VARIANCE PRESERVING INITIALIZATION FOR TRAINING DEEP NEUROMORPHIC PHOTONIC NETWORKS WITH SINUSOIDAL ACTIVATIONS

Nikolaos Passalis[1,3], *George Mourgias-Alexandris*[2], *Apostolos Tsakyridis*[2], *Nikos Pleros*[2] *and Anastasios Tefas*[1]

[1]Artificial Intelligence and Information Analysis Laboratory
[2]Photonic Systems and Networks Research Group
School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
[3]Faculty of Information Technology and Communication Sciences, Tampere University, Finland
nikolaos.passalis@tuni.fi, {mourgias, atsakyrid, npleros, tefas}@csd.auth.gr

ABSTRACT

Photonic neuromorphic hardware can provide significant performance benefits for Deep Learning (DL) applications by accelerating and reducing the energy requirements of DL models. However, photonic neuromorphic architectures employ different activation elements than those traditionally used in DL, slowing down the convergence of the training process for such architectures. An initialization scheme that can be used to efficiently train deep photonic networks that employ quadratic sinusoidal activation functions is proposed in this paper. The proposed initialization scheme can overcome these limitations, leading to faster and more stable training of deep photonic neural networks. The ability of the proposed method to improve the convergence of the training process is experimentally demonstrated using two different DL architectures and two datasets.

Index Terms— Neuromorphic Hardware, Photonic Neural Networks, Sinusoidal Activations

1. INTRODUCTION

Deep Learning (DL) provided powerful methods capable of achieving state-of-the-art performance on several difficult problems [1]. However, training and deploying DL models requires powerful and specialized hardware. It is worth noting that a significant part of the progress in DL has been attributed to the wide availability of such hardware that can be used for this task [1]. To this end, Graphical Processing Units (GPUs) are usually used to improve the training and inference speeds, as well as reduce the energy requirements, both of which are critical for successfully deploying DL models in large-scale applications. Apart from using general purpose hardware accelerators, like GPUs, developing neuromorphic hardware solutions, that directly implement the network architectures in hardware, have been proposed to further accelerate neural networks [2, 3, 4]. Even though these solutions are not yet widely used, it has been successfully demonstrated that it is indeed possible to improve the inference speed and reduce the energy requirements of deep neural networks, hinting that neuromorphic hardware can constitute an interesting alternative to the generic hardware accelerators that are predominantly used.

The use of *photonics* for providing hardware implementations of neural networks is perhaps among the most promising neuromorphic solutions [5], overcoming several limitations of existing neuromorphic hardware by using optical processing elements instead of purely eletrical ones. In neuromorphic photonics the input to a neural network is represented using an optical signal that is then appropriately manipulated through hardware components in order to provide the functionality of neural layers. The signal can propagate through the used optical hardware with speeds near to the speed of light, while the great bandwidth of optical systems allows for parallel processing of enormous amounts of data, potentially outperforming the currently used solutions by several orders of magnitude [5]. The momentum of photons to drive low-energy and ultra-fast processing engines has been already efficiently exploited in the implementation of reservoir computing [6] and all-passive optical Deep Neural Networks (DNNs) [7]. More recently, the employment of Photonic Integrated Circuits (PICs) for deploying integrated weighting banks moved towards an integrated neural network with sinusoidal activation elements [8].

However, the use of neuromorphic hardware always comes with limitations that are usually ignored when training neural networks that will be simulated instead of being actually implemented on hardware [9, 10]. Among the most important ones is that the activation functions, that are traditionally used in deep learning, cannot be precisely implemented. To this end, several different combinations of electrical and photonic elements have been proposed to simulate the effect of non-linear activation functions. In this work, the activation scheme proposed in [8] is consider, i.e., a Mach-Zehnder Modulator [11] is employed to electrically modulate an optical signal according to the weighted optical

output of a neuron. This configuration leads to the following transfer function:

$$P_{out} = P_{in} \sin^2(\frac{\pi}{2} \frac{V_{RF}}{V_{\pi}}), \quad (1)$$

where P_{out} is the output signal, P_{in} is the reference optical signal to be modulated, V_{π} is the voltage required for an exact π phase shift and V_{RF} is the input to the activation function. This configuration requires the use of photodetectors to convert the optical output of a neuron back to an electrical signal that is then used to modulate a reference optical signal. Even though this solution involves electronics, it can operate at high frequencies due to the use of Germanium-based photodetectors [12].

Using highly non-linear activation functions, such as the sigmoid function, can slow down (or even completely halt) the training of deep neural networks [13], due to several phenomena, including the vanishing of the input/back-propagated signals. The activation function described in (1) is significantly different from the activation functions that are usually used in deep neural networks. Its behavior is compared to the *ReLU* activation in Figure 1. Note the periodic and highly non-linear behavior of the employed *photonic* activation compared to the regular ReLU. It has been demonstrated that using *variance preserving* initialization schemes can significantly improve the speed and stability of the training process [14, 15], when similar non-linear activation functions were employed. However, it was established that different activation functions require using different initialization schemes to ensure that the input signal will not diminish and that the gradients will correctly back-propagate. Failing to use an initialization scheme that is correctly designed for the activation function at hand can stall the training process or lead to sub-optimal results [15]. Therefore, even though these photonic neuromorphic implementations can significantly improve the inference speed, further advances are required in the way that neural networks are designed and trained in order to fully exploit the potential of such photonic hardware.

The main contribution of this paper is the proposal of an initialization scheme for deep neural networks that is adapted to the quadratic sinusoidal function that is physically realized by the photonic neuromorphic processor proposed in [8]. An analytical expression for the optimal variance during initialization, following the variance preserving assumption [14], is derived. It is experimentally demonstrated, using two different datasets, that the proposed method can indeed significantly improve the convergence of deep neural networks allowing for training deep neural networks on photonic hardware. This work is related to training neural networks for neuromorphic hardware [9, 10, 16, 17]. However, to the best of our knowledge this is the first work that considers the implications arising from the use of different activation functions in photonic neuromorphic hardware. Note that the proposed approach can be used both for the *ex-situ* training of neuromorphic hardware, i.e., training using conventional hardware and then deploying the network on the neuromorphic hardware, as well as for the gradient descent-based *in-situ* training, i.e., training directly on the neuromorphic hardware.

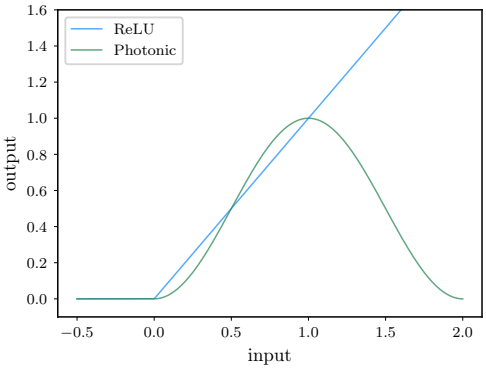

Fig. 1: Comparing the quadratic sinusoidal photonic activation to the *ReLU* activation

The rest of the paper is structured as follows. First, the proposed initialization scheme is analytically derived in Section 2. Then, the experimental evaluation is provided in Section 3, while conclusions are drawn in Section 4.

2. PROPOSED METHOD

A deep neural network is composed of multiple non-linear processing layers, while each layer is composed of several neurons. A neuron of the l-th layer receives its input, denoted by a vector $\mathbf{x}^{(l-1)} \in \mathbb{R}^{n_{l-1}}$, where n_{l-1} is the dimensionality of the input, and then the input is multiplied with the synaptic weights of the i-th neuron, denoted by $\mathbf{w}_i^{(l)} \in \mathbb{R}^{n_l}$, and then summed. This process is repeated for all the neurons of the l-th layer leading to the activation vector $\mathbf{u}^l \in \mathbb{R}^{n^l}$, where n_l is the number of neurons in the l-th layer, i.e., $\mathbf{u}^{(l)} = \mathbf{W}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)} \in \mathbb{R}^{n_l}$, $\mathbf{W}^{(l)} = [\mathbf{w}_,^{(l)}\mathbf{w}_2^{(l)}, \ldots, \mathbf{w}_{n_l}^{(l)}] \in \mathbb{R}^{n_l \times n_{l-1}}$ is the weight matrix of the l-th layer and $\mathbf{b} \in \mathbb{R}^{n_l}$ is a vector containing the bias terms. The final output of the neuron is obtained by applying a non-linear activation function $f(\cdot)$, i.e., $\mathbf{x}^{(l)} = f(\mathbf{u}^{(l)}) \in \mathbb{R}^{n_l}$. The terms activation and output of a neuron are not consistently used in the literature. In this paper, the term *activation* is used to denote the quantity $\mathbf{u}^{(l)}$, i.e., the output of the neuron *before* applying the activation function, while the term *output* is used to the denote the final output of a neuron $\mathbf{u}^{(l)}$ *after* applying the activation function $f(\cdot)$.

The transfer function of the activation element employed in [8] can be simplified as:

$$f(u) = \begin{cases} 0, & \text{if } u \leq 0 \\ sin^2(\frac{\pi}{2}u), & \text{if } u > 0 \end{cases}, \quad (2)$$

after considering the physical limitations of the components (negative power can not be represented) and that the quantity u can be appropriately scaled during the implementation (according to the voltage V_π required to achieve a phase shift of π).

As discussed in [14], using a proper initialization scheme before initiating the training process is of crucial importance. The characteristics of the initialization process, that should be used when photonic sinusoidal activation elements are employed, are derived in this paper following the hypothesis proposed in [14], i.e., that the variance of the input and back-propogated signals must be kept constant through the various layers of the networks. This allows the information to arrive to the output of the network, as well as, the gradients to be effectively back-propagated, avoiding vanishing gradients phenomena.

The weights of the l-th layer are initialized by drawing from a random variable denoted by $w^{(l)}$, while the notations $u^{(l)}$ and $x^{(l)}$ are used to refer to the random variables that correspond to the outputs $\mathbf{u}_i^{(l)}$ and $\mathbf{x}_i^{(l)}$ of the l-th layer. We assume that the elements of $\mathbf{W}^{(l)}$ share the same distribution, are mutually independent and have zero mean, i.e., $E[w^{(l)}] = 0$. Furthermore, the elements in $\mathbf{x}^{(l)}$ are also mutually independent and drawn from $x^{(l)}$, while it is assumed that the variables $w^{(l)}$ and $x^{(l)}$ are independent. For the scope of this analysis, the behavior of the employed activation function (for positive inputs) is approximated using a first-order Taylor expansion around u_0. Therefore, the activation function used for calculating the variance of the distribution that will be used for the initialization of the network is defined as:

$$\hat{f}(u) = \begin{cases} c_1(u - u_0) + c_0, & \text{if } u > 0 \\ 0, & \text{if } u \leq 0 \end{cases}. \tag{3}$$

where $c_1 = f'(u_0) = \frac{1}{2}\pi \sin(\pi u_0)$ and $c_0 = f(u_0) = \sin^2(\frac{1}{2}u_0)$. The feed-forward case is examined. The variance of the output of the l-th layer is calculated as:

$$\begin{aligned} Var[u^{(l)}] &= Var[\mathbf{u}_k^{(l)}] \\ &= Var[\sum_{i=1}^{n_{l-1}} \mathbf{W}_{ki}^{(l)} \mathbf{x}_i^{(l-1)}] \\ &= \sum_{i=1}^{n_{l-1}} Var[w^{(l)}] E[(\mathbf{x}_i^{(l-1)})^2]. \end{aligned} \tag{4}$$

Assuming that $w^{(l)}$ is symmetric around zero and the bias terms are initialized to 0, then it is easy to see that $E[u^{(l)}] = 0$ and $u^{(l)}$ is also symmetric around zero. By observing that the activation function maps half of its values to 0 and using a first-order Taylor series's approximation for the other half, then $E[(\mathbf{x}_i^{(l-1)})^2]$ can be calculated as:

$$\begin{aligned} E[(\mathbf{x}_i^{(l-1)})^2] &= \frac{1}{2} c_1^2 E[(\mathbf{u}_i^{(l-1)})^2] \\ &= \frac{1}{2} c_1^2 Var[u^{(l-1)}]. \end{aligned} \tag{5}$$

Therefore, the variance for the l-th layer is calculated as:

$$\begin{aligned} Var[u^{(l)}] &= \frac{1}{2} c_1^2 \sum_{i=1}^{n_{l-1}} Var[w^{(l)}] Var[u^{(l-1)}] \\ &= \frac{c_1^2}{2} n_{l-1} Var[w^{(l)}] Var[u^{(l-1)}] \\ &= Var[x^{(0)}] \prod_{i=1}^{l-1} \frac{c_1^2}{2} n_{i-1} Var[w^{(i)}], \end{aligned} \tag{6}$$

by plugging (5) into (4). Note that $\mathbf{x}^{(0)}$ corresponds to the input of the neural network and n_0 is the number of input dimensions. To ensure that the variance is equal across all the layers, the product that appears in (6) must be equal to 1. This ensures that the variance will be kept constant across the layers. Therefore, the weights of the l-th layer must be appropriately initialized from a distribution with the following variance:

$$Var[w^{(i)}] = \frac{2}{c_1^2 n_{i-1}}. \tag{7}$$

If a normal distribution is used for initializing the i-th layer, then its standard deviation must be set to $\sigma = \frac{1}{\sigma_1}\sqrt{\frac{2}{n_{i-1}}}$. If a uniform distribution is used, then the weights must be sampled from the interval $[-\alpha, \alpha]$, where $\alpha = \frac{1}{\sigma_1}\sqrt{\frac{6}{n_{i-1}}}$.

Similar analysis can be also used to derive the variance needed for successfully back-propagating the gradients through the network. However, as also proposed in [15], either of them can be used interchangeably in most of the cases without significant differences in the performance. Furthermore, note that the conducted analysis considers only the case of fully connected layers. However, it is straightforward to adapt the initialization scheme to other types of layers, e.g., convolutional layer, simply by correctly setting n_{i-1} to the *fan-in* of each neuron, as proposed in [15]. In the conducted experiments it is demonstrated that the proposed method works equally well for both fully connected and convolutional layers.

3. EXPERIMENTAL EVALUATION

The proposed initialization scheme was evaluated using two image datasets, the MNIST and Fashion MNIST datasets [18], as well as two different deep learning convolutional architectures: a) a 4-layer network and b) a 6-layer network. The first network is composed of two convolutional layers (3×3 kernels) with 32 and 64 filters, followed by two fully connected layers with 512 and 10 neurons respectively. Dropout with rate 0.5 is used before and after the first fully connected layer. Average pooling with kernel size 2×2 is used after each convolutional layer (note that currently there is no method proposed in the literature for implementing max pooling in

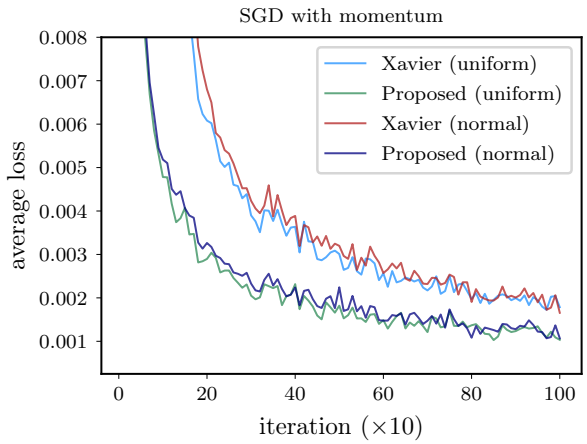

Fig. 2: Learning curves for two different initialization schemes (Xavier [14] and proposed) and two different random distributions (uniform and normal).

photonic networks). The second network follows a similar architecture, i.e., four convolutional layers with 32, 64, 128 and 256 filters (average pooling is used after the second and the fourth layers) are used, while the fully connected layer is composed of 1024 neurons. The cross entropy loss, together with the softmax activation function, is used for training the networks. Note that the softmax activation is only needed during the training process and it can be removed during the deployment. The initial weights were drawn from a uniform distribution for all the conducted experiments, unless otherwise stated. Finally, the activation function was approximated around $u_0 = \frac{1}{4}$, which consistently yielded the best results in all the conducted experiments.

First, the proposed initialization scheme is compared to the well-known Xavier initialization (as proposed in [14]) in Figure 2. The stochastic gradient descent method (the learning rate was set to $\eta = 0.1$) with momenum of 0.9 was used for these experiments. The proposed method leads to significantly faster convergence than the Xavier initialization regardless the distribution used for drawing the weights.

Then, the improved behavior of the proposed initialization scheme was also confirmed in the experimental results provided in Table 1 using the MNIST dataset and the first neural network architecture. The Adam optimizer was used for the conducted experiments [19]. The baseline (denoted by "ReLU") consists of a regular deep neural network with the same architecture using the ReLU activation (and the appropriate initialization scheme [15]) instead of the photonic activation. The proposed combination of the employed photonic activation and initialization scheme significantly improves the performance over using the Xavier initialization. It also leads to slightly improved performance over the baseline deep network (which was a surprising result, given the periodic and highly non-linear nature of the employed activation compared to the ReLU, hinting that such function can potentially have a positive regularization effect). Similar conclusions can be also drawn from the experiments conducted using the Fashion MNIST dataset and both network architectures, where the proposed initialization scheme also always improves both the training and testing error over the Xavier initilization. Note that the first model (4-layer network) was under-fitting the data, while the second one (6-layer network) was more powerful leading to better performance.

Table 1: Evaluation using the MNIST dataset. The optimization ran for 10 epochs (learning rate set to 0.001) with batch size of 128.

Model	**train err.**	**test err.**
ReLU	0.49	0.79
Photonic + Xavier	0.59	1.00
Photonic + proposed	**0.37**	**0.78**

Table 2: Evaluation using the Fashion MNIST dataset. The optimization ran for 20 iterations (learning rate set to 0.001) with batch size of 128, followed by 20 additional iterations (learning rate set to 0.0001).

Model	**Network**	**train err.**	**test err.**
ReLU	4 layer	10.61	10.28
Photonic + Xavier	4 layer	12.62	12.28
Photonic + Proposed	4 layer	**11.84**	**11.71**
ReLU	6 layer	5.62	6.56
Photonic + Xavier	6 layer	6.83	7.44
Photonic + Proposed	6 layer	**6.12**	**7.11**

4. CONCLUSIONS

In this work, a variance preserving initialization scheme that can be used to efficiently train deep photonic networks that employ quadratic sinusoidal activation functions was proposed. The proposed method allows for training deeper neural networks for neuromorphic photonic hardware, that can be used to significantly accelerate and reduce the energy requirements of Deep Learning (DL). The ability of the proposed method to improve the convergence of the training process was experimentally demonstrated using two different DL architectures and two datasets. Apart from studying the initialization schemes for deep photonic networks, several interesting future work directions exist, e.g., imposing additional constraints on the activations/weights to ensure the stable behavior of the network on photonic hardware and deriving methods to directly adapt a trained neural network for deployment on photonic hardware without training it from scratch.

5. REFERENCES

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

[2] 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.

[3] 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.

[4] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[5] 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.

[6] Kristof Vandoorne, Pauline Mechet, Thomas Van Vaerenbergh, Martin Fiers, Geert Morthier, David Verstraeten, Benjamin Schrauwen, Joni Dambre, and Peter Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, pp. 3541, 2014.

[7] 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.

[8] Alexander N Tait, Thomas Ferreira 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. 7430, 2017.

[9] Chris Yakopcic, Raqibul Hasan, and Tarek M Taha, "Memristor based neuromorphic circuit for ex-situ training of multi-layer neural network algorithms," in *Proceedings of the International Joint Conference on Neural Networks*, 2015, pp. 1–7.

[10] Eric Hunsberger and Chris Eliasmith, "Training spiking deep networks for neuromorphic hardware," *arXiv preprint arXiv:1611.05141*, 2016.

[11] Stelios Pitris, Charoula Mitsolidou, Theoni Alexoudi, Diego Pérez-Galacho, Laurent Vivien, Charles Baudot, Peter De Heyn, Joris Van Campenhout, Delphine Marris-Morini, and Nikos Pleros, "O-band energy-efficient broadcast-friendly interconnection scheme with sipho mach-zehnder modulator (mzm) & arrayed waveguide grating router (awgr)," in *Optical Fiber Communication Conference*, 2018, pp. Th1G–5.

[12] M Pantouvaki, SA Srinivasan, Y Ban, P De Heyn, P Verheyen, G Lepage, H Chen, J De Coster, N Golshani, S Balakrishnan, et al., "Active components for 50 gb/s nrz-ook optical interconnects in a silicon photonics platform," *Journal of Lightwave Technology*, vol. 35, no. 4, pp. 631–638, 2017.

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[14] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[16] Mirko Prezioso, Farnood Merrikh-Bayat, BD Hoskins, GC Adam, Konstantin K Likharev, and Dmitri B Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61, 2015.

[17] Steve K Esser, Rathinakumar Appuswamy, Paul Merolla, John V Arthur, and Dharmendra S Modha, "Backpropagation for energy-efficient neuromorphic computing," in *Proceedings of the Advances in Neural Information Processing Systems*, 2015, pp. 1117–1125.

[18] Han Xiao, Kashif Rasul, and Roland Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.