Photonic pattern reconstruction enabled by on-chip online learning and inference

Journal of Physics: Photonics

OPEN ACCESS

RECEIVED
4 December 2020

REVISED
16 January 2021

ACCEPTED FOR PUBLICATION
5 February 2021

PUBLISHED
25 February 2021

PAPER

Photonic pattern reconstruction enabled by on-chip online learning and inference

Bicky A Marquez[1], Zhimu Guo[1], Hugh Morison[1], Sudip Shekhar[2], Lukas Chrostowski[2], Paul Prucnal[3] and Bhavin J Shastri[1]

[1] Department of Physics, Engineering Physics and Astronomy, Queen's University, Kingston, ON KL7 3N6, Canada
[2] Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada
[3] Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, United States of America

E-mail: bama@queensu.ca

Keywords: neuromorphic photonics, brain-inspired computing, photonic integrated circuits, recurrent neural network, artificial intelligence hardware

Abstract

Recent investigations in neuromorphic photonics exploit optical device physics for neuron models, and optical interconnects for distributed, parallel, and analog processing. Integrated solutions enabled by silicon photonics enable high-bandwidth, low-latency and low switching energy, making it a promising candidate for special-purpose artificial intelligence hardware accelerators. Here, we experimentally demonstrate a silicon photonic chip that can perform training and testing of a Hopfield network, i.e. recurrent neural network, via vector dot products. We demonstrate that after online training, our trained Hopfield network can successfully reconstruct corrupted input patterns.

1. Introduction

The binary nature of conventional digital computers hinders the design of direct one-to-one maps between massively parallel neural systems and the digital machine [1]. An alternative way uses specialized analog machines, where the operations of a massively parallel neural architecture are embedded in the hardware itself. Analog devices have shown to perform efficient operations based on their device physics [2–5]. Therefore, an analog computer would be more suitable to model analog structures such as neurons. Thus, analog special-purpose hardware can be designed to emulate the behavior of artificial neural networks (ANNs).

One of the primary bottlenecks of digital networks' implementations is efficiently computing the matrix multiplications required for training and inference [6, 7]. Since the field of artificial intelligence (AI) is mostly perceptron-based [8], matrix multiplications became its core operation. Most significant advances in AI have been achieved using a perceptron as an artificial model of a neuron. Perceptrons encompass the most general functions of biological neurons, which can be summarized as weighted-additions nonlinearly transformed by activation functions [9]. Weighted-additions also represent the core operation for dot products between matrices. Therefore, dedicated hardware accelerators for perceptron-based neural networks would be designed to perform operations between matrices.

Dedicated analog hardware for ANNs would require to physically model every single individual component of such networks. This is an expensive demand considering that modern deep networks sizes scale up to thousands (or millions) of neurons to solve AI related tasks; for example, Google's state-of-the-art large-scale language model BERT is modeled by 110 M parameters which describe neurons [10]. In order to overcome such a challenge, the high speed and parallelism that analog photonic systems can achieve makes them natural candidates for efficient brain-inspired computing [11, 12]. In this work, we present a photonic integrated circuit able to perform online training and testing of a perceptron-based ANN for pattern reconstruction. We introduce a Hopfield network [13] with a simple training and testing scheme based around matrix multiplications. The task consists of training and testing this recurrent network three times to recognize three different faulty patterns. These patterns are represented as 4×4 matrices that model the

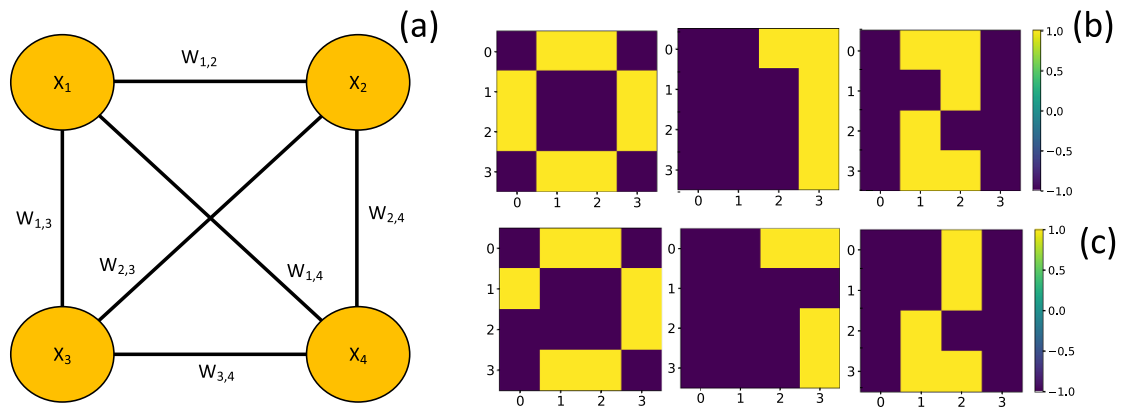

Figure 1. (a) Schematic diagram of a 4 × 4 Hopfield network. Images of (b) numbers {0, 1, 2} and (c) corrupted versions of the same numbers.

image of numbers 0, 1 and 2 with some defects. The motivation behind choosing a Hopfield network is that it can solve complex tasks with a simplified training and testing methodology.

Emerging technologies based around photonics attempt to enhance computing performance for AI applications. Due to their speed, energy efficiency and reconfigurability, such matrix multiplications will be performed using photonic devices. In particular, we consider the use of the broadcast-and-weight protocol as it has demonstrated to be able to carry out fully parallel matrix operations using wavelength division multiplexing [11, 14, 15]. This approach uses micro-ring resonators (MRRs) to directly encode different matrix elements as amplitude values in parallel optical channels [16].

In order to train and test a Hopfield network in our platform, we use a bank of off-chip tunable lasers and on-chip silicon MRRs to implement the elements of each matrix considered for both stages. Both the lasers and the MRRs encode matrix elements as optical amplitude modulated values. Tuning the power of a laser allows for a straightforward representation of matrix elements in optics. And tuning a given MRR on and off resonance changes the transmission of an optical signal through that MRR, effectively multiplying such a signal with a desired value. MRR-based architectures have proven to solve pattern-recognition related tasks as it can be found in the work of Feldmann *et al* with spiking-based neural nets [17]. In our present work, we will use a perceptron-based network to perform a pattern reconstruction task of corrupted patterns using a bank of on-chip MRRs.

2. Hopfield neural network

A Hopfield architecture is a recurrent neural network typically used as an associative memory [18]. Invented by J Hopfield in 1982 [13], this network memory property allows for pattern reconstruction of faulty datasets. In figure 1(a), we show a 4 × 4 symmetric Hopfield network, where each perceptron-based neuron is represented by each x_i-circle. The synaptic weights $w_{i,j}$ correspond to all-to-all connections between neurons, defined by bipolar numbers {−1,+1}. This fully connected network has inputs and outputs described by vectors whose elements are also bipolar numbers. The activation function is modeled by a sign function, $\mathrm{sgn}(\mathcal{O}) = +1$, if $\mathcal{O} \geqslant 0$; and $\mathrm{sgn}(\mathcal{O}) = -1$, otherwise. An input pattern of dimension 4 × 1 feeds all nodes simultaneously. In the case where an input pattern is represented by a 4 × 4 matrix, then each 4 × 1 vector composing such a matrix should be inputted separately. Therefore, 4 iterations would be required to complete this task. Figure 1(b) shows three examples of 4 × 4 input patterns corresponding to the images of numbers {0, 1, 2} that our Hopfield network has to memorize.

The network in the training stage uses these patterns to calculate the elements $w_{i,j}$ of three weight matrices — one matrix per image. For this particular task a weight matrix W can be estimated by multiplying every input pattern x with itself, and then set up all diagonal values equal to zero,

$$W = x^T \cdot x - I, \tag{1}$$

where T is the transpose function and I the identity matrix. Since the diagonal values are squared, they should be removed from the Hopfield network's memory to avoid keeping incorrect contributions in it. Notice that this step is important when we store multiple patterns in the network's memory. However, if only one pattern x_k is stored in the network's memory per task k, then the weight matrix can be calculated as

$W_k = x_k^T \cdot x_k$. In this work, we will store only one pattern k in memory per task, therefore we can skip the diagonal elements subtraction step, and define $k = 0, 1, 2$ related to each image $\{0, 1, 2\}$.

The inference stage consists of the reconstruction of partially broken input patterns. In figure 1(c), three patterns corresponding to corrupted versions of images $\{0, 1, 2\}$ are shown. The task of the network is to reconstruct an image based on a stored pattern similar to the corrupted input. For this to happen, we multiply a corrupted image χ_k by the weight matrix W_k,

$$O_k = \chi_k \cdot W_k, \tag{2}$$

where O_k is the result of the 4×4 matrix multiplication. Then, the output pattern is obtained once a sign activation function is applied to that result,

$$y_k = \text{sign}(\chi_k \cdot W_k). \tag{3}$$

The results of such experiments are determined by directly comparing output y_k and target x_k through the mean absolute error (MAE),

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_k^i - x_k^i|, \tag{4}$$

where N is the number of pixels per image. In this experiment we found that the network could reconstruct the numbers with high accuracy (1–MAE)%, i.e. low MAE. The corrupted versions of numbers 0, 1 and 2 were reconstructed with accuracies of 100%, 93.75% and 93.75%, respectively.

3. Photonic vector dot product

As the operations described above are based on dot products between vectors containing four elements each, we will demonstrate that such operations can be performed using a bank of four on-chip silicon MRRs. Let us define two vectors $\{\mathbb{A}, \mathbb{B}\}$ that will represent any theoretical set of vectors considered in the forthcoming experimental dot products $(\mathbb{A} \cdot \mathbb{B})$. In figure 2(a), the elements of such vectors are experimentally represented. To encode elements of the vector $\mathbb{A}$, we vary the power intensity P_i (with $i = 1, 2, 3, 4$) of four tunable lasers. Additionally, each laser provides optical signals to the on-chip circuit at wavelengths λ_i through a set of three 50/50 beam couplers and an erbium-doped fiber amplifier (EDFA). These devices multiplex and amplify optical signals of different wavelengths coming from the lasers before they enter the chip. The EDFA is utilized to match the launch power of the lasers with the power coupled on to the chip, which was attenuated by the beam couplers.

The elements of the second vector $\mathbb{B}$ are implemented by four on-chip add-drop MRRs. MRRs are devices capable of trapping light coming from the input (IN) port at frequencies λ_i at which they resonate, according to their physical characteristics. The resonance frequency can be obtained from the wavelength equation $\lambda_R = 2\pi R n_{\text{eff}}/m$, where R is the radius of the ring, m is an integer number and n_{eff} is the effective refractive index. The on-chip weight bank shown by figure 2(b) is fabricated on a silicon-on-insulator wafer with a silicon thickness of 0.22 μm and a buried oxide thickness of 2 μm. Each MRR$_i$ has an aluminum-based pad where the voltage will be applied and a common ground. These rings have slightly difference ring radii ($\{8.0, 8.1, 8.2, 8.3\}$ μm) to avoid resonance collision. The physical distance between one MRR to the next MRR is 50 μm. The quality factor of the MRR is $\sim$6000 for a gap of 0.2 μm between the ring and the bus waveguides. Each ring was designed with an n-doped heater [20, 21]. The heater was designed with an n-doped silicon rib waveguide on top of SiO_2, where two N++ doped overlayers on the side of the waveguide (750 nm of separation) serve as contacts. By applying a voltage V_i on such contacts, we can thermally tune the waveguide. Therefore, this n-doped heater functions as a thermo-optic tuner. Also, it can act as a detector by lowering the electrical resistance across it.

A wide variety of intensity values can be represented by an MRR through the tuning of the waveguide refractive index by means of an applied voltage V_i. The vector elements A and B are transferred to the experiment from the computer (PC) via power intensity P_i (tunable lasers) and applied voltage V_i (Keithley 2600 source meters). The summation of the four products is outputted by the photodetector, connected to the through (THRU) port of the last MRR, and stored in the PC. There, the activation function is applied and the prediction error is estimated.

In this work we only use IN and THRU ports to perform dot products. Figure 2(c) shows the optical spectrum of the on-chip MRR bank obtained with the fine sweep function of an optical spectrum analyzer (OSA), Aragon BOSA 400. The transmission vs wavelength profile with no-voltage applied to their embedded heaters. This configuration allows for analog parallel dot products using light as a medium for data processing.

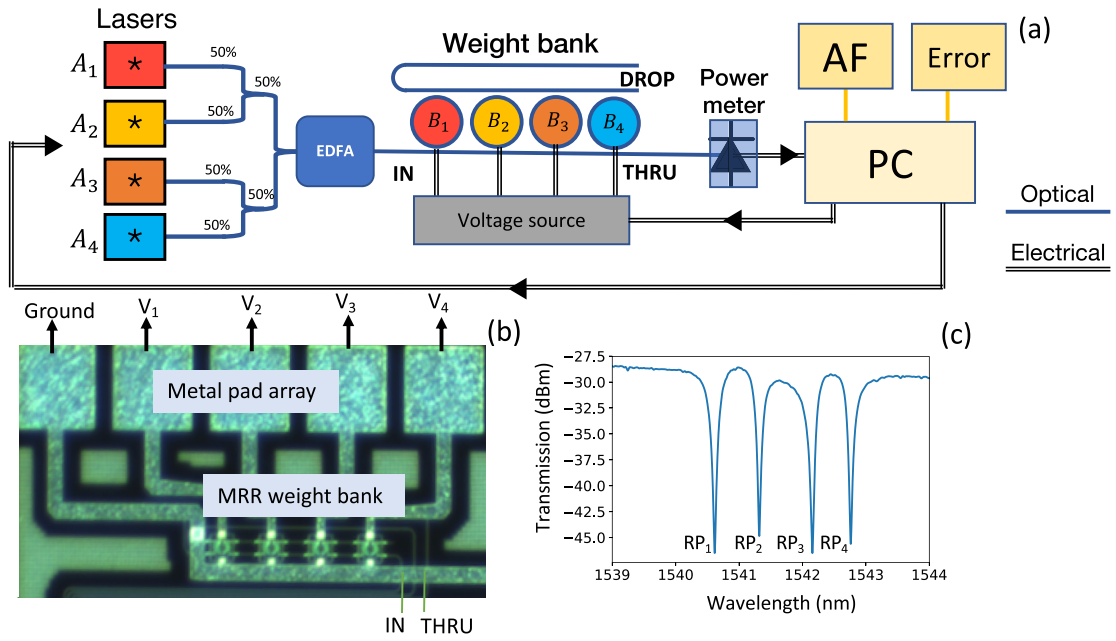

Figure 2. (a) Schematic illustration of the photonic vector dot product architecture between vectors $\mathbb{A}$ and $\mathbb{B}$. The elements of vectors $\mathbb{A}$ and $\mathbb{B}$ are implemented as power values of tunable lasers and transmission amplitudes of MRRs, respectively. A photodetector performs the summation part of the weighted addition. The data is collected by the PC where the activation function and the error are calculated. Data bidirectional flow from the PC to the devices (tunable lasers, power-meter and voltage sources) is performed through GPIB-to-USB interface adapters, and it is controlled by lightlab—an open source software developed by Princeton University [19]. (b) Picture of the on-chip MRR-based weight bank fabricated by Advanced Micro Foundry PTE LTD. The on-chip area covered by the weight bank is 540×760 μm. (c) Measured transmission vs wavelength plot of its through (THRU) port showing four resonance peaks (RP_i, with $i = 1, 2, 3, 4$).

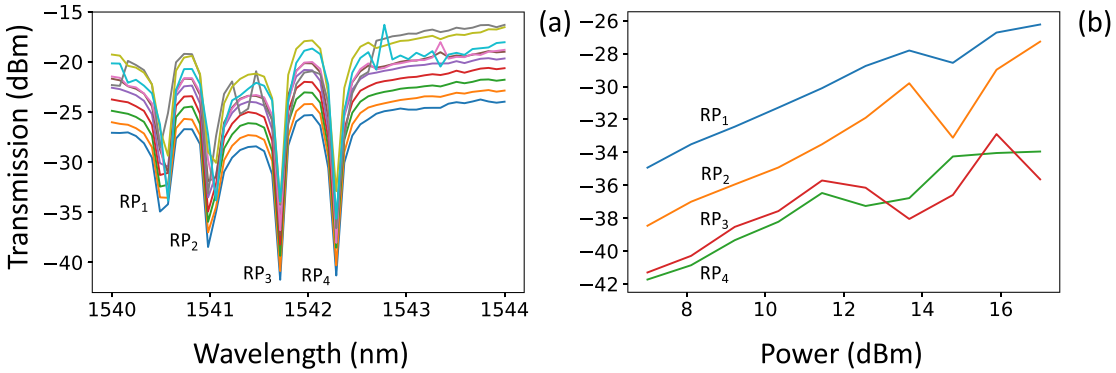

Figure 3. THRU-port measured: (a) transmission values at wavelengths $RP_1 = 1540.5$ nm, $RP_2 = 1541.0$ nm, $RP_3 = 1541.7$ nm and $RP_4 = 1542.4$ nm, as functions of the input power.

3.1. Experimental look-up tables

In order to experimentally represent elements of the two vectors $\{\mathbb{A}, \mathbb{B}\}$, the laser power intensity P_i as well as the the driving voltage V_i applied to the MRRs have to be properly chosen for the tasks. P_i and V_i are stored in the PC memory, and get transferred from it to the tunable lasers and voltage sources by GPIB-to-USB interface adapters. The construction of look-up tables for each device facilitates the process of properly chosen P_i and V_i for the task under consideration. To optically encode $\mathbb{A}$ elements we create a look-up table by sweeping the power of our Pure Photonics PPCL600 micro-ITLA tunable lasers around the resonance peak of each MRR as shown by figure 3(a). In the figure, the different curves correspond to different laser powers that are specified in the abscissa of figure 3(b), and the ordinate represents the evolution of transmission values corresponding to all resonances (RP_1, RP_2, RP_3, RP_4).

The transmission-voltage profiles show monotonous increments up to an input power of 11 dBm. After that point, RP_3 and RP_4 start behaving non-linearly. The source of non-linearity is the result of the interaction between high levels of optical power and the silicon material. High optical power levels in silicon result in non-linear effects like the Kerr effect and two-photon absorption which can result in free-carrier dispersion

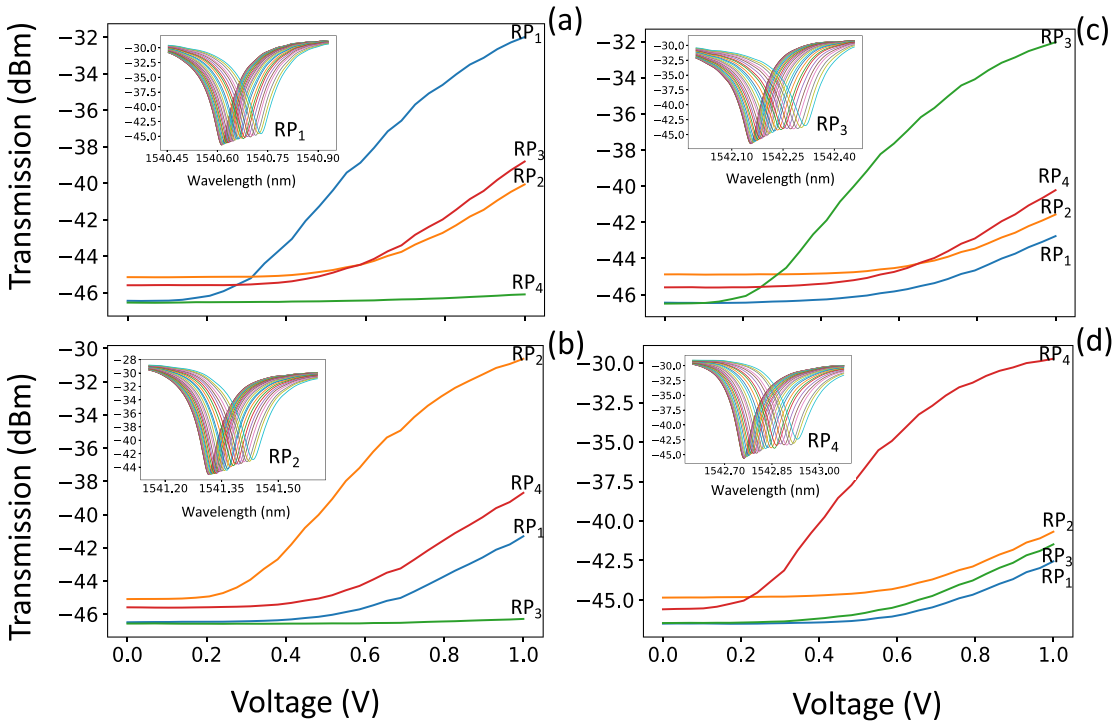

Figure 4. THRU-port transmission vs voltage profiles at each resonance peak RP_i for driving voltages applied to: (a) MRR_1, (b) MRR_2, (c) MRR_3 and (d) MRR_4.

and free-carrier absorption. These effects change the refractive index of the MRR affecting its resonances and desired weight value. Same happens with RP_1 and RP_2 after input power surpasses 13 dBm. This experiment shows how $\mathbb{A}$ elements can be represented in the optical domain as power intensities P_i. In this example, we represented 10 possible values that our processor can use for computation. For $N = 10$, the bit resolution that represents the number of vector elements that can be optically encoded by the tunable lasers is $\log_2(N) \approx 3.3$. For experiments that require a higher bit resolution, other alternatives to optically encode $\mathbb{A}$ elements should be found. Therefore, for experiments that require a higher bit resolution, alternative venues to optically encode $\mathbb{A}$ elements should be found.

The second look-up table will correspond to optical elements of vector $\mathbb{B}$. A driving voltage V_i applied to each MRR will allow us to shift the resonance peak around laser wavelength λ_i. Figure 4 shows plots of transmission as function of the applied voltage for the four on-chip MRRs used in this experiment. The transmission vs wavelength sub-plots were obtained by using an Aragon BOSA 400, where the OSA's internal tunable laser was used to finely characterize each MRR individually. In all panels of the figure, a monotonous increment of the transmission-voltage profiles is revealed for a constant input power of 9 dBm. In particular, figure 4(a) shows how RP_1 transmission value increases with a driving voltage to MRR_1, while $\{RP_2, RP_3, RP_4\}$ remain constant since 0 V are applied to all other MRRs. This behavior is maintained up to an applied voltage of 0.4 V on MRR_1, and 0 V for all other MRRs. An interesting phenomenon occurs for applied voltage values over 0.4 V, where $\{RP_2, RP_3, RP_4\}$ transmission values start increasing with the increment of the voltage. We associate this phenomenon with thermal crosstalk from MRR_1 that reaches neighbor MRRs [16]. Such a thermal crosstalk can also be seen in figures 4(b)–(d)), where the driving voltage was applied to MRR_2, MRR_3 and MRR_4, respectively.

We therefore consider a driving voltage of 0.4 V as a threshold value to avoid thermal crosstalk related distortions. This experiment shows that at least 20 values can be represented per MRR through driving voltages V_i—which translates in a bit resolution of 4.32. The existence of thermal crosstalk between MRRs separated by a physical distance of 50 μm reduces the measured bit resolution of MRRs by half. However, if we choose a range of driving voltages $V_i \in [0, 0.5]$, and smooth the sweep in that range, a higher bit resolution with insignificant thermal crosstalk can be achieved as demonstrated in reference [16].

In the following, these results will be used as a base to establish look-up tables for the Hopfield network. Such tables need to be properly constructed for each experiment. For instance, a photonic processor with less than 8 bits of resolution could not solve CIFAR-10 (training and testing included) [22]. As it will be demonstrated in the following section, the bit resolution of our photonic processor should not limit the

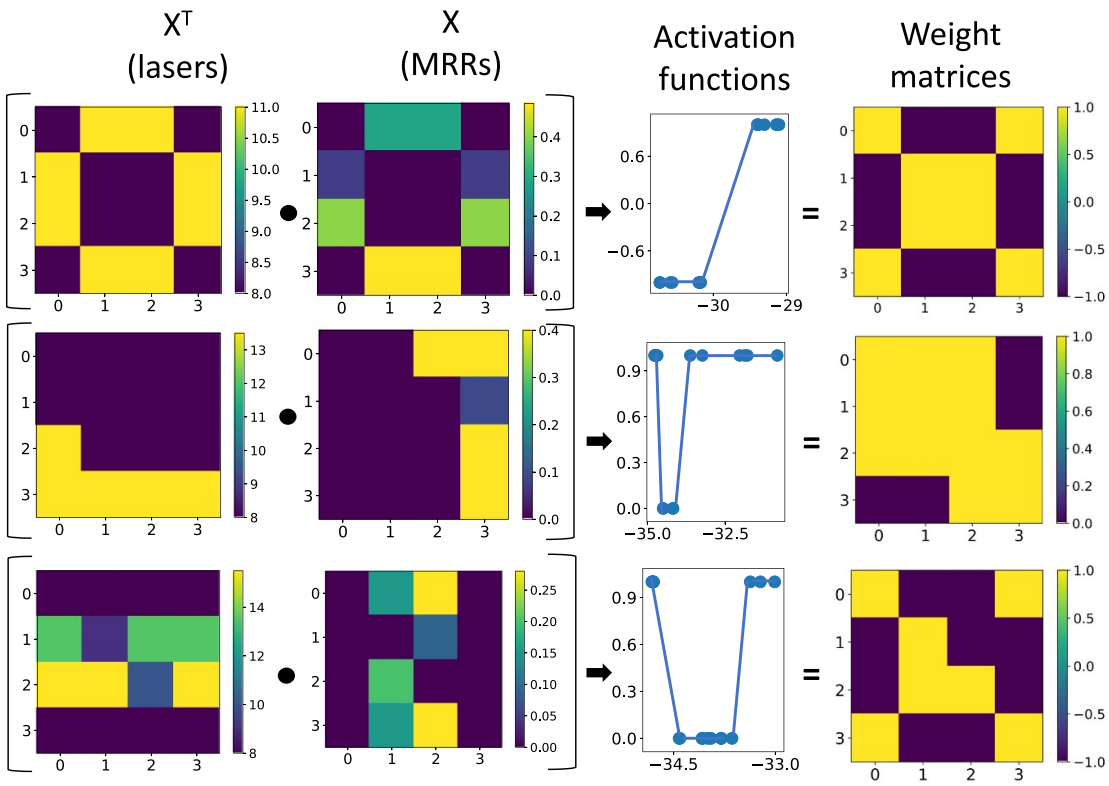

Figure 5. Training stage of the Hopfield network to memorize images 0, 1 and 2. The first two columns show matrices x_k^T and x_k defined by laser power and driving voltages, respectively. The next two columns show activation functions which allow for direct comparison between theoretical, and experimental weight matrices.

performance of the Hopfield network for pattern reconstruction, as it only requires two bits of resolution from our devices per vector dot product.

4. Online training of the Hopfield network

As described in section 2, training the Hopfield network to recognize one pattern x_k at a time requires the estimation of a weight matrix W_k through the dot product between input matrices $x_k^T \cdot x_k$. In the first two columns of figure 5, we show how this series of experiments can be done. To elaborate further on one example, let us define the image of number zero as a 4×4 matrix x_k constructed with $\{-1,+1\}$, followed by the same matrix but encoded with driving voltage values $V_0(V)$:

$$x_0 = \begin{bmatrix} -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 \\ -1 & +1 & +1 & -1 \end{bmatrix} \rightarrow V_0(V) = \begin{bmatrix} 0.00 & 0.28 & 0.28 & 0.00 \\ 0.09 & 0.00 & 0.00 & 0.09 \\ 0.40 & 0.00 & 0.00 & 0.40 \\ 0.00 & 0.48 & 0.48 & 0.00 \end{bmatrix} V. \tag{5}$$

This experiment takes into account that: (i) number -1 is encoded through driving voltages of 0 V, and number $+1$ is encoded by driving voltages of $\neq$0 V; (ii) any driven voltage set applied to two adjacent MRRs should have different values. This trick allows us to avoid very high levels of thermal crosstalk. For instance, such levels can be found when we try to encode the first column vector $[-1,+1,+1,-1]$ as $[0.00, 0.40, 0.40, 0.00]$. Here, the adjacent rings MRR_2 and MRR_3 will generate an important amount of heat which reaches adjacent MRRs. However, if the first column vector $[-1,+1,+1,-1]$ is encoded as $[0.00, 0.09, 0.40, 0.00]$, MRR_1 and MRR_4 do not get strongly affected by the heat released by the two middle rings.

The transpose of the number zero x^T is experimentally implemented with different power intensities of the tunable lasers as follows:

$$x_0^T = \begin{bmatrix} -1 & +1 & +1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 \\ -1 & +1 & +1 & -1 \end{bmatrix} \rightarrow \mathrm{P}_0(\mathrm{dBm}) = \begin{bmatrix} 8 & 11 & 11 & 8 \\ 11 & 8 & 8 & 11 \\ 11 & 8 & 8 & 11 \\ 8 & 11 & 11 & 8 \end{bmatrix} \mathrm{dBm}. \tag{6}$$

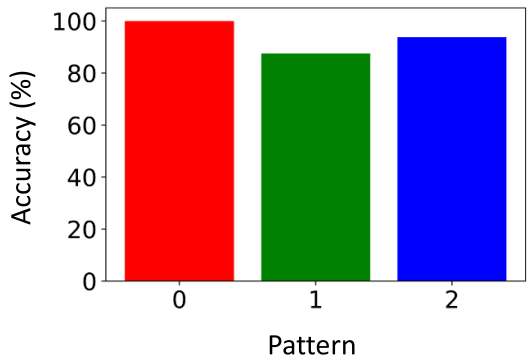

Figure 6. Accuracy ((1 − MAE)%) between theoretical and experimental weight matrices for patterns 0, 1 and 2.

In this case, numbers {−1,+1} are encoded as 8 and 11 dBm, respectively. Although, for all experiments, we considered number −1 encoded as any number in the range [7, 9] dBm, and number +1 as any number in [11, 16] dBm. This step allows for an enhancement of the action of a particular MRR while performing the dot product since each grating coupler has a 6 dB loss. Therefore, the estimated power coupled on to the chip will be attenuated by a factor of 6.

The experiment starts with a calibration that is carried out by a single tunable laser to sweep the wavelength across the range [1540, 1544] nm. This step allows us to identify where each RP_i is. The next step consists of programming the tunable lasers to match such wavelengths λ_i. Then, we proceed to upload values P_i and V_i from the look-up tables to the tunable lasers and Keithleys driving the MRRs, respectively. The voltage at the photodetector is consequently recorded. Each dot product between two 4×4 matrices x_k^T and x_k is carried out by our photonic processor in 16 iterations, since 16 vector dot products have to be performed to build up matrix W_k. A recalibration consisting of updating the lookup tables between sets of iterations is not mandatory but recommended to compensate for any drift of the RP_i—although the system was found to be stable for long periods of time (∼5 h). The stability of this system can be improved by adding a custom made temperature controller for this type of architecture.

The same process was followed with the purpose of determining W_1 and W_2 for images 1 and 2, respectively. All results were registered in the PC memory, followed by off-line post-processing. The post-processing phase is divided in two steps: (i) implementation of activation functions, and (ii) comparisons between experimental and theoretical weight matrices. Although training stages of Hopfield networks usually do not apply any activation function to the weight matrix, we apply them for comparison purposes. As we seek to compare theoretical and experimental W_k, we define piecewise activation functions for each experimental weight matrix such that the resultant matrix is described with numbers {−1,+1}—just like its theoretical counterpart. The third and fourth columns of figure 5 show the resultant activation functions and weight matrices.

The result of the post-processing phase is shown by figure 6, where the accuracy (defined as the percentage of (1-MAE)) of such an off-line experiment can be found. As it can be seen, post-processed weight matrices reached accuracies for over 85.6%. In the case of number zero, the achieved accuracy was 100%. Given the weight matrices for all images considered in this work, we can proceed to test the network with faulty patterns.

5. Online inference of corrupted patterns

Once the weight matrices have been estimated and compared to their theoretical counterparts, we proceed to test inference performance. The corrupted patterns corresponding to 0, 1 and 2 are shown in the first column of figure 7. These patterns have been encoded as laser power values in optics. Next, we use the post-processed weight matrices obtained in the previous section to carry out this inference stage (see second column of the same figure). Weight matrix elements have been encoded as voltage values through Keithleys driving the MRRs. Despite the fact that some of such weight matrices may negatively impact performance, their use in this experiment allows us to also test the training performance carried out in the previous section.

After all results were registered in the PC memory, we proceeded to implement off-line post-processing. The activations functions and the output patterns can be found in the last two columns of figure 7. We define piecewise activation functions for each experimental output matrix such that the resultant matrix is described with numbers {−1,+1}. The shapes of the three output patterns show qualitatively different

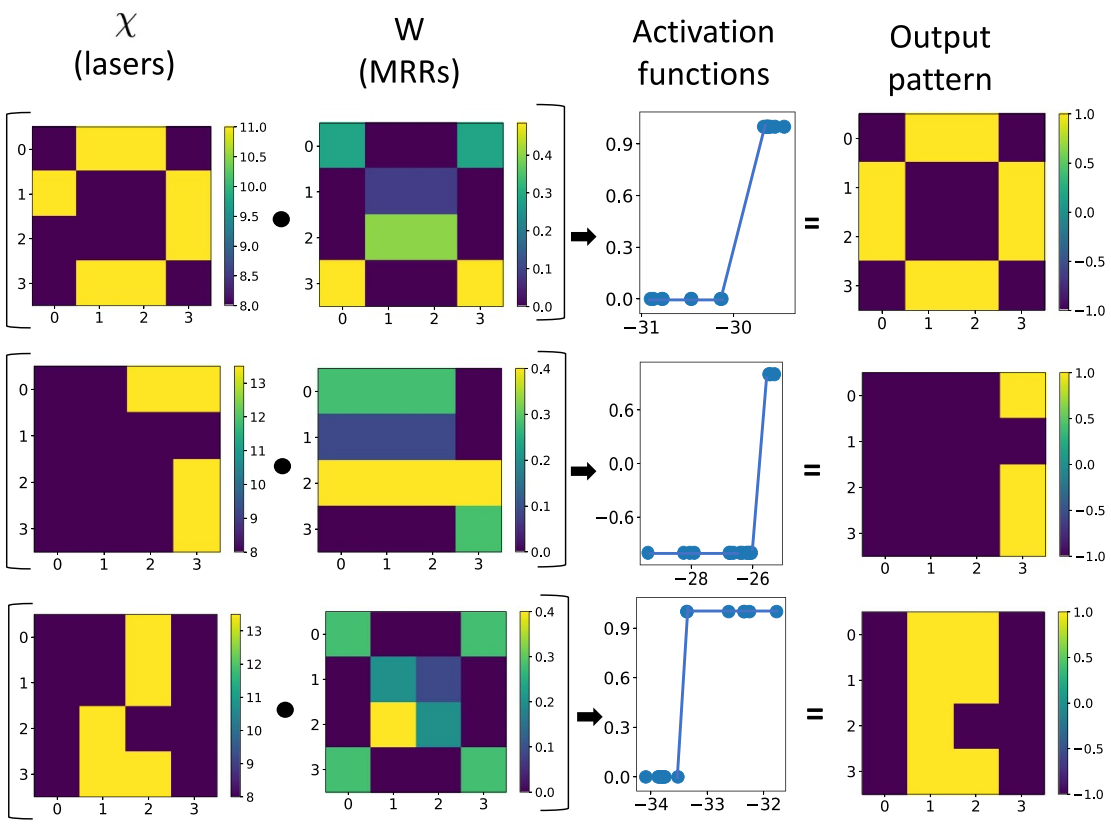

Figure 7. Inference stage of the Hopfield network to reconstruct corrupted images 0, 1 and 2. The first two columns show matrices χ_k and W_k defined by laser power and driving voltages, respectively. The next two columns show activation functions which allow for direct comparison between memorized, and reconstructed patterns 0, 1 and 2.

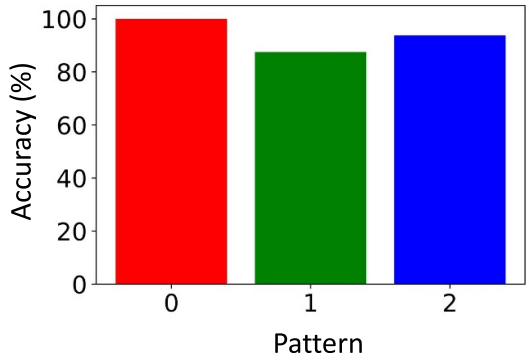

Figure 8. Accuracy ((1-MAE)%) between memorized and reconstructed patterns 0, 1 and 2.

performances. As it can be seen, the output pattern corresponding to number zero was properly reconstructed, while the other two images look less accurately reconstructed. The estimation of the accuracy will help us decide whether the trained network could reconstruct all the three patterns or not.

Figure 8 shows the percentage of the accuracy corresponding to the testing stage. Here, the reconstruction of number zero was accurately performed (100%). The reconstruction of numbers one and two showed lower performances. For instance, the recognition of number one was the less accurate of the three experiments (86.5%). The reconstruction case of number two was successful except for one pixel, therefore, the accuracy reached 93.75%. A direct comparison between theoretical and experimental accuracies shows that only the performance at reconstructing pattern one was negatively impacted.

Interestingly, both training and inference stages of the number one led to more errors than the other numbers. It seems that for this particular configuration, symmetric matrices are better calculated than asymmetric ones—where many MRRs are being used at once. Tasks where three MRRs are being used at once are associated to low performances in training and testing stages. The thermal crosstalk might be

behind issues associated with the use of several MRRs at once. Therefore, improvements should be made on our chips to avoid such a thermal crosstalk.

6. Conclusion

We have shown that MRR-based photonic integrated circuits can implement training and inference stages of a Hopfield network for pattern reconstruction. We demonstrated that experimental vector operations can be successfully carried out using a set of off-chip tunable lasers and on-chip MRRs. Despite the fact that the bit resolution can be reduced due to thermal crosstalk between adjacent rings, our experiment was not directly affected by such a limitation as only one bit of resolution per device was required both in training and testing stages. Thermal crosstalk was also found to be contributing to lower performances when three MRRs were used at once. Therefore, the separation between adjacent rings will be increased in further experiments if thermo-optic phase shifters are used. A possible solution to this problem is to actively cool down the surface of the chip, so that the heat released by each MRR remains local. For this to happen, we would need to change the testing methodology that we followed to perform this experiment, which is based on the use of a DC probe and a V-groove. More improvements can be achieved if graphene-based [23] or plasma dispersion (PN-junction) [24] modulators replace on-chip heaters, since they do not rely on heat to implement matrix elements in optics.

The reconfigurability feature of our photonic circuits allows for the design of special-purpose analog machines that can implement other types of ANNs as well, such as convolutional neural networks [25]. Due to the generality of their matrix multiplication architecture, our photonic processors can potentially become the core element of general-purpose analog computing solutions as described in reference [7].

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants Program and the Collaborative Research and Development (CRD) Grant with Huawei Canada.

ORCID iDs

Bicky A Marquez https://orcid.org/0000-0002-1644-8446
Bhavin J Shastri https://orcid.org/0000-0001-5040-8248

References

[1] Maley C 2018 Toward analog neural computation *Minds Mach.* **28** 77–97
[2] Merolla P A *et al* 2014 A million spiking-neuron integrated circuit with a scalable communication network and interface *Science* **345** 668–73
[3] Ríos C, Youngblood N, Cheng Z, Le Gallo M, Pernice W H P, Wright C D, Sebastian A and Bhaskaran H 2019 In-memory computing on a photonic platform *Sci. Adv.* **5** eaau5759
[4] Li C *et al* 2019 Long short-term memory networks in memristor crossbar arrays *Nat. Mach. Intell.* **1** 49–57
[5] Tait A N, Ferreira De Lima T, Nahmias M A, Miller H B, Peng H T, Shastri B J and Prucnal P R 2019 Silicon photonic modulator neuron *Phys. Rev. Appl.* **11** 1
[6] Fatahalian K, Sugerman J and Hanrahan P 2004 Understanding the efficiency of GPU algorithms for matrix-matrix multiplication *Proc. ACM SIGGRAPH/ Conf. on Graphics Hardware HWWS '04* (New York: Association for Computing Machinery) pp 133–7
[7] Shastri B J, Tait A N, de Lima T F, Pernice W H P, Bhaskaran H, Wright C D, and Prucnal P R 2020 Photonics for artificial intelligence and neuromorphic computing (arXiv:2011.00111)
[8] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[9] Rojas R 1996 *Neural Networks: A Systematic Introduction* (Berlin: Springer)
[10] Korhonen A, Traum D and Marquez L 2019 *Proc. 57th Annual Meeting of the Association for Computational Linguistics* (*Florence, Italy*) (Association for Computational Linguistics)
[11] Prucnal P R and Shastri B J 2017 *Neuromorphic Photonics* (Boca Raton, FL: CRC Press)
[12] Shen Y *et al* 2017 Deep learning with coherent nanophotonic circuits *Nat. Photon.* **11** 441–6
[13] Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *Proc. Natl Acad. Sci.* **79** 2554–8
[14] Tait A N, Nahmias M A, Shastri B J and Prucnal P R 2014 Broadcast and weight: an integrated network for scalable photonic spike processing *J. Lightwave Technol.* **32** 4029–41
[15] Tait A N, Wu A X, de Lima T F, Zhou E, Shastri B J, Nahmias M A and Prucnal P R 2016 Microring weight banks *IEEE J. Sel. Top. Quantum Electron.* **22** 312–25
[16] Huang C *et al* 2020 Demonstration of scalable microring weight bank control for large-scale photonic integrated circuits *APL Photonics* **5** 040803
[17] Feldmann J, Youngblood N, Wright C D, Bhaskaran H and Pernice W H 2019 All-optical spiking neurosynaptic networks with self-learning capabilities *Nature* **569** 208–14

[18] Pagiamtzis K and Sheikholeslami A 2006 Content-addressable memory (CAM) circuits and architectures: a tutorial and survey *IEEE J. Solid-State Circuits* **41** 712–27
[19] Lightwave Laboratory (Princeton University) lightlab 2018 *GitHub Repository* (https://github.com/lightwave-lab/lightlab/)
[20] Jayatilleka H, Murray K, Ángel Guillén-Torres M, Caverley M, Hu R, Jaeger N A F, Chrostowski L and Shekhar S 2015 Wavelength tuning and stabilization of microring-based filters using silicon in-resonator photoconductive heaters *Opt. Express* **23** 25084–97
[21] Tait A N, Jayatilleka H, Lima T F D, Ma P Y, Nahmias M A, Shastri B J, Shekhar S, Chrostowski L and Prucnal P R 2018 Feedback control for microring weight banks *Opt. Express* **26** 26422–43
[22] Krizhevsky A, Nair V and Hinton G 2010 CIFAR-10 Canadian Institute for Advanced Research p 5 (available at: www.cs.toronto.edu/~kriz/cifar.html)
[23] Marquez B A, Morison H, Guo Z, Filipovich M, Prucnal P R and Shastri B J 2020 Graphene-based photonic synapse for multi wavelength neural networks *MRS Adv.* **5** 1909–17
[24] Patel D, Ghosh S, Chagnon M, Samani A, Veerasubramanian V, Osman M and Plant D V 2015 Design, analysis and transmission system performance of a 41 GHz silicon photonic modulator *Opt. Express* **23** 14263–87
[25] Bangari V, Marquez B A, Miller H, Tait A N, Nahmias M A, de Lima T F, Peng H, Prucnal P R and Shastri B J 2020 Digital electronics and analog photonics for convolutional neural networks (DEAP-CNNs) *IEEE J. Sel. Top. Quantum Electron.* **26** 1–13