Activation Functions in Non-Negative Neural Networks

Marlon Becker[1] **Dominik Drees**[1] **Frank Brückerhoff-Plückelmann**[2]
Carsten Schuck[2] **Wolfram Pernice**[2] **Benjamin Risse**[1]
[1]Department for Computer Science & Institute for Geoinformatics
[2]Department for Quantum Technology
University of Muenster, Germany
{marlonbecker, dominik.drees, frank.bp, carsten.schuck,
wolfram.pernice, b.risse}@uni-muenster.de

Abstract

Optical neural networks (ONNs) have the potential to overcome scaling limitations of transistor-based systems due to their inherent low latency and large available bandwidth. However, encoding the information directly in the physical properties of light fields also imposes new computational constraints, for example the restriction to only positive intensity values for incoherent photonic processors. In this work, we address design and training challenges of physically constrained information processing with a particular focus on activation functions in non-negative neural networks (4Ns). Building on biological inspirations we revisit the concept of inhibitory (decreasing) and excitatory (increasing) activation functions, explore their effects experimentally and introduce a general approach for weight initialization of non-negative neural networks. Our results indicate the importance of both excitatory and inhibitory elements in activation functions in incoherent ONNs which should be considered for future design of optical activation functions for ONNs. Code is available at `https://nnnn.cvmls.org`.

1 Introduction

Modern deep learning models require an ever-growing amount of computational resources [17, 27]. Despite improved transistor technology [29, 35], physical constraints inevitably limit the scalability of semiconductor-based hardware as used in today's deep-learning accelerators [36] resulting in a steeply rising demand for alternative hardware-systems [18, 34].
Optical signal processing has the potential to overcome these limitations due to its ability to operate at higher rates while having a substantially smaller energy footprint [5, 23]. This is particularly true for deep neural networks which require vast amounts of independent scalar operations and can therefore further benefit from the intrinsic parallelization capabilities for wavelength- and spatial-multiplexing in optical processing [3, 38, 39], resulting in an increasing interest in so-called optical neural networks (ONNs) [31]. One intriguing class of ONNs is based on the incoherent superposition of several different light fields. Here, the analog input signals are encoded in the intensity of optical pulses of different frequency which are then processed by the photonic integrated circuit [3, 6, 11, 15, 38, 41, 42]. As a consequence, encoding negative numbers is not possible, constraining inputs and weights (and thus also activations and outputs) of neural networks to be non-negative (referred to as **N**on-**N**egative **N**eural **N**etworks, 4Ns, in the following). While approaches exist to circumvent these constraints by adjusting the physical system (e.g. electronic signal subtraction [33] or coherent, phase-sensitive photonic architectures [21, 32]), all of these inevitably imply additional computational demands or increase the complexity and sensitivity of the photonic circuit. Optimizing 4Ns instead enables physically more robust implementations of ONNs based on incoherent superposition and we identify the design of activation functions to be of particular importance for the success of 4Ns.
Despite the different approaches presented in the past [7, 10, 12, 13, 16, 24, 25, 26, 37, 40, 43]

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

only a few fulfill requirements to enable efficient training of 4Ns, which we will elaborate in this work. Also motivated by analog computing, 4Ns have already been a research topic of interest in the 1990's [4, 9, 28, 30], however, these considerations were mostly of theoretical nature and are not directly transferable to modern neural networks. In this work, we consider practical aspects of *designing* and *training* 4Ns with special consideration of non-negative activation functions and the resulting need for new weight initialization approaches.

2 Non-Negative Activation Functions

As established above, 4Ns impose constraints in two areas: 1) Model weights are non-negative and 2) inputs, outputs and all intermediate results are non-negative. Given non-negative weights and inputs, linear transformations of the network (such as fully connected layers or convolutions) inevitably yield non-negative results as well. Another (required) building block in terms of network architecture is the activation function, i.e., the non-linear function that is applied to the outputs of the linear layer (the activations). For 4Ns, its domain and image must be the non-negative real numbers. Notably, the ubiquitous ReLU activation function $R(x) = \max(x, 0)$ and its descendants (e.g., leaky ReLU [22]) do not qualify since they are strictly linear in the non-negative domain. This can be addressed by integrating an offset $c > 0$ (i.e., $R^{\text{shift}}(x) = \max(x - c, 0)$) so that the region of non-linearity (referred to as *activation center*) is located at c instead. Similarly, other activation functions such as sigmoid can be shifted/scaled to retain more of the defining range of the function in the positive domain.

However, a naive combination of the previously mentioned activation functions and linear layers immensely reduces the expressiveness of the network: All components of the network are monotonically increasing (for any combination of scalar input and output variables) and thus the whole network is also monotonic in the same sense. For classification tasks this effect is diminished via a softmax-layer $\sigma_i(\boldsymbol{x}) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$ for the final classification: In a two (or more) variable scenario, softmax is actually monotonically decreasing with respect to an input variable x_i and an output $\sigma_j(\boldsymbol{x})$ where $i \neq j$. However, results from intermediate layers are still limited to monotony, so the non-monotonic effect of the softmax in the last layer may not be sufficient to retain network expressiveness.

Similar to DeLaurentis et al. [9] and in reference to the biological origins, we refer to inhibitory/excitatory functions as monotonically decreasing/increasing functions, but also define an inhibitory/excitatory range of a function as a range on which the function is monotonically decreasing/increasing. A way to enable non-monotonically networks under non-negative constraints and thus increase the expressiveness of the network is the inclusion of inhibitory *as well as* excitatory elements. DeLaurentis et al. [9] proved the universal approximation theorem of Cybenko [8] in a similar setting using a combination of regular (excitatory) as well as sign-flipped (inhibitory) logistic functions. Another option is to choose an activation function that in itself includes inhibitory and excitatory elements. Possibly the simplest of such functions is the shifted absolute value function $A(x) = |x - c|$ for some $c > 0$.

3 Sequential Weight Initialization

Challenges remain in *training* of the 4N, including: 1) How to initialize non-negative weights before training? 2) How does the optimization process ensure that trainable weights remain non-negative? In this work, the second condition is ensured by clipping negative weights to zero after each optimization step. Other strategies (e.g., multiplicative updates [1, 2]) are conceivable as well, but are not considered further in the following. Instead, we focus on the aspect of weight initialization, for which commonly used methods (e.g., [14]) are not directly transferable to 4Ns: In the unconstrained setting, weight matrices $\boldsymbol{w}^l \in \mathbb{R}^{I,J}$ for each layer l are initialized stochastically around zero, i.e. $\mathbb{E}(w_{ij}) = 0$ and biases $\boldsymbol{b}^l$ to zero. As a result, the activations of layers are initially also distributed around zero, i.e., $\mathbb{E}(a_i^l) = 0$ where (for FC-layers) $\boldsymbol{a}^l = \boldsymbol{w}^l\boldsymbol{x} + \boldsymbol{b}^l$ for any activation layer input $\boldsymbol{x}$. This is a useful property for the commonly used nonlinearities since most established activation functions are centered at zero. In the 4N setting, this activation centering property is not achieved automatically: Since all weights are non-negative, the expected value (for any initialization other than a constant initialization of zero) of activations is also strictly positive, i.e, $\mathbb{E}(a_i^l) > 0$.

The simplest initialization method would be to initialize the network weights following a uniform distribution in the range $[0, b]$. We evaluate the impact of the upper bound b by training a CNN

on the CIFAR10 dataset [19]: A straight-forward convolutional neural network (CNN) with four convolutional layers followed by a shifted absolute value function as activation, two max pooling operations and a final fully connected layer before the classification was trained for 500 epochs with 5 instantiations per configuration. The convolutional layers width is controlled by the size factor $S \in \{1, 2, \dots 64\}$ resulting in channel numbers of $8S$, $8S$, $16S$, $16S$. A cosine learning rate scheduler with an initial learning rate of $\eta = 10^{-2}$ was applied. Results are shown in Figure 1A.
Similar to the effect which motivated Kaiming-initilization [14], we also observe that the optimal value for b depends on the network size. Too large upper bounds hinder convergence of the network while too small values result in a severe drop in performance. The optimal choice for b shifts towards smaller scales for wider networks demonstrating its dependence on network architecture. Thus a cumbersome hyperparameter search is needed for each new network architecture. However, Figure 1B shows that during training the mean of activation distribution for all layers shifts towards the activation center *regardless* of the initial choice for b.

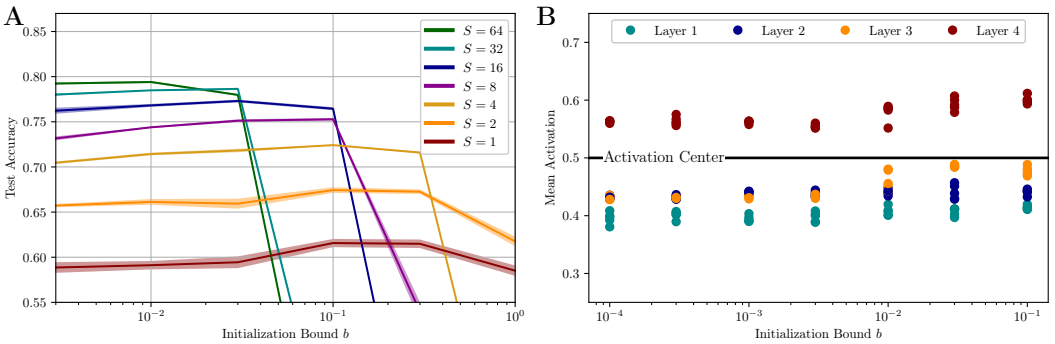

Figure 1: Evaluation of weight initialization with uniform distribution $[0, b]$ with constant initilization bound b. **A**: Achieved test accuracy for uniform initialization (solid lines) for different values of b. Different network widths require different values for b to achieve optimal performance. **B**: Mean activation distribution after training for networks with uniform weight initialization in the range $[0, b]$. *Independent* from the weight initialization values, the activation values tend towards the activation center after training (if convergent, i.e. $b \leq 0.1$).

This motivates an initialization of the weights, such that the activations are distributed around the center of the activation function. In order to achieve this property, we propose the following sequential initialization procedure: First all layers weights are initialized from an arbitrary positive distribution (e.g. a uniform distribution between 0 and some b). Subsequently, each layers weights are rescaled separately. For this, starting from the first layer, the mean activation m^l of layer l is computed for the entire training dataset

$$m^l := \frac{1}{T} \sum_{t=1}^{T} \frac{1}{I} \sum_{i=1}^{I} \sum_{j=1}^{J} w_{ij}^l x_j^{l,t} \tag{1}$$

where $x^{l,t}$ denotes the input of layer l for training sample $t \in \{1, ..., T\}$. The weights w_{ij} are scaled by $\frac{c^l}{m^l}$ where c^l denotes the activation center of the following activation function. Due to the linearity of the layer's affine operation, the modified activation is initialized around the activation function center. Notably, all biases in the linear layers are initialized with zeros.
While the initialization progresses from the input layers of the network to the output layers, changes to weights in deeper layers do not affect the activation distribution in earlier layers. As a result all activation distributions are initialized to the center of the following activation function.
The process of weight initialization based on the layer activation is illustrated in Figure 2 by showing the activation distributions next to the activation function for a four-layer CNN with the shifted absolute value as activation function. Additionally, the activation distributions after training (right-most) are also centered around the activation center, supporting the usefulness of the proposed method.

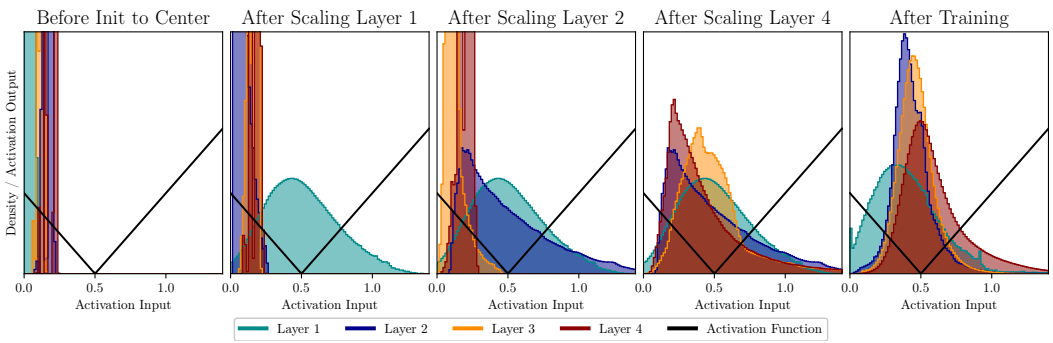

Figure 2: Illustration of the sequential initialization of the weights, so that the layer activation distribution mean matches with the activation function center. Activation distributions per layer next to activation function.

4 Evaluating Activation Functions in 4Ns

To explore the effect of different activation functions on 4Ns, two experiments were conducted. In first experiment a three layer multi layer perception (MLP; with hidden layers of size 100 each) was trained for 100 epochs on the MNIST dataset [20]. The second experiment once again evaluated the CNN described above on the CIFAR10 dataset [19] with fixed width of $S = 8$. Shifted variantes of the ReLU, Sigmoid (solely positive and with mixed signs [9]) and absolute value function were applied for 4Ns. Network weights were initialized using the proposed sequential method. As a baseline, standard ReLU, tanh and a centered absolute value function were used to train unconstrained networks. Each run was repeated 5 times and mean with 68% CI are reported. For equal base learning rates, all activation functions in the constrained setting showed similar conversion behavior. The final test accuracies in dependence on the learning rate are presented in Figure 3.
While performances of 4Ns with excitatory-only activation functions (sigmoid and ReLU) are reduced compared to unconstrained networks, 4Ns with activation functions comprising both inhibitory and excitatory elements, are yielding significantly higher test accuracies. This effect is especially pronounced for classification problems of higher complexity (CIFAR10) where the induced non-monotony of the softmax-layer is not sufficient. Therefore, it can be concluded that the inclusion of an additional inhibitory activation element yields significantly improved results.

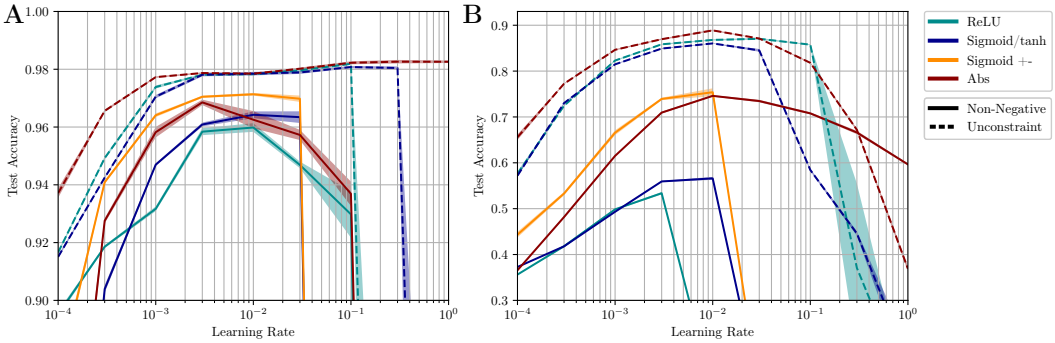

Figure 3: Evaluation of different activation functions for 4Ns and comparison with the unconstrained networks. **A**: MLP on MNIST. **B**: CNN on CIFAR10.

5 Conclusion

In this paper, we have investigated the effects of non-negative constraints arising in incoherent ONNs. Specifically, we have experimentally shown the extraordinary importance of excitatory and inhibitory elements in activations functions under these conditions which should inform design of future activation functions in incoherent ONNs and improve the performance of the resulting systems.

Furthermore, we have proposed a general solution for the problem of weight initialization arising specifically in 4Ns. Gains of specialized optimization methods for 4Ns (e.g., using [1, 2]) is still an ongoing research topic which we plan to investigate further in the future.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG CRC 1459 Intelligent Matter Project-ID 433682494).

References

[1] Bernstein, J., Vahdat, A., Yue, Y. and Liu, M. On the distance between two neural networks and the stability of learning. *Proceedings of NeurIPS*, volume 33, pages 21370–21381, 2020.

[2] Bernstein, J., Zhao, J., Meister, M., Liu, M.-Y., Anandkumar, A. and Yue, Y. Learning compositional functions via multiplicative weight updates. *Proceedings of NeurIPS*, volume 33, pages 13319–13330, 2020.

[3] Bernstein, L., Sludds, A., Hamerly, R., Sze, V., Emer, J. and Englund, D. Freely scalable and reconfigurable optical hardware for deep learning. *Scientific Reports*, 11(1):3144, 2021.

[4] Bradley, W. and Mears, R. Backpropagation learning using positive weights for multilayer optoelectronic neural networks. *Proceedings of LEOS*, volume 1, pages 294–295. IEEE, 1996.

[5] Brunner, D., Soriano, M. C., Mirasso, C. R. and Fischer, I. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Communications*, 4 (1):1364, 2013.

[6] Chang, J., Sitzmann, V., Dun, X., Heidrich, W. and Wetzstein, G. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific Reports*, 8(1):12324, 2018.

[7] Chen, X., Xue, Y., Sun, Y., Shen, J., Song, S., Zhu, M., Song, Z., Cheng, Z. and Zhou, P. Neuromorphic Photonic Memory Devices Using Ultrafast, Non-Volatile Phase-Change Materials. *Advanced Materials*, page 2203909, 2022.

[8] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

[9] DeLaurentis, J. M. and Dickey, F. M. A convexity-based analysis of neural networks. *Neural Networks*, 7(1):141–146, 1994.

[10] Demongodin, P., El Dirani, H., Lhuillier, J., Crochemore, R., Kemiche, M., Wood, T., Callard, S., Rojo-Romeo, P., Sciancalepore, C., Grillet, C. and Monat, C. Ultrafast saturable absorption dynamics in hybrid graphene/Si3N4 waveguides. *APL Photonics*, 4(7):076102, 2019.

[11] Feldmann, J., Youngblood, N., Karpov, M., Gehring, H., Li, X., Stappers, M., Le Gallo, M., Fu, X., Lukashchuk, A., Raja, A. S., Liu, J., Wright, C. D., Sebastian, A., Kippenberg, T. J., Pernice, W. H. P. and Bhaskaran, H. Parallel convolutional processing using an integrated photonic tensor core. *Nature*, 589(7840):52–58, 2021.

[12] Feldmann, J., Youngblood, N., Wright, C. D., Bhaskaran, H. and Pernice, W. H. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature*, 569(7755):208–214, 2019.

[13] Hazan, A., Ratzker, B., Zhang, D., Katiyi, A., Sokol, M., Gogotsi, Y. and Karabchevsky, A. MXene-Nanoflakes-Enabled All-Optical Nonlinear Activation Function for On-Chip Photonic Deep Neural Networks. *Advanced Materials*, 35(11):2210216, 2023.

[14] He, K., Zhang, X., Ren, S. and Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *Proceedings of ICCV*, pages 1026–1034. IEEE, 2015.

[15] Hughes, T. W., Minkov, M., Shi, Y. and Fan, S. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica*, 5(7):864–871, 2018.

[16] Jha, A., Huang, C. and Prucnal, P. R. Reconfigurable all-optical nonlinear activation functions for neuromorphic photonics. *Optics Letters*, 45(17):4819–4822, 2020.

[17] Jones, N. How to stop data centres from gobbling up the world's electricity. *Nature*, 561(7722): 163–166, 2018.

[18] Kaspar, C., Ravoo, B. J., van der Wiel, W. G., Wegner, S. V. and Pernice, W. H. P. The rise of intelligent matter. *Nature*, 594(7863):345–355, 2021.

[19] Krizhevsky, A., Hinton, G. and others, . Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009.

[20] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Lin, X., Rivenson, Y., Yardimci, N. T., Veli, M., Luo, Y., Jarrahi, M. and Ozcan, A. All-optical machine learning using diffractive deep neural networks. *Science*, 361(6406):1004–1008, 2018.

[22] Maas, A. L., Hannun, A. Y. and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. *Proceedings of ICML*, volume 30, page 3, 2013.

[23] Miller, D. A. B. Attojoule Optoelectronics for Low-Energy Information Processing and Communications. *Journal of Lightwave Technology*, 35(3):346–396, 2017.

[24] Miscuglio, M., Mehrabian, A., Hu, Z., Azzam, S. I., George, J., Kildishev, A. V., Pelton, M. and Sorger, V. J. All-optical nonlinear activation function for photonic neural networks. *Optical Materials Express*, 8(12):3851–3863, 2018.

[25] Mourgias-Alexandris, G., Tsakyridis, A., Passalis, N., Tefas, A., Vyrsokinos, K. and Pleros, N. An all-optical neuron with sigmoid activation function. *Optics Express*, 27(7):9620–9630, 2019.

[26] Opala, A., Panico, R., Ardizzone, V., Pietka, B., Szczytko, J., Sanvitto, D., Matuszewski, M. and Ballarini, D. Training a Neural Network with Exciton-Polariton Optical Nonlinearity. *Physical Review Applied*, 18:024028, 2022.

[27] Patterson, D. A., Gonzalez, J., Le, Q. V., Liang, C., Munguia, L., Rothchild, D., So, D. R., Texier, M. and Dean, J. Carbon Emissions and Large Neural Network Training. *arXiv preprint 2104.10350*, 2021.

[28] Pourzand, A. R. and Collings, N. Progress in the construction of a multilayer optical neural network. *Proceedings of Optics in Computing*, volume 3490, pages 439–442, 1998.

[29] Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S. and Kepner, J. Survey of Machine Learning Accelerators. *Proceedings of HPEC*, pages 1–12. IEEE, 2020.

[30] Saxena, I., Fiesler, E. and Moerland, P. A method for all-positive optical multilayer perceptrons. *Proceedings of ICECS*, pages 448–451. IEEE, 1996.

[31] Shastri, B. J., Tait, A. N., Lima, Ferreira de, T., Pernice, W. H. P., Bhaskaran, H., Wright, C. D. and Prucnal, P. R. Photonics for artificial intelligence and neuromorphic computing. *Nature Photonics*, 15(2):102–114, 2021.

[32] Shen, Y., Harris, N. C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., Sun, X., Zhao, S., Larochelle, H., Englund, D. and Soljačić, M. Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7):441–446, 2017.

[33] Tait, A. N., De Lima, T. F., Zhou, E., Wu, A. X., Nahmias, M. A., Shastri, B. J. and Prucnal, P. R. Neuromorphic photonic networks using silicon photonic weight banks. *Scientific Reports*, 7(1):7430, 2017.

[34] Theis, T. N. and Wong, H. P. The End of Moore's Law: A New Beginning for Information Technology. *Computing in Science and Engineering*, 19(2):41–50, 2017.

[35] Thompson, N. C., Greenewald, K. H., Lee, K. and Manso, G. F. The Computational Limits of Deep Learning. *arXiv preprint 2007.05558*, 2020.

[36] Waldrop, M. M. The chips are down for Moore's law. *Nature*, 530(7589):144–147, 2016.

[37] Wang, J., Zhang, L., Chen, Y., Geng, Y., Hong, X., Li, X. and Cheng, Z. Saturable absorption in graphene-on-waveguide devices. *Applied Physics Express*, 12(3):032003, 2019.

[38] Xu, X., Tan, M., Corcoran, B., Wu, J., Boes, A., Nguyen, T. G., Chu, S. T., Little, B. E., Hicks, D. G., Morandotti, R., Mitchell, A. and Moss, D. J. 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature*, 589(7840):44–51, 2021.

[39] Ying, Z., Feng, C., Zhao, Z., Dhar, S., Dalir, H., Gu, J., Cheng, Y., Soref, R., Pan, D. Z. and Chen, R. T. Electronic-photonic arithmetic logic unit for high-speed computing. *Nature Communications*, 11(1):2154, 2020.

[40] Yu, S., Wu, X., Wang, Y., Guo, X. and Tong, L. 2D Materials for Optical Modulation: Challenges and Opportunities. *Advanced Materials*, 29(14):1606128, 2017.

[41] Zhang, H., Gu, M., Jiang, X. D., Thompson, J., Cai, H., Paesani, S., Santagati, R., Laing, A., Zhang, Y., Yung, M. H., Shi, Y. Z., Muhammad, F. K., Lo, G. Q., Luo, X. S., Dong, B., Kwong, D. L., Kwek, L. C. and Liu, A. Q. An optical neural chip for implementing complex-valued neural network. *Nature Communications*, 12(1):457, 2021.

[42] Zhou, T., Lin, X., Wu, J., Chen, Y., Xie, H., Li, Y., Fan, J., Wu, H., Fang, L. and Dai, Q. Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nature Photonics*, 15(5):367–373, 2021.

[43] Zuo, Y., Li, B., Zhao, Y., Jiang, Y., Chen, Y.-C., Chen, P., Jo, G.-B., Liu, J. and Du, S. All-optical neural network with nonlinear activation functions. *Optica*, 6(9):1132–1137, 2019.