Machine-learning-aided abstraction of photonic integrated circuits in software-defined optical transport

(Article begins on next page)

PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Machine-learning-aided abstraction of photonic integrated circuits in software-defined optical transport

Khan, Ihtesham, Tunesi, Lorenzo, Chalony, Maryvonne, Ghillino, Enrico, Masood, Muhammad Umar, et al.

Ihtesham Khan, Lorenzo Tunesi, Maryvonne Chalony, Enrico Ghillino, Muhammad Umar Masood, Jigesh Patel, Paolo Bardella, Andrea Carena, Vittorio Curri, "Machine-learning-aided abstraction of photonic integrated circuits in software-defined optical transport," Proc. SPIE 11713, Next-Generation Optical Communication: Components, Sub-Systems, and Systems X, 117130Q (8 March 2021); doi: 10.1117/12.2578770

SPIE.

Event: SPIE OPTO, 2021, Online Only

Machine-learning-aided abstraction of photonic integrated circuits in software-defined optical transport

Ihtesham Khan[a], Lorenzo Tunesi[a], Maryvonne Chalony[b], Enrico Ghillino[b], Muhammad Umar Masood[a], Jigesh Patel[b], Paolo Bardella[a], Andrea Carena[a], and Vittorio Curri[a]

[a]Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy

[b]Synopsys, Inc., 400 Executive Blvd Ste 101, Ossining, NY 10562, United States.

ABSTRACT

In order to cope with the fast increase in data traffic demand, optical networks are fast evolving towards the disaggregation and progressive implementation of the openness paradigm. Such an evolution is enabling the application of the software-defined networking below the IP layer, down to the optical transmission (SD-OTN). SD-OTN is enabled by the capability of the network controller to automatized management of photonic switching systems, and allowing their full virtualization and softwarisation. To this purpose, one of the major matter of contention is an efficient utilization of routing strategies, which can be seamlessly incorporated into the control plane. In this work, we rely on data-driven science (DDS) to develop the machine learning (ML) model which is able to predict the routing strategies of generic $N \times N$ photonic switching system without any knowledge required of the topology. The dataset used for training and testing the ML model is generated “synthetically”. In particular, the training and testing of the proposed ML module is done in a completely topological and technological agnostic way and is able to perform its application in real-time. Furthermore, the scalability and accuracy of the proposed approach is verified by considering two different switching topologies: the Honey-Comb Rearrangeable Optical Switch and the Beneš network. Promising results are achieved in terms of predicting the control signals matrix for both of the considered topologies.

Keywords: Software-defined networking, Machine learning, Photonic integrated circuit, Optical switches, Microring resonators

1. INTRODUCTION

A dramatic increase in the global internet traffic, urged by the introduction of latest 5G technology along with the evolving concepts of connectivity such as Internet of things (IOT) are demanding for flexible and dynamic Network elements (NE) that can better optimize the complexities, power consumption and costs of optical networks. These attributes, set a pathway to the emerging technologies: Elastic optical networks (EONs) and Software-defined networking (SDN) in the optical networks. The EONs provide flexibility to the network controller to scale up or down resources according to the traffic requests in order to efficiently utilize the available spectrum.[1] In addition to this, the SDN implementation enables the management of each NE within a virtualized environment, so permitting a disaggregated approach to the network exploitation, enabling openness and virtual network slicing. Nowadays, NE are progressively exploiting photonic integrated circuits (PICs) to perform complex functions at the photonic level avoiding the bottleneck of opto-electronic conversion. This trend raises the management of such devices much more complicated. In this scenario, SDN becomes an important paradigm for the management of PICs. The PICs are a promising technology for next generation of photonic switches systems due to their low energy consumption, lower latency and small foot-print. These characteristic attributes of photonic switching systems will be widely adopted particularly in core optical networks and also enable emergent datacenter designs such as disaggregated hardware, dynamically reconfigurable networks, and application-dependent bandwidth allocation.

Further author information: ihtesham.khan@polito.it

Next-Generation Optical Communication: Components, Sub-Systems, and Systems X, edited by Guifang Li, Kazuhide Nakajima, Proc. of SPIE Vol. 11713, 117130Q
© 2021 SPIE · CCC code: 0277-786X/21/$21 · doi: 10.1117/12.2578770

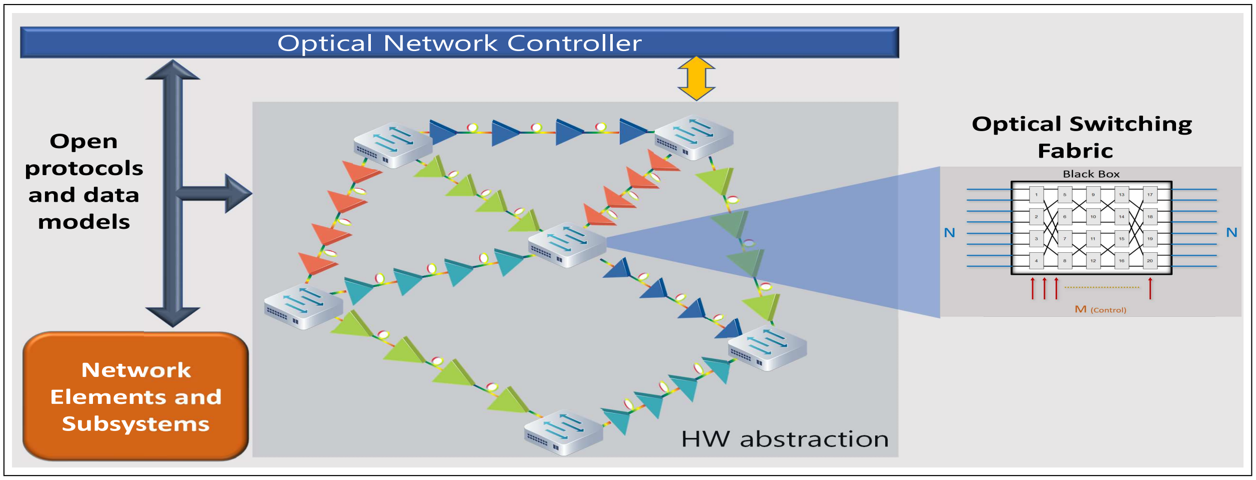


Figure 1: Software-defined Open Optical Networks

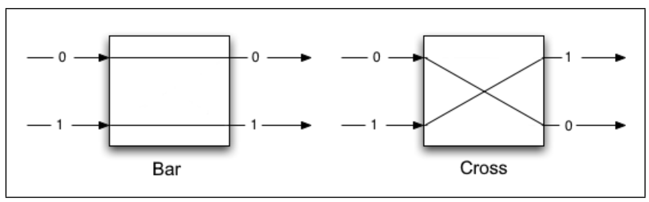
Typically, the basic building unit of PIC based optical switches are Mach-Zehnder (MZ) interferometers[2] or optical micro ring resonators(MRRs) .[3] The flow of light inside these building units can be routed to different paths by electrical control signals. Before augmenting PICs based solutions, a couple of different switching technologies have been proposed, such as three-dimensional micro-electro-mechanical systems (MEMS)[4] and beam-steering technique.[5] Both of these technologies are quite stable and mature for development of photonic switching system and also offer a reasonable degree of scalability. The major concern is about their precise calibration and installation of discrete components which make them much more expensive and massive. The current research on routing strategies of the optical switches has been readily reported. Unlike the electronic switching system, the optical switches have path dependent performance. The different variation in performance along different paths are due to fabrication and designed imperfection's, which eventually effects the routing strategies of the photonic switching system. Up-to the best of our knowledge, several deterministic routing algorithms were proposed in the literature which can efficiently determines the control/routing state of the internal switching element but most of them are topology dependent and are specifically designed for the particular internal topology and no generic algorithm exists.[6,7]

In contrast with topology-dependent algorithm, we propose a unique data-driven approach based on Machine Learning (ML) to obtain the control states of the internal switching element to obtain the requested wavelength permutation. The proposed approach is adopted in the Software-Defined Networking (SDN) domain which provides a flexible and softwarized control of any PIC based photonics switching system shown in Figure 1. The presented ML based approach works in a complete black-box scenario, as it requires a reasonable large amount of training dataset to build an appropriate model. The proposed generic approach does not require the insights knowledge of the internal topology. To generate the sufficient large instances of ML training and testing dataset, we mainly considered two switching devices: an Honey Comb Rearrangeable Optical Switch (HCROS) and a $N \times N$ Beneš network.[8–11] Regarding the opted HCROS we consider a typical configuration with 12 input and output ports (N) which are well controlled by 36 controlling ports/control signal (M). To further validate the scalability of the proposed technique, we also exploit three different sizes of Beneš switch having $N = 8$, 10, and 15, corresponding to specific configurations with M=20, 26, and 49 respectively. After extracting the dataset, the other fundamental part is to differentiate features and label. The present ML models exploits different combination of wavelengths (λ_1, λ_2, $\lambda_3 \ldots\lambda_n$) as features at the output ports whereas it maneuvers the M control signals as labels. In parallel to the proposed ML approach, we added an extra subsequent module which is based on a simple heuristic that is induced from observing error patterns on wrongly predicted M configurations. This specific heuristic based module provides assistance to the ML model by improving the prediction performance.

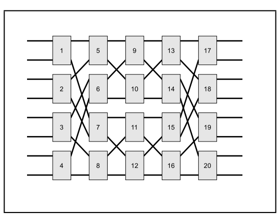
2. SWITCHING TOPOLOGIES

Typically, the switching system performance relies fundamentally on the choice of internal topology, switch blocking characteristics, total number of switching elements and number of cascading stages.[12] The switching topologies under analysis belong to the Banyan and Clos networks classes. These topologies are multi-stage architectures based on 2×2 Crossbar switching elements, and are able to route any possible permutation of the N inputs to the N output ports. Generally, the 2×2 Crossbar Switch has different physical implementation depending on the transmission characteristic of the device. For system where each input occupies a different frequency channel it can be implemented through the use of MRRs, while for monochromatic inputs it's typically designed through MZ interferometers.[13] Each unique permutation can be routed without conflict, as the non-blocking property implies, but differently from strict-sense non-blocking networks, where new input-output links can always be established without changing established links, in rearreangeable networks this is not guaranteed. Furthermore, the 2×2 Crossbar Switch has two characteristic states depending on the applied control bit: with the BAR state ($[0, 1] \rightarrow [0, 1]$) for $M = 0$ and the CROSS state ($[0, 1] \rightarrow [1, 0]$) for $M = 1$, as shown in Figure 2a.

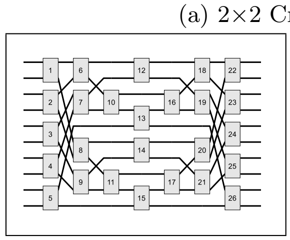
In this study, we analyzed two instances of multistage cross-over switching networks with non-blocking rearrange-ability: the Beneš network and the HCROS[8] shown in Figure 2(b–e). The Beneš switch is strictly



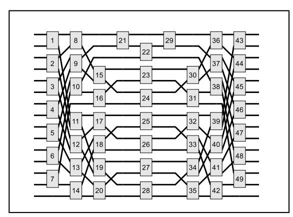
(a) 2×2 Crossbar switch



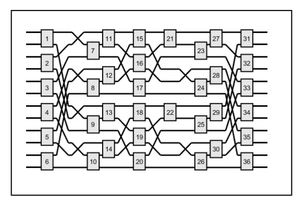
(b) 8x8 Beneš



(c) 10x10 Beneš



(d) 15x15 Beneš



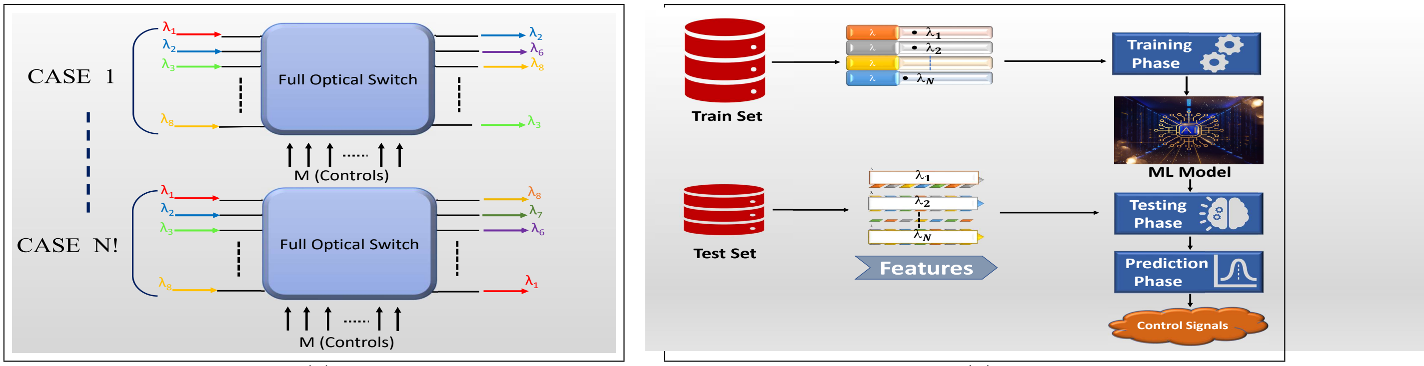
(e) 12x12 HCROS

Figure 2: Schematic of switch architectures

defined only for a number of inputs $N = 2^x, x \in \mathbb{N}$, although a generalization to any N is possible. This generalization is typically referred as Arbitrary Sized Beneš networks, or AS-Beneš.[14] While the Beneš is a highly symmetric and recursive network, as it can be built through a recursive algorithm, the custom tailored HCROS is an asymmetric and less regular structure. The HCROS[8] presents indeed a different topology, while maintaining equal number of elements to the equivalent sized Beneš switch. The original 6×6 HCROS, presented in,[8] has been expanded to a 12×12, due to the need of having a larger dataset, as well as the overall inadequacy of the original one to provide any meaningful results. The selection of HCROS switch has been made due to its similarity in properties but fundamental topological differences with respect to the Beneš.

3. SIMULATION MODEL & DATASET GENERATION

Typically, photonic switching systems are performing wavelengths routing from a definite optical line system (OLS) to the output of definite OLS, according to the switching matrix. For network abstraction, they define the spectral load at the input of each line system, so, their software abstraction must properly propagate signal information, including previously accumulated metrics. In the SDN perspective, the abstraction and softwarization of optical switch, at the control plane level, the request for any particular configuration of output wavelengths are received related to arbitrary switch with unknown topology. The ML-agent Application-program-interface (API) receives the desired configuration of output signals that are forwarded by the control plane and it must subsequently provides the M control signals, corresponding to the particular routing configuration requested.



(a) (b)

Figure 3: Representing $N!$ switch states for a $N \times N$ fabric (a), Description of the Machine Learning agent (b).

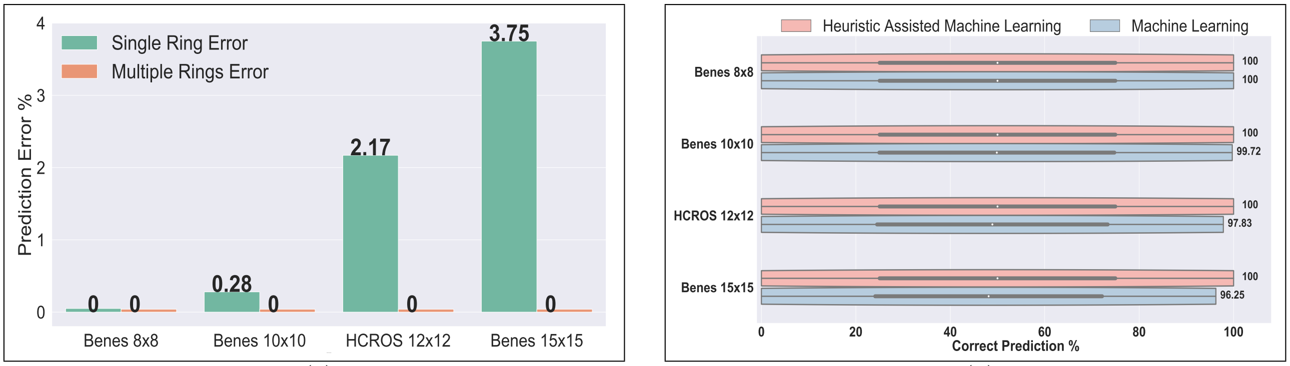
Table 1: Dataset Statistics

Network type **Size** ($N \times N$)	**Beneš** **8x8**	**Beneš** **10x10**	**HCROS** **12x12**	**Beneš** **15x15**
Permutations ($N!$)	40,320	3,628,800	479,001,600	1,307$\times 10^{9}$
Switches (M)	20	26	36	49
Combinations (2^M)	1,048,576	67,108,864	68$\times 10^{9}$	562$\times 10^{12}$
Dataset	100,000	300,000	300,000	1,000,000
Training set	70,000	210,000	210,000	700,000
Test set	30,000	90,000	90,000	300,000

After demonstrating the basic simulation environment, the most elegant part is the dataset generation which is fundamental unit for any data-driven technique. To generate the datasets needed for training and testing of ML model, the topologies have been implemented as a cascade of permutation matrices. This mathematical abstraction allows the evaluation of the network output by multiplying sequentially the stages matrices, without having to implement a more complex and time consuming graph structure. Each state of the network can be defined by its unique control vector, which is a binary array of length equal to number of switches, defining the CROSS or BAR state of each 2×2 Crossbar element. The training dataset is then built from randomly chosen vectors, with their respective output: the randomness ensures that the dataset is not biased toward preferential routing or switches configuration, as would happen by evaluating sequential control signals. The generated dataset may contain repetitions of the same output configuration, although with a different piloting vector: due to larger number of switching states with respect to the unique output configurations, equivalent routing is present in the network and, as an average, $(\frac{2^M}{N!})$ alternative routing paths exist for each output permutation as shown in Figure 3a.The size of the different switching architectures, as well as the generated dataset for the training and testing are shown in Table 1. The size of the dataset is chosen to be a small fraction of the complete solution of the network, for two main reason: due to the non-deterministic polynomial (NP) increase in complexity with respect to the input size N, the generation of complete dataset may not be possible in a reasonable time, and more importantly, if the full lookup table is computationally viable, the need for a routing algorithm is negated.

4. MACHINE LEARNING BLACK-BOX

The proposed ML based technique works in topological and technological agnostic way, requiring only a reasonable large amount of training data to learn the model, without taking into consideration the PIC internal topology. A Deep Neural Network (DNN) is selected as ML algorithm. The proposed model requires the definition of the features and labels representing the system inputs and outputs, respectively. The manipulated features include the various combinations of wavelengths at the output ports of the switch, and it exploits its M control signals as labels shown in Figure. 3b . The proposed DNN is developed by using higher-level APIs of the TensorFlow© platform .[15] The considered DNN is configured by several parametric values that have been optimized (such as the *training steps*, set to 1000), loaded with default *Adaptive Gradient Algorithm (ADAGRAD)* keras optimizer, with default *learning rate* set to 10^{-2} and default L_1 regularization set to 10^{-3}.[16] Moreover, *Relu* has been selected to implement DNN as non-linear activation functions.[17] Another important DNN parameter



(a) (b)

Figure 4: Errors in rings or control signals (a), prediction with ML and heuristic assisted ML (b).

is the number of *hidden-layers*. The model has been tuned testing several numbers of *hidden-layers* and neurons to achieve the best trade-off between precision and computational time. After this trade-off analysis, we decided upon a DNN with *three hidden-layers* with different number of cognitive neurons for each hidden layer.The conventional rule of 70% training and 30% testing dataset are used to train and test the model. In order to avoid over-fitting of the model, for each particular M we set the *training steps* as the stopping factor and the *Mean Square Error* (MSE) as the loss function. Furthermore, to improve the prediction performance of the ML module, we added an extra step based on a simple heuristic that we induced from observing wrong configurations of test dataset. In most of the cases where the ML agent is not able to predict the correct control state, a single switch is in an incorrect state. The heuristic we propose is to simply try the correction of these single-ring errors by flipping one switch at a time and comparing the output sequence against the desired output combination of wavelengths at the output ports.

5. RESULTS & CONCLUSION

In this section, we demonstrate the results obtained for the opted HCROS and different considered Beneš sizes in terms of predicting control states. The results in Figure. 4a shows the percentage of cases with a given number of errors in the control States, for HCROS and different considered Beneš sizes. The results in Fig. 4a depict that for HCROS and each size of Beneš the prediction of ML contains only single-ring errors i.e. one control signal is predicted wrong in all the test realization of incorrect routing. Finally, the statistics of correct prediction percentage using ML is summarized for HCROS and different considered Beneš sizes by the light-grey bars in Fig. 4b. We observe an excellent level of accuracy but with a decreasing trend of prediction ability when increasing the switch size N: correct predictions reach 100% for HCROS and 100% 99.72% and 96.25% for different considered Beneš sizes 8, 10 and 15, respectively. To further increase the predicting ability of the ML approach, we employed an auxiliary module based on a simple heuristic that we induced from the data shown in Fig. 4a. For HCROS and the considered Beneš network we labeled, all the instances of wrong routing carrying a single switch improperly set. The proposed heuristic actually try the correction of single-ring errors by simply flipping single switching-element at a time and start comparing the obtained sequence of wavelengths against the desired sequence of output wavelengths. This technique requires only M iterations and this number is reasonably small, so we can consider it also feasible for real-time implementations. For both HCROS and all Beneš sizes 8×8, 10×10 and 15×15, using the ML assisted by heuristic the accuracy raises to 100%.

In conclusion, we explore DDS to develop the ML model which is capable to effectively predict the control states for a generic $N \times N$ photonic switching system without considering its internal topology. The proposed ML based solution is scalable to large N as we demonstrate excellent level of accuracy with limited size dataset. Furthermore, the proposed simple heuristic assisted the ML model and subsequently increase the prediction performance with a small computational cost.

REFERENCES

[1] Gerstel et.al, O., "Elastic optical networking: A new dawn for the optical layer?," *IEEE Communications Magazine* **50**(2), s12–s20 (2012).

[2] Suzuki, K. et al., "Low-insertion-loss and power-efficient 32× 32 silicon photonics switch with extremely high-δ silica PLC connector," *Journal of Lightwave Technology* **37**(1), 116–122 (2019).

[3] Q., C. et al., "Ultralow-crosstalk, strictly non-blocking microring-based optical switch," *Photonics Research* **7**(2), 155–161 (2019).

[4] Jet, K. et al., "1100 x 1100 port MEMS-based optical crossconnect with 4-dB maximum loss," *IEEE Photonics Technology Letters* **15**(11), 1537–1539 (2003).

[5] Dames, A. N., "Beam steering optical switch," (June 17 2008). US Patent 7,389,016.

[6] Ding, M. et al., "Routing algorithm to optimize loss and IPDR for rearrangeably non-blocking integrated optical switches," in [*CLEO: Applications and Technology*], JTh2A–60, Optical Society of America (2015).

[7] Qian, Y. et al., "Crosstalk optimization in low extinction-ratio switch fabrics," in [*OSA*], Th1I–4, Optical Society of America (2014).

[8] Yahya, M. et al., "Honeycomb ROS: A 6 × 6 non-blocking optical switch with optimized reconfiguration for ONoCs," *Electronics* **8**, 844 (Jul 2019).

[9] Ghillino, E. et al., "Assessing the impact of design options for an optical switch in network routing impairments," in [*2019 21st ICTON*], 1–4, IEEE (2019).

[10] Khan, I. et al., "Machine learning assisted abstraction of photonic integrated circuits in fully disaggregated transparent optical networks," in [*2020 22nd ICTON*], 1–4 (2020).

[11] Khan, I. et al., "Effectiveness of machine learning in assessing qot impairments of photonics integrated circuits to reduce system margin," in [*2020 IEEE Photonics Conference (IPC)*], 1–2 (2020).

[12] Cheng, Q. et al., "Recent advances in optical technologies for data centers: a review," *Optica* **5**, 1354–1370 (Nov 2018).

[13] Islam, M. S. et al., "Mach Zehnder interferometer (MZI) as a switch for all optical network," in [*2018 International Conference on Innovation in Engineering and Technology (ICIET)*], 1–5 (2018).

[14] Chang, C. et al., "Arbitrary size Beneš networks," *Parallel Processing Letters* **07**(03), 279–284 (1997).

[15] Abadi, M. et al., "TensorFlow: A system for large-scale machine learning," in [*12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*], 265–283 (2016).

[16] Duchi, J. et al., "Adaptive subgradient methods for online learning and stochastic optimization," *JMLR* **12**(Jul), 2121–2159 (2011).

[17] Nwankpa, C. et al., "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378* (2018).