Survey on Activation Functions for Optical Neural Networks

OCEANE DESTRAS, Ecole polytechnique de Montreal, Canada
SÉBASTIEN LE BEUX, Concordia university, Canada
FELIPE GOHRING DE MAGALHÃES and GABRIELA NICOLESCU, Ecole polytechnique de Montreal, Canada

Integrated photonics arises as a fast and energy-efficient technology for the implementation of artificial neural networks (ANNs). Indeed, with the growing interest in ANNs, photonics shows great promise to overcome current limitations of electronic-based implementation. For example, it has been shown that neural networks integrating optical matrix multiplications can potentially run two orders of magnitude faster than their electronic counterparts. However, the transposition in the optical domain of the activation functions, which is a key feature of ANNs, remains a challenge. There is no direct optical implementation of state-of-the-art activation functions. Currently, most designs require time-consuming and power-hungry electro-optical conversions. In this survey, we review both all-optical and opto-electronic activation functions proposed in the state-of-the-art. We present activation functions with their key characteristics, and we summarize challenges for their use in the context of all-optical neural networks. We then highlight research directions for the implementation of fully optical neural networks.

CCS Concepts: • **Computing methodologies → Neural networks**; • **Hardware → Emerging optical and photonic technologies;**

Additional Key Words and Phrases: Optical neural network, non-linear optics, optical activation function, photonic integrated circuit

ACM Reference format:
Oceane Destras, Sébastien Le Beux, Felipe Gohring de Magalhães, and Gabriela Nicolescu. 2023. Survey on Activation Functions for Optical Neural Networks. *ACM Comput. Surv.* 56, 2, Article 35 (September 2023), 30 pages.
https://doi.org/10.1145/3607533

1 INTRODUCTION

Artificial Intelligence (AI)-powered systems are used for a variety of applications. However, they are usually resource hungry, requiring significant amounts of processing elements and memories. With the end of Moore's law, state-of-the-art architectures for AI algorithms will no longer enable the processing and the storage of data as predicted in Reference [1]. Furthermore, the need for power-efficient ANN implementations has been emerging. Indeed, an application of image

Authors' addresses: O. Destras, F. gohring de Magalhaes, and G. nicolescu, Ecole polytechnique de Montreal, 2500 Chemin de Polytechnique, Montréal, Québec, Canada, H3T 1J4; emails: oceane.destras@polymtl.ca, felipe.gohring-de-magalhaes@polymtl.ca, gabriela.nicolescu@polymtl.ca; S. Le Beux, Concordia University, 1455 De Maisonneuve Blvd. W., Montréal, Québec, Canada, H3G 1M8; email: slebeux@encs.concordia.a.

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

© 2023 Copyright held by the owner/author(s).
0360-0300/2023/09-ART35
https://doi.org/10.1145/3607533

recognition may have to implement billions of operations just for the processing of one image [2]. Solutions currently being explored involve the replacement of fully or partially integrated electronic circuits with photonic circuits [3].

A key feature of silicon photonics is **wavelength division multiplexing (WDM)**, which enables the parallel transmission of multiple signals on the same medium without interferences. In the context of **optical neural networks (ONNs)**, WDM allows parallel processing of multiple data at the same time. The potential to outperform electronic implementations in terms of speed and energy efficiency is another promising feature of ONNs [4]. For instance, the execution of operations on a conventional computer, such as matrix multiplications, is power and area hungry [5], while it may be computed at ultra-high speed using specific configurations of photonic networks [6]. Indeed, all-optical ANNs, which require no optoelectronics or electro-optical conversion other than the interface, allow matrix multiplications to be performed at the speed of light as optical signals propagate in waveguides. Silicon photonics enables the integration of photonic and electronic devices on the same platform [7]. The two most used optical modulators are **Mach-Zendher interferometers (MZIs)** and **microring resonators (MRRs)**. MZIs are bulkier, but they are less sensitive to process and temperature variations. This is because signal processing is accomplished by delaying the signal in one of the two branches. MRRs are more compact, and dot products are obtained by slightly detuning the resonant wavelength of the MRR from the input signal. While this enables WDM, the accurate calibration of the rings, for which resonance drifts with the temperature, leads to significant complexity and power overhead. Overall, we conclude that MZIs are, so far, a more robust approach to design ONNs compared to MRRs.

One of the main challenges in replicating an ANN with an ONN is to be able to implement optically every module of a classic ANN. The implementation of an optical matrix multiplication has been proven already [8] but the **activation function (AF)**, which is an essential function in an ANN, was not fully addressed. The matrix multiplication corresponds to the linear transformation that data undergoes in an ANN. However, to obtain optimal results, a non-linear transformation is also needed. In an ANN, this is achieved by the AF. The existing contributions offer either ONN implementations where the AF is performed on a computer or partly with electric components or, when implemented fully optically, using optical non-linearities at a material or device level. In the first case, the conversion of information from the optical circuit is carried out with the AF on a computer and then the output is converted back to the optical circuit. However, this means that the speed of the network can be restricted by electronic circuit limitations [3]. Further, optical-electrical conversions add noise, which degrades the accuracy. Also, converting data back and forth introduces expressive latency and bigger power consumption, which ends up jeopardizing the gain obtained by the optical implementation. Consequently, even though some works implementing the AF electrically (e.g., References [9, 10]) already promise highly competitive results, we believe that an optical AF is needed to attain the full potential of ONNs.

This survey presents a discussion on current trends for ONNs. To the best of our knowledge, this is the first discussion that focuses on this critical aspect of ONNs: an optical AF. Acknowledging tremendous advances in the computational part (i.e., matrix computation), there is still a gap in resolving the AF optically, which is explored in this work. The organization of this article is the following. We first introduce basic concepts on ANNs and Integrated photonic. In Section 4, we give an overview of the different tools that are available for the simulation of ONNs. Then, in Sections 5 and 6, we describe the electro-optical and all-optical solutions proposed to implement AFs in the current state-of-the-art. We present the technology used and highlight the performance of these solutions in terms of accuracy, power consumption, and speed. Finally, we compare the different solutions presented in this article in Section 7 and discuss future research directions in Section 8.

2 ARTIFICIAL NEURAL NETWORK

Machine learning (ML) is a subgroup of AI [11] that focuses on creating machines to solve problems humans encounter regularly and solve easily but are much harder for a machine to deal with (e.g., image recognition and language processing). A popular ML algorithm is **Deep Learning (DL)**. Its design is inspired by the biological structure of the human brain and is called an **Artificial Neural Network (ANN)**. The goal of DL algorithms is to build a mathematical model able to predict accurately outputs for a given task.

We can sort ANN architectures into three categories: **Spiking Neural Networks (SNNs)**, **Reservoir Computing (RC)**, and **Deep Neural Networks (DNNs)** [12]. SNNs and RC are called stateful (i.e., with memory) as opposed to DNNs, which are stateless [4]. The particularity of SNNs is to use spikes to encode the inputs of the network. Each spike is characterized by the timing of its occurrence and its amplitude. SNNs have shown great potential in terms of energy consumption [4]. However, some advancements still need to be made to compete with ANNs. Especially concerning the efficiency of their training algorithm [13]. RC architectures are mainly composed of an input layer, a reservoir and a readout layer. The reservoir is a cluster of non-linear neurons randomly connected to each other. It is the readout layer that is trained when the reservoir needs to be optimized for a given application. RC algorithms do not need a lot of computer resources as their optimization is linear and their training datasets are small [14]. In this article, we concentrate on the study of stateless ANNs, meaning DNNs. However, we do not completely put aside the progress made toward the integration of photonic RCs. Indeed, the non-linear neurons in both RC and DNN architectures are similar enough to be considered in this study. In addition, despite their differences, RC could potentially be used to reproduce the behaviour of DNNs [12].

The algorithm of a DNN is as follows. The network receives as input a dataset containing the information it must process and learn from, called the training set. This set of data undergoes a series of transformations and is fed in the forward direction of the DNN. This feed-forward network consists of matrix multiplications and AFs. Then, the algorithm revises the parameters of the model (backward propagation) to minimize the difference between the obtained result and the expected result. Finally, the model is tested on a test set, different from the train set, to simulate how it would perform on a new and unknown dataset [15].

2.1 Feed-forward Neural Network

A classic DNN is composed of different layers of neurons, each of them possessing its own function. We present here fully connected DNNs, where each neuron in a layer is connected to all the neurons of the next layer. Figure 1 illustrates the architecture of a fully connected DNN. There are three types of layers. The input layer is a vector defining one element of the dataset, with each neuron (blue in the figure) representing a value of the vector. Each element of the dataset is processed consecutively. The hidden layers are positioned after the input layer. Each neuron (grey in the figure) of a hidden layer, i, performs a weighted sum of each output of the neurons of the preceding layer. The result is subjected to a non-linear transformation σ, or AF as detailed thereafter. The output of the kth neuron in the ith layer is calculated with Equations (1) and (2):

$$z_{i,k} = \sum_{j=0}^{n} w_{j,k}^{i} \times a_{i-1,j}, \tag{1}$$

$$a_{i,k} = \sigma^{i}(z_{i,k}), \tag{2}$$

where n is the size (i.e., the number of neurons) of the $(i-1)$th layer and $w_{j,k}^{i}$ is the weight applied by the kth neuron of the ith layer on the jth neuron of the $(i-1)$th layer. The weighted

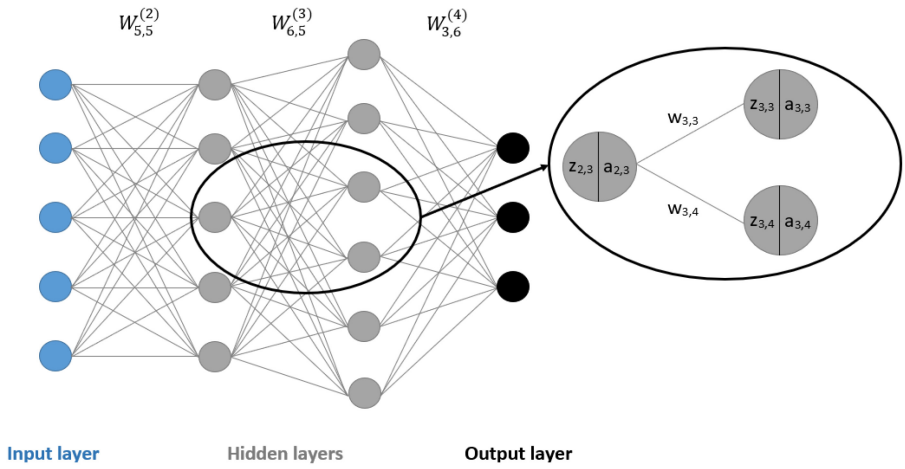


Fig. 1. Diagram of a fully connected neural network presented in Reference [16].

sum happens in each neuron of a layer simultaneously and is equivalent to identifying the layer as a vector A^i and multiplying this vector by a weight matrix W^i: $A^i = \sigma^i(W^i \times A^{i-1})$.

The output layer (black in the figure) is the final layer and behaves in the same ways as the hidden layers. Its number of neurons depends on the type of problem it is facing. In the case of a classification problem with two possible output classes, there will be two output neurons, each representing the probability of an element of the dataset to belong to each class. To obtain this probability, a function called the softmax function [17] is more commonly used as the final non-linear transformation.

2.2 Backward Propagation

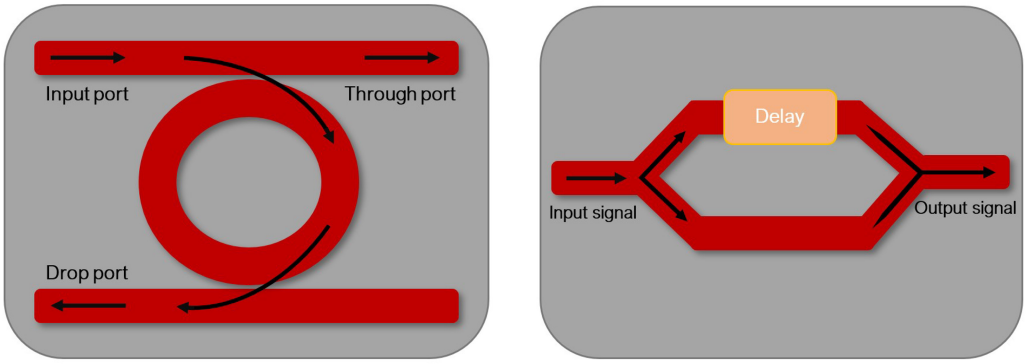
The role of the backward propagation, or backpropagation [18], is to update the parameters of the weight matrix after each data batch is processed to minimize the output error. This is achieved by progressively adjusting the weights on each step of the execution, following a given algorithm such as the gradient descent [19]. The gradient descent algorithm uses the partial derivative of the output error, also called the loss, with respect to each weight. The loss represents the difference between the expected output and the predicted output. It differentiates from the accuracy, which directly represents how the model performs on a given dataset. The goal is to find the value of W_{new}, which minimizes the loss. To update the weights, the equation used is Equation (3), where α is the learning rate that defines the speed at which the DNN model is updated:

$$W_{new} = W_{old} - \alpha \frac{\delta loss}{\delta W_{old}}. \tag{3}$$

2.3 Activation Function

The AF is an essential part of a **neural network (NN)**. Without it there is no interest in building deep NNs: NNs with multiple layers. Indeed, any NN of N linear layers, or N matrices, without AF can be simplified by a single linear layer NN. Also, it is important for a DNN to use differential AFs to enable the update of weights with the gradient descent algorithm. The following AFs are commonly used in NNs:

- The sigmoid function, which is a non-linear function whose outputs range between 0 and 1. Its mathematical formula is: $f(x) = \frac{1}{1+e^{-x}}$.



(a) (b)

Fig. 2. Schematics of the structure of a MRR (a) and a MZI (b).

- The **hyperbolic tangent function (tanh)** is a variant of the sigmoid function whose output values range from −1 and 1. Its mathematical formula is $f(x) = \frac{2}{1+e^{-2x}} - 1$.
- The ReLU function returns zero for negative input values and is linear for positive input values. The advantage of this function is to lessen the overall computation cost of the ANN and deactivate neurons that have a negative value and that could be considered as non-essential.

Sigmoid, ReLU and tanh functions are the most commonly used AFs in NN architectures. However, variants such as Leaky ReLU, Softplus, and **hard hyperbolic tangent (hardtanh)** are also popular [20].

3 INTEGRATED PHOTONIC

In this article, we review optical AFs in the context of integrated ONNs, where light travels through a waveguide, in opposition to free-space ONNs. **Photonic Integrated Circuits (PICs)** show great potential toward the low-cost production of high-speed ONNs [21]. This section explains what an optical component is and introduces some common ones. The basic structure of an ONN is then presented.

3.1 Optical Components

Optical components can be divided into two main categories: passive and active components. Passive components can only interact with the light and are incapable of emitting any. Active components can generate light and actively transform transmitted light. Photonic circuits are built using different components, such as directional couplers and modulators. Each component acts on transmitted light, interacting with it to achieve different operations.

A directional coupler is a common method to split and combine light using two parallel waveguides, in which the ratio of power coupled from one waveguide to another is defined by the coupling coefficient of the waveguides. The coupling coefficient is determined using a supermode analysis, a calculation including the effective indices of the coupled waveguides [22].

The MRR is an optical filter that selects a desired wavelength from a given input signal [23]. The design of the MRR is displayed in Figure 2(a). It consists of a "looped" waveguide that is bent over itself to form a circular structure. Tangential to this structure are placed one or two other waveguides. Depending on the coupling between the waveguide and the loop, as well as the wavelength, the light injected at the input port can be coupled into the loop and redirected through

the drop port or out of the loop, leaving via the through port. This creates a selective structure in which the light path is controlled by the looped waveguide.

A MZI enables the control of the amplitude of an optical wave. Figure 2(b) illustrates the schematic of a MZI. The input signal is split into two waves of equal phase and amplitude. One wave is transmitted unaltered through one of the branches. The second wave is delayed in time inducing a phase change. The merging of these two waves results in interferences and changes the amplitude of the output signal [23].

One of the main challenges related to the use of optical components is the power loss introduced by the interaction of light with the medium. In the context of architectures involving cascading of multiple components, the losses need to be compensated with higher signal power emitted at the source. Another solution to overcome this limit is the usage of a **semiconductor optical amplifier (SOA)** in the data path. A SOA enables amplifying the optical signal without converting it back to the electrical domain, introducing gains up to 30 db directly on the transmitted signal. Still, the usage of SOA implies the addition of noise in the transmitted signal and should be avoided when possible [22].

3.2 Optical Neural Network

ONNs are built using, among others, the aforementioned physical components. They are organized in the same manner as ANNs, i.e., with (i) an input layer, (ii) none, one or several hidden layers and (iii) an output layer.

The inputs of an ONN are optical signals generated by one or more optical sources. The outputs are more commonly the optical powers of the signals measured by photodetectors. As we saw in the previous section, a hidden layer can be broken down into two successive operations: a matrix multiplication and an AF. The same decomposition is followed in ONNs. Concerning the output layer, the softmax layer cannot be realized optically. It can either be replaced by an AF, which might impact the accuracy, or performed directly on a computer, following the photodetections of the output signals. In this article, we define the matrix multiplication and the AF transformation as, respectively, the linear and non-linear operations of an ONN.

Optical linear operations can be realized with the following methods: (i) Multiple plane light conversion, (ii) **Wavelength Division Multiplexing (WDM)**, and (iii) Meshes of MZIs.

Implementing an all-optical backpropagation is complex, because it requires regular updating of the optical components and a structure that can propagate the gradient. As a consequence, most work opted to perform this step on a computer. Once the training is done, the parameters of the model can be extracted and used to set an experimental demonstration. A first proposition for an optical backpropagation was presented in Reference [8].

The non-linear operation is not directly obtained by aforementioned components, as they must follow Maxwell's equations [24]. The next sections discuss solutions to implement these functions electro-optically or fully optically with integrated photonics.

4 TRAINING AND SIMULATION FRAMEWORKS FOR ONN

As we will observe in this survey, the reported ONN results were mostly simulated. In this section, we highlight the gap between the simulation of DNNs and PICs and the simulation of ONNs by introducing the different existing ONN simulation tools.

4.1 Overview

We present in Table 1 a non-exhaustive list of tools for DNN, PIC, and ONN simulations. For each tool, we specify the following: the date of publication and the date of the last update (e.g., for a github repository), their main programming language and whether they are open source. We

Table 1. List of Available Tools for the Simulation of DNNs, PICs, and ONNs

	Simulation tools	Published/Updated	Language	Open source
DNN	Tensorflow [25]	Nov. 2019/May 2022	C^{++}/Python	✓
	Keras [26]	Mar. 2015/May 2022	Python	✓
	Pytorch [27]	Sept. 2016/June 2022	C^{++}/Python	✓
	Scikit-learn [28]	June 2007/May 2022	Python	✓
	DL Toolbox [29]	2019/2022	Matlab	×
PIC	Rapid simulation of PICs [30]	2020	Verilog-A	×
	GDS Photonic Toolbox [31]	2014/2022	Matlab	✓
	Interactive sim. Toolbox [32]	2013/2022	Matlab	×
	Simphony [33]	May 2020/Feb. 2022	Python	✓
	SAX [34]	Aug. 2022/June 2023	Python	✓
	Photontorch [35]	2020	Python	✓
	Lumerical tools [36]	2003/Nov. 2021	—	×
	Photon Design tools [37]	2001/2021	—	×
ONN	Photonic neuron [38]	2022	Verilog-A	×
	Neuroptica [39]	2019/Apr. 2020	Python	✓
	Neurophox [40]	2019/Apr. 2021	Python	✓
	Imprecise ONN [41]	Mar. 2019	Python	✓
	Photontorch [35]	2020	Python	✓

observe that a majority of the DL and ONN frameworks are open source, which is not the case for photonic simulators. Indeed, DL and ONN tools are mainly implemented in the same language, Python, which is a widely used and user-friendly language with all of its documentation available online. It should be noted, though, that ONN implementations are not as up to date as DL and PIC simulation tools.

There is a real interest in implementing both the PIC and DL aspect of a simulation with the same language, as it facilitates the user experience. As such, full simulations working on Matlab have not been proposed yet but could be seen in the future. To the best of our knowledge, SAX and Simphony have not been exploited either in the context of ONNs. This is probably due to their lack of ability to simulate active optical devices. Indeed, the training of most ONNs requires to configure the transmission of components such as MZIs and phase shifters. As such, best candidates for the simulation and the training of ONNs are frameworks Neuroptica, Neurophox, and Photontorch, as discussed in the following.

4.2 Presentation of ONN Simulators

The ONN simulation tools listed in Table 1 all present their own particularities. In the following, we describe them independently and specify the context of their application.

Photonic neuron. In Reference [38], the authors present how Verilog-A can be used to co-simulate photonic and electronic devices in neuromorphic photonic circuits. They propose a methodology to combine discrete components such as photodetectors and phase-shifters to model more complex components such as MRRs and MZIs. The optical signal is defined by the electric field representation of light. They demonstrate the simulation of one neuromorphic neuron using a MRR to weight the neurons and the electro-optic AF presented in Reference [42]. The Verilog-A-based approach seems interesting to implement and simulate ONNs with predefined parameters. However, the lack of an interface for the DNN simulation aspect is a major drawback.

Neuroptica. The framework Neuroptica simulates ONNs using the *in situ* backpropagation technique depicted in Reference [8]. Meshes of MZIs modeled by unitary matrices are used to realize the matrix multiplication. Neuroptica is not based on any other DNN frameworks. It offers different levels of implementation:

(1) Component level: Beam-splitters, phase-shifters, and MZIs can be simulated individually.
(2) Component layer level: Build a layer of n components or an optical mesh composed of m layers of n components.
(3) Network layer level: Build layers of a DNN such as the AF or the matrix multiplication with optical components.

Neurophox. Neurophox is an ONN framework written in Python. The layers of an ONN can be built through the Tensorflow library. The various functions offered by this framework are thereby also accessible. Neurophox, like Neuroptica, is based on unitary mesh networks, implementing matrix multiplication with MZIs, yet, it differentiates itself by its training. Instead of using *in situ* backpropagation, it ensures that its optical layers can be accessed and updated by the optimizers of Tensorflow. Neurophox also provides functions to improve the mesh optimization. Those functions are detailed in Reference [43].

Imprecise ONN. In Reference [41], the authors propose an ONN architecture based on MZIs, which is more robust to component imprecision than the classic rectangular mesh [5] used by Neuroptica and Neurophox. To test it, they coded an ONN simulator based on Pytorch and made it available online. They do not directly train an ONN but rather a complex ANN. Once trained, the matrix multiplication layers are converted to the optical domain. This leads to a quicker convergence during training, as training ONNs requires more parameters to be tuned and elaborate training algorithms. As a result, while the code provided online is very application specific, the training method is easily adaptable.

Photontorch. Initially, Photontorch main goal is the simulation and optimization of photonic circuits. It relies on DL algorithms, accessible via the framework Pytorch, to optimize the parameters of a PIC. Thanks to Pytorch, the simulation of DNNs is also possible. In Reference [35], Photontorch is used to simulate a mesh of 384 MZIs to perform the recognition of the MNIST dataset. The optimization of 768 parameters (2 per MZI) takes them a few hours. An example of RC simulated by Photontorch is presented in Reference [44].

4.3 Comparison of ONN Simulators

In Table 2, we resume the advantages and drawbacks of each ONN simulator. Both the photonic neuron and the imprecise ONN are specific simulations developed in the context of a article. The first one proposes a methodology to implement electro-optic ONNs but lacks a proper DNN interface and only gives a very limited example of simulation: one photonic neuron. The second can train large models easily, thanks to its Pytorch implementation and the choice of training complex ANNs. However, its code is still very limited to the aims of the article in which it was published. Neuroptica, Neurophox and Photontorch seem to be the best alternatives for now to simulate ONNs. Even if their training is quite slow, the various levels of implementation of Neuroptica allow the user to build their ONN either using network layers directly or optical component by optical component. The Tensorflow implementation of Neurophox as well as its potential for mesh optimization renders it highly adaptable. Finally, the capacity to use DL algorithms to simulate PIC while accessing the framework Pytorch makes Photontorch very promising.

Following our observations, we conclude that Neuroptica, Neurophox and Photontorch are three interesting frameworks to build ONNs. However, the lack of update and amelioration brought to

Table 2. Advantages and Drawbacks of each ONN Simulator

Name	Advantages	Drawbacks
Photonic neuron [38]	Co-simulation electro-optic	Application specific Not publicly available No DL interface
Neuroptica [39]	Various levels of implementation	Slow training
Neurophox [40]	Tensorflow implementation Mesh optimization	Slow training
Imprecise ONN [41]	Pytorch implementation Training of complex ANNs	Application specific
Photontorch [35]	PyTorch implementation	Slow training

their code pulls downward their potential. Indeed, they do not propose any alternative for their training, which can take a significant amount of time when working with deeper ONNs.

5 ELECTRO-OPTICAL ACTIVATION FUNCTIONS

The AFs, which are intrinsically non-linear, are crucial in any ANN, since they complement data linear operations. Ideally, both linear and non-linear functions should be implemented optically in an ONN to minimize loss of resources and latency overhead. However, since the design and fabrication of such a fully optical ONN is a challenging task, there are several existing approaches that involve electro-optical AFs. We present these alternative options in this section. Although these solutions are not all optical, we show that they still offer important results in terms of speed and power consumption compared to electrical solutions.

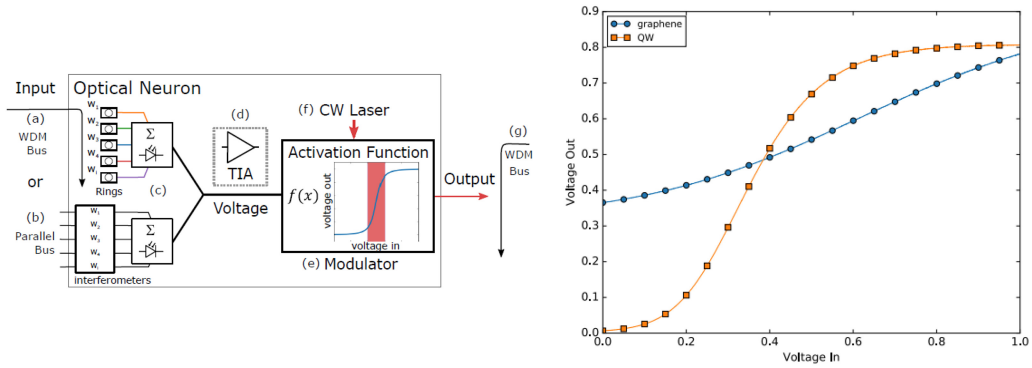
5.1 Electro-absorption Modulator

In Reference [45], the authors present an electro-optic neuron that performs both the linear and non-linear operations required for a forward propagation. First the neuron uses either MRRs or interferometers to weight and sum the inputs of the neurons. Then, a photodiode coupled to an **electro absorption modulator (EAM)** implement the electro-optic non-linearity: The transfer function of the EAM is directly used as the AF.

Figure 3(a) illustrates the optical neuron organization. Inputs are weighted either through MMRs with WDM (a) or interferometers using parallel buses (b). The weighted signals are summed and converted to a voltage with a photodiode (c). The voltage is amplified by a **transimpedance amplifier (TIA)** and is used to power the EAM. The latter absorbs the light of a **continuous wave (CW)** laser. The absorption rate is non-linearly dependent on the input voltage of the EAM, which is directly proportional to the optical power. Therefore, the light produced by the CW laser, and controlled by the modulator, possesses a non-linear relation with the input optical power. The resulting output optical power P_{out} is governed by Equation (4), with P_{in} the optical power of the CW laser (i.e., before attenuation), V_{in}, the input voltage of the modulator and, α_{dB}, the absorption in dB of the modulator:

$$P_{out} = 10^{\frac{(10\log_{10} P_{in} - \alpha_{dB}(V_{in}))}{10}}. \tag{4}$$

The operating speed of the optical device is superior to 10 GHz. The EAMs can be implemented with different materials such as graphene and **quantum well (QW)**, which leads to significant difference in the modulator absorption. Figure 3(b) reports the output voltage for a QW and graphene-based modulator for an input voltage ranging from 0.0 to 1.0 V. The transmission obtained for QW is very similar to a sigmoid function and returns values between 0 and 0.8 V.



(a) (b)

Fig. 3. (a) Optical neuron proposed in Reference [45] and (b) output voltage of an EAM based on QW (orange curve) and graphene (blue curve) according to the input voltage. Reprinted with permission from Reference [45] (2023 IEEE).

The non-linearity of the graphene is much less significant and outputs a voltage ranging from 0.4 to 0.8 V.

To determine the most fitting material authors also investigated the performance of both AFs. Keras [26] and TensorFlow [25] were used to simulate the DNN and to evaluate its performance on the MNIST dataset. The DNN model is composed of three layers of 100, 100, and 10 neurons. The AF in each neuron is defined by Equation (4). The parameter α_{dB} corresponds to the absorption, which depends on the material. Authors were able to reach up to 95% accuracy by considering laser powers as low as 5 and 30 mW for QW and graphene, respectively. Results show that the QW is more stable than the graphene as the accuracy of the QW DNN does not seem to be impacted much by the input laser power. On the contrary, the accuracy of the graphene DNN drops to 20% when an input power of 10 mW or less is used.

5.2 MRR Modulator

In Reference [42], the authors developed an optical modulator neuron that performs multiple functions: fan-in, optical-to-optical non-linearity (AF) and cascadability. They define the fan-in as the capacity of one neuron to convert several weighted inputs to one single output. As for the cascadability, it represents the possibility of chaining together multiple neurons without changing the characteristics of the signal.

The main component of the modulator is a MRR, which is responsible for the non-linearity of the neuron. The other components are balanced photodiodes that allow detection of small optical power variations while suppressing the common variations of the input signal [46]. The diagram of the electro-optical neuron is presented in Figure 4. The inputs $IN+$ and $IN-$ are two incoherent optical signals. They are converted into an electric current using a positive or negative photodetector. The electric currents are summed with a current bias I_b. Given that the source of the incoming current is either positive or negative, the following operations are realized: $I_b + i$ or $I_b - i$. The resulting current modifies the refractive index of the MRR modulator (*mod.*) using free carrier injection [47], which in turn modulates the intensity of the output signal. Finally, the output signal is the result of the modulation of a CW laser signal ($PUMP$) of wavelength λ_n by the MRR.

The experimental setup is controlled via lightlab[1] and the optical output is observed in a sampling oscilloscope. A 10 mW input pump power was used for their experiments. Different values of

[1]Lightlab is a python library specialized in remote laboratory monitoring [48].

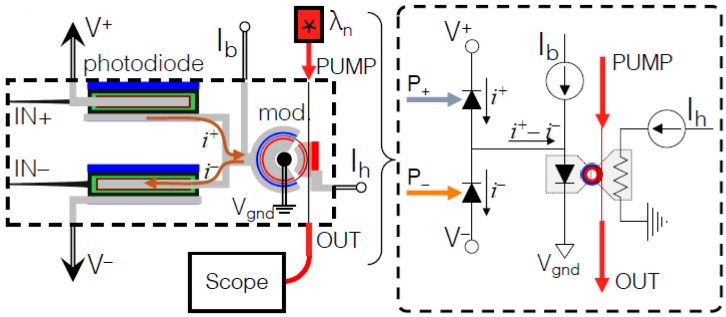


Fig. 4. Illustration of the setup of the neuron. Reprinted with permission from Reference [42] (2023 American Physical Society).

the bias current I_b result in various shapes of transmission curves, which translates into different types of AFs, including sigmoid-like and ReLU-like functions and their inverses. In addition, the MRR heater, which is controlled by the current I_h, allows the wavelength at which the nonlinearity occurs to be shifted without changing the transmission shape.

The overall loss of their neuron is governed by a total fiber-to-chip insertion loss of 18 dB, which leads to a photodetector responsivity of 0.76 A/W and causes a reduction of the pump power of 9 dB from the chip facet (−5.5 dBm) to the neuron (−14.5 dBm). The maximum tuning efficiency to maintain a chosen wavelength is 0.26 nm/mW. The setup involved two **erbium-doped fiber amplifiers (EDFA)** placed before the chip inputs before the oscilloscope. The authors noted that increasing the power of the PUMP signal would increase the gain. Their device operates at 1 GHz and its physical footprint is approximately 0.05 mm^2.

The neuron is considered cascadable if a chain of neurons of length N meets two criteria: gain cascadability and physical cascadability. In Reference [42], gain cascadability is defined by a neuron with (i) a gain of at least one for low amplitude signals and (ii) a gain greater than one for high amplitude signals. Physical cascadability is achieved when every neuron's input and output are optical and maintain the same wavelength. To show the cascadability of their neuron, the authors build an autapse circuit. An autapse is a neuron whose output optical power is fed back to itself, becoming the new input. With this circuit, the authors demonstrated that the gain is greater than unity, thus proving the gain cascadability of the neuron. More details on indefinite gain cascadability in an autapse circuit can be found in Reference [49]. As every input and output signals of the autapse are the same, physical cascadability is also confirmed. However, the authors needed to add to the feedback loop an EDFA to compensate for the high insertion losses.

In Reference [50], the architecture of the MRR modulator presented in Reference [42] is used to implement the AF in an ONN on-chip. The linear operations are realized with optical attenuators and the network is composed of two hidden layers and one output layer of 4, 3, and 2 neurons, respectively. For the image recognition of four handwritten letters, "p," "d," "a," and "t," the authors reach an average accuracy of 89.8%. As a point of comparison, they train a standard **convolutional neural network (CNN)** [51]: a DNN architecture typically used in an image recognition task [52]. They obtain 96% accuracy.

5.3 MZI-based Electro-optical Activation Function

An electro-optical architecture based on a MZI enabling non-linear transformation on optical signals is proposed in Reference [53]. The non-linearity is obtained by modulating the signal phase and amplitude. The authors postulate that the proposed design can be added or integrated with coherent ONNs to achieve, for instance, matrix multiplication.

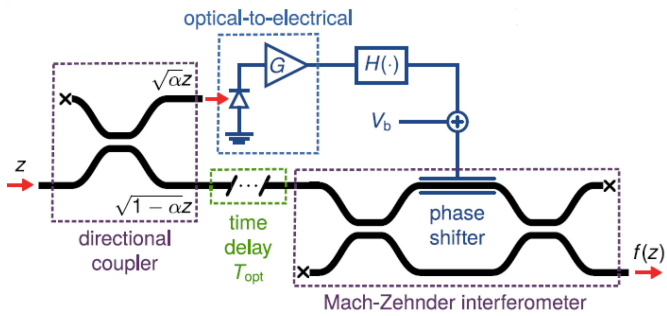


Fig. 5. Architecture of the electro-optic AF $f(Z)$. Reprinted with permission from Reference [53] (2023 IEEE).

Figure 5 illustrates the architecture. A directional coupler transmits a fraction, kZ, of the input optical signal amplitude, Z, toward a photodetector of responsivity R, which results in an electrical current: $I_{pd} = Rk^2|Z|^2$. The current is converted into a voltage with a gain G. A static bias voltage, V_b, is added to this electrical signal and the resulting voltage is used to regulate the phase shift ϕ_b of the MZI: $\phi_b = \pi(V_b + I_{pd}G)/V_\pi$. The non-coupled optical signal $\sqrt{1-k^2}Z$, propagates through the MZI where its amplitude undergoes a non-linear transformation, which depends on the phase shift ϕ_b. Eventually, this corresponds to a non-linear transformation in function on the intensity of the input signal Z, as defined by Equation (5), where $g_\phi = \pi\frac{k^2GR}{V_\pi}$ represents the phase gain parameter:

$$f(z) = j\sqrt{1-k^2}\exp\left(-j\left[\frac{g_\phi|z|^2}{2} + \frac{\phi_b}{2}\right]\right) \times \cos\left(\frac{g_\phi|z|^2}{2} + \frac{\phi_b}{2}\right)z. \tag{5}$$

Different AFs can be obtained by changing the bias voltage, since it is directly related to the phase shift. For instance, a phase shift of π would lead to ReLU-alike AF.

To validate the AF, the authors train two ONN models on two different tasks: the XOR logical operation and the classification of the MNIST dataset. The library Neuroptica [39] was used to simulate an ONN of two layers, with 4 neurons per layer, that obtains the final **mean-squared error (MSE)** [54] of 10^{-5} for XOR. For MNIST, a three-layer ONN of 16 neurons each was modelled using the neurophox library [40], which reached 94% accuracy.

Results show that a layer of 100 neurons would consume at least 10 W. This is significantly high with regards to the required 10 mW input power for solely the architecture to be functional. The power overhead is due to the use of **optical receiver amplifiers (ORA)** for the optical-to-electrical conversion of the AF. Its power consumption ranges from 10 to 150 mW. One ORA per neuron per layer is needed, causing the power consumption to escalate quickly with the size of the network. Nevertheless, they also predict that layers of 100 neurons would lead to the latency of 237 ps/layer, which is the equivalent of 121 ps/neuron, a footprint of 120 mm^2/layer, or 0.7 mm^2/neuron, and a performance of 10^{14} **multiply accumulate (MAC)** operations/s/layer, which is two orders of magnitude greater than the number of MACs that can be achieved in modern **graphics processing units (GPUs)**.

A similar solution is proposed in Reference [55], where the MZI is replaced by an MRR: a part of the optical input signal is transformed by a photodiode and used to detune the resonance of a pn-doped MRR. The remaining part of the signal is fed into the MRR. A major advantage of this

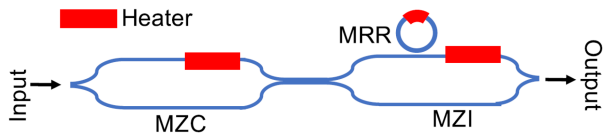


Fig. 6. Schematic of the device under test. Reprinted with permission from Reference [59] (2023 The Optical Society).

alternative is that it eliminates the need for an amplifier in the electro-optical conversion. The result is a reduction in system latency and power consumption.

5.4 Summary

From these three designs, we can conclude that the integration of electro-optical AFs is very promising. However the necessity to perform an **optical-to-electrical (O/E)** and **electrical-to-optical (E/O)** conversion at every neuron is their main weakness. Indeed, it introduces delays and increases the overall power consumption of the circuit. In References [42, 45], the E/O conversion requires a supplementary light source, either being directly the modulator in Reference [45] or an external laser in Reference [42]. In Reference [53], for the E/O conversion, only a fraction of the input optical signal is kept, which in an ONN would lead to the signal vanishing throughout the layers. To counteract this, optical amplifiers can be used, though it will increase the power consumption.

6 ALL-OPTICAL ACTIVATION FUNCTIONS

Electro-optic AFs show promising results in terms of speed and power consumption compared to modern GPUs. However, more aggressive and disruptive nanophotonic implementations [56] involving all-optical AFs may lead to further gain. In this section, we introduce different designs for the integration of all-optical AFs and their potential for ONNs. First, we present a reprogrammable AF exploiting the technology of a MRR-assisted MZI. Then, we review several implementations of AFs via SOAs: as a tanh in RC and as a reconfigurable sigmoidal AF. Finally, we consider the saturable absorption effect and its available implementations.

6.1 MRR-assisted MZI

The phase and amplitude of MRRs may change non-linearly depending on the optical input intensity due to the following optical effects: the Kerr effect, **two-photon absorption (TPA)**, **free carrier absorption (FCA)**, and **free carrier dispersion (FCD)**. These effects are explained in detail in Reference [57]. In References [58, 59], the authors exploit the potential for non-linearity of MRRs on silicon and propose an all-optical and reconfigurable AF, illustrated in Figure 6. A **Mach-Zehnder coupler (MZC)** is used upstream of the circuit, followed by a MZI equipped with a MRR on one of its arms. The FCD effect, which is the prevalent non-linearity in silicon waveguides, is triggered via cavity build up in the MRR. The interferences of the MZI depend on the MRR's phase. Consequently, the output signal power is non-linearly dependent on the input signal power. Furthermore, heaters placed on the MRR, the MZC, and the MZI enable control of (i) the shift between the MRR resonance wavelength and the input signal wavelength and (ii) the MZC coupling ratio and the MZI coupling ratio, respectively. This enables the configuration of various AF shapes, as discussed in the following.

Simulations are carried out to evaluate the impact of the MRR wavelength detuning $\Delta\lambda$ and the coupling ratio r of the resulting AF. For example, Scenario a reported in Figure 7, with $\Delta\lambda = 0.05$ nm and $r = 0.65$, corresponds to Clamped ReLU. In this scenario, the signal is initially

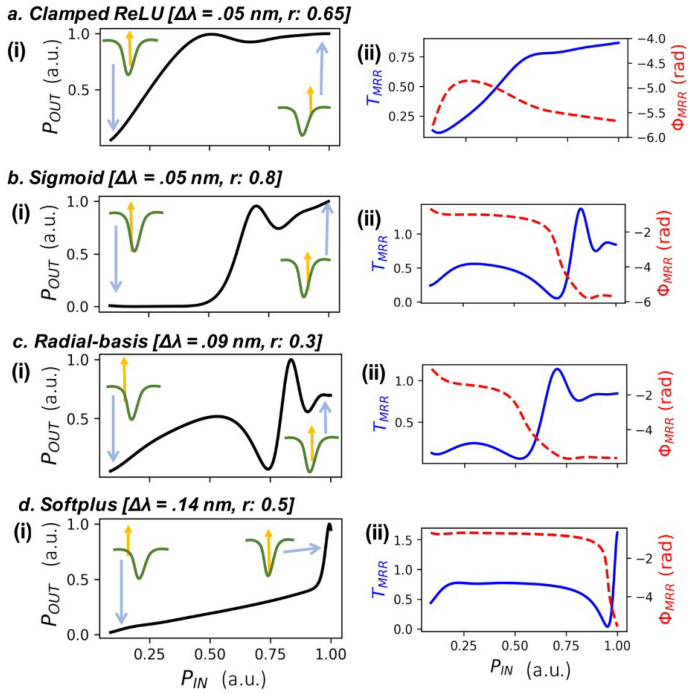


Fig. 7. Figure representing the different type of AFs simulated: (a) clamped ReLU, (b) sigmoid, (c) radial-basis, and (d) softplus. (i) Transfer functions with the output power plotted in function of the input power. At the start and end of each graph is specified the shift between the MRR resonance wavelength (green) and the input signal wavelength (yellow). (ii) The MRR transmission (T_{MRR}) and the non-linear phase change (Φ_{MRR}) are plotted in function of the input power. Reprinted with permission from Reference [59] (2023 The Optical Society).

red-detuned leading first to a linear increase in the output power. The generation of free carriers results in a non-linear proportional phase change (Figure 7(a)(ii)). Once the maximum phase change is reached, both the coupling into the MRR and the non-linear phase decrease due to the blue shifting of the resonance. Using the same hardware, the authors show that different settings for $\Delta\lambda$ and r lead to (b) Sigmoid, (c) radial basis [60], and (d) softplus [61], which is a smooth version of ReLU. Furthermore, the authors claim that the linear region of the clamped ReLU and Softplus transfer functions can be adapted by changing the MZC coupling ratio.

For the experimental setup, the input signal is coupled by free-space coupling to the DUT. An EDFA is added to the input of the DUT to offset the fiber-to-chip coupling loss of 8 dB. This setup is controlled remotely by lightlab. The power of the signal is measured by photodetectors and displayed on a sampling oscilloscope. In addition to the parameters tuned during the simulation, the authors adjust the coupling ratio of the MZI. As a result, they obtain a wide range of shapes for their transfer function. All of them are still classified among the four categories presented in Figure 7.

The authors use the clamped ReLU and the sigmoid function to challenge the integration of the MRR-assisted MZI as an AF into an ONN. From their experimental measurements, two functions are approximated by spline interpolations. These functions are implemented directly in the desired ANNs on a computer with the Pytorch library [27]. Two ANNs are built for two different classification tasks. The first network, a DNN, is composed of one hidden layer of two neurons and one

output layer of one neuron. The AF used is the sigmoid approximation. The DNN is trained on a XOR binary classification task and obtains 100% accuracy. The second network is a CNN. The activation function used is the clamped ReLU. The CNN is trained to classify the MNIST dataset and obtains 94% accuracy on the test set.

They increased the speed of their device from 400 Mbit/s to 2.5 Gbit/s by reducing the carrier lifetime [62]. They need to use an EDFA at the beginning of the circuit to compensate the coupling loss of 8 dB and to be sure to trigger the non-linearity of the MRR. For this purpose, the signal is amplified to 25 mW. An additional 2 dB loss is taken into account for the device insertion losses. For now, the phase efficiency of their thermo-optic heaters is 25 mW/π, but authors claim that it could reach 1.3 mW/π by using thermal isolation trenches [59]. The physical footprint of one AF is around 0.3 mm^2.

6.2 SOA-based

In this subsection, we review the existing implementation of SOAs-based AFs. First, we highlight the potential of SOA in RC. Then, we present a reconfigurable sigmoid AF designed using SOAs.

6.2.1 SOAs in Reservoir Computing. The benefits of SOAs as AFs seem to have been first studied with the implementation of optical RC. As mentioned in Section 2, RC is a type of ANN that connects its neurons in a recurrent manner, and each neuron performs a non-linear transformation on its input. This transformation is usually a tanh. A SOA is a good candidate to implement optically this operation. Indeed, it can be observed that above a sufficient optical input power, the gain of the SOA decreases: a phenomenon called gain saturation. This results in a transfer function similar to the positive part of the tanh [63].

In Reference [63], the focus is on the integration of coupled SOAs as AFs in photonic RC. At the time of publication of the paper, the software implementations of RC were quite slow and would greatly gain from an optical implementation. However, the goal of the authors is to showcase the potential of SOAs in optical RC, rather than optimizing its performance in terms of speed and power. The behaviour of the SOAs is modelled with the standard travelling-wave equations [64]. When the internal losses are neglected, the output power P_{out} and the output phase ϕ_{out} can be calculated by the following equations:

$$P_{out}(\tau) = P_{in} \exp h(\tau), \tag{6}$$

$$\phi_{out}(\tau) = \phi_{in} - \frac{1}{2}\alpha h(\tau), \tag{7}$$

with α being the linewidth enhancement factor, P_{in} the input power, ϕ_{in} the input phase and τ the spontaneous carrier lifetime. The gain integrated over the length of the SOA, $h(\tau)$, is dependent on the input power of the amplifier as well as its saturation power. As they use SOAs with anti-reflection coating on their facets, they neglect the influence of reflections.

The authors perform numerical simulations of photonic RC and represent the optical non-linearities using the toolbox presented in Reference [65]. The model is validated on a pattern recognition task (signals with triangular and square waveforms) and compared to the same model using tanh instead of the SOA function. Results reveal a slightly lower **error rate (ER)** with the implementation using SOAs. They use for the simulation a mesh of 25 SOAs. The delay between each SOA is 6.25 ps and the processing speed of one SOA is 0.5 GHz.

In Reference [66], the authors study in more depth the impact of the design and the fabrication of the device proposed in Reference [63]. They train their architecture to classify spoken digits from 0 to 9 from the ear model presented in Reference [67]. They introduce babble noises to the initial dataset of 500 samples. They compare their optical architecture to a similar one with tanh

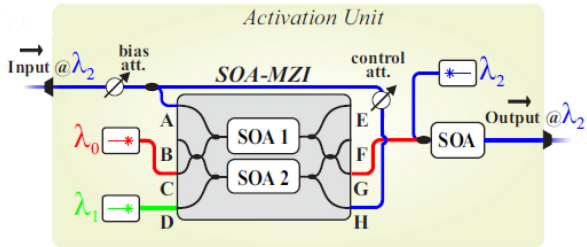


Fig. 8. Diagram of the activation unit. Reprinted with permission from Reference [68] (2023 The Optical Society).

AFs. The latter only use positive weights to compare with an optical reservoir whose values can only be positive as they are defined by the optical power of the signal. To compare these two architectures, they use a **word error rate (WER)**: the number of incorrectly classified samples on the total number of samples. The best WER obtained by a classical reservoir of tanh on this task is of 7.3%.

In terms of characteristics of the circuit, the input signals are continuous. The architecture can work with either coherent or incoherent light but achieve better results with a coherent circuit. They identify two main parameters that have an influence on the obtained WER: the delay of the interconnections between the SOAs and the phase delay. They found out that each application has an optimal phase delay. For instance, classification of spoken digits leads to an optimal phase delay of 190 ps, which is approximately half the duration of the audio signals. Then the performance of the optical reservoir depends essentially on their ability to control the phase change. In the case of a coherent circuit with delays of 190 ps, if the phase cannot be controlled, they still obtain results comparable to classical reservoirs. However, with perfect tuning of the phase change they outrun it. Finally, the physical footprint for a setup of 12 SOAs is 16 mm^2.

6.2.2 Reconfigurable SOAs to Build a Sigmoidal AF. In Reference [68], the implementation of a sigmoid AF is demonstrated using the saturation behaviour of SOAs. To achieve this, two SOAs are used to saturate the output signal for different ranges of input power, mimicking the asymptotes of a sigmoid. In their method, the authors compare experimental demonstration results with an analytical model they proposed. By using their method, the authors illustrate how the circuit can be configured. The following further detail their approach.

The activation unit is composed of a deeply saturated differentially biased SOA-MZI (cross phase modulation) [69] followed by a SOA-XGM (Cross gain modulation) **WC (wavelength converter)**. Figure 8 represents the optical AF implementation. The authors use a pattern generator to transform the CW input of the ONN into a pulse signal at λ_2. The resulting control signal is split into two identical streams by a directional coupler. The first stream enters port A of the SOA-MZI and the second one is attenuated and then sent to port H as a counter-propagating wave. Two CW signals of wavelength λ_0 (port C) and λ_1 (port D) and high optical power levels are sent through port C and D, forcing SOA 1 and SOA 2 into their saturated regime. Consequently, the output signal of the SOA-MZI is the inverted and saturated control signal with a wavelength of λ_0. A third (regular) SOA is used as a wavelength converter to restore the signal and the wavelength by inserting an additional input CW signal of wavelength λ_2.

The AF is measured experimentally with an oscilloscope, using 400 different input pulses with their pick powers ranging from −25 to 0 dB. They consider that the gains of the SOA 1 and 2 are recovered before the following pulse is inserted into the AF unit. They obtain a sigmoid like experimental transfer function, as shown in Figure 9 with the black dotted line. They find that this

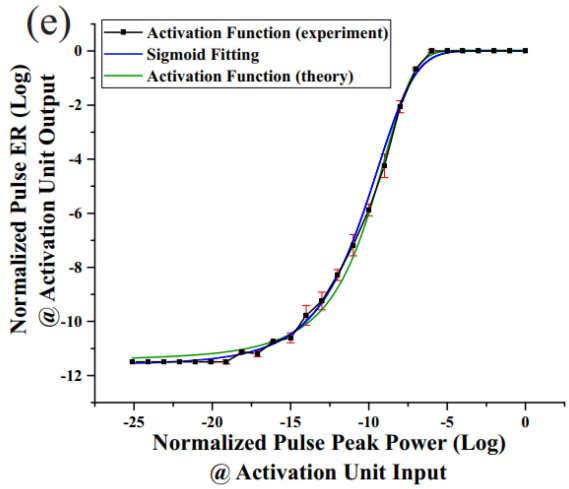


Fig. 9. Transfer function of the activation unit. Reprinted with permission from Reference [68] (2023 The Optical Society).

type of AF can be approximated with the following equation: $f(x) = A_2 + \frac{(A_1 - A_2)}{1+\exp(x-x_0)/d}$. To fit the AF in Figure 8, they use $A_1 = 0.060$, $A_2 = 1.005$, $x_0 = 0.145$, and $d = 0.033$, which gives the blue curve.

They use a theoretical approximation (green line in Figure 9) to investigate the impact of the control attenuation factor and the biasing attenuation factor and conclude that they, respectively, influence the slope and the position of the curve along the x-axis.

For the overall configuration of their activation unit, SOA 1, SOA 2, and the output SOA are, respectively, driven by direct currents of 240, 280, and 300 mA. Two CW optical signals of 3.5 and 4.5 mW go through port C and D. Finally, a CW signal of optical power 0.009 mW is added to the input of the output SOA. During the experimental demonstration of the neuron, the authors inject a periodic signal with 100 ps long pulses in the activation unit. Following the time traces plotted in Figure 5 of Reference [68], the AF operates at a speed of 2.5 GHz.

6.3 Saturable Absorber

In this subsection, we introduce the saturable absorption effect. We present several materials suitable for the implementation of saturable absorption as an AF in an ONN. A **saturable absorber (SA)** is an optical component that absorbs part of an electromagnetic signal, thereby decreasing its intensity. This absorption rate decreases with the increase in intensity of the input optical signal. Thus, the signal transmission of the SA is non-linear and possesses a saturated and non-saturated region [70].

6.3.1 Atomic Vapor. In Reference [70], the authors propose an ONN architecture that can perform both forward and backward propagation optically. The scheme is compatible with both integrated and free-space platforms. They use a SA to achieve the AF. Equation (8) represents the transfer function of the absorbent medium with $E_{p,in}$ and $E_{p,out}$, being the input and output electric field and α_0 the optical depth, which corresponds to the opacity of the medium. All fields are normalized by the saturation threshold:

$$E_{p,out} = exp\left(-\frac{\alpha_0/2}{1+E_{p,in}^2}\right)E_{p,in}. \tag{8}$$

The derivative of Equation (8), necessary to apply the gradient descent algorithm, is equal to the linear response of the SA up to a constant factor. Therefore, the backpropagation can be realized using the same optical components as the forward propagation. It should be noted that this approximation is not valid for the entire domain of the SA response. For the system presented in this article, the approximation holds within its non-linear region.

The authors observe that the optical depth of a SA directly influences the shape of its transfer function and, more precisely, its degree of non-linearity. Indeed, for an optical depth of 0 the function is linear. They carry out a numerical simulation of an ONN with the Pytorch library. The optical DNN is composed of two hidden layers of 128 neurons activated by SAs and is trained on the MNIST dataset. The best accuracy, 98.0 ± 0.2%, is observed for optical depths ranging from 10 to 30. As a point of comparison, they train a similar DNN using ReLUs as AFs and reach the same accuracy.

They specify that interlayers of SOAs are necessary to maintain the field amplitude in deep ONNs, specifically when using passive optical components for the SA. The authors point out that SOAs also have a non-linear effect, called saturable gain, similar to saturable absorption. Therefore, the SOAs could completely replace the SAs in their system.

For one neuron with SA built on the $^{87}RbD^2$ line, the total input power necessary is approximately 5×10^{-4} mW and the speed at which it is processed is 10 MHz. The energy used by the backpropagation is negligible. However, the forward propagation costs about 10 pJ every time the light passes through.

6.3.2 C_{60} Molecules. In Reference [56], a reverse SA is presented, in which the absorption increases with the input light intensity. The authors exploit the non-linear response of C_{60} molecules. The transfer function is obtained via a **finite difference time domain (FDTD)** solver. Specifically, the solver simulates the response of C_{60} molecules dispersed in polyvinyl alcohol with a concentration of 10 mM and the resultant set of points is approximated using quadratic functions. The AF is tested in a three-layer NN to classify MNIST. Over 99% accuracy on the test set is obtained. The processing speed of one AF is 10^3 GHz with only a few ps of latency.

6.3.3 Monolayer Graphene. Another component that shows potential of saturable absorption is graphene, or more precisely monolayer graphene. In Reference [71], the authors use monolayer graphene as a SA in a mode-locked laser and observe promising results, such as a capacity of saturation at a low excitation intensity: 0.53 M.W.cm^{-2} and a large modulation depth of 65.9%. The modulation depth is the maximum change of absorption caused by an input light [72]. They reached a speed of 5 MHz during experiments with the latency of a few ps.

6.4 Summary

The MRR-assisted MZI has the ability to perform several types of AFs such as ReLU and sigmoid. The circuit can be configured either before manufacturing, and therefore using passive components, or after using active components. Passive components have the advantage of not requiring any power consumption. Neither the MRR-assisted MZI nor the SA-based solutions use any additional power source for the non-linear transformation. On the one hand, the power consumption is limited to the generation of the input signal of the circuit, but, on the other hand, it implies that the amplitude of the optical signal decreases as the number of layers increases. Furthermore, these two types of architecture require a minimum signal amplitude to trigger the non-linearity, which prevents from designing ONNs with multiple layers. Two solutions to this problem have been proposed: (i) using a SOA to both amplify the signal amplitude and realize the AF and (ii) exploiting the capacity of parallelization of optical circuits. In References [63, 66], a single SOA is used as a neuron and perform the AF, but the AF is bound to be a tanh and has only been tested in the case

of RC. The authors in Reference [68] present a sigmoid AF built with SOAs and a MZI. However, it requires the use of three SOAs per optical AF unit. It is important to mention that using either one or three SOAs per AF in a deep NN may cause the power consumption to increase consequently.

Using a parallelized optical AF unit to process multiple inputs simultaneously would help to reduce the overall power consumption of the circuit. In the free space implementations proposed in References [73, 74], spatial parallelization takes place inside a single component. The authors exploit the property of saturable absorption and **electromagnetically induced transparency (EIT)** of atomic vapor, respectively. In both implementations, the non-linear transformation only takes place locally in the medium. As a consequence, multiple light rays undergo independent non-linear transformations when they are spatially distant from each other. However, these implementations also have disadvantages. First is their physical footprint: for ONNs of two [73] and three [74] layers, the authors, respectively, use five and four lenses with focal lengths of up to 30 cm. Additionally, the SLMs needed for the linear operations are quite power hungry and slow. They consume 10 W [73] of additional power per layer (minus the input layer) and commercially available SLMs are usually slower than computer implementations [74].

7 COMPARATIVE STUDY

To evaluate the performance of the optical and electro-optical AFs presented in this article, we propose two levels of comparison: the device-level comparison and the network level comparison. For the first one, we focus on the performance of each optical and electro-optical non-linearities when integrated in a PIC. As for the network level comparison, we highlight the efficiency of these non-linearities when used as an AF in an ONN.

7.1 Device-level Comparison

The device-level comparison is essential to investigate the scalability of existing optical AFs. Indeed, the non-linear function is a recurring function in a DNN. The deeper the circuit, the greater the impact of any latency or any power consumption overhead. For each AF, we consider the following criteria:

- The input signal power in mW.
- The insertion losses in dB.
- The speed in GHz.
- The footprint in mm^2.
- Whether the reported measurements were experimentally demonstrated.

The data is reported in Table 3. For each AF, the input signal power tested was limited to a set or interval of power levels. It gives an indication of the power necessary to trigger each non-linearity. The lowest reported results are obtained for SAs with an input power of 5×10^{-4} mW. For "insertion losses," we report the losses caused by the insertion of one signal into the AF. The high loss of 6 dB for [63] accounts for the internal losses of the SOA. Next, the speed refers to the frequency at which optical signals are transmitted through one AF. The SA [56] is simulated with the highest frequency, which is 10^3 GHz. Experimentally, a maximum frequency of 3.33 GHz is reported for an SOA [68]. Then, we indicate the physical footprint: the space taken up by an AF when integrated on-chip. Finally, we note that four of the AFs presented in this survey were studied experimentally: one electro-optic [42] and one for each of the technology presented: MRR-MZI [59], SOA [68], and SA [71].

Overall, electro-optical AFs seem competitive with all-optical technologies: the experimentally demonstrated MRR-based AF [42] reports the lowest physical footprint. The speed of the MZI [53] and EAM [45] is only surpassed by a SA [70]. To take into account the power consumption in

Table 3. Characteristics of Studied Optical AFs/Neurons

Technology		Ref.	Input signal power (mW)	Insertion losses (dB)	Speed (GHz)	Physical footprint (mm^2)	Exp. results
Electro-optic	EAM	[45]	5 (QW) 20 (graphene)	–	>10	–	No
	MRR	[42]	10	(1)	1	0.05	Yes
	MZI	[53]	[0.1, 10]	–	10	0.7	No
All optical	MRR-MZI	[59]	[14, 40]	2(2)	0.1	0.3	Yes
	SOA	[63]	≤5	6 (3)	0.5	–	No
		[66]	≤1	(3)	–	16 for 12 N	No
		[68]	{1, 2, 3.4, 7.4}	–	2.5	–	Yes
	SA	[70]	5×10^{-4}	–	10(4)	–	No
		[56]	1×10^{-3}	–	10^3	–	No
		[71]	$\leq 5\times10^{-3}$(5)	5.10	0.005	–	Yes
			$>5\times10^{-3}$(6)	0.14			

(1)Fiber-to-fiber insertion loss of 18 dB: 9 dB per grating coupler.
(2)Fiber-to-chip coupling loss of 8 dB + device insertion of 2 dB.
(3)SOAs' gain can be used to compensate for the losses.
(4)With an excited state lifetime of 26 ns.
(5,6)Respectively, input powers for the linear state and saturation state.

our assessment, we compare the all-optical AF presented in Reference [59] and the electro-optical MZI [53]. Both implementations rely on a MZI to which they apply a non-linear phase change with respect to the input signal. The power consumption of the electro-optic AF is dominated by the ORA in the optical-to-electrical conversion. It can be averaged to a 100 mW per neuron. In terms of input signal, only 0.1 mW is needed to trigger the non-linearity. In comparison, the MRR-MZI exhibits a non-linear behaviour for input power signals ranging from 14 to 40 mW. However, it does not have energy expenditures comparable to the one caused by the ORA. The power consumption of the reprogrammable shifters in both implementations can be considered negligible [53]. The effect of the ORA on the electro-optical neuron's consumption is significant. In a DNN involving a large number of neurons, the use of an ORA in each AF would significantly increase the total energy consumption. Overall, for the MZI technology, the all-optical implementation is less energy consuming but slower. We lack information to properly compare other all-optical and electro-optical AFs, in particular concerning the energy required to maintain the non-linearity.

As can be observed in Table 3, the data reported for each AF is highly inconsistent. In addition, only four of them propose an experimental demonstration of their device. We believe those works are only the first step toward the deployment of each technology. Further analysis should be performed to increase the performance of the circuit by either replacing some of the non-linear components or changing the physical characteristics of an optical component (e.g., the radius of the ring).

So far, our focus has been on the study of individual AFs. However, a layer of a DNN consists of a multitude of AFs. For instance, for the SA in Reference [70], the authors determine that the necessary input power for 1,000 neurons is 0.5 mW. In Reference [53], a thorough analysis of

their AF lead to the following results: for one layer of a hundred neurons, the power consumption would be 10 W, the latency 237 ps, the footprint 120 mm^2 and the speed would increase from 10^{10} MAC/s for one neuron to 10^{14} MAC/s. We observe that the input power needs to be multiplied by at least the number of neurons of the layer to trigger the non-linearity of each AF. When passive components (e.g., SAs, MZIs, and MRRs) are employed, the repetitive application of the non-linear function decreases the amplitude of the optical signal throughout the layers of an ONN. Using SOAs as AFs can compensate for the amplitude and power losses. However, the power consumption may drastically increase. To overcome this limitation, SOAs and passive components can be employed alternatively in the circuit.[2] Scalable AFs are essential for the implementation of deep, and therefore more accurate, ONNs. Besides the required power threshold, the scalability of an AF is ensured by maintaining the integrity of the optical signal during the non-linear transformation. However, except for the autapse [42], no complete study of scalability is proposed.

7.2 Network-level Comparison

In this subsection, we investigate the efficiency of each optical non-linearity to implement an AF. In Table 4, we present the ONNs in which the optical AFs were tested. Those ONNs are defined by the following criteria:

- The shape of the AF: some optical non-linearities may be compared to existing AFs (e.g., sigmoid, ReLU, tanh, etc.).
- The model of the AF used during the simulation. It is possible that no direct equation of a transmission function for a given technology exists. When it is the case, we specify if the transmission curve was **experimentally measured (EM)** or simulated with a solver (e.g., FDTD) and how the final equation was **approximated (app.)**.
- The dataset on which the network is trained.
- The size of the network. It is represented by the number of neurons of each layer, from the first hidden layer to the output layer. We ignore the input layer as it does not employ AFs nor weight multiplication. We do not detail the CNN architecture of Reference [58] but rather use a similar computer-based CNN [51] as a point of comparison. For References [63, 66] the number of neurons represents a "pool" of neurons instead of a classic ANN layer.
- The simulation tool used to implement and train the network.
- The percentage of accuracy (except in some cases where the metric is therefore specified) obtained with the simulation tool.
- Experimental demonstration: We indicate whether the data were obtained by experimental demonstration.

As a reference for the MNIST dataset, we present here the accuracy obtained on the validation set of three computer-based ANNs: two DNNs and one CNN. The highest reported accuracy with a fully connected DNN is 99.65% [76]. However, this network is composed of a significant number of neurons: five hidden layers of 2,500, 2,000, 1,500, 1,000, and 500 neurons and an output layer of 10 neurons. For a fair comparison, we present a smaller ANN with two hidden layers of 100 neurons and an output layer of 10 neurons, which reaches 98% accuracy with ReLU and 97.9% with sigmoid [77]. We also present a computer-based CNN, which reaches 98.9% accuracy using the LeNet-4 architecture [51].

We can see that the ONNs proposed in References [56, 70] are very competitive with the computer-based DNNs presented here. Reference [56] is even more promising given that a single

[2]In Reference [75], authors explore the non-linearity caused by the Kerr effect in a crystal and how this could replace the SOAs units of an ONN, as in Reference [63].

Table 4. Table Highlighting the Potential of Integration in ONNs of the AFs Studied in This Article

Technology		Ref.	Simulation parameters					Accuracy[3]	Exp. demons.
			AF		Network				
			Shape	Model	Dataset	Size	Tool		
Electro-optical	EAM	[45]	Sigmoid	Equation (4)	MNIST	100/100/10	TensorFlow	95%	No
	MRR	[50]	ReLU	NA[2]	Image recog.	4/3/2	NA[2]	89.8%	Yes
	MZI	[53]	ReLU	Equation (5)	XOR	4/4	Neuroptica	MSE<10^{-5}	No
					MNIST	16/16/16	Neurophox	94%	
All optical	MRR-MZI	[59]	Sigmoid	EM + spline app.	XOR	2/1	Pytorch	100%	No
			ReLU		MNIST	CNN		94%	
	SOA	[63]	Tanh	Equations (6) and (7)	Pattern recog.	25	Matlab [65]	ER: 2.5%	No
		[66]			Ear model	81		WER: 7.3%	No
		[68]	Sigmoid	NA	NA	NA	NA	NA	NA
	SA	[70]	Original[1]	Equation (8)	MNIST	128/128/10	Pytorch	98%	No
		[56]	Original[1]	FDTD + quad. app.	MNIST	100/10	TensorFlow	99.6%[4]	No
		[71]	Original[1]	NA	NA	NA	NA	NA	NA
Computer based		[76]	Tanh	–	MNIST	2,500/2,000/ 1,500/1,000/ 500/10	–	99.65%	–
		[77]	ReLU	–	MNIST	100/100/10	–	98.0%	–
			Sigmoid					97.9%	–
		[51]	Tanh	–	MNIST	CNN	–	98.9%	–

[1]We call the AF "original" when it cannot be compared to existing AFs (e.g., sigmoid, ReLU, tanh...).
[2]The ONN proposed in Reference [50] is completely simulated on-chip.
[3]When nothing is specified, the percentage displayed is the accuracy of the NN.
[4]Data extrapolated from Figure 5 of Reference [56].

hidden layer of 100 neurons reaches 99.6% accuracy. It should be noted that this level of accuracy seems to have been achieved in an isolated case. Training and testing the ONN several times on a randomized dataset and averaging the resulting accuracies would increase the fidelity of the prediction. Compared to the all-optical implementations, the electro-optical NNs do not perform as well on MNIST, having a maximum accuracy of 95%. However, all of these AFs rely on simulation models purely theoretical. The only experimentally measured AF is the MRR-MZI [59]. Its approximation is used in a CNN and trained on MNIST. The accuracy obtained is 94%, 4.9% lower than the computer-based one. Both CNNs have two convolution layers, two sub-sampling layers and two fully connected layers laid out in the same way. The main difference is that LeNet-4 uses tanh as its AF.

For datasets other than MNIST, XOR yields the best results. However, this dataset is too simple for a performance comparison to be based on. It is more useful as a first validation step of an ONN with AF, as it cannot be classified without non-linearity. In the case of RC for References [63, 66], the ONNs showed both comparable results to similar computer-based reservoirs. The ONN in Reference [50] is trained on an image recognition dataset different from MNIST. As a reference, the authors trained a CNN on the same task, achieving 96% accuracy. Nevertheless,

they acknowledge that the architecture of the CNN is remarkably larger. It must be noted that the simulation of this ONN is completely experimental. Therefore the accuracy can suffer from noise, for example from the electro-optic transformations.

Overall, it is tedious to identify the main cause for the performance gap of the simulated ONNs. Certainly, it could be due to the optical AF itself and its approximation. However, the performance may also be impacted by the choice of the simulation tool. Indeed, Neuroptica and Neurophox both implement their matrix multiplications using MZI models and therefore adapt their training algorithm accordingly. This may be one of the reasons for the lower accuracy of the electro-optical MZI [53]. Another factor may be the optimization of the hyperparameters used for the training of each network. Nonetheless, the optimization methods were not shared. Consequently, we assume the hyperparameters to have been carefully selected for an optimal accuracy.

We observe a gap between the small optical circuits used for experimental demonstrations and the simulated deeper ONNs used to solve tasks such as MNIST. The first one prioritizes the accuracy of the optical representation at the expense of the depth of its architecture. On the contrary, deeper ONNs are essentially simulated as DNNs with unique AFs, without proper study of the optical implementation. Advanced tools for the simulation of ONNs could help reduce this gap.

8 KEY RESEARCH DIRECTIONS

It is undeniable that photonic integrated circuits have the potential to accelerate NNs. They ensure high bandwidth and low power consumption. However, the research around the integration of ONNs is only starting and there is still progress to be made. In this section, we propose different research directions to consider for the development of efficient optical AFs. First, we present the advantages of reconfigurability in integrated ANNs and give the example of four reconfigurable and optical AFs introduced in this survey. Then, we point out the necessity to develop more comprehensive tools to test and compare ONNs. Finally, we discuss how WDM could impact the performance of ONNs.

8.1 Reconfigurable AFs

The main objective of reconfigurable integrated ANNs is to exploit the advantages of parallelization offered by hardware implementations, compared to software implementations, while remaining flexible. Many articles focus on the implementation of reprogrammable FPGAs for ANNs [78–80]. The ability to reconfigure the circuit is relevant for hardware implementations in the following cases:

(1) The dataset on which the ANN has been trained evolves.
(2) A better ANN architecture has been found in the state-of-the-art for the given task.
(3) We want to use the same circuit for several tasks to optimize space.
(4) We want to train the ANN directly on-chip.

Those apply for the reconfiguration of the AF too. For CNNs for example, the AFs of choice used to be tanh and sigmoid. However this recently shifted to ReLU and ReLU-like functions [81]. Research is always under way to find more suitable and more efficient AFs. In this context, an interest in research is the trainable AFs: AFs that can be optimized during the training process of the NN [82]. This approach could lead to a multitude of AFs being used in the same circuit, further justifying the need to use reconfigurable AFs. Some of the papers presented in this article provide the possibility to reconfigure their AF. We display these papers in Table 5, where we specify for each technology (i) the configuration method to modify the shape of the AF, (ii) the particular configurations used by each paper, and (iii) the resulting AF shapes.

Table 5. Reconfigurable AFs

Technology	Ref.	Configuration method	AF configurations	AF shape
MRR	[42]	Electrical bias: I_b Thermal bias: I_h	Q factor and depth: I_b Wavelength selection: I_h	Sigmoid ReLU RBF[(1)] Quadratic
MZI	[53]	Phase shift of the MZI: ϕ_b	$\phi_b = \{1.0\pi, 0.85\pi\}$	ReLU
			$\phi_b = \{0.0\pi, 0.5\pi\}$	Clipped[(2)]
MRR-MZI	[59]	Coupling ratio of the MZC: r Wavelength detuning[(3)]: $\Delta\lambda$	$\Delta\lambda$ = 0.05 nm; r=0.65	Clamped ReLU
			$\Delta\lambda$ = 0.05 nm; r=0.8	Sigmoid
			$\Delta\lambda$ = 0.09 nm; r=0.3	RBF
			$\Delta\lambda$ = 0.14 nm; r=0.5	Softplus
SOA	[68]	Control attenuation factor: af Bias attenuation factor: b	Slope of 7°: $af \in [0, 0.25]$ Bias of 4.5 dB: $b \in [0, 0.6]$	Sigmoid

(1) Radial Basis Function.
(2) Low input amplitudes lead to high output amplitudes and vice versa.
(3) Between the input signal and the MRR resonance.

The degree of reconfigurability differs from one technology to another. For example, in Reference [68], only one type of AF is programmed: a sigmoid. However both the slope and the bias of the function may be configured to tune the function. In contrast, the MRR-MZI [58] can be programmed to carry out four distinct AFs, but each AF has a very limited range of adjustment.

In situ training of an ONN is becoming more and more accessible. An integrated reconfigurable AF on-chip is essential to provide a full range of updateable parameters for the ONN. In fact, the type of AF used is an important hyper-parameter to achieve optimal accuracy. A reconfigurable AF with a broad range of AF shapes such as the MRR-MZI is a promising technology.

8.2 Simulation Tools for ONNs

From our comparative study, we notice a significant heterogeneity in terms of simulation level: from the all-theoretical simulation to the complete integration of an ONN on-chip. This is one of the factors contributing to the difficulty of comparison of the studied AFs. In addition, recent articles implementing optical non-linearities through the use of lithium niobate waveguides [83, 84] or based on homodyne detection [85] open new research directions. A comprehensive and accessible simulation tool for ONNs would allow the user the following projections:

- The performance comparison of different optical AFs implemented into the same network while trained on the same dataset.
- The analysis of the non-linear response of an optical component while modifying its physical characteristics, such as the radius of a ring or the phase shift in a MZI.
- The analysis of an optical AF when one of its optical components is interchanged (e.g., a SOA replaced by a crystal).

Given the diversity of the optical AFs in the state-of-the-art, detailed analysis such as the one presented above are essential to determine their strengths and weaknesses.

We introduced a list of ONNs simulators in Section 4, but except for Neurophox and Neuroptica with the electro-optical MZI [53], none were used for the ONNs presented in Table 4. Most of the

simulations were performed with DNN simulators with a more or less precise approximation of the AF. This allowed the authors to simulate larger networks at the cost of the accuracy of the optical representation. However, the experimental demonstrations of AFs implemented on-chip were either limited to the AF itself or to a very small network. Frameworks such as Neuroptica, Neurophox or Photontorch seem to be a good trade-off between these two extremes. However, their training speed is significantly slower than other DNNs simulators. The method employed in Reference [41], where the weights of the networks are complex during training and then converted to optical representation of MZIs, allows for faster training convergence. Overall, a good compromise could be to decompose the simulation of ONNs in two phases: the training phase and the test phase. During the first phase, the optical representation is simplified and the training speed is prioritized. Once the hyper-parameters of the network are learned, more complex models can be used to implement the various optical components during the test phase. For this second phase, libraries such as Photontorch, Symphony, or SAX can implement and analyse the resulting ONN circuits.

8.3 Wavelength Division Multiplexing

We notice that the use of nanophotonic is partly motivated by the potential for WDM to increase the processing speed of ONNs while decreasing the overall power consumption and physical footprint. As such, WDM can be used to build highly parallel matrix multiplications [86]. In the context of a complete ONN, after the matrix multiplication, multiple signals would arrive at one neuron to be transformed independently by the same AF. However, the current technology does not allow it. The different AFs seen through this article are based on components that react differently given the input intensity. At a given time t, these components cannot exhibit different behaviour according to entries of various intensities. There is an exception in the case of the two free-space implementations briefly presented in the summary of Section 6: References [73, 74]. As the non-linear component is a gas, if each input crosses it at a different point, given a minimum distance between them, they can be transformed independently. However, one must be careful of the total space that could be needed to transform at the same time multiple input vectors or matrices. To conclude, non-linear components such as atomic vapor, or other types of gas, show great potential toward the implementation of fully parallel ONNs. Articles such as References [87, 88], portraying examples of EIT on-chip, are a first step toward this direction.

9 CONCLUSION

In this work, we have reviewed state-of-the-art technologies and implementations of Activation Functions (AFs) in Optical Neural Networks (ONNs). In our comparison, we assume an optimal ONN training and, for each type of AF, we focus on the available information. To highlight their pros and cons, we have evaluated (i) the computation speed, (ii) their impact on the ONN accuracy, (iii) the losses and conversion required when crossing an AF, and (iv) the power requirement to trigger the AFs. We classified the AFs in two main categories: electro-optical and all optical. Both electro-optical and all optical AFs are very competitive in terms of speed. They reach frequencies of up to 10 GHz and more, which is at least 10 times faster than modern GPUs. However, electro-optical conversions are power-hungry as they need high amplification of the signal, which is not the case for all-optical AFs. In terms of simulation results, both electro-optical and all-optical NNs provide lower accuracy on the MNIST dataset with respect to computer-based implementations. However, all-optical NNs lead to higher accuracy than the electro-optical ones. Finally, electro-optical AFs are easily reconfigurable thanks to their electronic components. The MRR-assisted MZI and the SOA are also both all-optical technologies that can implement reconfigurable AFs. The use of reconfigurable circuits for most commonly used AFs (ReLU,

Sigmoid, tanh, etc.) would simplify the architecture and facilitate the fabrication process. In terms of scalability, the autapse is a first step toward the study of cascadable ONNs. Based on our observation, we conclude that, despite the growing interest in ONNs, significant research and development efforts are still needed for the design of reconfigurable and scalable AFs.

We also note that the level of detail for the AFs implementation discussed in the literature is highly heterogeneous, which lead to a challenging comparison. Indeed, the ONNs rely on different architectures, use different AFs and are trained on different datasets. Since the accuracy depends on numerous criteria, such as the optimization method and the preprocessing of the dataset, concluding on the efficiency of an AF solely is challenging, even when comparing ONNs trained on the same dataset. Numerous key characteristics of AF are also missing for comprehensive comparison of their performance, which raises the needs for standardized evaluation methods. The implementation of ONNs is a multidisciplinary field involving photonics and machine learning. Unfortunately, there is a lack of simulation tools for the integration of ONNs. On one side, photonic simulation tools such as Lumerical enable the design and the simulation of ONNs but do not cover the training and optimization phases. On the other side, NNs libraries such as Tensorflow and Pytorch allow the training of complex NNs but cannot simulate photonic components. Currently, Neuroptica, Neurophox, and Photontorch are the best compromise, including models of MZIs and phase-shifters to build ONNs. However, they require heavy computation and additional optical components such as AFs are needed for comprehensive design and simulation. A holistic framework dedicated to ONNs would enable the production of fair, comprehensive, and reproducible results for ONNs, which are needed to compare specific blocks such as AFs.

REFERENCES

[1] Ion Stoica et al. 2017. A Berkeley View of Systems Challenges for AI. Retrieved from https://arXiv:cs.AI/1712.05855

[2] Yu-Hsin Chen et al. 2016. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. *IEEE J. Solid-State Circ.* 52 (2016), 127–138.

[3] Yichen Shen, Nicholas C. Harris, Scott Skirlo, Mihika Prabhu, Tom Baehr-Jones, Michael Hochberg, Xin Sun, Shijie Zhao, Hugo Larochelle, Dirk Englund, et al. 2017. Deep learning with coherent nanophotonic circuits. *Nature Photon.* 11, 7 (2017), 441–446.

[4] Lorenzo De Marinis et al. 2019. Photonic neural networks: A survey. *IEEE Access* 7 (2019), 175827–175841.

[5] William R. Clements et al. 2016. Optimal design for universal multiport interferometers. *Optica* 3, 12 (Dec. 2016), 1460–1465.

[6] T. F. de Lima et al. 2019. Machine learning with neuromorphic photonics. *J. Lightwave Technol.* 37 (2019).

[7] Bowen Bai et al. 2020. Towards silicon photonic neural networks for artificial intelligence. *Sci. China Info. Sci.* 63 (2020).

[8] Tyler W. Hughes et al. 2018. Training of photonic neural networks through *in situ* backpropagation and gradient measurement. *Optica* 5, 7 (July 2018), 864–871. DOI: http://dx.doi.org/10.1364/OPTICA.5.000864

[9] Ryan Hamerly, Liane Bernstein, Alexander Sludds, Marin Soljačić, and Dirk Englund. 2019. Large-scale optical neural networks based on photoelectric multiplication. *Phys. Rev. X* 9 (May 2019), 021032. Issue 2. DOI: http://dx.doi.org/10.1103/PhysRevX.9.021032

[10] Alexander Sludds, Saumil Bandyopadhyay, Zaijun Chen, Zhizhen Zhong, Jared Cochrane, Liane Bernstein, Darius Bunandar, P. Ben Dixon, Scott A. Hamilton, Matthew Streshinsky, Ari Novack, Tom Baehr-Jones, Michael Hochberg, Manya Ghobadi, Ryan Hamerly, and Dirk Englund. 2022. Delocalized photonic deep learning on the internet's edge. *Science* 378, 6617 (Oct. 2022), 270–276. DOI: http://dx.doi.org/10.1126/science.abq8271

[11] Collins English Dictionary. 2020. Machine learning. In *Collins English Dictionary*. HarperCollins Publishers. Retrieved from https://www.collinsdictionary.com/dictionary/english/machine-learning

[12] Febin P. Sunny, Ebadollah Taheri, Mahdi Nikdast, and Sudeep Pasricha. 2021. A survey on silicon photonics for deep learning. *J. Emerg. Technol. Comput. Syst.* 17, 4, Article 61 (June 2021), 57 pages. DOI: http://dx.doi.org/10.1145/3459009

[13] Aashu Jha, Chaoran Huang, Hsuan-Tung Peng, Bhavin J. Shastri, and Paul R. Prucnal. 2021. Photonic spiking neural networks and graphene-on-silicon spiking neurons. *J. Lightwave Technol.* 40 (2021), 2901–2914.

[14] Daniel J. Gauthier, Erik M. Bollt, Aaron Griffith, and Wendson A. S. Barbosa. 2021. Next generation reservoir computing. Retrieved from https://arxiv.org/abs/2106.07688
[15] Ethem Alpaydin. 2004. Introduction to machine learning. In *Adaptive Computation and Machine Learning*.
[16] Oceane Destras. 2020. *Modelling of Artificial Neural Networks with Integrated Photonics*. Master's thesis. Polytechnique Montréal. Retrieved from https://publications.polymtl.ca/5553/
[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. Retrieved from http://www.deeplearningbook.org
[18] American Heritage Dictionary. 2020. Backpropagation. In *The American Heritage Dictionary of the English Language, Fifth Edition*. Houghton Mifflin Harcourt Publishing Company. Retrieved from https://www.ahdictionary.com/word/search.html?q=backpropagation
[19] Claude Lemaréchal. 2012. Cauchy and the gradient method. *Doc Math Extra* 251, 254 (2012), 10.
[20] Chigozie Nwankpa et al. 2018. Activation functions: Comparison of trends in Practice and Research for Deep Learning. Retrieved from https://arxiv.org/abs/1811.03378
[21] Pascal Stark, Folkert Horst, Roger Dangel, Jonas Weiss, and Bert Jan Offrein. 2020. Opportunities for integrated photonic neural networks. *Nanophotonics* 9, 13 (2020), 4221–4232. DOI: http://dx.doi.org/doi:10.1515/nanoph-2020-0297
[22] L. Chrostowski et al. 2015. *Passive Components*. Cambridge University Press.
[23] Felipe Gohring de Magalhaes. 2017. *High-level modelling of optical integrated networks-based systems with the provision of a low latency controller*. Ph.D. Dissertation.
[24] Mark G. Kuzyk. 2017. *Nonlinear Optics: A Student's Perspective: With Python Problems and Examples*. Createspace Independent Publishing Platform, North Charleston, SC.
[25] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*. 265–283.
[26] Francois Chollet et al. 2015. Keras. Retrieved from https://github.com/fchollet/keras
[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.
[29] Mathlab. 2019. Deep learning toolbox. Retrieved from https://www.mathworks.com/help/deeplearning/index.html
[30] M. d. Jubayer Shawon and Vishal Saxena. 2020. Rapid simulation of photonic integrated circuits using verilog-a compact models. *IEEE Trans. Circ. Syst. I: Reg. Papers* 67, 10 (2020), 3331–3341. DOI: http://dx.doi.org/10.1109/TCSI.2020.2983303
[31] Nicolas Ayotte. 2022. nicolasayotte/MatlabGDSPhotonicsToolbox. Retrieved from https://github.com/nicolasayotte/MatlabGDSPhotonicsToolbox) Accessed: July 18, 2022.
[32] Soeren Schmidt. 2022. Interactive simulation toolbox for Optics. Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/40093-interactive-simulation-toolbox-for-optics Accessed: July 18, 2022.
[33] Sequoia Ploeg, Hyrum Gunther, and Ryan M. Camacho. 2021. Simphony: An open-source photonic integrated circuit simulation framework. *Comput. Sci. Eng.* 23, 1 (Jan. 2021), 65–74. DOI: http://dx.doi.org/10.1109/mcse.2020.3012099
[34] Floris Laporte, Simon Bilodeau, and Jan-David-Black. 2022. S + Autograd + XLA: S-parameter based frequency domain circuit simulations and optimizations using JAX. Retrieved from https://github.com/flaport/sax Accessed: June 6, 2023.
[35] Floris Laporte, Joni Dambre, and Peter Bienstman. 2019. Highly parallel simulation and optimization of photonic circuits in time and frequency domain based on the deep-learning framework PyTorch. *Sci. Rep.* 9, 1 (Apr. 2019), 5918. DOI: http://dx.doi.org/10.1038/s41598-019-42408-2
[36] Lumerical. 2021. High-performance photonic simulation software. Retrieved from https://www.lumerical.com/
[37] Photon Design. 2021. Photon Design—Your source of photonics CAD tools. Retrieved from https://photond.com/index.htm
[38] Jagmeet Singh, Hugh Morison, Zhimu Guo, Bicky A. Marquez, Omid Esmaeeli, Paul R. Prucnal, Lukas Chrostowski, Sudip Shekhar, and Bhavin J. Shastri. 2022. Neuromorphic photonic circuit modeling in Verilog-A. *APL Photon.* 7, 4 (2022), Retrieved from arXiv:https://doi.org/10.1063/5.0079984
[39] Ben Bartlett, Momchil Minkov, Tyler Hughes, and Ian A. D. Williamson. 2019. Neuroptica: Flexible simulation package for optical neural networks. Retrieved from https://github.com/fancompute/neuroptica

[40] Sunil Pai. 2019. Neurophox: A simulation framework for unitary neural networks and photonic devices. Retrieved from https://github.com/solgaardlab/neurophox/

[41] Michael Y.-S. Fang, Sasikanth Manipatruni, Casimir Wierzynski, Amir Khosrowshahi, and Michael Robert DeWeese. 2020. Design of optical neural networks with component imprecisions. Retrieved from http://arxiv.org/abs/2001.01681

[42] Alexander N. Tait, Thomas Ferreira De Lima, Mitchell A. Nahmias, Heidi B. Miller, Hsuan-Tung Peng, Bhavin J. Shastri, and Paul R. Prucnal. 2019. Silicon photonic modulator neuron. *Phys. Rev. Appl.* 11, 6 (2019), 064043.

[43] Sunil Pai, Ben Bartlett, Olav Solgaard, and David A. B. Miller. 2019. Matrix optimization on universal unitary photonic devices. *Phys. Rev. Appl.* 11, 6 (June 2019). DOI: http://dx.doi.org/10.1103/physrevapplied.11.064044

[44] Sarah Masaad, Emmanuel Gooskens, Stijn Sackesyn, Joni Dambre, and Peter Bienstman. 2022. Photonic reservoir computing for nonlinear equalization of 64-QAM signals with a Kramers–Kronig receiver. *Nanophotonics* 12, 5 (2022). DOI: http://dx.doi.org/doi:10.1515/nanoph-2022-0426

[45] Jonathan George, Armin Mehrabian, Rubab Amin, Paul R. Prucnal, Tarek El-Ghazawi, and Volker J. Sorger. 2018. Neural network activation functions with electro-optic absorption modulators. In *Proceedings of the IEEE International Conference on Rebooting Computing (ICRC'18)*. IEEE, 1–5.

[46] D. Caputo, G. de Cesare, A. Nascetti, and M. Tucci. 2009. Amorphous silicon balanced photodiode for detection of ultraviolet radiation. *Sensors Actuat. A: Phys.* 153, 1 (2009), 1–4. DOI: http://dx.doi.org/10.1016/j.sna.2009.04.017

[47] M. Nedeljkovic, C. G. Littlejohns, A. Z. Khokhar, M. Banakar, W. Cao, J. Soler Penades, D. T. Tran, F. Y. Gardes, D. J. Thomson, G. T. Reed, H. Wang, and G. Z. Mashanovich. 2019. Silicon-on-insulator free-carrier injection modulators for the mid-infrared. *Opt. Lett.* 44, 4 (Feb. 2019), 915–918. DOI: http://dx.doi.org/10.1364/OL.44.000915

[48] Alex Tait and Thomas Ferreira de Lima. 2019. Lightlab. Retrieved from https://github.com/lightwave-lab/lightlab

[49] Alexander N. Tait, Thomas Ferreira De Lima, Ellen Zhou, Allie X. Wu, Mitchell A. Nahmias, Bhavin J. Shastri, and Paul R. Prucnal. 2017. Neuromorphic photonic networks using silicon photonic weight banks. *Sci. Rep.* 7, 1 (2017), 1–10.

[50] Farshid Ashtiani, Alexander J. Geers, and Firooz Aflatouni. 2022. An on-chip photonic deep neural network for image classification. *Nature* 606, 7914 (June 2022), 501–506. DOI: http://dx.doi.org/10.1038/s41586-022-04714-0

[51] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. DOI: http://dx.doi.org/10.1109/5.726791

[52] Myeongsuk Pak and Sanghoon Kim. 2017. A review of deep learning in image recognition. In *Proceedings of the 4th International Conference on Computer Applications and Information Processing Technology (CAIPT'17)*. 1–3. DOI: http://dx.doi.org/10.1109/CAIPT.2017.8320684

[53] Ian Williamson et al. 2019. Reprogrammable electro-optic nonlinear activation functions for optical neural networks. *IEEE J. Select. Top. Quant. Electr.* 26, 1 (2019), 1–12.

[54] Claude Sammut and Geoffrey I. Webb (Eds.). 2010. *Mean Squared Error*. Springer US, Boston, MA, 653–653. DOI: http://dx.doi.org/10.1007/978-0-387-30164-8_528

[55] Saumil Bandyopadhyay, Alexander Sludds, Stefan Krastanov, Ryan Hamerly, Nicholas Harris, Darius Bunandar, Matthew Streshinsky, Michael Hochberg, and Dirk Englund. 2022. Single chip photonic deep neural network with accelerated training. Retrieved on July 21, 2023 from https://arxiv.org/abs/2208.01623

[56] Mario Miscuglio et al. 2018. All-optical nonlinear activation function for photonic neural networks. *Optic. Mater. Express* 8, 12 (2018), 3851–3863.

[57] Thomas Ferreira de Lima, Hsuan-Tung Peng, Mitchell A. Nahmias, Chaoran Huang, Siamak Abbaslou, Alexander N. Tait, Bhavin J. Shastri, and Paul R. Prucnal. 2019. Enhancing SOI Waveguide Nonlinearities via Microring Resonators. In *Proceedings of the Conference on Lasers and Electro-Optics*. DOI: http://dx.doi.org/10.1364/CLEO_SI.2019.SW3H.7

[58] Chaoran Huang et al. 2019. Programmable silicon photonic optical thresholder. *IEEE Photon. Technol. Lett.* 31, 22 (2019), 1834–1837. DOI: http://dx.doi.org/10.1109/LPT.2019.2948903

[59] Aashu Jha, Chaoran Huang, and Paul R. Prucnal. 2020. Reconfigurable all-optical nonlinear activation functions for neuromorphic photonics. *Optics Lett.* 45, 17 (2020), 4819–4822.

[60] David S. Broomhead and David Lowe. 1988. Multivariable functional interpolation and adaptive networks. *Complex Syst.* 2 (1988).

[61] Charles Dugas, Y. Bengio, François Bélisle, Claude Nadeau, and Rene Garcia. 2000. Incorporating second-order functional knowledge for better option pricing. *Proceedings of the 13th International Conference on Neural Information Processing Systems (NIPS'00)*, MIT Press, Cambridge, MA, 451–457.

[62] Chaoran Huang, Aashu Jha, Thomas Ferreira de Lima, Alexander N. Tait, Bhavin J. Shastri, and Paul R. Prucnal. 2021. On-chip programmable nonlinear optical signal processor and its applications. *IEEE J. Select. Top. Quant. Electr.* 27 (2021), 1–11.

[63] Kristof Vandoorne, Wouter Dierckx, Benjamin Schrauwen, David Verstraeten, Roel Baets, Peter Bienstman, and Jan Van Campenhout. 2008. Toward optical signal processing using photonic reservoir computing. *Optics Express* 16, 15 (2008), 11182–11192.

[64] Govind P. Agrawal and N. Anders Olsson. 1989. Self-phase modulation and spectral broadening of optical pulses in semiconductor laser amplifiers. *IEEE J. Quant. Electr.* 25, 11 (1989), 2297–2306.
[65] David Verstraeten, Benjamin Schrauwen, Michiel d'Haene, and Dirk Stroobandt. 2007. An experimental unification of reservoir computing methods. *Neural Netw.* 20, 3 (2007), 391–403.
[66] Kristof Vandoorne, Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Peter Bienstman. 2011. Parallel reservoir computing using optical amplifiers. *IEEE Trans. Neural Netw.* 22, 9 (2011), 1469–1481.
[67] Richard F. Lyon. 1982. A computational model of filtering, detection, and compression in the cochlea. In *Proceedings of the International Conference on Acoustics, Speech, & Signal Processing (ICASSP'82)*.
[68] George Mourgias-Alexandris, A. Tsakyridis, N. Passalis, Anastasios Tefas, K. Vyrsokinos, and Nikolaos Pleros. 2019. An all-optical neuron with sigmoid activation function. *Optics Express* 27, 7 (2019), 9620–9630.
[69] Dimitrios Apostolopoulos, Konstantinos Vyrsokinos, Panagiotis Zakynthinos, Nikos Pleros, and Hercules Avramopoulos. 2009. An SOA-MZI NRZ wavelength conversion scheme with enhanced 2R regeneration characteristics. *IEEE Photon. Technol. Lett.* 21, 19 (2009), 1363–1365.
[70] Xianxin Guo, Thomas D. Barrett, Zhiming M. Wang, and A. I. Lvovsky. 2021. Backpropagation through nonlinear units for the all-optical training of neural networks. *Photon. Res.* 9, 3 (2021), B71–B80.
[71] Q. Bao et al. 2011. Monolayer graphene as a saturable absorber in a mode-locked laser. *Nano Res.* 4, 3 (2011), 297–307.
[72] Jiamin Liu, Zia Ullah Khan, Cong Wang, Han Zhang, and Siamak Sarjoghian. 2020. Review of graphene modulators from the low to the high figure of merits. *J. Phys. D: Appl. Phys.* 53, 23 (Apr. 2020), 233002. DOI: http://dx.doi.org/10.1088/1361-6463/ab7cf6
[73] Albert Ryou, James Whitehead, Maksym Zhelyeznyakov, Paul Anderson, Cem Keskin, Michal Bajcsy, and Arka Majumdar. 2021. Free-space optical neural network based on thermal atomic nonlinearity. *Photon. Res.* 9, 4 (2021), B128–B134.
[74] Y. Zuo et al. 2019. All-optical neural network with nonlinear activation functions. *Optica* 6, 9 (2019), 1132–1137.
[75] Martin Andre Agnes Fiers, Thomas Van Vaerenbergh, Francis Wyffels, David Verstraeten, Benjamin Schrauwen, Joni Dambre, and Peter Bienstman. 2013. Nanophotonic reservoir computing with photonic crystal cavities to generate periodic patterns. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 2 (2013), 344–355.
[76] Dan C. Ciresan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. 2010. Deep, big, simple neural nets for handwritten digit recognition. *Neural Comput.* 22 (2010), 3207–3220.
[77] Dabal Pedamonti. 2018. Comparison of non-linear activation functions for deep neural networks on MNIST classification task. Retrieved from https://arxiv.org/abs/1804.02763
[78] Zbigniew Hajduk. 2018. Reconfigurable FPGA implementation of neural networks. *Neurocomputing* 308 (2018), 227–234. DOI: http://dx.doi.org/10.1016/j.neucom.2018.04.077
[79] Janaina G. M. Oliveira, Robson Luiz Moreno, Odilon de Oliveira Dutra, and Tales C. Pimenta. 2017. Implementation of a reconfigurable neural network in FPGA. In *Proceedings of the International Caribbean Conference on Devices, Circuits and Systems (ICCDCS'17)*. 41–44. DOI: http://dx.doi.org/10.1109/ICCDCS.2017.7959699
[80] M. Porrmann, U. Witkowski, H. Kalte, and U. Ruckert. 2002. Implementation of artificial neural networks on a reconfigurable hardware accelerator. In *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*. 243–250. DOI: http://dx.doi.org/10.1109/EMPDP.2002.994279
[81] Yuancong Wu, J. J. Wang, Kun Qian, Yanchen Liu, Rui Guo, S. G. Hu, Q. Yu, T. P. Chen, Y. Liu, and Limei Rong. 2020. An energy-efficient deep convolutional neural networks coprocessor for multi-object detection. *Microelectr. J.* 98 (2020), 104737. DOI: http://dx.doi.org/10.1016/j.mejo.2020.104737
[82] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. 2021. A survey on modern trainable activation functions. *Neural Netw.* 138 (2021), 14–32. DOI: http://dx.doi.org/10.1016/j.neunet.2021.01.026
[83] Gordon H. Y. Li, Ryoto Sekine, Rajveer Nehra, Robert M. Gray, Luis Ledezma, Qiushi Guo, and Alireza Marandi. 2023. All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning. *Nanophotonics* 12, 5 (2023), 847–855. DOI: 10.1515/nanoph-2022-0137
[84] Qiushi Guo, Ryoto Sekine, Luis Ledezma, Rajveer Nehra, Devin J. Dean, Arkadev Roy, Robert M. Gray, Saman Jahani, and Alireza Marandi. 2022. Femtojoule femtosecond all-optical switching in lithium niobate nanophotonics. *Nature Photon.* 16, 9 (Sep. 2022), 625–631. DOI: http://dx.doi.org/10.1038/s41566-022-01044-5
[85] Zaijun Chen, Alexander Sludds, Ronald Davis, Ian Christen, Liane Bernstein, Lamia Ateshian, Tobias Heuser, Niels Heermeier, James A. Lott, Stephan Reitzenstein, Ryan Hamerly, and Dirk Englund. 2023. Deep learning with coherent VCSEL neural networks. *Nature Photonics* (2023). https://doi.org/10.1038%2Fs41566-023-01233-w
[86] Jingxi Li, Tianyi Gan, Bijie Bai, Yi Luo, Mona Jarrahi, and Aydogan Ozcan. 2023. Massively parallel universal linear transformations using a wavelength-multiplexed diffractive optical network. *Adv. Photon.* 5, 01 (Jan. 2023). DOI: http://dx.doi.org/10.1117/1.ap.5.1.016003

[87] Ang Li and Wim Bogaerts. 2017. Tunable electromagnetically induced transparency in integrated silicon photonics circuit. *Opt. Express* 25, 25 (Dec. 2017), 31688–31695. DOI: http://dx.doi.org/10.1364/OE.25.031688

[88] Bin Wu, John F. Hulbert, Evan J. Lunt, Katie Hurd, Aaron R. Hawkins, and Holger Schmidt. 2010. Slow light on a chip via atomic quantum state control. *Nature Photon.* 4, 11 (Nov. 2010), 776–779. DOI: http://dx.doi.org/10.1038/nphoton.2010.211

Received 7 June 2022; revised 19 June 2023; accepted 22 June 2023