Neural Computing and Applications (2022) 34:15589–15601
https://doi.org/10.1007/s00521-022-07243-z

ORIGINAL ARTICLE

Photonic-aware neural networks

Emilio Paolini[1,2,3] · Lorenzo De Marinis[1] · Marco Cococcioni[4] · Luca Valcarenghi[1] · Luca Maggiani[3] · Nicola Andriolli[2]

Received: 18 January 2022 / Accepted: 29 March 2022 / Published online: 25 April 2022

Abstract
Photonics-based neural networks promise to outperform electronic counterparts, accelerating neural network computations while reducing power consumption and footprint. However, these solutions suffer from physical layer constraints arising from the underlying analog photonic hardware, impacting the resolution of computations (in terms of effective number of bits), requiring the use of positive-valued inputs, and imposing limitations in the fan-in and in the size of convolutional kernels. To abstract these constraints, in this paper we introduce the concept of Photonic-Aware Neural Network (PANN) architectures, i.e., deep neural network models aware of the photonic hardware constraints. Then, we devise PANN training schemes resorting to quantization strategies aimed to obtain the required neural network parameters in the fixed-point domain, compliant with the limited resolution of the underlying hardware. We finally carry out extensive simulations exploiting PANNs in image classification tasks on well-known datasets (MNIST, Fashion-MNIST, and Cifar-10) with varying bitwidths (i.e., 2, 4, and 6 bits). We consider two kernel sizes and two pooling schemes for each PANN model, exploiting 2×2 and 3×3 convolutional kernels, and max and average pooling, the latter more amenable to an optical implementation. 3×3 kernels perform better than 2×2 counterparts, while max and average pooling provide comparable results, with the latter performing better on MNIST and Cifar-10. The accuracy degradation due to the photonic hardware constraints is quite limited, especially on MNIST and Fashion-MNIST, demonstrating the feasibility of PANN approaches on computer vision tasks.

Keywords Photonic neural networks · Analog computations · Effective number of bits · Quantization

1 Introduction

The recent advances in Machine Learning (ML), and specifically in Deep Learning (DL), have undoubtedly driven the current success of artificial intelligence. ML and DL models are deployed in an ever increasing number of applications, ranging from computer vision [1, 2] to speech recognition [3, 4] to fraud detection [5] and many others [6].

DL relies on Deep Neural Networks (DNNs), i.e., structures whose basic element is vaguely inspired by

✉ Emilio Paolini
emilio.paolini@santannapisa.it

Lorenzo De Marinis
lorenzo.demarinis@santannapisa.it

Marco Cococcioni
marco.cococcioni@unipi.it

Luca Valcarenghi
luca.valcarenghi@santannapisa.it

Luca Maggiani
luca.maggiani@sma-rty.com

Nicola Andriolli
nicola.andriolli@ieiit.cnr.it

[1] Scuola Superiore Sant'Anna, 56124 Pisa, Italy

[2] Institute of Electronics, Computer and Telecommunication Engineering (CNR-IEIIT), National Research Council of Italy, 56122 Pisa, Italy

[3] Sma-RTy Italia Srl, 20061 Carugate, Italy

[4] Department of Information Engineering, University of Pisa, 56122 Pisa, Italy

biological neurons. Neural networks have been traditionally implemented in electronic platforms (both CPUs and GPUs) based on the von Neumann architecture [7]. The high degree of programmability and the advancements in terms of processing capabilities of digital electronic architectures of the last decades enabled the great achievements of DNNs [8]. However, with the end of Moore's law and the exponential increase of DNN model complexity (at a pace of doubling every 3.4 months [9]), electronics is failing to keep up with the processing speed and energy efficiency required for large-scale deployment of complex models [10, 11].

Many research efforts are, therefore, investigating alternative hardware solutions for the acceleration of DNNs [12, 13]. In this context, photonics-based solutions attracted a lot of interest with the promise of outperforming electronic counterparts in speed, power consumption, and computing density [14]. Over the last few years, several Photonic Neural Network (PNN) architectures have been proposed exploiting integrated, fiber or free-space optics with the aim of accelerating DNN inference with analog computations in the photonic domain [15]. In [14], the authors highlight the advantages of performing multiply-accumulate (MAC) operations in photonics in terms of energy ($> 10^2$), speed ($> 10^3$), and computing density ($> 10^2$). Photonics has also the potential to outperform high-speed GPUs in performing convolutions with a lower power consumption [16].

Furthermore, the photonic implementation of the non-linearities needed in the activation functions of neurons has been recently investigated [17, 18], in some cases exploiting an electro-optic hardware platform [19, 20].

In the context of PNNs, the need of software frameworks for simulating training and inference operations of photonic neuromorphic architectures has been soon recognized. For this purpose, specialized frameworks, such as neuroptica [21] and neurophox [22], have been developed. These tools aim to emulate and train PNNs based on Mach–Zehnder interferometer (MZI) meshes. In particular, neuroptica provides several levels of abstractions, from the direct control of the MZI phase shifters, to the training of stacked structures using a Keras-like API. On the other hand, neurophox allows to train these chips using the Haar random technique developed in [23]. These simulators are, therefore, powerful tools, albeit specifically aimed to develop photonic accelerators based on MZI meshes.

In this paper, we focus on the abstraction of the physical layer constraints arising from generic analog photonic hardware, namely the limited resolution of the computations (in terms of effective number of bits), the requirement to use positive-valued inputs, and the limitations in the fan-in and in the size of convolutional kernels. Based on these constraints, (i) we introduce the concept of *Photonic-Aware Neural Network (PANN) architectures*, developing photonic-hardware compliant DNN models exploiting the Larq library [24]; (ii) we devise *PANN training schemes* resorting to quantization strategies aimed to obtain suited neural network parameters. The performance of PANNs is then assessed in a case study concerning image classification on well-kwown datasets, demonstrating the limited accuracy degradation due to the constraints coming from the use of photonic hardware.

The remainder of this paper is structured as follows: in Sect. 2, we review the main photonic architectures aimed at DNN inference acceleration, with a focus on integrated approaches. Section 3 firstly highlights the constraints arising from the adoption of photonic hardware. Afterward, a training-to-inference strategy is discussed, as well as the developed photonic-aware DNN models. Section 4 presents and discusses the image classification results obtained on the different datasets. Finally, Section 5 concludes the paper.

2 Photonics for neural networks: state of the art

A recurrent research theme at the border of optics and computing fields is the concept of a photonics-based computer [25, 26], aimed to overcome, through optics the speed, the energy consumption and the computing density limits imposed by electronics. While some notable results have been achieved at the beginning of the century [27–29], the interest in developing a digital optical computer based on logic gates has then faded. Despite the lower speed of electronics, the level of integration and energy efficiency enabled by CMOS process advancements overcame Photonic Integrated Circuits (PIC) in performing logic operations.

However, in recent years, following the breakthrough of deep learning [30], a significant research effort was put to explore non-conventional architectures to reduce the hardware resources and the energy necessary to run DNNs [31–33]. In this scenario, photonics gained a renewed interest as an alternative platform to implement analog neuromorphic functionalities [14, 15, 34]. The goal of optical neuromorphic processors is not to replace digital computers, but to enable analog computations with high bandwidth, low latency, and high energy efficiency [35].

In this paper, we focus on photonic devices for accelerating deep learning inference. These photonic engines rely on the inherent parallelism and speed of optics to perform DNN computations (e.g., matrix–vector multiplications and pooling). As an example of the computing capacity enabled by photonic devices, 11 Tera-operations

per second have been recently demonstrated using a fiber-based approach [36]. By pushing on integrated photonics solutions, optical processors promise also to significantly lower the power consumption for DNN computations, while increasing at the same time the footprint efficiency [37]. To unleash the potential of a drastic reduction in power consumption, several photonic solutions leverage passive components to implement weights, i.e., not requiring energy besides input generation and output acquisition. These solutions have the drawback of a limited speed at which weights can vary, in the hundreds or even tens of kHz range. Nevertheless, in weight-sharing architectures such as convolutional neural networks, this is not a heavy constraint as these parameters are slowly changing. In the following, we report a few photonic implementations of deep learning inference accelerators, with a focus on integrated solutions.

The coherent matrix multiplier depicted in Fig. 1a is a milestone in this field [38]. The silicon photonics-based device exploits a mesh of MZI to perform matrix multiplications on an input vector of coherent lightwaves. This is done by physically implementing the singular value decomposition theorem. In terms of DNN, this device implements a layer of fully connected (FC) neurons and thus, stacking several devices, a FC-DNN can be obtained. In the demonstration, however, the nonlinear activation function was emulated in software. Another experimentally validated coherent approach is shown in Fig. 1b, namely the optical linear algebra unit (OLAU). The OLAU basic element is composed of four MZIs in the form of a dual in-phase and quadrature (IQ) modulator [39], typically used in optical communications, composed of two MZIs with a phase shifter at one MZI output, as highlighted in Fig. 1b. In the OLAU each dual IQ modulator implements two input-weight multiplications, whose results are sent to optical 3dB combiners to perform the accumulation. In this way, the OLAU carries out matrix–vector multiplications in a distributed manner and implements again an FC layer. Despite a great potential, the actual scalability of these devices is still limited due to impairments and losses [40].

Other solutions rely on Wavelength Division Multiplexing (WDM), i.e., the use of multiple channels routed on the same waveguides at different wavelengths. The strength of these architectures is that multiple wavelengths, and thus multiple inputs, can be broadcast to multiple neurons. In this context, the architecture exploiting a microring resonator (MRR) bank recently gained a lot of attention as several proof-of-concept PICs have been fabricated with this approach. These solutions exploit resonant structures, the MRRs, to selectively weigh the different wavelengths within the same waveguide, as exemplified in Fig. 1c. The result of several multiply-and-accumulate operations is encoded in the photocurrent of a balanced photodetector placed after the MRR bank. Thanks to the hybrid photonic–electronic approach, a nonlinearity can be applied to the photocurrent, modulating the photonic neuron output. An MRR-bank-based PIC has been recently used for compensating fiber nonlinearities in a long-haul transmission experiment [41]. On the other hand, in [42] the use of a Semiconductor Optical Amplifier (SOA)-based cross-connect to perform weighing in a WDM architectures is proposed, as sketched in Fig. 1d. The experimental demonstration has been carried out with an Indium Phosphide (InP)-based photonic integrated device working on the iris dataset and still exploiting digital electronic hardware for the nonlinear activation fuction. Nonetheless, the strength of this approach relies on the fact that SOAs can inherently compensate for optical losses and exhibit all-optical nonlinearities.

Focusing on serial architectures, Fig. 1e reports a photonic electronic multiply-accumulate neuron [43]. This architecture relies on an hybrid opto-electronic approach where multiplications are performed at high speed in the optical domain, while the accumulation is carried out by an analog electrical frontend. Two high-speed MZIs are employed to impress inputs and weights on an incoming lightwave. The modulated signals are received in a balanced photodetector whose photocurrent encodes the multiplication result. Successive results are accumulated as a charge onto a capacitance and ultimately read by an analog-to-digital converter with a nonlinear input–output characteristic, so that the nonlinear activation function is inherently applied.

3 Photonic-aware neural networks

In this section, we outline the process adopted to develop and train PANNs. In the first part, we describe the main constraints derived from the use of analog optical technologies. In the second part, we report the solutions adopted to translate hardware limits in software with a focus on the DNN training-to-inference strategy. Finally, we present the developed DNN models and the computer vision datasets used to assess their performance. We call the devised DNNs *PANN Architectures* and the method for obtaining the DNN parameters *PANN Training*.

3.1 Limitations due to the photonic hardware

As surveyed in Sect. 2, photonic engines are analog processors that perform matrix–vector multiplications at high speed and low power consumption, while exploiting integrated photonic solutions to reach unprecedented computing densities [14]. Although in principle analog values can vary in a continuous set of values, noise and distortions

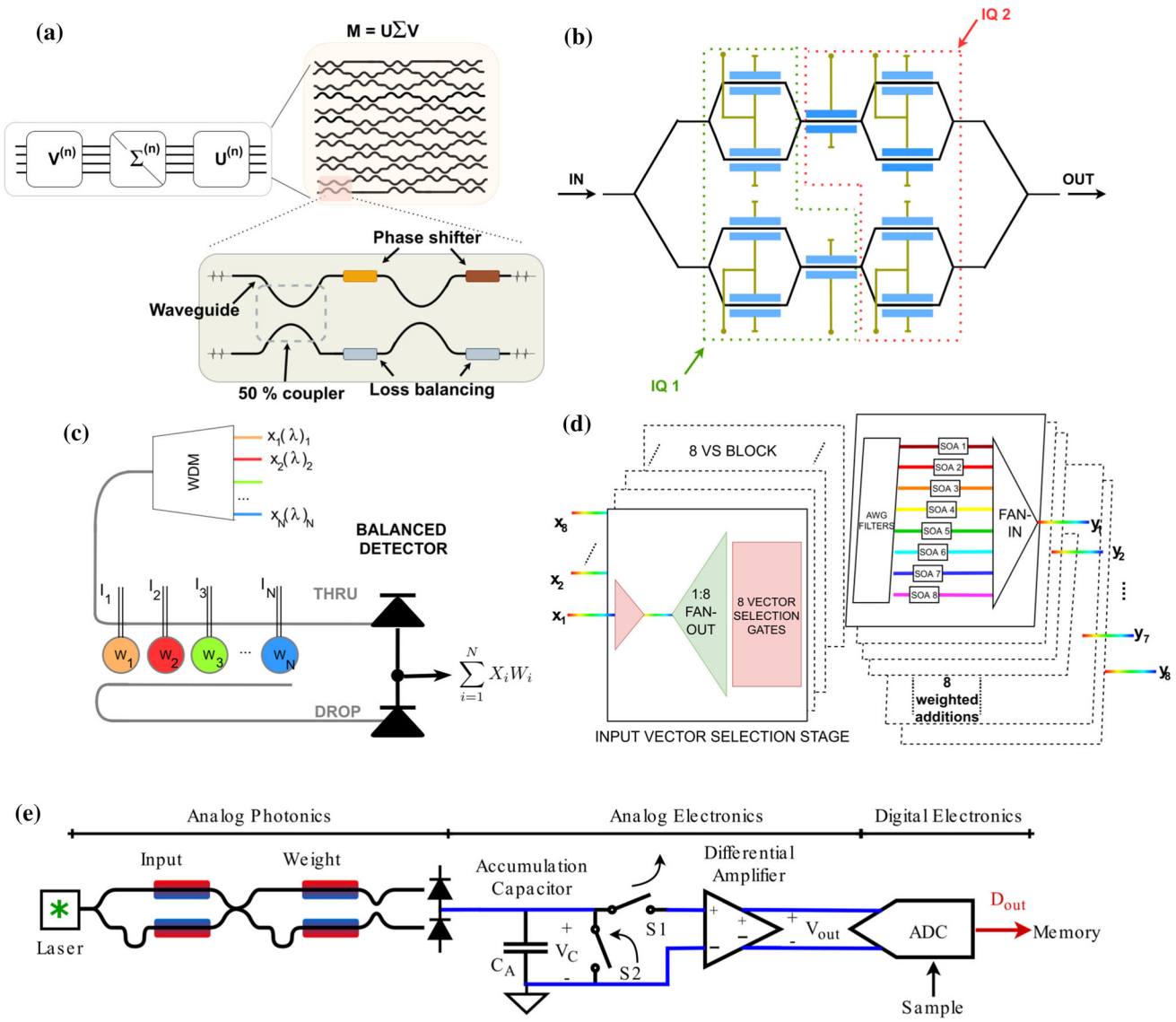


Fig. 1 Photonic architectures for optical matrix–vector multiplications. **a** Coherent matrix multiplier. **b** Coherent dual IQ modulator, acting as a basic element of an optical linear algebra unit. **c** Microring Resonator (MRR) weight bank. **d** InP SOA-based cross-connect. **e** Hybrid photonic–electronic multiply-accumulate neuron

prevent distinguishing different values with infinite precision, limiting the resolution of analog computations. Indeed, different values of analog signals can be distinguished only if they are "far enough", meaning that their distance cannot be closed by noise fluctuations. Furthermore, devices such as photonic neuromorphic engines are characterized by essentially constant noise intervals [44], meaning that the noise does not depend on the represented values. This translates into the fact that analog engines can distinguish only a finite number of different equally spaced levels. In this scenario, the Effective Number of Bits (ENOB) can be used to assess the equivalent resolution of analog signals, relating the number of distinguishable values into the corresponding number of bits needed for digital storage. For instance, if 16 levels can be distinguished, the signal has an ENOB of 4. For photonic neural networks, the typical bit resolution goes well below classical 32-bit floats down to very small values (i.e., ≤ 6 bits [14, 42, 43, 45]).

An additional limitation imposed by photonic hardware concerns the inputs of the PANN architecture. Indeed, in most configurations inputs are required to be positive-valued, as they are encoded in the intensity of optical signals. This impacts the normalization function of the input data, as well as the activation function used in each neuron that should have positive-only outputs, such as the ReLU [46] or the Sigmoid [47]. Another constraint on the input side of photonic neurons is related to the maximum number of

inputs (i.e., fan-in) to each neuron. Different photonic architectures are characterized by different constraints, as for instance MZI meshes can perform block operations at the expense of several electro-optical conversions, while in [43] the constraint arises from the electronic accumulation phase: the maximum number of inputs is about 200 for these implementations.

A final limitation due to the underlying hardware concerns the maximum kernel size in convolutional layers. Given the current experimentally validated photonic convolutional kernel implementations [40–42], the maximum kernel size is equal to 3×3 elements.

The main constraints imposed by the photonic hardware are summarized in Table 1.

3.2 Photonic-aware neural network computations

As pointed out in Sect. 3.1, noise and distortions limit the resolution of analog computations, allowing to distinguish a limited set of equally spaced-levels, leading to a very coarse bit resolution (i.e., ≤ 6 bits). For this reason, the floating point type, typically used for classical neural network computation, cannot be exploited to emulate photonic hardware.

To overcome this issue, in this paper, we propose to exploit reduced-precision fixed-point type for PANN inference and present a suited training approach. Indeed, the direct quantization of the weights computed using floats typically significantly reduces the accuracy of the obtained DNN [48, 49]. Dealing with this aspect, many approaches in the literature allow to use equally spaced types in neural networks [50–53]. All these strategies perform the training phase using the floating point type, however they take into account that the inference phase will be carried out using low bitwidth equally spaced types, consequently adjusting the weights. In particular, Rastegari et al. [51], Courbariaux and Bengio [52] implement binary operations in neural networks to achieve better performance at the expense of a very coarse granularity of inputs and weights, while [53] extends those works to equally spaced types with arbitrary precision by introducing a bitwidth-dependent quantization function.

Table 1 Summary of the PANN constraints

Photonic constraints	Value
Equally spaced levels	$\leq$ 6 bits
Inputs	Positive
Neuron fan-in	$\leq$ 200
Kernel size	$\leq 3 \times 3$

We now focus on training DNNs taking into account the resolution constraints from the photonic hardware. The approach presented in this paper aims to emulate the underlying photonic architecture, satisfying the bitwidth constraint (i.e., the ENOB) by exploiting quantized weights in the fixed-point domain.

The behavior of the quantization process carried out during the training phase is reported in Fig. 2. The input quantizer and the kernel quantizer describe the way of quantizing the incoming inputs and weights, respectively. A quantized layer computes the activation y as:

$$y = \sigma(f(q_{\text{kernel}}(w), q_{\text{input}}(x)) + b)$$

with full precision weights w, arbitrary precision input x, layer operation f, output σ, and bias b.

A very important aspect is the operation of the kernel quantizer $q_{\text{kernel}}(w)$ during the training phase. The concept of *latent weights* is introduced aside quantized weights to be used by the photonic hardware [54]. Basically, latent weights are a full precision copy of the weights used throughout the training process. Their purpose is to accumulate the small changes from the gradients without loss of precision. During the forward pass, a quantized version of these weights is used. When the vector of updates for the weights is obtained, the latent weights are updated instead of the quantized weights. Once the model is trained, only the quantized weights are used for inference.

Besides weights, we need also to consider the derivation of the quantization function during back-propagation. Being discrete-valued, its derivative is 0 almost everywhere: thus, its gradient would stop the network from learning. To overcome this problem, the pseudo-gradient method is used for optimizing weights during back-propagation [55]. One of the most commonly used pseudo-gradients is the Straight-Through Estimator (STE), defined as:

$$\frac{\partial q_{\text{kernel}}(w)}{\partial w} = \begin{cases} 1 & |w| \leq 1, \\ 0 & \text{otherwise} \end{cases}$$

Essentially, it ignores the derivative of the quantization function, and the incoming gradient is evaluated as if the function was a clipped identity function.

Regarding the inputs, they are quantized on the same number of bits used for weights. We implement the proposed PANN training-to-inference strategy by relying on the DoReFa quantizers implemented in Larq [24, 53], since it provides a convenient way to flexibly define the bitwidths on both inputs and weights.

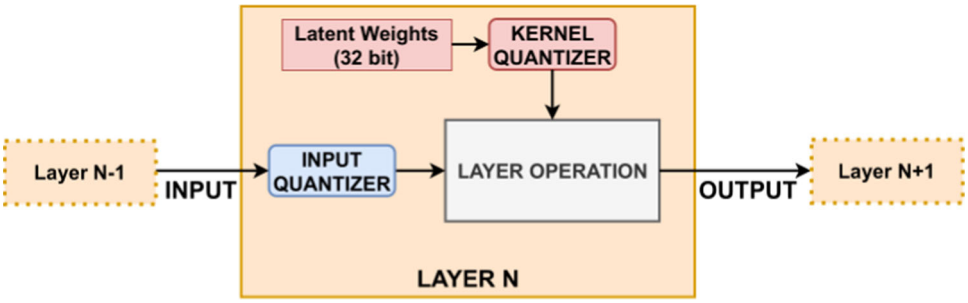


Fig. 2 PANN training. A layer is defined along with an input quantizer and a kernel quantizer

3.3 Datasets and models

After introducing the photonic constraints and the quantization strategies, we focus on the datasets and models used in the experiments. In particular, three common benchmark datasets are used:

- *MNIST* [56]: it is a dataset composed of 28×28 8-bit grayscale images containing handwritten digits from 0 to 9. The training set is made of 60, 000 images, while the test set is composed of 10, 000 samples.
- *Fashion-MNIST* [57]: it is an alternative to MNIST dataset, containing images of Zalando's articles. The structure of the images and the number of samples of this dataset are the same as the MNIST.
- *Cifar-10* [58]: it is a dataset made of 32×32 RGB images from 10 different classes (i.e., airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck). There are 50, 000 images in the training set and 10, 000 images in the test set.

All the input images for the three datasets, already satisfying the photonic constraint regarding positive-valued inputs, have been pre-processed to normalize all samples to the interval [0, 1]. This, however, leads to input values in floating-point domain (i.e., 32-bit numbers), not satisfying the photonic hardware constraints. For this reason, the developed PANN architectures include an input quantizer in the first layer. This differs from traditional quantization approaches, that typically keep the inputs in full-precision [53].

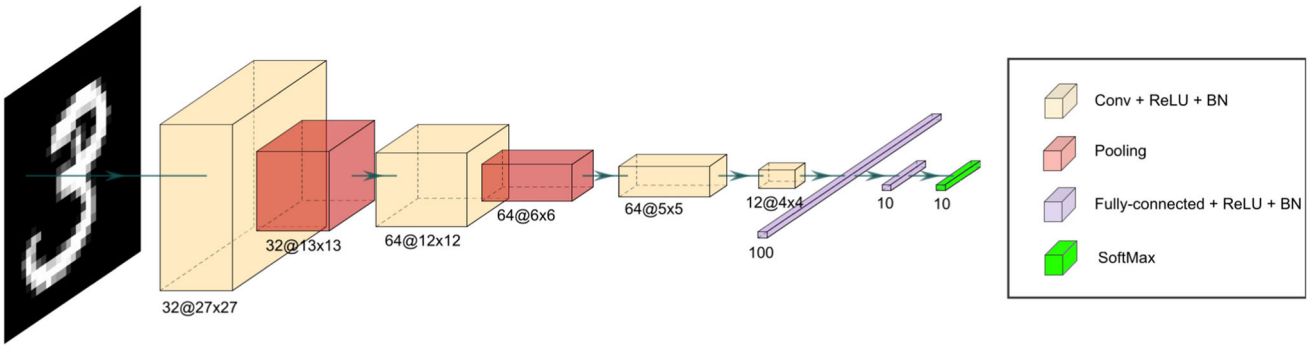
Regarding the models, two versions for each neural network model have been developed, exploiting 2×2 and 3×3 convolutional kernels, respectively. Specifically, the models used for MNIST and Fashion-MNIST are reported in Fig. 3. Instead the models used in Cifar-10 are slightly deeper, reported in Fig. 4. In all models, the ReLU has been exploited as the activation function, which ensures that the next layer inputs are positive. Moreover, a SoftMax layer has been employed as a final layer to carry out the classification.

In detail, the MNIST/Fashion-MNIST models have been developed based on the LeNet-5 architecture [59], with an increased depth to counteract the lower precision [60]. Four convolutional layers have been devised, each of them followed by a batch normalization (BN) layer. These layers are used to accelerate the training by reducing internal covariate shift [61]. Furthermore, a pooling layer is added after the first and second convolutional layers. Both an average and a max pooling have been tested. It is worth noting that the average pooling is of particular interest in the context of PNNs, since it is a linear operation, and hence, it can be easily implemented in photonics [62]. At the end of the structures, there are two fully connected (FC) layers, composed of 100 and 10 neurons, respectively.

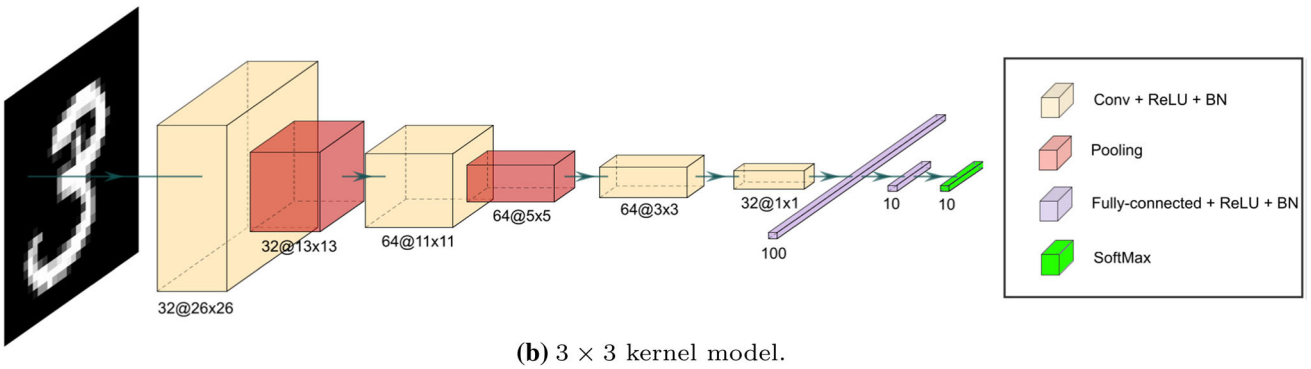
The developed models are compliant with the fan-in limitation of 200: this is particularly critical in the flattening operation, i.e., when passing from a convolutional layer to an FC one. In the 2×2 kernel model, the last convolutional layer has 12 feature maps of size 4×4. This results in 192 features, compliant with the fan-in for the subsequent FC layer. Similarly, with 3×3 kernel, 32 one-element feature maps are obtained after the last convolutional layer. In this way, these two models are photonic-compliant and can be used on the MNIST and Fashion-MNIST datasets.

Concerning the Cifar-10 dataset, the same strategy has been used: two models have been defined, one for each kernel size. The developed architectures are derived from the Binary Nets [63], with some modifications to satisfy photonic constraints. The 2×2 model exploits 5 convolutional layers, each one followed by a ReLU activation function and a BN layer. After the second, third, and fourth convolutional layer, pooling layers are employed. The third pooling layer allows the developed model to further reduce the number of features aimed to comply with the fan-in constraint. After flattening, three FC layers have been used, composed of 200, 100, and 10 neurons, respectively. The 3×3 model is very similar to the 2×2 model with one except for the absence of the third pooling layer: in this case, the fan-in constraint is satisfied due to the larger kernel size that shrinks down more the number of features.

Regarding the training phase, all the models have been trained using Adam as optimizer [64]. In details, the MNIST/Fashion-MNIST models have been trained for 30

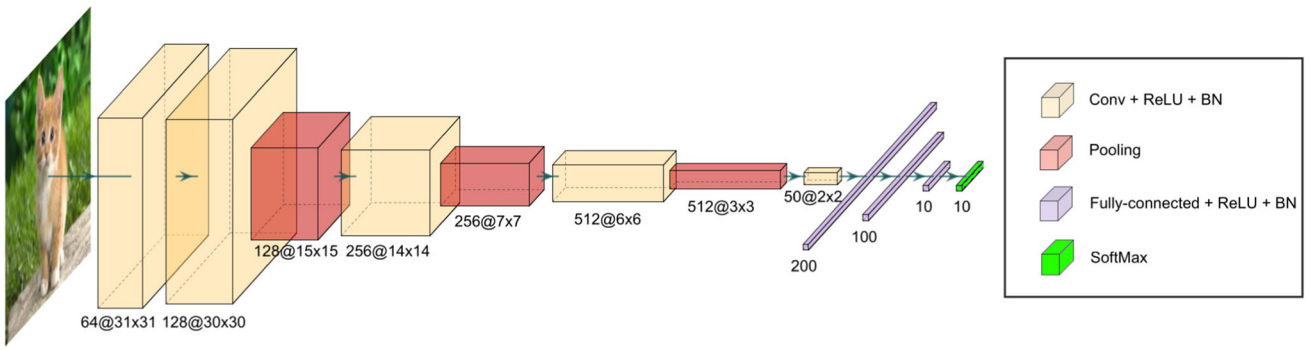


(a) 2×2 kernel model.

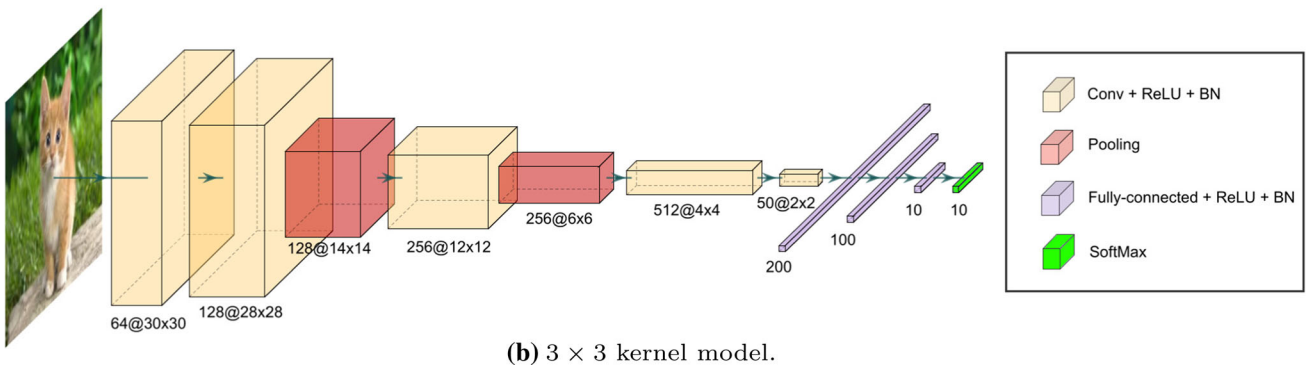


(b) 3×3 kernel model.

Fig. 3 PANN architectures for experiments on MNIST and Fashion-MNIST datasets



(a) 2×2 kernel model.



(b) 3×3 kernel model.

Fig. 4 PANN architectures for experiments on Cifar-10 dataset

epochs, while the Cifar-10 models have exploited 50 epochs. All the training phases have used a batch size equal to 64.

4 PANN results

In this section, we discuss the results obtained by the models suited for photonic implementation on the three benchmark datasets i.e., MNIST, Fashion-MNIST, and Cifar-10. The experiments have been conducted using DoReFa with a varying number of bit resolution (i.e., 2, 4, and 6 bits) on two model versions, with 2×2 and 3×3 kernels. Additionally, for each model we have considered two pooling variants, exploiting either max pooling or average pooling.

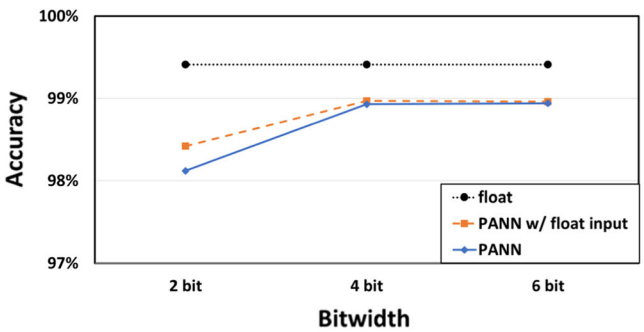
To make a fair comparison, PANN architectures (blue line in the following figures) have been compared with two baselines: (i) *float*, exploiting a 32-bit floating-point architecture (black line), and (ii) *PANN w/ float input* where just the input of the first layer is not quantized (orange line).

4.1 MNIST

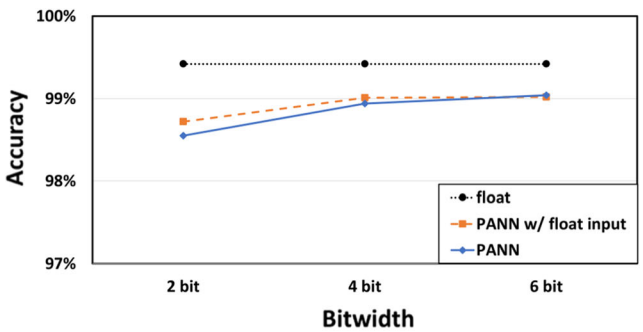
The accuracy of the models with max pooling on MNIST dataset is shown in Fig. 5 as a function of the bit resolution of the quantized parameters.

The figure shows a limited accuracy degradation when exploiting PANN models. Indeed, in the worst case (i.e., 2×2 kernel with 2 bits) an accuracy drop of about 1.3% can be observed with respect to the float baseline. Moreover, it can be noticed that the input quantization itself has a certain impact on the accuracy, especially at low bitwidths.

The results obtained on MNIST dataset using average pooling are reported in Fig. 6. Both 2×2 and 3×3 kernel sizes show very good performance. In the case of 3×3 kernel using 6 bits, the accuracy degradation with respect to the baseline model is very low (i.e., 0.26%), showing that the photonic hardware can approach the performance of the electronic counterpart. Comparing the pooling options, slightly better results are obtained with the average pooling, even if the difference are very small (0.15% on average).

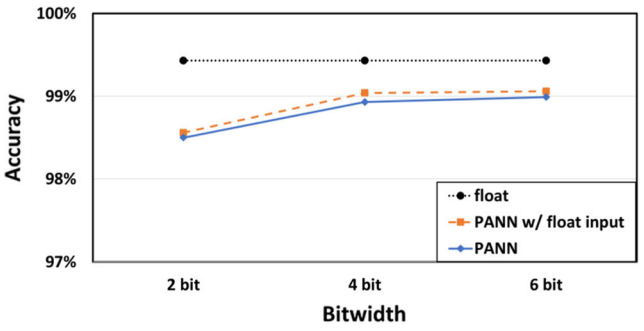


(a) 2×2 kernel.

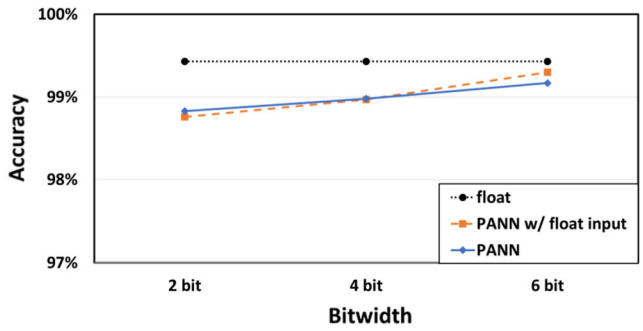


(b) 3×3 kernel.

Fig. 5 Results on MNIST dataset with max pooling



(a) 2×2 kernel.



(b) 3×3 kernel.

Fig. 6 Results on MNIST dataset with average pooling

4.2 Fashion-MNIST

The results obtained by PANN models on the Fashion-MNIST dataset using max pooling are reported in Fig. 7. The performance degradation with respect to the float baseline is slightly increased due to the more complex structure of the dataset. When using 2 bits, the accuracy drop is 5.8% (5.4%) with the 2×2 (3×3) kernel. However, by exploiting a higher number of bits, the difference between the photonic models and the floating-point architectures can be significantly reduced, up to 1.1% using six bits in both kernel configurations. Considering the PANN with floating point input, the input quantization reduces the accuracy up to 1.3%, however this impact can be made negligible by increasing the bitwidth, as shown in the 6-bit configurations.

The results with average pooling are shown in Fig. 8.

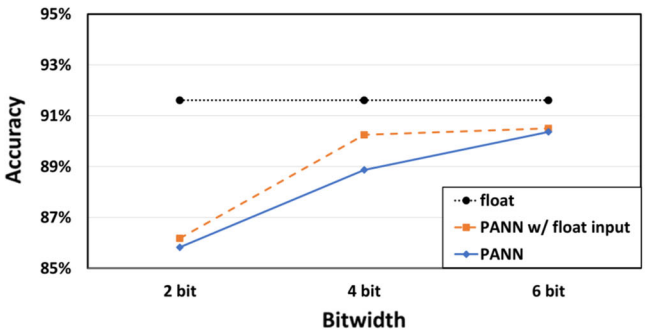
The overall behavior is similar to max pooling: indeed even in this case the 3×3 kernel models perform better than 2×2 models. The PANN accuracy decreases with respect to max pooling except for the 3×3 on two bits. The performance drop with respect to the float baseline is 2.1% in the best case, i.e., 3×3 kernel on six bits.

4.3 Cifar-10

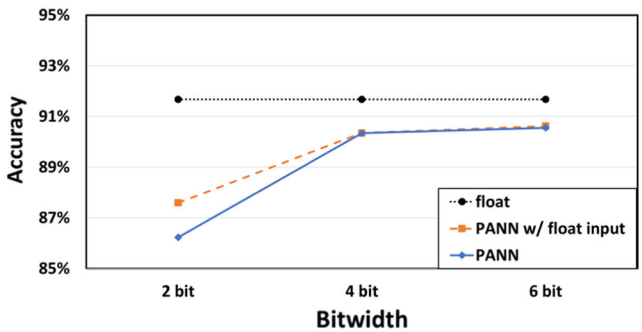
The accuracy of PANN models on Cifar-10 dataset using max pooling is shown in Fig. 9. In this case, the gap between the PANN models and the floating point baseline is higher, about 8% in the best case (i.e., 3×3 kernel on six bits). When using just two bits, the accuracy drop reaches 20% (18.2%) for the 2×2 (3×3) kernel models, with an 8% drop caused by the input quantization. This is due to the complexity of the Cifar-10 dataset, which is composed of RGB images (i.e., 3-channel images) with more features compared to the MNIST/Fashion-MNIST datasets. Again, the input quantization issue can be solved by using higher bitwidths, indeed no impact can be observed when operating at a bitwidth of 6.

The performance achieved by the average pooling on Cifar-10 is reported in Fig. 10.

In this scenario, the accuracy is slightly higher compared to the max pooling, reaching a drop of 6.4% in the best case (3×3 kernel size on 6 bits). Thus, the average pooling allows to decrease the gap with both the baselines (i.e., also reducing the impact of the input quantization).

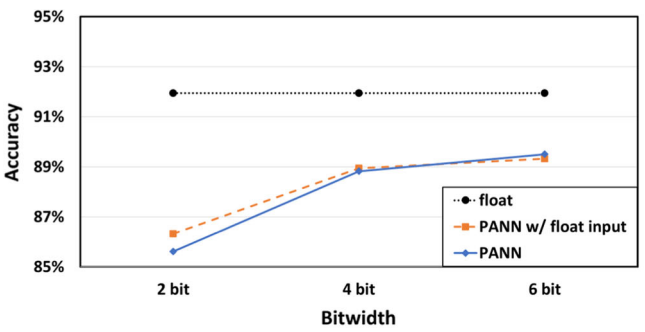


(a) 2×2 kernel.

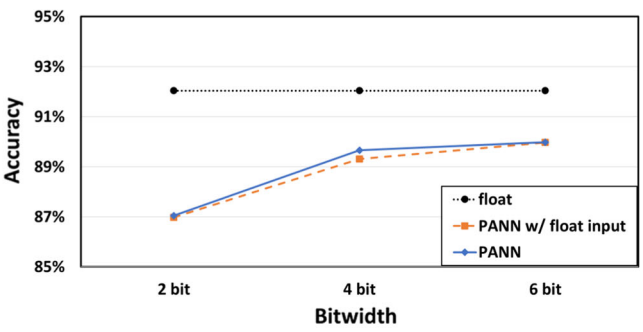


(b) 3×3 kernel.

Fig. 7 Results on Fashion-MNIST dataset with max pooling

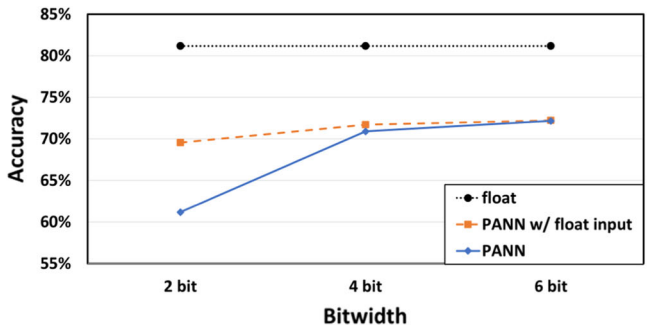


(a) 2×2 kernel.

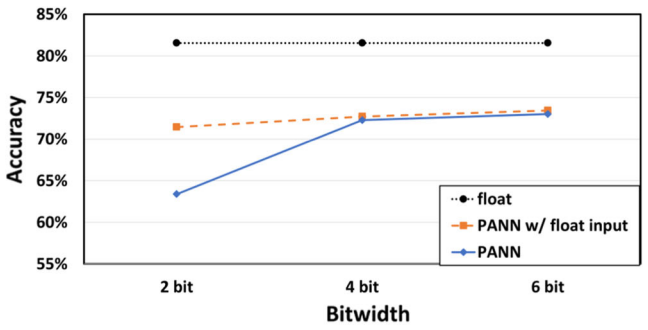


(b) 3×3 kernel.

Fig. 8 Results on Fashion-MNIST dataset with average pooling

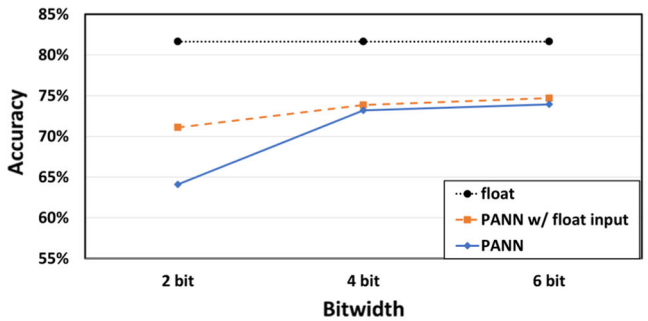


(a) 2×2 kernel.

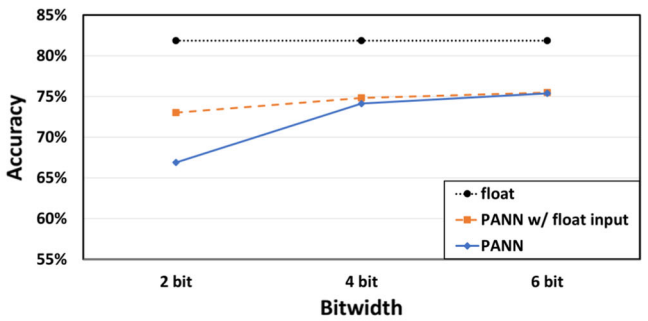


(b) 3×3 kernel.

Fig. 9 Results on Cifar-10 dataset with max pooling



(a) 2×2 kernel.



(b) 3×3 kernel.

Fig. 10 Results on Cifar-10 dataset with average pooling

5 Conclusion

The breakthrough of deep learning has recently led to a renewed interest on photonics for computing. Several optical architectures for deep learning have been investigated, focusing on integrated approaches. Such photonic accelerators promise to bring substantial improvements in terms of speed up and power consumption and footprint reduction for DNN inference.

Exploiting these analog processors comes at the cost of some limitations: computations are carried out on analog signals with limited resolution (≤ 6 bits with equally spaced levels) with positive-valued inputs, the fan-in to photonic neurons is limited to a couple of hundreds inputs, and convolutional kernels are demonstrated up to a 3×3 size. These limitations mainly derive from the lack of mature photonic technologies, leading to undesired losses and impairments. While in the foreseeable future some of these aspects are likely to be improved, the exploitation of photonic hardware for DNN acceleration in the short-medium term requires AI models that are compliant with current technology.

In this paper, we, therefore, introduced the concept of PANN architectures, i.e., DNN models compliant with the constraints imposed by the photonic hardware. Moreover, we devised a quantization-based PANN training-to-inference scheme to obtain neural network weights in the fixed-point domain suited for the underlying photonic architecture.

The performance of these DNN models has been then assessed in computer vision tasks. During our experiments, we considered two kernel sizes, namely 2×2 and 3×3, and two pooling schemes, i.e., max pooling model and average pooling. The impact of the different bitwidths (2, 4, and 6 bits) on the accuracy has been reported and discussed. Thanks to the higher number of parameters and the higher computational precision, models with larger kernel size and higher bitwidths achieve higher accuracy. Indeed in all three datasets, the highest accuracy is reached by the 3×3 kernel-sized model with six bits. Specifically, in the best case we are able to reach an accuracy of 99.2% on MNIST, 90.2% on Fashion-MNIST, and 75.4% on Cifar-10. When compared to the floating point baseline the PANN architectures suffer a very limited drop in accuracy on MNIST and Fashion-MNIST (i.e., 0.3% and 1.1% in the best case, respectively) and slightly higher on Cifar-10 (i.e., 6.5%). The impact of input quantization is negligible for all tested configurations, unless just two bits are exploited on Cifar-10 dataset. Moreover, max and average pooling schemes achieve similar results, with the latter

even outperforming the former in all configurations of both MNIST and Cifar-10, thus enabling the use of all-optical average pooling, which can be easily realized with passive devices.

These results show the feasibility of DNN operations using photonic hardware. However, further development and investigations are required to improve the scalability of the underlying photonic hardware in terms of bit resolution and trade-off with weight update frequency, neuron fan-in and kernel size [65].

Funding Open access funding provided by CRUI-CARE.

Declarations

Conflict of interests All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Sinha R, Pandey R, Pattnaik R (2018) Deep learning for computer vision tasks: a review
2. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. Neural Inf Process Syst. https://doi.org/10.1145/3065386
3. Dahl GE, Yu D, Deng L, Acero A (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans Audio Speech Lang Process 20(1):30–42. https://doi.org/10.1109/TASL.2011.2134090
4. Soundarya B, Krishnaraj R, Mythili MS (2021) Visual speech recognition using convolutional neural network. In: IOP conference series: materials science and engineering, vol 1084, p 012020. https://doi.org/10.1088/1757-899X/1084/1/012020
5. Zhang Z, Huang S (2020) Credit card fraud detection via deep learning method using data balance tools. In: 2020 international conference on computer science and management technology (ICCSMT), pp 133–137. https://doi.org/10.1109/ICCSMT51754.2020.00033
6. Dong S, Wang P, Abbas K (2021) A survey on deep learning and its applications. Comput Sci Rev 40:100379
7. Von Neumann J (1993) First draft of a report on the EDVAC. IEEE Ann Hist Comput 15(4):27–75
8. Guo K, Zeng S, Yu J, Wang Y, Yang H (2019) [DL] A survey of FPGA-based neural network inference accelerators. ACM Trans Reconfig Technol Syst 12(1):1–26. https://doi.org/10.1145/3289185
9. Azghadi MR, Lammie C, Eshraghian JK, Payvand M, Donati E, Linares-Barranco B, Indiveri G (2020) Hardware implementation of deep network accelerators towards healthcare and biomedical applications. IEEE Trans Biomed Circuits Syst 14(6):1138–1159. https://doi.org/10.1109/TBCAS.2020.3036081
10. Williams RS (2017) What's next? [the end of Moore's law]. Comput Sci Eng 19(2):7–13. https://doi.org/10.1109/MCSE.2017.31
11. Hamerly R, Sludds A, Bernstein L, Prabhu M, Roques-Carmes C, Carolan J, Yamamoto Y, Soljačić M, Englund D (2019) Towards large-scale photonic neural-network accelerators. In: 2019 IEEE international electron devices meeting (IEDM), pp 22.8.1–22.8.4. https://doi.org/10.1109/IEDM19573.2019.8993624
12. Schuman CD, Potok TE, Patton RM, Birdwell JD, Dean ME, Rose GS, Plank JS (2017) A survey of neuromorphic computing and neural networks in hardware. arXiv preprint arXiv:1705.06963
13. Shin D, Yoo H-J (2020) The heterogeneous deep neural network processor with a non-von Neumann architecture. Proc IEEE 108(8):1245–1260. https://doi.org/10.1109/JPROC.2019.2897076
14. Nahmias MA, de Lima TF, Tait AN, Peng H-T, Shastri BJ, Prucnal PR (2020) Photonic multiply-accumulate operations for neural networks. IEEE J Sel Top Quantum Electron 26(1):1–18. https://doi.org/10.1109/JSTQE.2019.2941485
15. De Marinis L, Cococcioni M, Castoldi P, Andriolli N (2019) Photonic neural networks: a survey. IEEE Access 7:175827–175841. https://doi.org/10.1109/ACCESS.2019.2957245
16. Bangari V, Marquez BA, Miller HB, Tait AN, Nahmias MA, de Lima TF, Peng H-T, Prucnal PR, Shastri BJ (2020) Digital electronics and analog photonics for convolutional neural networks (DEAP-CNNs). IEEE J Sel Top Quantum Electron 26:1–13
17. Miscuglio M, Mehrabian A, Hu Z, Azzam SI, George J, Kildishev AV, Pelton M, Sorger VJ (2018) All-optical nonlinear activation function for photonic neural networks. Opt Mater Express 8(12):3851–3863. https://doi.org/10.1364/OME.8.003851
18. Mourgias-Alexandris G, Tsakyridis A, Passalis N, Tefas A, Vyrsokinos K, Pleros N (2019) An all-optical neuron with sigmoid activation function. Opt Express 27(7):9620–9630
19. George JK, Mehrabian A, Amin R, Meng J, De Lima TF, Tait AN, Shastri BJ, El-Ghazawi T, Prucnal PR, Sorger VJ (2019) Neuromorphic photonics with electro-absorption modulators. Opt Express 27(4):5181–5191
20. Williamson IAD, Hughes TW, Minkov M, Bartlett B, Pai S, Fan S (2020) Reprogrammable electro-optic nonlinear activation functions for optical neural networks. IEEE J Sel Top Quantum Electron 26(1):1–12. https://doi.org/10.1109/JSTQE.2019.2930455
21. Bartlett B, Minkov M, Hughes T, Williamson IAD (2019) Neuroptica: flexible simulation package for optical neural networks. GitHub https://github.com/fancompute/neuroptica
22. Pai S, Williamson IA, Hughes TW, Minkov M, Solgaard O, Fan S, Miller DA (2019) Parallel fault-tolerant programming of an arbitrary feedforward photonic network. arXiv preprint arXiv:1909.06179
23. Pai S, Bartlett B, Solgaard O, Miller DAB (2019) Matrix optimization on universal unitary photonic devices. Phys Rev Appl 11(6):064044. https://doi.org/10.1103/PhysRevApplied.11.064044
24. Geiger L, Team P (2020) An open-source library for training binarized neural networks. J Open Source Softw 5(45):1746. https://doi.org/10.21105/joss.01746

25. Krishnamoorthy AV, Ho R, Zheng X, Schwetman H, Lexau J, Koka P, Li G, Shubin I, Cunningham JE (2009) Computer systems based on silicon photonic interconnects. Proc IEEE 97(7):1337–1361
26. Wetzstein G, Ozcan A, Gigan S, Fan S, Englund D, Soljačić M, Denz C, Miller DA, Psaltis D (2020) Inference in artificial intelligence with deep optics and photonics. Nature 588(7836):39–47
27. Stubkjaer KE (2000) Semiconductor optical amplifier-based all-optical gates for high-speed optical processing. IEEE J Sel Top Quantum Electron 6(6):1428–1435. https://doi.org/10.1109/2944.902198
28. Kim JH, Jhon YM, Byun YT, Lee S, Woo DH, Kim SH (2002) All-optical XOR gate using semiconductor optical amplifiers without additional input beam. IEEE Photon Technol Lett 14(10):1436–1438. https://doi.org/10.1109/LPT.2002.801841
29. Bogoni A, Wu X, Bakhtiari Z, Nuccio S, Willner AE (2010) 640 Gbits/s photonic logic gates. Opt Lett 35(23):3955–3957
30. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
31. Chang H-Y, Narayanan P, Lewis SC, Farinha NC, Hosokawa K, Mackin C, Tsai H, Ambrogio S, Chen A, Burr GW (2019) AI hardware acceleration with analog memory: microarchitectures for low energy at high speed. IBM J Res Dev 63(6):8:1-8:14
32. Haensch W, Gokmen T, Puri R (2018) The next generation of deep learning hardware: analog computing. Proc IEEE 107(1):108–122
33. Paliy M, Strangio S, Ruiu P, Rizzo T, Iannaccone G (2020) Analog vector-matrix multiplier based on programmable current mirrors for neural network integrated circuits. IEEE Access 8:203525–203537
34. Brunner D, Soriano MC, Van der Sande G (2019) Photonic reservoir computing. De Gruyter 8:19
35. Shastri BJ, Tait AN, de Lima TF, Pernice WH, Bhaskaran H, Wright CD, Prucnal PR (2021) Photonics for artificial intelligence and neuromorphic computing. Nat Photon 15(2):102–114
36. Xu X, Tan M, Corcoran B, Wu J, Boes A, Nguyen TG, Chu ST, Little BE, Hicks DG, Morandotti R et al (2021) 11 TOPS photonic convolutional accelerator for optical neural networks. Nature 589(7840):44–51
37. Totović AR, Dabos G, Passalis N, Tefas A, Pleros N (2020) Femtojoule per MAC neuromorphic photonics: an energy and technology roadmap. IEEE J Sel Top Quantum Electron 26(5):1–15
38. Shen Y, Harris NC, Skirlo S, Prabhu M, Baehr-Jones T, Hochberg M, Sun X, Zhao S, Larochelle H, Englund D et al (2017) Deep learning with coherent nanophotonic circuits. Nat Photon 11(7):441–446
39. Mourgias-Alexandris G, Totović A, Tsakyridis A, Passalis N, Vyrsokinos K, Tefas A, Pleros N (2019) Neuromorphic photonics with coherent linear neurons using dual-IQ modulation cells. J Lightw Technol 38(4):811–819
40. De Marinis L, Cococcioni M, Liboiron-Ladouceur O, Contestabile G, Castoldi P, Andriolli N (2021) Photonic integrated reconfigurable linear processors as neural network accelerators. Appl Sci 11(13):6232
41. Huang C, Fujisawa S, De Lima TF, Tait AN, Blow E, Tian Y, Bilodeau S, Jha A, Yaman F, Batshon HG et al (2020) Demonstration of photonic neural network for fiber nonlinearity compensation in long-haul transmission systems. In: 2020 optical fiber communications conference and exhibition (OFC), pp 1–3. IEEE
42. Shi B, Calabretta N, Stabile R (2019) Deep neural network through an InP SOA-based photonic integrated cross-connect. IEEE J Sel Top Quantum Electron 26(1):1–11
43. De Marinis L, Catania A, Castoldi P, Bruschi P, Piotto M, Andriolli N (2021) A codesigned photonic electronic MAC neuron with ADC-embedded nonlinearity. In: CLEO: science and innovations, pp 3–4. Optical Society of America
44. de Lima TF, Tait AN, Saeidi H, Nahmias MA, Peng H-T, Abbaslou S, Shastri BJ, Prucnal PR (2020) Noise analysis of photonic modulator neurons. IEEE J Sel Top Quantum Electron 26(1):1–9. https://doi.org/10.1109/JSTQE.2019.2931252
45. Stark P, Horst F, Dangel R, Weiss J, Offrein BJ (2020) Opportunities for integrated photonic neural networks. Nanophotonics 9(13):4221–4232
46. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Gordon G, Dunson D, Dudík M (eds) Proceedings of the fourteenth international conference on artificial intelligence and statistics. Proceedings of machine learning research, vol 15. PMLR, Fort Lauderdale, FL, USA, pp 315–323. https://proceedings.mlr.press/v15/glorot11a.html
47. Cococcioni M, Rossi F, Ruffaldi E, Saponara S (2020) Fast deep neural networks for image processing using posits and ARM scalable vector extension. J Real-Time Image Process 17(3):759–771. https://doi.org/10.1007/s11554-020-00984-x
48. Wang M, Rasoulinezhad S, Leong PHW, So HK-H (2022) NITI: training integer neural networks using integer-only arithmetic. IEEE Trans Parallel Distrib Syst. https://doi.org/10.1109/TPDS.2022.3149787
49. Shin S, Hwang K, Sung W (2016) Quantized neural network design under weight capacity constraint. arXiv preprint arXiv:1611.06342
50. Hubara I, Courbariaux M, Soudry D, El-Yaniv R, Bengio Y (2017) Quantized neural networks: training neural networks with low precision weights and activations. J Mach Learn Res 18(1):6869–6898
51. Rastegari M, Ordonez V, Redmon J, Farhadi A (2016) XNOR-Net: ImageNet classification using binary convolutional neural networks. CoRR arXiv:abs/1603.05279
52. Courbariaux M, Bengio Y (2016) BinaryNet: training deep neural networks with weights and activations constrained to +1 or − 1. CoRR arXiv:abs/1602.02830
53. Zhou S, Ni Z, Zhou X, Wen H, Wu Y, Zou Y (2016) DoReFa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients. CoRR arXiv:abs/1606.06160
54. Helwegen K, Widdicombe J, Geiger L, Liu Z, Cheng K-T, Nusselder R (2019) Latent weights do not exist: rethinking binarized neural network optimization. Adv Neural Inf Process Syst 32:1–12
55. Bengio Y, Léonard N, Courville AC (2013) Estimating or propagating gradients through stochastic neurons for conditional computation. CoRR arXiv:abs/1308.3432
56. LeCun Y, Cortes C (2010) MNIST handwritten digit database
57. Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. CoRR arXiv:abs/1708.07747
58. Krizhevsky A, Nair V, Hinton G. CIFAR-10 (Canadian Institute for Advanced Research)
59. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
60. Guo Y (2018) A survey on methods and theories of quantized neural networks. arXiv preprint arXiv:1808.04752
61. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. CoRR arXiv:abs/1502.03167
62. De Marinis L, Nesti F, Cococcioni M, Andriolli N (2020) A photonic accelerator for feature map generation in convolutional neural networks. In: Photonics in switching and computing. Optical Society of America, pp 1–3

63. Courbariaux M, Bengio Y, David J (2015) BinaryConnect: training deep neural networks with binary weights during propagations. CoRR arXiv:abs/1511.00363
64. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980
65. Sunny FP, Taheri E, Nikdast M, Pasricha S (2021) A survey on silicon photonics for deep learning. J Emerg Technol Comput Syst 17(4):1–57. https://doi.org/10.1145/3459009