Research Article

ADVANCED PHOTONICS

Training neural networks with end-to-end optical backpropagation

James Spall,[a,b,†] Xianxin Guo,[a,b,*,†] and Alexander I. Lvovsky[a,b,*]
[a]University of Oxford, Clarendon Laboratory, Oxford, United Kingdom
[b]Lumai Ltd., Wood Centre for Innovation, Oxford, United Kingdom

Abstract. Optics is an exciting route for the next generation of computing hardware for machine learning, promising several orders of magnitude enhancement in both computational speed and energy efficiency. However, reaching the full capacity of an optical neural network (NN) necessitates that the computing be implemented optically not only for inference but also for training. The primary algorithm for network training is backpropagation, in which the calculation is performed in the order opposite to the information flow for inference. Although straightforward in a digital computer, the optical implementation of backpropagation has remained elusive, particularly because of the conflicting requirements for the optical element that implements the nonlinear activation function. We address this challenge for the first time, we believe, with a surprisingly simple scheme, employing saturable absorbers for the role of activation units. Our approach is adaptable to various analog platforms and materials and demonstrates the possibility of constructing NNs entirely reliant on analog optical processes for both training and inference tasks.

Keywords: optical computing; optical neural networks; machine learning.

Received Aug. 7, 2024; revised manuscript received Dec. 2, 2024; accepted for publication Dec. 11, 2024; published online Feb. 4, 2025.

[DOI: 10.1117/1.AP.7.1.016004]

1 Introduction

Machine learning, one of the most revolutionary scientific breakthroughs in the past decades, has completely transformed the technology landscape, enabling innovative applications in fields ranging from natural language processing to drug discovery. As the demand for increasingly sophisticated machine-learning models continues to escalate, there is a pressing need for faster and more energy-efficient computing solutions. In this context, analog computing has emerged as a promising alternative to traditional digital electronics.[1–7] A particularly exciting platform for analog neural networks (NNs) is optics, in which the interference and diffraction of light during propagation implement the linear part of every computational layer.[8,9]

Most of the current analog computing research and development is aimed at using the NN for inference.[8,10] Training such NNs, on the other hand, is a challenge. This is because the backpropagation algorithm,[11] the workhorse of training in digital NNs, requires the calculation to be performed in the order opposite to the information flow for inference, which is difficult to implement on an analog physical platform. Hence, analog models are typically trained offline (*in silico*), on a separate digital simulator, after which the parameters are transferred to the analog hardware. In addition to being slow and inefficient, this approach can lead to errors arising from imperfect simulation and systematic errors ("reality gap"). In optics, for example, these effects may result from dust, aberrations, spurious reflections, and inaccurate calibration.[12]

To enable learning in analog NNs, different approaches have been proposed and realized.[13] Several groups explored various "hardware-in-the-loop" schemes, in which, although the backpropagation was done *in silico*, the signal acquired from the analog NN operating in the inference regime was incorporated into the calculation of the feedback for optimizing the NN parameters.[12,14–20] This has partially reduced the training error but has not addressed the low speed and inefficiency of *in silico* training.

*Address correspondence to Xianxin Guo, xianxin.guo@lumai.co.uk; Alexander I. Lvovsky, alex.lvovsky@physics.ox.ac.uk

†These authors contributed equally to this work.

Recently, several optical neural networks (ONNs) were reported that were trained online (*in situ*) using methods alternative to backpropagation. Bandyopadhyay et al.[21] trained an ONN based on integrated photonic circuits using simultaneous perturbation stochastic approximation, i.e., randomly perturbing all ONN parameters and using the observed change of the loss function to approximate its gradient. Filipovich et al.[22] applied direct feedback alignment, wherein the error calculated at the output of the ONN is used to update the parameters of all layers. However, both these methods are inferior to backpropagation, as they take a much longer time to converge, especially for sufficiently deep ONNs.[23]

An optical implementation of the backpropagation algorithm was proposed by Hughes et al.,[24] and recently demonstrated experimentally,[25] showing that the training methods of current digital NNs can be applied to analog hardware. However, their scheme omitted a crucial step for optical implementation: backpropagation through nonlinear activation layers. Their method requires digital nonlinear activation and multiple opto-electronic interconversions inside the network, complicating the training process. Furthermore, the method applies only to a specific type of ONN that uses interferometer meshes for the linear layer and does not generalize to other ONN architectures. Complete implementation of the backpropagation algorithm in optics, through all the linear and nonlinear layers that can be generalized to many ONN systems, remains a highly challenging goal.

In this work, we address this long-standing challenge and present what we believe is the first complete optical implementation of the backpropagation algorithm in a two-layer ONN. The gradients of the loss function with respect to the NN parameters are calculated by light traveling through the system in the reverse direction. The main difficulty of all-optical training lies in the requirement that the nonlinear optical element used for the activation function needs to exhibit different properties for the forward and backward propagating signals. Fortunately, as demonstrated in our earlier theoretical work[26] and explained below, there does exist a group of nonlinear phenomena that exhibit the required set of properties with sufficient precision.

We optically train our ONNs to perform classification tasks, and our results surpass those trained with a conventional *in silico* method. Our optical training scheme can be further generalized to other platforms using different linear layers and analog activation functions, making it an ideal tool for exploring the vast potential of analog computing for training NNs. Optical backpropagation offers faster convergence in training compared with alternative algorithms,[21–23] delivers higher ONN inference accuracy than traditional digital methods, and unlocks the potential benefits of optical computing for training tasks.

1.1 Optical Training Algorithm

We consider a multilayer perceptron—a common type of NN that consists of multiple linear layers that establish weighted connections among neurons, interlaid by activation functions that enable the network to learn complex nonlinear functions. To train the NN, one presents it with a training set of labeled examples and iteratively adjusts the NN parameters (weights and biases) to find the correct mapping between the inputs and outputs.

The training steps are summarized in Fig. 1(d), and the complete analysis is presented in Note 1 in the Supplementary Material. The weight matrices, denoted $W^{(i)}$ for the i'th layer, are first initialized with random values. Each iteration of training starts by entering the input examples from the training set as input vectors $x = a^{(0)}$ into the NN and forward propagating through all of its layers. In every layer i, one performs a matrix–vector multiplication (MVM) of the weight matrix and the activation vector,

$$z^{(i)} = W^{(i)} \times a^{(i-1)}, \tag{1}$$

followed by element-wise application of the activation function $g(\cdot)$ to the resulting vector,

$$a^{(i)} = g(z^{(i)}). \tag{2}$$

The output $y = a^{(L)}$ of an L-layer NN allows one to compute the loss function $\mathcal{L}(y, t)$ that determines the difference between the network predictions y and ground-truth labels t from the training set. The backpropagation algorithm helps calculate the gradient of this loss function with respect to all the parameters in the network, through what is essentially an application of the chain rule of calculus. The network parameters are then updated using these gradients and optimization algorithms such as stochastic gradient descent. The training process is repeated until convergence.

The gradients we require are given by[11]

$$\frac{\partial \mathcal{L}}{\partial W^{(i)}} = \delta^{(i)} \otimes a^{(i-1)}, \tag{3}$$

where $\delta^{(i)}$ is referred to as the "error vector" at the i'th layer, and $\otimes$ denotes the outer product. The error vector is calculated as

$$\delta^{(i)} = (W^{(i+1)T} \times \delta^{(i+1)}) g'(z^{(i)}), \tag{4}$$

going through layers in reverse sequence. The expression for the error vector $\delta^{(L)}$ in the last layer depends on the choice of the loss function, but for the common loss functions of mean-squared error and cross-entropy (with an appropriate choice of activation function), it is simply the difference between the NN output and the label: $\delta^{(L)} = y - t$. Therefore, to calculate the gradients at each layer, we need one vector from the forward pass through the network (the activations) and one vector from the backward pass (the errors).

We see from Eq. (4) that the error backpropagation consists of two operations. First, we must perform an MVM, mirroring the feed-forward linear operation, Eq. (1). In an ONN, this can be done by light that propagates backward through the same linear optical arrangement.[27] The second operation consists of modulation of the MVM output by the activation function derivative and poses a notable challenge for optical implementation. This is because most optical media exhibit similar properties for forward and backward propagation. On the other hand, our application requires an optical element that is (1) nonlinear in the forward direction, (2) linear in the backward direction, and (3) modulates the backward light amplitude by the derivative of the forward activation function.

We have solved this challenge with our optical backpropagation protocol, which calculates the right-hand side of Eq. (4) entirely optically with no opto-electronic conversion or digital processing. The first component of our solution is the observation that many optical media exhibit nonlinear properties for strong optical fields but are approximately linear for weak

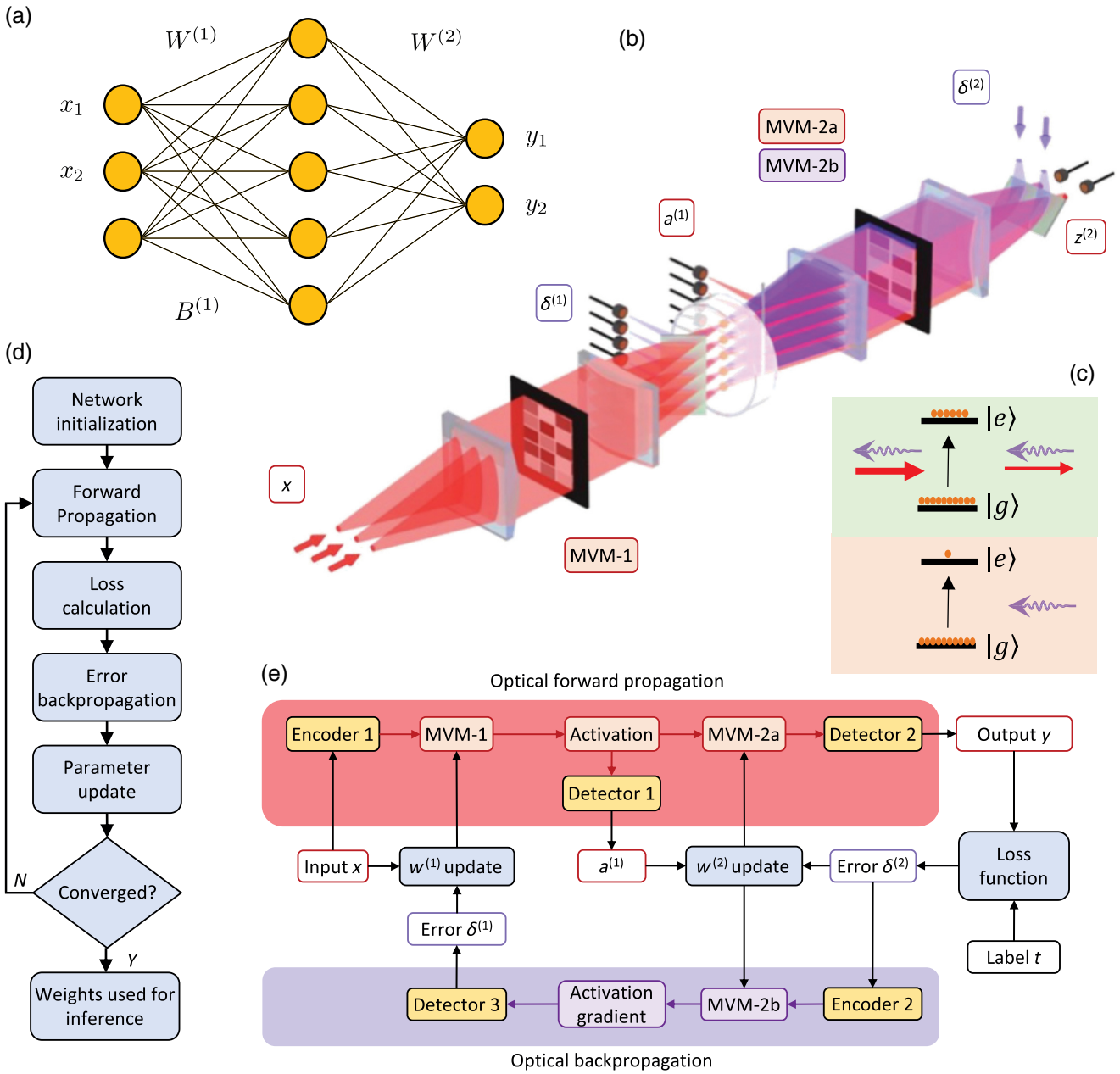

Fig. 1 Illustration of optical training. (a) Network architecture of the ONN used in this work, which consists of two fully connected linear layers and a hidden layer. (b) Simplified experimental schematic of the ONN. Each linear layer performs optical MVM with a cylindrical lens and an SLM that encodes the weight matrix. Hidden layer activations are computed using SA in an atomic vapor cell. Light propagates in both directions during optical training. (c) Working principle of SA activation. The forward beam (pump) is shown by solid red arrows and the backward (probe) by purple wavy arrows. The probe transmission depends on the strength of the pump and approximates the gradient of the SA function. For high forward intensity (top panel), a large portion of the atoms are excited to the upper level. Stimulated emission produced by these atoms largely compensates for the absorption due to the atoms at the ground level. For the weak pump (bottom panel), the excited level population is small, and the absorption is significant. (d) NN training procedure. (e) Optical training procedure. Both signal and error propagations in the two directions are fully implemented optically. Loss function calculation and parameter update are left for electronics without interrupting the optical information flow.

fields. Hence, we can satisfy conditions 1 and 2 by maintaining the back-injected beam at a much lower intensity level than the forward. Furthermore, there exists a set of nonlinear phenomena that also addresses the requirement (3). An example is saturable absorption (SA), whose nonlinear response of the forward incoming light field $z^{(i)}$ is

$$g(z^{(i)}) = \exp\left(-\frac{\alpha_0/2}{1+(z^{(i)})^2}\right) z^{(i)}, \tag{5}$$

where α_0 is the optical depth. The exponential term in Eq. (5) represents the transmissivity of the SA medium, which is set by the intensity of the forward incoming light. A weak beam in the backward direction experiences the same transmissivity, which is independent of the weak beam's intensity. Therefore, the SA medium is nonlinear for the forward beam and linear for the backward beam. Furthermore, the derivative of the forward nonlinear response is

$$g'(z^{(i)}) = \left(1+\frac{\alpha_0 (z^{(i)})^2}{(1+(z^{(i)})^2)^2}\right) \cdot \exp\left(-\frac{\alpha_0/2}{1+(z^{(i)})^2}\right). \tag{6}$$

As shown in our prior work,[26] the term before the central dot in Eq. (6) is approximately constant over a wide range of input values $z^{(i)}$, such that the transmissivity of the backward beam approximates the derivative $g'(z^{(i)})$ up to a constant factor. This constant factor can be absorbed into the learning rate, so the actual value is unimportant. Moreover, as revealed in our previous numerical simulation, during the ONN training, most $z^{(i)}$ values tend to distribute in a narrow range, and the approximation error is very small.

1.2 Multilayer ONN

Our ONN as shown in Figs. 1(a) and 1(b) is implemented in a free-space tabletop setting. The neuron values are encoded in the transverse spatial structure of the propagating light-field amplitude. Spatial light modulators (SLMs) are used to encode the input vectors and weight matrices. The NN consists of two fully connected linear layers implemented with optical MVM[28] following our previously demonstrated experimental design.[29] This design has a few characteristics that make it suitable for use in a deep NN. First, it is reconfigurable, so that both neuron values and network weights can be arbitrarily changed. Second, multiple MVM blocks can be cascaded to form a multilayer network, as the output of one MVM naturally forms the input of the next MVM. Using a coherent beam also allows us to encode both positive- and negative-valued weights. Finally, the MVM works in both directions, meaning the inputs and outputs are reversible, which is critical for the implementation of our optical backpropagation algorithm. The hidden layer activation between the two layers is implemented optically by means of SA in a rubidium atomic vapor cell [Fig. 1(c)].

2 Results

2.1 Linear Layers

We first set up the linear layers that serve as the backbone of our ONN, and we make sure that they work accurately and simultaneously in both directions—a highly challenging task that has never been achieved before, to our best knowledge. This involves three MVMs: first layer in the forward direction (MVM-1), second layer in both forward (MVM-2a) and backward (MVM-2b) directions. MVM-2b is the transpose of MVM-2a because the matrix elements are the same, but the fan-in directions are perpendicular for the forward and backward propagating beams. To characterize these MVMs, we apply random vectors and matrices and simultaneously measure the output of all three: the results for 300 random MVMs are presented in Fig. 2(a). To quantify the MVM performance, we define the signal-to-noise ratio (SNR, see Appendix for details). As illustrated by the histograms, MVM-1 has the greatest SNR of 14.9, and MVM-2a has a lower SNR of 7.1 as a result of noise accumulation from both layers and the reduced signal range. MVM-2b has a slightly lower SNR of 6.7 because the optical system is optimized for the forward direction. Comparing these experimental results with a simple numerical model, we estimate 1.3% multiplicative noise in our MVMs, which is small enough not to degrade the ONN performance.[12]

2.2 Nonlinearity

With the linear layers fully characterized, we now measure the response of the activation units in both directions. With the vapor cell placed in the setup and the laser tuned to resonance with the atomic transition, we pass the output of MVM-1 through the vapor cell in the forward direction. The response as presented in Fig. 2(b) shows strong nonlinearity. We fit the data with the theoretically expected SA transmissivity (see Supplementary Material for details), thereby finding the optical depth to be $\alpha_0 = 7.3$, which is sufficient to achieve high accuracy in ONNs.[26] The optical depth and the associated nonlinearity can be easily tuned to fit different network requirements by controlling the temperature of the vapor cell. In the backward direction, we pass weak probe beams through the vapor cell and measure the output. Both the forward and backward beams are simultaneously present in the vapor cell during the measurement.

In Fig. 2(c), we measure the effect of the forward amplitude $z^{(1)}$ on the transmission of the backward beam through the SA. The theoretical fit for these data—the expected backward transmissivity calculated from the physical properties of SA—is shown by the red curve. For comparison, the orange curve shows the rescaled exact derivative $g'(z^{(1)})$ of the SA function, which is the dependence required for the calculation of Eq. (4) of the training signal. Although the two curves are not identical, they both match the experimental data for a broad range of neuron values generated from the random MVM; hence, the setting is appropriate for training.

2.3 All-Optical Classification

After setting up the two-layer ONN, we perform end-to-end optical training and inference on classification tasks: distinguishing two classes of data points on a two-dimensional plane (Fig. 3). We implement a fully connected feed-forward architecture, with three input neurons, five hidden layer neurons, and two output neurons (Fig. 1). Two input neurons are used to encode the input data point coordinates (x_1, x_2), and the third input neuron of constant value is used to set the first layer bias. The class label is encoded by a "one-hot" vector $(0, 1)$ or $(1, 0)$, and we use categorical cross-entropy as the loss function.

We optically train the ONN on three 400-element datasets with different nonlinear boundary shapes, which we refer to

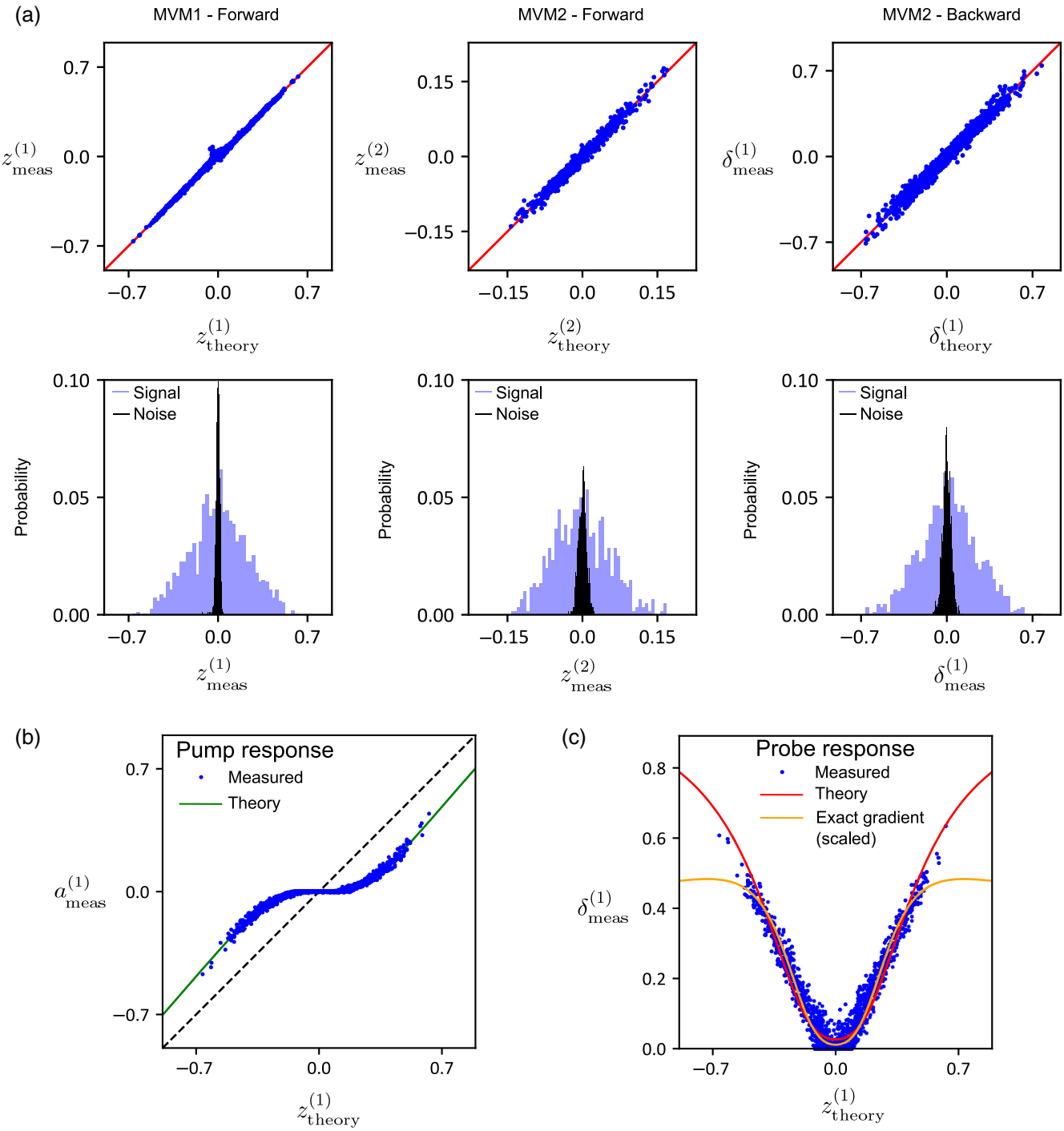

Fig. 2 Multilayer ONN characterization. (a) Scatterplots of measured-against-theory results for MVM-1 (first layer forward), MVM-2a (second layer forward), and MVM-2b (second layer backward). All three MVM results are taken simultaneously. Histograms of the signal and noise error for each MVM are displayed underneath. (b) First layer activations $a^{(1)}_{\text{meas}}$ measured after the vapor cell, plotted against the theoretically expected linear MVM-1 output $z^{(1)}_{\text{theory}}$ before the cell. The green line is a best-fit curve of the theoretical SA nonlinear function. (c) Amplitude of a weak constant probe passed backward through the vapor cell as a function of the pump $z^{(1)}_{\text{theory}}$, with a constant input probe. Measurements for both forward and backward beams are taken simultaneously.

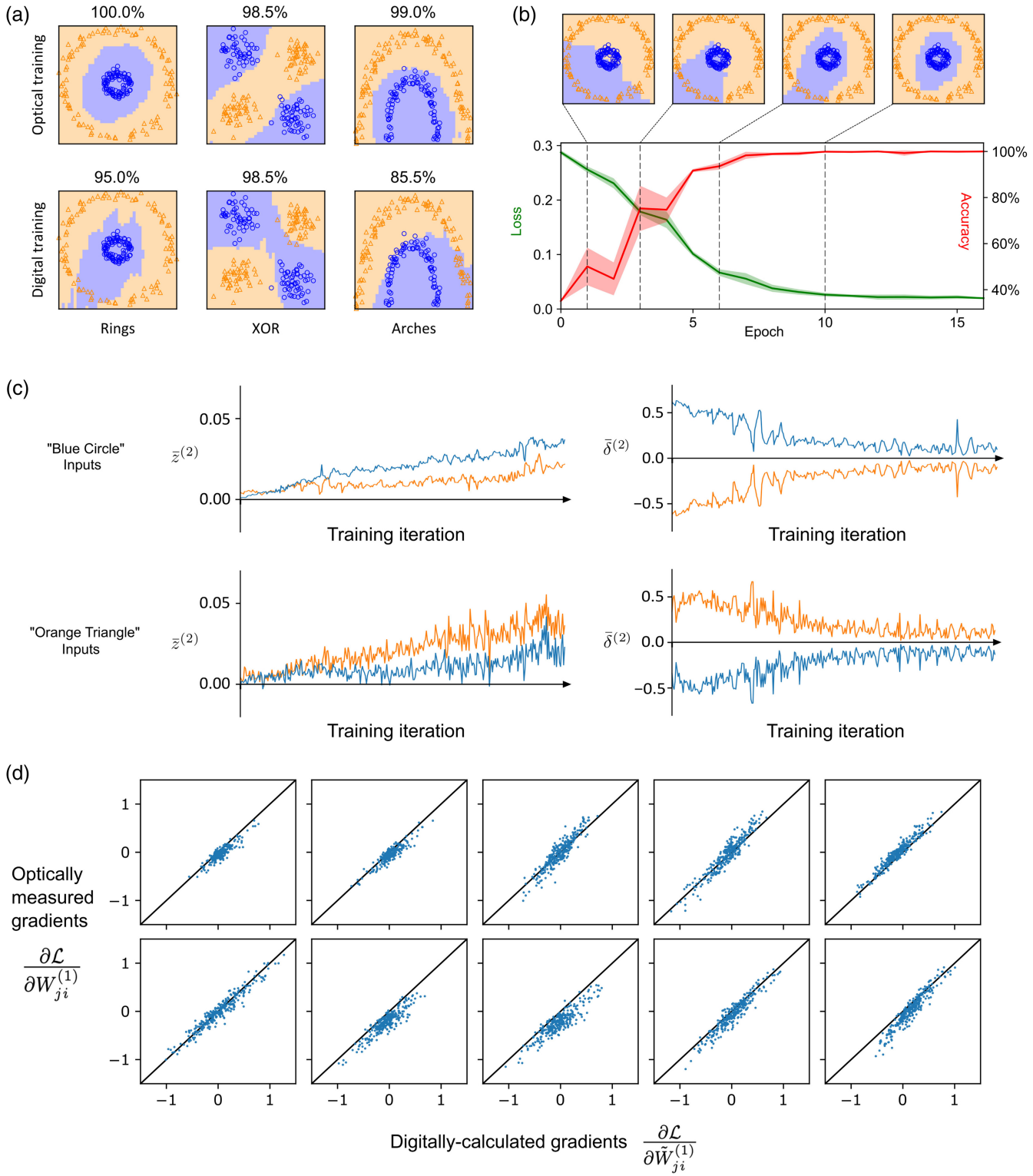

Fig. 3 Optical training performance. (a) Decision boundary charts of the ONN inference output for three different classification tasks, after the ONN has been trained optically (top) or *in silico* (bottom). (b) Learning curves of the ONN for classification of the "Rings" dataset, showing the mean and standard deviation of the validation loss and accuracy averaged over five repeated training runs. Shown above are decision boundary charts of the ONN output for the test set, after different epochs. (c) Evolution of output neuron values and output errors, for the training set inputs of the two classes. (d) Comparison between optically measured and digitally calculated gradients. Each panel shows gradients for each of the 10 weight matrix elements.

as "Rings," "XOR," and "Arches" [Fig. 3(a)]. Another 200 similar elements of each set are used for validation, i.e., to measure the loss and accuracy after each epoch of training. The test set consists of a uniform grid of equally spaced (x_1, x_2) values. The optical inference results for the test set are displayed in Fig. 3(a) with light purple and orange backgrounds, whereas the blue circles and orange triangles show the training set elements.

For all three datasets, each epoch consists of 20 minibatches, with a minibatch size of 20, and we use the Adam optimizer to update the weights and biases from the gradients. We tune hyperparameters such as learning rate and number of epochs to maximize network performance. Table 1 summarizes the network architecture and hyperparameters used for each dataset.

The optical training performance on the "Rings" dataset is shown in Fig. 3(b). We perform five repeated training runs and plot the loss and accuracy for the validation set after each epoch of training. To visualize how the network is learning the boundary between the two classes, we also run a test dataset after each epoch. Examples of the network output after 1, 3, 6, and 10 epochs are shown. We see that the ONN quickly learns the nonlinear boundary and gradually improves the accuracy to 100%. This indicates a strong optical nonlinearity in the system and a good gradient approximation in optical backpropagation. Details of the training procedure are provided in the Appendix, and results for the other two datasets in Note S3 in the Supplementary Material.

To better understand the optical training process, we explore the evolution of the output neuron and error vector values in Fig. 3(c). First, we plot the minibatch mean value of each output neuron, $\overline{z}_j^{(2)}$, for the inputs from the two classes separately in the upper and lower panels, over the course of the training iterations. We see the output neuron values diverge in opposite ways for the two classes, such that the inputs can be distinguished and correctly classified.

Second, we similarly plot the evolution of the minibatch mean output error, $\overline{\delta}^{(2)}$, for each neuron. This is calculated as the difference between the network output vector $a^{(2)}$ and the ground-truth label y, averaged over each minibatch. As expected, we see the output errors converge toward zero as the system learns the correct boundary.

To prove that the convergence is truly from optical gradient descent, we further compare the optically estimated gradients with digitally calculated gradients in Fig. 3(d). With the ONN well calibrated, optical gradients should track digital gradients. In Fig. 3(d), we see some deviation of optically estimated gradients from the digital gradients, which is a signature of imperfection in the physical system. In spite of the deviation, these plots still show a high degree of correlation between the optical and digital gradients for all 10 weight matrix elements. It shows that the ONN is optically trained in the correct direction according to digital calculation, subject to some perturbation by stochastic noise. The evolution of optical and digital gradients during the training process is further analyzed in Note 3 in the Supplementary Material.

2.4 Optical Training versus in silico Training

To demonstrate the optical training advantage, we perform *in silico* training of our ONN as a comparison. We digitally model our system with an NN of the equivalent architecture, including identical learning rate, number of epochs, and all other hyperparameters. The hidden layer nonlinearity and the associated gradient are given by the best-fit curve and theoretical probe response of Fig. 2(b). The trained weights are subsequently used for inference with our ONN. The top and bottom rows in Fig. 3(a) plot the network output of the test boundary set, after the system has been trained optically and digitally, respectively, for all three datasets. In all cases, the optically trained network achieves almost perfect accuracy, whereas the digitally trained network is clearly not optimized, with the network prediction not matching the data. This is further evidence of the already well-documented advantages of hardware-in-the-loop training schemes.

3 Discussion and Conclusion

According to simple estimates, optical implementation will enhance the energy efficiency of an NN by 3 orders of magnitude in comparison with its digital electronic counterpart. Our surprisingly simple and effective optical training scheme is capable of offering the same advantage factor to training as was previously promised for inference. It adds minimal computational overhead to the network because it does not require *in silico* simulation or intricate mapping of network parameters to physical device settings. Our method also imposes minimal hardware complexity on the system, as it requires only a few additional beam splitters and detectors to measure the activation and error values for parameter updates.

Our scheme can be generalized and applied to many other analog NNs with different physical implementations of the linear and nonlinear layers. We list a few examples in Table 2. Common optical linear operations include MVM, diffraction, and convolution. Compatible optical MVM examples include our free-space multiplier and photonic crossbar array,[30] as they are both bidirectional, in the sense that the optical field propagating backward through these arrangements gets multiplied by the transpose of the weight matrix. Diffraction naturally works in both directions; hence, diffractive NNs constructed using different programmable amplitude and phase masks also satisfy the requirements.[31] Optical convolution, achieved with the Fourier transform by means of a lens, and mean pooling, achieved through an optical low-pass filter, also work in both directions. Therefore, a convolutional NN can be optically trained as well. Detailed analysis of the generalization to these linear layers can be found in Note 4 in the Supplementary Material.

Table 1 Summary of network architecture and hyperparameters used in both optical and digital training.

Dataset	Input neurons	Hidden neurons	Output neurons	Learning rate	Epochs	Batches per epoch	Batch size
Rings				0.01	16		
XOR	2	5	2	0.005	30	20	20
Arches				0.01	25		

Table 2 Generalization of the optical training scheme.

Network layer	Function	Implementation example
Linear layer	MVM	Free-space optical multiplier and photonic crossbar array
	Diffraction	Programmable optical mask
	Convolution	Lens Fourier transform
Nonlinear layer	SA	Atomic vapor cell, semiconductor absorber, and graphene
	Saturable gain	EDFA, SOA, and Raman amplifier

EDFA, erbium-doped fiber amplifier; SOA, semiconductor optical amplifier.

Regarding the generalization to other nonlinearity choices, the critical requirement is the ability to acquire gradients during backpropagation. Our pump–probe method is compatible with multiple types of saturable effects, including SA and saturable gain.[32] Using saturable gain as the nonlinearity offers the added advantage of loss compensation in a deep network. This is important for scaling ONNs to real-world workloads, which may otherwise be limited to only a few layers if optical losses are not overcome. Importantly, both SA and saturable gain nonlinearities can be implemented not only in free space but also in integrated ONN settings.[33,34]

In our ONN training implementation, some computational operations remain digital, specifically the calculation of the last layer error $\delta^{(2)}$ and the outer product between the activation and error vectors, Eq. (3). Both these operations can be performed optically.[27] The error vector can in many cases be obtained by subtracting the ONN output from the label vector by way of destructive interference.[12] Interference can also be utilized to compute the outer product by fanning out the two vectors and overlapping them in a criss-cross fashion onto a pixelated photosensor. The data from that photosensor would then be digitized to determine the weight update on the SLM. Additional errors that can be brought about by this process can be minimized with careful calibration, and as we have shown previously, "physics-in-the-loop" training methods are quite robust to static additive and multiplicative errors; hence, these errors are not likely to be critical.[12] However, optical computation of the outer product requires a square array of photodetectors. This adds additional overhead in power consumption, system cost, and challenging optical implementation, so the value of including such a scheme in an experiment requires further study.

The compute performance and energy efficiency of our proof-of-principle experiment were not optimized to be comparable to its digital counterpart and were primarily bottlenecked by the slow refresh rate of the liquid crystal spatial light modulator (LC-SLM) and data communication time between optical devices and the host PC. However, because our scheme is applicable to a variety of implementations, the optical training algorithm is not limited to the devices used in this demonstration. Our optically trained ONN can therefore be scaled up to improve computing performance. In a previous experimental setup, we demonstrated an ONN with 100 neurons per layer.[12] In addition, an ONN capable of interconnecting 1000 neurons can be realized using high-resolution SLMs. ONN input data can be switched at speeds up to 100 GHz using advanced optical transceiver components. Therefore, computational speeds up to 10^{17} operations per second during inference are possible using 1000 high-speed optical transceivers as ONN input and output, fast-relaxation material for activation, and high-resolution SLMs as weight matrix displays. Our optical training method is compatible with such an optical system. The current bottleneck for training time is the slow speed of the SLM update, which may limit the training speed if the input batch sizes are not sufficiently large.

4 Appendix: Materials and Methods

4.1 Multilayer ONN

To construct the multilayer ONN, we connect two optical multipliers in series. For the first layer (MVM-1), the input neuron vector x is encoded into the field amplitude of a coherent beam using a digital micromirror device (DMD), DMD-1. This is a binary amplitude modulator, and every physical pixel is a micromirror that can reflect light at two angles representing 0 or 1. The model used was DLP Light Crafter 6500EVM, with a resolution of 1920×1080 pixels and a pixel pitch of $7.56\,\mu$m. By grouping 128 physical pixels in a row to form a "logical pixel," we are able to represent 7-bit positive-valued inputs on DMD-1, with each input value proportional to the number of binary physical pixels turned "on" in each logical pixel.

As MVM requires performing dot products of the input vector with every row of the matrix, we create multiple copies of the input vector pattern on DMD-1, replicating the logical pixel patterns vertically. We image DMD-1 onto the $W^{(1)}$ matrix mask—a phase-only LC-SLM, SLM-1—for element-wise multiplication. We use LC-SLMs with a phase grating modulation method[35] that enables arbitrary and accurate control of both field amplitude and phase of the coherent beam. The LC-SLM model used was Santec SLM-100, with a resolution of 1440×1050 pixels and a pixel pitch of $10.4\,\mu$m. This allows realizing real-valued weights (i.e., both positive and negative values) with 8-bit resolution.

The DMD-1 logical pixels are imaged to blocks of pixels on SLM-1 representing matrix elements using a simple $4f$ imaging telescope with unity magnification. The phase grating on SLM-1 was a 45-deg grating with both horizontal and vertical periods of 10 pixels, such that each matrix element was represented by a block of pixels covering at least six grating periods. Figures S1–S4 in Note 2 in the Supplementary Material provide more experimental details.

The MVM-1 result $z^{(1)}$ is obtained by summing the element-wise products encoded in the optical field after SLM-1, using a cylindrical lens (first optical "fan-in"), and passing 50% of the zero spatial frequency component of the beam through a 60-μm wide slit.

The beam passes through a rubidium vapor cell to apply the activation function, such that immediately after the cell the beam encodes the hidden layer activation vector, $a^{(1)}$. The spatial profile of modes in the cell is determined by diffraction on the slit and is hence independent of the number of physical pixels in each logical pixel (which encodes the input vector element), so the nonlinearity acts on a precise and consistent linear input. The beam continues to propagate and becomes the input for the second linear layer. Another cylindrical lens is used to expand the beam (first optical "fan-out"), before modulation by the second weight matrix mask, SLM-2. Finally, summation by a third

cylindrical lens (second optical "fan-in") completes the second MVM in the forward direction (MVM-2a), and the final beam profile encodes $z^{(2)}$.

To read out the activation vectors required for the optical training, we insert beam splitters at the output of each MVM to tap off a small portion of the beam. The real-valued vectors are measured by high-speed cameras, using coherent detection techniques detailed in Note 2 in the Supplementary Material.

At the output layer of the ONN, we use a digital softmax function to convert the output values into probabilities and calculate the loss function and output error vector, which initiates the optical backpropagation.

4.2 Optical Backpropagation

The output error vector, $\delta^{(2)}$, is encoded in the backward beam using DMD-2 to modulate a beam obtained from the same laser as the forward propagating beam. The backward beam is introduced to the system through one of the arms of the beam splitter placed at the output of MVM-2a and carefully aligned to overlap with the forward beam. SLM-2 performs element-wise multiplication by the transpose of the second weight matrix. The cylindrical lens that performs "fan-out" for the forward beam performs "fan-in" for the backward beam into a slit, completing the second layer backward MVM (MVM-2b). Passing through the vapor cell modulates the backward beam by the transmissivity set by the intensity of the forward signal, which, to a close approximation, is the derivative of the activation function. The beam then encodes the hidden layer error vector $\delta^{(1)}$. Another beam splitter and camera are used to tap off the backward beam and measure the result.

In our experiment, different areas of a single DMD were used as DMD-1 and DMD-2. The entire DMD area is mapped to SLM-1. The area of SLM-1 mapped by DMD-1 region is used to encode the weight matrix, whereas the area of SLM-1 mapped by DMD-2 is used to encode the sign of the error vector. The forward and backward beams are separated by a pick-off mirror after SLM-1. A schematic of this setup is provided in Fig. S2(a) in Note 2 in the Supplementary Material.

Each training iteration consists of optically measuring all of $a^{(1)}$, $z^{(2)}$, and $\delta^{(1)}$. These vectors are used, along with the inputs $x = a^{(0)}$ to calculate the weight gradients according to Eq. (3) and weight updates, which are then applied to the LC-SLMs. This process is repeated for all the minibatches until the network converges.

4.3 SA Activation

The cell with atomic rubidium vapor is heated to 70 deg by a simple heating jacket and temperature controller. The laser wavelength is locked to the D_2 transition at 780 nm. The optical depth as measured with the pump beam is 8.6, and the saturation power of each hidden neuron with a beam size of 60 μm$\times$ 120 μm is measured to be 3 μW. The power of the forward propagating beam is adjusted to ensure the beam at the vapor cell is intense enough to saturate the absorption, whereas the maximum power of the backward propagating beam is attenuated to $\sim$2% of the maximum forward beam power to ensure a linear response when passing through the cell in the presence of a uniform pump.

In the experiment, the backward probe response does not match perfectly with the simple two-level atomic model, due to two factors.

First, the probe does not undergo 100% absorption, even with the pump turned off. Second, a strong pump beam causes the atoms to fluoresce in all directions, including along the backward probe path. Therefore, the backward signal has a background offset proportional to the forward signal. To compensate for these issues, three measurements are taken to determine the probe response $\delta^{(1)}$ for each training iteration: pump only, probe only, and both pump and probe. In this way, the background terms due to pump fluorescence and unabsorbed probe could be negated. Further details on the series of intensity measurements made for each training iteration are provided in Note 2 in the Supplementary Material.

Finally, using a single vapor cell to perform nonlinear activation on all hidden layer vector elements limited the achievable hidden layer dimension, due to cross talk among modes. We were able to accommodate five individual modes without substantially increasing the nonlinear activation noise due to cross talk. A greater number of modes could be accommodated by encoding the activation vector in both transverse dimensions or by dividing the cell into multiple physically separated microcells. Alternatively, one can use a different activation mechanism altogether, e.g., coupling each hidden layer mode to an erbium-doped fiber amplifier.

More experimental details are available in the thesis by Dr. James Spall.[36]

Disclosures

The authors declare that they have no competing interests.

Code and Data Availability

Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Author Contributions

X.G. and A.L. conceived the experiment. J.S. carried out the experiment and performed the data analysis. All the authors jointly prepared the paper. This work was done under the supervision of A.L.

Supplemental Documentation

Aside from the Supplementary Material, the following video is available: Time-lapse of one complete end-to-end optical training process for the Rings dataset, Video 1, MP4, 3.27 MB [URL: https://doi.org/10.1117/1.AP.7.1.016004.s1].

Acknowledgments

This work was supported by the Innovate UK Smart (Grant No. 10043476). X.G. acknowledges support from the Royal Commission for the Exhibition of 1851 Research Fellowship.

References

1. K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature* **575**(7784), 607–617 (2019).
2. P. Yao et al., "Fully hardware-implemented memristor convolutional neural network," *Nature* **577**(7792), 641–646 (2020).
3. X. Xu et al., "11 TOPS photonic convolutional accelerator for optical neural networks," *Nature* **589**(7840), 44–51 (2021).

4. J. Feldmann et al., "Parallel convolutional processing using an integrated photonic tensor core," *Nature* **589**(7840), 52–58 (2021).
5. A. Sludds et al., "Delocalized photonic deep learning on the internet's edge," *Science* **378**(6617), 270–276 (2022).
6. X. Lin et al., "All-optical machine learning using diffractive deep neural networks," *Science* **361**(6406), 1004–1008 (2018).
7. Y. Shen et al., "Deep learning with coherent nanophotonic circuits," *Nat. Photonics* **11**(7), 441–446 (2017).
8. B. J. Shastri et al., "Photonics for artificial intelligence and neuromorphic computing," *Nat. Photonics* **15**(2), 102–114 (2021).
9. J. Wu et al., "Analog optical computing for artificial intelligence," *Engineering* **10**(1), 133–145 (2022).
10. G. Wetzstein et al., "Inference in artificial intelligence with deep optics and photonics," *Nature* **588**(7836), 39–47 (2020).
11. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**(7553), 436–444 (2015).
12. J. Spall, X. Guo, and A. I. Lvovsky, "Hybrid training of optical neural networks," *Optica* **9**(7), 803–811 (2022).
13. S. M. Buckley et al., "Photonic online learning: a perspective," *Nanophotonics* **12**, 833–845 (2023).
14. L. G. Wright et al., "Deep physical neural networks trained with backpropagation," *Nature* **601**(7894), 549–555 (2022).
15. T. Zhou et al., "Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit," *Nat. Photonics* **15**(5), 367–373 (2021).
16. S. M. Tam et al., "Learning on an analog VLSI neural network chip," in *IEEE Int. Conf. Syst., Man, and Cybern. Conf. Proc.*, IEEE, pp. 701–703 (1990).
17. L. Mennel et al., "Ultrafast machine vision with 2D material neural network image sensors," *Nature* **579**(7797), 62–66 (2020).
18. C. Li et al., "Efficient and self-adaptive *in-situ* learning in multilayer memristor neural networks," *Nat. Commun.* **9**(1), 2385 (2018).
19. Z. Wang et al., "*In situ* training of feed-forward and recurrent convolutional memristor networks," *Nat. Mach. Intell.* **1**(9), 434–442 (2019).
20. J. Feldmann et al., "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature* **569**(7755), 208–214 (2019).
21. S. Bandyopadhyay et al., "Single-chip photonic deep neural network with forward-only training," *Nat. Photonics* **18**, 1335–1343 (2022).
22. M. J. Filipovich et al., "Silicon photonic architecture for training deep neural networks with direct feedback alignment," *Optica* **9**(12), 1323–1332 (2022).
23. S. Bartunov et al., "Assessing the scalability of biologically-motivated deep learning algorithms and architectures," in *32nd Conf. Neural Inf. Process. Syst*, ACM, pp. 9390–9400 (2018).
24. T. W. Hughes et al., "Training of photonic neural networks through *in situ* backpropagation and gradient measurement," *Optica* **5**(7), 864–871 (2018).
25. S. Pai et al., "Experimentally realized *in situ* backpropagation for deep learning in photonic neural networks," *Science* **380**(6643), 398–404 (2023).
26. X. Guo et al., "Backpropagation through nonlinear units for the all-optical training of neural networks," *Photonics Res.* **9**(3), B71–B80 (2021).
27. K. Wagner and D. Psaltis, "Multilayer optical learning networks," *Appl. Opt.* **26**(23), 5061–5076 (1987).
28. J. W. Goodman, A. Dias, and L. Woody, "Fully parallel, high-speed incoherent optical method for performing discrete Fourier transforms," *Opt. Lett.* **2**(1), 1–3 (1978).
29. J. Spall et al., "Fully reconfigurable coherent optical vector–matrix multiplication," *Opt. Lett.* **45**(20), 5752–5755 (2020).
30. S. Ohno et al., "Si microring resonator crossbar array for on-chip inference and training of the optical neural network," *ACS Photonics* **9**(8), 2614–2622 (2022).
31. T. Zhou et al., "*In situ* optical backpropagation training of diffractive optical neural networks," *Photonics Res.* **8**(6), 940–953 (2020).
32. R. W. Boyd, *Nonlinear Optics*, Academic Press (2020).
33. J. Wang et al., "Saturable absorption in graphene-on-waveguide devices," *Appl. Phys. Express* **12**(3), 032003 (2019).
34. A. E. Siegman, "Gain-guided, index-antiguided fiber lasers," *J. Opt. Soc. Am. B* **24**, 1677–1682 (2007).
35. V. Arrizón et al., "Pixelated phase computer holograms for the accurate encoding of scalar complex fields," *J. Opt. Soc. Am. A* **24**(11), 3500–3507 (2007).
36. J. Spall, "Training neural networks with end-to-end optical backpropagation," PhD thesis, University of Oxford (2024).

James Spall completed his PhD at the University of Oxford in the Atomic and Laser Physics Department. His research into optical neural networks and optical computing hardware has resulted in publications in a range of leading scientific journals, numerous conference talks, multiple patents, and co-founding the optical computing start-up Lumai. He previously completed his master's degree in mathematics and physics at Durham University, winning numerous awards and graduating top of his cohort.

Xianxin Guo is co-founder and head of research at Lumai, an Oxford-based startup developing optical computing products. After earning his PhD in physics from Hong Kong University of Science and Technology in 2018, he served as an RCE 1851 research fellow at Oxford and lecturer at Keble College, bringing a decade of international experience in optics and quantum physics.

Alexander I. Lvovsky is an award-winning educator and experimental physicist with expertise in quantum and classical optics and optical neural networks, best known for his work on quantum light. He grew up in Moscow and completed his PhD at Columbia University in 1998. After working in several academic positions throughout the world, he became a professor at Oxford University in 2018. Aside from his research, he is a popular public speaker, quantum science evangelist, and education outreach leader.