Learning photonic neural network initialization for noise-aware end-to-end fiber transmission

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

***Abstract*—Deep Learning (DL) has dominated a wide range of applications due to its state-of-the-art performance. Novel approaches introduce Artificial Neural Networks (ANNs) on fiber communication channels to be used as intensity modulation/direct detection (IM/DD) systems and optimized in an end-to-end fashion. Despite the potential of these methods, the demanding nature of DL models limits their applications in such domains, where fast inference and low power consumption is required. Indeed, these limitations fueled the research on neuromophic architectures, including neuromorphic photonics, which holds the credentials for unlocking matrix multiplications at high frequencies, while minimizing energy consumption. However, at the same time, photonic architectures impose new challenges to DL training due to the underlying hardware constraints. In this paper, we present a trainable data-driven noise-aware initialization method oriented to easily saturated activation functions, such as those typically used in optical neurons on the transmitter and receiver side of a noisy IM/DD system intercepted by a noisy channel. The proposed method is evaluated on a fully optical IM/DD system using different fiber lengths, overcoming issues such vanishing gradient phenomena that profoundly hinders the training of the receiver and transmitter photonic neural networks (PNNs), while also improving robustness to noise.**

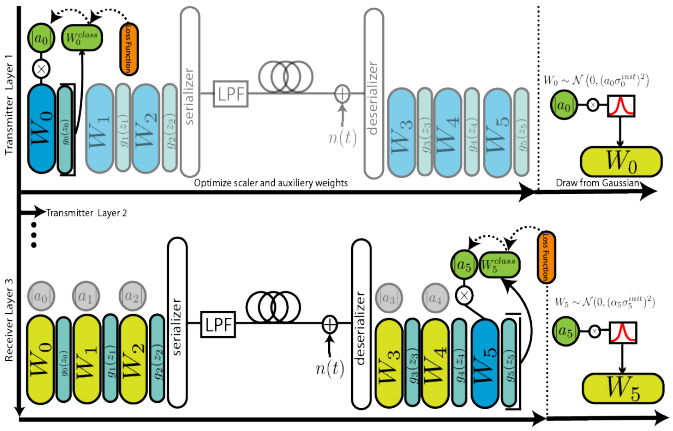

Fig. 1. Schematic representation of the proposed noise-aware training method. From top to bottom: the proposed initialization method is employed iteratively to all layers for both transmitter and receiver. From left to right: first the appropriate variance is estimated using the auxiliary task and then weights of the layer are initialized. After initializing the weights of a layer, the auxiliary parameters are discarded.

I. Introduction

DL has been extensively applied by both academic community and industry leading to state-of-the-art performance [1]. Over the recent years, there is an increasing interest in employing DL in the communication domain [2], ranging from wireless [3] to optical fiber communications [4], exploiting the intrinsic ability of ANNs to compensate noise, especially when they are trained to withstand it [5], [6]. Such approaches design the communication system by carrying out the optimization in a single end-to-end process, including the transmitter, receiver and communication channel, with the ultimate goal to achieve an optimal end-to-end performance by acquiring a robust representation of the input message [7].

Such novel approaches, which treat the optimization process in an end-to-end fashion, gaining attention especially for fiber optical communication, such as IM/DD systems, which are currently the preferred choice in many datacenters, access, metro and backhaul applications [8]. IM/DD systems have limitations due to the nonlinear impairments originating from the fiber dispersion followed by the square-law direct detection. On top of that, various noise sources including the shot noise of the photodiode (PD), the $sin^2(x)$ transfer function of Mach-Zehnder Interferometric modulator (MZM) and the low-pass frequency response of almost every component, deteriorates further the signal quality of such optical communication links.

Although it has been shown that DL can compensate these phenomena that occur in communication channels [7], its application is hindered by the high complexity of DL models, which increasingly demands more powerful and energy consuming hardware [9]. Consequently, specialized hardware accelerators have been developed, ranging from Tensor Processing Units [10] to advanced neuromorphic hardware [11], increasing both training and inference speed, while also reducing power and energy consumption.

To this end, *photonic* hardware is gaining attention as a very promising approach, due to their ability to provide ultra fast matrix-based operations with very low power consumption [12], [13]. In neuromorphic photonics, signals are encoded using light, instead of electrical quantities, which are then manipulated to provide the neuron's functionality [14], [15]. Such approaches, ranging from purely optical components to advanced combinations of electro-optical devices [16], [17], have great advantages over their electronic counterparts due to their massive parallelism potential, enabled by their enormous bandwidth [18], [19], as well as the ability to operate in high frequencies [20].

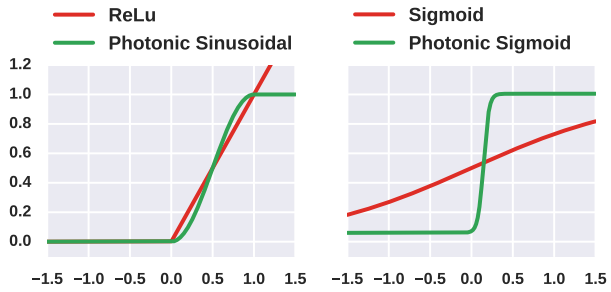

Fig. 2. Photonic activations functions can be easily saturated compared to typically ones used in most DL architectures

However, the unique nature of photonic hardware dictates constraints on the neuromorphic photonic implementation that hinders its application. More precisely, PNNs rely mostly on sigmoid [17] and sinusoidal [21] based activation functions that are susceptible to early saturation in contrast to the traditionally used functions (such as ReLU [22]), as depicted in Fig. 2. As a result, photonic activation functions work on smaller regions of the input domain leading to narrow activation windows making them easily saturated and susceptible to vanishing gradient phenomena [23]. Vanishing gradient phenomena can lead to significant performance deterioration since the model can be trapped in a bad local minimum or even halt the training in early stages, making PNNs highly sensitive to the initialization schemes.

Vanishing gradients phenomena have been extensively studied for the regular DL training and several enhancements have been proposed to this end, such as variance-preserving initialization schemes [24], [25]. However such schemes typically target traditionally used activation functions (such as ReLU), limiting their usefulness in photonic architectures. Even specifically tailored initialization methods that have been proposed can still lead to suboptimal results compared to regular DL models [21]. This behavior can be attributed to the assumptions involved in the methods, e.g., data are assumed to be normally distributed and/or linear approximations are employed for the activation functions, some of which are not always satisfied, limiting the performance of the models.

Our work introduces an end-to-end deep learning fiber communication transceiver design inspired by [7], emphasizing on training by examining all optical activation schemes and respective limitations present in realistic demonstrations. More specifically, we focus on training photonic architectures which employ all optical activation schemes [17], [21], by simulating their given transfer functions. Thus, the main contribution of this paper is a data-driven noise-aware initialization method that is capable of initializing PNNs by taking into account the actual data distribution, noise sources as well as the unique nature of photonic activation functions, as shown in Fig. 1. This allows for reducing the effect of vanishing gradient phenomena, as well as improving the ability of networks coupled with communication systems to withstand noise, e.g., due to the optical transmission link. To this end, we employ an auxiliary task in order to approximate the optimal initialization scheme that will allow the information to flow through the initial state of the networks connected through the employed communication channel, greatly improving the efficiency of backpropagation. As experimentally demonstrated, the proposed method significantly resists the degradation that occurred when easily saturated photonic activations are employed as well as significantly improves the signal reconstruction of the all optical IM/DD system. To the best of our knowledge, this is the first approach that is capable of appropriately initializing photonic transmitter and receiver networks by taking into account the limited activation range of photonic activation functions, as well as the non-linear corruption that exists in the optical channel due to noise source and fiber dispersion.

The rest of the paper is structured as follows. In Section II we present the employed IM/DD setup, while in Section III we present the proposed training method. Then, we demonstrate the effectiveness of the proposed method on the employed setup in Section IV. Finally, in Section V the conclusions are drawn.

II. End-to-End Learning in IM/DD systems

Same as in software-based DL, photonic neuromorphic implementations rely on the perceptron with its ultimate goal to approximate a function f^* by mapping an input $\boldsymbol{x} \in \mathbb{R}^M$, where M is the number of observations fed to an ANN, to a category (typically $\boldsymbol{y} \in \mathbb{R}^N$ when one-hot encoding is used) for classification problems, or to a continuous vector $\boldsymbol{y} \in \mathbb{R}^N$ for regression problem, i.e., $f^*(\boldsymbol{x}) = \boldsymbol{y}$, where N is the number of categories or values to regress respectively. In turn, a multi-layer perceptron approximates f^* by stacking many different layers, $f^{(n)}(...(f^{(2)}(f^{(1)}(\boldsymbol{x};\boldsymbol{\theta_1});\boldsymbol{\theta_2});\boldsymbol{\theta_n}) = \boldsymbol{y}$, and learn parameters $\boldsymbol{\theta}_i$, where $1 \leq i \leq n$ with $\boldsymbol{\theta}_i$ typically consisting of weights $\boldsymbol{W_i} \in \mathbb{R}^{M_i \times N_i}$ and biases $\boldsymbol{b}_i \in \mathbb{R}^{N_i}$. M_i and N_i refer to the number of input features and output neurons of the i-th layer. Accordingly, the linear part of the i-th layer of the model is defined as $u^{(i)}(\boldsymbol{y}_{i-1}) = \boldsymbol{W}_i^\top \boldsymbol{y}_{i-1} + \boldsymbol{b}_i$. The output of the linear part is then fed to the employed activation function $g(\cdot)$ to get the response of each layer as $\boldsymbol{y}_i = g(u^{(i)}(\boldsymbol{y}_{i-1}))$. Finally, the learning process aims to minimize the loss function $J(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{t})$ with respect to weights $\boldsymbol{W}$ and biases $\boldsymbol{b}$ using the backpropagation algorithm, where $\boldsymbol{t} \in \mathbb{R}^N$ denotes the training targets. For multi-class problems, we typically use the cross entropy loss defined as $J(\boldsymbol{\theta}; \boldsymbol{y}, \boldsymbol{t}) = -\sum_{c=1}^{N} t_c \log y_c$.

In this paper, we are using two photonic activation functions that correspond to different photonic configurations that can be used for providing the non-linear behavior required by ANNs. The first one is the photonic sigmoid introduced in [17]:

$$g(z) = A_2 + \frac{A_1 - A_2}{1 + e^{(z - z_0)/d}}, \tag{1}$$

where the parameters were set to $A_1 = 0.060$, $A_2 = 1.005$, $z_0 = 0.145$ and $d = 0.033$ according to the experimental observations from real hardware implementation, as reported in [17]. For the second activation function the layout proposed in [26] is used, which is based on a MZM [27] to appropriately

TABLE I
PHYSICAL CHARACTERISTICS OF THE CHANNEL

Sampling rate	336Gsa/s
LPF bandwidth	32GHz - Gaussian Filter, $\sigma = 0.7$
Fiber dispersion	18 ps/nm/km
Fiber attenuation	0.18 dB/km
MZM	$A(t) = sin(t)$
Fiber dispersion	$D(z, \omega) = \exp j(\beta_2/2)\omega^2 z$

modulate an optical signal, along with a diode [21]. The behavior of this activation is described by the transfer function:

$$g(z) = \begin{cases} 0, & \text{if } z < 0 \\ \sin^2 \frac{\pi}{2} z & \text{if } 0 \leq z \leq 1 \\ 1, & \text{if } z > 1 \end{cases}. \quad (2)$$

The limited range in which those optical activation functions work, as depicted in Fig. 2, imposes a careful design for the employed initialization scheme. Furthermore, high slopes, especially in photonic sigmoid, are apt to culminate in saturated area at early stages of training and result in a worse local minimum.

In this work, we employed the aforementioned activation functions in both the neural transmitter and receiver of the IM/DD system which is trained in an end-to-end fashion as a single feed-forward ANN, as proposed in [7]. The system accumulates the non-linear noise induced by the fiber dispersion and builds robust representations of the transmitted messages targeting to minimize the Bit Error Rate of the obtained signal on the transmitter side. The operation of the IM/DD system is also summarized, along with the proposed method, in Fig. 1.

On one end, the neural transmitter is composed of 3 fully connected layers and gets an input of a 6-bit symbol that is encoded into one-hot vector of size 64. In between is the channel which transmits the signal employing a non-linear transfer function with additional Gaussian noise. The major limitation of the channel, which is also responsible for the noise, is the intersymbol interference arising from optical fiber dispersion [28]. In order to accurately simulate the channel, the outputs of the transmitter are concatenated in a block of 11 neighboring samples to be passed into the channel, since the fiber dispersion introduces memory between several consecutive symbols. In turn, the signal passes through a Low-Pass Filter (LPF) to account for the finite bandwidth of the system. Table II summarizes mathematical expressions for the channel components and respective operational settings used for the software-based implementation. Finally, the symbols are deserialized to be fed into the receiver network. The receiver gets the deserialized output of the channel as an input of 48 features. Similarly to the transmitter, the receiver consists of 2-hidden layers and an output layer with softmax activation function. Note that we can train the system in an end-to-end manner since the transfer function and the derivative of channel's components can be analytical computed, as showed in [28]. This can in principle allow to run a simulated backpropagation through the channel and train the transmitter and receiver to account for the signal degradation introduced by the communication channel.

III. PROPOSED METHOD

Considering the aforementioned architecture we propose an initialization method that takes advantage of the effectiveness of training shallow ANNs (up to two layers) to incrementally estimate the most appropriate variance for initializing the weights of more complex architectures. The proposed method takes into account the noise and corruption that exist in the whole system, as well as the slope of the activation units. This is in contrast to the traditional initialization methods, such as Xavier [25] and He [24], that consider only the size of every layer (fan-in and fan-out) ignoring the synergistic effects of different components on the models. In this way, the proposed method is capable of modeling the effect of fiber dispersion on the system to better estimate the most appropriate initialization scheme to be applied.

The main hypothesis behind the proposed method is that there is an appropriate Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ for each layer (in both transmitter and receiver), which can maximize the information flow through the network during the initial steps of the training process by taking into account both the distribution of the training samples and the noise that arises through the noisy channel. Indeed, maximizing the available information during the initial stages of gradient descent ensures that information would not be blocked in the early layers of models, since the information lost in one layer cannot be recovered in the subsequent ones [29]. This is crucial for the proposed optical IM/DD system considering the synergistic effects of fiber dispersion, noise and high sloped photonic activation functions.

However, learning the parameters μ and σ directly is not possible, since the Gaussian distribution is not differentiable. At the same time, using an information-theoretic measure for estimating information flow, such as mutual information, is especially challenging and inefficient in high-dimension spaces [30]. To this end, we propose an auxiliary task based on the expectation that the information for the task at hand increases when the loss between the extracted representation and the target variable is minimized. In this way, fitting an auxiliary classification (or regression) layer can be used as a proxy approximator. Therefore, instead of directly learning the distribution parameters to maximize the mutual information between each layer's input and system's output, we proposed to optimize the parameters in order to maximize the information that can be extracted using a linear classifier by maximizing classification accuracy. Apart from using an efficient proxy for optimization, we employ a trainable parameter to rescale a fixed distribution $\mathcal{N}(0, 1)$. This allows for effectively providing a differentialable expression for optimizing a Gaussian distribution $\mathcal{N}(\mu, \tilde{\sigma}^2)$ using regular back-propagation.

The aforementioned procedure can be described as follows. First, an additional scale factor a_i is introduced for each layer,

$$\tilde{\boldsymbol{y}}_i = f(|a_i| \boldsymbol{W}_i^\top \boldsymbol{y}_{i-1} + \boldsymbol{b}_i) \in \mathbb{R}^{N_i}, \quad (3)$$

where $\tilde{\boldsymbol{y}}_i$ denotes the proxy output of the i-th layers which is used to learn the parameters of the Gaussian distribution and

$|\cdot|$ the absolute value operator. Assuming that the weights are initialized by drawing from a Gaussian distribution with zero mean and unit variance, altering the scaling factor results in adjusting initialization variance for each layer. In turn, to optimize scaling factor a_i an auxiliary linear classification layer is required, $\boldsymbol{W}_i^{class} \in \mathbb{R}^{N_i \times N}$, where N is the number of classes. Therefore, the output of the auxiliary linear branch is calculated as:

$$\mathbf{z}_i = \left(\boldsymbol{W}_i^{class}\right)^T \tilde{\boldsymbol{y}}_i \in \mathbb{R}^N. \quad (4)$$

In this way, a_i and $\boldsymbol{W}_i^{class}$ are those terms that need to be optimized, while the actual weights of the network are kept fixed. Then, the outputs $\mathbf{z}_i$ from the auxiliary classification layer can be directly used in loss function $J(\boldsymbol{W}_i^{class}, a_i; \boldsymbol{z}_i, \boldsymbol{t})$, where $J(\cdot)$ is the cross-entropy loss. We also propose to add an extra regularization term, denoted by $\Omega(a_i)$, in order to penalize the scaling factor when saturating the activation function. Specifically, after forward passing the linear part of the layer, $\tilde{\boldsymbol{u}}_i = |a_i|\boldsymbol{W}_i\boldsymbol{y}_{i-1} + \boldsymbol{b}_i$, we calculate $\Omega(a_i)$ as:

$$\Omega(a_i) = \frac{1}{N_i}\sum_{j=1}^{N_i} \max\{p_{min} - \tilde{u}_{ij}, \tilde{u}_{ij} - p_{max}, 0\}, \quad (5)$$

where p_{min} and p_{max} are the lower and upper bounds of the activation region and $\max\{\cdot\}$ denotes the maximum element in the set. Therefore, the final loss function $J'(\boldsymbol{W}_i^{class}, a_i; \boldsymbol{X}, \boldsymbol{y})$ is formulated as:

$$J'(\boldsymbol{W}_i^{class}, a_i; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{W}_i^{class}, a_i; \boldsymbol{X}, \boldsymbol{y}) + c\Omega(a_i), \quad (6)$$

where c is the weight for the relative contribution of the *vanishing gradient penalty*. Finally, scaling factor a_i and classification weights W_i^{class} are optimized using gradient descent:

$$\Delta a_i = -\eta\frac{\partial J}{\partial a_i} - \eta c\frac{\partial \Omega}{\partial a_i},\ \Delta \boldsymbol{W}_i^{class} = -\eta\frac{\partial J}{\partial \boldsymbol{W}_i^{class}}, \quad (7)$$

where η is the used learning rate.

After the optimization has been completed, the weights of i-th layer can be re-initialized using the optimized scaling factor a_i. It is worth noting that the classification weights are no longer needed after the initialization process is completed and can be discarded. Starting from the transmitter to receiver, all layers of both networks, from input to output, are iteratively initialized with the aforementioned procedure. It should be mentioned that during the initialization of the receiver, the proposed method takes into account the corruption that occurred in the fiber link and estimates the variance accordingly. After this process has been completed, the model is ready to be trained in an end-to-end fashion using regular back-propagation. The proposed method is presented schematically in Fig. 1.

IV. Experimental Results

We train the neural optical IM/DD system described in Section II in an end-to-end fashion using the RMSprop [31] optimizer for 1.5 millions iterations using mini-batches of 253 randomly generated 6-bit symbols. These symbols are encoded in one-hot vectors of size $M = 64$. For both the transmitter and receiver we use MLPs with two hidden layers (each one with 128 neurons). The size of the input layer of the transmitter is 64, equal to the size of the receiver's classification layer. Transmitter outputs 48 points of a time-series concatenated with another 10 sets of points by the serializer (as described in Section II) to construct the transmitted time-series before it passes through the optical channel. In turn, signal is deserialized and fed to the receiver PNN that consists also of 48-neuron in the input layer. We applied the proposed initialization method for 3 and 10 epochs for transmitter and receiver respectively using RMSprop optimizer. The learning rates of the transmitter and receiver are calculated according to $(\eta_r, \eta_t) = (\eta_f\eta, \eta)$, where η denotes the system's learning rate, which is set to 0.001. The $\eta_f = \{\eta_{sig}, \eta_{sin}\}$ denotes the different learning rates for each activation function, where $\eta_{sig} = 0.1$ is referred to sigmoid-based and $\eta_{sin} = 1$ to sinusoidal activations. Finally, for all different cases the weight for the vanishing gradient penalty term c is set to 0.6. The whole system is trained with a decaying learning rate starting from 10^{-4} to 10^{-6}.

We evaluated the proposed method in different fiber lengths using different neural IM/DD system configurations. More specifically, the models were trained and evaluated on ranges between 30 and 70 kilometers (km) employing different activation functions (sigmoid, photonic sigmoid and photonic sinusoidal) comparing the proposed method with traditionally used initialization schemes (Xavier and He). We report the average evaluation loss over 80,000 randomly generated 6-bit symbols in Table II. On the first column, the fiber length is reported and the rest of the columns present, from left to right, the following initialization schemes: a) Xavier initialization, b) first initializing the network with Xavier and then employing the proposed method, c) He initialization, and d) first initializing the network with He and then employing the proposed method.

As presented in Table II, the proposed method achieves impressive performance improvements in the sigmoid case with the evaluation loss reduced by about an order of magnitude. This behavior is observed in both cases, regardless of the employed initialization method, with the He method performing greater than the Xavier when combined with the proposed initialization method for fiber lengths longer than 30 km. The role of the initialization is also depicted in the overall performance and convergence of the network, since the proposed method constantly achieves lower evaluation loss compared to the baseline initialization approaches.

For the photonic sigmoid architecture, the contribution of the proposed approach is even greater when the model is first initialized using the He method. However, combining the proposed method with the Xavier method leads to the overall best results. It is worth noting that for longer distances, such as for 60 and 70 km, in which the corruption in which the signal is significantly higher, the proposed method reduces the evaluation loss by over 40%. For the photonic sinusoidal

TABLE II
EVALUATION LOSS FOR DIFFERENT FIBER LENGTHS

L	Xavier	+ Proposed	He	+ Proposed
Sigmoid				
30	7.38×10^{-3}	$\mathbf{5.65 \times 10^{-5}}$	9.90×10^{-4}	$\mathbf{8.20 \times 10^{-5}}$
40	5.15×10^{-3}	$\mathbf{3.41 \times 10^{-3}}$	3.26×10^{-3}	$\mathbf{3.68 \times 10^{-4}}$
50	1.11×10^{-2}	$\mathbf{4.79 \times 10^{-3}}$	5.30×10^{-3}	$\mathbf{1.39 \times 10^{-3}}$
60	7.38×10^{-3}	$\mathbf{3.06 \times 10^{-3}}$	7.66×10^{-3}	$\mathbf{2.77 \times 10^{-3}}$
70	6.94×10^{-3}	$\mathbf{5.85 \times 10^{-3}}$	6.76×10^{-3}	$\mathbf{5.42 \times 10^{-3}}$
Photonic Sigmoid				
30	2.31×10^{-4}	$\mathbf{3.14 \times 10^{-5}}$	1.23×10^{-3}	$\mathbf{6.86 \times 10^{-5}}$
40	1.63×10^{-3}	$\mathbf{2.21 \times 10^{-4}}$	3.04×10^{-3}	$\mathbf{2.74 \times 10^{-4}}$
50	2.93×10^{-3}	$\mathbf{1.31 \times 10^{-3}}$	5.06×10^{-3}	$\mathbf{1.70 \times 10^{-3}}$
60	6.49×10^{-3}	$\mathbf{1.97 \times 10^{-3}}$	8.01×10^{-3}	$\mathbf{4.08 \times 10^{-3}}$
70	6.38×10^{-3}	$\mathbf{2.34 \times 10^{-3}}$	8.27×10^{-3}	$\mathbf{5.03 \times 10^{-3}}$
Photonic Sinusoidal				
30	1.53×10^{-5}	$\mathbf{1.17 \times 10^{-5}}$	2.91×10^{-5}	$\mathbf{1.02 \times 10^{-5}}$
40	6.30×10^{-5}	$\mathbf{6.28 \times 10^{-5}}$	1.20×10^{-4}	$\mathbf{1.18 \times 10^{-4}}$
50	3.70×10^{-4}	$\mathbf{3.15 \times 10^{-4}}$	9.65×10^{-4}	$\mathbf{4.32 \times 10^{-4}}$
60	1.27×10^{-3}	$\mathbf{9.18 \times 10^{-4}}$	1.95×10^{-3}	$\mathbf{1.80 \times 10^{-3}}$
70	2.11×10^{-3}	$\mathbf{1.49 \times 10^{-3}}$	3.46×10^{-3}	$\mathbf{2.27 \times 10^{-3}}$

case, the proposed method still improves the performance, even though the improvements are lower. However, this is an expected behavior, since the sinusoidal photonic activation is closer to the behavior of the ReLU function, which is the activation function He initialization targets. Therefore, in all evaluated cases, i.e., different fiber lengths and activation functions, the proposed method improves the evaluation loss.

V. CONCLUSION

In this work, we presented a trainable initialization method for photonic neural IM/DD systems that takes into account easily saturated activation functions that are often used in PNN, data distribution and corruption that occurs in the signal in optical channels due to fiber dispersion and other noise sources. The experimental results, that include evaluation using different initialization schemes and fiber lengths, demonstrate that the proposed initialization method can significantly increase the system's performance, highlighting its effectiveness.

Acknowledgment: 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.

REFERENCES

[1] Y. LeCun *et al.*, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[3] S. Dörner *et al.*, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.

[4] G. Dabos *et al.*, "End-to-end deep learning with neuromorphic photonics," in *Integrated Optics: Devices, Materials, and Technologies XXV*, vol. 11689, International Society for Optics and Photonics. SPIE, 2021, pp. 56 – 66.

[5] N. Passalis *et al.*, "Training noise-resilient recurrent photonic networks for financial time series analysis," in *2020 Proc. of the European Signal Processing Conf. (EUSIPCO)*, 2021, pp. 1556–1560.

[6] G. Mourgias-Alexandris *et al.*, "A silicon photonic coherent neuron with 10gmac/sec processing line-rate," in *Proc. of the Optical Fiber Communications Conf. and Exhibition (OFC)*, 2021, pp. 1–3.

[7] B. Karanov *et al.*, "End-to-end deep learning of optical fiber communications," *Journal of Lightwave Technology*, vol. 36, no. 20, pp. 4843–4855, 2018.

[8] M. H. Eiselt *et al.*, "Direct detection solutions for 100g and beyond," in *Optical Fiber Communication Conf.* Optical Society of America, 2017, p. Tu3I.3.

[9] P. J. Freire *et al.*, "Performance versus complexity study of neural network equalizers in coherent optical systems," *Journal of Lightwave Technology*, vol. 39, no. 19, p. 6085–6096, Oct 2021.

[10] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 1–12.

[11] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, p. 73, 2011.

[12] Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, no. 7, p. 441, 2017.

[13] N. Pleros *et al.*, "Compute with light: Architectures, technologies and training models for neuromorphic photonic circuits," in *Proc. of the European Conf. on Optical Communication (ECOC)*, 2021, pp. 1–4.

[14] G. Giamougiannis *et al.*, "Silicon-integrated coherent neurons with 32gmac/sec/axon compute line-rates using eam-based input and weighting cells," in *Proc. of the European Conf. on Optical Communication (ECOC)*, 2021, pp. 1–4.

[15] M. Moralis-Pegios *et al.*, "Photonic neuromorphic computing: Architectures, technologies, and training models," in *2022 Optical Fiber Communications Conf. and Exhibition (OFC)*, 2022, pp. 01–03.

[16] X. Lin *et al.*, "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018.

[17] G. Mourgias-Alexandris *et al.*, "An all-optical neuron with sigmoid activation function," *Opt. Express*, vol. 27, no. 7, pp. 9620–9630, Apr 2019.

[18] G. Mourgias-Alexandris *et al.*, "l response-aware photonic neural network accelerators for high-speed inference through bandwidth-limited optics," *Opt. Express*, vol. 30, no. 7, pp. 10 664–10 671, Mar 2022.

[19] G. Mourgias-Alexandris *et al.*, "25gmac/sec/axon photonic neural networks with 7ghz bandwidth optics through channel response-aware training," in *2021 European Conf. on Optical Communication (ECOC)*, 2021, pp. 1–4.

[20] M. Moralis-Pegios *et al.*, "Neuromorphic silicon photonics and hardware-aware deep learning for high-speed inference," *Journal of Lightwave Technology*, pp. 1–1, 2022.

[21] N. Passalis *et al.*, "Training deep photonic convolutional neural networks with sinusoidal activations," *IEEE Trans. Emerging Topics in Computational Intelligence*, pp. 1–10, 2019.

[22] X. Glorot *et al.*, "Deep sparse rectifier neural networks," in *Proc. of the International Conf. on Artificial Intelligence and Statistics*. JMLR Workshop and Conf. Proc., 2011, pp. 315–323.

[23] R. Pascanu *et al.*, "On the difficulty of training recurrent neural networks," in *Proc. of the International Conf. on Machine Learning*, 2013, pp. 1310–1318.

[24] K. He *et al.*, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," 2015.

[25] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. of the International Conf. on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[26] A. N. Tait *et al.*, "Neuromorphic photonic networks using silicon photonic weight banks," *Scientific Reports*, vol. 7, no. 1, pp. 1–10, 2017.

[27] S. Pitris *et al.*, "O-band energy-efficient broadcast-friendly interconnection scheme with sipho mach-zehnder modulator (MZM) & arrayed waveguide grating router (AWGR)," in *Proc. of the Optical Fiber Communication Conf.* Optical Society of America, 2018, p. Th1G.5.

[28] G. P. Agrawal, *Fiber-optic communication systems*. John Wiley & Sons, 2012, vol. 222.

[29] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. of the IEEE Information Theory Workshop*, 2015, pp. 1–5.

[30] L. Paninski, "Estimation of entropy and mutual information," *Neural computation*, vol. 15, no. 6, pp. 1191–1253, 2003.

[31] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.