We show that optical waves passing through a nanophotonic medium can perform artificial neural computing. Complex information, such as an image, is encoded in the wave front of an input light. The medium continuously transforms the wave front to realize highly sophisticated computing tasks such as image recognition. At the output, the optical energy is concentrated to well defined locations, which for example can be interpreted as the identity of the object in the image. These computing media can be as small as tens of wavelengths and offer ultra-high computing density. They exploit sub-wavelength linear and nonlinear scatterers to realize complex input-output mapping far beyond the capabilities of traditional nanophotonic devices. All structural degrees of freedom can be used as training weights, forming a vast parameter space with strong expressive power.

Artificial neural networks (ANN) have shown exciting potential in a wide range of applications, but they also require ever-increasing computing power. This has prompted an effort to search for alternative computing methods that are faster and more energy efficient. One interesting approach is optical neural computing [1–7]. This analog computing method can be passive, with minimal energy consumption, and more importantly, its intrinsic parallelism can greatly accelerate computing speed.

Most optical neural computing follows the architecture of digital ANNs, using a layered feed-forward network as shown in Fig. 1a. Free-space diffraction[4, 8] or integrated waveguides[1, 3, 9] are used as the connections between layered activation units. Similar to the way digital signals propagate in ANN, optical signals pass through optical networks in the forward direction once. Light reflection propagating in the backward direction is avoided or neglected. However, it is the reflection that provides the feedback mechanism which gives rise to rich wave physics. It holds the key to the miniaturization of optical devices such as laser cavities [10], photonic crystals [11], meta-materials [12], and ultra-compact beam splitters [13–15]. Here we show that by leveraging optical reflection instead of avoiding it, it is possible to go beyond the paradigm of layered feed-forward networks to realize artificial neural computing in a continuous and layer-free fashion. Fig.1b shows the proposed Nanophotonic Neural Medium (NNM). An optical signal enters from the left and the output is the energy distribution on the right side of the medium. Computation is performed by a host material, such as SiO$_2$, with numerous inclusions. The inclusions can be air holes, or any other material with an index different from that of the host medium. These inclusions strongly scatter light in both the forward and backward directions. The scattering spatially mixes the input light, rendering it a counterpart to linear matrix multiplication (Fig.1c) in a digital ANN. The nonlinear operation is realized via inclusions made of saturable absorbers, where they perform distributed nonlinear activation (Fig.1d). The locations and shapes of inclusions are the equivalent of weight parameters in digital ANNs, and their sizes are typically sub-wavelength.

![FIG. 1.](image)

(a) A conventional ANN architecture where the information propagates only in the forward direction (green arrow). (b) Proposed Nanophotonic Neural Medium (NNM). Passive neural computing is performed by light passing through nanostructured medium with both linear and nonlinear scatterers. (c) Full-wave simulation of light scattered by nanosstructures, which spatially redistribute the optical energy to different directions. (d) The output intensity of light passing through a saturable absorber with a thickness of $\lambda/2$. It is a nonlinear function of the incident wave intensity. This material is used as nonlinear activation as indicated by light blue color.

Fig.2 shows a NNM in action, where a two-dimensional (2D) medium is trained to recognize gray-scale handwritten digits. The dataset contains 5,000 different images, representative ones of which are shown in Fig.2a. Each time, one image, represented by $20 \times 20$ pixels, is converted to a vector, and then encoded as the spatial inten-
sity of input light incident on the left. Inside the NNM, nanostructures create strong interferences and light is guided toward one of ten output locations depending on the digit that the image represents, where the output with the highest share of energy intensity is categorized as the inferred class. Fig. 2b shows the fields created by two different hand-written 2 digits. Because of different shapes, the field patterns created by these two images are quite different but both lead to the same hot spot at the output, which correctly identifies the identity information as the number 2. As another example, Fig. 2c shows the case of two handwritten 8 digits that result in another hot spot. Here, the field is simulated by solving a nonlinear wave equation using Finite-Difference Frequency-Domain (FDFD) method. The size of the NNM is 80A by 20A, where $\lambda$ is the wavelength of light used to carry and process the information. The average recognition accuracy reaches over 75% for a test set made up of 1,000 images. The limited reported accuracy is due to the heavy constraints we set during the optimization for fabrication concerns. These constraints keep the medium dense, where it would have been otherwise made up of sparse sections of air and SiO$_2$. By relaxing these requirement or using larger medium sizes, the accuracy can be further improved.

NNM can provide ultra-high computing density by tapping into sub-wavelength features. In theory, the number of weight parameters is infinite: every atom in this medium can be varied to influence the wave propagation. In practice, a change below 10 nm would be considered too challenging for fabrication. Even at this scale, the potential number of weights exceeds 10 billion parameters per square millimeter for a 2D implementation. This is much greater computing density than both free-space [8] and on-chip optical neural networks [11] [14]. In addition, NNM has a few other attractive features. It has stronger expressive power than layered optical networks. In fact, layered networks are a subset of NNM, as a medium can be shaped into connected waveguides as a layered network. Furthermore, it does not have the issue of diminishing gradients in deep neural networks. Maxwell’s equations, as the governing principle, guarantee that the underlying linear operation is always unitary, which does not have diminishing or exploding gradients[15]. Lastly, NNM does not have to follow any specific geometry, and thus it can be easily shaped and integrated into existing vision or communication devices as the first step of optical preprocessing.

We now discuss the training of NNM. Although, one could envision in-situ training of NNM using tunable optical materials [29], here we focus on training in the digital domain and use NNM only for inference. The underlying dynamics of the NNM are governed by the nonlinear Maxwell’s equations, which, in the frequency domain, can be written as

$$L_{(r,E(r))}E_{(r)} = -i\omega J_{(r)} \tag{1}$$

where $L_{(r,E(r))} = (\nabla \times \nabla \times) / \mu - \omega^2 \varepsilon_{(r,E(r))}$, and $\mu$ and $\varepsilon$ are the permeability and permittivity. $J$ is the current source density which represents the spatial profile of the input light and is only non-zero on the left side of the medium. Waveguide modes or plane waves can also be used as the input, which are also implemented as current sources in numerical simulation. For a classification problem, the probability of the $i^{th}$ class label is given by

$$h_i = \left( \frac{\int |E(r)|^2 R_i(r)}{\sum_{i=1}^{10} \int |E(r)|^2 R_i(r)} \right) \tag{2}$$

which represents the percentage of energy at the $i^{th}$ receiver relative to the total optical energy that reaches all receivers. Here the profile function $R_i(r)$ defines the location of receivers, and is only non-zero at the position of the $i^{th}$ receiver. The training is performed by optimizing the dielectric constant $\varepsilon(r, E)$ similar to how weight parameters are trained in traditional neural networks. The cost function $C$ is defined by the cross entropy between the output vector $h$ and the ground truth $y$.

$$C = -\sum_{i=1}^{10} y_i \log(h_i) + (1 - y_i) \log(1 - h_i) \tag{2}$$

The ground truth $y$ is a one-hot vector. Digit 8 is represented as $y = (0, 0, 0, 0, 0, 0, 0, 0, 1, 0)$, for instance. The

FIG. 2. (a) NNM trained to recognize handwritten digits. The input wave encodes the image as the intensity distribution. On the right side of NNM, the optical energy concentrate to different locations depending on the image’s classification labels. (b) Two samples of the digit 2 and their optical fields inside NNM. As it can be seen, although the field distributions differ for the images of the same digit, they are classified as the same digit. (c) the same as (b) but for two samples of the digit 8.
accuracy of the cost function with respect to the dielectric constant $\varepsilon$ can be calculated point by point. For example, one could assess the effect of changing $\varepsilon$ at one spatial point; the change is only kept if the loss function decreases. This method has achieved remarkable success in simple photonic devices [13]. However, each gradient calculation requires solving full-wave nonlinear Maxwell’s equations. It is prohibitively costly for NNM, which could easily have millions of gradients. Here, we use Adjoint State Method (ASM) to compute all gradients in one step:

$$\frac{dC}{d\varepsilon} = -2\omega^2 \text{Re} \{ \lambda(r) E(r) \}$$

Here $\lambda(r)$ is a Lagrangian multiplier, which is the solution to the adjoint equation (Eq.4), in which the electric field $E(r)$ is obtained by solving Eq.1.

$$\frac{\partial C}{\partial E(r)} + \lambda(r)(L(r,E(r)) + \frac{\partial L(r,E(r))}{\partial E(r)} E(r))$$

$$+ \lambda(r) \frac{\partial L(r,E(r))}{\partial E(r)} E(r) = 0$$

The training process, as illustrated in Fig.3a, minimizes the summation of the cost functions $C$ for all training instances through Stochastic Gradient Descent (SGD). The process starts with one input image as the light source, for which we solve the nonlinear Maxwell’s equations in an iterative process (pink block in Fig.3a). The initial field is set to be random $E_0(r)$, which allows us to calculate the dielectric constant $\varepsilon(r, E_0(r))$. Then FDFD simulation is used to solve Eq.1, and the resulting electric field $E_1(r)$ is then used to update the dielectric constant. This iteration continues until the field converges, which usually takes around 10 iterations. The next step is to compute the gradient based on Eq.3. Once the structural change is updated, the training of this instance is finished.

The above process is repeated again, but for the next image in the training queue, instead of the same image. This gradient descent process is stochastic, which is quite different from typical use of ASM in nanophotonics [14] [15] where gradient descent is performed repeatedly for very few inputs until the loss function converges. In these traditional optimizations, the device needs to only function for those few specific inputs. If such processes were used here, the medium would do extremely well for particular images but fail to generalize and recognize other images.

For the example of hand-written digits shown in Fig.3, the training data set contains 4,000 images and their labels. Training typically converges after three epochs, which equates to solving wave equations 120,000 times. This process could take many days on a single CPU, so here we use two techniques to significantly accelerate the training. First, we note that the SGD is stochastic in nature. Therefore, there is no need to obtain an accurate solution to the nonlinear Maxwell’s equations. We can thus reduce the number of iterations from over 10 to around two or three iterations. This method is inspired by the single-shot optimization method used in aerodynamic design optimization [19]. Secondly, we use batch training with a batch size of 100 images. The gradient is calculated based on the sum of all cost functions for these 100 images, which allows us to perform 100 FDFD simulations in parallel instead of sequentially. Leveraging a high-throughput computing facility [20], the total training time is reduced by almost a factor of 100.

The gradient descent process treats the dielectric constant as a continuous variable, but in practice, its value is discrete, depending on the material used at the location. For example, in the case of a medium with $SiO_2$...
host material and linear air inclusions, the dielectric constants can either be 2.16 or 1. Discrete variables remain effective for neural computing [21]. Here, we need to take special care to further constrain the optimization process. This is done by using a level set function [22], where each of the two materials (host material and the linear inclusion material), is assigned to each of the two levels in the level-set function \( \phi(r) \) similar to ref [14] [13].

\[
\varepsilon(r) = \begin{cases} 
\varepsilon_{SiO_2} & \phi(r) < 0 \\
\varepsilon_{Air} & \phi(r) > 0 
\end{cases}
\]

The training starts with randomly distributed inclusions, both linear and nonlinear, throughout the host medium. The boundaries between two materials evolve in the training. Specifically, the level set function is updated by \( -v(r)|\nabla\phi| \), where \( v(r) \) is the gradient calculated by ASM and \( |\nabla\phi| \) indicates the boundary between the two constituent materials. Therefore, at each step, this method essentially decides whether any point on the boundary should be switched from one material to the other. Nonlinear sections perform the activation function and their location and shape are fixed in this optimization. They could also be optimized, which would be equivalent to optimizing structural hyperparameters in layered neural networks [23] [25].

As a specific example, we now discuss the training of the 2D medium shown in Fig. 2. The structural evolution is shown in Fig. 3b-d during the training. We start by randomly seeding the domain with dense but small inclusions. As the training progresses, the inclusions move and merge, eventually converging. The recognition accuracy for both training and test group improve during this process.

Next, we show another example based on a threedimensional (3D) medium, whose size is \( 4\lambda \times 4\lambda \times 6\lambda \). The inputs can be an image projected on the top surface of the medium. For example, we use a plane wave to illuminate a mask with its opening shaped into a hand-written digit as shown in Fig. 4. The full-wave simulation shows the optical energy is concentrated on the location with correct class label, in this case 6. (b) The confusion matrix. The rows on the matrix show true labels of the images that have been presented as input, and the columns depict the labels that the medium has classified each input. Therefore the diagonal elements show the number of correct classifications out of every ten samples.

![Diagram showing 3D nanophotonic neural medium case. Different colors illustrate varying values of permittivity. The input image is projected onto the top surface. Computing is performed while the wave propagates through the 3D medium. The field distribution on the bottom surface is used to recognize the image. Full-wave simulation shows the optical energy is concentrated on the location with correct class label, in this case 6.](image)

**CONCLUSION**

From the perspective of optics, the functions of most nanophotonic devices can be described as mode mapping [27]. In traditional nanophotonic devices, mode mapping mostly occurs between eigen modes. For example, a polarization beam splitter [13] maps each polarization eigen-mode to a spatial eigen-mode. Here, we introduce a class of nanophotonic media that can perform complex and nonlinear mode mapping equivalent to artificial neural computing. In comparison, today’s optical neural computing mostly follows layered structures. While highly efficient for digital computing, layered structures could be counter-productive in optical analog computing. For practical applications that routinely use millions of connections, it can be challenging to implement deep and dense layers of optical components in a compact form. Nevertheless, the concept of NNM shows that any nanostuctures could be optimized to perform neural computing without the rigid constraint of layer structures. Combined with ultra-high computing density, NNM could be used in a wide range of information devices as the analog preprocessing unit.

I. ACKNOWLEDGEMENTS

The authors thank W. Shin for his help on improving the computational speed of the implementation of this method. The work was financially supported by DARPA.
Young Faculty Award program.

[1] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic, Nature Photonics 11, 441 (2017).

[2] M. Hermans, M. Burn, T. Van Vaerenbergh, J. Dambre, and P. Bienstman, Nature communications 6, 6729 (2015).

[3] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, Optica 5, 864 (2018).

[4] S. R. Skinner, E. C. Behrman, A. A. Cruz-Cabrera, and J. E. Steck, Applied optics 34, 4129 (1995).

[5] P. R. Prucnal and B. J. Shastry, Neuromorphic photonics (CRC Press, 2017).

[6] H. G. Chen, S. Jayasuriya, J. Yang, J. Stephen, S. Sivaramakrishnan, A. Veeraraghavan, and A. Molnar, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016) pp. 903–912.

[7] J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, Optica 5, 756 (2018).

[8] X. Lin, Y. Rivenson, N. T. Yardime, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, Science 361, 1004 (2018).

[9] M. Hermans and T. Van Vaerenbergh, arXiv preprint arXiv:1510.03776 (2015).

[10] H.-G. Park, S.-H. Kim, S.-H. Kwon, Y.-G. Ju, J.-K. Yang, J.-H. Baek, S.-B. Kim, and Y.-H. Lee, Science 305, 1444 (2004).

[11] J. D. Joannopoulos, S. G. Johnson, J. N. Winn, and R. D. Meade, Photonic crystals: molding the flow of light (Princeton university press, 2011).

[12] W. Cai and V. Shalaev, Optical metamaterials: fundamentals and applications (Springer Science & Business Media, 2009).

[13] B. Shen, P. Wang, R. Polson, and R. Menon, Nature Photonics 9, 378 (2015).

[14] A. Y. Piggott, J. Petykiewicz, L. Su, and J. Vucković, Scientific Reports 7, 1786 (2017).

[15] L. Su, A. Y. Piggott, N. V. Sapra, J. Petykiewicz, and J. Vuckovic, ACS Photonics (2017).

[16] W. Shin, “MaxwellFDFD Webpage,” (2015), https://github.com/wsshin/maxwellfdfd.

[17] H. J. Caulfield, J. Kinser, and S. K. Rogers, Proceedings of the IEEE 77, 1573 (1989).

[18] L. Jing, Y. Shen, T. Dubcek, J. Peurifoy, S. Skirlo, Y. LeCun, M. Tegmark, and M. Soljačić, in Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 70, edited by D. Precup and Y. W. Teh (PMLR, International Convention Centre, Sydney, Australia, 2017) pp. 1733–1741.

[19] S. Günther, N. R. Gauger, and Q. Wang, Journal of Computational Physics 328, 387 (2017).

[20] CHTC contributors, “Center for high throughput computing (chtc).”

[21] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, arXiv preprint arXiv:1602.02830 (2016).

[22] C. Li, C. Xu, C. Gui, and M. D. Fox, IEEE transactions on image processing 19, 3243 (2010).

[23] J. Bergstra and Y. Bengio, Journal of Machine Learning Research 13, 281 (2012).

[24] J. Snoek, H. Larochelle, and R. P. Adams, in Advances in neural information processing systems (2012) pp. 2951–2959.

[25] S. Saxena and J. Verbeek, in Advances in Neural Information Processing Systems (2016) pp. 4053–4061.

[26] G. D. Marshall, A. Politi, J. C. Matthews, P. Dekker, M. Ans, M. J. Withford, and J. L. O’Brien, Optics express 17, 12546 (2009).

[27] D. A. Miller, Optics Express 20, 23985 (2012).