

# Nearest Unitary and Toeplitz matrix techniques for adaptation of Deep Learning models in photonic FPGA

Georgios Agrafiotis\*, Eftychia Makri, Ilias Kalamaras, Antonios Lalas, Konstantinos Votis, and Dimitrios Tzovaras

Centre for Research and Technology Hellas  
6th km Xarilaou - Thermi, Thessaloniki, 57001, Greece

## Abstract

Photonic circuits pave the way to extremely quick computation and real-time inference in critical applications, such as imaging flow cytometry (IFC). Nevertheless, current photonic FPGA implementations display intrinsic limitations that restrict the complexity of Deep Learning (DL) models that could be sustained. One of these restrictions implies the weight matrices to be unitary. Thus, machine learning mechanisms to transform weight matrices to their nearest unitary one, are essential for the effective deployment of such demanding tasks. Furthermore, DL models that perform convolutions, require special handling so as to fit in the photonic system. In this work, several methods have been investigated for conversion of non-unitary matrices to unitary ones, as well as, linear algebra techniques for the transformation of Convolutional Neural Networks (CNNs) to Feed-Forward models, under the prism of discovery of the best candidate for the photonic FPGA in terms of accuracy and restrictions. Experimental results proved that post-training or iterative techniques to find the nearest unitary weight matrix can be applied for photonic chips with the minimum loss in accuracy, while CNNs adapted well in a photonic configuration employing a Toeplitz matrix implementation. The proposed approach envisions efficient tackling of DL models limitations for deployment in photonic FPGAs.

\*Corresponding Author: gagrafio@iti.gr

This work was funded by the European's Union Horizon 2020 Research and Innovation Program through NEuromorphic Reconfigurable Integrated photonic Circuits as artificial image processor (NEoteRIC) under Grant Agreement No. 871330.

## 1 Introduction

Recent advances in artificial neural networks (ANNs) have shown enhanced performance in manipulation and recognition of images, videos, text, and audio, at a variety of applications. Despite the high accuracy and generalization ability of the produced models, their complexity often results in time-consuming inference, which prevents their adoption in time-critical applications. An example is imaging flow cytometry (IFC), where the demanding image processing performed on each cell passing through the flow channel is a significant bottleneck preventing high-throughput applications (>100K cells/sec) [10]. Deep Learning (DL) methods can provide advanced analysis per single cell, but at the cost of lower throughput.

This fact has led to intense investigations and implementation of hardware-based neuromorphic computing architectures, i.e., hardware platforms that can mimic human brain functions [7], [4]. Photonics can provide a proliferating platform for the development of ANNs due to inherent merits such as high wall-plug efficiency, parallel processing through time-wavelength and space multiplexing along with unparalleled operational bandwidth. Linear transformations can be performed in the optical domain by propagating light through a properly configured photonic structure, unlocking negligible power consumption per ANN operation [31]. On the other hand, photonic platforms are still plagued by high footprints and lack of circuit adaptability. With respect to the implementation of ANN architectures, photonic circuits place restrictions on the morphology of the networks that can be implemented [39].

In this paper, different algorithms are applied to DL models, namely Feed-Forward (FFNN) and Convolutional Neural Networks (CNNs), regarding the transformation of weight matrices to unitary ones, as well as a Toeplitz matrix implementation, in order to adapt those models in the hardware-restricted photonic FPGA. A discussion of the findings, their wider implications, and the potential for this work’s extension and experimental realization concludes our paper.

## 2 Related Work

Several implementations of ANNs on photonic circuits have been recently proposed, in order to exploit the high speed and inherent parallelism of optics [31]. It is important to note that optical deep neural networks and very fast processing photonic engines have both been proven in the context of reservoir computing [33]. Additionally, integrated weighting banks have recently been implemented using photonic integrated circuits (PICs), enabling the development of completely integrated photonic neural networks that use sinusoidal activation components [32]. Although neuromorphic technology can significantly enhance speed or energy consumption, it always has extra limits and constraints compared to ANNs that are installed on general-purpose hardware in the form of simulations [36]. Another constraint of photonic-based neuromorphic architectures is the ability to directly implement the activation functions that are usually employed in deep learning, e.g., ReLU [13], [17], [22]. To provide the functionality of non-linear activation functions, often a Mach-Zehnder Modulator (MZM) or Interferometer (MZI) [27] is employed to properly adjust an optical signal depending on a neuron’s output.

CNNs are a powerful category of ANNs and have been widely used in photonic implementations. In [3], a photonic integrated circuit architecture of a three layer CNN network was presented, capable of performing a million inferences per second while a similar idea is employed for realizing 11 tera operation per second (TOPS) photonic convolutional accelerator for optical neural network [35]. An energy efficient approach was proposed in [38], where deeply pipelined multi-FPGA architecture is presented along with an algorithm, in order to map the

CNN layers to multiple FPGA boards, achieving better throughput and latency than single-FPGA implementations. The goal for fully optical neuromorphic computing has been also examined in [9], where a novel photonics-based backpropagation accelerator for high performance deep learning training was implemented. This work, attempts to overcome the hardware restrictions imposed by a photonic FPGA, by transforming the weight matrices of neural networks to unitary weight matrices, with respect to the non-linearity idea of an MZI or MZM, while, moreover, attempting to adapt the convolution execution in a photonic platform by employing Toeplitz matrix implementations.

Our approach, in contrast to existing unitary learning techniques [2],[18] that incorporate other NN designs and demand calculations in the complex domain, exploits the advantages of FFNN or CNN while and in the same time transforms those models in quantum computing [21] algorithms form with both nearest unitary matrices techniques and Toeplitz matrix implementations. Our method has the benefit of being less computationally complex and achieving minimum loss in accuracy. It should be mentioned that our iterative training scheme is a variation of projected gradient descent [34]. Last but not least, this method could be expanded to a great variety of hardware morphology photonic chips, because of its generalization in terms of weight matrices form.

## 3 Fitting DL in Photonics

PICs aim at configuring arbitrary designs by employing a large-scale set of programmable integrated beam splitters and phase actuators. Adjustable circuits with high complexity in terms of programming, demand the integration of many Tunable Basic Units (TBUs) and the optimization of their performance that can be affected by a series of factors namely minimum number of parameters, optical loss and power consumption. As mentioned by Lopez et al. in [19], a simple approach for a TBU is using a “balanced MZI with an independent phase actuator on each arm”.

The minimal matrix multiplication may be implemented using MZI, a fundamental minimum matrix operation unit that can be built on a silicon substrate. Without experiencing any significant

loss, the photonic matrix network created by MZIs may be expanded to accommodate any matrix multiplication, although regarding the Quantum Computing nature of the photonic FPGAs, those matrices must be unitary [26].

The triangular decomposition algorithm was a generic approach initially suggested in [28], as a practical realization of any  $n \times n$  unitary matrix. In this scenario, a structure made up of mirrors, phase shifters, and beam splitters, placed in a precise order can produce unitary matrix transformations [8]. Since basic optical devices like multimode interferometers (MMIs) and phase shifters (PSs) necessarily encourage analog unitary transformations in the electromagnetic waves, provided that the insertion loss of such devices is negligible [20], the different ways to modify weight matrices to be unitary are of great significance.

Furthermore, when considering the adaptation of deep learning models in photonic FPGAs, CNNs are the first option when it comes to image classification tasks. CNNs could be implemented on photonic chips, with a back-propagation algorithm, as already mentioned in Section 2. Regarding the complexity and resources of a back-propagation algorithm, as shown in [11], our goal is to develop a CNN model that could be transformed to a FFNN model, a form which is more wieldy for a photonic configuration, because of its simple linear algebra executions, that take part in the Fully-Connected (FC) layer, executions that could be implemented on photonic chip [19].

As a result, each application’s requirements must be carefully taken into account while designing the training algorithms ensuring that a) the trained network behaves correctly and stays within the imposed hardware limits (e.g. unitary weight matrices) and b) the transfer functions of the various components are appropriately modeled [24].

## 4 Proposed Methods

The deployment of a neural network directly on PIC introduces some constraints as explained in [19]. Fundamentally, only linear transformations can be applied. Moreover, the bias term found in a conventional fully connected neural network cannot be used ( $Y = Wx + b$ ), [14]. Additionally, the non-linear activation function should be consistent with

a photonic structure and the weight matrix ( $W$ ) must be unitary. The function  $|x|^2$ , which emulates the output of a photonic diode can be selected, as described in [33]. Finally, since the linear operations involve unitary matrices, only square matrices can be used with dimensions being enforced by the input data.

Consequently, each photonic neural network structure must resemble a fully connected neural network with bias terms set to zero, unitary weight matrices, and a non-linear activation function that is specific to the hardware. We aim to emulate a photonic ANN by using open source machine learning tools such as Tensorflow [1] and Pytorch [25] to create neural network models based on the aforementioned hardware limitations that may then be directly implemented on FPGAs.

### 4.1 Nearest Unitary Learning

Finding the closest unitary weight matrix is an ad hoc way to implement neural network designs on the PIC. The matrix  $U$  that develops in the decomposition of  $M$  into the product of an orthogonal matrix and a positive definite matrix is the nearest orthogonal matrix to a specific nonsingular matrix  $M$ , as explained in [15].

$$U = M(M^T M)^{-1/2} \quad (1)$$

Since the weight matrices of a neural network only include real values, the nearest orthogonal is computed as opposed to the nearest unitary.

While there is an error introduced by this method, the computational resources are significantly reduced: only one matrix is used per network layer, as opposed to three. Details about the three matrix implementation by performing Singular Value Decomposition (SVD) can be found in [31]. The SVD approach requires the implementation of two unitary matrices through FPGAs and a diagonal matrix using an optical amplifier or attenuator. It should be noted that the SVD method introduces no error in contrast with the Nearest Unitary method.

#### 4.1.1 Post-training Method

A straightforward method to derive orthogonal weight matrices in a multilayer perceptron is to first train the network and then apply the nearest

orthogonal approach to each matrix. The disadvantage of this approach is that the inaccuracy resulting from each weight matrix stated in Section 4 is amplified by passing through the rest of the network, notably the non-linearities. An approach to lessen the error’s amplitude is described below.

Assuming we have a N-layer perceptron, we will train the network N times, applying the closest orthogonal approach at the end of each training phase to the corresponding layer, which denotes that the training phase and the layer number should coincide. The weights of each layer that the closest orthogonal approach is applied to are frozen at every subsequent training session. It is possible to reduce the impact of each layer’s error by training the network in this staggered fashion.

#### 4.1.2 Iterative Method

An additional approach of achieving a better fit for the closest orthogonal weight matrices, is to apply the Nearest Orthogonal formula discussed in Section 4.1 iteratively. An N-layer perceptron is considered undergoing the default train session. At the end of each epoch, after the backpropagation is employed to the network weights, the closest unitary approach will be applied at each weight matrix. Since there should always be an optimal orthogonal solution, by maintaining a low learning rate, this strategy is driving the network towards this direction as training progresses. By using this method, we expect the weight matrices to converge to an orthogonal solution that is close to a local optimal.

Weight initialization has been shown to enhance the network training process, [16] and [23]. We suggest a weight initialization strategy that may improve our proposed method. In more detail, the network will go through N training iterations, and only the associated layer will be changed during each training phase. At each training step the corresponding layer weights will be updated by applying nearest orthogonal approach at every epoch as described above and by the end of each training session the associated weight matrix will have orthogonal properties. All previous layers will be frozen during a particular training cycle to keep their orthogonal characteristics. The weight matrices will all be orthogonal at the conclusion of all training sessions, serving as weight initialization before the

network is retrained using the method outlined at the beginning of this section.

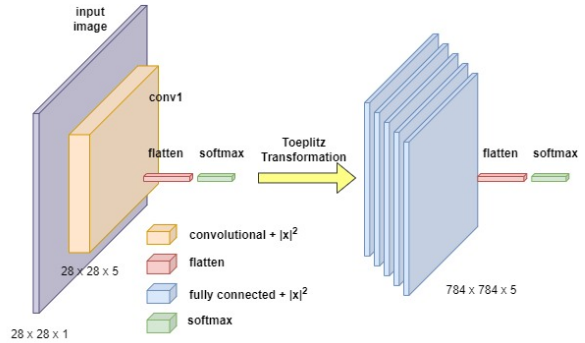


Figure 1: Schematic process of CNN to FC with Toeplitz implementation

## 4.2 Handling CNN

A common architecture in classical neural networks is the convolutional network, as already mentioned. CNNs are particularly well-suited for computer image recognition problems, because they illustrate a straightforward but significant observation: since detecting an object is a mainly separate task of where the object appears in an image, the network should be equivariant to translations [37]. Consequently, the linear transformation in a CNN is not fully-connected, while, according to [19] and the aforementioned photonic restrictions in an MZI implementation, convolution could be achieved, if transformed to matrix multiplication followed by the non-linear activation function. Thus, instead of using for-loops to perform 2D convolution on images we can convert the filter to a Toeplitz matrix and an image to a vector and do the convolution just by one matrix multiplication.

In order to implement the aforementioned process, we were based on the repository of Ali Salehi [29] combined with the proper modifications on the padding parameter to achieve "same" padding. By applying "same" as padding method to the CNN model, we are forcing the Toeplitz matrix to be square. This is a necessary contribution because a) as mentioned in Section 4, quantum computing (e.g. photonic FPGAs) is only able in handling square matrices and b) the methods referred in Section 4.1 are only applicable in square matrix forms,

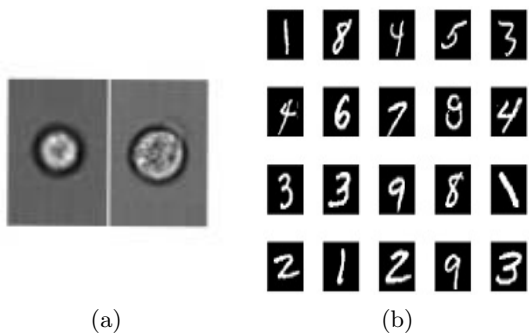


Figure 2: (a) Representative images of G1 and G2 phase, (b) MNIST handwritten digits example

since unitary matrices are by definition square.

It is worth noted, that the Toeplitz nature of the  $N \times N$  matrix  $P$  gives us only  $N$  degrees of freedom, as opposed to  $N^2$  degrees of freedom encoded in the weights of a fully-connected deep neural network, as mentioned in [6]. Nevertheless, Toeplitz implementation, allows photonic FPGAs to handle CNNs, with the less possible complexity, without exploiting back-propagation algorithms.

## 5 Applications

### 5.1 Cell image dataset

Aiming to test the efficiency of the models in a cytometry relevant scenario, a single-cell image classification application was put into practice. The dataset [12] consists of raw IFC images of 7.362 asynchronously growing immortalized human T-lymphocyte cells (Jurkat cells) which can be classified into two different stages of cell cycle, as depicted in Figure 2: the G1 phase, where the cell grows physically and increases the volume of both protein and organelles, and the G2 phase, which involves further cell growth and organization of cellular contents. Being able to distinguish between the “small” G1 cells and the “big” G2 cells is important as it can be extended to distinguishing between normal or cancerous cells, since cancerous cells tend to be larger. The size-based classification strategy was based on the work of B. Shashni et al. in [30].

### 5.2 Single-cell image identification

For the identification of the single-cell images, a comparison to known benchmarks is required. One popular dataset used for this purpose is the MNIST collection of handwritten numbers [5]. For the classification processes, two experiments were conducted, concerning the methods mentioned in Section 4. In every experiment, a  $|x|^2$  mathematic relation was employed as an activation function, in order to satisfy the photodiode necessity as discussed in Section 4. The deep learning models applied to the datasets were the following: (i) A single 2025x2025-neurons Fully Connected (FC) layer regarding the Jurkat and a single 784x784-neurons FC for the MNIST dataset, (ii) A two layer FC, consisting of 2025x2025-neurons each for the Jurkat and a two layer FC, consisting of 784x784-neurons for the MNIST, and (iii) A CNN. The cross entropy loss, together with the softmax activation function, is used for training the networks.

## 6 Results

The models mentioned in Section 5.2 were used in three studies, which all targeted the MNIST and Jurkat datasets. The performance of the single-layered network is assessed in two scenarios in the first experiment. In the first situation, the post-training approach described in Section 4.1.1 is used, while in the second, the iterative method described in Section 4.1.2 is employed. The experiment’s findings are collected in Table 1.

Subsequently, the second experiment tests the two-layered network in four scenarios. The post training strategy suggested in Section 4.1.1 is used in the first case immediately following the completion of a training iteration, and the second scenario uses the same method by completing two training phases while freezing the first layer in the second phase as indicated. Moreover, the two remaining scenarios are utilizing the iterative method discussed in Section 4.1.2. Specifically, the first situation updates both weight matrices concurrently as described. Finally, the last training scenario utilizes the weight initialization strategy described in Section 4.1.2. The results of the experiment are presented in Table 2.

The last experiment involves conventional train-

ing of the CNN described in Section 5.2 followed by its transformation into a matrix multiplication followed by a non-linear activation function as discussed in Section 4.2. Furthermore, since the CNN contains five filters, five Toeplitz matrices will be generated and the output will be concatenated right before softmax layer. The performance of the network is then compared before and after using the post-training nearest orthogonal technique. It should be emphasized that since retraining the network would behave as a fully connected implementation rather than a convolutional approach, the iterative method cannot be combined with the Toeplitz transformation. The experiment’s outcomes are shown in Table 3.

Table 1 demonstrates that there is some accuracy loss when using the post-training method in comparison to the initial model and the iterative approach produces superior outcomes. The post training approach also has a flaw that was discovered via experimentation: weight initialization has a significant impact on the model’s output. This is a natural conclusion given that each weight initialization may produce distinct weight matrices, each of which has a large closest orthogonal distance and a large inaccuracy. The iterative method, on the other hand, is unaffected by this issue, since it compels the weight matrices to continually be close to a unitary, by using a low learning rate in conjunction with the updating strategy already explained.

Table 2 illustrates that due to the large inaccuracy indicated in Section 4.1.1, the post-training cannot be implemented in a network with more than one layer without employing the freezing approach. Additionally, the iterative technique once more outperforms post-training. Moreover, the weight initialization discussed in Section 4.1.2 can improve network performance.

Table 3 depicts that, as anticipated, the CNN implementation performs better than the Fully Connected. Additionally, employing the post-training Nearest Unitary technique on the CNN produces better results than employing it on a Fully Connected network, suggesting that certain CNN properties may still exist. Finally, Toeplitz transformation can be combined with the SVD technique stated in Section 4.1 if the desired property is model accuracy, though this will require more processing power as explained.

Table 1: Image classification results for single-layer FFNN with different nearest unitary techniques

Method	MNIST		Jurkat	
	Accuracy	Precision	Accuracy	Precision
<i>Initial Model</i>	0.982	0.982	0.947	0.922
<i>Post-training</i>	0.88	0.86	0.939	0.903
<i>Iterative</i>	0.976	0.976	0.941	0.911

Table 2: Image classification results for multi-layer FFNN with different nearest unitary techniques

Method	MNIST		Jurkat	
	Accuracy	Precision	Accuracy	Precision
<i>Initial Model</i>	0.984	0.984	0.95	0.941
<i>Post – training/No Freeze</i>	0.801	0.789	0.769	0.732
<i>Post – training/Freeze</i>	0.904	0.899	0.9397	0.901
<i>Iterative</i>	0.979	0.979	0.942	0.911
<i>Iterative-weights init.</i>	0.981	0.981	0.949	0.942

## 7 Conclusion

When deploying DL models, photonic hardware has the potential to significantly outperform and use less energy than traditional hardware. Additionally, the operations of PICs are limited to linear operations employing unitary matrices. In order to create fully connected neural networks that are in compliance with the aforementioned criteria, we provide a number of ways of utilizing the closest unitary approach. The recommended techniques produce excellent results while using 66% less hardware than other suggested implementations. Additionally, we offer a technique for realizing CNN on PIC by converting the convolution process to matrix multiplication exhibiting high performance.

Fruitful directions for future work include further investigation on DL models implementation on FPGAs such as Recurrent Neural Network and Transformers, as well as optimization techniques for hardware requirements. The proposed approaches are deemed promising for accelerating the employment of neuromorphic computing via the extensive utilisation of DL-based PICs in various applications.

Table 3: Image classification results for CNN with post-training nearest unitary technique

Method	MNIST		Jurkat	
	Accuracy	Precision	Accuracy	Precision
<i>Initial Model</i>	0.984	0.984	0.958	0.948
<i>Post – Training</i>	0.901	0.897	0.944	0.908

## References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [2] M. Arjovsky, A. Shah, and Y. Bengio. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pages 1120–1128. PMLR, 2016. doi: 10.48550/arXiv.1511.06464.
- [3] H. Bagherian, S. Skirlo, Y. Shen, H. Meng, V. Ceperic, and M. Soljacic. On-chip optical convolutional neural networks. *arXiv preprint arXiv:1808.03303*, 2018. doi: 10.48550/arXiv.1808.03303.
- [4] B. Bai, H. Shu, X. Wang, and W. Zou. Towards silicon photonic neural networks for artificial intelligence. *Science China Information Sciences*, 63(6):1–14, 2020. doi: 10.1007/s11432-020-2872-3.
- [5] A. Baldominos, Y. Saez, and P. Isasi. A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9(15):3169, 2019. doi: 10.3390/app9153169.
- [6] J. R. Basani, M. Heuck, D. R. Englund, and S. Krastanov. All-photonic artificial neural network processor via non-linear optics. *arXiv preprint arXiv:2205.08608*, 2022. doi: doi.org/10.48550/arXiv.2205.08608.
- [7] D. Brunner, B. Penkovsky, B. A. Marquez, M. Jacquot, I. Fischer, and L. Larger. Tutorial: Photonic neural networks in delay systems. *Journal of Applied Physics*, 124(15):152004, 2018. doi: 10.1063/1.5042342.
- [8] J. Cheng, H. Zhou, and J. Dong. Photonic matrix computing: from fundamentals to applications. *Nanomaterials*, 11(7):1683, 2021. doi: 10.3390/nano11071683.
- [9] D. Dang, S. V. R. Chittamuru, S. Pasricha, R. N. Mahapatra, and D. Sahoo. Bplight-cnn: A photonics-based backpropagation accelerator for deep learning. *CoRR*, abs/2102.10140, 2021. doi: 10.1145/3446212. URL <https://arxiv.org/abs/2102.10140>.
- [10] M. Doan, I. Vorobjev, P. Rees, A. Filby, O. Wolkenhauer, A. E. Goldfeld, J. Lieberman, N. Barteneva, A. E. Carpenter, and H. Hennig. Diagnostic potential of imaging flow cytometry. *Trends in biotechnology*, 36(7):649–652, 2018. doi: 10.1016/2017.12.008.
- [11] S. K. Esser, R. Appuswamy, P. Merolla, J. V. Arthur, and D. S. Modha. Backpropagation for energy-efficient neuromorphic computing. *Advances in neural information processing systems*, 28, 2015.
- [12] P. Eulenberger, N. Köhler, T. Blasi, A. Filby, A. E. Carpenter, P. Rees, F. J. Theis, and F. A. Wolf. Reconstructing cell cycle and disease progression using deep learning. *Nature communications*, 8(1):1–6, 2017. doi: 10.1038/s41467-017-00623-3.
- [13] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.
- [14] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.
- [15] B. K. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135, 1988. doi: 10.1364/JOSAA.5.001127.
- [16] S. K. Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017. doi: 10.48550/arXiv.1704.08863.

- [17] H. Li, B. Wu, W. Tong, J. Dong, and X. Zhang. All-optical nonlinear activation function based on germanium silicon hybrid asymmetric coupler. *IEEE Journal of Selected Topics in Quantum Electronics*, 2022. doi: 10.1109/JSTQE.2022.3166510.
- [18] S. Lloyd and R. Maity. Efficient implementation of unitary transformations. *arXiv preprint arXiv:1901.03431*, 2019. doi: 10.48550/arXiv.1901.03431.
- [19] D. P. López. Programmable integrated silicon photonics waveguide meshes: optimized designs and control algorithms. *IEEE Journal of Selected Topics in Quantum Electronics*, 26(2):1–12, 2019. doi: 10.1109/JSTQE.2019.2948048.
- [20] A. Macho-Ortiz, D. Pérez-López, and J. Capmany. Optical implementation of  $2 \times 2$  universal unitary matrix transformations. *Laser & Photonics Reviews*, 15(7):2000473, 2021. doi: 10.1002/lpor.202000473.
- [21] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, et al. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, 2022. doi: 10.1038/s41586-022-04725-x.
- [22] M. Miscuglio, A. Mehrabian, Z. Hu, S. I. Azam, J. George, A. V. Kildishev, M. Pelton, and V. J. Sorger. All-optical nonlinear activation function for photonic neural networks. *Optical Materials Express*, 8(12):3851–3863, 2018. doi: 10.1364/OME.8.003851.
- [23] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1):291–322, 2022. doi: 10.1007/s10462-021-10033-z.
- [24] N. Passalis, G. Mourgias-Alexandris, A. Tsakyridis, N. Pleros, and A. Tefas. Training deep photonic convolutional neural networks with sinusoidal activations. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(3):384–393, 2019. doi: 10.1109/TETCI.2019.2923001.
- [25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimselshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. doi: 10.48550/arXiv.1912.01703.
- [26] F. Pavosevic and J. Flick. Polaritonic unitary coupled cluster for quantum computations. *The Journal of Physical Chemistry Letters*, 12(37):9100–9107, 2021. doi: 10.1021/acs.jpcclett.1c02659.
- [27] S. Pitris, C. Mitsolidou, T. Alexoudi, D. Pérez-Galacho, L. Vivien, C. Baudot, P. De Heyn, J. Van Campenhout, D. Marris-Morini, and N. Pleros. O-band energy-efficient broadcast-friendly interconnection scheme with siphon mach-zehnder modulator (mzm) & arrayed waveguide grating router (awgr). In *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pages 1–3. IEEE, 2018.
- [28] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani. Experimental realization of any discrete unitary operator. *Physical review letters*, 73(1):58, 1994. doi: 10.1103/PhysRevLett.73.58.
- [29] A. Salehi. My Research Software. 2022. doi: 10.5281/zenodo.1234. URL [https://github.com/alisaalehi/convolution\\_as\\_multiplication](https://github.com/alisaalehi/convolution_as_multiplication).
- [30] B. Shashni, S. Ariyasu, R. Takeda, T. Suzuki, S. Shiina, K. Akimoto, T. Maeda, N. Aikawa, R. Abe, T. Osaki, et al. Size-based differentiation of cancer and normal cells by a particle size analyzer assisted by a cell-recognition pc software. *Biological and Pharmaceutical Bulletin*, 41(4):487–503, 2018. doi: 10.1248/bpb.b17-00776.
- [31] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun,



- S. Zhao, H. Larochelle, D. Englund, et al. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441–446, 2017. doi: 10.1038/nphoton.2017.93.
- [32] A. N. Tait, T. F. De Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal. Neuromorphic photonic networks using silicon photonic weight banks. *Scientific reports*, 7(1):1–10, 2017. doi: doi.org/10.1038/s41598-017-07754-z.
- [33] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature communications*, 5(1):1–6, 2014. doi: 10.1038/ncomms4541.
- [34] K. Wood, G. Bianchin, and E. Dall’Anese. Online projected gradient descent for stochastic optimization with decision-dependent distributions. *IEEE Control Systems Letters*, 6: 1646–1651, 2021. doi: 10.1109/LCSYS.2021.3124187.
- [35] X. Xu, M. Tan, B. Corcoran, J. Wu, A. Boes, T. G. Nguyen, S. T. Chu, B. E. Little, D. G. Hicks, R. Morandotti, et al. 11 tops photonic convolutional accelerator for optical neural networks. *Nature*, 589(7840):44–51, 2021. doi: 10.1038/s41586-020-03063-0.
- [36] C. Yakopcic, R. Hasan, and T. M. Taha. Memristor based neuromorphic circuit for ex-situ training of multi-layer neural network algorithms. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2015. doi: 10.1109/IJCNN.2015.7280813.
- [37] G.-Y. Yang, X.-L. Li, R. R. Martin, and S.-M. Hu. Sampling equivariant self-attention networks for object detection in aerial images. *arXiv preprint arXiv:2111.03420*, 2021. doi: 10.48550/arXiv.2111.03420.
- [38] C. Zhang, D. Wu, J. Sun, G. Sun, G. Luo, and J. Cong. Energy-efficient cnn implementation on a deeply pipelined fpga cluster:. pages 326–331, 08 2016. doi: 10.1145/2934583.2934644.
- [39] J. Zhu and P. Sutton. Fpga implementations of neural networks—a survey of a decade of progress. In *International conference on field programmable logic and applications*, pages 1062–1066. Springer, 2003. doi: 10.1007/978-3-540-45234-8\_120.