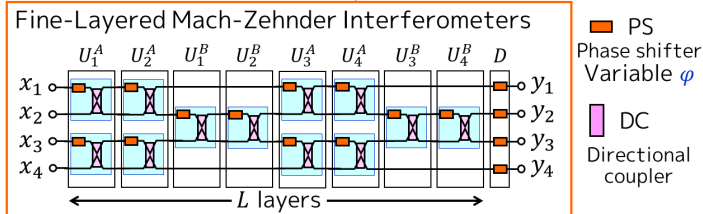
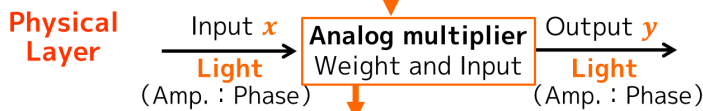
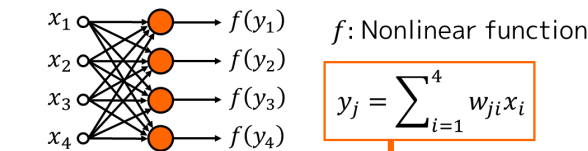


Abstract

An optical neural network (ONN) is a promising system due to its high-speed and low-power operation. The ONN has a multiple-layered structure of programmable Mach-Zehnder interferometers (MZIs). Due to this structure, it takes a lot of time to learn MZI parameters with a conventional automatic differentiation (AD). To solve the time-consuming problem, we develop a function module implemented in C++ to collectively calculate input-output values in a multiple-layered structure, where novel customized derivatives for an MZI are utilized in backpropagation. We demonstrate that our learning method works 50 times faster than the conventional AD when a pixel-by-pixel MNIST task is performed in a complex-valued recurrent neural network. Our approach supports ONN design and contributes to realize green-computing AI's instead of conventional ones consuming a lot of energy.

Optical Neural Network

Conventional Neural Network



Mathematical Model

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = D \dots \begin{pmatrix} 1 & U_1^B & 0 & 0 \\ 0 & U_{1[1]}^B & 0 & 0 \\ 0 & 0 & U_{2[1]}^A & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U_2^A & 0 & 0 \\ 0 & 0 & U_{2[2]}^A \\ 0 & 0 & 0 & U_{1[1]}^A \\ 0 & 0 & 0 & U_{1[2]}^A \end{pmatrix} \begin{pmatrix} U_1^A & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & U_{1[1]}^A \\ 0 & 0 & 0 & U_{1[2]}^A \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \uparrow \text{dim}$$

$$U_{p[q]}^R = \frac{1}{\sqrt{2}} \begin{pmatrix} e^{i\varphi_{p[q]}^R} & i \\ ie^{i\varphi_{p[q]}^R} & 1 \end{pmatrix} \quad U_p^R: \text{Unitary matrix} \quad R = A, B$$

$$\varphi_{p[q]}^R: \text{Parameter} \quad \begin{matrix} 1 \leq p \leq L/2 \\ 1 \leq q \leq \text{dim}/2 \end{matrix}$$

Problem to Solve

Physical restriction: Difficulty in manufacture of large-scale circuits
 → Use of recurrent neural networks (RNN)
 Fine-layered structure: One layer → One linear circuit
 → **Learning very deep neural networks**
A lot of computational time required by the conventional automatic differentiation (AD)

Accelerated Learning Method

Key 1. Customized derivatives: CD

Update of parameter: $\varphi \leftarrow \varphi - \eta \left(\frac{\partial L}{\partial \varphi} \right)$ L : Loss func. η : Learning rate

Forward

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} e^{i\varphi} & i \\ ie^{i\varphi} & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Backward

Conjugate transpose

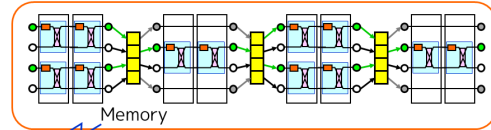
$$\begin{pmatrix} \frac{\partial L}{\partial x_1^*} \\ \frac{\partial L}{\partial x_2^*} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} e^{-i\varphi} & -ie^{-i\varphi} \\ -i & 1 \end{pmatrix} \begin{pmatrix} \frac{\partial L}{\partial y_1^*} \\ \frac{\partial L}{\partial y_2^*} \end{pmatrix}$$

$$\frac{\partial L}{\partial \varphi} = 2 \cdot \text{Im} \left(x_1^* \frac{\partial L}{\partial x_1^*} \right)$$

Update by multiplication of **only two values**

Key 2. Pointer Rewiring in C++: PR

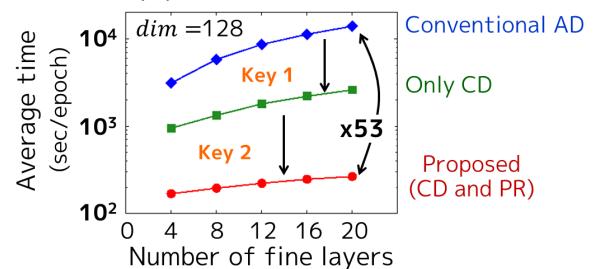
Fine-layered structure → one function module in C++



Data read by direct access to stored-data address

Experimental Results

- RNN with optical-circuit hidden unit
- Pixel-by-pixel MNIST classification task



References

[1] K. Aoyama, H. Sawada, "Accelerated method for learning fine-layered optical neural networks," in *Proc. of IEEE/ACM the 40th International Conference on Computer-Aided Design*, 2021.

Contact

Kazuo Aoyama / Learning and Intelligent Systems Research Group, Innovative Communication Laboratory
 Email: cs-openhouse-ml@hco.ntt.co.jp