

Backpropagation and the brain

Timothy P. Lillicrap^{1,2*}, Adam Santoro^{1*}, Luke Marris¹,
Colin Akerman³, Geoffrey Hinton^{4,5}

Abstract

During learning the brain modifies synapses to improve behaviour. In the cortex synapses are embedded within multi-layered networks, making it difficult to determine the effect of an individual synaptic modification on the behaviour of the system. The backpropagation algorithm solves this problem in deep artificial neural networks, but has historically been viewed as biologically problematic. Nonetheless, recent developments in neuroscience and the successes of artificial neural networks have reinvigorated interest in whether backpropagation offers insights for understanding learning in the cortex. The backpropagation algorithm learns quickly by computing synaptic updates using feedback connections to deliver error signals. While feedback connections are ubiquitous in the cortex, it is difficult to see how they could deliver the error signals required by strict formulations of backpropagation. Here we build on past and recent developments to argue that feedback connections may instead induce neural activities whose differences can be used to locally approximate these signals, and hence drive effective learning in deep networks in the brain.

Introduction

The brain learns by modifying the synaptic connections between neurons¹⁻⁵. Although synaptic physiology helps explain the rules and processes behind individual modifications, it does not explain how individual modifications coordinate to achieve a network’s goal. Since learning cannot be just a blind accumulation of myopic, synapse-specific events that do not consider downstream behavioural consequences, we need to uncover the principles orchestrating plasticity across whole networks if we are to understand learning in the brain.

Within machine learning, researchers study ways of coordinating synaptic updates to improve performance in artificial neural networks, without being constrained by biological reality. They start by defining the architecture of the neural network, which comprises the number of neurons and how they are connected. For example, they often use deep networks with many layers of neurons since these architectures have proved to be very effective for many tasks. Next, researchers define an error function⁶ that quantifies how poorly the network is currently achieving its goals, and then they search for learning algorithms that compute synaptic changes that reduce the error (see Figure 1).

In machine learning, backpropagation of error (“backprop”)⁷⁻¹⁰ is the algorithm most often used to train deep neural networks (see Box 1), and is the most successful learning procedure for deep neural

* Denotes equal contribution.

¹DeepMind; ²University College London; ³University of Oxford; ⁴University of Toronto; ⁵Google Brain
Correspondence: Timothy Lillicrap countzero@google.com, Geoffrey Hinton geoffhinton@google.com

networks. Networks trained with backprop are at the heart of recent successes of machine learning, including state-of-the-art speech¹¹ and image recognition^{12,13}, as well as language translation¹⁴. Backprop also underpins recent progress in unsupervised learning problems such as image and speech generation^{15,16}, language modelling¹⁷, and other next-step prediction tasks¹⁸. As well, *combining* backprop with reinforcement learning has given rise to significant advances in solving control problems, such as mastering Atari games¹⁹, and beating top human professionals in the games of Go^{20,21} and Poker²².

Backprop uses error signals sent through feedback connections to adjust synapses, and has classically been described in the supervised learning setting (i.e., with explicit, externally provided targets). On the other hand, the brain appears to use its feedback connections for different purposes^{23,24}, and is thought to learn in a predominantly unsupervised fashion^{1,25–27}, building representations that make explicit the structure that is only implicit in the raw sensory input. It is natural to wonder, then, whether backprop has anything to tell us about learning in the brain^{25,28–30}.

Here, we will argue that in spite of these apparent differences, the brain has the capacity to implement the core principles underlying backprop. The main idea is that the brain could compute effective synaptic updates by using feedback connections to induce neuron activities whose locally computed differences encode backpropagation-like error signals. We link together a seemingly disparate set of learning algorithms into this framework, which we call Neural Gradient Representation by Activity Differences (NGRAD)^{9,27,31–41}. The NGRAD framework demonstrates that it is possible to embrace the core principles of backpropagation while sidestepping many of its problematic implementation requirements. These considerations may be relevant to any brain circuit that incorporates both feedforward and feedback connectivity. We nevertheless focus on the cortex, which is defined by its multi-laminar structure and hierarchical organisation, and so has long been viewed as exhibiting many of the architectural features associated with deep networks.

Credit assignment in neural networks

This review emphasizes the role of learning. It should be acknowledged that brains undoubtedly have prior knowledge optimized by evolution (e.g., in the form of neural architectures and default connectivity strengths). Priors may ensure that only a limited amount of learning based on a relatively small amount of task error or feedback information is needed throughout an animal’s lifetime to acquire all of the skills it exhibits. Nonetheless, though animals often display impressive behaviours right from birth, they are also capable of extraordinary feats that could not have been tuned by evolution and instead require long bouts of learning. Some examples of such feats in humans are: playing Go and chess; programming a computer and designing a video game; writing and playing a piano concerto; learning the vocabulary and grammar of multiple languages; recognizing thousands of objects; and diagnosing a medical problem and performing vascular microsurgery. Recent work in machine learning suggests that these behaviours depend on a powerful and general learning algorithm^{12,20}. Our interest here, then, is in characterizing this learning algorithm, and specifically how it can assign credit across multiple layers of neurons.

Correlative methods

Synaptic weights determine neural activity, neural activity determines the network’s output, and the network’s output determines the network’s error. In artificial networks, we can therefore reduce the error slightly by making small changes in the synaptic weights. But it is non-trivial to decide whether to increase or decrease any particular weight because a synapse strength does not influence the network’s output directly; rather, it influences its immediate post-synaptic neurons, they influence their post-synaptic neurons, and so on all the way to the output of the network. The radius of the synapse’s influence – its projective field – rapidly expands, so the effect of changing the synapse strength depends on the strengths of many subsequent synapses in the network (e.g., the red connections originating from neuron j in Figure 1).

A conceptually simple way to decide whether to strengthen or weaken a synapse is to *measure* the effect of changing a synapse strength on the error. Such a measurement is easy to make in artificial networks: First, some input is injected into the network and the network’s baseline error is recorded. Next, noise is added to a particular synapse, and the same input is injected back into the network. Finally, one accepts the modified (/noised) synaptic weight if the network’s new error is less than the baseline error, and rejects the modification if the new error is larger than the baseline error^{42–44}. This can be implemented very simply by a learning rule that broadcasts a global scalar representing the overall change in performance of the network (see Figure 1a). Suppose the performance of a network is captured by an error function that computes the degree to which the network’s outputs $[y_1, \dots, y_M]$ deviate from their target values $[t_1, \dots, t_M]$, e.g. via the squared error, $E = \frac{1}{2} \sum_l (y_l - t_l)^2$. To improve the error we simply update a weight in the network W_{ij} via: $\Delta W_{ij} = -\eta(E' - E)\xi_{ij}$, where η is a learning rate, E is the error computed before noise is added, and E' is the error computed after Gaussian noise $\xi_{ij} \sim \mathcal{N}(0, \sigma)$ is added to W_{ij} . Although this method works and is easy to understand, it is extremely inefficient to measure the error of the whole network in order to know how to change a single synapse.

If changes at some synapses have more effect on performance than changes at other synapses, we can do a bit better by measuring the effects of making N different synaptic changes simultaneously (in parallel), but this does not really solve the efficiency issue because we then require about N trials before we can reliably infer whether increasing any particular synapse strength will increase or decrease the error^{44–46}. These kinds of ‘weight perturbation’ methods can be further improved by perturbing the outputs of neurons instead of weights^{42,47}. Such ‘node perturbation’ methods compute local derivatives of a neuron’s activity with respect to its own weights to speed up learning, but these methods are still very slow, and the performance gap increases as network size increases⁴⁴. We think it likely that the brain employs perturbation methods for *some* kinds of learning. However, it is striking that there has not yet been any successful application of these methods to training large, deep networks for difficult problems, such as classifying natural images of many different types of object.

The backprop algorithm addresses the efficiency issues present in perturbation methods by *computing* rather than measuring how a change in a synapse strength affects the network’s error (see Box 1). This computation is possible because we have access to the exact causal relationship between the synapse strengths and the network’s output. By contrast, the causal relationship between the genotype and the phenotype generally depends upon unknown aspects of the environment, so

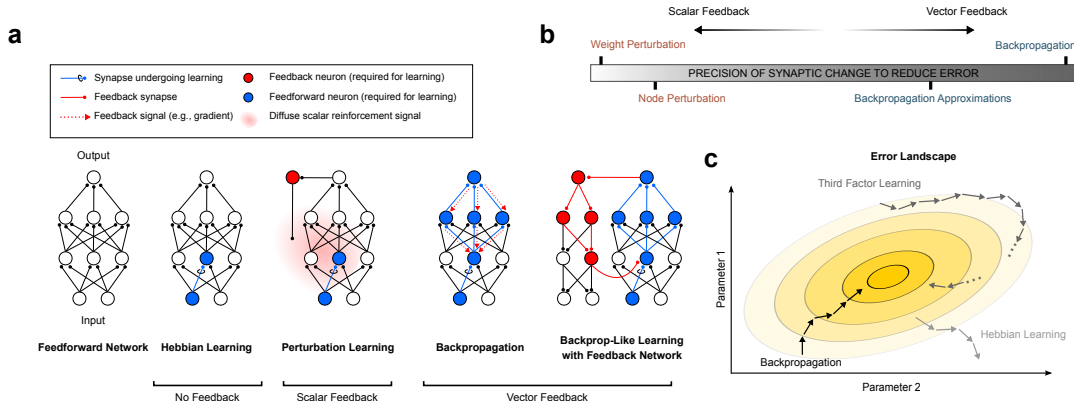


Figure 1: (a) Left to right: a neural network computes an output through a series of simple computational units. To improve its outputs for a task it adjusts the synapses between these units. Simple Hebbian learning cannot make meaningful changes to the blue synapse, because it does not consider this synapse’s downstream effect on the network output. Perturbation methods *measure* the change in error caused by random perturbations to neural activities (node perturbation) or synapse strengths⁴⁴ (weight perturbation) and use this measured change as a global scalar reinforcement signal that controls whether a proposed perturbation is accepted or rejected. The backprop algorithm instead *computes* the synapse update required to most quickly reduce the error. In backprop, vector error signals are delivered backwards along the original path of influence for a neuron. In the brain, vector feedback might be delivered in a variety of ways, including via a separate network. (b) Backpropagation and perturbation algorithms fall along a spectrum with respect to the specificity of synaptic change they prescribe. (c) Algorithms on this spectrum learn at different speeds. Without feedback, synaptic parameters wander randomly on the error surface. Scalar feedback does not require detailed feedback circuits, but learns slowly. Since the same signal is used to inform learning at all synapses the difficulty of deciding whether to strengthen or weaken a synapse scales with the number of synapses in the network: If millions of synapses are changed simultaneously, the effect of one synapse change is swamped by the noise created by all the other changes and it takes millions of trials to average away this noise^{43–46}. The inverse scaling of learning speed with network size makes global reinforcement methods extremely slow, even for moderately sized neural networks. Precise vector feedback via backprop learns quickly. In real networks, it is not possible to make perfect use of the internal structure of the network to compute per-synapse changes, but the brain may have discovered ways to approximate the speed of backprop.

measuring the effects of genetic changes may be the only reasonable algorithm for evolution.

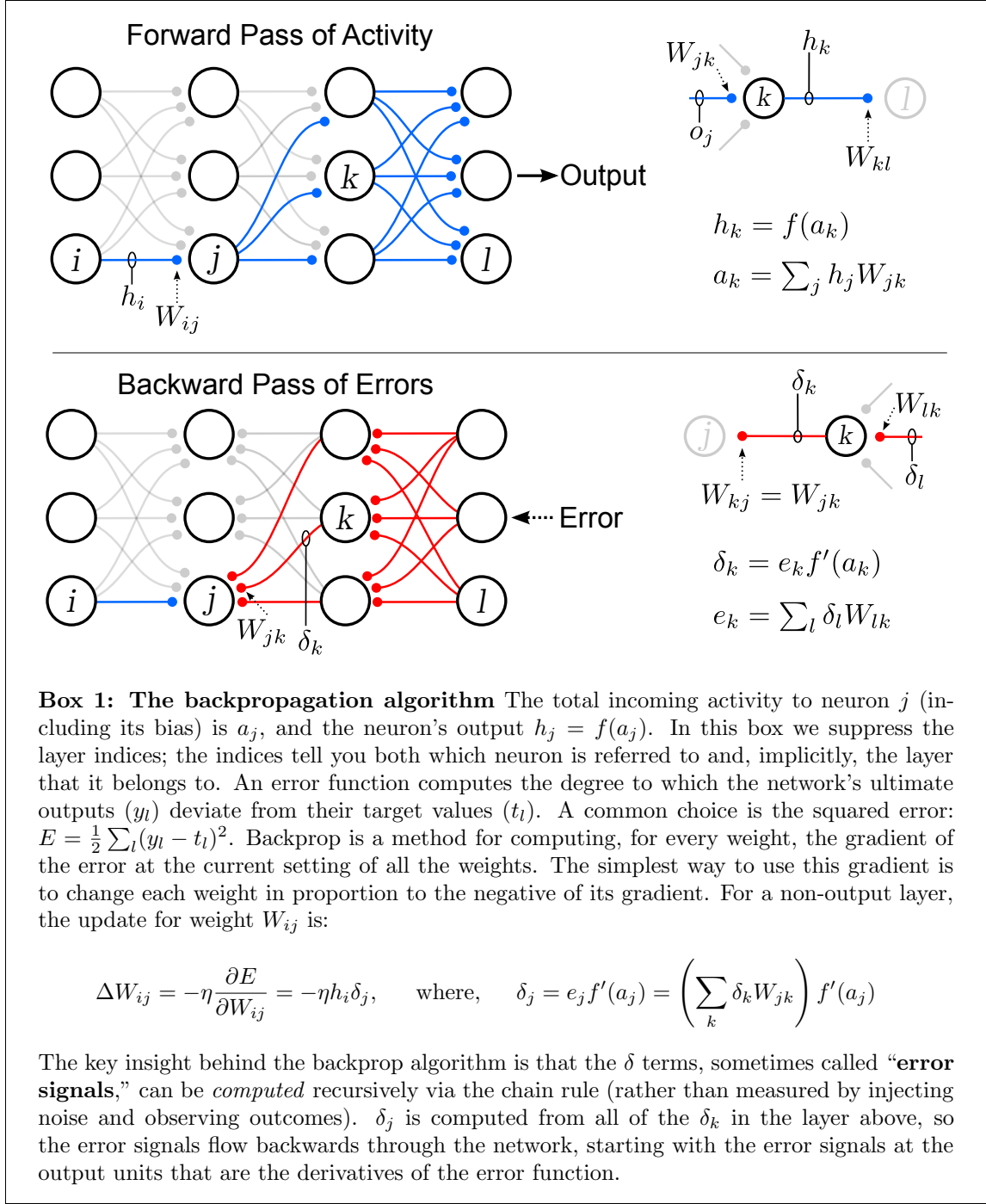
Backpropagation

Backprop *computes* how slightly changing each synapse strength would change the network’s error using the chain rule of calculus. Moreover, it does this computation for all the synapse strengths at the same time and it only requires the same amount of computation as is needed for a forward propagation pass through the network. Its key insight is to implement the chain rule of calculus using a recursive computation of “**error signals**” (see Box 1 for backprop’s algorithmic details). In a hierarchical, multi-layer neural network, the error signals for the neurons in one layer are computed from the error signals in the layer above. Thus, error computations start in the final layer and flow backwards, leading to the notion of errors “backpropagating” through the network. Once error signals have been computed for every neuron, the final output error can be reduced by changing the incoming weights of each neuron so as to push its post-synaptic activity in the direction specified by the error signal.

Backprop is often presented as requiring explicit output targets that are paired with corresponding input patterns. In fact, backprop’s recursive application of the chain rule provides a more general mechanism for computing how changes in the activity of one part of a network effect activities downstream. This mechanism is thus broadly applicable to credit assignment in multi-layer networks. For simplicity, we follow the supervised learning paradigm, but note that backpropagated signals do not need to be a difference from a supervised target. They can also be a temporal difference error or a policy gradient in reinforcement learning^{19,46,48–50}, or a reconstruction or prediction error for an unsupervised algorithm (see Section in the Supplementary Information). All of these can be self constructed by an organism without reference to an external target.

An important empirical feature of backprop—and perhaps a key reason for its success—is its ability to quickly find good **internal representations** of inputs⁵¹ when training deep neural networks. Internal representations are not specified explicitly by the input or the output targets. Instead they must be discovered over the course of learning. Internal representations comprise useful building blocks – such as representations for edges, fragments of shapes, or semantic features of words, etc. – that the network’s intermediate layers use to efficiently code many different entities by using combinations of shared features. The distributed representation of input data as an activity vector of reusable, mix-and-match features allows networks to represent novel data as new combinations of familiar features, and this allows the network to **generalize** to new data from the same distribution as the training data, and in some cases to data that is outside this distribution.

The backprop algorithm has two main features that are critical for its operation. These features are remarkably consistent with biological networks. First is the prescription of synapse-specific changes. Synaptic plasticity mechanisms are widely accepted to exhibit synapse-specificity in biological networks undergoing learning^{5,52}, such as in STDP^{2–4}, where the specific pre- and post-synaptic spike trains influence the particular synaptic modifications between two neurons. Second is the requirement for feedback connections to deliver error information to neurons deep within the network so that they can compute the necessary synaptic changes. We will refer to a learning algorithm as ‘backprop-like’ if it optimizes a downstream objective by using detailed vector feedback to help



prescribe synapse specific updates. Feedback connections between areas permeate every network of the cortex and, critically, can act to modulate the spiking of “feedforward” neurons. These feedback connections can take the form of direct ‘top-down’ corticocortical connections from higher processing cortical areas to lower processing areas, like those that exist between V2 and V1 within the visual system²³. Equally, feedback connections could be routed through thalamus, via cortico-thalamo-cortical loops that can deliver higher order information to cortical regions and individual neurons that receive lower order information^{53–55}.

It is not clear in detail what role feedback connections play, so we cannot say that cortex employs backprop-like learning. Yet, if feedback connections modulate spiking, and spiking determines the adaptation of synapse strengths, the information carried by the feedback connections must clearly influence learning! Backpropagation can be viewed as a very good candidate for what this influence *should be* if the cortex is to be an efficient learning machine. This still leaves open the details of exactly how the feedback connections could approximate backpropagation and there are good arguments to suggest that some of the most obvious implementations are biologically unrealistic (discussed below)^{25,28–30,56,57}. But these arguments do not mean that backprop should be abandoned as a guide to understanding learning in the brain^{57–59}; its core ideas – that neural networks can learn by computing synapse-specific changes using information delivered in an intricate web of feedback connections – have now proven to be so powerful in so many different applications that we need to investigate less obvious implementations.

It should be noted at the outset that the cortex differs from artificial neural networks in many significant ways. For example: (1) There is no straightforward mapping between layers in an artificial network and layers (i.e. Layers 1-6) or areas (e.g. V1 and V2) in the cortex, (2) Cell types, connectivity, and gene expression differ between different areas of the cortex^{60–63}, (3) Cortical areas send and receive different kinds of connections to and from a variety of cortical and sub-cortical areas, etc. Nevertheless, there are also overarching similarities across cortical areas, such as the prevalence of microcolumns^{64,65} and canonical connectivity patterns both within and between cortical areas^{54,66}, which suggest common computations⁶⁷, and we believe that understanding these common computations will be useful. Specifically, we imagine that an algorithm akin to backprop is required to coordinate synaptic changes.

Is there evidence for backprop in the brain?

There is no direct evidence that the brain uses a backprop-like algorithm for learning. Past work has, however, shown that backprop-trained models can account for observed neural responses, such as the response properties of neurons in posterior parietal cortex⁶⁸ and primary motor cortex⁶⁹. A new wave of evidence from neuroscience modelling of the visual cortex carries this trend forward^{58,70–72}. This work reveals that multi-layer models trained with backprop to classify objects tend to have representations that better match those along the visual ventral stream in primates (Figure 2). Models that are not trained with backprop (such as bio-inspired models using Gabor filters^{73,74}, or networks that used non-backprop optimization⁵⁹) do not perform as well as backprop-optimized networks, and their representations do not match those found in IT cortex as well as those discovered by backprop trained models. Representational matches of networks trained by backprop are by no means perfect, and recent work indicates that current models do not explain aspects of human object

classification⁷⁵. Nevertheless, the general phenomena appears to be widespread, with related work demonstrating that auditory neuron responses are also better predicted by multi-layer networks trained by backprop⁷⁶. This does not prove that the cortex learns via backprop-like mechanisms, but as the authors of Cadieu *et al.* [58] state, it shows that “[the] possibility cannot be ruled out merely on representational grounds.”

Performance and representational matches can never on their own establish that backprop-like mechanisms are employed by the brain. The proliferation of computing power and the discovery of better priors may one day allow researchers to train high-performing networks for complex tasks using slower learning algorithms that do not make use of vector feedback. What we can say is that the practicality and efficiency of backprop are at least suggestive that the brain ought to harness detailed, error-driven feedback for learning; to our knowledge, no one in the machine learning community has been able to train high-performing deep networks on difficult tasks such as ImageNet using any algorithm besides backprop. In particular, attempts to rely on algorithms that use only scalar feedback signals, such as genetic algorithms⁷⁷ or REINFORCE⁴⁶, have failed by a wide margin to reach backprop’s level of performance.

In addition to producing models that better match representations observed in the brain, backprop trained deep networks can also help explain the size and timing of receptive field changes in perceptual learning^{72,78}, as well as stage-like transitions observed during some biological learning⁷⁸. Other work has demonstrated that neurons in layers 2/3 of cortex appear to compute detailed mismatches between actual and predicted sensory events⁷⁹, and that neural dynamics at successive stages of the visual cortex are consistent with hierarchical error signals⁸⁰. These findings are consistent with the hypothesis that feedback connections in the cortex drive learning across multiple layers of representation. In the penultimate section we review recently described neural mechanisms that offer additional support for this hypothesis.

A contemporary view of the difficulties in implementing backprop in cortex

Backprop’s computations

While there is mounting evidence that multi-layer networks trained with backprop can help explain neural data, there are difficult questions about how backprop-like learning could be implemented in cortex. Equation 1 gives the synaptic updates prescribed by backprop in matrix/vector notation:

$$\Delta W_\ell = -\eta \frac{\partial E}{\partial W} = -\eta \boldsymbol{\delta}_\ell \mathbf{h}_{\ell-1}^\top \quad \text{where} \quad \boldsymbol{\delta}_\ell = \mathbf{e}_\ell \circ \mathbf{f}'(\mathbf{a}_\ell) = (W_{\ell+1}^\top \boldsymbol{\delta}_{\ell+1}) \circ \mathbf{f}'(\mathbf{a}_\ell) \quad (1)$$

Where bold symbols are vectors, \cdot^\top is the transpose operation, and \circ is element-wise multiplication, and $\mathbf{h}_\ell = \mathbf{f}(\mathbf{a}_\ell)$. Since this equation does not require indices for individual neurons (as in Box 1), we use subscripts to denote the layers. The equation says that the presynaptic weights, W_ℓ , in layer

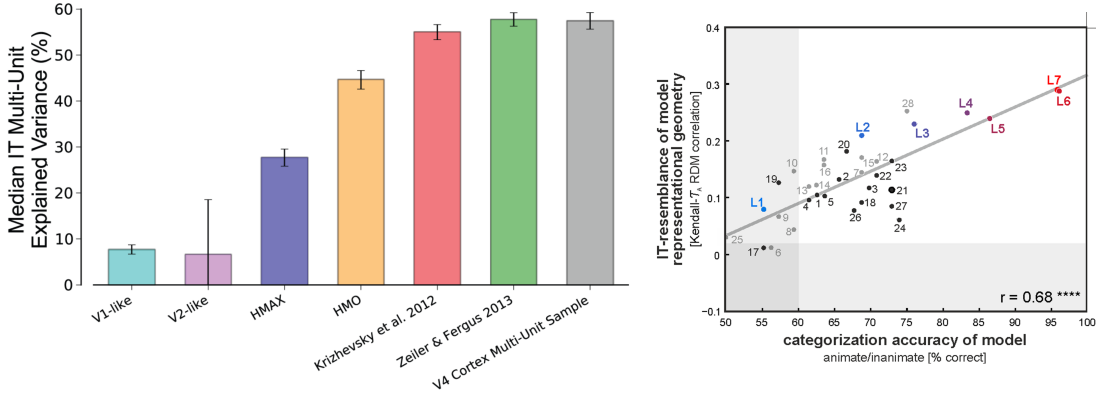


Figure 2: Comparison of backprop trained networks with neural responses in visual ventral cortex. (Left) Cadieu *et al.* [58] showed that backprop trained models^{12,81} (red and green) explain IT multi-unit responses better than other models. (Right) Khaligh-Razavi & Kriegeskorte [70] showed that models with better classification performance more closely resemble IT representations; each numbered dot corresponds to a model, whilst the colored dots L1-L7 correspond to successively deeper network layers. Moreover, neurons in deeper layers within the backprop trained network, match IT better than neurons in earlier layers.

ℓ are updated according to the product of the error signals δ_ℓ and the presynaptic activities which are the outputs of the previous layer $\mathbf{h}_{\ell-1}$. The error signals δ_ℓ are computed by multiplying the error signals from the layer above, $\delta_{\ell+1}$, by the transpose of the post-synaptic weights $W_{\ell+1}^\top$ and then multiplying by the derivative of the activity function $\mathbf{f}'(\mathbf{a}_\ell)$. It's perhaps worth noting that, when presented in the following form: $\Delta W_\ell = -\eta(\mathbf{e}_\ell \circ \mathbf{f}'(\mathbf{a}_\ell))\mathbf{h}_{\ell-1}^\top$, the update can be seen as a local Hebbian-like rule — where the post-synaptic activity is replaced by $\mathbf{f}'(\mathbf{a}_\ell)$ — that is modulated by a third factor⁸², \mathbf{e}_ℓ , which is computed via feedback connections. In the subsequent sections, we will refer to Equation 1 when outlining three major difficulties in implementing backprop in biological circuits.

Backprop demands synaptic symmetry in the forward and backward direction

A naive implementation of backprop requires the delivery of error signals through feedback connections that have exactly the same strength as the feedforward connections. In equation 1, errors, $\delta_{\ell+1}$, travel along feedback weights, $W_{\ell+1}^\top$, that are symmetric to their feedforward counterparts. On a computer, the backprop algorithm sends error information backward using a set of error derivative variables that are distinct from the activity variables used in the forward pass. Soon after backprop was introduced^{83,84}, it was suggested that, in the brain, error information could be delivered by a distinct “error delivery network”, with each neuron in this backward network carrying update information for a matched neuron in the “forward” network. Early work on how backprop might be implemented in the brain also explored the idea that error signals might travel retrogradely along the feedforward axons^{85,86} (with, somehow, equivalent strength as in the anterograde direction). But this idea has been abandoned because retrograde communication is much

too slow to support backprop^{87–89}. The need to have the same weight on two different connections has been called the “weight transport” problem^{25,28,83} and would still be a major hurdle with a second error delivery network. Previous work has considered the use of symmetric learning rules to establish and maintain this weight symmetry^{28,34,39,83,90–92}, but the cortex does not exhibit the requisite point-to-point reciprocal connectivity.

Fortunately, recent work demonstrates that this symmetry is unnecessary^{57,93–100}. Remarkably, networks with fixed *random* feedback weights learn to approximately align their feedforward synaptic weights to their feedback weights. In a display of neural pragmatism, the fake error derivatives computed using the random feedback weights cause updates to the feedforward weights that make the true error derivatives closer to the fake derivatives. This surprising phenomenon, called “feedback alignment”, suggests that feedback connections do not need to be symmetric to their feedforward counterparts in order to deliver information that can be used for fast and effective weight updates⁵⁷. Feedback alignment thus offered early evidence that the kind of precise symmetry employed by backprop is not always required for effective learning. Random feedback weights may be insufficient for learning in deeper networks¹⁰¹, but subsequent work has demonstrated that simple learning mechanisms could shape the backward pathway to ensure that effective feedback is delivered^{100,102}, even in very deep networks trained on a complex task¹⁰².

Error signals are signed and potentially extreme valued

In backprop the information sent backwards through a network to inform updates is conveyed in the form of signed error signals, δ . During training these signals often vary across many orders of magnitudes, a phenomena referred to as exploding and vanishing gradients¹⁰³. While there exists evidence for signed error delivery in apparently single-layered structures such as the cerebellum^{104,105}, feedback of signed error in *deep* networks, such as the cortex, appears problematic. A separate set of “error neurons”^{83,84,102}, as suggested in the previous section, could be used to implement a backward pass with firing rates above a certain value communicating positive error and below this value negative error. However, they must then perform complicated bookkeeping to coherently integrate signed information coming from multiple feedback connections, and to continue the propagation of this information across multiple layers. When thinking about backprop in the brain this problem is still unsolved, but we will explore a potential solution that avoids propagation of error signals altogether.

Feedback in brains alters neural activity

In error backpropagation, feedback connections deliver error signals that do not influence the activity states of neurons produced by feedforward propagation. Rather, the information delivered via δ *only influences synaptic updates*. The role of feedback connections in the brain appears to be fundamentally different. In the cortex for example, these connections influence the neural activities produced by feedforward propagation, and are thought to serve a number of functional roles. For example, top-down control through feedback connections has a well-established link with gain control – i.e., the enhancement or suppression of neural responses depending on, for example, attention

to a particular feature in the visual field^{65,106–118}. Interestingly, feedback connections in cortex can also *drive* activity, rather than just modulate, or enable it.

This idea is corroborated by experiments showing that conduction velocities and types of excitatory synaptic communication are often comparable between feedback and feedforward axons^{119,120}, evidence that early visual areas are activated during visual mental imagery^{121,122}, evidence that top-down feedback is actively involved in bottom-up processing^{123,124}, and by lesion experiments demonstrating cessation of activity following inactivation of feedback^{125,126}. Consequences of this idea appear in a number of proposals, such as Reverse Hierarchy Theory¹²⁷, and hierarchical Bayesian inference (HBI) for perception^{36,128–131}, both of which draw inspiration from Helmholtz’s view of perception as unconscious inference¹³².

Although neuroscientists have proposed several different functions for feedback connections, they have rarely considered the possibility that their *primary* function is to drive learning, see e.g.,²³. There is, however, a long but lesser known history of work in the machine learning literature that uses feedback connections to alter the activities produced during feedforward propagation (unlike backprop), and then uses these alterations to guide learning^{9,27,31–35,37–41}. Here, we suggest that the most important role for feedback connections is to make alterations in neural activities to convey the information required for effective multi-layer learning; that is, the activity *alterations* induced by feedback dictate synaptic weight changes that improve feedforward processing in deep networks.

The NGRAD hypothesis

Using neural activity differences to encode errors

There have been many proposed learning mechanisms that use differences in activity states to drive synaptic changes, rather than propagating or diffusing signals that directly represent gradients explicitly. Around the time that backprop entered the mainstream⁸, several neural network learning algorithms – including the Boltzmann machine^{35,133} – explored this idea by using temporal differences between activities inferred during two phases of propagation to compute updates. Several recently introduced approaches instead use activity differences between sets of neurons in a local circuit¹³⁴, or between different compartments within a neuron^{135,136}.

We call learning mechanisms that use differences in activity states to drive synaptic changes NGRADs, which stands for **N**eural **G**radient **R**epresentation by **A**ctivity **D**ifferences. The idea that the cortex uses an NGRAD to perform an approximation to gradient descent will be called the NGRAD hypothesis. The main attraction of this hypothesis is that it avoids the need to propagate two quite different types of quantity: activities and error derivatives. Instead, NGRADs are based on the idea that higher level activities – coming from a target, another modality, or a larger spatial- or temporal-context – can nudge^{27,34,135} lower level activities towards values that are more consistent with higher-level activity or a desired output. Moreover, the induced change in lower level activities can then be used to compute backprop-like weight updates using only locally available signals. Thus, the fundamental idea is that top-down driven *activities* drive learning without carrying explicit error information between layers.

One concrete example of such an algorithm is GeneRec³⁴, which combines insights from the Boltzmann Machine algorithm¹³³ and the recirculation algorithm²⁷. GeneRec trains multi-layer recurrent networks as follows: in a “negative phase”, the input is provided and recurrent activities are allowed to settle to equilibrium. In a “positive phase”, input is provided to the network while the output neurons are clamped to, or nudged towards, their target values, and activities are again allowed to settle to equilibrium. GeneRec’s learning rule is simple and local: each synaptic weight change should be proportional to the difference between the product of the pre- and post-synaptic activities from the positive and negative phases.

A number of other algorithms, such as Contrastive Hebbian Learning (CHL)³⁷, the Almeida/Pineda algorithms^{31–33}, and the Wake-Sleep algorithm in the Helmholtz Machine^{36,131} use similar logic as the backbone for learning. The most important contribution in our context is their use of locally available information – activity states at different points in time or across different spatial compartments – to capture the error information that guides learning. New work on the biological plausibility of backprop has returned to these ideas^{134,135,137,138}: e.g. the recently introduced Equilibrium-propagation¹³⁸ employs the same essential elements as GeneRec and CHL. Further, several models^{134,135} have examined how NGRAD learning might be achieved without a separate “negative phase”. These models use ideas from predictive coding to make effective updates using locally computed differences across neurons or their compartments, rather than across time. In spite of the fact that NGRADs compute error vectors locally within a layer, rather than transmitting them across layers as in backprop and feedback alignment⁵⁷, many algorithms in this class can be shown to make updates that approximately follow (and in some cases exactly follow) the gradient computed by backprop^{39,134,136,137}.

To gain intuition into how activity differences computed within a layer can be used to guide learning, we will examine a simpler proposal put forward at the first NIPS workshop on deep learning¹³⁹ (see Supplementary Figure 1), and later developed in Lee *et al.* [41]. Fundamental to the proposal is the use of autoencoders^{35,140} to send top-down activities to improve earlier layer activations, and the use of the induced differences to make weight updates. In the following sections we describe autoencoders and then show how they can be used as the basis of a deep learning algorithm motivated by biological constraints.

Autoencoders

We begin by developing the idea of an autoencoder^{35,140}, which is a network that aims to reconstruct its own input. The simplest autoencoder takes a vector input \mathbf{x} , uses this input vector to produce an activity vector in its hidden layer via a weight matrix W and non-linearity: $\mathbf{h} = f(\mathbf{x}; W) = \sigma(W\mathbf{x})$, then uses the hidden activity vector to reconstruct an approximation to the input vector via a backward weight matrix: $\hat{\mathbf{x}} = g(\mathbf{h}; B) = \sigma(B\mathbf{h})$. Autoencoders can be trained without requiring explicit labels because the difference between the original input and the reconstruction, $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$ is used as the error to drive learning. This difference can be computed locally by neurons in the input layer and used to adjust the weights from the hidden layer to the input layer. The input-to-hidden weights might also be adjusted without requiring backprop by using the recirculation algorithm^{27,141}. Another view of this idea is that the feedback connections in an autoencoder learn an approximate inverse function $g(\cdot; B)$ that transforms the hidden activity back to the associated

point in the input space so that $g(f(\mathbf{x}; W); B) \approx \mathbf{x}$. Many applications of autoencoders use hidden layers that are much smaller than the input, but when learning a precise inverse we may wish them to have roughly the same number of units. Most importantly, for error assignment in the NGRAD framework, autoencoders can be used to propagate detailed *activity targets* at higher layers backwards to provide targets for earlier layers, which can in turn be used to compute local differences that are appropriate for driving learning.

Target propagation

Figure 3a sketches target propagation^{142,143}, the essential idea behind using a stack of autoencoders for deep learning. We propagate activity forward through successive layers of a network to produce a predicted output. Then we propagate an output *target* backwards through inverse functions (i.e. via feedback connections) that are learned through layer-wise autoencoding of the forward layers. This backward propagated target induces hidden activity targets that *should have been* realized by the network. In other words, if the network had achieved these hidden activities during feedforward propagation, then it would have produced the correct output. The direction in the activity space between the feedforward activity and the feedback activity indicates the direction in which the neurons' activities should move in order to improve performance on the data. Learning proceeds by updating forward weights to minimize these local layer-wise activity differences, and it can be shown that under certain conditions the updates computed using these layer-wise activity differences approximate those that would have been prescribed by backprop.

Formally, suppose that we have a stack of autoencoders in which the hidden units of one autoencoder are the input units for the next autoencoder: we have forward functions, $\mathbf{h}_\ell = \mathbf{f}(\mathbf{h}_{\ell-1}; W_\ell) = \sigma(W_\ell \mathbf{h}_{\ell-1})$ for layers $\ell \in \{1, \dots, L\}$, and backward functions $\tilde{\mathbf{h}}_\ell = \mathbf{g}(\tilde{\mathbf{h}}_{\ell+1}; B_{\ell+1}) = \sigma(B_{\ell+1} \tilde{\mathbf{h}}_{\ell+1})$ for layers $\ell \in \{L-1, \dots, 1\}$. For notational convenience we define the input to the network to be $\mathbf{x} = \mathbf{h}_0$, the output of the network to be $\mathbf{y} = \mathbf{h}_L$, and the output target to be $\mathbf{t} = \tilde{\mathbf{h}}_L$. Also for notational convenience we absorb the weight matrices into the subscript and write: $\mathbf{f}_\ell(\mathbf{h}_{\ell-1}) = \mathbf{f}(\mathbf{h}_{\ell-1}; W_\ell)$ and $\mathbf{g}_{\ell+1}(\tilde{\mathbf{h}}_{\ell+1}) = \mathbf{g}(\tilde{\mathbf{h}}_{\ell+1}; B_{\ell+1})$. Suppose further that the autoencoders are perfect so that we have exact inverse functions that map back from each higher layer to the layer below, i.e. $\mathbf{g}_\ell(\cdot) = \mathbf{f}_\ell^{-1}(\cdot)$ so that $\mathbf{g}_\ell(\mathbf{f}_\ell(\mathbf{h}_{\ell-1})) = \mathbf{h}_{\ell-1}$. After forward and backward passes are complete, and assuming that one is computed after the next, the temporal difference between the feedforward activity and the feedback activity target, $\mathbf{e}_\ell = \tilde{\mathbf{h}}_\ell - \mathbf{h}_\ell$, drives plasticity via: $\Delta W_\ell = -\eta \delta_\ell \mathbf{h}_{\ell-1}^\top$ where $\delta_\ell = \mathbf{e}_\ell \circ \mathbf{h}'_\ell$, and \mathbf{h}'_ℓ is the derivative of the activation function in layer ℓ . This idea of using autoencoders to induce targets for deep updates is elegant, but problematic in practice^{41,101,143}, perhaps most obviously because it may be impossible to obtain perfect inverses.

Difference target propagation

We described target propagation above as using perfect autoencoders to convey targets to earlier layers. This constraint is unrealistic but can be fixed by training the backward weights. During the forward pass we try to reconstruct neural activity from the activity in the subsequent layer: $\hat{\mathbf{h}}_\ell = \mathbf{g}_{\ell+1}(\mathbf{h}_{\ell+1})$, shown as light grey dots in Figure 3b. The backward path autoencoders thus induce layer-wise errors, $\mathbf{e}_\ell^B = (\mathbf{h}_\ell - \hat{\mathbf{h}}_\ell)$, which we use to update the feedback weights via: $\Delta B_{\ell+1} =$

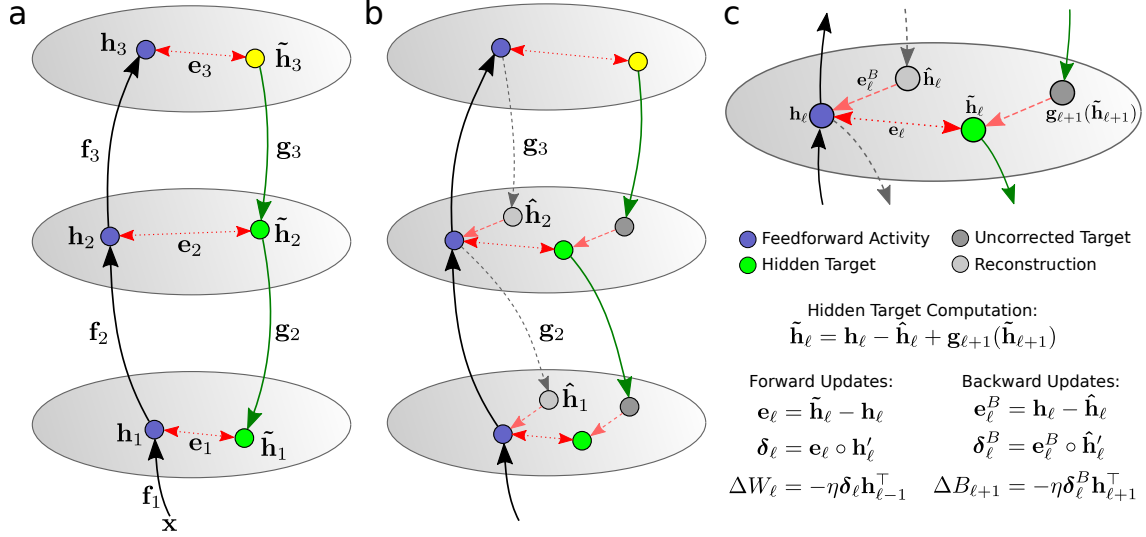


Figure 3: **Target propagation algorithms:** (a) Schematic of target propagation that uses perfect inverses, $\mathbf{g}_\ell(\cdot) = \mathbf{f}_\ell^{-1}(\cdot)$, at each layer. For illustration, high dimensional activity vectors at each layer are represented as points in a 2D space. Local layer-wise errors, $\mathbf{e}_\ell = \tilde{\mathbf{h}}_\ell - \mathbf{h}_\ell$, are computed between the forward pass activities (\mathbf{h}_ℓ , blue), and the top-level ($\tilde{\mathbf{h}}_3$, yellow) and induced targets ($\tilde{\mathbf{h}}_\ell$, green). Synaptic weights, W_ℓ , associated with forward mapping $\mathbf{f}_\ell(\cdot)$ are updated to move forward activity vectors closer to the targets. (b) Difference target propagation helps correct for the fact that the feedback connections may not implement perfect inverses. For each layer, \mathbf{h}_ℓ , we compute a reconstruction, $\hat{\mathbf{h}}_\ell$, from the layer immediately above via $\mathbf{g}_{\ell+1}(\cdot)$. Then, to compensate for imperfections in the autoencoders, we add the reconstruction error, $\mathbf{e}_\ell^B = \mathbf{h}_\ell - \hat{\mathbf{h}}_\ell$, to the uncorrected target $\mathbf{g}_{\ell+1}(\tilde{\mathbf{h}}_{\ell+1})$ (dark grey), computed from the layer above in the backward pass. (c) Schematic for a single layer of DTP. Forward synaptic weights, W_ℓ , are updated to move the forward pass hidden activity closer to the corrected hidden target. Note that the light grey, dark grey, and green circles do not represent separate sets of neurons, but rather different stages of processing performed in the same neurons. Backward synaptic weights, $B_{\ell+1}$, are updated to reduce autoencoder reconstruction errors. The hidden target, $\tilde{\mathbf{h}}_\ell$, is computed as a mixture of the bottom up activity with top-down feedback. Crucially, errors are computed with signals local to the neurons in each layer, rather than propagated between layers as in backprop.

$-\eta \delta_\ell^B \mathbf{h}_{\ell+1}^\top$, where $\delta_\ell^B = \mathbf{e}_\ell^B \cdot \hat{\mathbf{h}}'_\ell$, so that $\mathbf{g}_{\ell+1}$ is moved closer to an approximate inverse for \mathbf{f}_ℓ . In this sense, the circuit learns-to-learn, a phenomena common to many proposed approximations for backprop^{34,91,93,94,102}. We then send the modified target $\hat{\mathbf{h}}_{\ell+1}$ at level $\ell + 1$ backward through these approximate inverses and use the result to make a linear correction to the target at level ℓ : $\hat{\mathbf{h}}_\ell = \mathbf{h}_\ell - \hat{\mathbf{h}}_\ell + \mathbf{g}_\ell(\hat{\mathbf{h}}_{\ell+1})$, shown by green dots in Figure 3b-c. Under certain assumptions⁴¹, this correction allows the autoencoders to perform perfectly *for this particular input*. Finally, we use these corrected targets to update the forward weights as before: $\Delta W_\ell = -\eta \delta_\ell \mathbf{h}_{\ell-1}^\top$. This learning procedure is called difference target propagation (DTP)⁴¹, and is shown along with the layer-wise weight updates in Figure 3b,c.

Difference target propagation effectively trains multi-layer neural networks on classification tasks such as MNIST and CIFAR⁴¹, and it learns in a fraction of the time required by algorithms that use weight or node perturbation to update weights. The performance of algorithms like DTP is still being explored on more challenging datasets and more complex architectures. Recent work shows that straightforward implementations of DTP do not perform as well as backprop on the ImageNet task with large convolutional networks¹⁰¹. The DTP algorithm also does not address questions of online learning or how forward and backward pathways could communicate in biological circuits. Nevertheless, the algorithm provides a compelling example of how locally generated activity differences can be used to drive learning updates for multi-layer networks, and recent work suggests avenues for recovering performance for large-scale tasks^{98,99,102}.

We have emphasized algorithms that send the same kind of signal in both forward and backward directions, and use activity differences local to a layer to compute errors. But it is possible that the brain employs approaches that are closer in spirit to backprop. One may conceive of algorithms wherein neurons switch between propagating ‘feature’ information forward and errors backward, though we’re not aware of evidence for the kind of fast switching between modes that ought to be induced. Another idea would be to use a second set of specially designated neurons to carry errors backward across multiple layers and deliver them to the forward pathway without interfering with its feature processing^{84,100,102}. Both of these approaches require that signed information be carried backwards across multiple layers via unsigned spiking activity. We are not aware of effective solutions to this issue, but these ideas present interesting alternatives to NGRADs that should not be ignored as we seek to understand how multi-layer credit assignment might be implemented in neural circuits. No existing algorithm for multi-layer credit assignment can be straightforwardly squared with what we know about the neurophysiology of cortex. But aspects of the algorithms explored here may help us form the next round of empirical inquiry.

Implementing NGRADs in biological circuits

Existing NGRADs may offer high-level insights into how the brain could approximate backprop, but there are many questions about how such algorithms could be implemented in neural tissue. To function in neural circuits, NGRADs needs to be able to: (1) Coordinate interactions between feedforward and feedback pathways, (2) Compute differences between patterns of neural activities, and (3) Use this difference to make appropriate synaptic updates. It is not yet clear in detail how biological circuits could support these operations, but recent empirical studies present an expanding set of potential solutions to these implementation requirements (see Figure 4).

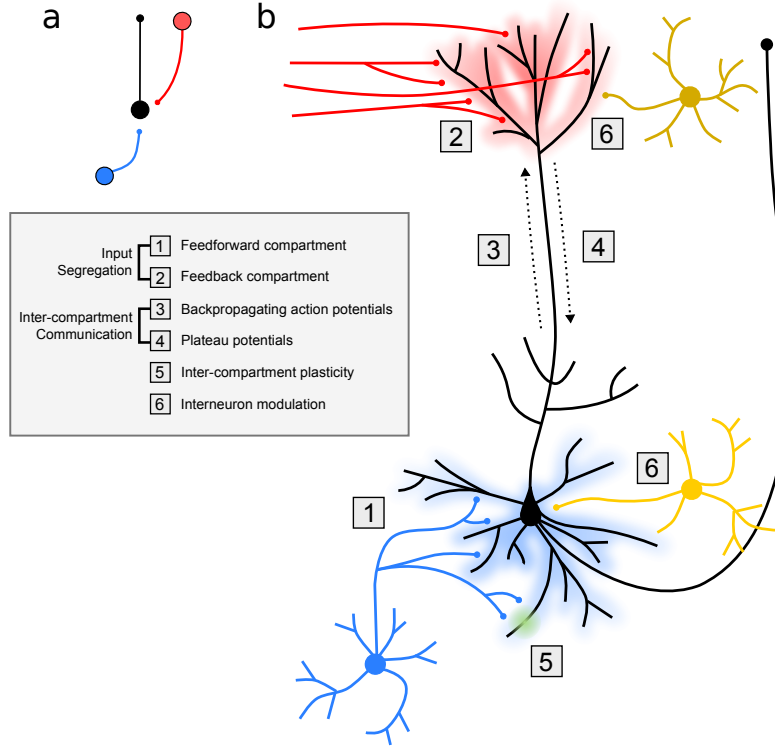


Figure 4: **Empirical findings suggest new ideas for how backprop-like learning might be approximated by the brain.** (a) When backprop was first published, a neuron (black cell) was typically conceived of, and modelled, as a single voltage compartment into which feedforward (blue; e.g. from a lower-order cortical area) and feedback (red; e.g. from a higher-order cortical area) signals would arrive undifferentiated. (b) A contemporary schematic of a cortical pyramidal neuron (black cell). Feedforward and feedback inputs are thought to be treated differently [1] & [2]. They arrive at different compartments of the cell (e.g. basal and apical dendrites, respectively) and may be electrotonically segregated. Compartments can communicate selectively via backpropagating action potentials that are triggered by spikes in the soma, and via calcium spike induced plateau potentials generated in the apical dendrite [3] & [4]. Plasticity in one compartment may depend both on local synaptic events and events triggered in another compartment [5]. For example, ‘forward’ basal synaptic plasticity may be altered by the arrival of apically generated plateau potentials. Finally, local inhibitory neurons (yellow cells) [6] can regulate the communication between the subcellular compartments and can themselves be differentially recruited by higher-order inputs, and thus modulate the interactions between forward and backward pathways.

On a computer it is trivial to compute and store memories of separate feedforward and feedback passes in order to take differences and then use these for learning. There have been several proposals for how the same might be achieved in neural tissue. One of the earliest, proposed for backprop, was

to employ a second set of “error” neurons which could act as the feedback variables^{83,84}. The same idea could work for DTP, but there is no evidence for entirely segregated feedforward neurons in cortex that are unaffected by feedback activity. Feedback tends to be area-wise reciprocal in cortex so that if area A sends forward connections to area B, it will tend to receive feedback from neurons in B^{54,66}. If, as it appears, feedforward and feedback paths in the brain “reuse” the same neurons, this has important implications for how backprop-like learning might be achieved in cortex. Naively, sharing the same neurons requires some form of strong time-multiplexing, where the forward pass occurs and then is replaced by a backwards pass. Again, there is no direct support for this kind of staggered activity — in the brain both feedforward and feedback pathways appear to be active and interacting, at least to some extent, simultaneously.

When thinking about biological implementations of backprop, it is common to assume that feedforward and feedback synapses have similar functional effects on a neuron^{34,37}. If this assumption is correct, something like time-multiplexing is required in order to compute the differences needed for learning. In reality, however, biological neurons are known to contain functionally and anatomically distinct compartments and activity arriving at these different subcellular domains can have varying effects on the cell.

For example, apical tuft dendrites of layer-V cortical pyramidal neurons are electrotonically isolated from somatic and basal compartments. The apical dendritic compartment can receive feedback connections from higher cortical areas^{144,145}, or via higher order thalamic nuclei^{146,147}, and can act as semi-independent reservoirs that only communicate with the somatic compartment under certain conditions¹⁴⁸. As well, “feedforward” connections are thought to predominantly target the basal dendrites of pyramidal neurons, which act as functionally distinct regions that probably have different plasticity rules^{148–153}.

Neurocomputational models have begun to employ more realistic neuron models with segregated spatial compartments and unique computational properties per compartment^{92,154,155}. These more complicated neurons avoid many of the problems that vex point-process neuron models. If information can be separated in space within a cell, then the two propagation phases – feedforward and feedback – do not require strict temporal segregation, and hence might occur simultaneously. Interactions between compartments can occur in a variety of ways, including backpropagating action potential (BAPs) and plateau potentials¹⁵⁶. Further, these bidirectional interactions between cellular compartments can be controlled by other factors, such as inhibitory interneuron activity that arrives at specific locations on pyramidal neurons¹⁵⁷. Segregated dendrites may also underlie “burst ensemble multiplexing” schemes for representing feedforward and feedback streams of information simultaneously in cortex¹⁵⁵; this work shows that high frequency bursts and singlet spikes can be driven by different compartments and may convey different kinds of information in forward and backward paths.

Empirical studies also suggest that plateau potentials generated from electrotonically segregated dendrites can alter the plasticity in feedforward connections^{158–160}, sometimes with as few as five pairings of subthreshold presynaptic activity and at behavioural time scales^{161,162}. These plasticity findings may help explain how dendritic segregation may be used to compute differences and inform synaptic updates. Urbanczik & Senn [154] describe how errors might be computed across somatic and dendritic compartments to drive learning. Guerguiev *et al.* [163] build on this work to show how segregated dendrites could underlie a biologically motivated variant of DTP in a spiking

network simulation. More recently, Sacramento *et al.* [135] have proposed a variant of equilibrium propagation¹³⁸ that makes use of both dendritic segregation and local interneuron circuitry in order to achieve NGRAD learning without assuming two phases. Nevertheless, how forward and feedback pathways in biological tissue coordinate their activities to achieve effective credit assignment remains an open question.

Another intriguing detail related to biological implementation lies in the fact that many NGRAD algorithms, but not backprop, compute their learning updates using feedback driven improvements to lower layer activity^{34,35,37,134,136,138}. This aspect of these algorithms fits well with evidence from physiology which suggests that top-down feedback is actively involved in improving bottom-up information processing^{23,123,124}. Taken together, these new findings and related theories are expanding the horizon of possible credit assignment mechanisms that might be considered biologically plausible.

While empirical work has broadened our understanding of how backprop-like algorithms might operate, there remain difficult challenges in terms of how to test for such learning mechanisms in the brain. One simple question that has been neglected in the literature, but is predicted by backprop and NGRADs alike, is that feedback circuitry can influence the sign and amplitude of plasticity that is elicited at feedforward synapses. To test this, we could build upon previous approaches^{147,164} to examine the effects of synapse-specific plasticity protocols at feedforward connections to a cortical pyramidal neuron’s basal dendrites, whilst simultaneously controlling activity patterns delivered via feedback connections to the neuron’s apical dendrite. This type of experiment would not constitute *proof* of backprop, or of a particular NGRAD algorithm, but since both require that detailed feedback activity can alter feedforward learning, this kind of experimental protocol would be a natural starting place for understanding multi-layer learning.

Summary

The way in which the cortex modifies synapses so as to improve the performance of complicated multi-stage networks remains one of the biggest mysteries in neuroscience. The introduction of backpropagation generated excitement in the neuroscience community as a possible source of insight about learning in cortex²⁸. But its relevance to the cortex was quickly cast in doubt – partly because it failed to produce truly impressive performance in artificial systems, and partly because, interpreted literally, it has obvious biological implausibilities^{25,28,29,57}.

With the advent of greater computing power, bigger data-sets and a few technical improvements, backprop can now train multi-layer neural networks to be competitive with human abilities^{19,20,22}. We believe that backprop offers a conceptual framework for understanding how the cortex learns, but there are many remaining mysteries around how the brain could approximate it. Some of these mysteries are minor and easily addressed. For example, backprop networks are typically rate based rather than spiking (see Supplementary for clarity), and violate Dale’s Law¹⁶⁵: that neurons in the brain form *either* excitatory or inhibitory connections¹⁶⁶, whereas artificial neurons are usually able to form both^{167–170}. Others, however, such as the computation and backwards delivery of error signals, pose deeper conceptual issues. NGRADs resolve significant implausibilities of backprop in a way that is intuitive and consistent with how we believe biological circuits operate. They do

away with the explicit propagation of error derivatives, and instead compute them locally through differences of propagated activities.

There are many pieces missing from a story that would firmly connect backprop with learning in the brain. Nevertheless, the situation now is very much reversed from 30 years ago²⁸, when it was thought that neuroscience may have little to learn from backprop because aspects of the algorithm seem biologically unrealistic. The reality is that in deep neural networks, learning by following the gradient of a performance measure works really well. It therefore seems likely that a slow evolution of the thousands of genes that control the brain would favor getting as close as possible to computing the gradients needed for efficient learning of the trillions of synapses it contains.

Competing Interests:

The authors declare no competing interests.

Glossary:

Learning: The modification of network parameters, such as synaptic weights, to enable better performance according to some measure, such as an error function.

Error function: An explicit quantitative measure for determining the quality of a network's output. Also frequently called a loss or objective function.

Credit Assignment: The determination of the degree to which a particular parameter, such as a synaptic weight, contributes to the magnitude of the error.

Deep Learning: Learning in networks that consist of hierarchical stacks, or layers, of neurons. Deep learning is especially difficult because of the difficulty inherent in assigning credit to a vast number of synapses situated deep within the network.

Backpropagation of Error (backprop): An algorithm for explicitly computing the changes to prescribe to synapses in deep networks to improve performance. It involves the flow of error signals through feedback connections from the output of the network towards the input.

Error signals: The contribution to the error by the activities of neurons situated closer to the output. In backpropagation, these signals are sent backwards through the network to inform learning.

Target: The desired output of a network given some input. Deviation from the target is quantified with an error function.

Supervised Learning: Learning in which the error function involves an explicit target. The target tends to contain information unavailable to the network, such as a ground truth labels.

Unsupervised Learning: Learning in which the error function does not involve a separate output target. Instead, errors are computed using other information readily available to the network, such as the input itself, or the next observation in a sequence.

Reinforcement Learning: Learning in an interactive trial-and-error loop whereby an agent acts stochastically in an environment and uses correlations between actions and accumulated scalar rewards to improve performance.

Generalization: The capacity of a network to produce a correct output for an input that it has not observed.

Internal representations: Hidden activity of a network that represents the network's input data. "Useful" representations tend to be those that efficiently code for redundant features of the input data and lead to good generalization, such as the existence of oriented edges in handwritten digits.

Auto-encoding Unsupervised learning in which the target is the input itself. One application of auto-encoding is the training of feedback connections to coherently carry 'targets' to earlier layers.

Weights Network parameters that determine the strength of neuron-neuron connections. A pre-synaptic neuron connected to a post-synaptic neuron with a high weight will greatly influence the activity of the post-synaptic neurons, and vice versa.

References

1. Hebb, D. O. *The organization of behavior: A neuropsychological approach* (John Wiley & Sons, 1949).
2. Markram, H. & Sakmann, B. *Action potentials propagating back into dendrites trigger changes in efficacy of single-axon synapses between layer V pyramidal neurons* in *Soc. Neurosci. Abstr* **21** (1995), 2007.
3. Markram, H., Lübke, J., Frotscher, M. & Sakmann, B. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* **275**, 213–215 (1997).
4. Gerstner, W., Kempter, R., van Hemmen, J. L. & Wagner, H. A neuronal learning rule for sub-millisecond temporal coding. *Nature* **383**, 76–78 (1996).
5. Bliss, T. V. & Lømo, T. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of physiology* **232**, 331–356 (1973).
6. Bishop, C. M. *Neural networks for pattern recognition* (Oxford university press, 1995).
7. Werbos, P. Beyond regression: New tools for prediction and analysis in the behavioral sciences (1974).
8. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. *Learning internal representations by error propagation* tech. rep. (DTIC Document, 1985).
9. LeCun, Y. in *Proceedings of Cognitiva 85, Paris, France* (1985).
10. Parker, D. B. Learning logic (1985).
11. Hannun, A. *et al.* Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567* (2014).
12. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *Imagenet classification with deep convolutional neural networks* in *Advances in neural information processing systems* (2012), 1097–1105.
13. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 770–778.
14. Vaswani, A. *et al.* *Attention is all you need* in *Advances in Neural Information Processing Systems* (2017), 6000–6010.
15. Oord, A. v. d., Kalchbrenner, N. & Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759* (2016).
16. Van den Oord, A. *et al.* Wavenet: A generative model for raw audio (2016).
17. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N. & Wu, Y. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410* (2016).
18. Oh, J., Guo, X., Lee, H., Lewis, R. L. & Singh, S. *Action-conditional video prediction using deep networks in atari games* in *Advances in Neural Information Processing Systems* (2015), 2863–2871.
19. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).

20. Silver, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
21. Silver, D. *et al.* Mastering the game of go without human knowledge. *Nature* **550**, 354 (2017).
22. Moravčík, M. *et al.* Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* **356**, 508–513 (2017).
23. Gilbert, C. D. & Li, W. Top-down influences on visual processing. *Nature Reviews Neuroscience* **14**, 350–363 (2013).
24. Tong, F. Cognitive neuroscience: primary visual cortex and visual awareness. *Nature Reviews Neuroscience* **4**, 219 (2003).
25. Grossberg, S. Competitive learning: From interactive activation to adaptive resonance. *Cognitive science* **11**, 23–63 (1987).
26. Marr, D. Simple memory: a theory for archicortex. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* **262**, 23 (1971).
27. Hinton, G. E. & McClelland, J. L. *Learning representations by recirculation in Neural information processing systems* (1988), 358–366.
28. Crick, F. The recent excitement about neural networks. *Nature* **337**, 129 (1989).
29. Roelfsema, P. R. & Holtmaat, A. Control of synaptic plasticity in deep cortical networks. *Nature Reviews Neuroscience* **19**, 166 (2018).
30. Whittington, J. C. & Bogacz, R. Theories of error back-propagation in the brain. *Trends in cognitive sciences* (2019).
31. Almeida, L. B. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment in *Artificial neural networks* (1990), 102–111.
32. Pineda, F. J. Generalization of back-propagation to recurrent neural networks. *Physical review letters* **59**, 2229 (1987).
33. Pineda, F. J. Dynamics and architecture for neural computation. *Journal of Complexity* **4**, 216–245 (1988).
34. O’Reilly, R. C. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural computation* **8**, 895–938 (1996).
35. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. A learning algorithm for Boltzmann machines. *Cognitive science* **9**, 147–169 (1985).
36. Hinton, G. E., Dayan, P., Frey, B. J. & Neal, R. M. The “wake-sleep” algorithm for unsupervised neural networks. *Science* **268**, 1158 (1995).
37. Movellan, J. R. *Contrastive Hebbian learning in the continuous Hopfield model in Connectionist models: Proceedings of the 1990 summer school* (1991), 10–17.
38. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M. & Huang, F. A tutorial on energy-based learning. *Predicting structured data* **1** (2006).
39. Xie, X. & Seung, H. S. Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural computation* **15**, 441–454 (2003).
40. Bengio, Y. How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv preprint arXiv:1407.7906* (2014).

41. Lee, D.-H., Zhang, S., Fischer, A. & Bengio, Y. *Difference target propagation* in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2015), 498–515.
42. Mazzoni, P., Anderson, R. A. & Jordan, M. I. A more biologically plausible learning rule for neural networks. *Proceedings of the National Academy of Sciences* **88**, 4433–4437 (1991).
43. Seung, H. S. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* **40**, 1063–1073 (2003).
44. Werfel, J., Xie, X. & Seung, H. S. Learning curves for stochastic gradient descent in linear feedforward networks. *Neural computation* **17**, 2699–2718 (2005).
45. Spall, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control* **37**, 332–341 (1992).
46. Williams, R. J. in *Reinforcement Learning* 5–32 (Springer, 1992).
47. Flower, B. & Jabri, M. *Summed weight neuron perturbation: An $O(n)$ improvement over weight perturbation* in *Advances in neural information processing Systems* (1993), 212–219.
48. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).
49. Deisenroth, M. P., Neumann, G., Peters, J., *et al.* A survey on policy search for robotics. *Foundations and Trends® in Robotics* **2**, 1–142 (2013).
50. Lillicrap, T. P. *et al.* Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).
51. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagation errors. *Nature* **323**, 533–536 (1986).
52. Andersen, P., Sundberg, S., Sveen, O., Swann, J. & Wigström, H. Possible mechanisms for long-lasting potentiation of synaptic transmission in hippocampal slices from guinea-pigs. *The Journal of Physiology* **302**, 463–482 (1980).
53. Guillery, R. & Sherman, S. M. Thalamic relay functions and their role in corticocortical communication: generalizations from the visual system. *Neuron* **33**, 163–175 (2002).
54. Sherman, S. M. & Guillery, R. Distinct functions for direct and transthalamic corticocortical connections. *Journal of neurophysiology* **106**, 1068–1077 (2011).
55. Viaene, A. N., Petrof, I. & Sherman, S. M. Properties of the thalamic projection from the posterior medial nucleus to primary and secondary somatosensory cortices in the mouse. *Proceedings of the National Academy of Sciences* **108**, 18156–18161 (2011).
56. Abdelghani, M., Lillicrap, T. & Tweed, D. Sensitivity Derivatives for Flexible Sensorimotor Learning. *Neural Computation* **20**, 2085–2111 (2008).
57. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications* **7** (2016).
58. Cadieu, C. F. *et al.* Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput Biol* **10**, e1003963 (2014).
59. Yamins, D. L. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences* **111**, 8619–8624 (2014).
60. Elston, G. N. Cortex, cognition and the cell: new insights into the pyramidal neuron and prefrontal function. *Cerebral Cortex* **13**, 1124–1138 (2003).

61. Oh, S. W. *et al.* A mesoscale connectome of the mouse brain. *Nature* **508**, 207 (2014).
62. Jiang, X. *et al.* Principles of connectivity among morphologically defined cell types in adult neocortex. *Science* **350**, aac9462 (2015).
63. Tasic, B. *et al.* Shared and distinct transcriptomic cell types across neocortical areas. *BioRxiv*, 229542 (2017).
64. Mountcastle, V. B. *et al.* Modality and topographic properties of single neurons of cat's somatic sensory cortex. *J neurophysiol* **20**, 408–434 (1957).
65. Mountcastle, V. B., Motter, B., Steinmetz, M. & Sestokas, A. Common and differential effects of attentive fixation on the excitability of parietal and prestriate (V4) cortical visual neurons in the macaque monkey. *The Journal of neuroscience* **7**, 2239–2255 (1987).
66. Douglas, R. J., Martin, K. A. & Whitteridge, D. A canonical microcircuit for neocortex. *Neural computation* **1**, 480–488 (1989).
67. Bastos, A. M. *et al.* Canonical microcircuits for predictive coding. *Neuron* **76**, 695–711 (2012).
68. Zipser, D. & Andersen, R. A. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* **331**, 679–684 (1988).
69. Lillicrap, T. P. & Scott, S. H. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron* **77**, 168–179 (2013).
70. Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput Biol* **10**, e1003915 (2014).
71. Kriegeskorte, N. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science* **1**, 417–446 (2015).
72. Wenliang, L. K. & Seitz, A. R. Deep neural networks for modeling visual perceptual learning. *Journal of Neuroscience* **38**, 6028–6044 (2018).
73. Pinto, N., Cox, D. D. & DiCarlo, J. J. Why is real-world visual object recognition hard? *PLoS Comput Biol* **4**, e27 (2008).
74. Freeman, J. & Simoncelli, E. P. Metamers of the ventral stream. *Nature neuroscience* **14**, 1195–1201 (2011).
75. Ullman, S., Assif, L., Fetaya, E. & Harari, D. Atoms of recognition in human and computer vision. *Proceedings of the National Academy of Sciences* **113**, 2744–2749 (2016).
76. Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V. & McDermott, J. H. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron* **98**, 630–644 (2018).
77. Mitchell, M. *An introduction to genetic algorithms* (MIT press, 1998).
78. Saxe, A. M. *Deep linear neural networks: A theory of learning in the brain and mind* PhD thesis (Stanford University, 2015).
79. Zmarz, P. & Keller, G. B. Mismatch receptive fields in mouse visual cortex. *Neuron* **92**, 766–772 (2016).
80. Issa, E. B., Cadieu, C. F. & DiCarlo, J. J. Neural dynamics at successive stages of the ventral visual stream are consistent with hierarchical error signals. *eLife* **7**, e42870 (2018).

81. Zeiler, M. D. & Fergus, R. *Visualizing and understanding convolutional networks* in *European conference on computer vision* (2014), 818–833.
82. Schultz, W., Dayan, P. & Montague, P. R. A neural substrate of prediction and reward. *Science* **275**, 1593–1599 (1997).
83. Zipser, D. & Rumelhart, D. The Neurobiological Significance of the New Learning Models. *Computational Neuroscience* (ed. Schwartz E.L. Ch. 15), 192–200 (1990).
84. Stork, D. G. *Is Backpropagation Biologically Plausible?* in *International Joint Conference on Neural Networks* **2** (1989), 241–246.
85. Brandt, R. D. & Lin, F. *Can supervised learning be achieved without explicit error back-propagation?* in *Neural Networks, 1996., IEEE International Conference on* **1** (1996), 300–305.
86. Brandt, R. D. & Lin, F. *Supervised learning in neural networks without feedback network* in *Intelligent Control, 1996., Proceedings of the 1996 IEEE International Symposium on* (1996), 86–90.
87. Oztas, E. Neuronal tracing. *Neuroanatomy* **2**, 5 (2003).
88. Harris, K. D. Stability of the fittest: organizing learning through retroaxonal signals. *Trends in Neurosciences* **31**, 130–136 (2008).
89. Venkateswararao, L. C. *Adaptive optimal-control algorithms for brainlike networks* PhD thesis (University of Toronto, 2010).
90. Hinton, G. The ups and downs of Hebb synapses. *Canadian Psychology/Psychologie canadienne* **44**, 10 (2003).
91. Kolen, J. F. & Pollack, J. B. *Back-propagation Without Weight Transport* in *IEEE World Congress on Computational Intelligence* **3** (Orlando, Florida, 1994), 1375–1380.
92. Körding, K. P. & König, P. Supervised and unsupervised learning with two sites of synaptic integration. *Journal of computational neuroscience* **11**, 207–215 (2001).
93. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random feedback weights support learning in deep neural networks. *arXiv preprint arXiv:1411.0247* (2014).
94. Nøkland, A. *Direct Feedback Alignment Provides Learning in Deep Neural Networks* in *Advances In Neural Information Processing Systems* (2016), 1037–1045.
95. Guergiev, J., Lillicrap, T. P. & Richards, B. A. Deep learning with segregated dendrites. *arXiv preprint arXiv:1610.00161* (2016).
96. Liao, Q., Leibo, J. Z. & Poggio, T. How important is weight symmetry in backpropagation? *arXiv preprint arXiv:1510.05067* (2015).
97. Samadi, A., Lillicrap, T. P. & Tweed, D. B. Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural computation* (2017).
98. Moskovitz, T. H., Litwin-Kumar, A. & Abbott, L. Feedback alignment in deep convolutional networks. *arXiv preprint arXiv:1812.06488* (2018).
99. Xiao, W., Chen, H., Liao, Q. & Poggio, T. Biologically-plausible learning algorithms can scale to large datasets. *arXiv preprint arXiv:1811.03567* (2018).
100. Amit, Y. Deep learning with asymmetric connections and Hebbian updates. *Frontiers in computational neuroscience* **13**, 18 (2019).

101. Bartunov, S. *et al.* Assessing the scalability of biologically-motivated deep learning algorithms and architectures in *Advances in Neural Information Processing Systems* (2018), 9390–9400.
102. Akrou, M., Wilson, C., Humphreys, P. C., Lillicrap, T. & Tweed, D. Using Weight Mirrors to Improve Feedback Alignment. *arXiv preprint arXiv:1904.05391* (2019).
103. Pascanu, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks in *International conference on machine learning* (2013), 1310–1318.
104. Coesmans, M., Weber, J. T., De Zeeuw, C. I. & Hansel, C. Bidirectional parallel fiber plasticity in the cerebellum under climbing fiber control. *Neuron* **44**, 691–700 (2004).
105. Yang, Y. & Lisberger, S. G. Purkinje-cell plasticity and cerebellar motor learning are graded by complex-spike duration. *Nature* **510**, 529–532 (2014).
106. Li, W., Piëch, V. & Gilbert, C. D. Contour saliency in primary visual cortex. *Neuron* **50**, 951–962 (2006).
107. Motter, B. C. Focal attention produces spatially selective processing in visual cortical areas V1, V2, and V4 in the presence of competing stimuli. *Journal of neurophysiology* **70**, 909–919 (1993).
108. Moran, J. & Desimone, R. Selective attention gates visual processing in the extrastriate cortex. *Frontiers in cognitive neuroscience* **229**, 342–345 (1985).
109. Spitzer, H., Desimone, R. & Moran, J. Increased attention enhances both behavioral and neuronal performance. *Science* **240**, 338–340 (1988).
110. Chelazzi, L., Miller, E. K. & Duncan, J. A neural basis for visual search in inferior temporal cortex. *Nature* **363**, 27 (1993).
111. Chelazzi, L., Miller, E. K., Duncan, J. & Desimone, R. Responses of neurons in macaque area V4 during memory-guided visual search. *Cerebral Cortex* **11**, 761–772 (2001).
112. Treue, S. & Maunsell, H. motion processing in cortical. *Nature* **382**, 8 (1996).
113. Luck, S. J., Chelazzi, L., Hillyard, S. A. & Desimone, R. Neural mechanisms of spatial selective attention in areas V1, V2, and V4 of macaque visual cortex. *Journal of neurophysiology* **77**, 24–42 (1997).
114. Ito, M. & Gilbert, C. D. Attention modulates contextual influences in the primary visual cortex of alert monkeys. *Neuron* **22**, 593–604 (1999).
115. McAdams, C. J. & Maunsell, J. H. Effects of attention on orientation-tuning functions of single neurons in macaque cortical area V4. *The Journal of Neuroscience* **19**, 431–441 (1999).
116. Reynolds, J. H. & Desimone, R. Interacting roles of attention and visual salience in V4. *Neuron* **37**, 853–863 (2003).
117. Abbott, L. F., Varela, J., Sen, K. & Nelson, S. Synaptic depression and cortical gain control. *Science* **275**, 221–224 (1997).
118. Chance, F. S., Abbott, L. & Reyes, A. D. Gain modulation from background synaptic input. *Neuron* **35**, 773–782 (2002).
119. Girard, P., Hupé, J. & Bullier, J. Feedforward and feedback connections between areas V1 and V2 of the monkey have similar rapid conduction velocities. *Journal of neurophysiology* **85**, 1328–1331 (2001).

120. De Pasquale, R. & Sherman, S. M. Synaptic properties of corticocortical connections between the primary and secondary visual cortical areas in the mouse. *Journal of Neuroscience* **31**, 16494–16506 (2011).
121. Kosslyn, S. M. & Thompson, W. L. When is early visual cortex activated during visual mental imagery? *Psychological bulletin* **129**, 723 (2003).
122. Bridge, H., Harrold, S., Holmes, E. A., Stokes, M. & Kennard, C. Vivid visual mental imagery in the absence of the primary visual cortex. *Journal of neurology* **259**, 1062–1070 (2012).
123. Manita, S. *et al.* A top-down cortical circuit for accurate sensory perception. *Neuron* **86**, 1304–1316 (2015).
124. Fyall, A. M., El-Shamayleh, Y., Choi, H., Shea-Brown, E. & Pasupathy, A. Dynamic representation of partially occluded objects in primate prefrontal and visual cortex. *ELife* **6**, e25784 (2017).
125. Mignard, M. & Malpeli, J. G. Paths of information flow through visual cortex. *Science* **251**, 1249–1252 (1991).
126. Markov, N. T. & Kennedy, H. The importance of being hierarchical. *Current opinion in neurobiology* **23**, 187–194 (2013).
127. Ahissar, M. & Hochstein, S. The reverse hierarchy theory of visual perceptual learning. *Trends in cognitive sciences* **8**, 457–464 (2004).
128. Lee, T. S. & Mumford, D. Hierarchical Bayesian inference in the visual cortex. *JOSA A* **20**, 1434–1448 (2003).
129. Lewicki, M. S. & Sejnowski, T. J. Bayesian unsupervised learning of higher order structure. *Advances in neural information processing systems*, 529–535 (1997).
130. Knill, D. C. & Richards, W. *Perception as Bayesian inference* (Cambridge University Press, 1996).
131. Dayan, P., Hinton, G. E., Neal, R. M. & Zemel, R. S. The helmholtz machine. *Neural computation* **7**, 889–904 (1995).
132. Von Helmholtz, H. & Southall, J. P. C. *Treatise on physiological optics* (Courier Corporation, 2005).
133. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. in *Readings in Computer Vision* 522–533 (Elsevier, 1987).
134. Whittington, J. C. & Bogacz, R. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation* (2017).
135. Sacramento, J., Costa, R. P., Bengio, Y. & Senn, W. Dendritic error backpropagation in deep cortical microcircuits. *arXiv preprint arXiv:1801.00062* (2017).
136. Sacramento, J., Costa, R. P., Bengio, Y. & Senn, W. *Dendritic cortical microcircuits approximate the backpropagation algorithm* in *Advances in Neural Information Processing Systems* (2018), 8721–8732.
137. Scellier, B. & Bengio, Y. Towards a biologically plausible backprop. *arXiv preprint arXiv:1602.05179* **914** (2016).

138. Scellier, B. & Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience* **11** (2017).
139. Hinton, G. How to do backpropagation in a brain. *NIPS 2007 Deep Learning Workshop* (2007).
140. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. *Extracting and composing robust features with denoising autoencoders* in *Proceedings of the 25th international conference on Machine learning* (2008), 1096–1103.
141. Smolensky, P. *Information processing in dynamical systems: Foundations of harmony theory* tech. rep. (1986).
142. LeCun, Y. in *Disordered systems and biological organization* 233–240 (Springer, 1986).
143. LeCun, Y. *Modèles connexionnistes de l'apprentissage* PhD thesis (PhD thesis, These de Doctorat, Université Paris 6, 1987).
144. Coogan, T. & Burkhalter, A. Conserved patterns of cortico-cortical connections define areal hierarchy in rat visual cortex. *Experimental brain research* **80**, 49–53 (1990).
145. D'Souza, R. D. & Burkhalter, A. A laminar organization for selective cortico-cortical communication. *Frontiers in neuroanatomy* **11**, 71 (2017).
146. Wimmer, V. C., Bruno, R. M., De Kock, C. P., Kuner, T. & Sakmann, B. Dimensions of a projection column and architecture of VPM and POM axons in rat vibrissal cortex. *Cerebral cortex* **20**, 2265–2276 (2010).
147. Williams, L. E. & Holtmaat, A. Higher-order thalamocortical inputs gate synaptic long-term potentiation via disinhibition. *Neuron* **101**, 91–102 (2019).
148. Larkum, M. E., Zhu, J. J. & Sakmann, B. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature* **398**, 338–341 (1999).
149. Gordon, U., Polsky, A. & Schiller, J. Plasticity compartments in basal dendrites of neocortical pyramidal neurons. *Journal of Neuroscience* **26**, 12717–12726 (2006).
150. Branco, T., Clark, B. A. & Häusser, M. Dendritic discrimination of temporal input sequences in cortical neurons. *Science* **329**, 1671–1675 (2010).
151. Branco, T. & Häusser, M. Synaptic integration gradients in single cortical pyramidal cell dendrites. *Neuron* **69**, 885–892 (2011).
152. Losonczy, A., Makara, J. K. & Magee, J. C. Compartmentalized dendritic plasticity and input feature storage in neurons. *Nature* **452**, 436–441 (2008).
153. Polsky, A., Mel, B. W. & Schiller, J. Computational subunits in thin dendrites of pyramidal cells. *Nature neuroscience* **7**, 621–627 (2004).
154. Urbanczik, R. & Senn, W. Learning by the dendritic prediction of somatic spiking. *Neuron* **81**, 521–528 (2014).
155. Naud, R. & Sprekeler, H. Burst Ensemble Multiplexing: A Neural Code Connecting Dendritic Spikes with Microcircuits. *bioRxiv*, 143636 (2017).
156. Schiess, M., Urbanczik, R. & Senn, W. Somato-dendritic synaptic plasticity and error-backpropagation in active dendrites. *PLoS computational biology* **12**, e1004638 (2016).
157. Klausberger, T. & Somogyi, P. Neuronal diversity and temporal dynamics: the unity of hippocampal circuit operations. *Science* **321**, 53–57 (2008).

158. Sjöström, P. J. & Häusser, M. A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron* **51**, 227–238 (2006).
159. Richards, B. A. & Lillicrap, T. P. Dendritic solutions to the credit assignment problem. *Current opinion in neurobiology* **54**, 28–36 (2019).
160. Muller, S. Z., Zadina, A., Abbott, L. & Sawtell, N. Continual Learning in a Multi-Layer Network of an Electric Fish. *CELL-D-19-01360* (2019).
161. Bittner, K. C. *et al.* Conjunctive input processing drives feature selectivity in hippocampal CA1 neurons. *Nature neuroscience* **18**, 1133 (2015).
162. Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S. & Magee, J. C. Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science* **357**, 1033–1036 (2017).
163. Guerguiev, J., Lillicrap, T. P. & Richards, B. A. Towards deep learning with segregated dendrites. *Elife* **6**, e22901 (2017).
164. Kwag, J. & Paulsen, O. The timing of external input controls the sign of plasticity at local synapses. *Nature neuroscience* **12**, 1219 (2009).
165. Dale, H. *Pharmacology and nerve-endings* 1935.
166. Osborne, N. N. Is Dale’s principle valid? *Trends in Neurosciences* **2**, 73–75 (1979).
167. O’Donohue, T. L., Millington, W. R., Handelmann, G. E., Contreras, P. C. & Chronwall, B. M. On the 50th anniversary of Dale’s law: multiple neurotransmitter neurons. *Trends in Pharmacological Sciences* **6**, 305–308 (1985).
168. Draye, J.-P., Cheron, G., Libert, G. & Godaux, E. Emergence of clusters in the hidden layer of a dynamic recurrent neural network. *Biological cybernetics* **76**, 365–374 (1997).
169. De Kamps, M. & van der Velde, F. From artificial neural networks to spiking neuron populations and back again. *Neural Networks* **14**, 941–953 (2001).
170. Parisien, C., Anderson, C. H. & Eliasmith, C. Solving the problem of negative synaptic weights in cortical models. *Neural computation* **20**, 1473–1494 (2008).
171. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. & Bengio, Y. *Binarized neural networks* in *Advances in neural information processing systems* (2016), 4107–4115.
172. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15**, 1929–1958 (2014).
173. Bengio, Y., Mesnard, T., Fischer, A., Zhang, S. & Wu, Y. Stdp-compatible approximation of backpropagation in an energy-based model. *Neural computation* (2017).
174. Bengio, Y., Lamblin, P., Popovici, D. & Larochelle, H. *Greedy layer-wise training of deep networks* in *Advances in neural information processing systems* (2007), 153–160.
175. Ngiam, J. *et al.* *Multimodal deep learning* in *Proceedings of the 28th international conference on machine learning (ICML-11)* (2011), 689–696.
176. Srivastava, N., Mansimov, E. & Salakhutdinov, R. *Unsupervised learning of video representations using lstms* in *International conference on machine learning* (2015), 843–852.
177. Mikolov, T., Karafiát, M., Burget, L., Černocký, J. & Khudanpur, S. *Recurrent neural network based language model* in *Eleventh Annual Conference of the International Speech Communication Association* (2010).

178. Gemici, M. *et al.* Generative Temporal Models with Memory. *arXiv preprint arXiv:1702.04649* (2017).
179. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
180. Gregor, K., Danihelka, I., Graves, A., Rezende, D. J. & Wierstra, D. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623* (2015).
181. Rumelhart, D. E., McClelland, J. L., Group, P. R., *et al.* *Parallel distributed processing* (IEEE, 1988).
182. Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* **78**, 1550–1560 (1990).
183. Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**, 544–557 (2009).
184. Miconi, T. Biologically plausible learning in recurrent neural networks reproduces neural dynamics observed during cognitive tasks. *Elife* **6** (2017).
185. Bellec, G. *et al.* Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets. *arXiv preprint arXiv:1901.09049* (2019).
186. Ba, J., Hinton, G. E., Mnih, V., Leibo, J. Z. & Ionescu, C. *Using fast weights to attend to the recent past* in *Advances in Neural Information Processing Systems* (2016), 4331–4339.

Supplementary Information

Spiking neurons

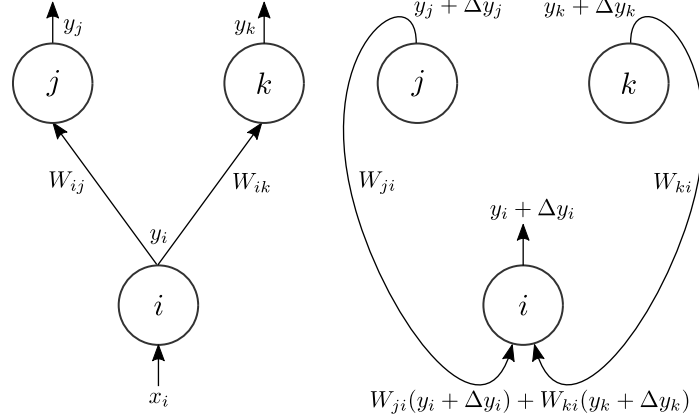
While neurons in the brain communicate using spikes, most artificial neural networks are trained using neurons that communicate real values. This discrepancy has sometimes been viewed as a stumbling block for linking backprop with learning in the brain^{28,95,97}. However, recent work in machine learning suggests this apparent issue may not present a significant impediment to understanding how cortex approximates backprop^{93,171,172}. In a spiking neuron, the spike train is often a noisy realization of the underlying firing rate. In this case, if errors are represented as activity differences, the post-synaptic term in the learning rule needs to measure changes in this underlying rate. It can do this by applying a derivative filter to the post-synaptic spike train. The derivative filter compares the firing rate just before a pre-synaptic spike with the firing rate just after a pre-synaptic spike. This looks exactly like spike-time dependent plasticity^{139,173}. Naturally, the output of the derivative filter will be a very noisy estimate of the change in the underlying firing rate, but stochastic gradient descent is extremely robust to noise, provided the noise is unbiased. Indeed, adding random noise to neural activities during training has been shown to greatly improve the ability of neural networks to generalize well to novel data¹⁷², so rather than viewing spikes as a clumsy way to convey an underlying firing rate, we can view them as a very effective regularizer that allows us to fit large neural nets to relatively small amounts of data.

Connections with unsupervised learning

A distinction is often made between backprop and unsupervised algorithms²⁵. However, this is a false dichotomy that likely arose from the fact that backprop was first developed in a supervised setting⁵¹. Unsupervised algorithms are characterized by the lack of output labels for targets, but there is no issue with employing backprop in these algorithms. Without output labels, learning may occur *within* a modality by trying to predict one part of an input from the remainder of the input^{15,140,174}. Unsupervised learning can also occur across modalities¹⁷⁵ or across time^{176–178}: i.e., where the future activity of a network is the target for the prediction. Not only is backprop compatible with unsupervised learning, it underlies the most powerful unsupervised algorithms developed to date^{15,179,180}. In short, the issue of effective learning across multiple layers exists in the case of unsupervised learning as well and backpropagation is well suited to the task.

Backpropagation-through-time

In this review we address the question of how the brain might learn across networks with multiple layers. We have not addressed the difficult issue of how the brain might optimize recurrent networks that process time-varying inputs. To learn from temporal data, artificial neural networks make extensive use of backpropagation-through-time^{181,182} (BPTT). It is much harder to see how BPTT could be implemented in cortex because each neuron must remember its activity value at many different time steps during the forward pass and then use these remembered activities to compute



Supplementary Figure 1: The two last layers of a simple feedforward network. The total input to unit i is x_i and its output is $y_i = f(x_i)$ where f is a smooth non-linear function. After an initial forward pass, the network's initial output vector (y_j, y_k) is compared with the target and is moved towards the target by a small amount, $(\Delta y_j, \Delta y_k)$, that is proportional to the difference. For linear output units with a quadratic loss or for logistic output units with a cross-entropy loss, this ensures that $\Delta y_j, \Delta y_k$ represent the derivatives of the loss with respect to the *total inputs*, x_j and x_k to the output units. To compute the derivatives in earlier layers, we can now make use of the following curious fact: If the perturbation in the *input* to a unit represents the derivative of the loss w.r.t. the *output* of the unit, the resulting perturbation of its *output* represents the derivative w.r.t. its *input*. The modified output vector is used to reconstruct the activity in the previous layer via backwards connections that have the same weights as the forward connections (i.e. $W_{ij} = W_{ji}$ and $W_{ik} = W_{ki}$). We assume that these weights have already been trained to be a perfect autoencoder, so if the output vector had not been perturbed, the top-down input of $W_{ij}y_j + W_{ik}y_k$ to unit i would produce the same output, y_i , as was computed on the forward pass. The small additional input $W_{ij}\Delta y_j + W_{ik}\Delta y_k$ will be converted into a small additional output which will be the additional input times the gradient of the non-linear function f . So, to first order, $\Delta y_i = dy_i/dx_i(W_{ij}\Delta y_j + W_{ik}\Delta y_k)$. This is exactly the derivative prescribed by backpropagation, so a perturbation in the output layer that represents the derivatives of the loss with respect to the inputs to that layer, causes a perturbation in the previous layer that represents the same quantity for the previous layer. This can be repeated for as many earlier layers as required. The learning rule is then to modify each incoming weight in proportion to the product of the pre-synaptic activity and the change in the post-synaptic activity.

weight updates during the subsequent backward pass. Rather than using BPTT, we suspect the cortex may rely on approximations such as eligibility traces^{183–185}, or approaches wherein fast temporary changes in synaptic weights are used to store recent hidden activity vectors¹⁸⁶. In the latter case, the cortex could do associative retrieval of relevant recent activity vectors in order to learn long-term dependencies without having to explicitly go back through the intermediate time-steps.