet update was 173.58°/s (S.D. = 39.84°/s), which was 48.22% of the speed imposed during training. During the period of counter-clockwise rotation, the mean speed of packet update was 206.33°/s (S.D. = 2.89°/s), which was 57.31% of the speed imposed during training. These results are displayed in Table 7.9.

The model can thus learn to perform path integration of head direction when the learning rates k^2 and k^3 were set to a lower value of 0.01. The percentage accuracy of the path integration is similar to the percentage accuracy for the model simulated with the same parameters, and trained at the same rotational velocity, but with the learning rates k^2 and k^3 set to a value of 0.1 (Table 7.7). The current model does exhibit a decrease in the percentage accuracy of the clockwise rotation in comparison to the results given in Table 7.7.

Further simulations were conducted in which all of the learning rates, k^1 to k^4, were set to 0.01.

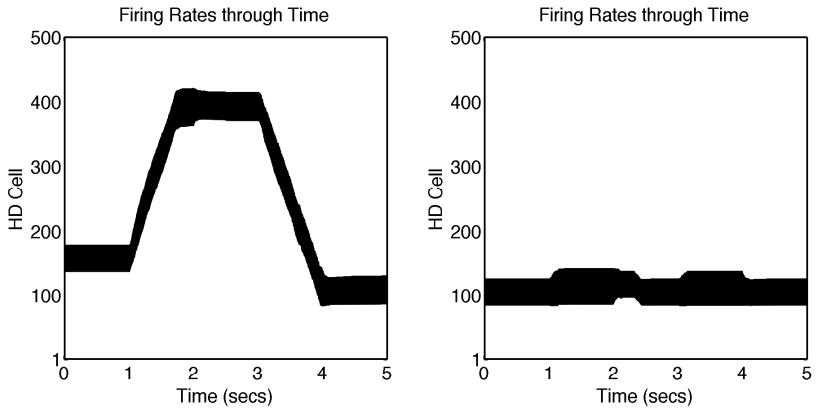

Figure 7.13: The firing rates of the head direction cells recorded during the testing phase of Experiment 7.4. The left-hand plot displays the results of a simulation conducted with the learning rates k^2 and k^3 set to 0.01. The model has learned to perform path integration. The right-hand plot displays the results of a simulation conducted with the learning rates k^2 and k^3 set to 0.001. The model cannot learn to perform path integration of head direction.

Table 7.9: The speed of movement of the head direction cell activity packet during testing

Experiment 7.4: k^2 and k^3 set to 0.01		
	Clockwise	**Counter-Clockwise**
Mean Speed	173.58°/s	206.33°/s
Standard Deviation	39.84°/s	2.89°/s
Percentage	48.22%	57.31%

The results displayed are the average of five simulations conducted with identical model parameters, but with different random synaptic weight initializations and different random synaptic connectivities. With the learning rates k^2 and k^3 set to 0.01, τ^{COMB} set to 100ms, and a rotational velocity of 360°/s, the model learns to perform path integration at approximately the speed imposed during training.

The results are not shown, but are qualitatively very similar to the results obtained when only k^2 and k^3 are set to 0.01

Learning Rates k^2 and k^3 set to 0.001

For the simulations comprising this part of the experiment, the learning rates k^2 and k^3 were set to 0.001. The learning rates k^1 and k^4 were set to 0.1. The model was simulated with a time constant τ^{COMB} of 100ms, and a rotational velocity imposed during training in the light of 360^{o}/s. The remaining network parameters were set to the values given in Table 7.2.

The right-hand plot of Figure 7.13 displays the firing rates of the head direction cells during testing in the dark. With the learning rates k^2 and k^3 set to 0.001, the model cannot learn to perform path integration of head direction.

Further simulations, with all learning rates k^1 to k^4 set to 0.001, produced qualitatively similar results to the simulations with only k^2 and k^3 set to 0.001.

The learning rates k^2 and k^3 must be implemented with relatively high values in order for the model to learn to perform path integration of head direction.

7.4.5 Experiment 7.5: Long-Term Depression in the Learning Rules

Long-Term Depression (LTD) can increase the specificity of learned cell responses because a learning rule that incorporates LTD permits the strength of a synapse w_{ij} to decrease as well as increase. This experiment explored the effect of incorporating first homosynaptic LTD, and then heterosynaptic LTD,

into the learning rules governing the synaptic weight update.

Homosynaptic LTD

Homosynaptic LTD promotes a decrease in the strength of a synapse w_{ij} when strong firing of the presynaptic cell j is temporally conjunctive with below-average firing of the postsynaptic cell i.

Equation (6.5) now becomes

$$\frac{dw_{ij}^{2}(t)}{dt} = k^{2}\left[r_{i}^{\mathrm{HD}}(t) - \langle r_{i}^{\mathrm{HD}}\rangle\right] r_{j}^{\mathrm{COMB}}(t) \tag{7.4}$$

where k^2 is the learning rate, $r_i^{\mathrm{HD}}(t)$ is the firing rate of postsynaptic head direction cell i at time t, $\langle r_i^{\mathrm{HD}}\rangle$ is the time-average firing rate of postsynaptic head direction cell i, and $r_j^{\mathrm{COMB}}(t)$ is the firing rate of presynaptic combination cell j at time t. The average firing rate $\langle r_i^{\mathrm{HD}}\rangle$ is computed by, at each model update, summing together all the firing rates r_i for a particular postsynaptic cell i that have occurred at each model update so far, and then dividing by the total number of model updates.

Equation (6.9) now becomes

$$\frac{dw_{ij}^{3}(t)}{dt} = k^{3}\left[r_{i}^{\mathrm{COMB}}(t) - \langle r_{i}^{\mathrm{COMB}}\rangle\right] r_{j}^{\mathrm{HD}}(t) \tag{7.5}$$

where k^3 is the learning rate, $r_i^{\mathrm{COMB}}(t)$ is the firing rate of postsynaptic combination cell i at time t, $\langle r_i^{\mathrm{COMB}}\rangle$ is the time-average firing rate of postsynaptic combination cell i, and $r_j^{\mathrm{HD}}(t)$ is the firing rate of presynaptic head direction cell j at a time t. The average firing rate $\langle r_i^{\mathrm{COMB}}\rangle$ is computed in the same way as the average firing rate $\langle r_i^{\mathrm{HD}}\rangle$ described above.

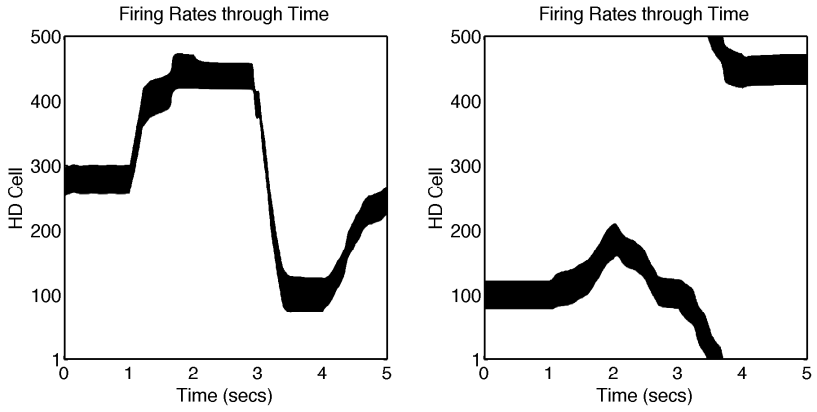

Figure 7.14: The firing rates of the head direction cells during the testing phase of Experiment 7.5 The left-hand plot displays data recorded from a model simulated with homosynaptic LTD in the w^2 and w^3 synaptic learning rules. The packet of head direction cell activity does not update in a continuous manner, and is not stable when the agent is not rotating. Therefore, the model has not learned to perform path integration of head direction with much accuracy. The right-hand plot displays data recorded from a model simulated with heterosynaptic LTD in the w^2 and w^3 learning rules. The packet of head direction cell activity does not remain stable when the agent is stationary. Therefore, the model has not learned to perform path integration of head direction.

The simulations were conducted with the network parameters given in Table 7.2. The time constant τ^{COMB} was set to 100ms, and a rotational velocity of 360^{o}/s was imposed during training in the light.

The left-hand plot of Figure 7.14 displays the firing rates of the head direction cells during testing in the dark. The head direction cell activity packet does not update in a continuous manner, and does not remain stable when the agent is stationary. Thus, the model has not learned to successfully perform path integration of head direction. This is in contrast to the corresponding Experiment 6.4.5 of Chapter 6, where the model incorporating axonal conduction delays does learn to perform path integration of head direction with homosynaptic LTD in the w^2 and w^3 learning rules.

Further simulations incorporated homosynaptic LTD into all of the learning rules for all of the synapses w^1 to w^4. The results are not shown, but are qualitatively very similar to the results with homosynaptic

LTD in the w^2 and w^3 learning rules only.

Heterosynaptic LTD

Heterosynaptic LTD promotes a decrease in the strength of a particular synapse w_{ij} when below-average firing of the presynaptic cell j is temporally conjunctive with strong firing of the postsynaptic cell i.

Equation (6.5) now becomes

$$\frac{dw_{ij}^2(t)}{dt} = k^2 r_i^{\mathrm{HD}}(t) \left[r_j^{\mathrm{COMB}}(t) - \langle r_j^{\mathrm{COMB}} \rangle \right] \tag{7.6}$$

where k^2 is the learning rate, $r_i^{\mathrm{HD}}(t)$ is the firing rate of postsynaptic head direction i at time t, $r_j^{\mathrm{COMB}}(t)$ is the firing rate of presynaptic combination cell j at time t, and $\langle r_j^{\mathrm{COMB}} \rangle$ is the time-average of the firing rate of presynaptic combination cell j, which is calculated by summing the firing rates r_j over successive model updates and then dividing by the total number of model updates so far.

Equation (6.9) now becomes

$$\frac{dw_{ij}^3(t)}{dt} = k^3 r_i^{\mathrm{COMB}}(t) \left[r_j^{\mathrm{HD}}(t) - \langle r_j^{\mathrm{HD}} \rangle \right] \tag{7.7}$$

where k^3 is the learning rate, $r_i^{\mathrm{COMB}}(t)$ is the firing rate of postsynaptic combination i at time t, $r_j^{\mathrm{HD}}(t)$ is the firing rate of presynaptic head direction cell j at time t, and $\langle r_j^{\mathrm{HD}} \rangle$ is the time-average firing rate of presynaptic head direction cell j, which is calculated in the same way as described for the average $\langle r_j^{\mathrm{COMB}} \rangle$ above.

The model was simulated with the network parameters given in Table 7.2. The time constant τ^{COMB} was set to 100ms, and a rotational velocity of 360^{o}/s imposed during training in the light.

The right-hand plot of Figure 7.14 displays the firing rates of the head direction cells during testing in the dark. The packet of head direction cell activity continuously moves, and does not move in the correct direction at the correct time. The packet also does not remain stably in one position when the agent is not rotating. Thus, a model incorporating heterosynaptic LTD in the w^2 and w^3 learning rules cannot learn to perform path integration of head direction.

Further simulations incorporated heterosynaptic LTD into all of the learning rules for the w^1 to w^4 synapses. The results are not shown, but are qualitatively very similar to the results obtained with heterosynaptic LTD in the w^2 and w^3 learning rules only.

Thus, the current model cannot learn path integration of head direction with reasonable accuracy when either homosynaptic LTD or heterosynaptic LTD are incorporated into the synaptic learning rules. There is a difference in learned behaviour compared to the model, reported in Chapter 6, that incorporates axonal conduction delays, which could learn path integration of head direction with homosynaptic LTD present in the w^2 and w^3 learning rules.

7.4.6 Experiment 7.6: Learning during the Testing Phase

In all of the simulations reported in this chapter so far, during the testing phase no learning is allowed to take place. As discussed in the corresponding Experiment 6.6 of Chapter 6, this protocol was

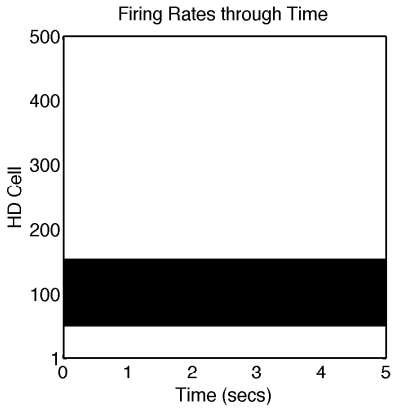

Figure 7.15: The firing rates of the head direction cells during the testing phase of Experiment 7.6. When learning is permitted during the testing phase, the model cannot perform path integration of head direction.

adopted in order to isolate the effects of the testing phase without the potentially confusing factor of constant synaptic modification. It is not clear, however, how learning could be arbitrarily turned on and off in the real brain. This experiment thus investigated the effect of allowing learning during the testing phase of the model.

During the testing phase, the associative earning and weight normalization equations for synapses w^1 to w^4 were simulated. These are Equations (6.3), (6.4), (6.5), (6.6), (6.9), (6.10), (6.11), and (6.12). The model parameters are given in Table 7.2. The time constant τ^{COMB} was set to 100ms, and a rotational velocity of 360^{o}/s was imposed during training in the light.

Figure 7.15 displays the firing rates of the head direction cells recorded during testing in the dark. When learning is permitted in the testing phase, the model cannot perform path integration of head direction. The model will learn, during the training phase, the correct synaptic weight structure to perform path integration of head direction. The further learning during the testing phase will, however, overwrite this synaptic structure and prevent path integration from occurring.

7.5 Discussion

This chapter provides the operational principles and simulation results of a model that can learn to perform path integration of head direction at approximately the speed experienced during training. In summary, the main points of the chapter are:

- The model architecture comprises a layer of head direction cells that are reciprocally linked to a layer of combination cells. A layer of rotational velocity cells signals when the agent is rotating in either a clockwise or counter-clockwise direction.

- The computational task to be solved by the model is to learn to update a packet of head direction cell activity using idiothetic (self-motion) signals at the same speed as the activity packet was updated in the presence of visual input.

- Setting the neuronal time constant τ^{COMB} to a relatively large value of either 100ms or 150ms, compared to the neuronal time constant τ^{HD}, which is set to 1ms, produces an effective time delay Δt in the w^3 synaptic connections. This is because the activity profile of a postsynaptic combination cell will be delayed in time Δt from the activity profile of the driving presynaptic head direction cell. There is, however, no corresponding effective time delay Δt in the w^2 synaptic connections. Thus, a postsynaptic combination cell will, through competitive learning, represent a particular rotational velocity and a head direction θ occurring at a time $t - \Delta t$ in the past. The activity in the combination cell layer at a time t is associated, due to the absence of a time delay Δt in the w^2 synapses, with activity in the head direction cell layer occurring at the same time t. Thus, the model can learn associations between head direction θ_1 occurring at time t_1, and head direction θ_2 occurring at time t_2. In the absence of visual input, these associations

enable the model to update packet of head direction cell activity at the correct speed.

- In simulation, the model learns to perform path integration of head direction at approximately the speed that was imposed during training in the light. This ability is robust across two different values of the time constant τ^{COMB}, and two different rotational velocities.

- The time constant τ^{COMB} must be set to a large value, otherwise the model cannot learn to perform path integration of head direction.

- The model can learn path integration of head direction when a distribution of values of the time constant τ^{COMB} are incorporated across the combination cells.

- When the learning rates k^2 and k^3 are set to 0.01 the model can still learn path integration of head direction. When the learning rates k^2 and k^3 are set to 0.001 the model cannot learn path integration of head direction.

- Incorporating either homosynaptic LTD or heterosynaptic LTD into the w^2 and w^3 synaptic learning rules either destroys the accuracy of path integration, or abolishes path integration of head direction altogether.

- Allowing learning to occur during the testing phase also abolishes path integration of head direction.

The current model is very similar, architecturally and functionally, to the model incorporating axonal conduction delays that is reported in Chapter 6. The experiments reported in the current chapter were carried out as careful replications of the experiments reported in Chapter 6. This approach was adopted in order to highlight the similarities and differences between the two models.

Importantly, the principle of using natural time delays as a mechanism for producing accurate path integration of head direction has been shown to be a general principle. Two separate physical instantiations of this principle have now been shown to produce the desired path integration of head direction behaviour: axonal conduction delays, and neuronal time constants.

There are differences between the two models. A major difference is that the current model, with a large value for the neuronal time constant τ^{COMB}, undershoots the true speed of path integration consistently by a greater amount than the model incorporating axonal conduction delays. A possible explanation for this is that the model incorporating axonal conduction delays includes a time delay Δt in the Hebbian associative learning rules for updating the w^2 and w^3 synaptic weights, whereas the current model does not. The presence of the time delay Δt in the learning rules of the model incorporating axonal conduction delays may facilitate more accurate path integration of head direction by encouraging the specificity in the synaptic weight structure that is required to update a packet of head direction cell activity at the correct speed. That is, in the model incorporating axonal conduction delays, the w^2 and w^3 synaptic weights converge more readily to a stable specific state. The current model can still perform path integration of head direction with the learning rates k^2 and k^3 set to 0.01, whereas the model incorporating axonal conduction delays cannot. On the other hand, introducing either homosynaptic LTD or heterosynaptic LTD into the w^2 and w^3 learning rules severely degrades or abolishes the path integration of head direction in the current model, whereas the model incorporating axonal conduction delays can still perform path integration of head direction with homosynaptic LTD incorporated into the w^2 and w^3 learning rules.

The two models also utilize different operational mechanisms. The results of Experiment 7.2 demonstrate that, when the time constants τ^{HD} and τ^{COMB} are set to the same small value of 1ms then some other mechanism, e.g. axonal conduction delays, is required in order to produce path integration of head direction.

The accuracy of the path integration may be improved by simulating the model with integrate-and-fire neurons. It may also be that the w^1_{ij} excitatory recurrent collateral synapses provide a source of friction that damps the speed of update of the head direction cell activity packet. It is possible to implement a continuous attractor neural without recurrent collateral synapses, and doing so may also improve the accuracy of path integration. Incorporating more realistic angular head velocity cells into the current model will allow the model to generalize across different speeds of path integration, which cannot currently be achieved.

As was discussed in Section 6.5 of Chapter 6, there is no immediately obvious computational reason why the current model consistently undershoots the correct speed of path integration, apart from the lack of convergence in the w^2 and w^3 synaptic weights as stated above. Future simulation work is required in order to identify the factors that have the most influence upon the accuracy of path integration.

As per Section 6.5, a similar question can also be asked about the validity of the current model given the consistent undershoot of the speed of path integration. A similar answer can also be provided.

There is no computational reason why the model cannot learn to perform path integration at the correct speed. The current model does undershoot the correct speed, and performs worse than the model reported in Chapter 6 for reasons stated above, but taken together with the results in Chapter 6 these two models provide the first steps in addressing two different mechanisms by which a model can learn, in a biologically plausible manner, to perform path integration at the correct speed. These results can be improved upon with further research. Thus, the consistent undershoot of the correct speed of path integration that is exhibited by the current model is not a fatal flaw.

In conclusion, a model that uses the value of the neuronal time constant τ^{COMB} to produce an effective time delay can learn to perform path integration of head direction at approximately the speed that was experienced during training.

Chapter 8

Conclusion

Within the rat brain, there are classes of cells that signal spatial information. Place cells signal the location of the rat within its environment (O'Keefe and Dostrovsky, 1971; O'Keefe and Nadel, 1978). Entorhinal cortex grid cells also signal the location of the rat within its environment, but each grid cell has a repeating receptive field that tessellates into a hexagonal grid-like pattern (Hafting et al., 2005; Sargolini et al., 2006). Head direction cells signal the direction that the head of the rat is facing towards (Ranck, 1985; Taube et al., 1990a,b).

The experiments reported in this thesis have explored the computational neuroscience of head direction cells. Neural network models, that self-organize through learning, were employed as the research methodology. These neural network models were constructed as the minimal synaptic architectures required to produce the desired learned behaviour in each experiment. For each neural network model, the minimal synaptic architecture facilitated a detailed investigation into the effect that controlled alterations of the model parameters, and training environments, have upon the learned behaviour of the model.

The research conducted in this thesis has investigated two computational properties of the head direction cell system. Firstly, the development, through learning in the presence of visual input, of well-established head direction cell response properties in response to visual cues has been investigated. Second, the development, through learning, of the ability of the head direction cell system to update its current representation of head direction using internal idiothetic (self-motion) signals in the absence of visual input, i.e. path integration of head direction, was investigated.

8.1 The Development of Head Direction Cell Visual Responses

Individual head direction cells exhibit similar response properties. In particular, each head direction cell has an approximately Gaussian response profile centred on a unique preferred head direction (Ranck, 1985; Taube et al., 1990a). The preferred firing direction of a head direction cell can be altered by the manipulation of a visual cue or cues: in the paradigmatic experiment, this cue is a single, visually-distinctive, cue card (Taube et al., 1990b; Goodridge and Taube, 1995).

Different visual cues can differ in their salience with respect to the head direction cell system. Previous research has shown that distal visual cues have a stronger influence upon the preferred firing directions of individual head direction cells than do proximal visual cues (Zugaro et al., 2001, 2004; Yoganarasimha et al., 2006).

This might be expected because only the visual input from distal visual cues will directly reflect the current head direction of an animal as it moves through different locations within its visual envi-

ronment. In contrast, individual proximal cues will appear to lie on different compass bearings as the animal moves through different locations within the environment.

An important question then is how head direction cells develop their sensitivity to the distal visual cues, and therefore head direction, given visual input from a natural environment containing a mixture of proximal and distal visual cues? The challenge is to explain how a subpopulation of head direction cells can learn to respond only to the distal visual cues, and ignore the proximal cues, in order to develop their sensitivity purely to the head direction of the animal.

The results reported in Chapters 2 and 3 show how a one-layer competitive neural network is able to form separate output representations of the distal visual cues, and a single proximal visual object, when trained on visual input from a training environment containing both of these kinds of cues. The network thus develops, through visually-guided learning, a subpopulation of output layer cells that are anchored to the distal rather than proximal visual cues. These output layer cells will then respond to preferred head directions, much like head direction cells in the rat brain.

The competitive network is able to achieve this result by exploiting the statistical decoupling that will exist between the visual input from the distal cues and the visual input from the proximal object as the agent visits different locations in the environment. It has been demonstrated that this statistical decoupling leads to different subpopulations of output layer cells learning to respond to either the distal cues or the proximal object, but not both. The subpopulation of output layer cells that learn to respond to the distal cues fire with an approximately Gaussian profile centred on individually preferred

head directions, and in this respect behave like head direction cells.

8.1.1 Learning to Discriminate between Distal and Proximal Visual Cues

A neural network model comprising a layer of input cells connected, by a set of feedforward synapses, to a layer of output cells, can learn, under the correct conditions, to form separate representations of distal and proximal visual cues that are presented simultaneously as input to the model.

Orthogonal Input Representations

Chapter 2 presents the results of experiments simulating this model with orthogonal input representations. When exposed to a training environment containing ten training locations, one proximal object, and a circular space of distal objects, the model is able to learn to form separate output layer cell representations of the distal and proximal object spaces. That is, after training, individual output layer cells respond exclusively to a narrow region of either the space of distal objects or the proximal object space, but not both.

Further experiments reported in Chapter 2 explored the effect upon the learned model behaviour of controlled alterations of the model and training environment parameters. An important result is that the simulated agent must visit a minimum number of training locations, ≥ 5, in the training environment in order for output layer cells to develop that respond exclusively to a narrow region of either the space of distal objects or the proximal object space, but not both. This results corresponds to the requirement that a rat must move through its environment in order to discriminate between distal and proximal visual cues.

The learned model performance was shown to be robust to the amount of training that the model receives, and performance is also robust to the size of the neural network. The learned model performance is dependent, however, upon the values of the learning rate k, and the sparseness a. In both cases, the model requires that these parameters be set to relatively small values if the model is to produce output layer cells that respond exclusively to one of the distal or proximal object spaces, but not both. When either of these parameters is set to a relatively large value, the model develops a distributed representation in the output layer. It is still possible to discriminate between the distal and proximal object spaces with a distributed output representation, but this type of output representation is not, however, the desired output representation of the model.

When the number of proximal objects in the training environment is increased to be ≥ 2, the output layer cells also develop a distributed representation.

Distributed Input Representations

Chapter 3 presents the results of experiments simulating a model with distributed input representations. This model was simulated to demonstrate that the general principle of using the statistical independence between the distal and proximal visual cues to aid the competitive learning of separate representations of these cues is robust with respect to a more distributed input representation. Similar to the results of Chapter 2, the model with distributed input representations can learn, under the correct conditions, to develop output layer cells that respond exclusively to a narrow region of either the space of distal objects or the proximal object space, but not both.

In simulation, the model with distributed input representations was shown to be robust to the per-

centage of overlap between the input layer cell subsets, i.e. the number of input cells shared between the subsets. It was also shown that a minimum number of training locations, ≥ 5, must be visited by the agent in order for output layer cells to develop responses that are exclusive to either the space of distal objects or the proximal object, but not both.

Similar to the results obtained for the model with orthogonal input representations, the model with distributed input representations requires relatively small values of the learning rate k and the sparseness a. Otherwise, a distributed representation of the input spaces develops in the output layer cells. The model performance is reasonably robust to the amount of training received, although the model with distributed input representations requires a higher number of training epochs to be experienced, compared to the model with orthogonal input representations, before behaviour is produced that approaches the learned behaviour of the fully-trained model. The model performance is also robust to the size of the model.

Simulating the model with ≥ 2 proximal objects in the training environment results in the model developing an output layer distributed representation of the distal and proximal object spaces.

The experiments reported in Chapter 3 are careful replications of the experiments reported in Chapter 2. This approach was employed in order to highlight the similarities and differences between the model with orthogonal input representations and the model with distributed input representations. In general, when implemented with a particular set of model parameters or a particular training environment, the two models produce qualitatively similar results. Therefore, using the statistical properties

of distal and proximal visual cues as an aid to a competitive learning mechanism that forms separate representations of these visual cues is a general principle that is robust to the distributed nature of the input cell representation.

Future Research

In both the model with orthogonal input representations and the model with distributed input representations, the visual input is represented in a highly idealized manner. That is, the individual input layer cells respond, with a Gaussian response profile, to narrow regions of either the space of distal objects, or a proximal object space. This abstract, idealized visual input was used to provide a high degree of control over the model so that the effect upon the learned model behaviour of altering particular model parameters could be clearly isolated.

Such idealized visual input is not, however, representative of either the complex visual scenes experienced by a rat, or the early stages of visual processing in the rat brain. A next step in the research into the development of head direction cell visual responses will be to embed the neural network model in a rich 3D visual training environment. This type of training environment can be created using 3D modelling software such as Blender,[1] or a low-level graphics programming library such as OpenGL.[2] The visual input from these 3D training environments can be processed with Gabor filters, which simulate the edge and bar detection cells in the early mammalian visual system (Jones and Palmer, 1987).

The advantage of using richer visual training environments is not only an added degree of realism,

[1] http://www.blender.org
[2] http://www.opengl.org

but also the fact that it will be possible to replicate experimental setups from the single-cell recording literature. For example, the paradigmatic cue-card rotation within a cylinder experiment (Muller and Kubie, 1987; Taube et al., 1990b). Replication of standard experimental environments will allow for a closer correspondence between simulation results and experimental results. I have already conducted some preliminary simulations in which the neural network model was fed visual input from a 3D visual training environment containing distal and proximal visual cues. These simulations, however, proved too computationally expensive to pursue within the time constraints of a doctoral thesis.

Another simplification in the neural network models reported in Chapters 2 and 3 is that the simulated agent visits discrete training locations within the training environment and effectively jumps between the training locations. In a real environment, a rat will visit a continuum of training locations as it traces a route through its environment. A next step will be to allow the simulated agent to visit any possible location in the training environment. This can be achieved by either having the agent follow a preset path through the training environment, or by simulating the agent tracing a stochastic, random walk through its environment where the training locations the agent visits are decided probabilistically, but the agent can only visit training locations that are adjacent to its current location. This will produce a more realistic path of the agent through its environment. The random walk can be implemented first in the idealized visual training environment, and then in the rich 3D visual training environment. Allowing the agent to visit a continuum of training locations will increase the statistical decoupling between the distal and proximal object spaces, which could improve the learned behaviour of the model.

In the training protocols for the models reported in Chapters 2 and 3, the simulated agent rotates through 360^{o} at each training location in the training environment. It can be argued that this is not a particularly realistic training regime, and that a more realistic protocol would be to have the agent follow a path through the training environment as discussed above, but with the head of the agent remaining within a small interval of the direction the agent is walking. In the simplified case where the head of the agent *always* faces the direction the agent is walking, it is predicted that the output layer cells in the model will learn to form separate representations of the distal and proximal object spaces and may out-perform the models reported in Chapters 2 and 3.

This is because, if the agent follows a suitably long path, and its head always remains at the same directional heading whilst the agent traverses this path, the agent will constantly receive the same view of the space of distal objects, but will receive constantly changing views of the proximal objects as the agent walks towards and then passes these proximal objects. In the model, the input cells representing the view of the distal object space will always be active as the agent traverses the path, but different subsets of input layer cells representing different views of the proximal objects will activate and deactivate as the agent receives different views of the proximal objects. The output layer cells in the model will continuously strengthen the feedforward synapses they receive from the input layer cells representing the view of the space of distal objects, but will not equivalently consistently strengthen the feedforward synapses they receive from the input layer cells representing a single particular view of one of the proximal objects. For the reasons outlined in Chapter 1, this will have the computational effect that the output layer cells will learn to respond exclusively to a particular view of the space of distal objects, and will not respond to any of the proximal objects. Thus, a change in

the training protocol – allowing the agent to maintain the same directional heading whilst following a path – is predicted to significantly improve the ability of the model to discriminate between the distal and proximal object spaces.

An interesting question that deserves further research is the maturity of the head direction cell responses in young rats. The research of Langston et al. (2010) and Wills et al. (2010) suggests that head direction cell response profiles approach their mature state almost immediately after the infant rat opens its eyes. That is, the head direction cells display single, approximately Gaussian, response profiles that are centred on a single preferred firing direction, and which remain stable across repeat visits to a particular environment. These results raise two further important questions. Is the head direction cell system genetically predetermined, i.e. hard-wired? And, do the head direction cells in infant rats exhibit the full range of response properties that are displayed by head direction cells in adult rats?

In the studies of Langston et al. (2010) and Wills et al. (2010), the amount of visual experience undergone by the infant rats was not controlled for. Thus, it is hard to dissociate whether the head direction cell responses are the result of a genetically predetermined head direction cell system, or visual training. Certainly, the results reported in Chapters 2 and 3 of this thesis demonstrate that, in order to exhibit mature head direction cell responses, the required visual learning can be accomplished within minutes of exposure to visual input.

Research by Rudy et al. (1987) demonstrated that young rats can see distal visual cues, but cannot process these cues as distal visual cues in a spatial configuration with proximal visual cues until the

rat becomes more mature. This result suggests that the full range of head direction cell response properties only develops after the rat has further experience of exploring environments beyond the initial opening of the eyes. Thus, the full range of head direction cell responses is not hard-wired, and requires further learning in order to fully develop.

It would be very interesting to test the full range of the head direction cell response properties in young rats. Recording from head direction cells in rats that are the same age as the rats in the studies of Langston et al. (2010) and Wills et al. (2010), the infant rat could be tested with the paradigmatic cue-card rotation experiment (Muller and Kubie, 1987; Taube et al., 1990b), and with the distal and proximal cue manipulation experiment of Zugaro et al. (2001). By carefully controlling the amount of visual experience the infant rat undergoes, it should be possible to dissociate whether the head direction cell responses in young rats are the result of genetic pre-specification or visual learning. Moreover, it will be possible to test the exact maturity of the head direction cell responses. It is predicted that further learning is required in order for head direction cells in young rats to show full maturity in their responses, and therefore it will not be possible to replicate the results of Taube et al. and Zugaro et al..

A problem with the models reported in Chapters 2 and 3 is that increasing the number of proximal objects in the training environment degrades the learned model behaviour: the model produces a distributed representation of the distal and proximal object spaces in the output layer. The original hypothesis, however, was that the model would produce output layer cells that had learned to respond exclusively to a narrow region of one of the distal or proximal object spaces. The future research

steps outlined so far may help to improve this aspect of the learned model behaviour when there are multiple proximal objects in the training environment.

For instance, allowing the agent to visit a continuum of training locations, in a more realistic 3D visual training environment, may also increase the statistical decoupling between the distal and proximal object spaces for the same reasons that increasing the number of discrete training locations increases the statistical decoupling between the distal and proximal object spaces. It may also be the case that including internal idiothetic (self-motion) signals in the model may improve the learned model performance. The reasons for this will be discussed in Section 8.3 of this chapter. All of these suggestions will be a next research step.

8.1.2 The Isomapping of Preferred Firing Directions

Within a population of head direction cells, the angular distance between the preferred firing directions of any two head direction cells is maintained across different environments (Taube et al., 1990b; Taube and Burton, 1995; Zugaro et al., 2004; Yoganarasimha et al., 2006). This is defined as an isomapping of head direction cell preferred firing directions across environments.

In the neural network model, a layer of input cells is connected, through feedforward synapses, to a layer of output cells, which is, in turn, connected to itself through excitatory recurrent collateral synapses. The model can learn to produce an isomapping of the output layer cell preferred firing directions across two visual training environments.

The isomapping is produced because output layer cells that, through competitive learning in the first

training environment, develop preferred firing directions close to one another in terms of angular distance, will also develop strong excitatory recurrent collateral synapses with one another.

In the second training environment, the firing of the input layer cells will initially lead, through the initially untrained feedforward synapses, to a random assortment of the output layer cells firing. The strong excitatory recurrent collateral synapses will then help to force the firing of a cluster of output layer cells that have preferred firing directions close to one another in the first training environment. Hebbian learning will strengthen the feedforward synaptic connections between the currently firing input layer cells, representing orientations of the agent that are nearby in the second environment, and the currently firing output layer cells, which will come to represent these orientations. Thus, an isomapping of output layer cell preferred firing directions is learned across two environments.

The results of the experiments reported in Chapter 4 show that nature of the mapping of output layer cell preferred firing directions is dependent upon strong excitatory recurrent collateral synaptic input within the output layer. Weak, or non-existent, excitatory recurrent collateral input produces a complex remapping, rather than isomapping, of output layer cell preferred firing directions. The development of an isomapping of preferred firing directions is robust to the amount of training received by the model, and is also robust to the size of the model. The isomapping is, however, dependent upon some form of long-term depression being present in the learning rules.

Future Research

The simulation results of the isomapping model demonstrate that the head direction cell system can learn a behaviour such as isomapping on the basis of visual input alone. To this end, it would be very

interesting to embed the current isomapping model within the type of 3D visual training environment described above and then examine the influence that the richer visual input has upon the learned performance of the model.

Another next step will be to incorporate proximal visual objects into the training environment and explore the type of mapping of output layer cell preferred firing directions that the model learns to produce when proximal objects are present. Given that the real head direction cell system can produce an isomapping of head direction cell preferred firing directions in environments containing both distal and proximal objects, it is clear that producing an isomapping in a more complex environment is possible. The richer visual input will allow for a full exploration of the behaviour of the model.

It is predicted that the excitatory recurrent collateral synapses will help the isomapping model to effectively "wash-out" the proximal objects such that the model will only learn to represent, and form an isomapping between, the distal objects in an environment containing both distal and proximal objects. This is because the firing rates received by each output layer cell through the excitatory recurrent collateral synapses will only be coherent with the firing rates received by each output layer cell from the input layer cells representing the space of distal objects. That is, the firing rates of the input layer cells representing particular views of the proximal objects will change too often to establish any sort of stable relationship with the firing rates received through the excitatory recurrent collateral synapses, where the firing rates received through the recurrent collateral synapses will reflect the most stable input to the output layer cells, i.e. the input from the input layer cells representing the space of distal objects.

It is only when there is a stable and consistent relationship between the firing rates of a particular presynaptic cell and a particular postsynaptic cell that the synapse between these cells can be strengthened through Hebbian learning. Thus, the constant firing of a particular subset of input layer cells, which represent a particular view of the space of distal objects, will promote strengthening of the excitatory recurrent collateral synapses with the output layer cells because these recurrent collaterals will relay the stable firing from the input layer. The strengthening of the excitatory recurrent collateral synapses will, in turn, promote strengthening of the feedforward synapses between the input layer cells representing a particular view of the space of distal objects and the output layer cells. Thus, the output layer cells will learn to represent only the distal objects – and will only learn an isomapping between the distal objects – when the training environment contains both distal and proximal objects. It is further predicted that if the training protocol is altered to incorporate the agent maintaining a fixed directional heading as it traverses a path, as discussed in Section 8.1.1, then the model will be better able to form representations of only the distal objects.

It will also be interesting to simulate the agent visiting more than two training environments. Given that the model has demonstrated isomapping across two training environments, the model should, in principle, be able to demonstrate isomapping across multiple environments. The multiple training environments can initially be the same type of idealized environment as simulated in Chapter 4. It will then be possible to test the model in multiple 3D visual training environments.

8.2 The Path Integration of Head Direction

Head direction cells can track the current head direction of the rat using internal idiothetic (self-motion) signals in the absence of visual input (Goodridge et al., 1998). This is the path integration of head direction. Previous research has shown that, with the appropriate driving input, a continuous attractor neural network model of head direction cells can perform path integration of head direction (e.g. Skaggs et al., 1995; Zhang, 1996; Song and Wang, 2005; Stringer and Rolls, 2006). However, no previous research has addressed how a model may learn, in a biologically plausible manner, to update the internal representation of head direction at the same speed as the rat is rotating its head.

Axonal Conduction Delays

A continuous attractor neural network of head direction cells is reciprocally linked with a layer of combination cells, which also receive input from a layer of rotational velocity cells. Incorporating axonal conduction delays to provide an effective time delay, this model can learn to perform path integration of head direction, in the absence of visual input, at approximately the speed imposed during training in the presence of visual input

The results of the experiments reported in Chapter 6 provide a detailed exploration of the properties of this neural network model.

The model can learn to perform path integration of head direction, at approximately the speed experienced during training, across two different values for the axonal conduction delays and for two rotational velocities imposed during training in the light. Importantly, the model performance is ro-

bust to a uniform distribution of axonal conduction delays within the w^2 and w^3 synapses. The model also functions correctly with axonal conduction delays in either the w^2 or the w^3 synapses, but not both sets of synapses.

The model requires a high value for the learning rate k, and can function with homosynaptic LTD incorporated into the learning rule, but cannot function with heterosynaptic LTD incorporated into the learning rule. Permitting learning to occur during the testing phase of the model abolishes the learned path integration.

Neuronal Time Constants

A neural network model, with the same general architecture as the model described in Chapter 6, employs a large value for the neuronal time constant τ^{COMB} and a small value for the neuronal time constant τ^{HD}. This neural network model can also learn to perform path integration of head direction, in the absence of visual input, at approximately the speed imposed during training in the presence of visual input. This model does not incorporate axonal conduction delays.

The experiments reported in Chapter 7 are a careful replication of the experiments conducted in Chapter 6. This approach highlights the fact that exploiting natural time delays is a general mechanism that facilitates learning the path integration of head direction, in the absence of visual input, at approximately the speed experienced during training in the presence of visual input. This approach also allows a comparison of the similarities and differences between the model reported in Chapter 6 and the model reported in Chapter 7.

The neural network model reported in Chapter 7 can learn to perform path integration of head direction at approximately the speed experienced during training in the light, across two different values of the neuronal time constant τ^{COMB}, and at two different rotational velocities imposed during training in the light.

If the model is simulated with the same small values for the time constants τ^{HD} and τ^{COMB}, then the model does not learn to perform path integration of head direction. This result highlights the fact that another timing mechanism, i.e. axonal conduction delays, is required in order for a model simulated with small neuronal time constants to learn path integration of head direction. Thus, the model reported in Chapter 6 employs a different timing mechanism to the model reported in Chapter 7.

The model is robust to a uniform distribution for the values of τ^{COMB}. The model does not learn to perform path integration accurately when either homosynaptic LTD, or heterosynaptic LTD, is incorporated in the learning rules.

Future Research

Both the model incorporating axonal conduction delays, Chapter 6, and the model using large neuronal time constants τ^{COMB}, Chapter 7, consistently undershoot, during path integration in the absence of visual input, the rotational velocity imposed during training in the light. The model using large neuronal time constants τ^{COMB} consistently undershoots this value by more than the model incorporating axonal conduction delays.

A next step will be to simulate both models with integrate-and-fire neurons. Previous research has

shown that a neural network model of head direction cells can update the cell response profiles within tens of milliseconds of the presentation of a new stimulus (Degris et al., 2004). The fast response of integrate-and-fire neurons to a stimulus may ameliorate the undershoot of the recorded speed of path integration in the absence of visual input, compared to the imposed speed of path integration during training in the presence of visual input.

A further advantage of simulating the models in Chapters 6 and 7 with integrate-and-fire neurons is that this type of neural modelling approach allows for the incorporation of spike-time dependent plasticity (STDP) (Markram et al., 1997; Bi and Poo, 1998). STDP requires a presynaptic action potential, or spike, to occur within a temporal window of a few tens of milliseconds before a postsynaptic action potential in order for long-term potentiation of the synapse connecting the two cells to occur. The temporal precision required for learning to occur under conditions of STDP may also improve the learned performance, in terms of the speed of head direction cell activity packet update, for the models reported in Chapters 6 and 7.

It is predicted that the inclusion of an STDP mechanism into the models of path integration will produce more accurate path integration because the accuracy of the path integration mechanism is dependent upon associating the firing a particular head direction cell at a time t with the firing of another head direction cell at a time $t + \Delta t$. If the head direction cell firing at time t forms a strong synaptic connection with a head direction cell that commences firing either before or after the precise time interval Δt, then the accuracy of the path integration mechanism will be degraded. STDP will help produce an accurate path integration mechanism because a synapse is only strengthened if a presynap-

tic spike occurs within a precise time interval before the postsynaptic spike. As the association to be learned between a head direction cell with a preferred firing direction θ_1 firing at a time t, and another head direction cell with a preferred firing direction θ_2 firing at a time $t + \Delta t$, is strongly regulated by the natural timing mechanisms of either axonal conduction delays or neuronal time constants, the STDP mechanism should ensure high specificity in terms of strengthening the correct synapses to produce accurate path integration.

In the two models of path integration, rotational velocity cells provide a signal of a fixed speed of rotation in either the clockwise or counter-clockwise direction. This was a simplifying assumption that allowed the investigation into the model performance to focus upon the nature of the timing mechanism. In the real rat brain, however, angular velocity cells in the dorsal tegmental nucleus and lateral mammillary nuclei exhibit approximately linear response functions as the angular velocity of the head increases (Bassett and Taube, 2001b, 2005; Sharp, 2005). The response functions of these cells plateau as the angular velocity of the head increases further. Some of these angular velocity cells respond in a symmetric manner to both clockwise and counter-clockwise directions of rotation, whereas other angular velocity cells respond in an asymmetric manner to either clockwise rotation or counter-clockwise rotation, but not both.

A next step will be to incorporate these biologically more realistic angular velocity cells into both of the models reported in Chapters 6 and 7. In order for these path integration models to continue to function, the combination cells must still learn to respond to combinations of a particular head direction and angular head velocity. If the angular velocity cells possess response functions that are

variable with different gradients then the population firing vector of angular velocity cells will point in a unique direction for each unique angular velocity. A layer of combination cells incorporating competitive learning may thus be able to learn to respond to these unique population firing vectors such that individual combination cells will still learn to respond to combinations of a particular head direction and an angular head velocity.

In the models presented in Chapters 6 and 7, the agent is trained under conditions of continuous motion for both clockwise and counter-clockwise rotation. It can be argued that this is not particularly realistic as a training protocol, and an interesting question is what would the effect be upon the performance of the model if the agent is not trained under conditions of continuous motion? In the case where the agent performs a series of small rotations in either a clockwise or counter-clockwise direction, but never rotates fully through 360^{o} in either direction, then it is predicted that the model will still learn to perform path integration *as long as* the series of small rotations takes the agent through all head directions from $0^{\text{o}} - 360^{\text{o}}$. In other words, the models presented in Chapters 6 and 7 will not be able to generalize across portions of the 360^{o} head-direction space that the agent has not visited, but the computational principles by which the two models work do not require the agent to visit all of the 360 head direction in one continuous motion.

To see that this is the case, consider the situation in which the agent makes a continuous rotation through 360^{o} in either the clockwise or counter-clockwise direction. Accurate path integration requires the model to form strong synaptic connections between the firing of a head direction cell with a preferred firing direction θ_1 at a time t, and the firing of a head direction cell with a preferred firing

direction θ_2 at a time $t + \Delta t$. In the case of a continuous rotation through 360°, the head direction cell with a preferred firing direction θ_2 that fires at a time $t + \Delta t$ must then form a strong synaptic connection with a head direction cell with a preferred firing direction θ_3 at a time $t + 2\Delta t$. In the case of non-continuous rotations, an association must still be learned between the head direction cell with the preferred firing direction θ_2 and the head direction cell with the preferred firing direction θ_3, but it is now the case that the head direction cell with the preferred firing direction θ_2 fires at a time t, and the head direction cell with the preferred firing direction θ_3 now fires at a time $t + \Delta t$. For training under both continuous and non-continuous motion, the association between the head direction cells that is to be learned is the same, but the times at which the head direction cell with the preferred firing direction θ_2 fires is allowed to differ. This is why, provided all the required associations between the head direction cells are learned, the model can cope with training under conditions of non-continuous rotation. If, however, not all associations are learned then the models presented in Chapters 6 and 7 will not generalize to unseen associations, and the model will not correctly perform the path integration of head direction.

It may be possible to use the oscillations in the activity of a population of cells as a timing mechanism that introduces an effective time delay (Traub et al., 1999; Buzsáki, 2006). For instance, gamma oscillations (30-100 Hz) may provide a clock or timing mechanism that can act as a reference for the temporal order of neural activity (Sejnowski and Paulsen, 2006; O'Keefe, 2007). If both the head direction cell layer, and the combination cell layer, sustain population oscillatory firing behaviour, potentially out of phase with one another, then it may be possible to learn associations through time between successive head directions by associating the head directions with peaks in the oscillatory

firing behaviour. It may also be possible for one of the head direction cell or combination cell layers to oscillate at a slower theta frequency (6-12 Hz), while the other layer oscillates at the faster gamma frequency. Associations can then be learned across the gamma sub-cycles within a single theta cycle (Jensen and Lisman, 1996b,a; Lisman, 1999). Previous research has suggested that population oscillations play a role in facilitating successful path integration (Buzsáki, 2005; Blair et al., 2008; Hasselmo, 2008; Hasselmo and Brandon, 2008).

8.3 A Unified Model of the Head Direction Cell System

Bassett and Taube (2005) have suggested that the head direction cell system can be broadly classified into two separate, but convergent, processing streams.

The "ascending" processing stream carries idothetic, primarily vestibular, signals. The stream follows an approximate path of medial vestibular nucleus $\rightarrow$ nucleus prepositus hypoglossi $\rightarrow$ dorsal tegmental nucleus $\rightarrow$ lateral mammillary nuclei $\rightarrow$ anterodorsal thalamic nucleus $\rightarrow$ postsubiculum.

The "descending" processing stream carries visual signals, and follows an approximate path of either visual cortex $\rightarrow$ postsubiculum, or visual cortex$\rightarrow$ retrosplenial cortex $\rightarrow$ postsubiculum. The ascending and descending processing streams are shown schematically in Figure 8.1a.

The models reported in Chapters 2, 3, and 4 of this thesis are representative of the type of processing occurring in the descending visual stream of the head direction cell system. The models reported in Chapters 6 and 7 of this thesis are representative of the type of processing that occurs in the ascending

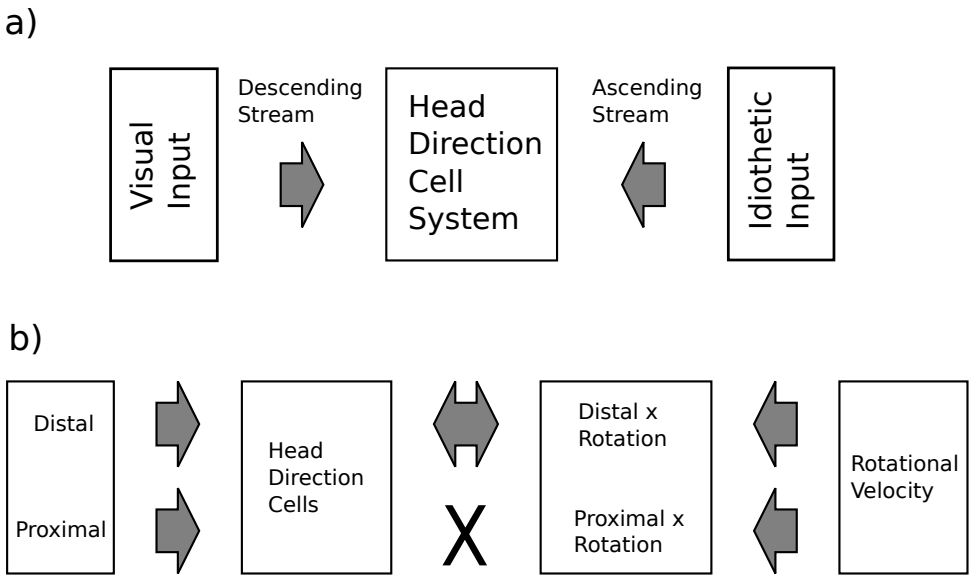

Figure 8.1: a) Ascending and descending input streams to the head direction cell system as proposed by Bassett and Taube (2005). The ascending stream provides idiothetic input, such as self-motion signals, to the head direction cell system. The descending stream provides visual input to the head direction cell system. b) A unified model of the head direction cell system. The left-hand side of the model provides visual input to the head direction cell system, and is hypothesized to involve the same type of computation as the models reported in Chapters 2 and 3, i.e. the computation can form separate representations of distal and proximal visual cues. The right-hand side of the model is a system that can achieve path integration of head direction as reported in Chapters 6 and 7. Only the distal visual cues will maintain a stable and consistent relationship with the rotational velocity inputs, and therefore the only learned association that will develop in the head direction cells is an association between visual input from distal cues and idiothetic input from the rotational velocity cells. Thus, the head direction cells will show a strong learned preference for stimulation by distal visual cues.

idiothetic stream of the head direction cell system.

A next step will be to incorporate the two processing streams to produce a unified model of the head direction cell system. This model will self-organize to produce the known responses of head direction cells in response to both visual and idiothetic signals. A schematic representation of the unified model is shown in Figure 8.1b.

The principle of operation for the unified model of the head direction cell system can be understood by considering the case of a single, untrained, head direction cell. Throughout learning, this head

direction cell will receive visual input from cells that can discriminate distal and proximal visual cues, shown as the left-hand side box of Figure 8.1b. The single head direction cell will also constantly receive idiothetic inputs. A layer of rotational velocity cells will signal the clockwise and counter-clockwise rotation of the animal. This layer of cells is represented by the right-hand side box in Figure 8.1b. The layer of rotational velocity cells will project onto a second layer of cells, which are reciprocally connected with a layer of head direction cells. This second layer of cells is represented by the middle-right box in Figure 8.1b. Thus, the single head direction cell under consideration will continuously receive direct input from the cells in the visual input stream and the second layer of cells in the idiothetic input stream, and indirect input from the layer of rotational velocity cells in the idiothetic input stream. The head direction will, in turn, project directly onto the second layer of cells in the idiothetic input stream.

The visual inputs will provide constant stimulation to the head direction cell. The head direction cell will, in turn, provide constant stimulation to, and receive constant stimulation from, the second layer of cells in the idiothetic input stream. The second layer of cells will also receive constant stimulation from the layer of rotational velocity cells. As the animal moves through its environment, the visual and idiothetic inputs to the head direction cell will change according to both the current rotational velocity, and the current head direction θ, of the animal.

In the second layer of cells in the idiothetic input stream, individual cells will initially represent combinations of the current rotational velocity of the animal and head direction cells that are stimulated by either distal or proximal visual cues. However, the only *consistent* visual input that the example head

direction cell will receive is from cells that represent distal visual cues. That is, the head direction cell will still receive constant stimulation from cells representing proximal visual cues but, amongst this subset of visual input cells, different cells will be active – and thus provide stimulation to the head direction cell – for a particular head direction θ at different locations in the environment. In comparison, amongst the subset of cells representing distal visual cues, the same cells will be active and provide stimulation to the head direction cell for a particular head direction θ at different locations in the environment.

Thus, as the rat moves through its environment, the only consistent relationship between the visual and idiothetic inputs to the head direction cell system is between those cells representing the distal visual cues and the rotational velocity cells. Consequently, through Hebbian associative learning, cells will develop that represent combinations of distal cue visual input and rotational velocity. No cells will develop that represent combinations of proximal cue visual input and rotational velocity. Preliminary simulation results, not shown, suggest that incorporating a vestibular signal into the orthogonal model reported in Chapter 2 improves the performance of the model to the extent that, in a standard training environment containing ten training locations and one proximal object, the grand majority of the output layer cells learn to respond exclusively to a narrow region of the space of distal objects. Hardly any output layer cells learn to respond to the proximal object space, or to both of the distal and proximal object spaces.

Because the second layer of "combination" cells projects onto the head direction cells, a functional consequence of the learning described above is that a head direction cell system will develop where

the individual head direction cells will show strong preferences for stimulation by distal visual cues, can maintain their preferred firing directions in the absence of visual input, and the system as a whole can perform path integration of head direction at the same speed as the animal is moving its head. The model will also develop an isomapping of the head direction cell preferred firing directions across environments. The research presented in this thesis has shown how the individual parts of this unified model can develop their response properties.

This unified model will also be a first step towards producing a self-organizing neural network model of the head direction cell system that closely corresponds to the known physiology of the head direction cell system, both in terms of the brain areas known to contain head direction cells and the responses of the head direction cells in these different brain areas. For instance, the origin of the head direction cell signal, and the location of the putative head direction cell continuous attractor network, is argued to be in the reciprocal loop formed between the dorsal tegmental nucleus and the lateral mammillary nuclei (Blair et al., 1998; Bassett and Taube, 2001a; Song and Wang, 2005). Explicitly modelling the head direction cell continuous attractor network as located in this reciprocal loop may then aid in generating biologically realistic firing properties in the head direction cells located in brain areas downstream from the continuous attractor.

For example, the firing rates of the head direction cells located in brain areas at the beginning of the ascending vestibular stream anticipate the head direction of the animal. That is, the firing profiles of these head direction cells are highly correlated with the head direction of the rat at some t in the future. The firing profiles of head direction cells in the anterodorsal nucleus of the thalamus anticipate

the head direction of the rat by approximately 23ms (Blair and Sharp, 1995), and the firing profiles of head direction cells in the lateral mammillary nuclei anticipate head direction by approximately 38ms (Blair et al., 1998). A full biophysically accurate and self-organizing model of the head direction cell system would permit a detailed investigation of how the head direction cells in different brain regions acquire their known response properties. This unified model, if embedded in a 3D visual training environment, will also permit the replication of the recording protocol and environment in single-cell recording studies, which will hopefully produce a two-way interaction between computational research and experimental research into the head direction cell system.

8.4 Final Remarks

In summary, this thesis has explored how neural network models may learn to produce particular well-known properties of the head direction cell system. Two models, with orthogonal and distributed input representations respectively, have shown that it is possible, on the basis of visual input alone, to learn to form separate output representations of distal and proximal visual cues that are presented simultaneously to the model as input. This type of computation would be required in order for head direction cells to preferentially respond to the manipulation of distal visual cues. However, so far, these models are only able to form separate output representations when there is a single proximal object present in the training environment. A third neural network model demonstrated how, on the basis of visual input alone, a complex property such as the isomapping of preferred firing directions can be learned. This thesis also explored the influence of idiothetic signals upon the head direction cell system. Two models of path integration, using axonal conduction delays and large neuronal time constants τ^{COMB} respectively, produce a natural time delay that enables the network to learn to

perform path integration of head direction at approximately the speed imposed during training in the light. Future research, including incorporating the models reported in this thesis into a unified model of the head direction cell system, was also considered.

Bibliography

Allen, G. V. and Hopkins, D. A. (1988). Mammillary body in the rat: a cytoarchitectonic, golgi, and ultrastructural study. *Journal of Comparative Neurology*, 275:39–64.

Allen, G. V. and Hopkins, D. A. (1989). Mammillary body in the rat: topography and synaptology of projections from the subicular complex, prefrontal cortex, and midbrain tegmentum. *Journal of Comparative Neurology*, 286:311–336.

Allen, G. V. and Hopkins, D. A. (1990). Topography and synaptology of mammillary body projections to the mesencephalon and pons in the rat. *Journal of Comparative Neurology*, 301:214–231.

Alyan, S. and Jander, R. (1994). Short-range homing in the house mouse, mus musculus: stages in the learning of directions. *Animal Behaviour*, 48:285–298.

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87.

Bassett, J. P. and Taube, J. S. (2001a). Lesions of the dorsal tegmental nucleus of the rat disrupt head direction cell activity in the anterior thalamus. *Society of Neuroscience Abstracts*, 25:852.29.

Bassett, J. P. and Taube, J. S. (2001b). Neural correlates for angular head velocity in the rat dorsal tegmental nucleus. *Journal of Neuroscience*, 21:5740–5751.

Bassett, J. P. and Taube, J. S. (2005). Head direction signal generation: Ascending and descending information streams. In Wiener, S. I. and Taube, J. S., editors, *Head Direction Cells and the Neural Mechanisms of Spatial Orientation*, pages 83–109. MIT Press, Cambridge, MA.

Bennett, J. and Stringer, S. M. (2011). Using independent motion to separate representations of objects that are always seen together. In preparation.

Bi, G. G. and Poo, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18:10464–10472.

Blair, H. T., Cho, J., and Sharp, P. E. (1998). Role of the lateral mammillary nucleus in the rat head direction circuit: a combined single unit recording and lesion study. *Neuron*, 21:1387–1397.

Blair, H. T., Gupta, K., and Zhang, K. (2008). Conversion of a phase- to a rate-coded position signal by a three-stage model of theta cells, grid cells, and place cells. *Hippocampus*, 18:1239–1255.

Blair, H. T. and Sharp, P. E. (1995). Anticipatory head-direction cells in anterior thalamus: Evidence for a thalamocortical circuit that integrates angular head motion to compute head direction. *Journal of Neuroscience*, 15:6260–6270.

Bostock, E. M., Muller, R. U., and Kubie, J. L. (1991). Experience-dependent modifications of hippocampal place cell firing. *Hippocampus*, 1:193–205.

Boucheny, C., Brunel, N., and Arleo, A. (2005). A continuous attractor network model without recurrent excitation: maintenance and integration in the head direction cell system. *Journal of Computational Neuroscience*, 18:205–227.

Buzsáki, G. (2005). Theta rhythm of navigation: Link between path integration and landmark nvaigation, episodic and semantic memory. *Hippocampus*, 15:827–840.

Buzsáki, G. (2006). *Rhythms of the Brain*. Oxford University Press, Oxford.

Carpenter, G. A. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Visiion, Graphics, and Image Processing*, 37:54–115.

Carpenter, G. A. and Grossberg, S. (1988). The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, March 1988:77–88.

Carr, C. E. (1993). Processing of temporal information in the brain. *Annual Review of Neuroscience*, 16:223–243.

Carr, C. E. and Konishi, M. (1988). Axonal delay lines for time measurement in the owl's brainstem. *Proceedings of the National Academy of Sciences of the USA*, 85:8311–8315.

Carr, C. E. and Konishi, M. (1990). A circuit for detection of interaural time differences in the brainstem of the barn owl. *Journal of Neuroscience*, 10:3227–32246.

Chen, L. L., Lin, L. H., Green, E. J., Barnes, C. A., and McNaughton, B. L. (1994). Head-direction cells in the rat posterior cortex. I. Anatomical distribution and behavioral modulation. *Experimental Brain Research*, 101:8–23.

Cho, J. and Sharp, P. E. (2001). Head direction, place, and movement correlates for cells in the rat retrosplenial cortex. *Behavioral Neuroscience*, 115:3–25.

Collett, T. S. and Zeil, J. (1998). Places and landmarks: An arthropod perspective. In Healy, S., editor, *Spatial Representation in Animals*. Oxford University Press, Oxford.

Cressant, A., Muller, R. U., and Poucet, B. (1997). Failure of centrally placed objects to control the firing fields of hippocampal place cells. *Journal of Neuroscience*, 17:2531–2542.

Dayan, P. and Abbott, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA.

Degris, T., Brunel, N., Sigaud, O., and Arleo, A. (2004). Rapid response of head direction cells to reorienting visual cues: A computational model. *Neurocomputing*, 58–60c:675–682.

Etienne, A. S., Berlie, J., Georgakopoulos, J., and Maurer, R. (1998). Role of dead reckoning in navigation. In Healy, S., editor, *Spatial Representation in Animals*. Oxford University Press, Oxford.

Etienne, A. S. and Jeffery, K. J. (2004). Path integration in mammals. *Hippocampus*, 14:180–192.

Etienne, A. S., Joris-Lambert, S., Reverdin, B., and Téroni, E. (1993). Learning to recalibrate the role of dead reckoning and visual cues in spatial navigation. *Animal Learning and Behavior*, 21:266–280.

Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78.

Girard, P., Hupé, J. M., and Bullier, J. (2001). Feedforward and feedback connections between areas V1 and V2 of the monkey have similar rapid conduction velocities. *Journal of Neurophysiology*, 85:1328–1331.

Goodridge, J. P., Dudchenko, P. A., Worboys, K. A., Golob, E. J., and Taube, J. S. (1998). Cue control and head direction cells. *Behavioral Neuroscience*, 112:749–761.

Goodridge, J. P. and Taube, J. S. (1995). Preferential use of the landmark navigational system by head direction cells in rats. *Behavioral Neuroscience*, 109:49–61.

Goodridge, J. P. and Taube, J. S. (1997). Interaction between the postsubiculum and anterior thalamus in the generation of head direction cell activity. *Journal of Neuroscience*, 17:9315–9330.

Goodridge, J. P. and Touretzky, D. S. (2000). Modeling atractor deformation in the rodent head-direction system. *Journal of Neurophysiology*, 83:3402–3410.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23–63.

Hafting, T., Fyhn, M., Molden, S., Moser, M. B., and Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436:801–806.

Hahnloser, R. H. R. (2003). Emergence of neural integration in the head-direction system by visual supervision. *Neuroscience*, 120:877–891.

Hasselmo, M. E. (2008). Temporally structured replay of neural activity in a model of entorhinal cortex, hippocampus and postsubiculum. *European Journal of Neuroscience*, 28:1301–1315.

Hasselmo, M. E. and Brandon, M. P. (2008). Linking cellular mechanisms to behavior: Entorhinal persistent spiking and membrane potential oscillations may underlie path integration, grid cell firing, and episodic memory. *Neural Plasticity*.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Addison Wesley, Wokingham, UK.

Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544.

Hopfield, J. J. (1982). Neural networks and systems with emergent selective computational abilities. *Proceedings of the National Academy of Sciences of the USA*, 79:2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the USA*, 81:3088–3092.

Jensen, O. and Lisman, J. E. (1996a). Hippocampal CA3 region predicts memory sequences: Accounting for the phase precession of place cells. *Learning and Memory*, 3:279–287.

Jensen, O. and Lisman, J. E. (1996b). Theta/gamma networks with slow NMDA channels learn sequences and encode episodic memory: Role of NMDA channels in recall. *Learning and Memory*, 3:264–278.

Jones, J. P. and Palmer, L. A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58:1233–1258.

Khabbaz, A., Feel, M. S., Tsien, J. Z., and Tank, D. W. (2000). A compact converging-electrode microdrive for recording head direction cells in mice. *Society of Neuroscience Abstracts*, 26:984.

Knierim, J. J., Kudrimoti, H. S., and McNaughton, B. L. (1995). Place cells, head direction cells, and the learning of landmark stability. *Journal of Neuroscience*, 15:1648–1659.

Knierim, J. J., Kudrimoti, H. S., and McNaughton, B. L. (1998). Interactions between idiothetic cues and external landmarks in the control of place cells and head direction cells. *Journal of Neurophysiology*, 80:425–446.

Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press, New York.

Koch, C., Rapp, M., and Segev, I. (1996). A brief history of time (constants). *Cerebral Cortex*, 6:93–101.

Konishi, M. (1991). Deciphering the brain's codes. *Neural Computation*, 3:1–18.

Langston, R. F., Ainge, J. A., Couey, J. J., Canto, C. B., Bjerknes, T. L., Witter, M. P., Moser, E. I., and Moser, M. B. (2010). Development of the spatial representation system in the rat. *Science*, 328:1576–1580.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitée comme une polarisation. *Journal de Physiologie et Pathologie Général*, 9:620–635.

Leutgeb, S., Ragozzino, K. E., and Mizumori, S. J. (2000). Convergence of head direction and place information in the CA1 region of the hippocampus. *Neuroscience*, 100:11–19.

Lisman, J. E. (1999). Relating hippocampal circuitry to function: Recall of memory sequences by reciprocal dentate-CA3 interactions. *Neuron*, 22:233–242.

Markram, H., Lubke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–215.

Markus, E. J., Qin, Y. L., Leonard, B., Skaggs, W., McNaughton, B. L., and Barnes, C. A. (1995). Interactions between location and task affect the spatial and directional firing of hippocampal neurons. *Journal of Neuroscience*, 15:7079–7094.

Martin, P. D. and Berthoz, A. (2002). Development of spatial firing in the hippocampus of young rats. *Hippocampus*, 12:465–480.

McNaughton, B. L., Chen, L. L., and Markus, E. J. (1991). Dead reckoning, landmark learning, and the sense of direction: A neurophysiological and computational hypothesis. *Journal of Cognitive Neuroscience*, 3:190–202.

Mittelstaedt, H. and Mittelstaedt, M. L. (1982). Homing by path integration. In Papi, F. and Wallraff, H. G., editors, *Avian Navigation*. Springer, Berlin.

Mittelstaedt, M. L. and Mittelstaedt, H. (1980). Homing by path integration in a mammal. *Naturwissenschaften*, 67:566–567.

Mizumori, S. J., Puryear, C. B., Gill, K. M., and Guazzelli, A. (2005). Head direction cells in hippocampal afferent and efferent systems: What functions do they serve? In Wiener, S. I. and Taube, J. S., editors, *Head Direction Cells and the Neural Mechanisms of Spatial Orientation*. MIT Press, Cambridge, MA.

Mizumori, S. J., Ragozzino, K. E., and Cooper, B. G. (2000). Location and head direction representation in the dorsal striatum of rats. *Psychobiology*, 28:441–462.

Mizumori, S. J. and Williams, J. D. (1993). Directionally selective mnemonic properties of neurons in the lateral dorsal nucleus of the thalamus of rats. *Journal of Neuroscience*, 13:4015–4028.

Morris, R. G. M. (1981). Spatial localization does not require the presence of local cues. *Learning and Motivation*, 12:239–260.

Muir, G. M. and Taube, J. S. (2002). Firing properties of head direction cells, place cells and theta cells in the freely-moving chinchilla. *Society of Neuroscience Abstracts*, 584:4.

Muller, R. U. and Kubie, J. L. (1987). The effects of changes in the environment on the spatial firing of hippocampal complex spike cells. *Journal of Neuroscience*, 7:1951–1968.

Oja, E. (1982). A simplified neuron model as a principal component analyser. *Journal of Mathematical Biology*, 15:267–273.

O'Keefe, J. (2007). Hippocampal neurophysiology in the behaving animal. In Andersen, P., Morris, R., Amaral, D., Bliss, T., and O'Keefe, J., editors, *The Hippocampus Book*. Oxford University Press, Oxford.

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely moving rats. *Brain Research*, 34:171–175.

O'Keefe, J. and Nadel, L. (1978). *The Hippocampus as a Cognitive Map*. Clarendon Press, Oxford.

Ranck, Jr., J. B. (1985). Head direction cells in the deep cell layer of dorsolateral presubiculum in freely moving rats. In Buzsáki, G. and Vanderwolf, C., editors, *Electrical Activity of the Archicortex*. Akadémiai Kiadó, Budapest.

Redish, A. D. (1999). *Beyond the Cognitive Map*. MIT Press, Cambridge, MA.

Redish, A. D., Elga, A. N., and Touretzky, D. S. (1996). A coupled attractor model of the rodent head direction system. *Network: Computation in Neural Systems*, 7:671–685.

Robertson, R. G., Rolls, E. T., and Georges-François, P. (1999). Head direction cells in the primate pre-subiculum. *Hippocampus*, 9:206–219.

Rolls, E. T. and Deco, G. (2002). *Computational Neuroscience of Vision*. Oxford University Press, Oxford.

Rolls, E. T. and Stringer, S. M. (2006). Invariant visual object recognition: A model, with lighting invariance. *Journal of Physiology, Paris*, 100:43–62.

Rolls, E. T. and Treves, A. (1990). The relative advantages of sparse versus distributed encoding for associative neuronal networks in the brain. *Network*, 1:407–421.

Rolls, E. T. and Treves, A. (1998). *Neural Networks and Brain Function*. Oxford University Press, Oxford.

Royer, S. and Paré, D. (2003). Conservation of total synaptic weight through balanced synaptic depression and potentiation. *Nature*, 422:518–522.

Rubin, J., Terman, D., and Chow, C. (2001). Localized bumps of activity sustained by inhibition in a two-layer thalamic network. *Journal of Computational Neuroscience*, 10:313–331.

Rudy, J. W., Stadler-Morris, S., and Albert, P. (1987). Ontogeny of spatial navigation behaviors in the rat: Dissociation of "proximal"- and "distal"- cue-based behaviors. *Behavioral Neuroscience*, 101:62–73.

Rumelhart, D. E. and Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science*, 9:75–112.

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B. L., Witter, M. P., Moser, M. B., and Moser, E. I. (2006). Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312:758–762.

Save, E., Cressant, A., Thinus-Blanc, C., and Poucet, B. (1998). Spatial firing of hippocampal place cells in blind rats. *Journal of Neuroscience*, 18:1818–1826.

Schenk, F. (1985). Development of place navigation in rats from weaning to puberty. *Behavioral and Neural Biology*, 43:69–85.

Séguinot, V., Maurer, R., and Etienne, A. S. (1993). Dead reckoning in a small mammal: The evaluation of distance. *Journal of Comparative Physiology A*, 173:103–113.

Sejnowski, T. J. and Paulsen, O. (2006). Network oscillations: Emerging computational principles. *Journal of Neuroscience*, 26:1673–1676.

Sharp, P. E. (2005). Regional distribution and variation in the firing properties of head direction cells. In Wiener, S. I. and Taube, J. S., editors, *Head Direction Cells and the Neural Mechanisms of Spatial Orientation*. MIT Press, Cambridge, MA.

Sharp, P. E., Blair, H. T., and Brown, M. (1996). Neural network modeling of the hippocampal formation spatial signals and their possible roles in navigation: A modular approach. *Hippocampus*, 6:720–734.

Sharp, P. E., Tinkelman, A., and Cho, J. (2001). Angular velocity and head direction signals recorded from the dorsal tegmental nucleus of Gudden in the rat: implications for path integration in the head direction cell circuit. *Behavioral Neuroscience*, 115:571–588.

Shibata, H. (1987). Ascending projections to the mammillary nuclei in the rat: A study using retrograde and anterograde transport of wheat germ agglutinin conjugated to horseradish peroxidase. *Journal of Comparative Neurology*, 264:205–215.

Skaggs, W. E., Knierim, J. J., Kudrimoti, H. S., and McNaughton, B. L. (1995). A model of the neural basis of the rat's sense of direction. In Tesauro, G., Touretzky, D., and Leen, T., editors, *Advances in Neural Information Processing Systems*, volume 7, pages 173–180. MIT Press, Cambridge, MA.

Song, P. and Wang, X.-J. (2003). A three-population attractor network model of rodent head direction system without recurrent excitation. *Society of Neuroscience Abstracts*, 29:939.3.

Song, P. and Wang, X.-J. (2005). Angular path integration by moving "hill of activity": A spiking neuron model without recurrent excitation of the head-direction system. *Journal of Neuroscience*, 25:1002–1014.

Stackman, R. W. and Taube, J. S. (1998). Firing properties of rat lateral mammillary single units: head direction, head pitch, and angular head velocity. *Journal of Neuroscience*, 18:9020–9037.

Stent, G. S. (1973). A psychological mechanism for Hebb's postulate of learning. *Proceedings of the National Academy of Sciences of the USA*, 70:997–1001.

Stratton, P., Wyeth, G., and Wiles, J. (2010). Calibration of the head direction network: A role for symmetric angular head velocity cells. *Journal of Computational Neuroscience*, 28:527–538.

Stringer, S. M., Perry, G., Rolls, E. T., and Proske, J. H. (2006). Learning invariant object recognition in the visual system with continuous transformations. *Biological Cybernetics*, 94:128–142.

Stringer, S. M. and Rolls, E. T. (2006). Self-organizing path integration using a linked continuous attractor and competitive network: Path integration of head direction. *Network: Computation in Neural Systems*, 17:419–445.

Stringer, S. M. and Rolls, E. T. (2008). Learning transform invariant object recognition in the visual system with multiple stimuli present during training. *Neural Networks*, 21:888–903.

Stringer, S. M., Rolls, E. T., and Tromans, J. M. (2007). Invariant object recognition with trace learning and multiple stimuli present during training. *Network: Computation in Neural Systems*, 18:161–187.

Stringer, S. M., Trappenberg, T. P., Rolls, E. T., and De Araujo, I. E. T. (2002). Self-organizing continuous attractor networks and path integration: One dimensional models of head direction cells. *Network: Compuation in Neural Systems*, 13:217–242.

Suzuki, S., Augerinos, G., and Black, A. H. (1980). Stimulus control of spatial behavior on the eight-arm maze in rats. *Learning and Motivation*, 11:1–8.

Taube, J. S. (1995). Head direction cells recorded in the anterior thalamic nuclei of freely moving rats. *Journal of Neuroscience*, 15:70–86.

Taube, J. S. (1998). Head direction cells and the neurophysiological basis for a sense of direction. *Progress in Neurobiology*, 55:225–256.

Taube, J. S. (2007). The head direction signal: Origins and sensory-motor integration. *Annual Review of Neuroscience*, 30:181–207.

Taube, J. S. and Burton, H. L. (1995). Head direction cell activity monitored in a novel environment and during a cue conflict situation. *Journal of Neurophysiology*, 74:1953–1971.

Taube, J. S., Muller, R. U., and Ranck, Jr., J. B. (1990a). Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *Journal of Neuroscience*, 10:420–435.

Taube, J. S., Muller, R. U., and Ranck, Jr., J. B. (1990b). Head-direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations. *Journal of Neuroscience*, 10:436–447.

Taylor, J. G. (1999). Neural 'bubble' dynamics in two dimensions: Foundations. *Biological Cybernetics*, 80:393–409.

Téroni, E., Portenier, V., and Etienne, A. S. (1987). Spatial orientation of the golden hamster in conditions of conflicting location-based and route based information. *Behavioral Ecology and Sociobiology*, 20:389–397.

Traub, R. A., Jeffreys, J. G. R., and Whittington, M. A. (1999). *Fast Oscillations in Cortical Circuits*. MIT Press, Cambridge, MA.

Wallis, G. and Rolls, E. T. (1997). Invariant face and object recognition in the visual system. *Progress in Neurobiology*, 51:167–194.

Wehner, R. and Wehner, S. (1990). Insect navigation: Use of maps or Ariadne's thread? *Ethology, Ecology, and Evolution*, 2:27–48.

Wiener, S. I. (1993). Spatial and behavioral correlates of striatal neurons in rats performing a self-initiated navigation task. *Journal of Neuroscience*, 13:3802–3817.

Wills, T. J., Cacucci, F., Burgess, N., and O'Keefe, J. (2010). Development of the hippocampal cognitive map in preweanling rats. *Science*, 328:1573–1576.

Wirtshafter, D. and Stratford, T. R. (1993). Evidence for gabaergic projections from the tegmental nucleus of gudden to the mammillary body in the rat. *Brain Research*, 630:188–194.

Xie, X., Hahnloser, R. H. R., and Seung, H. S. (2002). Double-ring network model of the head-direction system. *Physical Review E*, 66:041902.

Yoganarasimha, D. and Knierim, J. J. (2005). Coupling between place cells and head direction cells during relative translations and rotations of distal landmarks. *Experimental Brain Research*, 160:344–359.

Yoganarasimha, D., Yu, X., and Knierim, J. J. (2006). Head direction cells maintain internal coherence during conflicting proximal and distal cue rotations: Comparison with hippocampal place cells. *Journal of Neuroscience*, 26:622–631.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory. *Journal of Neuroscience*, 16:2112–2126.

Zugaro, M. B., Arleo, A., Berthoz, A., and Wiener, S. I. (2003). Rapid spatial reorientation and head direction cells. *Journal of Neuroscience*, 23:3478–3482.

Zugaro, M. B., Arleo, A., Déjean, C., Burguière, E., Khamassi, M., and Wiener, S. I. (2004). Rat anterodorsal thalamic head direction neurons depend upon dynamic visual signals to select anchoring landmark cues. *European Journal of Neuroscience*, 20:530–536.

Zugaro, M. B., Berthoz, A., and Wiener, S. I. (2001). Background, but not foreground, spatial cues are taken as references for head direction responses by rat anterodorsal thalamus neurons. *Journal of Neuroscience*, 21 RC154:1–5.

Zugaro, M. B., Tabuchi, E., and Wiener, S. I. (2000). Influence of conflicting visual, inertial and substratal cues on head direction cell activity. *Experimental Brain Research*, 133:198–208.