Implementation of Software Defined Electro-Optical Neural Networks

J. F. Kollmorgen, H. F. Santana, D. W. Feyisa, N. Calabretta, B. Shi, R. Stabile

EHCI, Eindhoven University of Technology, 5600MB, Eindhoven, The Netherlands

Abstract *- In recent years, the rising artificial intelligence workload has increased the demand for computational power, challenging traditional hardware like GPUs. Optical neural networks (ONNs) promise to offer faster computation with lower power consumption, but face issues such as scalability and high costs. In light of these challenges, we propose an alternative electro-optical neural network using off-the-shelf optical components, high-speed transceivers, and FPGA control, focusing on a single-layer implementation of the ONNs, to be iteratively re-used. We develop the design of the system, the encoding and decoding schemes, and the implemented training methods. Initial tests confirm the successful operation of perceptrons with TensorFlow models that consider the beforementioned physical implementation for XOR and Iris flower classification. A system efficiency of about 150 pJ/MAC is obtained from preliminary experiments with 10 Gb/s transceivers, which can be improved in future to 33.97 pJ/MAC when utilizing all system components.*

Introduction

Artificial intelligence has seen a rapid development in recent years and has led to numerous disruptive industry transformations not only in image recognition and data analysis but also in solving more complex problems with large language models such as ChatGPT. As a result of the recent advancements, the global artificial intelligence market is expected to grow at a 72 % compound annual growth rate [1]. Along with the increasing number of applications, the model complexity is expected to significantly rise. For instance, ChatGPT required 1.5 billion parameters to train and operate GPT-2 and already 175 billion parameters for GPT-3 [2]. While the demand for more computational power and resources is currently provided by conventional electronic hardware, such as Graphic Processing Units (GPUs), scaling these technologies with more Floating Point Operations per Second (FLOPS) at the same energy costs is a significant challenge: although transistor scaling continues to follow Moore's law, and the number of FLOPS improve as a result, the power efficiency does not scale according to Dennard's law [3].

Optical neural networks (ONNs) could be a promising alternative hardware, leveraging the light's properties for faster, higher throughput and low-energy processing. Free-space architectures such as the 4-F System or integrated photonic circuits have already demonstrated sub-nanosecond latencies and near-zero energy consumption for running neural network inference [4], but also face issues with misalignments and slow reconfigurability, as well as scalability when moving towards miniaturized photonics. Moreover, running all-optical training is still a challenge.

With this paper we propose an alternative software-defined electro-optical neural network architecture, using off-the-shelf data transceivers, active and passive optical components and an FPGA for control and reconfiguration. Our aim is to outline how the electro-optical architecture operates as a closed-loop one-layer network and to preliminary demonstrate that the inference of simple classification problems can be operated using this hardware with tens of Gb/s data throughput and about 33 pJ/MAC energy consumption.

Electro-Optical Neural Networks and Methods

To implement an optical neural network with off-the-shelf components, one must first create a single perceptron or McCulloch-Pitts neuron, as described in (1):

$$\hat{y} = \varphi\left(b_0 + \sum_{i=1}^{f} x_i w_i\right), \tag{1}$$

where $\hat{y}$ is the output of the neuron, b_0 is the bias, w_i the weight and x_i the input value for the i^{th} input. Therefore, one needs to be able to apply weights to multiple inputs, accumulate them, if required add a bias to the result, and then pass this to a non-linear activation function φ, e.g. a threshold function, that is used to normalize the output for the next neuron. The envisioned high-level system is shown as a sketch in Fig. 1a and the following components are utilized to perform the necessary operations:

Small Form-factor Pluggable (SFP) Transceiver: 10 Gb/s transceivers (SolidOptics) ranging from 1550nm to 1559nm, with 200 GHz spacing provide the encoded input signals for the neural network, as well as the receiver for detecting the signal at the output of the perceptron. The thresholder at the receiver can act as an activation function.

Variable Optical Attenuator (VOA): A 32-channel electro-optical NanoSpeed^{TM} VOA rack (Photonwares) is used to weight the optical signal coming from each transceiver via attenuating its strength. Switching speeds up to 300 ns and a 30 dB attenuation range.

Multiplexer (MUX): A 16-channel 200 GHz DeMUX DWDM rack from Simac Electronics is used to multiplex the signals together and thus perform the summation of the weighted inputs.

Splitters: Off-the-shelf splitters scale the system to multiple perceptrons per layer.

FPGA: The HTG-Z922 Xilinx ZYNQ FPGA board is used to control transceivers and VOAs and soft-define different neural network layouts. While the transceivers are controlled by the GTH IP core high-speed serial transceiver interface via SMA ports, the VOA is controlled via UART (Universal Asynchronous Receiver / Transmitter).

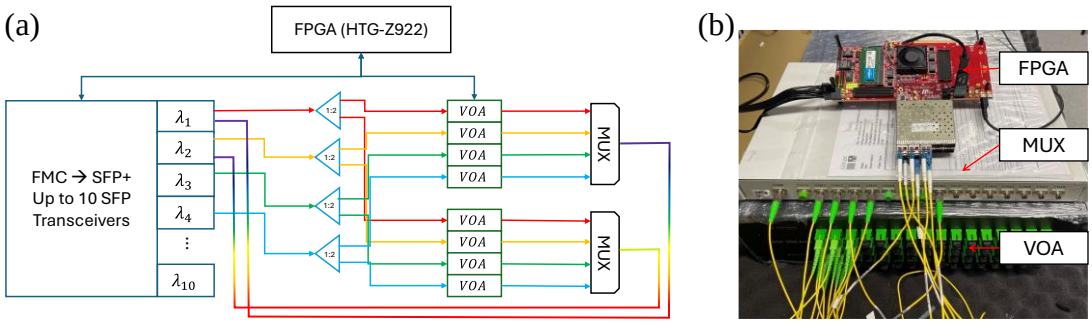

Figure 1: (a) The schematic of the envisioned high-level system with transceivers for the input / output of the perceptron, VOA to apply weights, multiplexer to summate the signal and FPGA to control all active components (b) Experimental setup for a single layer perceptron with maximum 4 inputs.

The 1:2 splitters in Fig. 1a can be used in future to scale the system and to operate multiple perceptrons in parallel, while Fig. 1b displays the experimental setup used to test individual perceptrons with up to 4 inputs and 1 bias signal.

Since the transceivers use On-Off-Keying (OOK) modulation, the input data is discretized and transmitted as a series of bits. Therefore, the input data is quantized into bits, so that the data encoding is in binary, while the output of the perceptron is dichotomous. Moreover, the receiver's built-in decision circuit can be modeled as:

$$y_i = f(z_i) = \begin{cases} 1 \; if \; z_i \geq \theta \\ 0 \; if \; z_i < \theta' \end{cases} \quad (2)$$

where z_i is the linear combination of the transmitted bits for input i, and θ is the threshold of the receiver to decide if the received bit is a "1" or "0". Therefore, the receiver's decision circuit can be viewed as a Heaviside step function or a steep sigmoidal function with a shifted threshold value at θ. Moreover, since the VOA can only attenuate the optical signal, the multiplexer receives an optical signal range between $P_{out} \in [0, P_{in}]$, where P_{in} is the input power before attenuation. Therefore, one must constrain the weights to the range [0,1], where one is the maximum normalized output power to P_{in}.

Neural Networks, Training Accuracies and Experimental Results

After defining the encoding/decoding scheme and considering the system constraints, two common neural network problems are solved: the XOR logic gate and the iris flower classification problem. To train the neural networks in TensorFlow, the steep differentiable sigmoid function $f(x) = \frac{1}{1+e^{-\alpha(x-\theta)}}$ is used, where $\theta = 0.5$ is the offset due to the weights being between [0,1] and α is the steepness factor, equal to 30 for the training and 1000 for the inference, in order to improve the training but also simulate the actual activation function during inference. Using the system constraints, first the XOR problem is solved using two inputs, each in the range [0, 1]. The output is 1 if the inputs are different and 0 if they are the same. With a dataset of 10 000 samples, which is split into 70% training and 30% testing, the neural network made out of only 3 logic gate-like perceptrons, shown in Fig. 2a, achieves a prediction accuracy of 100%.

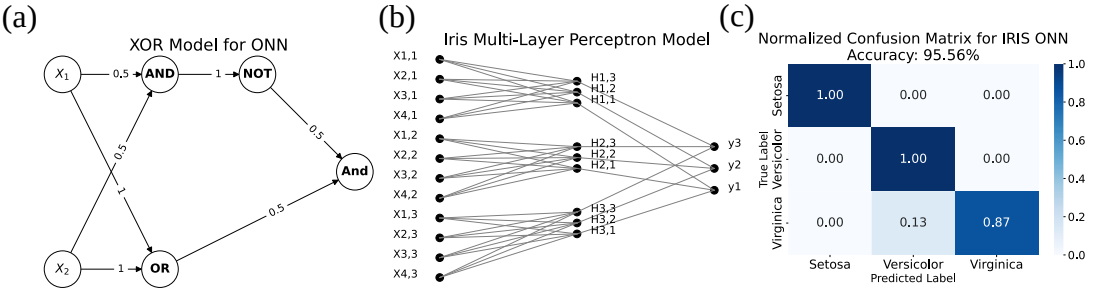

Figure 2: (a) NN topology for the XOR problem with fundamental logic gates. (b) NN topology for the iris flower classification. (c)Normalized confusion matrix of the trained iris neural network model with an accuracy of 95.56%. The true and predicted label of the model are shown.

More complex problems like the iris flower classification can also be solved, where based on the petal length/width and sepal length/width, 3 types of iris flowers can be classified (Setosa, Versicolor, Virginica). To train the samples in TensorFlow, the continuous input data is first quantised into 3-bit binary representations and the neural network in Fig. 2b is designed such that each bit of the 3-bit representation is processed individually in a single-layer implementation, where the result of the perceptrons in the hidden layer is temporarily stored on the FPGA before being passed on to the final perceptrons. This reduces the throughput but one can rely on a binary high accuracy neural network [5]: by splitting the 150-sample dataset into 70% training and 30% testing, a prediction accuracy of 95.56 % is obtained.

Besides training neural networks for the ONN, a successful operation of individual perceptrons is demonstrated. To reduce the power consumption and to synchronize the SFP transceivers on a bit level, the low power mode (LPM) equalization mode is used for

the decision circuit. Before operating the neural network, the transmitters are successfully aligned in phase with the TX phase interpolator controller from the Ultrascale IP core. In Fig. 3a, different hexadecimal values are transmitted at 10Gb/s by 2 different TX and are perfectly aligned. Moreover, by fixing the attenuation of the bias transceiver, the translation of the weights from a domain of [0,1] to the attenuation value is determined experimentally and visualized in Fig. 3b. While it is possible to control the VOA via

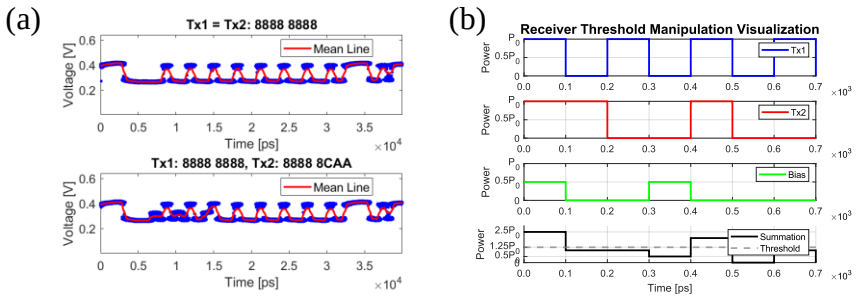

Figure 3: (a) Analog signal of the multiplexed Tx1&Tx2 signal, measured with the Agilent infiniium DCA 86100B wide-bandwidth oscilloscope. Encoded input value shown in hexadecimal in the title. (b) Multiplexed signal and threshold decision visualization of the receiver

UART from the FPGA and to run individual perceptrons manually, the full closed-loop system is yet to be tested. Nevertheless, the system's performance metrics can already be anticipated to some extent. Currently, 4 transceivers are used at 10Gb/s which can be increased to 10 transceivers yielding a total throughput of 100Gb/s and thus 100GMAC/s. Moreover, the total system power consumption is 6.139W, with 1mW per transceiver, 175mW avg. per VOA channel, and 5.259W for the FPGA (from Vivado's power analysis tool), yielding a system-level energy efficiency of 153.5pJ/MAC. However, when utilizing all 32 VOA channels, the system energy efficiency improves to 33.97pJ/MAC.

Conclusion and Discussion

In summary, an alternative electro-optical neural network is presented that uses off-the-shelf components. After discussing the encoding/decoding scheme and the system constraints, 2 newly designed and trained neural networks are presented. We demonstrated that it is possible to run individual perceptrons and the full neural network is yet to be tested. Although this paper only presents simple neural network models that can be run on the system, it can be extended to more complex problems and easily reconfigured by adding more components and by closing the loop. While the power efficiency does not outperform current ONNs, this system can be used as a testbench and implemented into a more integrated and optimized system in future, with faster and more efficient components. Moreover, the efficiency improves with the number of VOAs.

References

[1] S. Gantori, D. K. Limas, B. Chu, and K. Dennean, "TechGPT: Raising AI revenue forecast by 40% TechGPT series," Jan. 2024. Accessed: Aug. 13, 2024. [Online].

[2] K. S. Kalyan, "A Survey of GPT-3 Family Large Language Models Including ChatGPT and GPT-4," Oct. 2023, [Online]. Available: http://arxiv.org/abs/2310.12321

[3] L. Xiu, "Time Moore: Exploiting Moore's Law from the Perspective of Time," Dec. 01, 2019, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/MSSC.2018.2882285.

[4] M. Miscuglio and V. J. Sorger, "Photonic Tensor Cores for Machine Learning," Washington, Sep. 2020. doi: https://doi.org/10.1063/5.0001942.

[5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations," Sep. 2016, [Online].