ALoRA : Hardware-Aware Fine Tuning for Photonic Large Neural Networks

Taniguchi, Taichi
Department of Electrical and Electronic Engineering, Kyushu University

Nakajima, Mitsumasa
NTT Device Technology Labs., Nippon Telegraph and Telephone Corp

Ikeda, Kohei
NTT Basic Research Labs., Nippon Telegraph and Telephone Corp.

Hashimoto, Toshikazu
NTT Device Technology Labs., Nippon Telegraph and Telephone Corp

他

ALoRA: Hardware-Aware Fine Tuning for Photonic Large Neural Networks

Taichi Taniguchi[1], Mitsumasa Nakajima[2], Kohei Ikeda[3], Toshikazu Hashimoto[2], Satoshi Kawakami[1]
[1]Department of Electrical and Electronic Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka, Japan
[2]NTT Device Technology Labs., Nippon Telegraph and Telephone Corp., 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa, Japan
[3]NTT Basic Research Labs., Nippon Telegraph and Telephone Corp., 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa, Japan
[1] taniguchi@con.ed.kyushu-u.ac.jp, kawakami@ed.kyushu-u.ac.jp

Abstract: *We investigated the impact of analog errors on large-scale photonic neural networks using an experimentally obtained error model. Performance degradation due to analog error can be recovered by proposed hardware-aware fine-tuning with ~1.6% trainable parameters.*
Keywords: *Highly parallelized scalable photonic computing architectures and devices, Deep learning for photonic device and applications, Optical transformer*

I. Introduction

The rapidly increasing size of deep-learning models has motivated the development of alternative computational engines that can dramatically reduce the energy cost of running state-of-the-art deep neural networks (DNNs) [1]. One promising candidate as an alternative processor is a photonic neural network (PNN) with a photonic matrix-vector multiplication (MVM) engine, including both coherent [2, 3] and incoherent [4, 5] systems. Considering a practical use case by referring to existing edge computing devices, we will use such a processor by downloading and fine-tuning the parameters of various existing digitally pre-trained DNNs and transferring them to the PNN as shown in Fig. 1 (a). However, PNNs generally suffer from inherent errors, namely input/output quantization errors for digital-to-analog/analog-to-digital conversions (DA/ADC), and analog errors between the ideal and implemented matrix weight δW due to crosstalk and fabrication/control errors. These analog errors significantly degrade the inference performance of PNNs [6]. Accordingly, it is important to understand their effect on the model scale and task complexity by considering actual error levels in implemented photonic processors. In addition, an error mitigation algorithm is highly desired towards large-scale PNN [7-9].

Here, we investigate the impact of analog errors on various DNN models up to a practical parameter size (~8.7×10^7) by simulation based on an experimentally observed error model with a GPU cluster. To mitigate the observed significant performance degradation even under the observed low-level error (~1.2%) in our processor, we also demonstrate a hardware-aware fine-tuning method named "Analog Low-Rank Adaption" (ALoRA), inspired by a recent efficient fine-tuning algorithm [10-13], which could recover the performance while ~99% of parameters are untrained. Our investigation clarified the scalability of PNNs even under implementable analog error levels.

II. Simulation of Photonic MVM Based on Experimental Results

Fig. 1 (a) is a schematic illustration of our simulation setup. We transfer the pre-trained DNN weights W to the photonic MVM engine by quantizing each parameter to an n-bit value. The input x is quantized to an m-bit value by DAC. The quantized input x' is sent to the photonic MVM engine, which returns the MVM results with analog error $(W'+\delta W)x'$,

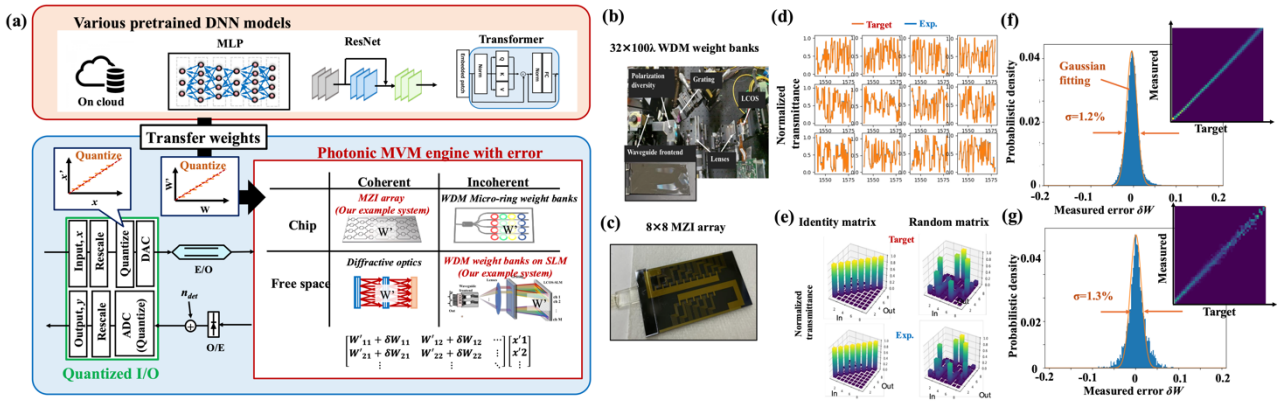

Fig. 1 (a) Schematic illustration of photonic MVM engine with transferring pre-trained DNN parameters. Images of our fabricated (b) incoherent and (c) coherent photonic MVM engines. Comparison of target and measured weight for (d) incoherent and (e) coherent processor. Measured histogram of (f) incoherent and (g) coherent processors.

where W' and δW are the quantized weight and its analog error. On the detector side, output values are quantized again by ADC. Note that to consider the impact of the fidelity of the photonic MVM engine itself, we neglected the detection noise (thermal and shot noise), which depends on the type of photodetector. To consider the effective the error model of δW in the photonic MVM engine, we measured δW in our developed incoherent [Fig. 1 (b)] and coherent [Fig. 1 (c)] photonic processors [3, 5]. By implementing random weights, we estimated the difference between target and measured weights. Examples of measurements are shown in Figs. 1 (d) and (e). Figs. 1 (f) and (g) show the histogram of measured δW for each photonic processor. Both histograms could be fitted by a gaussian function, suggesting that the error can be modeled by gaussian noise by considering the measured standard deviation (SD) σ (σ=1.2% and 1.3% for the incoherent and coherent processors, respectively). The bit number of the processors is m=n=8, which was determined by our digital-analog interface. We have implemented an in-house simulator based on PyTorch [14] and Hugging Face [15] to investigate the impact of analog errors on various large models. The simulator performs processing based on the inference flow in Fig. 1. By applying gaussian noise based on the standard deviation of the measured error in units of MVM execution, the simulator achieves both high speed and high accuracy.

III. Impact of Analog Error on Photonic MVM with Post-Training Quantization

We investigated the impact of the analog error on the model and task complexity by using the in-house simulator on GPU cluster (NVIDIA H100 × 8), enabling the investigation on the larger models beyond the previous studies [e.g., 5]. In the training phase, the weights on pretrained DNN models (FP32) are simply quantized (INT16, INT8) which are well known as PTQ (Post-Training Quantization). Then, the quantized weights are transferred to photonic MVM simulator in inference phase. Fig. 2 (a) – (d) show the simulated test accuracies as a function of the SD of error for the 4-layer multilayer perceptron (MLP), 4-layer convolutional DNN (CNN), 18-layer residual DNN (ResNet-18) [16], and 12-layer vision transformer (ViT) [17, 18]. The MNIST (red), CIFAR10 (blue), and CIFAR100 (green) tasks were tested with m=n=8 (solid lines) and 16 bits (dashed lines). As can be seen in the figures, the error robustness was highly dependent on the model size, while the task dependency was relatively small. For instance, the test accuracies for the simple MLP model were highly stable against both quantizing and weight errors, but those for the ViT significantly depended on the analog error (only 1% error and 8-bit quantization are critical). This suggests that a method for mitigating these errors is important. Although previous studies have suggested various hardware-aware training for the error mitigation [7-9], they typically require full-parameter tuning from the vanilla state, meaning that substantial computational time and energy are required for the retraining.

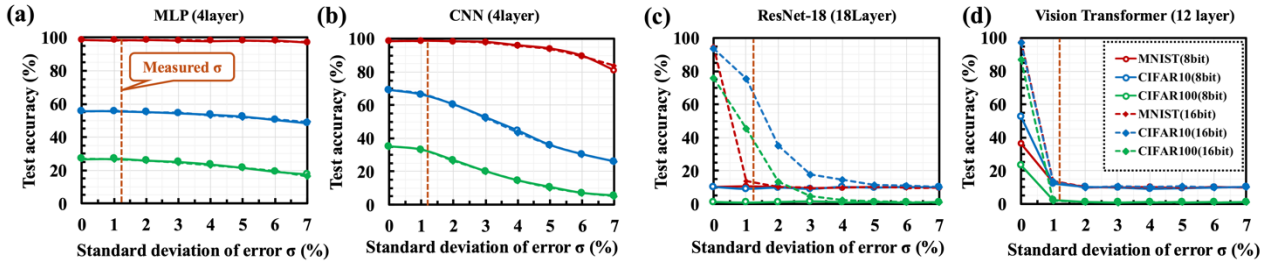

Fig. 2 Test accuracies as a function of weight error for (a) MLP, (b) CNN, (c) ResNet, and (d) Vision Transformer.

IV. ARoLA: Hardware-aware Fine-tuning for Error Mitigation

For mitigating the error, we also demonstrate a hardware-aware fine-tuning method named "Analog Low-Rank Adaption" (ALoRA), inspired by a recent efficient three fine-tuning algorithm: QAT (Quantized-Aware Training) [10], NAT (Noise-Aware Training) [11], and LoRA (Low-Rank Adaptation) [12, 13]. Fig. 3 (a) shows the overview of our simulation flow and proposed scheme "ALoRA". QAT is a training technique designed to improve the robustness of deep neural networks when deployed with quantized weights and activations. Fig. 3 (b) shows the details of the QAT procedure. In general, there are feed-forward (inference) and back-propagation (training) processes in the fine-tuning. The quantization error is incorporated only in the feed-forward process, and the back-propagation process is performed in single-precision (FP32). In the quantization, the values contained in a single MVM process are scaled according to their dynamic range, converted to integer type (int), and then converted to single precision again. Unlike PTQ, QAT simulates quantization effects during training, allowing the model to learn to compensate for precision loss. This is particularly important for photonic neural networks (PNNs), where digital-to-analog and analog-to-digital conversions introduce quantization errors that can degrade performance. Noise-Aware Training (NAT) focuses on enhancing model robustness by explicitly incorporating noise effects into the training process. As shown in Fig. 3 (c), just like QAT, noise is applied only to the feed-forward process, and all back-propagation is processed in single precision (FP32). In our simulator, adding gaussian noise of an arbitrary standard deviation to the weights W before matrix multiplication is possible. Furthermore, it is also possible to investigate the effects of experimental system errors by adding measurement errors

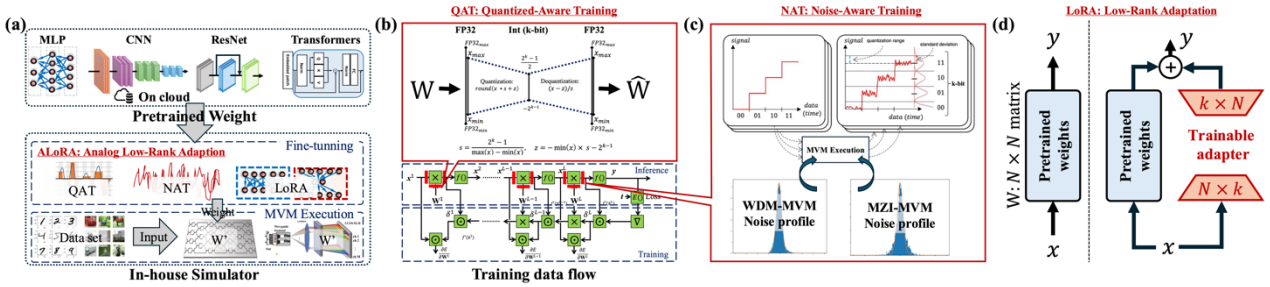

Fig. 3 (a) Schematic illustration of ALoRA fine-tuning. (b) Training process overview and QAT: Quantized-Aware Training (c) NAT: Noise-Aware Training (d) LoRA: Low-Rank Adaptation.

such as those shown in Fig. 1 (f, g). In the context of PNNs, NAT helps mitigate performance degradation caused by analog errors such as fabrication imperfections and crosstalk by injecting noise into the model during training. This approach allows the network to adapt to real-world error conditions, improving its resilience in practical deployment scenarios. Finally, LoRA is an efficient fine-tuning method that reduces the number of trainable parameters while maintaining performance. It was originally inspired by singular value decomposition to capture task-specific adjustments with minimal computational overhead effectively. Instead of updating all model parameters, LoRA introduces low-rank trainable matrices to capture task-specific adaptations. As shown in Fig. 3 (d), we add trainable adapter layers ($N\times k$ and $k\times N$ full connections, where k is typically much smaller than N) for each $N\times N$ layer. The original transferred $N\times N$ layer remains untrained, and only the adapters are trainable, drastically reducing the training parameters. The adapter layer can be considered a low-rank $N\times N$ matrix, assuming that this matrix would be sufficient to capture task-specific features.

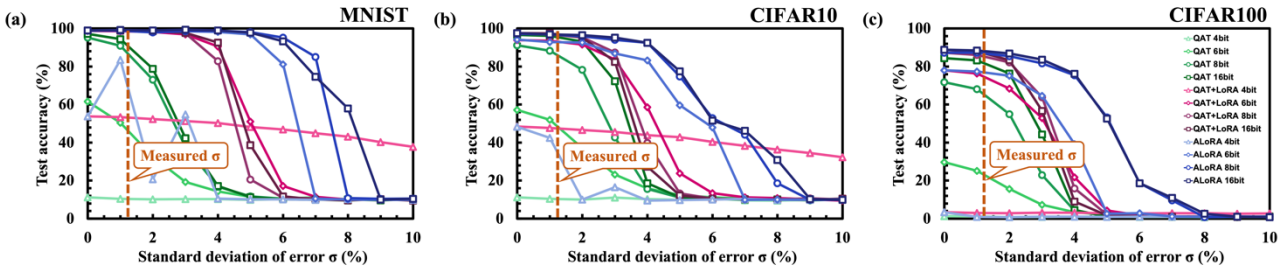

Fig. 4 Test accuracies after ALoRA training as a function of weight error in ViT for (a) MNIST, (b) CIFAR10, and (c) CIFAR100 tasks.

Our proposed method, ALoRA, efficiently combines these methods to enable rapid fine-tuning of large-scale models. When applied to ViT, adding a parallel adapter to layers such as key, query, value, mha output, fc1, fc2, and mlp head reduces the model's 8.7×10^7 parameters to 0.14×10^7 (~1.6%). This means that ViT, which previously required a few days of training on our GPU cluster, can be trained in 40 minutes. Fig. 4 (a)–(c) shows the test accuracies of the ViT after 5-epoch ALoRA training as a function of the weight error for MNIST, CIFAR10, and CIFAR100. First, we can see that QAT improves robustness against quantization and weight errors compared to PTQ as shown in Fig. 2 (d). Furthermore, QAT+LoRA allows many parameters training with the same resources, resulting in improved quantization tolerance. For example, even with 6-bit quantization for MNIST and CIFAR10 and 8-bit quantization for CIFAR100, we achieved accuracy comparable to 16-bit quantization. Finally, we found that ALoRA further improves robustness against quantization and weight errors. The fine-tuned ViT model became highly robust against both quantizing and weight errors, suggesting that the PNNs can scale to practical scale models even under the observed error levels.

V. Conclusions

In this study, we investigated the impact of analog errors on large-scale PNNs using an experimentally derived error model. Our simulations demonstrated that even low-level analog errors (~1.2%) could significantly degrade inference accuracy, particularly for complex models such as vision transformers. To address this issue, we proposed and evaluated a hardware-aware fine-tuning method, ALoRA, which efficiently mitigates accuracy degradation while requiring only ~1.6% of parameters to be trained. Our findings highlight the scalability of PNNs despite inherent analog errors, suggesting that the proposed fine-tuning approach enables the practical deployment of large-scale photonic deep learning models. The combination of QAT, NAT, and LoRA within ALoRA enhances robustness against quantization and weight errors, making it a promising solution for energy-efficient, high-performance photonic computing. Future work includes extending ALoRA to broader model architectures and further optimizing training efficiency for various applications.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number JP 22H05194, Japan.

REFERENCES

[1] K. Kitayama, M. Notomi, M. Naruse, K. Inoue, S. Kawakami, and A. Uchida, "Novel frontier of photonics for data processing - Photonic accelerator -," APL Photonics, vol. 4, no. 9, pp.090901, Sep. 2019.

[2] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, "Deep learning with coherent nanophotonic circuits," Nature Photonics, vol. 11, no. 7, pp. 441-446, July 2017.

[3] K. Ikeda, M. Nakajima, S. Kita, A. Shinya, M. Notomi, and T. Hashimoto, "High-Fidelity WDM-Compatible Photonic Processor for Matrix-Matrix Multiplication," CLEO, Technical Digest Series (Optica Publishing Group, 2024), paper JTh2A.87, 2024.

[4] A. N. Tait, T. F. de Lima, E. Zhou, A. X. Wu, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Neuromorphic photonic networks using silicon photonic weight banks," Scientific Reports, vol. 7, no. 1, pp. 7430, 2017.

[5] M. Nakajima, K. Tanaka, K. Inoue, K. Nakajima and T. Hashimoto, "Densely Parallelized Photonic Tensor Processor on Hybrid Waveguide/Free-Space-Optics," International Conference on Photonics in Switching and Computing (PSC), 2023.

[6] V. Shah and N. Youngblood, "AnalogVNN: A fully modular framework for modeling and optimizing photonic neural networks," APL ML, vol. 1, no. 2, pp. 026116, 2023.

[7] L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, "Deep physical neural networks trained with backpropagation," Nature, vol. 601, no. 7894, pp. 549-555, 2022.

[8] M. Nakajima, K. Inoue, K. Tanaka, Y. Kuniyoshi, T. Hashimoto, and K. Nakajima, "Physical deep learning with biologically inspired training method: gradient-free approach for physical hardware," Nat. commun., vol. 13, no. 1, pp. 7847, 2022.

[9] M. Moralis-Pegios, G. Mourgias-Alexandris, A. Tsakyridis, G. Giamougiannis, A. Totovic, G. Dabos, N. Passalis, M. Kirtas, T. Rutirawut, F. Y. Gardes, A. Tefas, and N. Pleros, "Neuromorphic silicon photonics and hardware-aware deep learning for high-speed inference," Journal of Lightwave Technology (JLT), vol. 40, no. 10, pp. 3243-3254, 2022.

[10] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018, pp.2704-2713.

[11] M. J. Rasch, C. Mackin, M. L. Gallo, A. Chen, A. Fasoli, F. Odermatt, N. Li, S. R. Nandakumar, P. Narayanan, H. Tsai, G. W. Burr, A. Sebastian, and V. Narayanan, "Hardware-aware training for large-scale and diverse deep learning inference workloads using in-memory computing-based accelerators," Nature Communications, vol. 14, Aug. 2023, pp. 5282.

[12] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," Proc. of Int. Conf. on Learning Representations (ICLR), Apr. 2022.

[13] A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, "QLoRA: Efficient Finetuning of Quantized LLMs," Proc. of Advances in neural information processing systems (NeurIPS), Dec. 2023.

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," Proc. of Advances in neural information processing systems (NeurIPS), Dec. 2019, pp. 8024–8035.

[15] https://huggingface.co/

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 770-778.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Proc. of 9th Int. Conf. on Learning Representations (ICLR), Jan. 2021.

[18] https://huggingface.co/google/vit-base-patch16-224