Power-Aware Pruning for Ultrafast, Energy-Efficient, and Accurate Optical Neural Network Design

ABSTRACT

With the rapid progress of the integrated nanophotonics technology, the optical neural network (ONN) architecture has been widely investigated. Although the ONN inference is fast, conventional densely connected network structures consume large amounts of power in laser sources. We propose a novel ONN design that finds an ultrafast, energy-efficient, and accurate ONN structure. The key idea is power-aware edge pruning that derives the near-optimal numbers of edges in the entire network. Optoelectronic circuit simulation demonstrates the correct functional behavior of the ONN. Furthermore, experimental evaluations using tensorflow show the proposed methods achieved 98.28% power reduction without significant loss of accuracy.

KEYWORDS

artificial neural network (NN), optical neural network (ONN), sparsely connected network, integrated nanophotonics

ACM Reference Format:

. 2021. Power-Aware Pruning for Ultrafast, Energy-Efficient, and Accurate Optical Neural Network Design. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the last decade, an optical circuit emerged as a promising paradigm to resolve the latency issue of CMOS LSI circuits in advanced technology node. Traditionally, optical communication technologies have been rapidly growing over the past several decades for long-distance communications. However, the recent innovation in nanophotonics makes it possible to migrate power-efficient light-based communication into ever-shorter distances and move onto silicon chips as optical networks-on-chips [14]. Concurrently, significant efforts have been made on the architecture development of optical neural networks (ONN) [1, 3–6, 8, 11–13], motivated by the rapid evolution of neural networks (NNs).

Very recently, Hattori et al. proposed an ONN architecture based on a coherent optical vector-matrix multiplication (VMM) using wavelength division multiplexing, which enables inference processing with ultra-wideband[4]. Moreover, this architecture is based on a sparsely connected multi-layer perceptron, reducing the number of edges and power losses in the power splitters and combiners. Thanks to the sparse architecture, [4] significantly saves the power dissipation required for the laser source in the input layer without sacrificing the inference speed and accuracy. Originating from these excellent features, the architecture based on [4] is one of the most promising approaches for realizing the ultrafast, energy-efficient, and accurate ONN.

However, [4] only applies the edge-pruning to the input layer in an ad-hoc manner and does not consider the power reduction in the hidden and output layers. As explained later, we found that the hidden and output layers could highly dominate the overall power consumption, which is a severe limitation of [4] in terms of energy efficiency. Therefore, the power reduction methodology for the entire circuit is strongly demanded.

This paper proposes a novel ONN design method that identifies an ultrafast, energy-efficient, and accurate ONN circuit structure. The key idea of the proposed method is power-aware edge pruning that finds the near-optimal numbers of edges in the entire circuit, which is guided by an edge-conscious power consumption model for our ONNs. Consequently, the pruning method dramatically saves the power dissipation while satisfying the high inference accuracy.

Our experimental results show that the proposed algorithm reduces the power dissipation by two orders of magnitude without accuracy degradation compared to the architecture presented in [4]. In summary, the main contributions of this work are as follows.

- A novel ONN design based on power-aware edge pruning is proposed for the first time. Whereas huge number of pruning algorithms for traditional NNs are developed, we newly propose the pruning method for the VMM based ONN, aiming at ultrafast, energy-efficient, and accurate NN. Our pruning method takes into account the power dissipation model of ONN, which is totally different from the conventional NN. Thanks to the power model and pruning strategy, the proposed pruning dramatically saves the power dissipation of state-of-the-art VMM-based ONN.
- We quantitatively evaluate the functional behavior and power efficiency by an optoelectronic circuit simulator and a virtual environment for machine learning, i.e., TensorFlow. Experimental evaluations demonstrate that the proposed ONN correctly behaves while achieving the power saving by 98.28%.

The rest of the paper is organized as follows. Section 2 summarizes several previous works on the ONNs. Section 3 proposes a detailed architecture of our ONN. Section 4 shows experimental results obtained with the optoelectronic circuit simulator and TensorFlow. Section 5 concludes this paper.

2 RELATED WORKS

The following subsections summarize recent architectures proposed for optical neural networks.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.1 Coherent ONN with Mach-Zehnder Interferometer Array

In [12], a fully optical neural network (ONN) architecture is presented for implementing general deep neural network algorithms using nanophotonic circuits that process coherent light. The core part of the ONN architecture is a matrix multiplication unit composed of a reconfigurable Mach-Zehnder Interferometer (MZI) array. Once a NN is trained, the architecture can be passive, and we can perform computation at the speed of light without additional energy input. These features could enable ONNs that are substantially more energy-efficient and faster than their electronic counterparts. As described in [12], the energy consumption introduced by the switching activity is extremely small in this architecture. However, one big drawback in the ONN architecture described above is high photonic component utilization and area cost. Considering a single fully connected layer with an $n \times m$ weight matrix, the ONN architecture in [12] requires $O(n^2 + m^2)$ MZIs for implementation. If the number of neurons in the network increases, the area for the implementation increases quadratically. In [3], a more compact ONN architecture based on fast Fourier transform (FFT) is proposed. It improves the area efficiency of the ONN by a factor of 2.2 to 3.7. However, the area required for the implementation is still very large as the number of MZIs required is still quadratic to the number of neurons.

2.2 ONN based on Incoherent Accumulation

An optical circuit structure for vector-matrix multiplication (VMM) based on wavelength division multiplexing (WDM) is proposed in [13]. The incoming WDM carrier waves are weighted by continuous-valued filters called microring (MRR) weight banks. This operation corresponds to a parallel multiplication. The number of different multiplications performed at a time in the weight banks is equal to the number of different wavelengths multiplexed in the WDM carrier waves. The weighted optical signals are then summed as photocurrent by a photodetector. This operation corresponds to an accumulation. Unlike the coherent ONN presented in the previous subsection, this approach does not use optical coherence for the accumulation. Therefore, it is referred to as incoherent accumulation [2]. This architecture is a very area-efficient and low-power consumption. The complexity is $O(1)$ regardless of the number of weights. However, this approach has the following drawbacks. If more than one optical signal with different wavelengths (i.e., WDM signals) are given to the photodetector, undesirable oscillation in photocurrent occurs. This oscillation is known as a beat note. One straightforward approach to eliminate the beat note is low-pass filtering with an electronic low-pass filter. This approach is very simple but prevents VMMs from exploiting the ultra-high-speed nature of lights since the time constant of the low-pass filter is more than two orders of magnitude bigger than that of the operations using coherent light. Another approach for eliminating the beat note is using wavelengths that are sufficiently apart from each other. Since the oscillation frequency depends on the difference between the wavelengths of the lights, the frequency can be set to the out of the highest possible oscillation-band of the photodetector by setting the wavelengths sufficiently apart from each other. However, this approach limits the number of different wavelengths used in WDM signals and limits the scalability of the VMM.

In [1, 2, 6, 8, 9, 11], similar optical VMMs based on the MRR weight banks are proposed. The architectures are very compact and achieve the $O(1)$ VMM calculation using WDM signals and photodetector-based accumulation. However, they also have the oscillation issue in the photodetector, which prevents the light-speed operation of optical VMM.

2.3 Fully Coherent Optical VMM using WDM

A fully coherent vector-matrix multiplier (VMM) based on wavelength division multiplexing (WDM) is proposed in [5]. For solving the beat note issue incurred in the incoherent VMM presented in the previous subsection, the architecture in [5] employs an optical accumulation circuit using optical coherence. Each layer is composed of an MRR weight bank [13], parallelized multipliers based on Mach Zehnder Modulator (MZM), an accumulator based on an optical combiner [7], and optoelectronic activation circuits based on [10] as depicted at the bottom of Fig. 1. The output signals of the activation circuits are passed to the next layer as electrical voltage signals, which are used as inputs of the MZM-based multipliers in the next layer. Assuming the number of nodes in the input layer is X and that in the next layer is h, $X \times h$ micro-ring arrays as the MRR weight bank is needed. The number of wavelengths needed is h, and the number of rows in the bank is X. The h different carrier waves are given to every row in parallel. Then the $X \times h$ weight values are individually weighted to the carrier waves in parallel. The MZM next to the MRR weight bank functions as a parallel multiplier. The number of MZMs is X, and the number of wavelengths used in the WDM signals given to each MZM is h. Therefore, $X \times h$ multiplications are performed in the MZMs concurrently. Once the outputs of the MZMs are given to the power combiner as WDM signals, coherent accumulations are performed in parallel. Optical signals with the same wavelength are accumulated using optical coherence in the power combiner. This accumulation is performed for every different wavelength in parallel. Since the coherent light is used throughout the multiply and accumulation operation, the

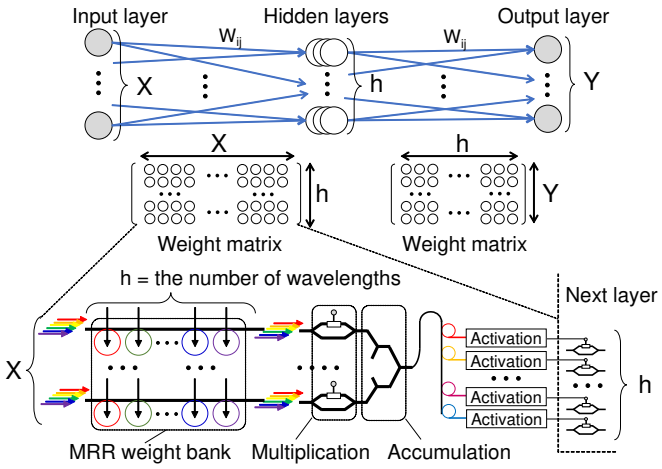

Figure 1: WDM-based optical neural network overview.

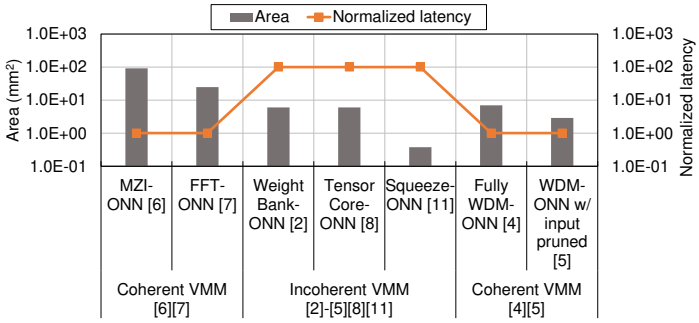

Figure 2: Comparison of optical vector-matrix multipliers (VMMs). Coherent VMM [4, 5] are well-balanced in terms of the circuit area and latency.

VMM calculation at the speed of light is possible in this architecture. Finally, the accumulated values are extracted by micro-ring resonators and given to activation circuits separately. The outputs of the activation circuits are passed to the next layer as inputs of the MZM-based multipliers in the next layer.

2.4 Comparison among Existing Optical VMMs

Figure 2 summarizes area and latency comparison among existing optical VMMs. The area estimation is based on [2, 4]. The latency is estimated based on the photocurrent-based accumulation discussed in subsection 2.2. The VMMs are composed of multipliers and accumulators. Among the optical VMMs, the coherent VMMs such as [3–5, 12] are very fast since they use optical coherence for both the multiplier and accumulator. The areas estimated for these VMMs include both those of multipliers and accumulators. Contrarily, incoherent VMMs such as [1, 2, 8, 9, 13] do not use the coherence of light for the accumulator. Instead of using optical coherence, they use photocurrent for the accumulation. Without using the optical coherence for the accumulation, the latency might be large since the accumulation has to be done with an electrical counterpart. It is typically two orders of magnitude slower than the optical circuits, such as a linear optical circuit presented in [7] that can complete the optical accumulation in the order of a few picoseconds. Although the incoherent VMMs are very compact, this slow accumulation may spoil the ultra-fast nature of light. In conclusion, since the WDM-VMM architecture presented in [4] is ultra-fast, wide-band, and low-footprint, this paper uses this as a baseline architecture to optimize.

3 PROPOSED EDGE PRUNING FOR MAXIMIZING POWER EFFICIENCY

This section describes the key idea behind of the proposed ONN design method. Section 3.1 explains the overview of our ONN design method. Section 3.2 proposes the power-aware edge-pruning algorithm.

3.1 Overview of Proposed ONN Design Method

An overview of the proposed ONN design method is depicted in Fig. 3. Given the power constraint, the proposed design method performs the power-aware edge pruning. The proposed method prunes the edges to save the power dissipation in laser sources,

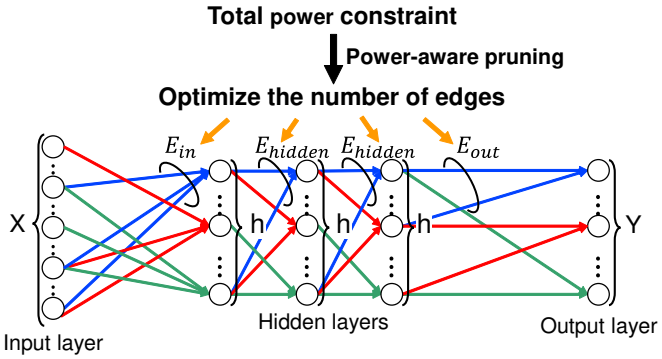

Figure 3: An overview of the proposed ONN design method with power-aware pruning.

which is based on the similar consideration of [4]. On the other hand, our approach is totally different from [4] since we aim at pruning edges in all the input, hidden, and output layers, whereas the architecture proposed in [4] focuses on the input layer only. Remind that our method takes into account the power dissipation model of ONN, which is a distinct target difference from conventional NN. In Fig. 3, E_{in}, E_{hidden}, and E_{out} represent the number of "non-pruned" edges leading to each node in each layer, and they are the key design parameters in the pruning step. As explained in Sec. 3.2, the proposed method takes into account the variance of weights and the number of nodes in each layer as the pruning cost, aiming at the maximization of the inference accuracy under the power constraint. Remind that the power constraint P is given as explained in Sect. 3.1.

Note that this work focuses on the power consumed in the laser source, as the laser power is dominant for ONNs with many branches which are implemented by splitters and combiners. The minimum laser power can be determined so that the minimum detectable signal power at the output of the circuit (i.e., photodetector) is ensured even if it is attenuated in the circuit. The edge pruning which corresponds to the branch pruning in the splitters and combiners can quadratically reduce the signal attenuation of the circuit. This leads to a quadratic power reduction in the laser source, as the minimum detectable power at the photodetector is fixed.

3.2 Proposed Power-Aware Pruning Algorithm

As explained in Sec. 3.1, if we adequately prune the edge connections, a significant power saving can be expected with a slight loss of accuracy. This pruning step can be regarded as the optimization problem that determines pruned edges from all possible combinations taking into account the inference accuracy and power dissipation. A brute force searching approach is not feasible since the accuracy calculation requires a relatively long computational time.

Here, in preliminary analysis of ONN training, we found that the average weight value of neurons is close to 0, which is an important observation. In this case, if the weight variance is small, many neurons have weight values near 0, which are easily pruned. Namely, by aggressively pruning the edges in small variance layers, we can approximately but efficiently distill the "important" edges

that impact the inference accuracy. On the other hand, if we prune edges of one layer too aggressively, the representation ability of the layer may degrade prohibitively. Therefore, for keeping the high inference accuracy, we need to select the pruned edges carefully taking into account both the weight value and the number of non-pruned edges in each layer.

Based on the above consideration, we propose the power-aware edge pruning method, maximizing the inference accuracy under the power constraint. The proposed method derives the pruning cost of each layer from the variance of weights and the number of nodes in each layer, determines the number of pruning edges in each layer, and prunes edges whose weight is smaller than the threshold value. By exploiting the weight variance and the number of non-pruned edges, the proposed method efficiently reduces the computational time required to search near-optimal pruned design.

Alg. 1 shows the proposed pruning algorithm. This algorithm repeats the procedures from lines 3 to 17 until the iteration number reaches the pre-determined upper bound, considering the algorithm's computational time and convergence problem. The input is a pre-pruned ONN, e.g., a fully connected ONN or a sub-pruned ONN, and the output is a power-aware pruned ONN.

Algorithm 1 Proposed power-aware edge-pruning algorithm

Require: X: training datum, $\{W_k : 0 \le k \le G\}$: the weight matrix of edges in ONN for each k-th layer, $\{M_k : 0 \le k \le G\}$: the binary masking matrix for each k-th layer, $\{E_k, \widehat{E}_k : 0 \le k \le G\}$: the number of non-pruned edges and the maximum number of non-pruned edges per nodes for each k-th layer, N_k: the numbers of nodes for each k-th layer, P: total power constraint.

Ensure: $\{\widehat{W}_k : 0 \le k \le G\}$: the updated parameter matrix for each k-th layer.

1: Initialize $M_k \leftarrow 1, \forall 0 \le k \le G$, $E_k \leftarrow N_k$, $\widehat{E}_k \leftarrow N_k$, and iter $\leftarrow 0$
2: **repeat**
3: Choose a minibatch of network input from X
4: Generate $\widehat{W}_k$ by quantizing $(W_k \odot M_k)$ by formula(1)
5: Forward propagation and loss calculation with the quantized weights
6: Backward propagation of the model output and generate ∇L
7: **for** $k = 0, \ldots, G$ **do**
8: Update W_k with the current loss function gradient ∇L
9: Update the weight variance β_k
10: Update M_k by formula(2) with W_k
11: Generates the each k-th layer's cost C_k by formula(3) with the weight variance ratio β_k and the number of nodes N_k
12: Update $\widehat{E}_k$ by formula(4) with the total power constraint P and the total cost C
13: **end for**
14: iter $\leftarrow$ iter + 1
15: **if** $\widehat{E}_k < E_k$ **then**
16: $E_k \leftarrow E_k - 1$
17: **end if**
18: **until** iter reaches its desired maximum

After initialization, quantization is performed according to Eq. (1) in line 4. For each k-th layer, this step prunes the edges with the masking matrix M_k. After pruning, non-pruned edges are binarized to -1 and 1. These weights are passed to the forward propagation, loss calculation, and backpropagation (lines 5 and 6).

$$\widehat{W_k^{i,j}} = \begin{cases} 0 & if\ M_k^{i,j} = 0\ (pruned), \\ -1 & if\ M_k^{i,j} = 1\ \&\ W_k^{i,j} < 0, \\ 1 & if\ M_k^{i,j} = 1\ \&\ W_k^{i,j} > 0. \end{cases} \tag{1}$$

Next, as described in lines 7 to 13, the algorithm updates the weight matrix W_k (line 8), the weight variance β_k (line 9), and masking matrix M_k (line 10), derives the pruning cost C_k (line 11), and updates the maximum number of pruned edges $\widehat{E}_k$ (line 12) for each k-th layer. Each element in M_k is updated by Eq. (2).

$$M_k^{i,j} = \begin{cases} 0 & if\ |T_{E_k,j}| > |W_k^{i,j}|, \\ 1 & if\ |T_{E_k,j}| \le |W_k^{i,j}|, \end{cases} \tag{2}$$

where $T_{E_k,j}$ is the threshold value. $T_{E_k,j}$ is the E_k-th highest value of weight list, which is the descending order of j-th column in W_k. If the absolute value of weight is smaller than that of $T_{E_k,j}$, we enable the masking and thus prune the edge. The cost of each layer (C_k) for updating the maximum number of non-pruned edges ($\widehat{E}_k$) is calculated by Eq. (3).

$$C_k = \beta_k \times N_k, \tag{3}$$

where β_k and N_k are the variance ratio of weights, and the number of nodes in the k-th layer, respectively. As shown in in Eq. (4), $\widehat{E}_k$ is computed considering the number of nodes N_k, total power constraint P, total cost C, and the cost C_k. Note that the total cost C corresponds to the summation of each C_k.

$$\widehat{E}_k = \left\lfloor \sqrt{\frac{P \times \frac{C_k}{C}}{N_k}} \right\rfloor. \tag{4}$$

Then, the algorithm checks and updates the number of non-pruned edges E_k (lines 15 and 16).

4 EXPERIMENTAL EVALUATION

This section first validates the functional behavior of the proposed ONN. The power efficiency and inference accuracy of the ONN optimized by the proposed pruning algorithm are then evaluated. Section 4.1 examines the functional behavior of our ONN via the optoelectronic circuit simulator. Then, using TensorFlow, Section 4.2 demonstrates the power saving effects thanks to the proposed pruning methods.

4.1 Optoelectronic Circuit Simulation

4.1.1 Evaluation Setup. As a test circuit, we design a 4×4 VMM circuit, as shown in Fig. 4. This circuit shows a sparsely connected MLP with two edges connected to each output node. In Fig. 4, x_1 to x_4 are given to the first layer of the MLP as electrical inputs, and y_1 to y_4 are obtained as electrical outputs. These outputs are passed to the next MLP layer as electrical inputs. Similarly, z_1 to z_4 are obtained as outputs of the second layer.

As shown at the bottom left of Fig. 4, the four WDM optical carrier waves are given and divided into two ways by an optical power splitter. Note that the WDM optical carrier waves are consisted

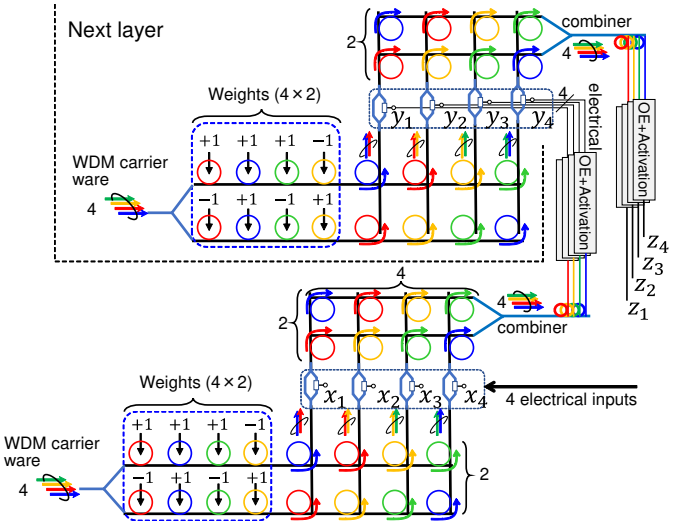

Figure 4: Test circuit with pruned vector-matrix multiplication, O-E conversion, and ELU-based activation circuits. The lower part represents the first layer and the upper part is the second layer in MLP.

of the optical waves with four different wavelengths. Then, those carrier waves are individually weighted by the MRR weight banks. The weighted optical signals are selectively passed to the four Mach Zehnder modulators (MZM) and individually multiplied with the electrical inputs x_1, x_2, x_3, and x_4. The accumulation is performed by the optical power combiner. The multiply-accumulation results are OE converted by photodetectors and then passed to the activation circuits. Finally, y_1 to y_4 are obtained as the outputs of the first MLP layer. The outputs y_1 to y_4 can be formulated as Eq. (5) where f is the activation function, and x_1 to x_4 are the inputs.

$$\begin{aligned} y_1 &= f(-1 \times x_1 + 1 \times x_2), \\ y_2 &= f(1 \times x_1 + 1 \times x_4), \\ y_3 &= f(-1 \times x_3 + 1 \times x_4), \\ y_4 &= f(1 \times x_2 - 1 \times x_3). \end{aligned} \tag{5}$$

To verify the designed circuit's behavior, we use Optisystem and OptiSPICE, which are commercial optoelectronic circuit simulators. In addition to MOS transistors, Optisystem and OptiSPICE can simulate optoelectric conversion in photodetectors and linear interference in MZMs and combiners at the transistor level. In the experiment, we used an exponential linear unit (ELU) function as the activation function, and performed the optoelectric circuit simulation. The inputs x_1 through x_4 are set to between -2 and 2, respectively. Then, y_1 to y_4 are updated by Eq. (5) and selected as inputs to the next layer. The outputs z_1 to z_4 can be similarly calculated in the second MLP layer from the inputs y_1 to y_4.

4.1.2 Evaluation Results. Figure 5 shows the simulation results for the test circuit. In the upper four results of Fig. 5, the black lines show simulation inputs x_1 to x_4 given to the first MLP layer. The colored lines show the simulation results. From Fig. 5, we can confirm that the outputs, i.e., y_1 to y_4, converge to the expected values in the first MLP layer. Similarly, in the second MLP layer, the outputs z_1 to z_4 converge to the expected values. We experimentally confirmed that the proposed sparsely connected ONN circuit works correctly in terms of functional operation from the above.

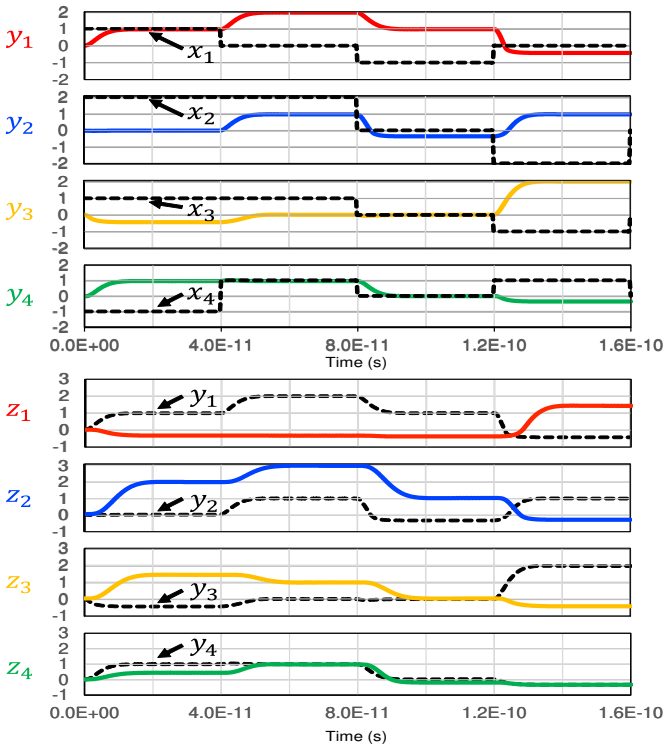

Figure 5: Simulation results for digital optical MAC, ELU activation function. The upper and lower parts correspond to the first and the second layers of the MLP, respectively. The colors of the outputs correspond to the colors of each wavelength in Fig. 4

4.2 Evaluation of Power Saving

4.2.1 Evaluation Setup. As a target circuit, we select the multilayer perceptron (MLP) with the MNIST dataset. This NN consists of one input layer, five hidden layers, and one output layer. The input layer has 784 (28 × 28) nodes, and the output layer has ten nodes. The numbers of nodes in the hidden layers are set to 98. In addition, we quantize the values of the weight matrices to binary (1 bit) for eliminating digital-to-analog converters (DAC) and thus saving the circuit area, which is based on the similar consideration of [4]. Leaky-ELU proposed in [4] is adopted to each node in hidden layers as the activation function. Besides, the Batch Normalization is added before the activation function.

Next, we apply the proposed design method to the test circuit. In the proposed pruning step, we prepare five different constraints for the total power consumption, i.e., 0.1 W, 0.3 W, 0.5 W, 1.0 W, and 1.5 W. We implement the above-designed circuits using TensorFlow and evaluate their inference accuracy. In addition, the power dissipation is estimated with identical methods of [4] for a fair comparison.

4.2.2 Evaluation Results. First, let us introduce the power-saving effects thanks to the proposed pruning method. Figure 6 compares the power consumption and inference accuracy between the fully connected ONN [5], the input pruned ONN [4], and the proposed design without loop structure. Note that Fig. 6 plots the proposed design whose power constraint is 1.0 W. In Fig. 6, the line graph represents the total power consumption, and bar graphs show the

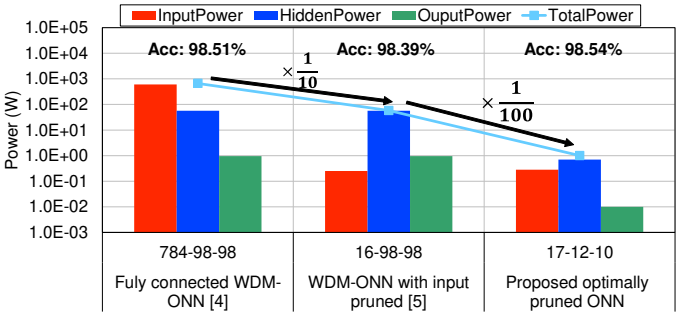

Figure 6: Power saving effects thanks to the proposed pruned ONN compared to [5] and [4]. The proposed pruning method significantly reduces the power dissipation without accuracy degradation. The numbers in the entries represent the number of connected edges to the nodes between layers. For example, "17-12-10" indicates that the number of connected edges between the input and first hidden layer, between the hidden layers, and between the last hidden layer and output layer are 17, 12, and 10, respectively.

power consumption in input, hidden, and output layers. Besides, the inference accuracy is described at the top of the figure, e.g., 98.51% for fully connected WDM-ONN [5]. From Fig. 6, we can see that the proposed pruning method significantly reduces the total power consumption without accuracy degradation. For example, whereas input-pruned ONN [4] achieves 98.39% inference accuracy with the power dissipation of 57.7 W, the proposed ONN achieves 98.54% accuracy with 0.99 W. Namely, the proposed ONN achieves the power reduction of 98.28% from 57.7 W to 0.99 W without any loss of accuracy. This significant reduction is due to the power reduction for hidden layers, as depicted with the blue bar graph. From this case study, we experimentally confirm that the proposed pruning method dramatically improves the power efficiency of WDM-based ONN.

Then, we discuss error rate reduction effects thanks to the proposed optimal method. Figure 7 shows the comparison results between the proposed pruned design and the naive design. From Fig. 7, we can see that the proposed method reduces the error rate compared with the naive method. For example, with the power

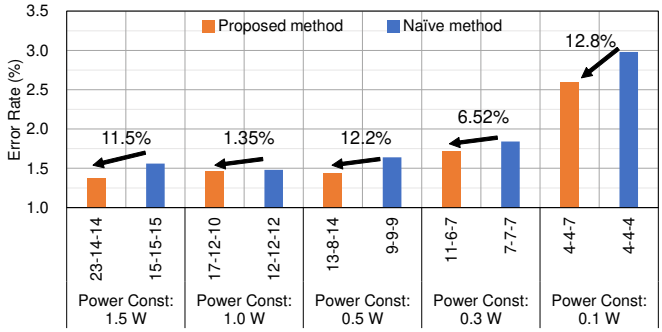

Figure 7: Error rate comparison between proposed optimal pruning method and naive pruning method with different power constraints. The numbers in the entries represent the number of connected edges to the nodes between layers.

constraint of 0.5 W, the error rate is reduced from 1.64% to 1.44%, which is more than 10% reduction in the error rate. In summary, we experimentally confirm that the proposed pruning method dramatically reduces the power dissipation while maintaining high inference accuracy. Since the ONN design based on [5] intrinsically has advantages, as discussed in Sec. 2.4, the power reduction thanks to the proposed design method contribute to realizing the ultrafast, energy-efficient, and accurate ONN.

5 CONCLUSION

This paper proposed the novel ONN design method that finds the ultrafast, energy-efficient, and accurate ONN structure. The key idea of the proposed method is power-aware edge pruning that optimizes the numbers of edges in input, hidden, and output layers. Thanks to the proposed pruning method, the inference accuracy is maximized under the constraint of the total power consumption in laser sources. Optoelectronic circuit simulation demonstrated the correct functional behavior of the ONN. Furthermore, experimental evaluations using TensorFlow showed that the proposed pruning method achieved 98.28% power saving without significant loss of inference accuracy.

REFERENCES

[1] J Feldmann, N Youngblood, M Karpov, H Gehring, X Li, M Stappers, M Le Gallo, X Fu, A Lukashchuk, AS Raja, et al. 2021. Parallel convolutional processing using an integrated photonic tensor core. *Nature* 589, 7840 (2021), 52–58.

[2] Jiaqi Gu, Chenghao Feng, Zheng Zhao, Zhoufeng Ying, Mingjie Liu, Ray T. Chen, and David Z. Pan. 2021. SqueezeLight: Towards Scalable Optical Neural Networks with Multi-Operand Ring Resonators. In *Proc. DATE*. 238–243.

[3] Jiaqi Gu, Zheng Zhao, Chenghao Feng, Zhoufeng Ying, Mingjie Liu, Ray T. Chen, and David Z. Pan. 2021. Towards Hardware-Efficient Optical Neural Networks: Beyond FFT Architecture via Joint Learnability. *IEEE TCAD* 40, 9 (Sep. 2021), 1796–1809.

[4] Naoki Hattori, Yutaka Masuda, Tohru Ishihara, Jun Shiomi, Akihiko Shinya, and Masaya Notomi. 2021. Optical-electronic implementation of artificial neural network for ultrafast and accurate inference processing. In *Proc. SPIE, AI and Optical Data Sciences II*, Vol. 11703. 117031E.

[5] Tohru Ishihara, Jun Shiomi, Naoki Hattori, Yutaka Masuda, Akihiko Shinya, and Masaya Notomi. 2019. An Optical Neural Network Architecture based on Highly Parallelized WDM-Multiplier-Accumulator. In *Proc. IEEE/ACM Workshop on Photonics-Optics Technology Oriented Networking, Information and Computing Systems*. 15–21.

[6] N. Janosik, Q. Cheng, M. Glick, Y. Huang, and K. Bergman. 2019. High-resolution Silicon Microring based Architecture for Optical Matrix Multiplication. In *Proc. CLEO*.

[7] Shota Kita, Kengo Nozaki, Kenta Takata, Akihiko Shinya, and Masaya Notomi. 2020. Ultrashort Low-Loss ψ Gates for Linear Optical Logic on Si Photonics Platform. *Communications Physics* 3, 33 (Mar. 2020).

[8] W. Liu, W. Liu, Y. Ye, Q. Lou, Y. Xie, and L. Jiang. 2019. HolyLight: A Nanophotonic Accelerator for Deep Learning in Data Centers. In *Proc. DATE*. 1483–1488.

[9] Mario Miscuglio and Volker J. Sorger. 2020. Photonic Tensor Cores for Machine Learning. *Applied Physics Reviews* 7, 031404 (2020).

[10] Kengo Nozaki, Shinji Matsuo, Takuro Fujii, Koji Takeda, Akihiko Shinya, Eiichi Kuramochi, and Masaya Notomi. 2019. Femtofarad Optoelectronic Integration Demonstrating Energy-Saving Signal Conversion and Nonlinear Functions. *Nature Photonics* 13 (July 2019), 454–459.

[11] Mehmet Berkay On, Hongbo Lu, Humphry Chen, Roberto Proietti, and S. J. Ben Yoo. 2020. Wavelength-Space Domain High-Throughput Artificial Neural Networks by Parallel Photoelectric Matrix Multiplier. In *Proc. IEEE Optical Fiber Communications Conference and Exhibition*. 1–3.

[12] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. B.-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljacic. 2017. Deep Learning with Coherent Nanophotonic Circuits. *Nature* 11, 7 (June 2017), 441–446.

[13] A. N. Tait, T. F. de Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal. 2017. Neuromorphic Photonic Networks using Silicon Photonic Weight Banks. *Sci. Rep.* 7, 1 (Aug. 2017).

[14] Xiaowen Wu, Jiang Xu, Yaoyao Ye, Zhehui Wang, Mahdi Nikdast, and Xuan Wang. 2014. SUOR: Sectioned Undirectional Optical Ring for Chip Multiprocessor. *JETC* 10, 4 (April 2014), 1–25.