

ARTICLE

Open Access

Deep learning with photonic neural cellular automata

Gordon H. Y. Li¹, Christian R. Leefmans¹, James Williams², Robert M. Gray², Midya Parto^{2,3} and Alireza Marandi^{1,2}✉

Abstract

Rapid advancements in deep learning over the past decade have fueled an insatiable demand for efficient and scalable hardware. Photonics offers a promising solution by leveraging the unique properties of light. However, conventional neural network architectures, which typically require dense programmable connections, pose several practical challenges for photonic realizations. To overcome these limitations, we propose and experimentally demonstrate Photonic Neural Cellular Automata (PNCA) for photonic deep learning with sparse connectivity. PNCA harnesses the speed and interconnectivity of photonics, as well as the self-organizing nature of cellular automata through local interactions to achieve robust, reliable, and efficient processing. We utilize linear light interference and parametric nonlinear optics for all-optical computations in a time-multiplexed photonic network to experimentally perform self-organized image classification. We demonstrate binary (two-class) classification of images using as few as 3 programmable photonic parameters, achieving high experimental accuracy with the ability to also recognize out-of-distribution data. The proposed PNCA approach can be adapted to a wide range of existing photonic hardware and provides a compelling alternative to conventional photonic neural networks by maximizing the advantages of light-based computing whilst mitigating their practical challenges. Our results showcase the potential of PNCA in advancing photonic deep learning and highlights a path for next-generation photonic computers.

Introduction

Deep learning models have demonstrated remarkable capabilities in numerous domains, ranging from computer vision to natural language processing, scientific discovery, and generative art^{1–4}. However, as the complexity and scale of these models continue to surge, a critical challenge emerges: the need for efficient and scalable hardware solutions to handle the ever-increasing computational demands. For example, recent trends show that the compute requirements for deep learning models are doubling approximately every 5–6 months⁵. This is far outpacing improvements in conventional digital electronic computers, which has spurred the use of application-specific hardware accelerators such as Graphics

Processing Units and Tensor Processing Units⁶. In this context, the convergence of deep learning with photonics has emerged as a promising frontier, poised to redefine the landscape of neural network computation. By leveraging the distinct characteristics of light, photonic hardware can unlock unprecedented processing speeds, parallelism, and energy efficiencies that surpass the capabilities of traditional electronic architectures^{7,8}. To enable this new paradigm of photonic deep learning, much of the focus so far has been on developing the fundamental devices needed for crucial neural network operations. Indeed, there have been impressive demonstrations of photonics for linear operations such as matrix multiplication and convolutions^{9–11}, as well as nonlinear activation functions such as rectified linear unit^{12–14}. These photonic building blocks are now comparable to or surpass their electronic counterparts in certain important computing metrics.

However, studying system-level architectures for photonic neural networks (PNNs) beyond single devices

Correspondence: Alireza Marandi (marandi@caltech.edu)

¹Department of Applied Physics, California Institute of Technology, Pasadena, CA, USA

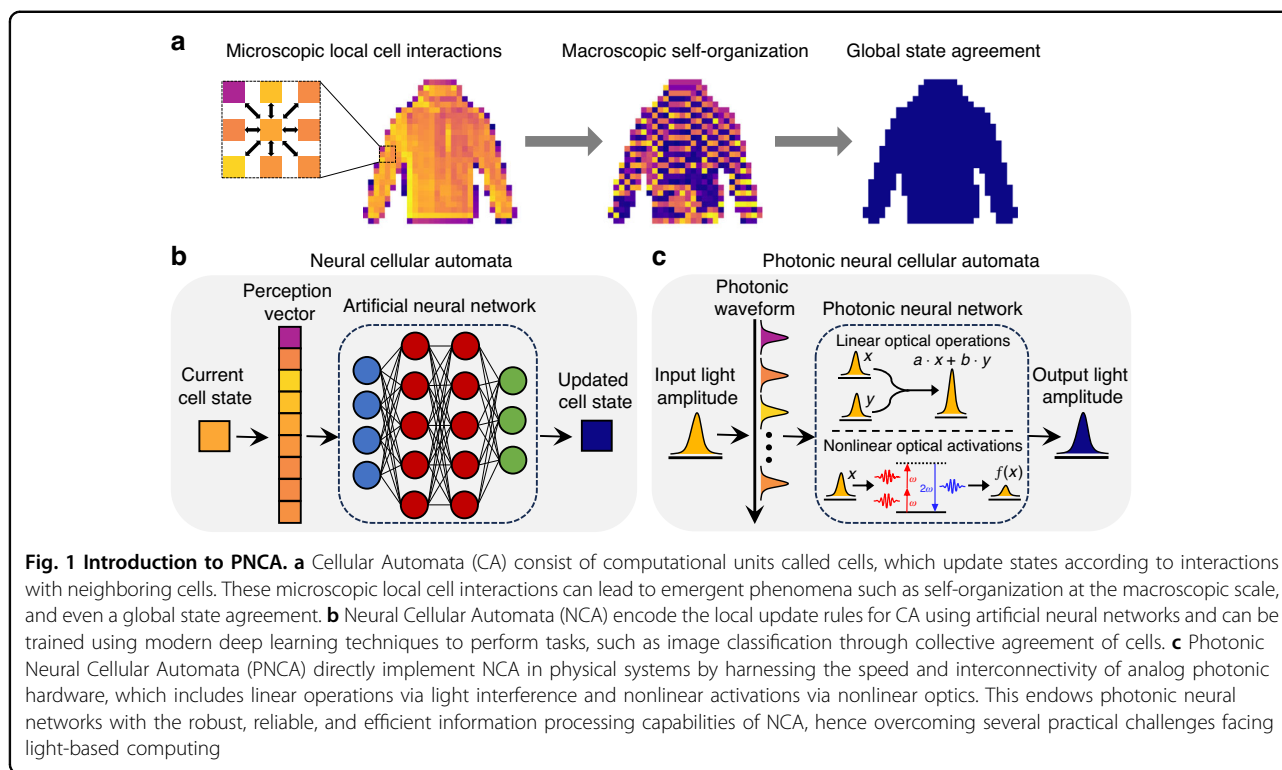
²Department of Electrical Engineering, California Institute of Technology, Pasadena, CA, USA

Full list of author information is available at the end of the article

© The Author(s) 2024



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.



is also of vital importance. This is crucial since photonics and electronics operate in entirely different regimes¹⁵. The computational advantages of photonic building blocks can quickly diminish when used to implement conventional neural network architectures that were optimized for digital electronics¹⁶. Advancing photonic deep learning towards end-to-end and scalable photonic systems requires properly considering neural network architectures that can benefit from implementation with specific photonic hardware. One important hurdle is that conventional deep learning architectures such as Multi-layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs), which have so far been mainstays for PNNs, require densely-connected layers with large numbers of parameters, which are challenging to realize in typical photonic platforms and current demonstrations of PNNs. For example, integrated PNNs can possess fast input update rates (>1 GHz) but feature a small number of programmable parameters (<10³)^{9,10,14}, whereas free-space PNNs can contain a large number of parameters (>10⁶) but have slow input update rates (<10 kHz)^{17–19}. Finally, PNNs are usually operated with fixed weights that cannot be rapidly updated in real-time. This constraint makes it difficult for PNNs to efficiently implement the complex structures of modern deep learning models and also poses reliability concerns when generalizing to out-of-distribution data.

To overcome these apparent disparities between photonics capabilities and conventional neural network architectures, we propose and experimentally demonstrate a novel type of PNN based on Neural Cellular Automata (NCA)²⁰. Cellular automata (CA) are computational models composed of a lattice of cells with states that follow an update rule, which defines how the state of a cell evolves over time based on the states of its neighboring cells (Fig. 1a)^{21,22}. Inspired by biological systems, the local interactions between cells governed by the update rule gives rise to complex phenomena and emergent patterns at the global-scale²³ (Supplementary Information Section I). Unlike conventional human-designed update rules, NCA (Fig. 1b) harness the complex dynamics of cellular automata by using modern deep learning techniques to learn the local update rules needed to perform specific tasks such as regenerating patterns²⁰, self-classifying images²⁴, and texture generation²⁵. Our Photonic Neural Cellular Automata (PNCA) combines the advantages of photonic hardware with NCA to achieve self-organized image classification (Fig. 1c). The PNCA leverages a completely different methodology for computer vision tasks compared to previous PNNs based on MLPs or CNNs. This enables noise-robust processing, as well as convenient measures of uncertainty for identifying anomalies and out-of-distribution data. Furthermore, PNCA achieves parameter-efficient solutions since the photonic hardware can operate with fixed weights and

only needs to encode the parameters for local update rules instead of global network weights. The proposed PNCA approach can be generalized to suit a wide variety of existing photonic hardware, which can potentially greatly increase the functionality of PNNs and addresses several important challenges facing photonic deep learning.

Results

PNCA architecture

The key concepts of the general PNCA architecture are shown in Fig. 1, which can be adapted to suit a wide range of different photonic hardware platforms (e.g., see Supplementary Information Section II). For computer vision tasks, each pixel in the input image corresponds to a cell in the PNCA. Cells are designated as either *alive* or *dead* through an alive masking procedure. This can be done by setting a threshold for the initial pixel value, below which the cell is considered dead. Only alive cells are actively updated by the PNCA, whereas dead cells can influence the updates of alive cells but are otherwise quiescent. The cell state updates according to a rule that depends on the cells in a local m -cell neighborhood. For example, Fig. 1a shows the prototypical Moore neighborhood composed of the cell and the 8 cells that surround it. Other types of local cell neighborhoods are also possible. In the PNCA, the optical field corresponding to each cell is split into m optical paths to define the desired m -cell neighborhood for the local update rule. The local update rule for the PNCA is encoded by the photonic hardware (Fig. 1c), which accepts the m inputs given by the m -cell neighborhood and outputs the next cell state. Although Fig. 1a only shows each cell state having a single channel, this can also be extended to multiple channels (e.g., RGB color image channels) by increasing the inputs and outputs accordingly. In general, the programmable photonic hardware contains feed-forward layers with linear operations which can be implemented through meshes of Mach–Zehnder interferometers⁹, photonic crossbar arrays¹⁰, micro-ring resonator weight banks²⁶, or other linear photonic devices^{11,14}. In addition, there must also be layers performing nonlinear activations such as photonic devices based on optoelectronic measurement-feedback^{14,27} or nonlinear-optical crystals^{12,13}. This kind of feed-forward programmable photonic hardware specifying a single input-output function has been used in previous PNNs. However, for PNCA, the key difference is that the photonic hardware only needs sparse connections and enough parameters to encode for the local update rule, which is usually orders-of-magnitude fewer than the number of parameters needed to encode global network weights in fully-connected layers for MLPs or CNNs. In other words, the parameter-efficient PNCA architecture can enable existing PNN hardware with relatively few parameters to perform larger and more complicated tasks

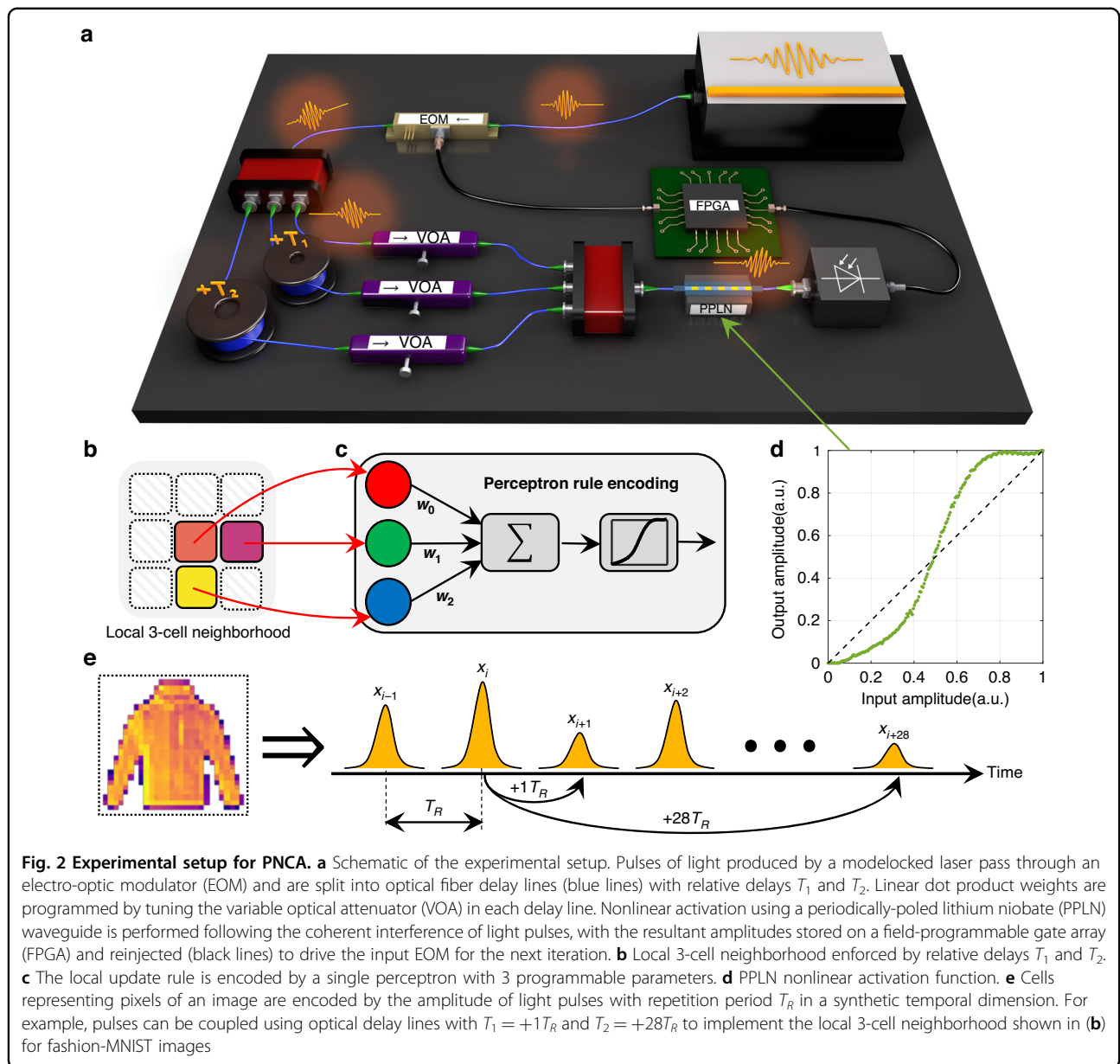
than otherwise possible in conventional neural network architectures. Furthermore, this local update rule can more easily tolerate the use of fixed weights after training since every cell follows the same update rule. Note that the weights/parameters encoding the local update rule for cells do not vary across cell index or time step iteration, which avoids the need for costly parameter updates in photonic hardware. Finally, the output is recurrently fed back to update the cell state for the next iteration. This can be accomplished by photodetection and electro-optic feedback or by using all-optical feedback lines (e.g., see Supplementary Information Section IV).

Unlike conventional CA with discrete cell states²¹, NCA use cell states that are continuous-valued²⁰, which allows the model to be end-to-end differentiable and compatible with gradient-descent based learning algorithms. In this work, we consider the task of self-organized image classification. The target output after the final iteration is to have every alive cell in the state that corresponds to the class label for the input image. The alive cells must form this collective agreement through only the local interactions defined by repeated iteration of the update rule. This can be interpreted as a kind of recurrent neural network, which can be trained (Fig. S3) using the standard backpropagation-through-time algorithm²⁸. Using a cell-wise L_2 loss was found to give better performance compared to cross-entropy loss of labels, which is more commonly used for image classification tasks²⁰. The training can either be done in situ by performing the forward pass in PNCA to more accurately capture the physics, or completely digitally by simulating the photonic hardware with noise^{29,30}.

Experimental realization of PNCA

We used a time-multiplexed scheme and commercially-available optical-fiber components to experimentally demonstrate proof-of-concept for a simple version of PNCA as shown in Fig. 2. Each cell state is given by the amplitude of a laser light pulse generated by a mode-locked laser with a fixed repetition rate such that the cells are inputted one at a time in a flattened 1D lattice by raster scanning across the 2D image. In this way, each cell occupies a time-bin site in a synthetic temporal dimension³¹. Therefore, distances in a real-space lattice correspond to time-differences in the temporal dimension and cells at different lattice sites can be made to interact by using temporal delay lines.

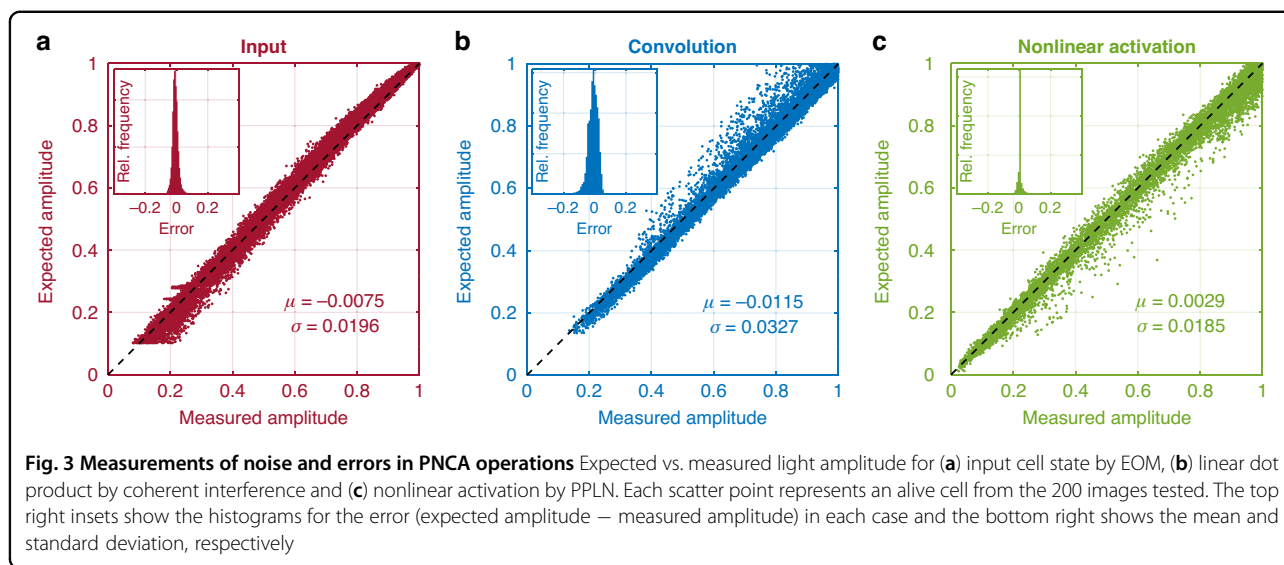
The pulse amplitude/phase representing the cell state is set using an electro-optic modulator (EOM), and the pulse is then split between 3 temporal optical delay lines with relative delays T_1 and T_2 chosen to enforce the desired 3-cell local neighborhood shown in Fig. 2b. In this simple example, the local update rule is encoded by a single perceptron neuron shown in Fig. 2c, which consists



of a linear dot product followed by a nonlinear activation function. The dot product is achieved by coherent interference of the optical delay lines, each equipped with a variable optical attenuator (VOA) to program the desired weights, which can be either positive (in-phase/constructive interference) or negative (out-of-phase/destructive interference). The nonlinear activation is performed using pump-depleted second harmonic generation (see Supplementary Information Section VI) in a reverse-proton exchange periodically-poled lithium niobate waveguide³². This produces a sigmoid-like function as shown in Fig. 2d. Thus, the computations in the local update rule are achieved all-optically. Overall, the local update rule contains only 3 programmable parameters,

but can still perform complex tasks. Finally, the cell state is measured using a photodetector, stored on a field-programmable gate array (FPGA), and electro-optically re-injected for the next iteration after alive-cell masking.

A crucial aspect of photonic hardware is that it is analog and noisy. A key advantage of the PNCA architecture is that it is robust to noise due to the self-organizing nature of the cell states. We rigorously characterized the noise and errors in our PNCA implementation, which arises from three main operations: (1) the input cell state due to thermal and electronic noise in the EOM, (2) the linear dot product due to phase noise and imperfect pulse temporal overlap in the coherent interference, and (3) the nonlinear activation due to thermal noise and



photorefractive effects in the PPLN. We characterized these errors using 200 test images. The expected vs. measured amplitudes of alive cells in these images are shown in Fig. 3. The mean and standard deviation of the errors (expected amplitude – measured amplitude) achieved in our system are typical of photonic hardware, and we show that this is tolerable for the PNCA architecture due to its noise-robustness.

Self-organized image classification

We trained the experimental PNCA to perform binary image classification using the fashion-MNIST dataset consisting of 28×28 pixel gray-scale images of clothing items³³. For example, Fig. 4a shows how the PNCA can classify images of sneakers and trousers. The alive cell masking is performed by designating any pixel with initial value $\alpha > 0.1$ as an alive cell, and all other pixels as dead cells with constant value of zero. Each input image was iterated for $t = 21$ time steps in the PNCA, which was sufficient for the cells to reach an approximate global agreement. The alive cells self-organize to have state values close to zero (unity) for images of sneakers (trousers). Finally, the predicted image label is obtained in postprocessing (see Supplementary Information Section III) by performing global average pooling of the final alive cell states followed by softmax classification. In this case, a global average closer to zero (unity) indicates that the predicted image label is sneaker (trouser).

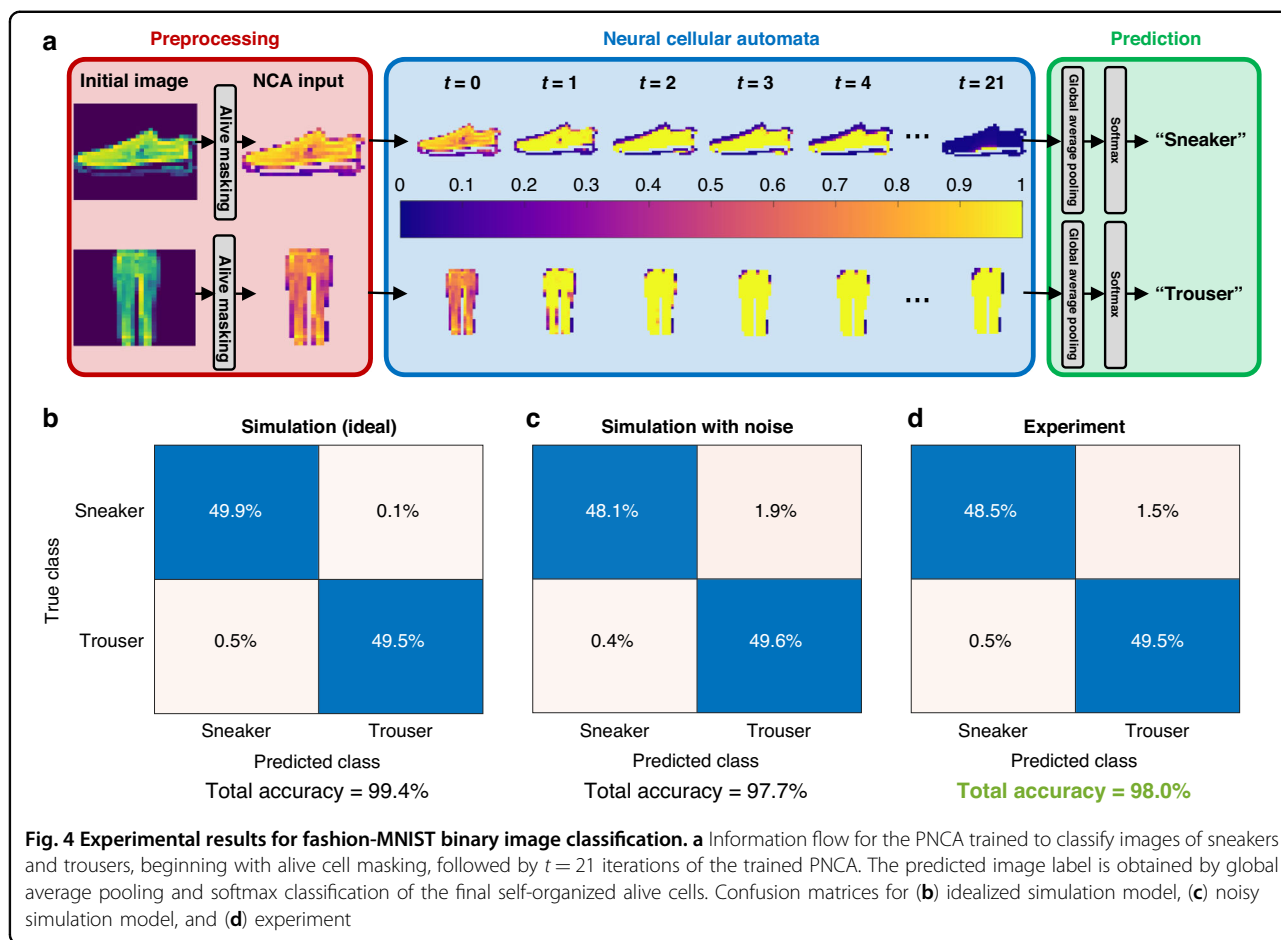
The training procedure was performed digitally using an idealized simulation model of the PNCA that had no noise. The confusion matrix for the idealized model is shown in Fig. 4b, which yielded a final test accuracy of 99.4%.

Next, the trained model parameters were frozen, and the model was tested again but with additional simulated

Gaussian noise for each operation, matching the noise characteristics shown in Fig. 3. The confusion matrix for the noisy model is shown in Fig. 4c, which has a slightly lower final test accuracy of 97.7%. The trained model parameters were implemented in the experimental PNCA by appropriately tuning the VOAs. The confusion matrix for the experimental result tested on the same 200 images (100 for each class) used to characterize the noise in Fig. 3 is shown in Fig. 4d and has a final test accuracy of 98.0%. This experimental test accuracy is in close agreement with the simulated noisy model, which shows that the PNCA operates as desired and can successfully tolerate the use of noisy photonic hardware. No special training or noise regularization techniques were used for the PNCA. We emphasize that the robustness emerges through the local interactions between cells forming a global agreement. Therefore, even if one cell fails, the collective state can still persist (Supplementary Information Section VIII).

Out-of-distribution data

Furthermore, conventional neural networks are prone to making overconfident predictions and failing to generalize to out-of-distribution data³⁴. This lack of reliability is especially problematic for photonic deep learning in which the weights are fixed and online learning is not practical. The NCA approach addresses this shortcoming by using the average state value of all alive cells as a built-in measure of uncertainty. We experimentally demonstrated this for PNCA by using the same network as before that was trained on images of sneakers and trousers. Now, we test the PNCA on images of bags, which is an out-of-distribution class that the PNCA was not exposed to during training. The distributions for the alive cell averages of the sneaker, trouser, and bag classes are shown for the initial test

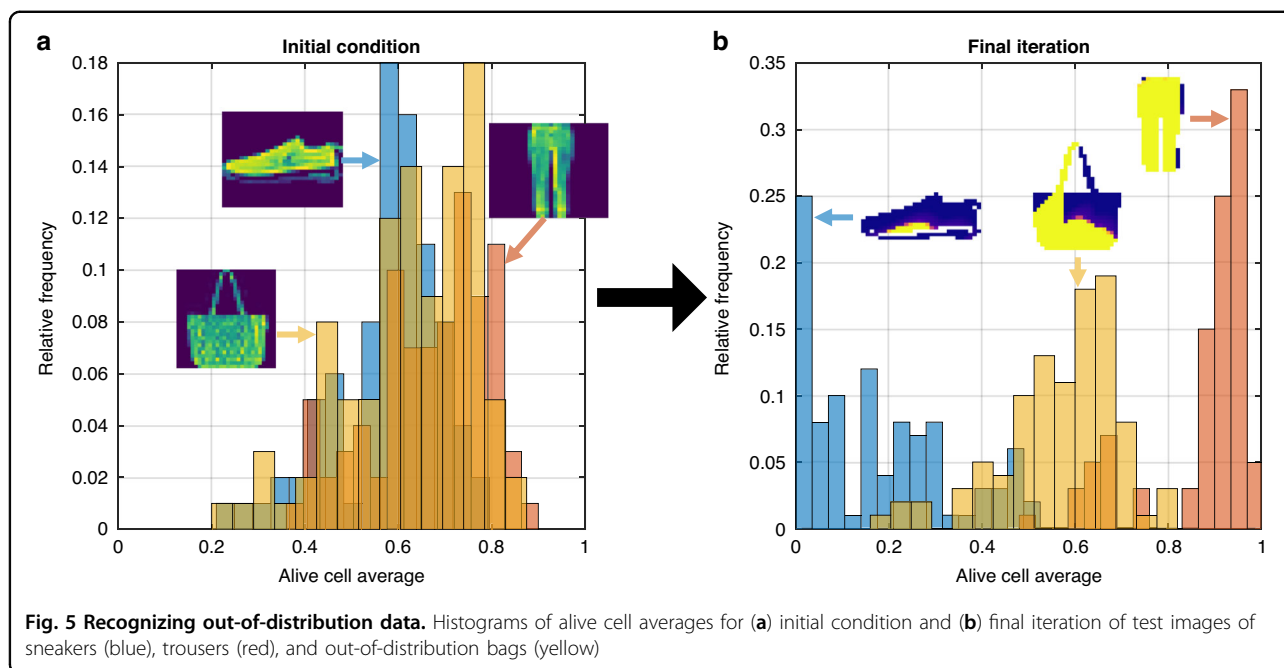


images in Fig. 5a. It clearly shows that the initial distributions for alive cell averages closely overlap between all classes. Upon iteration of the local update rule that was learned during training, the PNCA is able to successfully separate the distributions for sneaker and trouser, with final alive cell averages of 0.1743 and 0.8742, respectively, as shown in Fig. 5b. In this case, the difference between the final alive cell average and zero/one indicates the uncertainty in the prediction. However, the final alive cell average for out-of-distribution test images of bags is 0.5682, which is close to 0.5 and means that the cells did not reach a global agreement. This shows that the PNCA can use the alive cell average as a proxy for uncertainty and to detect out-of-distribution data. Unlike for conventional neural network architectures, neither special training/inference techniques nor additional training data are required.

Simulated benchmarks

In the current experimental implementation of PNCA, we represented the local update rule using only a single neuron with 3 parameters. However, the PNCA architecture can also be used with more advanced PNN

hardware that can represent the local update rule using a greater number of neurons/parameters. In general, a greater number of neurons/parameters can allow for more complicated tasks and higher classification accuracy while the hardware complexity remains far less demanding than other neural network architectures. Table 1 shows the simulated binary classification accuracy of the proposed PNCA with different numbers of neurons/parameters encoding the local update rule (see Methods). The simulated PNCA was tested on different classes within the fashionMNIST dataset, as well as other similar benchmark datasets including EMNIST (28 × 28 grayscale images of handwritten letters)³⁵, MNIST (28 × 28 grayscale images of handwritten digits)³⁶, and CIFAR10 (32 × 32 RGB images of animals and vehicles)³⁷. The corresponding classification accuracies for conventional MLPs with different numbers of hidden neurons/parameters are also shown for reference. It can be seen that the PNCA requires far fewer parameters and achieves comparable (or sometimes even better) classification accuracy to MLPs across a wide variety of examples. Therefore, the PNCA architecture may provide an efficient way for PNNs with only few parameters¹⁴ to



increase their task performance with minimal changes to existing hardware.

Discussion

We note that CA with simple rules and only nearest-neighbor connections are known to be Turing-universal models of computation³⁸. This means that CA can, at least in principle, compute any function that a fully-connected network (neural or otherwise) can compute. There is no fundamental loss of computational power or information processing ability imposed by the sparsity. Therefore, given enough time, the PNCA approach (albeit with more advanced input encoding schemes) must be able to achieve at least the same accuracy as conventional neural networks such as MLPs. However, in practice, the time steps are truncated to be finite, which means the classification accuracy may not always be the same as MLPs. It is difficult to determine a priori on which examples the PNCA will perform better/worse compared to MLPs. In the proposed PNCA architecture, the maximum throughput is ultimately limited by the speed of the nonlinear activation function. We chose to utilize ultrafast nonlinear optics since it can be orders of magnitude faster than digital electronics for performing nonlinear activations. The reverse-proton exchange PPLN waveguide³² used in the experiment utilizes strong $\chi^{(2)}$ optical nonlinearity and has a phase-matching bandwidth of ~ 100 GHz, which determines the maximum possible computational clock rate. This is an important step towards achieving end-to-end PNNs since it is much faster than other nonlinear activation

methods utilizing optoelectronics¹⁴, slower optical nonlinearities^{39,40}, or spectral shaping⁴¹. Note that in our experiment, we used optoelectronic conversions after the PPLN nonlinear activation to perform feedback between iterations, however, this was not a fundamental limitation and can in principle be replaced by an all-optical feedback loop in the form of a sufficiently long optical fiber (see Supplementary Information Section IV). The scalability and performance can be further improved by using nanophotonic PPLN waveguides, which were recently demonstrated to achieve a maximum speed >10 THz and energy of ~ 10 fJ per nonlinear activation¹³.

In summary, we have proposed and experimentally demonstrated a novel approach to photonic deep learning based on PNCA. It addresses several system-level challenges in previous PNNs and can serve as a general architecture for a wide variety of photonic hardware platforms. In particular, we showed that PNCA enables noise-robust image classification through local interactions between cells with an inherent measure of uncertainty based on alive cell averages. Moreover, the efficient PNCA model encoding requires orders of magnitude fewer parameters compared to MLPs or CNNs. Our single perceptron neuron rule encoding can be straightforwardly extended to a shallow neural network with a greater number of programmable parameters to perform more complicated and larger-scale computer vision tasks. For example, we focused on binary image classification for simplicity, but it is possible to perform image classification with more classes

Table 1 Simulated benchmarks

Dataset	Classes	PNCA			Multi-layer perceptron		
		1 neuron, 10 params.	10 neurons, 111 params.	100 neurons, 1101 params.	784 → 1 → 1, 787 params.	784 → 10 → 1, 7861 params.	784 → 100 → 1, 78,601 params.
fashionMNIST (28 × 28, grayscale)	trouser/sneaker	99.4	99.9	99.9	99.8	99.9	99.9
	t-shirt/pullover	87.0	93.5	94.1	94.0	95.8	96.8
	coat/sandal	98.8	99.3	99.3	99.7	99.8	99.8
	dress/boot	94.7	98.0	98.2	99.8	99.9	99.9
	shirt/bag	89.5	95.6	95.8	95.0	95.2	97.3
EMNIST (28 × 28, grayscale)	A/B	83.7	99.5	99.5	95.3	97.5	98.1
	C/D	97.4	98.5	98.6	98.5	98.5	99.6
	P/Q	89.3	97.8	98.0	97.7	98.9	99.1
	R/S	87.9	96.1	96.5	96.6	98.3	99.6
	Y/Z	99.0	99.3	99.3	96.4	98.3	98.6
MNIST (28 × 28, grayscale)	one/two	99.4	99.5	99.5	98.6	99.5	99.7
	three/four	99.3	99.6	99.7	98.5	99.6	99.7
	five/six	95.0	97.4	97.4	95.1	95.3	97.6
	seven/eight	91.2	99.1	99.3	96.5	98.3	99.1
	nine/zero	90.1	96.5	96.6	98.2	99.0	99.4
CIFAR10 (32 × 32, RGB)	automobile/bird	83.0	88.8	89.7	60.1	80.7	84.7
	dog/truck	78.6	90.4	90.5	78.9	80.2	86.1
	horse/ship	77.3	84.7	85.6	72.9	78.9	85.2

Image binary classification accuracy (%) for PNCA with 1, 10, and 100 neurons (10, 111, and 1101 parameters), respectively, for examples of classes in datasets: fashionMNIST (28 × 28 grayscale images of fashion items), EMNIST (28 × 28 grayscale images of handwritten letters), MNIST (28 × 28 grayscale images of handwritten digits), and CIFAR10 (32 × 32 RGB images of animals and vehicles). The corresponding accuracy for a conventional multi-layer perceptron (MLP) with 1, 10, and 100 hidden layer neurons (787, 7861, and 78,601 parameters), respectively, is also shown for reference

(e.g., the full 10-class MNIST image classification) if the number of output neuron channels is increased (e.g., see Supplementary Information Section II). Furthermore, we only used standard backpropagation training and did not employ any special training or regularization techniques. More advanced noise-aware or physics-aware training schemes²⁹ are also compatible with the PNCA architecture and may further increase performance. We used a time-multiplexed photonic network based on a synthetic temporal dimension, however, it is also possible to use an analogous PNCA approach based on other synthetic dimensions such as frequency dimensions^{40,42}. In addition to robustness to noise, it has also been previously shown that NCA are generally robust against sudden changes or failures in the underlying cell states^{20,24}. This fault-tolerance property has not yet been explored for optical implementations and can be an interesting avenue for future work on PNCA. Our work therefore highlights a clear path to advancing photonic deep learning based on PNCA and paves the way for next-generation photonic computers.

Materials and methods

Experimental setup

A more detailed schematic of the experimental setup is shown in Fig. S1. A femtosecond laser source (MenloSystems FC1500-250-WG) produces pulses of light at a fixed repetition rate of ~250 MHz. The light pulses are filtered using a 200 GHz band-pass filter with center wavelength ~1550 nm to stretch the pulse length to ~5 ps and reduce the effects of dispersion. The light pulses are photodetected (MenloSystems FPD610-FC-NIR) as a reference clock signal for the FPGA (Xilinx Zynq UltraScale+ RFSoc) to eliminate timing drift between the optical and electronic signals. The FPGA drives an EOM (IXblue MXAN-LN-10) that is used to modulate the amplitude of the light pulses. The light pulses are split into a 3-path interferometer by cascading 50:50 optical fiber splitters. Two paths of the interferometer have delays $+1T_R$ and $+28T_R$, respectively, relative to the shortest path, where T_R is the repetition period of the light pulses. The relative delays in each arm are set using a combination of optical fiber patch cords and free-space delay

stages. Tuning the free-space coupling efficiency also acts a VOA to set the relative amplitude weight in each arm. The output of the 3-arm interferometer is tapped using a 90:10 optical fiber splitter. The 10% tap is photodetected (Newport New Focus Model 2053) and used as an electronic locking signal input to a proportional-integral derivative controller (Red Pitaya). The electronic locking signal output is amplified (Thorlabs Piezo Controller MDT693B) and drives fiber phase-shifters (General Photonics FPS-002-L) that stabilize the relative phases of each delay arm. The 90% output of the 3-arm interferometer is amplified using an erbium-doped fiber amplifier (Thorlabs Fiber Amplifier 1550 nm PM) and filtered using a 200 GHz band-pass filter to reduce the amplified spontaneous emission noise. The amplified light pulses pass through a 40 mm long reverse-proton exchange PPLN waveguide³² that is heated to $\sim 52^\circ\text{C}$ with a thermocouple controller. The PPLN waveguide contains a wavelength division multiplexer on the output to separate the fundamental harmonic centered at ~ 1550 nm and the second harmonic centered at ~ 775 nm. The second harmonic output is dumped and the fundamental harmonic is photodetected (Thorlabs DET08CFC). The final photodetected signal is read as a time trace using an oscilloscope (Tektronix MSO6B) and light pulse amplitude values are stored on the FPGA to be electrooptically reinjected. A single photodetector can be used for tasks only requiring positive-valued inputs/outputs, such as the image classification tasks considered in this work. However, the electro-optic feedback scheme can also handle negative-valued outputs by instead using a local-oscillator with balanced photodetector. All optical fiber paths are single-mode polarization-maintaining (PM)

Photonic neural cellular automata model

The NCA comprises a lattice of cells indexed by lattice site number $i \in \mathbb{N}$ with states $\mathbf{x}_i \in \mathbb{C}^d$, where d is the number of channels for each cell. Each cell interacts locally in an m -cell neighborhood \mathcal{M}_i according to a fixed update rule. We consider discrete-time synchronous updates $t \in \mathbb{N}$ for cells:

$$\mathbf{x}_i(t+1) = f_\theta(\mathbf{x}_{m_{i1}}(t), \mathbf{x}_{m_{i2}}(t), \mathbf{x}_{m_{i3}}(t), \dots) \quad (1)$$

where $m_{i1}, m_{i2}, m_{i3}, \dots \in \mathcal{M}_i$ are the lattice sites in the local neighborhood of the i th cell and $f_\theta : (\mathbb{C}^d)^m \rightarrow \mathbb{C}^d$ is the local update rule. The local update rule is parameterized by $\{\theta\}$ and is differentiable so that it can be trained using modern deep learning techniques. For example, f_θ can represent a neural network. The key aspect is that the update rule f_θ is the same for all cells and all time steps.

We experimentally demonstrated a simple version of NCA implemented directly on analog photonic hardware, which we call PNCA. In PNCA, lattice sites are

represented by laser light pulses in time bins of a synthetic temporal dimension with a fixed repetition period T_R and cell states are represented by the complex amplitude of the light pulses. For simplicity, we consider a single image channel $d=1$ and the local update rule f_θ encoded by a single perceptron neuron with an $m=3$ neighborhood as shown in Fig. 2b, c. The temporal delay lines $T_1 = +1T_R$ and $T_2 = +2T_R$ set the desired local cell neighborhood and the VOAs in each arm of the 3-arm interferometer set the desired weights $\{w_0, w_1, w_2\} \in [-1, +1]$. The PIDs are used to enforce a relative phase of 0 for constructive interference, or conversely a relative phase of π for destructive interference. Therefore, at the output of the 3-arm interferometer, the combined result of the delay lines, VOAs, and phases can be summarized as a linear dot product or sliding convolutional filter:

$$y_i(t) = w_0 x_i(t) + w_1 x_{i+T_1}(t) + w_2 x_{i+T_2}(t) \quad (2)$$

where the result of the linear operation $y_i(t)$ is fed into a PPLN to perform a nonlinear activation function:

$$x_i(t+1) = g(y_i(t)) \quad (3)$$

where g is the sigmoid-like function shown in Fig. 2d. The PNCA approach is very general and Eq. (1) can be implemented using more complicated photonic hardware platforms with different cells neighborhoods, more neurons, deeper layers, and more programmable parameters (see Supplementary Information Section II).

Experimental procedure

The input modulator was calibrated by using a sequence of 200 consecutive light pulses and performing a linear voltage sweep of the input EOM, which was DC-biased open. The peak pulse amplitude or maximum value in each time bin (i.e., pulse repetition period) of the measured time trace was used to construct a look-up table for the voltage-to-light amplitude conversion. To input a specific 28×28 fashion-MNIST image, the 2D-pixel map was unrolled column-wise to form a 784×1 vector of input cell values. Alive masking was applied such that any initial pixel value < 0.1 was designated as a dead cell. The accuracy of the input operation was checked by measuring the difference between the measured input cell states and the expected value, such as shown in Fig. S2. The aggregate results are shown in Fig. 2a. Each desired weight in the linear dot product was set by tuning the coupling efficiency of a free-space section contained within each VOA in the 3-arm interferometer. Note that the VOAs were completely passive and did not consume any power. The optical power was directly measured in each arm to roughly tune the attenuation factor, and then fine-tuning of the weight was performed by checking the result of the linear interference matched the expected value like in

Table 2 Simulated benchmark parameters

Dataset	Iterations	Alive cell threshold	Training images per class	Validation images per class	Test images per class	Learning rate	Epochs
fashionMNIST	25	0.1	5000	420	1000	0.01	500
EMIST	15	0.1	500	420	500	0.01	1000
MNIST	10	0.1	5000	420	800	0.01	1000
CIFAR10	10	0.15	500	420	500	0.005	2000

Fig. S2. A standard Pound–Drever–Hall locking scheme was used to stabilize the relative phases in each delay arm to either 0 or π to ensure coherent interference. It is also possible to make use of the full complex amplitude of light, although we restricted our attention to only real values. The relative delays in each temporal delay line was set roughly using optical fiber patch cords, then fine-tuned using free-space delay stages to ensure maximal temporal overlap between interfering light pulses. The aggregate results of the linear dot product or convolution operation are shown in Fig. 2b. The temperature of the PPLN was fine-tuned around 52 °C until maximal average power was measured on the output second-harmonic given a small input fundamental harmonic average power ~ 1 mW. The PPLN nonlinear activation function shown in Fig. 2d was measured using a sequence of consecutive light pulses with linearly increasing input amplitude. To ensure stable operation over long-periods of time (>12 h) throughout the experiment, we regularly check that the calibrated PPLN nonlinear activation function remains the same and does not change significantly due to photo-refractive or thermal effects. The measured values for PPLN nonlinear activations were also compared against the expected simulated values as shown in Fig. S2. The aggregate results of the PPLN nonlinear activation operation are shown in Fig. 2c. To perform self-organized image classification using the experimental PNCA, the input modulator was first calibrated. Then, the PPLN nonlinear activation function was measured, and a simulated digital model of the PNCA was trained (see “Model Training”) to determine the optimal weights to be set in the temporal delay lines. The light pulse amplitudes were stored digitally on the FPGA in between iterations, however, the iteration feedback can also be performed all-optically using an optical fiber cavity (Supplementary Information Section IV).

Model training

The PNCA can be trained using the standard backpropagation-through-time algorithm (Fig. S3) for recurrent neural networks if a differentiable model of the update rule f_θ is known. The goal is to learn the

parameters $\{\theta\}$ for a particular task such as self-organized image classification. We consider a cell-wise L_2 loss at each time step:

$$L = \frac{1}{T} \sum_{t=1}^T \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i(t) - \mathbf{z}_i\|^2 \quad (4)$$

where \mathbf{z}_i is the target state for the i th cell. The parameter values are updated using stochastic gradient descent:

$$\theta^{[l+1]} = \theta^{[l]} - \alpha \nabla L(\theta^{[l]}) \quad (5)$$

where l is the epoch number and $\alpha > 0$ is the learning rate. The gradient ∇L is calculated by unrolling the network in time for T time steps and applying the chain rule or automatic differentiation. More complicated gradient-based optimization such as stochastic gradient descent with momentum or adaptive moment estimation can also be used to perform parameter updates. We trained a PNCA to perform binary image classification of sneakers and trousers classes from the fashion-MNIST dataset using 5000 training and 420 validation images for each class, learning rate of $\alpha = 0.002$, and 200 training epochs. An example of a training curve is shown in Fig. S4.

Simulation procedure

For the simulated benchmarks, we considered PNCA using the classic Moore neighborhood (composed of the current cell plus its 8 neighboring cells in a square lattice). The local update rule f_θ was encoded by a 2-layer fully-connected network $9 \rightarrow N \rightarrow 1$, where N is the number of hidden neurons. The simulation parameters are shown below in Table 2. For the CIFAR10 dataset examples, we applied the *same* local update rule channel-wise to each RGB input channel for the images, then averaged over the channels for the final classification. For the simulated MLPs used for comparison, we used a 2-layer fully-connected network $784 \rightarrow N \rightarrow 1$, where N is the number of hidden neurons. We used clipped ReLU nonlinear activation function $f(x) = \min(1, \max(0, x))$ to ensure that the final output probability is in range $[0, 1]$. Each MLP was trained using binary cross-entropy loss, the same number

of training/validation/test images as for the corresponding PNCA, learning rate of 0.0001, and 500 epochs. The images were flattened column-wise to form the input to the MLPs and we resampled the images to be 28×28 grayscale for CIFAR10 since MLPs can only accept inputs with a fixed dimension, whereas PNCA can handle arbitrary image input sizes

Acknowledgements

The authors acknowledge support from ARO grant no. W911NF-23-1-0048, NSF grant no. 1846273 and 1918549, Center for Sensing to Intelligence at Caltech, and NASA/JPL. The authors thank NTT Research for their financial and technical support. The authors thank Martin M. Fejer and Carsten Langrock for providing the PPLN waveguide. G.H.Y.L. acknowledges support from the Quad Fellowship.

Author details

¹Department of Applied Physics, California Institute of Technology, Pasadena, CA, USA. ²Department of Electrical Engineering, California Institute of Technology, Pasadena, CA, USA. ³Physics and Informatics Laboratories, NTT Research Inc., Sunnyvale, CA, USA

Author contributions

All authors contributed to this manuscript.

Data availability

The data used to generate the plots and results in this paper are available from the corresponding author upon reasonable request.

Code availability

The code used to analyze the data and generate the plots for this paper is available from the corresponding author upon reasonable request.

Conflict of interest

A.M. has financial interest in PINC Technologies Inc., which is developing photonic integrated nonlinear circuits. The remaining authors declare no competing interests.

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41377-024-01651-7>.

Received: 13 January 2024 Revised: 17 September 2024 Accepted: 22 September 2024

Published online: 08 October 2024

References

- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
- Vaswani, A. et al. Attention is all you need. In *Proc. 31st International Conference on Neural Information Processing Systems*. 6000–6010 (Curran Associates Inc., Long Beach, CA, USA, 2017).
- Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
- Goodfellow, I. J. et al. Generative adversarial nets. In *Proc. 27th International Conference on Neural Information Processing Systems*. 2672–2680 (MIT Press, Montreal, Canada, 2014).
- Sevilla, J. et al. Compute trends across three eras of machine learning. In *Proc. 2022 International Joint Conference on Neural Networks (IJCNN)*. 1–8 (IEEE, Padua, Italy, 2022).
- Jouppi, N. P. et al. In-datacenter performance analysis of a tensor processing unit. In *Proc. 44th Annual International Symposium on Computer Architecture*. 1–12 (ACM, Toronto, ON, Canada, 2017).
- Shastri, B. J. et al. Photonics for artificial intelligence and neuromorphic computing. *Nat. Photonics* **15**, 102–114 (2021).
- Wetzstein, G. et al. Inference in artificial intelligence with deep optics and photonics. *Nature* **588**, 39–47 (2020).
- Shen, Y. C. et al. Deep learning with coherent nanophotonic circuits. *Nat. Photonics* **11**, 441–446 (2017).
- Feldmann, J. et al. Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589**, 52–58 (2021).
- Xu, X. Y. et al. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* **589**, 44–51 (2021).
- Feldmann, J. et al. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature* **569**, 208–214 (2019).
- Li, G. H. Y. et al. All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning. *Nanophotonics* **12**, 847–855 (2023).
- Ashtiani, F., Geers, A. J. & Aflatouni, F. An on-chip photonic deep neural network for image classification. *Nature* **606**, 501–506 (2022).
- Miller, D. A. B. Are optical transistors the logical next step? *Nat. Photonics* **4**, 3–5 (2010).
- Sze, V. et al. Efficient processing of deep neural networks: a tutorial and survey. *Proc. IEEE* **105**, 2295–2329 (2017).
- Farhat, N. H. et al. Optical implementation of the Hopfield model. *Appl. Opt.* **24**, 1469–1475 (1985).
- Zhou, T. K. et al. Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nat. Photonics* **15**, 367–373 (2021).
- Lin, X. et al. All-optical machine learning using diffractive deep neural networks. *Science* **361**, 1004–1008 (2018).
- Mordvintsev, A. et al. Growing neural cellular automata. *Distill* **5**, e23 (2020).
- Wolfram, S. Statistical mechanics of cellular automata. *Rev. Mod. Phys.* **55**, 601–644 (1983).
- Li, G. H. Y. et al. Photonic elementary cellular automata for simulation of complex phenomena. *Light Sci. Appl.* **12**, 132 (2023).
- Gardner, M. The fantastic combinations of John Conway's new solitaire game "life". *Sci. Am.* **223**, 120–123 (1970).
- Randazzo, E. et al. Self-classifying MNIST digits. *Distill* **5**, e00027–002 (2020).
- Niklasson, E. et al. Self-organising textures. *Distill* **6**, e00027–003 (2021).
- Tait, A. N. et al. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* **7**, 7430 (2017).
- Williamson, I. A. D. et al. Reprogrammable electro-optic nonlinear activation functions for optical neural networks. *IEEE J. Sel. Top. Quantum Electron.* **26**, 7700412 (2020).
- Werbos, P. J. Backpropagation through time: what it does and how to do it. *Proc. IEEE* **78**, 1550–1560 (1990).
- Wright, L. G. et al. Deep physical neural networks trained with backpropagation. *Nature* **601**, 549–555 (2022).
- Pai, S. et al. Experimentally realized in situ backpropagation for deep learning in photonic neural networks. *Science* **380**, 398–404 (2023).
- Leeffmans, C. et al. Topological dissipation in a time-multiplexed photonic resonator network. *Nat. Phys.* **18**, 442–449 (2022).
- Langrock, C. & Fejer, M. M. Fiber-feedback continuous-wave and synchronously-pumped singly-resonant ring optical parametric oscillators using reverse-proton-exchanged periodically-poled lithium niobate waveguides. *Opt. Lett.* **32**, 2263–2265 (2007).
- Xiao, H., Rasul, K. & Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. Preprint at <https://doi.org/10.48550/arXiv.1708.07747> (2017).
- Guo, C. et al. On calibration of modern neural networks. In *Proc. 34th International Conference on Machine Learning*. 1321–1330 (PMLR, Sydney, Australia, 2017).
- Cohen, G. et al. EMNIST: extending MNIST to handwritten letters. In *Proc. 2017 International Joint Conference on Neural Networks (IJCNN)*. 2921–2926 (IEEE, Anchorage, AK, USA, 2017).
- Deng, L. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **29**, 141–142 (2012).
- Krizhevsky, A. *Learning Multiple Layers of Features from Tiny Images*. MSc thesis, University of Toronto (2009).
- Cook, M. Universality in elementary cellular automata. *Complex Syst.* **15**, 1–40 (2004).
- Porte, X. et al. A complete, parallel and autonomous photonic neural network in a semiconductor multimode laser. *J. Phys. Photonics* **3**, 024017 (2021).
- Zhou, T. K. et al. Ultrafast dynamic machine vision with spatiotemporal photonic computing. *Sci. Adv.* **9**, eadg4391 (2023).
- Yildirim, M. et al. Nonlinear optical feature generator for machine learning. *APL Photonics* **8**, 106104 (2023).
- Yuan, L. Q. et al. Synthetic dimension in photonics. *Optica* **5**, 1396–1405 (2018).