TRAINING NOISE-RESILIENT RECURRENT PHOTONIC NETWORKS FOR FINANCIAL TIME SERIES ANALYSIS

N. Passalis[1], M. Kirtas[1], G. Mourgias-Alexandris[2], G. Dabos[2], N. Pleros[2] and A. Tefas[1]
[1]*Artificial Intelligence and Information Analysis Lab*
[2]*Wireless and Photonic Systems and Networks Group*
Dept. of Informatics, *Aristotle University of Thessaloniki*, Thessaloniki, Greece
{passalis, eakirtas, mourgias, ntamposg, npleros, tefas}@csd.auth.gr

***Abstract*—Photonic-based neuromorphic hardware holds the credentials for providing fast and energy efficient implementations of computationally complex Deep Learning (DL) models. At the same time, the unique nature of neuromorphic photonics also imposes a number of limitations that hinders its application, including the need to re-train DL models in order to be compliant with the underlying hardware architecture, as well as the existence of various noise sources, which are prevalent in virtually all neuromorphic photonic architectures and negatively affect the accuracy of the deployed models. In this paper we propose a novel noise-aware approach for training neural networks realized on photonic hardware, which can alleviate some of these limitations. To this end we first provide an extensive characterization of the various noise sources that affect sigmoid-based recurrent photonic architectures, as well as provide an extensive study on the effect of various signal-to-noise-ratios (SNRs) levels on the performance of such DL models. The effectiveness of the proposed method is demonstrated on a challenging forecasting problem that involves high frequency financial time series using a state-of-the-art recurrent photonic architecture, which naturally fits the requirements of such latency-critical applications. Apart from providing more accurate models, the proposed method opens several interesting future research directions on co-designing neuromorphic photonics, including developing DL models that can work on lower SNRs, leading to more energy efficient solutions.**

***Index Terms*—Photonic Deep Learning, Neural Network Initialization, Noise-aware Training**

I. Introduction

Deep Learning (DL) allows for obtaining state-of-the-art performance on a wide range of different problems [1]. Despite being very successful on tackling such challenging tasks, DL models are hindered by their high complexity, since they are typically trained and deployed using powerful hardware, which increases the cost of adopting DL, as well as restricts its potential applications [2], [3]. This, in turn, fueled the development of specialized hardware accelerators that are capable of accelerating DL, leading to faster training and inference and reducing energy and power requirements. Graphics Processing Units (GPUs) [4] are among the predominantly used accelerators employed to this end, due to their low cost and ability to accelerate matrix-based calculations, which cover a significant fraction of the calculations involved during the training and inference of DL models. At the same time, many other hardware accelerators have emerged, ranging from Tensor Processing Units (TPUs) [5], to advanced neuromorphic hardware platforms [6], [7].

A very promising and increasingly popular research area is the use of *photonic* hardware to provide such neuromorphic architectures [8]. In neuromorphic photonics signals are encoded using light, instead of using electrical quantities, which are then manipulated to provide the functionality of neurons. Several approaches have been proposed to this end, ranging from using purely optical components [9], [10], to advanced combinations of electro-optical devices [11]. Neuromorphic photonics provides several advantages over traditional electronic accelerators, since they are capable of operating at very high frequencies, signals are propagated near to the speed of light, while they also provide a massive parallelism potential due to their enormous bandwidth.

The unique nature of neuromorphic photonics also imposes a number of limitations that hinders their applications when DL models are used. First, the vast majority of currently available photonic architectures are facing challenges to support the activation functions that are traditionally used in DL, such as ReLU [12], with most of them relying either on sinusoidal activations [13], or sigmoidal activations [10]. This in turn requires training the DL models specifically for neuromorphic photonic architectures, taking into account the transfer functions of the employed components [13]. Furthermore, while photonic hardware implementations of neurons leveraging advances in materials and waveguide technologies [14], [15] enable ultra-fast analog processing and vector-matrix multiplications with almost zero power consumption [8], these implementations are susceptible to a number of different noise sources that typically do not exist on software-based DL implementations. For example, state-of-the-art Digital-to-Analog (DAC) circuits employed in photonic neurons constitute a significant noise source to the input of

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871391 (PlasmoniAC). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

DL models. Besides the inherent noise characteristics of the employed DACs, thermal crosstalk being present almost in all photonic weighting architectures can further increase the imposed noise [11], [16]. Finally, additional noise stemming from optical activation functions is also contributing to the signal quality deterioration [17], as explained in more detail in Section II.

Despite noise being prevalent in virtually all neuromorphic photonic architectures, most of the existing works are limited on evaluating the effect of noise during the deployment/inference and after the network has already been trained. In fact, noise is usually treated as a characteristic that concerns only the hardware and for which appropriate hardware architectures must be devised in order to achieve the required SNR that ensures that DL models will operate as intended. However, this approach overlooks the fact that DL models are *intrinsically* resistant to noise, especially when they are first appropriately trained to withstand noisy signals. Indeed, preliminary results reported in [13], demonstrated that it is possible to derive models that can withstand such noise, especially when the noise sources are appropriately modeled and used during the training process. It is worth noting that following such *co-design* approach, i.e., training the models taking into account the actual limitations of the hardware, can also lead to other benefits as well. For example, having DL models that are capable of correctly operating using lower SNRs can also allow for reducing the power requirements, leading to more energy efficient architectures.

In this paper we propose a novel noise-aware approach for training neural networks implemented with photonic hardware. The contributions of this work are three-fold: a) we provide an extensive characterization of the various noise sources that affect sigmoidal-based recurrent photonic architectures, b) we propose a training method that goes beyond traditional noise-aware training approaches, by also taking into account the initialization of photonic DL models, which can significantly improve their performance, and c) we provide an extensive study on the effect of various noise levels (SNRs) to the performance of the network. To the best of our knowledge, the proposed method is the first that is capable of appropriately initializing the network by taking into account the noise sources that exist in the actual photonic implementation. Note that DL involves non-convex optimization problems that almost always end up on local minima [18]. Therefore, an improved initial point for the learning process can have a significant effect on the optimization process. The effectiveness of the proposed method is demonstrated on a challenging forecasting problem that involves high frequency financial time series data [19], using a state-of-the-art recurrent photonic architecture, which naturally fits the requirements of such latency-critical applications.

The rest of the paper is structured as follows. First, the sigmoid-based recurrent photonic architecture is introduced in Section II. Then, the proposed method is described in Section III, while the experimental evaluation is provided in Section IV. Finally, conclusions are drawn in Section V.

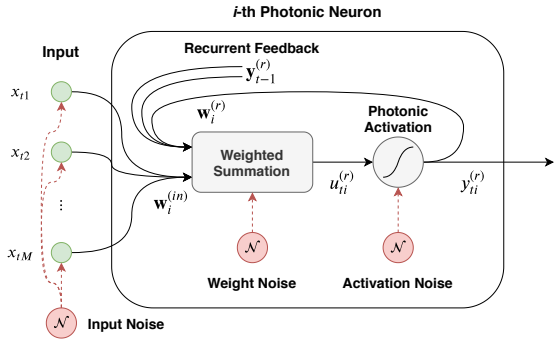

Fig. 1: Photonic Recurrent Neuron Architecture

II. Sigmoid-based Recurrent Photonic Neurons

The employed recurrent photonic architecture is composed of multiple recurrent photonic neurons. The architecture of each recurrent photonic neuron is shown in Fig. 1. Let $\mathbf{x}$ be a multi-dimensional time series, while let $\mathbf{x}_t \in \mathbb{R}^M$ denote M observations fed to the neuron at the t-th time-step. Then, the input signal is weighted by the i-th neuron using the input weights $\mathbf{w}_i^{(in)} \in \mathbb{R}^M$. Furthermore, the recurrent feedback signal, denoted by $\mathbf{y}_{t-1}^{(r)} \in \mathbb{R}^N$, which corresponds to the output of the N recurrent neurons at the previous time-step, is also weighted by the recurrent weights $\mathbf{w}_i^{(r)} \in \mathbb{R}^N$. The final weighted output of the i-th recurrent neuron is then calculated as:

$$u_{ti}^{(r)} = \mathbf{w}_i^{(in)^T} \mathbf{x}_t + \mathbf{w}_i^{(r)^T} \mathbf{y}_{t-1}^{(r)}. \quad (1)$$

Note that we omitted the bias term to simplify the employed notation. Then, this weighted output is fed to the employed photonic non-linearity $f(\cdot)$ to acquire the final activation of the neuron as $y_{ti}^{(r)} = f(u_{ti}^{(r)})$.

This architecture does not employ gating mechanisms, which are typically present in many established recurrent architectures, such as LSTMs [20] and GRUs [21]. This design choice allows to directly use existing feed-forward photonic neuron designs [10], [22] to implement the employed recurrent model simply by adding a feedback loop, while keeping the complexity of photonic hardware low compared to the already demonstrated feed-forward layouts. More specifically, the input signal is encoded using ultra-fast DACs that can produce signals up to 100GBaud. Then, the weighted summation can be easily implemented using any of the already demonstrated weighting techniques, such as employing WDM layouts based on micro-ring resonators [11], Mach-Zehnder-based coherent weighting [23], or even using phase-change materials [24]. For the purpose of this study, we assume that a coherent weighting scheme based on a Mach-Zehnder interferometer is used. However, note that this is without loss of generality, since similar analysis can be conducted for the other two cases, after appropriately modeling the prevailing noise sources during the signal weighting and summation. Finally, the photonic activation function is implemented using the layout proposed in [10], where two cascaded wavelength converters are realized by a saturated differentially-

biased SOA-MZI followed by an SOA that performs cross-gain modulation on its small-signal gain regime. The transfer function of this photonic activation is approximated by fitting a generalized logistic function to the experimental behavior of the proposed scheme as $f(x) = A_2 + \frac{A_1 - A_2}{1+e^{((x-x_0)/d)}}$, where $A_1 = 0.060$, $A_2 = 1.005$, $x_0 = 0.145$, and $d = 0.033$. The exact values of fitted function were calculated based on the experimental observations reported in [10].

Following the brief description of possible noise sources in a photonic neuron, we elucidate their origin by mapping the underpinning photonic devices used to deliver necessary functions of the neuron. First, noise on data encoding is often attributed to the intrinsic noise of the employed DACs, modulators and lasers. Second, crosstalk induced noise being prevalent in weighting elements originates from the employed control circuits and usually gets increased by the limited resolution of the used DACs. Finally, noise owing to photonic activation functions [10], [17], [25], e.g., gain fluctuation will further increase the noise level. This kind of cumulative noise has been thoroughly studied in optical communication links pointing out that all the above-mentioned different source of noises can be modeled as a channel with an Additive White Gaussian Noise (AWGN) [26], [27]. Therefore, the BER in such optical links can be measured precisely by simply calculating the SNR of the system considering AWGN and hence the noise in photonic neurons can modeled by stochastically corrupting the signal by drawing the corruption values from a Gaussian distribution with zero mean and variance that corresponds to the energy of the noise. The operation of the neuron is modeled as:

$$u_{ti}^{(r)} = \sum_{j=1}^{M} w_{ij}^{(in)} \left(x_{tj} + \mathcal{N}(0, \sigma_i^2)\right) + \mathbf{w}_i^{(r)^T} \mathbf{y}_{t-1}^{(r)} + \mathcal{N}(0, \sigma_w^2), \tag{2}$$

where σ_i^2 is the variance of the noise that affects the input and σ_w^2 is the variance of the noise that affects the weighting process. Also, the final output of the neuron is modeled as:

$$y_{ti}^{(r)} = f(u_{ti}^{(r)}) + \mathcal{N}(0, \sigma_a^2), \tag{3}$$

where σ_a^2 is the variance of the noise induced by the activation function. Finally, note that fully connected layers are also used in the employed neural network architectures. For these layers we used the same photonic architecture (after removing the recurrent optical feedback loops), activation function and noise models.

III. Noise-aware Adaptive Initialization

The response of the i-th layer of the employed architecture is denoted by $\mathbf{y}_t^{(i)} \in \mathbb{R}^{m^{(i)}}$, where the notation $m^{(i)}$ is used to refer to the number of neurons of the corresponding layer. Note that this definition covers both recurrent and feed-forward layers. Furthermore, we use the notation $\mathbf{y}_t^{(0)} = \mathbf{x}_t$ to refer to the input of the model, while the response of the neurons is calculated as:

$$\mathbf{u}_t^{(i)} = \mathbf{W}_i \mathbf{y}_t^{(i-1)} + \mathbf{b}_i, \tag{4}$$

where the notation $\mathbf{W}_i \in \mathbb{R}^{m^{(i)} \times m^{(i-1)}}$ is used to refer to the weights of the layer, while $\mathbf{b}_i \in \mathbb{R}^{m^{(i)}}$ denotes the corresponding biases. This definition covers also the recurrent architecture described in (1), since the response vector can be extended to include the recurrent feedback, while the weights can be similarly extended to include both the input and recurrent weights. The final output of the layer is calculated by feeding the weighted responses to the employed activation function, i.e., $\mathbf{y}_t^{(i)} = f(\mathbf{u}_t^{(i)})$. To simulate the effect of noise during the training process, we can simply add the corresponding noise sources in the computational graph of the model, as described in (2) and (3). Note that the SNR is controlled by changing the variance of the corresponding noise source, i.e., σ_i^2 for the input, σ_w^2 for the weighting processing, and σ_a^2 for the activation function.

Existing initialization methods usually employ variance preserving approaches, that ensure that the activation variance will be maintained across the different layers of the network. These approaches are activation-specific, i.e., different initialization schemes are derived for different activation functions [12], [13], [28], and ignore the existence of noise sources, as well as their effects on the model. In this paper, we argue that the initialization of the network must also take into account the noise that might exist in the network architecture, instead of just modeling the effect of the activation function on the variance of the activations. To this end, we derive an activation-agnostic initialization scheme that is data-driven and takes into account the actual distribution of the activations, instead of just the theoretical one [12], [13], [28] and/or just relying on an ideal (noise-less) model of the network [29]. In this way, the proposed method is capable of modeling the effect of noise on the various layers of the network in order to better estimate the most appropriate initialization scheme to be applied.

More specifically, we assume that the weights are initialized by drawing from a Gaussian distribution with zero mean and variance σ_i^2, denoted by $\mathcal{N}(0, \sigma_i^2)$. The proposed method still aims to estimate the optimal value for the initialization variance of each layer σ_i^2. However, instead of focusing only on a few properties of the network, such as the size of each layer and the employed activation function [12], [28], we propose a data-driven approach that exploits the effectiveness of training shallow neural networks (up to 2 layers) to incrementally estimate the most appropriate value for the variance. To this end, we introduce an additional scaling factor α_i for each layer:

$$\mathbf{y}_t^{(i)} = f(|\alpha_i| \mathbf{W}_i \mathbf{y}_{t-1}^{(i-1)} + \mathbf{b}_i), \tag{5}$$

where $|\cdot|$ denotes the absolute value operator. Altering the value for this scaling factor is equivalent to altering the initialization variance of the corresponding layer, i.e., it is equivalent to initializing the layer by drawing from $\mathcal{N}\left(0, (\alpha_i \sigma_i)^2\right)$. Then, an auxiliary classification layer is trained directly on the representation extracted from a specific layer. The weights of this layer are denoted by $\mathbf{W}_i^{class} \in \mathbb{R}^{m^{(i)} \times N_C}$, where N_C is the number of classes (for classification problems) or the

number of values to regress (for regression problems). To estimate the optimal initialization variance, all the layers of the base network are fixed and the auxiliary classification weights $\mathbf{W}_i^{class}$, as well as the the scaling factor α_i, are estimated using gradient descent, i.e., $\Delta\alpha_i = -\eta\frac{\partial\mathcal{L}}{\partial\alpha_i}$, and $\Delta\mathbf{W}_i^{class} = -\eta\frac{\partial\mathcal{L}}{\partial\mathbf{W}_i^{class}}$, where $\mathcal{L}$ denotes the loss used for training the network and η is the learning rate. This process allows to effectively estimate the initialization variance of the i-th layer in order to facilitate the task at hand. To understand the effectiveness of the proposed method consider that the back-propagated gradients only pass through one non-linearity, lowering the risk of vanishing gradient phenomena, while the scale of the weights of the i-th layer are indirectly adapted through α_i to facilitate the task at hand and the employed activation, which is equivalent to initializing the layer with a different variance (since none of the actual weights are actually modified during this process).

After estimating the variance for the first layer, the layer is appropriately re-initialized and the weights $\mathbf{W}_i^{class}$ are discarded. Next, the initialization process continues with the remaining layers. Then, after all the layers have been initialized using the proposed method, the network is trained in an end-to-end fashion, as any regular deep neural network architecture. Note again that during the initialization process none of the layers are actually trained. Therefore, the representations extracted from them are random transformations / projections of the input data. Such kind of transformations come with strong theoretical guarantees, since it has been shown that they can be used as universal function approximators [30], [31]. This further explains why the proposed method is capable of estimating the initialization variance despite relying on random projections of the input.

IV. Experimental Evaluation

For evaluating the proposed method a challenging large-scale high-frequency limit order book dataset (FI-2010) is employed [19]. The FI-2010 dataset contains more than 4 million limit orders collected from 5 Finnish companies. The anchored evaluation setup and pre-processing scheme proposed and described in [19] were used for the evaluation (splits 1 to 5 were used for the conducted experiments). The forecasting task concerns the prediction of the direction of the future mid-price movement (up, down or stationary) after 10 time-steps.

The DL model used for the conducted experiments is composed of a recurrent photonic layer with 32 neurons, as described in Section II. The output of this layer is then fed to two fully connected layers with 512 and 3 output neurons respectively. Again, the same photonic architecture is employed for these layers [10]. The length of the time series fed to the model was restricted to 10 (current and 9 previous time-steps), while the optimization was performed using the RMSprop algorithm with a learning rate of $\eta = 10^{-4}$. The DL models were trained for 20 epochs. The scaling factors α_i were estimated using 10 training epochs, while to accelerate the converge, only 5 recurrent time-steps were performed and the learning rate for the scaling factor was set to 0.1.

TABLE I: Evaluating the effect of noise on various parts of a recurrent photonic architecture

σ^2	Regular Training	Noisy Training	Adaptive Init.	Proposed
Weight Noise ($\sigma_w = \sigma, \sigma_{in} = \sigma_a = 0$)				
.1	0.0807	0.1146	0.1498	**0.1712**
.3	0.0486	0.1257	0.0809	**0.1514**
.5	0.0138	0.1199	0.0096	**0.1228**
Activation Noise ($\sigma_a = \sigma, \sigma_{in} = \sigma_w = 0$)				
.1	0.0884	0.1079	0.1693	**0.1897**
.3	0.0681	0.1197	0.1410	**0.1732**
.5	0.0267	0.1041	0.0674	**0.1351**
Input noise ($\sigma_{in} = \sigma, \sigma_a = \sigma_w = 0$)				
.1	0.0923	0.1102	0.1324	**0.1626**
.3	0.0635	0.1128	0.0975	**0.1632**
.5	0.0179	0.1003	0.0381	**0.1068**
Weight/Activation/Input Noise ($\sigma_{in} = \sigma_a = \sigma_w = \sigma$)				
.1	0.0627	0.1195	0.1351	**0.1566**
.3	0.0274	0.1085	0.0552	**0.1253**
.5	0.0058	0.0917	0.0078	**0.1000**
No noise + Adaptive Initialization: 0.1738				

The evaluation results are reported in Table I. The κ statistic is used to measure the forecasting accuracy [32], since the dataset is highly unbalanced, while the mean value over the 5 evaluation splits is reported. We separately evaluate the effect of noise on the weights (rows 3-5), activation function (rows 7-9) and input (rows 11-13). Furthermore, we also evaluated the cumulative effect of noise on all the components of the architecture (rows 15-17). We also evaluated four different training approaches: a) directly training the models using Xavier initialization (column 2) [28], b) training the models using Xavier initialization and applying noise during the training process (column 3), c) training the models by estimating the most appropriate initialization variance (column 4, without taking into account the existence of noise [29]) and d) using the proposed adaptive initialization method (column 5), which estimates the initialization variance taking into account the existence of noise (by applying the noise during the initialization process as well). Note that noise is always applied during the inference process, regardless the employed training process. The exact amount of noise used for each experiment is reported in column 1.

Several interesting conclusions can be drawn from the results reported in Table I. First, note that the performance of the models always increases when noisy training is used. Actually, as shown in column 2, the forecasting accuracy is only slightly affected by the employed noise, since in most of the cases we obtain a κ value of around 0.1, which is significantly improved compared to regular training, where κ collapses to less than 0.02, when higher levels of noise are introduced ($\sigma^2 = 0.5$). Just applying adaptive initialization by itself can improve the results when small amounts of noise exist, leading to an impressive κ value of 0.17, but quickly degenerates to values close to zero when the SNR falls further. At this point it is worth noting that a Photonic RNN with the same architecture trained with adaptive initialization and evaluated without noise reaches a κ value of 0.1738. Finally,

significant improvements are observed when the proposed method is applied, since the best κ value is obtained in every case. It is actually worth noting that when the noise is applied on the activations only and using relatively high SNR ($\sigma^2 = 0.1$) the proposed method performs even better than the baseline model, highlighting the effectiveness of the proposed method and confirming the regularizing nature of small amounts of noise [33].

V. Conclusions

A novel noise-aware approach for training recurrent photonic neural networks was proposed. The proposed method models the various noise sources that usually exist in most existing photonic platforms, providing a way to train models that are more robust to noise, without requiring any additional hardware components or design to mitigate the effects of noise. Apart from providing more accurate models, the proposed method opens several interesting future research directions on co-designing nerumorphic photonics, including developing DL models that can work on lower SNRs, leading to more energy efficient solutions.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[2] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, 2018.

[3] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2339–2348.

[4] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer, "cudnn: Efficient primitives for deep learning," *arXiv:1410.0759*, 2014.

[5] Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al., "In-datacenter performance analysis of a tensor processing unit," in *Proc. of ACM/IEEE Annual Int. Symposium on Computer Architecture*, 2017, pp. 1–12.

[6] Giacomo Indiveri, Bernabé Linares-Barranco, Tara Julia Hamilton, André Van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih-Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, et al., "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, pp. 73, 2011.

[7] Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B Bhadviya, Pinaki Mazumder, and Wei Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[8] Yichen Shen, Nicholas C Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, et al., "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, no. 7, pp. 441, 2017.

[9] Xing Lin, Yair Rivenson, Nezih T Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan, "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018.

[10] G Mourgias-Alexandris, A Tsakyridis, N Passalis, A Tefas, K Vyrsokinos, and N Pleros, "An all-optical neuron with sigmoid activation function," *Optics express*, vol. 27, no. 7, pp. 9620–9630, 2019.

[11] Alexander N Tait, Thomas Ferreira De Lima, Ellen Zhou, Allie X Wu, Mitchell A Nahmias, Bhavin J Shastri, and Paul R Prucnal, "Neuromorphic photonic networks using silicon photonic weight banks," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. Int. Conf. on Computer Vision*, 2015, pp. 1026–1034.

[13] Nikolaos Passalis, George Mourgias-Alexandris, Apostolos Tsakyridis, Nikos Pleros, and Anastasios Tefas, "Training deep photonic convolutional neural networks with sinusoidal activations," *IEEE Trans. on Emerging Topics in Computational Intelligence*, 2019.

[14] Zengguang Cheng, Carlos Ríos, Wolfram HP Pernice, C David Wright, and Harish Bhaskaran, "On-chip photonic synapse," *Science advances*, vol. 3, no. 9, pp. e1700160, 2017.

[15] J Feldmann, N Youngblood, CD Wright, H Bhaskaran, and WHP Pernice, "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature*, vol. 569, no. 7755, pp. 208–214, 2019.

[16] Alexander N Tait, Allie X Wu, Thomas Ferreira De Lima, Mitchell A Nahmias, Bhavin J Shastri, and Paul R Prucnal, "Two-pole microring weight banks," *Optics letters*, vol. 43, no. 10, pp. 2276–2279, 2018.

[17] T. F. de Lima, A. N. Tait, H. Saeidi, M. A. Nahmias, H. Peng, S. Abbaslou, B. J. Shastri, and P. R. Prucnal, "Noise analysis of photonic modulator neurons," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–9, Jan 2020.

[18] Kenji Kawaguchi, "Deep learning without poor local minima," in *Proc. of the Advances in Neural Information Processing Systems*, 2016, pp. 586–594.

[19] Paraskevi Nousi et al., "Machine learning for forecasting mid-price movements using limit order book data," *IEEE Access*, vol. 7, pp. 64722–64736, 2019.

[20] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber, "Lstm: A search space odyssey," *IEEE Trans. on Neural Netw. and Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, 2016.

[21] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv:1412.3555*, 2014.

[22] Hsuan-Tung Peng, Mitchell A Nahmias, Thomas Ferreira De Lima, Alexander N Tait, and Bhavin J Shastri, "Neuromorphic photonic integrated circuits," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 24, no. 6, pp. 1–15, 2018.

[23] G. Mourgias-Alexandris, A. Totovic, A. Tsakyridis, N. Passalis, A. Tefas, K. Vyrsokinos, and N. Pleros, "A linear photonic neuron using an extended iq modulator cell for enabling all-optical coherent sigmoid neurons," in *Proc. European Conf. on Optical Communication*, 2019.

[24] Indranil Chakraborty, Gobinda Saha, Abhronil Sengupta, and Kaushik Roy, "Toward fast neural computing using all-photonic phase change spiking neurons," *Scientific reports*, vol. 8, no. 1, pp. 12980, 2018.

[25] C. Huang, T. Ferreira De Lima, A. Jha, S. Abbaslou, A. N. Tait, B. J. Shastri, and P. R. Prucnal, "Programmable silicon photonic optical thresholder," *IEEE Photonics Technology Letters*, vol. 31, no. 22, pp. 1834–1837, Nov 2019.

[26] X. Li, R. Mardling, and J. Armstrong, "Channel capacity of im/dd optical communication systems and of aco-ofdm," in *2007 IEEE Int. Conference on Communications*, June 2007, pp. 2128–2133.

[27] R. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," *Journal of Lightwave Technology*, vol. 28, no. 4, pp. 662–701, Feb 2010.

[28] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[29] Nikolaos Passalis, George Mourgias-Alexandris, Apostolos Tsakyridis, Nikos Pleros, and Anastasios Tefas, "Initializing photonic feed-forward neural networks using auxiliary tasks," *Neural Networks*, 2020.

[30] Ali Rahimi and Benjamin Recht, "Random features for large-scale kernel machines," in *Proc. Advances in Neural Information Processing Systems*, 2008, pp. 1177–1184.

[31] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, vol. 42, no. 2, pp. 513–529, 2011.

[32] Mary L McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica: Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[33] Chris M Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.