A robust, quantization-aware training method for photonic neural networks

A. Oikonomou[1], M. Kirtas[1][0000−0002−8670−0248],
N. Passalis[1][0000−0003−1177−9139], G. Mourgias-Alexandris[2][0000−0002−9646−3119],
M. Moralis-Pegios[2][0000−0002−9401−730X], N. Pleros[2][0000−0003−2931−4540], and
A. Tefas[1][0000−0003−1288−3667]

[1] Computational Intelligence and Deep Learning Group, AIIA Lab.
[2] Wireless and Photonic Systems and Networks Group
Dept. of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
{athoikgeo, eakirtas, passalis, mourgias, mmoralis,
npleros, tefas}@csd.auth.gr

Abstract. The computationally demanding nature of Deep Learning (DL) has fueled the research on neuromorphics due to their potential to provide high-speed and low energy hardware accelerators. To this end, neuromorphic photonics are increasingly gain attention since they can operate in very high frequencies with very low energy consumption. However, they also introduce new challenges in DL training and deployment. In this paper, we propose a novel training method that is able to compensate for quantization noise, which profoundly exists in photonic hardware due to analog-to-digital (ADC) and digital-to-analog (DAC) conversions, targeting photonic neural networks (PNNs) which employ easily saturated activation functions. The proposed method takes into account quantization during training, leading to significant performance improvements during the inference phase. We conduct evaluation experiments on both image classification and time-series analysis tasks, employing a wide range of existing photonic neuromorphic architectures. The evaluation experiments demonstrate the effectiveness of the proposed method when low-bit resolution photonic architectures are used, as well as its generalization ability.

Keywords: Photonic Neural Networks · Neuromorphic Computing · Neural Network Quantization

1 Introduction

Over the recent years, the applications that are using Deep Learning (DL) are constantly expanding both in industrial and academic communities since they are achieving state-of-the-art performance in complex tasks, such as image classification and time-series forecasting [17]. Despite the fact that DL can effectively tackle such demanding tasks, its application is often restricted because of its high computational cost. High-end hardware accelerators are required to achieve fast computational operations, such as matrix multiplication that occupies a significant fraction of operations in DL. This demanding nature of DL has fueled the research on low energy and ultra-fast hardware accelerators. Initially, Graphics Processing Units (GPUs) have been used to serve the high computational cost of the training and inference. Nowadays, energy consumption is an increasingly relevant issue [42] and more advanced technologies, such as Tensor Processing Units (TPUs) [13] and novel neuromorphic hardware architectures [11], are applied, achieving even higher frequency rates with lower power consumption.

Neuromorpic photonics is an upcoming and promising technology that has been increasingly gaining more attention in the academic communities since it is able to propagate optical signals in very high frequencies with extremely lower power consumption, employing them to provide the neuron's functionality [1, 3, 37]. To achieve this, there is a great variety of proposed hardware architectures that use only optical [8,41] and/or conjunctions of electro-optical hardware devices [19]. However, there are limitations that restrict the application of neuromorphic photonics in DL due to their unique nature. Although photonic hardware has great advantages in the development of materials and waveguide technologies [6], managing very fast analog processing and vector-matrix operations with ultra low energy and power consumption [41] in reference to its electronic counterparts, such implementations include analog-to-digital and digital-to-analog conversions significantly degrade bit-resolution [29,40]. Furthermore, most of the photonic architectures currently available face difficulties in deploying traditional activation functions that are typically used in DL, such as ReLU [9]. Instead, PNNs are usually implied on sinusoidal [34] and/or sigmoidal activations [23]. Therefore, training ANNs that are oriented to neuromorphic photonics should consider both the photonic activation function [1,25], and take into account the corruptions that exist due to the use of DACs and ADCs.

Typically, ADCs can be simulated through a quantization process that converts a continuous signal to a discrete one by mapping its continuous set to a finite set of discrete values [35]. This can be achieved by rounding and truncating the values of the input signal. Despite the fact that quantization techniques are widely studied by the DL community [12,16,18], they typically target large convolutional neural networks (CNNs) containing a great amount of surplus parameters with a minor contribution to the overall performance of the model [5, 44]. These large architectures are easily compressed, in contrast to smaller networks, such those currently developed for neuromorphic photonics, in which every single parameter has a great contribution on the final output of the model [12]. Furthermore, existing works mainly target dynamic quantization methods, which require extra parameters during inference, or focus on partially quantized models that ignore input and bias [10, 12]. These limitations, which are further exaggerated when high-slope photonic activations are used, dictate employing different training paradigms that take into account the actual physical implementation [22].

Indeed, neuromorphic photonics impose new challenges on DL models quantization, requiring the appropriately adaption of the existing methodologies to the unique limitations of photonic substrates, e.g., using smaller models. Furthermore, the quantization scheme applied in neuromorphics is a very simple uniform quantization because it depends on the DAC/ADC modules that quantize the signals equally and symmetrically [29,40]. This differs from the approaches traditionally used in trainable quantization schemes for DL models [32]. Finally, being able to operate on low-precision networks during the deployment can further improve the potential use of analog computing by lowering even more the energy consumption of the developed accelerators [28,39].

This work focuses on training PNNs while taking into account the quantization that occurs during the deployment, employing photonic activation functions. As has been shown, considering actual hardware limitations and corruptions during training can significantly improve the performance of the model during the deployment phase [26,27,33]. To this end, we propose an activation-agnostic, quantization-aware training method oriented for PNNs that enables us to effectively train models in lower precision without significant performance degradation. The proposed quantization-aware training method considers the input and model parameter variances during training and quantizes them accordingly. We evaluate the proposed method on two different photonic architectures used on two traditional image classification tasks applying multi layer perceptron (MLP) and CNNs, respectively, as well as on a challenging time-series forecasting task that involves high frequency financial time series using a state-of-the-art recurrent photonic architecture.

The rest of this paper is structured as follows. Section 2 provides the necessary background on photonic DL, while the proposed method is introduced and described in Section 3. Finally, the experimental evaluation is provided in Section 4, while the conclusion is drawn in Section 5

2 Background

Similarly to the software implemented ANNs, photonic ones are based on perceptron with the ultimate goal of approximating a function f^*. More precisely, the input signal of the photonic ANN is denoted as $\boldsymbol{x} \in \mathbb{R}^M$, where M represents the number of features. Each sample in the train data set is labeled with a vector $\boldsymbol{t} = \mathbf{1}_n \in \mathbb{R}^N$ where the n-th element equals to 1 and the other elements are 0 if it is a classification task (N denotes the number of classes) or a continuous vector $\boldsymbol{t} \in \mathbb{R}^N$ if it is a regression task (N denotes the number of regression targets). MLPs approximate f^* by using more than one layer, i.e., $f_n(...(f_2(f_1(\boldsymbol{x};\boldsymbol{\theta}_1)\boldsymbol{\theta}_2;)\boldsymbol{\theta}_n) = \boldsymbol{z}_n$ and learn the parameters $\boldsymbol{\theta}_i$ where $0 \leq i \leq n$ with $\boldsymbol{\theta}_i$ consisting of the weights $\boldsymbol{w}_i \in \mathbb{R}^{N_i \times M_i}$ and biases $\mathbf{b}_i \in \mathbb{R}^{N_i}$. Subsequently, each layer's output is denoted as $\boldsymbol{z}_i = f_i(\boldsymbol{y}_{i-1}) = \boldsymbol{w}_i\boldsymbol{y}_{i-1} + \boldsymbol{b}_i$. The output of the linear part of a neuron is fed to a non-linear function $g(\cdot)$, named activation function, to form the final output of the layer, $\boldsymbol{y}_i = g(\boldsymbol{z}_i)$

The training of an ANN is achieved by updating its parameters, using the backpropagation algorithm [14], aiming to minimize a loss function $J(\boldsymbol{y}, \boldsymbol{t})$, where

$\boldsymbol{t}$ represents the training labels and $\mathbf{y}$ the output of the network. Cross-entropy loss is often used in multi-class classification cases $J(\boldsymbol{y}, \boldsymbol{t}) = -\sum_{c=1}^{N} t_c \log y_c$.

Except for the feed-forward ANNs, in this paper we also employ a simple-to-apply recurrent neuromorphic photonic architecture. The applied recurrent architecture is benefited from the existing photonic feed-forward implementations [24, 38] while using a feedback loop. Following the above notation and the fact that the recurrent architectures accept sequential data as input, let $\mathbf{x}$ be a multidimensional time series, while let $\mathbf{x}_t \in \mathbb{R}^M$ denote M observations fed to the input 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 $\mathbf{y}_{t-1}^{(r)} \in \mathbb{R}^{N_r}$, which corresponds to the output of the N_r recurrent neurons at the previous time-step, is also weighted by the recurrent weights $\mathbf{w}_i^{(r)} \in \mathbb{R}^{N_r}$. The final weighted output of the i-th recurrent neuron is calculated as $\boldsymbol{u}_{ti}^{(r)} = {\mathbf{w}_i^{(in)}}^T \mathbf{x}_t + {\mathbf{w}_i^{(r)}}^T \mathbf{y}_{t-1}^{(r)}$. 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 $\boldsymbol{y}_{ti}^{(r)} = f(\boldsymbol{u}_{ti}^{(r)})$.

In this case of study two photonic activation functions are used. First, the photonic sigmoid activation function is defined as [24]:

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

in which the parameters $A_1 = 0.060, A_2 = 1.005, z_0 = 0.154$ and $d = 0.033$ are tuned to fit the experimental observations as implemented on real hardware devices [24].

Also, a photonic sinusoiudal activation function is applied on the experimental evaluations. The photonic layout corresponds to the employing a Mach-Zender Modulator device (MZM) [36] that converts the data into an optical signal along with a photodiode [2]. The formula of this photonic activation function is the following:

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

It is worth noting that because of the narrow range of the input domain these photonic activations have, training is even more difficult, since the networks tend to be easily saturated, leading to slower convergence or even halting the training.

3 Proposed Method

In this work, we propose a quantization-aware training framework that takes into account the quantization error arisen from DACs and ADCs modules in PNNs. In this way, we exploit the intrinsic ability of ANNs to resist to known noise sources when they are first trained to withstand them [25, 33]. In this way, the training procedure is adjusted on lower-precision signals, and consequently the quantization error is considered at the loss function and minimized through the optimization process. As a result, the networks trained in a quantization-aware fashion can significantly improve the accuracy during the inference process.

Under the proposed quantization-aware training framework, which is inspired and extends the quantization scheme in [12], every signal that is involved in the response of the i-th layer is first quantized in a specific floating range $[p^{(i)}_{min}, \dots, p^{(i)}_{max}]$. More specifically, it *simulates* the quantization process *during the forward pass*, which means that the input, model's parameters and activation values are stored as floating point numbers enabling us to perform backpropagation as usual. However, during the forward-pass quantization error ϵ is injected by deploying the rounding of quantization arithmetically in floating point. More precisely, the inputs and the model's parameters are quantized before forward-pass is applied to the layer. In turn, the linear output of the layer is quantized before it is fed to the photonic activation function. As a result, quantization divides the signal by the number of quantization levels in a range depending on the specific bit resolution.

First, every signal involved $p^{(i)}$ is converted to a bit representation by applying the function $Q : \mathbb{R} \rightarrow \mathbb{N}$ formulated as following:

$$p^{(i)}_q = Q(p^{(i)}, s^{(i)}_p, \zeta^{(i)}_p) = clip\left\{ \left\lfloor \frac{p^{(i)}}{s^{(i)}_p} + \zeta^{(i)}_p \right\rceil, q_{min}, q_{max} \right\} \in \mathbb{N}$$

where $p^{(i)} \in \mathbb{R}$, $p^{(i)}_q \in [0 \dots, 2^B - 1]$ and B denotes the bit resolution of the signal. Variables $s^{(i)}_p \in \mathbb{R}^+$ and $\zeta^{(i)}_p \in \mathbb{N}$ define the quantization parameters of the quantization function Q named *scale* and *zero-point* respectively. The *scale* value is typically represented in the software as a floating-point number and is calculated as follows:

$$s^{(i)}_p = \frac{p^{(i)}_{max} - p^{(i)}_{min}}{q_{max} - q_{min}} \in \mathbb{R}^+ \tag{3}$$

where $q_{min} \in \mathbb{N}^+$ and $q_{max} \in \mathbb{N}^+$ denote the range of an B-bit resolution (0 and $2^B - 1$ respectively) while $p^{(i)}_{max} \in \mathbb{R}$ and $p^{(i)}_{min} \in \mathbb{R}$ represents the working range, i.e., maximum and minimum, of a signal. In turn, the zero point is calculated:

$$\zeta^{(i)}_p = clip\left\{ \left\lfloor q_{min} - \frac{p^{(i)}_{min}}{s^{(i)}_p} \right\rceil, q_{min}, q_{max} \right\} \in \mathbb{N} \tag{4}$$

In contrast to [12], we convert $p^{(i)}_q \in [0 \dots, 2^B - 1]$ to discrete floating arithmetics $p^{(i)}_q \in [p^{(i)}_{min}, \dots, p^{(i)}_{min}]$ using a dequantization function $D : \mathbb{N} \rightarrow \mathbb{R}$ formulated as following:

$$p^{(i)}_f = D(p^{(i)}_q, s^{(i)}_p, \zeta^{(i)}_p) = s^{(i)}_p(p^{(i)}_q - \zeta^{(i)}_p) \in \mathbb{R} \tag{5}$$

Following the above notation the linear response of i-th layer is given by:

$$\boldsymbol{z}_f^{(i)} = Quant(\boldsymbol{w}_f^{(i)} \cdot \boldsymbol{y}_f^{(i-1)} + \boldsymbol{b}_f^{(i)}) \in \mathbb{R}^{N_i} \tag{6}$$

where $Quant(\boldsymbol{x})$ denotes process of quantization followed by dequantization of a vector or matrix $\boldsymbol{x} \in \mathbb{R}^{M_i}$, while $\boldsymbol{w}_f^{(i)} \in [w_{min}^{(i)}, \ldots, w_{max}^{(i)}]^{N_i \times M_i}$ and $\boldsymbol{b}_f^{(i)} \in [b_{min}^{(i)}, \ldots, b_{max}^{(i)}]^{N_i}$ denote the quantized weights and biases of i-th layer. Note that the $Quant(\boldsymbol{x})$ function is applied in an element-wise fashion.

Finally, the output $\boldsymbol{z}_f^{(i)}$ passes through the photonic activation function, $g(\cdot)$ of the neuron $\boldsymbol{y}_f^{(i)} = Quant(g(\boldsymbol{z}_f^{(i)}))$. In this way, all signals involved in the layer's output are distributed in a uniform floating range between $p_{min}^{(i)}$ and $p_{max}^{(i)}$ and they can be represented using B bits. Thus, the quantization error is propagated through the network as a noise signal that is taken into account during the training, and the network learns to be aware of it during the deployment. The proposed method is presented for feedforward networks, but without loss of generality it can be applied to RNN architectures as well. Consequently, we can represent the forward pass as a procedure that involves a quantization error that is introduced in the inputs, weights, and activations. We should note that during the training, the quantization effect is simulated while the backpropagation happens as usual, meaning that the original parameters are updated according to the propagated loss.

What significantly affects the amount of quantization error, both in training and inference, is the selected working range, i.e., minimum (p_{min}) and maximum (p_{max}) values, of a signal on which the scale of uniform buckets depends. To this end, we propose computing the exponential moving average (EMA) for p_{min} and p_{max}. We use EMA to eliminate outliers in vectors and matrices and smoothen the process of quantization during the training. In this way, the model becomes more robust to outlier values, especially at the beginning of the training process.

Since the distribution of every signal is transformed during the training, the preferable boundaries of a signal are calculated incrementally at every timestep t as following:

$$\tilde{p}_{max,t}^{(i)} = (\beta/t)p_{max,t}^{(i)} + (1 - (\beta/t))\tilde{p}_{max,t-1}^{(i)} \tag{7}$$

$$\tilde{p}_{min,t}^{(i)} = (\beta/t)p_{min,t}^{(i)} + (1 - (\beta/t))\tilde{p}_{min,t-1}^{(i)} \tag{8}$$

where t denotes the training iteration, β is the weighting parameter of the EMA and the update is applied for $t > \lceil b \rceil$. Note that we calculate the min and max values per vector and/or matrix. Therefore, we use the same min and max for the activations of the same layer, but different ones for different layers.

4 Experimental Evaluation

We evaluate the proposed method on two traditional image classification tasks, more specifically on MNIST [4] and CIFAR10 [15], using MLPs and CNNs respectively. Additionally, we employed RNN, to sufficiently cover all possible scenarios, on a challenging forecasting task using high frequency time series limit order book data (FI-2020) [31]. Photonic sigmoid and sinusoidal activation functions are employed in the aforementioned architectures, as given by equations 1 and 2 respectively. We evaluate the performance of the proposed method on different bit resolutions. Also, we compared the proposed method with a post training quantization approach in which the quantization is ignored during the training procedure and is applied during the inference. On the baseline approach, the p_{min} and p_{max} values for each parameter are calculated using the minimum and maximum values of each parameter vector or matrix. This corresponds to the case where the models are deployed directly in photonic hardware, as is currently done in most photonic DL approaches [7, 8, 21]. In the proposed method, the parameter β is set to 2.

4.1 Image classification

We report the average accuracy and the corresponding variance of the evaluation accuracy over 10 training runs in Table 1 and 2 for MNIST and CIFAR10 datasets respectively. More precisely, MNIST [4] dataset consists of handwritten digits, including 60,000 train samples and 10,000 test samples. The digits have been size-normalized, centered in a fixed size, and flattened to one dimension, leading to 784 features per sample. The input flattened images are fed to the first fully connected layer which consists of 10 neurons, then to the second fully connected layer which consists of 20 neurons, and finally to the output layer which consists of 10 neurons. The models are optimized for 100 epochs using the RMSProp optimizer [43] with a learning rate equal to 0.0001. The cross-entropy loss was used as the objective function, while mini-batches of 256 samples were used.

Table 1. Evaluating the proposed method on MNIST. Classification accuracy (%) is reported.

	Photonic Sinusoidal		**Photonic Sigmoid**	
Bits	Post training	Proposed	Post training	Proposed
8	90.12 ± 1.52	$\mathbf{91.21 \pm 0.49}$	91.02 ± 0.81	$\mathbf{91.17 \pm 0.33}$
6	86.93 ± 1.47	$\mathbf{91.04 \pm 0.51}$	87.01 ± 0.12	$\mathbf{91.02 \pm 0.29}$
4	09.95 ± 0.00	$\mathbf{90.26 \pm 0.77}$	09.34 ± 0.00	$\mathbf{90.20 \pm 0.39}$
2	09.97 ± 0.00	$\mathbf{67.63 \pm 2.28}$	09.98 ± 0.00	$\mathbf{61.00 \pm 1.95}$

As demonstrated in Table 1, the performance of the post training quantization method (columns 2 and 4) is collapsed when the bit resolution is lowered, especially in 2 and 4 bits. The proposed method, which takes into account the quantization during the training phase, can significantly improve the performance in low bit resolutions and resist the corruption occurring in post training quantization. This can be also attributed to the fact that the proposed method is taking into account the bounds of each signal incrementally, since it computes the EMA of the minimum and maximum value of each involved parameter. In this way, it can eliminate outliers that can lead to a wide range of buckets with barely any values. The proposed method (column 3 and 5) exceeds in terms of performance the post training method in all cases irrespective of the model's resolution and/or the applied photonic activation function.

The CIFAR10 dataset includes 50,000 images in the training set and 10,000 in the evaluation set with 32 × 32 color image samples containing one of the 10 object classes. The applied CNN consists of four convolutional layers followed by two linear layers. In more detail, the first two convolutional layers consist of 3×3 kernel size with 32 and 64 filters, followed by a 2×2 average pooling layer. Then, the other 2 convolutional layers are applied with 128 and 256 filters of size 3×3 followed by an 2×2 average pooling. Finally, the features that are extracted are flattened and fed to a linear layer that consists of 512 neurons followed by the final classification layer. The networks are optimized for 250 epochs using RMSProp optimizer [43] using mini-batches of 256 samples with a learning rate equal to 0.0001.

Table 2. Evaluating the proposed method on CIFAR10. Classification accuracy (%) is reported.

	Photonic Sinusoidal		**Photonic Sigmoid**	
Bits	Post training	Proposed	Post training	Proposed
8	15.22 ± 1.15	$\mathbf{67.64 \pm 1.24}$	16.39 ± 1.15	$\mathbf{66.23 \pm 1.23}$
6	15.10 ± 1.81	$\mathbf{66.56 \pm 1.38}$	15.76 ± 0.73	$\mathbf{66.50 \pm 1.61}$
4	16.62 ± 2.44	$\mathbf{29.48 \pm 10.43}$	16.38 ± 0.25	$\mathbf{65.25 \pm 1.96}$

In contrast to the MNIST case, in this experimental evaluation the performance of the baseline approach (columns 2 and 4) collapses even when 8-bit resolution is used. On the other hand, the proposed method (columns 3 and 6), similar to the fully connected case, can significantly resist to such collapse, since it outperforms the baseline approach in all cases. At 4-bit resolution the proposed method cannot fully recover the loss in the accuracy (for the photonic sinusoidal case), yet it can still lead to improvements. Therefore, we can safely draw the conclusion that the proposed can be generalized to CNNs, since it improves the performance of models during the inference for all the experiments conducted.

4.2 Forecasting financial time series analysis

Finally, the dataset that is used to evaluate the photonic recurrent architecture is a high frequency financial time series limit order book dataset (FI-2020) [31] that consists of more than 4,000,000 limit orders which come from 5 Finnish companies. The data processing scheme and evaluation procedure are described extensively in [30]. For the following experiments, splits 1 to 5 were used. The task of the forecast is to predict the movement of the future mid-price after the next 10 time steps which can go down, up or remain stationary.

The DL network that is used for the experiment consists of a recurrent photonic layer with 32 neurons, as described in Section 2. The output of the recurrent layer is fed to two fully-connected layers with the first fully connected layer consisting of 512 neurons and the second of 3 neurons. The length of the time series that is fed to the model is 10, which is the current and the past 9 timesteps. The model is optimized for 20 epochs with the RMSprop optimizer, and the learning rate is set to 10^{-4}.

Table 3. Evaluating the proposed method on FI2020. Cohen's κ metric is reported.

	Photonic Sinusoidal		**Photonic Sigmoid**	
Bits	Post training	Proposed	Post training	Proposed
8	0.0502 ± 0.0218	$\mathbf{0.1189 \pm 0.0105}$	0.0653 ± 0.0081	$\mathbf{0.1262 \pm 0.0072}$
6	0.0647 ± 0.0015	$\mathbf{0.1170 \pm 0.0115}$	0.0656 ± 0.0061	$\mathbf{0.1242 \pm 0.0132}$
4	0.0645 ± 0.0057	$\mathbf{0.1091 \pm 0.0210}$	0.0648 ± 0.0062	$\mathbf{0.1232 \pm 0.0517}$
2	0.0370 ± 0.0069	$\mathbf{0.0601 \pm 0.0031}$	0.0377 ± 0.0126	$\mathbf{0.0918 \pm 0.0033}$

The evaluation results are reported in Table 3. More precisely, we report the mean value of 5 splits using Cohen's κ metric [20] to evaluate the performance of the models since the dataset is extremely imbalanced. We observe that the benefits of the proposed method (columns 3 and 6) are crucial for the performance of the models during the inference phase since the post quantization training method (columns 2 and 4) is unable to sustain a reasonable performance. Indeed, the proposed method can significantly improve the inference accuracy irrespective of the photonic activation that is employed, highlighting once again its activation agnostic scope.

5 Conclusion

Neuromorphic photonics are an upcoming technology promising to overcome limitations that have become relevant over the recent years providing ultra-high speed and low energy consumption accelerators. At the same time, their application is hindered since it introduces new challenges on the training and deployment of DL, such as easily saturated activation functions and susceptible to different noise source ANNs, e.g., due to quantization. In this paper, we propose a novel activation-agnostic quantization-aware training method that is capable of compensating for quantization noise that arises from ADCs/DACs. As experimentally evaluated, the proposed method is capable of significantly improving the performance of low-bit resolution PNNs by considering quantization during the training. The proposed method builds a robust representation enabling us to decrease memory requirements and computational cost by lowering the bit resolution without significant performance degradation. The proposed method is evaluated on both image classification and time-series analysis tasks, employing a wide range of photonic architectures, outperforming the evaluated baselines.

Acknowledgements The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.), Greece under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 4233)

References

1. Dabos, G., Mourgias-Alexandris, G., Totovic, A., Kirtas, M., Passalis, N., Tefas, A., Pleros, N.: End-to-end deep learning with neuromorphic photonics. In: Integrated Optics: Devices, Materials, and Technologies XXV. vol. 11689, p. 116890I. Int. Society for Optics and Photonics (2021)
2. Danial, L., Wainstein, N., Kraus, S., Kvatinsky, S.: Breaking through the speed-power-accuracy tradeoff in adcs using a memristive neuromorphic architecture. IEEE Trans. on Emerging Topics in Computational Intelligence **2**(5), 396–409 (2018)
3. De Marinis, L., Cococcioni, M., Castoldi, P., Andriolli, N.: Photonic neural networks: A survey. IEEE Access **7**, 175827–175841 (2019)
4. Deng, L.: The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012)
5. Esser, S.K., McKinstry, J.L., Bablani, D., Appuswamy, R., Modha, D.S.: Learned step size quantization (2020)
6. Feldmann, J., Youngblood, N., Wright, C., Bhaskaran, H., Pernice, W.: All-optical spiking neurosynaptic networks with self-learning capabilities. Nature **569**(7755), 208–214 (2019)
7. Feldmann, J., Youngblood, N., Karpov, M., Gehring, H., Li, X., Stappers, M., Le Gallo, M., Fu, X., Lukashchuk, A., Raja, A.S., et al.: Parallel convolutional processing using an integrated photonic tensor core. Nature **589**(7840), 52–58 (2021)
8. Giamougiannis, G., Tsakyridis, A., Mourgias-Alexandris, G., Moralis-Pegios, M., Totovic, A., Dabos, G., Passalis, N., Kirtas, M., Bamiedakis, N., Tefas, A., Lazovsky, D., Pleros, N.: Silicon-integrated coherent neurons with 32gmac/sec/axon compute line-rates using eam-based input and weighting cells. In: Proc. European Conf. on Optical Communication (ECOC). pp. 1–4 (2021)
9. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proc. Int. Conf. on Computer Vision. pp. 1026–1034 (2015)
10. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations. The Journal of Machine Learning Research **18**(1), 6869–6898 (2017)
11. Indiveri, G., Linares-Barranco, B., Hamilton, T.J., Van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.C., Dudek, P., Häfliger, P., Renaud, S., et al.: Neuromorphic silicon neuron circuits. Frontiers in Neuroscience **5**, 73 (2011)
12. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition pp. 2704–2713 (2018)

13. Jouppi, N.P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., et al.: In-datacenter performance analysis of a tensor processing unit. In: Proc. Annual Int. Symposium on Computer Architecture. pp. 1–12 (2017)
14. KELLEY, H.J.: Gradient theory of optimal flight paths. ARS Journal **30**(10), 947–954 (1960)
15. Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research) http://www.cs.toronto.edu/ kriz/cifar.html
16. Kulkarni, U., Meena, S., Gurlahosur, S.V., Bhogar, G.: Quantization friendly mobilenet (qf-mobilenet) architecture for vision based applications on embedded platforms. Neural Networks **136**, 28–39 (2021)
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
18. Lee, D., Wang, D., Yang, Y., Deng, L., Zhao, G., Li, G.: Qttnet: Quantized tensor train neural networks for 3d object and video recognition. Neural Networks (2021)
19. Lin, X., Rivenson, Y., Yardimci, N.T., Veli, M., Luo, Y., Jarrahi, M., Ozcan, A.: All-optical machine learning using diffractive deep neural networks. Science **361**(6406), 1004–1008 (2018)
20. McHugh, M.L.: Interrater reliability: the kappa statistic. Biochemia medica **22**(3), 276–282 (2012)
21. Miscuglio, M., Sorger, V.J.: Photonic tensor cores for machine learning. Applied Physics Reviews **7**(3), 31404 (2020)
22. Mourgias-Alexandris, G., Moralis-Pegios, M., Tsakyridis, A., Passalis, N., Kirtas, M., Tefas, A., Rutirawut, T., Gardes, F., Pleros, N.: Channel response-aware photonic neural network accelerators for high-speed inference through bandwidth-limited optics. Optics Express **30**(7), 10664–10671 (2022)
23. Mourgias-Alexandris, G., Tsakyridis, A., Passalis, N., Tefas, A., Vyrsokinos, K., Pleros, N.: An all-optical neuron with sigmoid activation function. Optics express **27**(7), 9620–9630 (2019)
24. Mourgias-Alexandris, G., Tsakyridis, A., Passalis, N., Tefas, A., Vyrsokinos, K., Pleros, N.: An all-optical neuron with sigmoid activation function. Opt. Express **27**(7), 9620–9630 (2019)
25. Mourgias-Alexandris, G., Moralis-Pegios, M., Simos, S., Dabos, G., Passalis, N., Kirtas, M., Rutirawut, T., Gardes, F.Y., Tefas, A., Pleros, N.: A silicon photonic coherent neuron with 10gmac/sec processing line-rate. In: Proc. Optical Fiber Communications Conf. and Exhibition. pp. 1–3 (2021)
26. Mourgias-Alexandris, G., Moralis-Pegios, M., Simos, S., Dabos, G., Passalis, N., Kirtas, M., Rutirawut, T., Gardes, F.Y., Tefas, A., Pleros, N.: A silicon photonic coherent neuron with 10gmac/sec processing line-rate. In: Proc. Optical Fiber Communications Conf. and Exhibition (OFC). pp. 1–3 (2021)
27. Mourgias-Alexandris, G., Tsakyridis, A., Passalis, N., Kirtas, M., Tefas, A., Rutirawut, T., Gardes, F.Y., Pleros, N., Moralis-Pegios, M.: 25gmac/sec/axon photonic neural networks with 7ghz bandwidth optics through channel response-aware training. In: Proc. European Conf. on Optical Communication (ECOC). pp. 1–4 (2021)
28. Murmann, B.: Mixed-Signal Computing for Deep Neural Network Inference. IEEE Trans. on Very Large Scale Integration (VLSI) Systems **29**(1), 3–13 (2021)
29. Nahmias, M.A., de Lima, T.F., Tait, A.N., Peng, H.T., Shastri, B.J., Prucnal, P.R.: Photonic multiply-accumulate operations for neural networks. IEEE Journal of Selected Topics in Quantum Electronics **26**(1), 1–18 (2020)

30. Nousi, P., et al.: Machine learning for forecasting mid-price movements using limit order book data. IEEE Access **7**, 64722–64736 (2019)
31. Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. Journal of Forecasting **37**(8), 852–866 (2018)
32. Park, E., Ahn, J., Yoo, S.: Weighted-entropy-based quantization for deep neural networks. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 7197–7205 (2017)
33. Passalis, N., Kirtas, M., Mourgias-Alexandris, G., Dabos, G., Pleros, N., Tefas, A.: Training noise-resilient recurrent photonic networks for financial time series analysis. In: Proc. 28th European Signal Processing Conf. pp. 1556–1560 (2021)
34. Passalis, N., Mourgias-Alexandris, G., Tsakyridis, A., Pleros, N., Tefas, A.: Training deep photonic convolutional neural networks with sinusoidal activations. IEEE Trans. on Emerging Topics in Computational Intelligence (2019)
35. Pearson, C.: High-speed, analog-to-digital converter basics. Texas Instruments Application Report, SLAA510 (2011)
36. Pitris, S., Mitsolidou, C., Alexoudi, T., Pérez-Galacho, D., Vivien, L., Baudot, C., Heyn, P.D., Campenhout, J.V., Marris-Morini, D., Pleros, N.: O-band energy-efficient broadcast-friendly interconnection scheme with sipho mach-zehnder modulator (mzm) & arrayed waveguide grating router (awgr). In: Proc. Optical Fiber Communication Conf. Optical Society of America (2018)
37. Pleros, N., Moralis-Pegios, M., Totovic, A., Dabos, G., Tsakyridis, A., Giamougiannis, G., Mourgias-Alexandris, G., Passalis, N., Kirtas, M., Tefas, A.: Compute with light: Architectures, technologies and training models for neuromorphic photonic circuits. In: Proc. European Conf. on Optical Communication (ECOC). pp. 1–4 (2021)
38. Rosenbluth, D., Kravtsov, K., Fok, M.P., Prucnal, P.R.: A high performance photonic pulse processing device. Opt. Express **17**(25), 22767–22772 (Dec 2009)
39. Sarpeshkar, R.: Analog versus digital: extrapolating from electronics to neurobiology. Neural computation **10**(7), 1601–1638 (1998)
40. Shastri, B.J., Tait, A.N., de Lima, T.F., Pernice, W.H., Bhaskaran, H., Wright, C.D., Prucnal, P.R.: Photonics for artificial intelligence and neuromorphic computing. Nature Photonics **15**(2), 102–114 (2021)
41. Shen, Y., Harris, N.C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D., et al.: Deep learning with coherent nanophotonic circuits. Nature Photonics **11**(7), 441 (2017)
42. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp. arXiv preprint arXiv:1906.02243 (2019)
43. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4**(2), 26–31 (2012)
44. Wu, J., Leng, C., Wang, Y., Hu, Q., Cheng, J.: Quantized convolutional neural networks for mobile devices. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. 4820–4828 (2016)