Diffractive optical computing with the characteristics of complex neural networks

[a]ChengLiang Deng

[a]School of Physics and Optoelectronic Engineering, Beijing University of Technology, Beijing 100124, China

ABSTRACT

Traditional neural networks are trained and simulated with real values, while diffractive neural networks based on optical diffraction propagation belong to complex-valued operations. Existing studies have shown that complex-valued neural networks exhibit stronger generalization performance and can achieve complex tasks such as image recognition. This paper introduces an optical implementation method for logical operations along with detailed derivation processes, and discusses some characteristics of complex-valued neural networks.

1. Introduction

In the post-Moore era, the performance improvement of electronic chips has slowed down. However, the further maturity of technologies such as cloud computing, big data, and the Internet of Things has generated massive data, which has increased the huge demand for backend computing power. Optical computing is an ideal tool for massive data processing due to its characteristics of high bandwidth, high processing speed, and low power consumption. Traditional neural networks are limited by the computing speed of electronic chips, while diffractive optical neural networks perform data operations at the speed of light, making them the fastest computing architecture for certain single problems except for quantum computing. Current optical computing is mainly divided into on-chip integrated photonic chips and free-space diffractive neural networks. Optical computing integrated on photonic chips has enabled efficient convolution operations[1], matrix multiplication[2], and parallel tensor computation cores[3].Addition, it is difficult to set the function of nonlinear in Free-space diffractive neural networks. But, Free-space diffractive neural networks have realized the functions of handwritten digit image recognition and nonlinear dataset classification [4-7].

Diffractive neural networks (DNNs) are, in essence, a multi-layered structure composed of computer-generated phase holograms spaced at fixed intervals. These networks process optical fields modulated by input values through multiple layers of diffraction, enabling end-to-end computation. Unlike traditional neural networks that rely on real-valued operations, DNNs manipulate the phase and amplitude of propagating optical fields, with all internal parameters represented as complex numbers. This complex-valued framework introduces unique properties absent in real-valued networks, providing theoretical insights into tasks such as image classification. Section 2.3 of this paper delves into this topic. Currently, the vast majority of existing neural networks rely entirely on real-valued arithmetic, whereas complex arithmetic may offer a significant advantage. For instance, the detection of symmetry problem and XOR problem can be easily solved by a single complex-valued neuron with orthogonal decision boundaries, but cannot be done with a single real-valued neuron[8]. Meanwhile, recent studies suggest that complex-valued arithmetic would significantly improve the performance of neural networks by offering rich representational capacity[9], strong generalization[10].

2. Methods

2.1 Parallel Logical Operation

This section presents an optical implementation method for parallel logical operations. Multiple layers of phase-only modulation surfaces are configured, each taking 16 logical operation modes as a single instruction set input to control network computation. For the image pairs to be computed inserted into the intermediate layer, the mean squared error between the network output images and the theoretically computed values is calculated. When training with N pairs of input images, the average gradient of the output errors (16N in total, corresponding to all logical operations for all input image pairs) is computed during one iteration. This gradient is then backpropagated to update the phase values. Notably, since the parameters here are complex numbers, the gradient formula for backpropagation is a crucial component, which will be discussed in Section 2.2. As illustrated in Figure 1 left, the computation processes under three input modes (AND, OR, and NOR) are shown. The trained network can perform logical operations on arbitrary corresponding positions of image pairs in the intermediate input layer via diffraction, which is referred to as parallel logical operations. The right figure shows the theoretical calculation result (b) and the simulated calculation result (c) of the network for the input image pair (a).

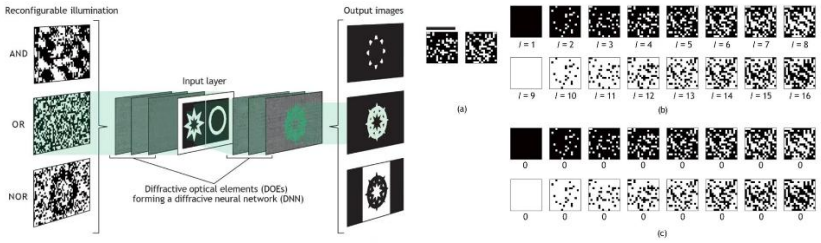

Fig1. The principle of diffraction casting[11]

Table 1 Logic operations defined on input pair.

Input pair f_{left} f_{right}	Operation index l	Logic operation	Boolean 0 0 1 1 / 0 1 0 1	Operation index l	Logic operation	Boolean 0 0 1 1 / 0 1 0 1
Output $\hat{g}_l$	1	0	0 0 0 0	9	1	1 1 1 1
	2	$f_{left} \wedge f_{right}$ (AND)	0 0 0 1	10	$\overline{f_{left} \wedge f_{right}}$ (NAND)	1 1 1 0
	3	$f_{left} \wedge \overline{f_{right}}$	0 0 1 0	11	$\overline{f_{left}} \vee f_{right}$	1 1 0 1
	4	f_{left}	0 0 1 1	12	$\overline{f_{left}}$	1 1 0 0
	5	$\overline{f_{left}} \wedge f_{right}$	0 1 0 0	13	$f_{left} \vee \overline{f_{right}}$	1 0 1 1
	6	f_{right}	0 1 0 1	14	$\overline{f_{right}}$	1 0 1 0
	7	$f_{left} \oplus f_{right}$ (XOR)	0 1 1 0	15	$\overline{f_{left} \oplus f_{right}}$ (XNOR)	1 0 0 1
	8	$f_{left} \vee f_{right}$ (OR)	0 1 1 1	16	$\overline{f_{left} \vee f_{right}}$ (NOR)	1 0 0 0

Table 1 shows the 16 logical operations from Ref. [1] that control the network's computation on input image pairs, where f_{left} and f_{right} denote the left and right images of the input pair, respectively. For example, the AND operation result for the entries (0, 0, 1, 1) and (0, 1, 0, 1) within the red box is shown as (0, 0, 0, 1) at the intersection of the dashed lines.

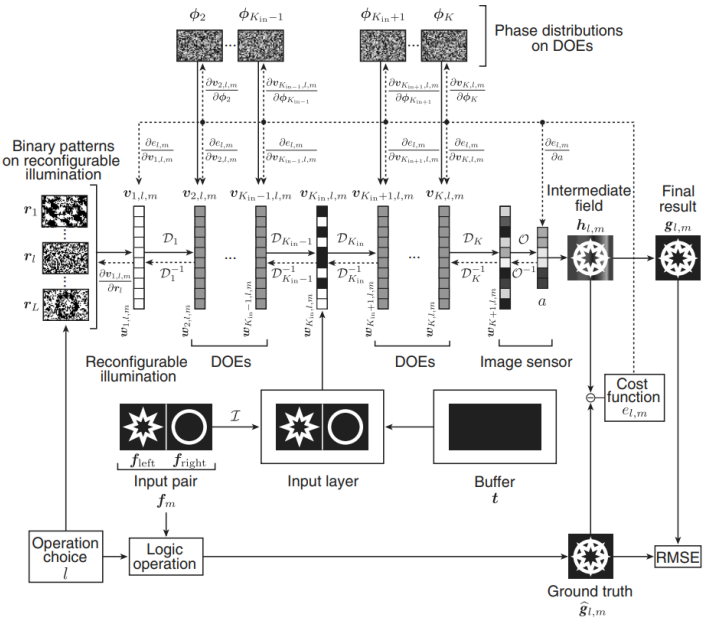

Fig2. model training

As shown in Figure 2, for the M pairs of input images trained under the 16 computation modes, the average error obtained is expressed by Equation (1).

$$E = \frac{1}{LM}\sum_{l,m} e_{l,m} \tag{1}$$

Calculate the gradient of the calculation error with respect to the input mode parameter r that controls the computation, and respect to the phase parameters of each layer.

$$\frac{\partial E}{\partial r_l} = \sum_{m} \frac{\partial e_{l,m}}{\partial r_l} \tag{2}$$

$$\frac{\partial E}{\partial \phi_k} = \sum_{l,m} \frac{\partial e_{l,m}}{\partial \phi_k} \tag{3}$$

For the phase update of the k-th layer, the gradient descent method is used to derive the phase value at the $j+1$ -th iteration from the value at the j-th iteration.

$$\phi^{j+1} = \phi^{j} - \alpha \cdot Real\left(\frac{\partial E}{\partial \phi_k}\right) \tag{4}$$

2.2 Gradient Descent

Diffraction propagation is a strictly linear transformation process. The derivation of the gradient descent formula for diffraction propagation can be analogously obtained by studying the gradient under Fourier transform. As shown in Figure 3, it is a visualization of the computational graph of the Fourier propagation operator F. It can be seen that the complex amplitude of each point on the CCD is a weighted sum of all points on the input plane U. The square corresponding to the black arrow in the upper left corner represents $A_{u,v}$.

$$A_{u,v} = FFT(U) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} U_{m,n} \cdot e^{-j\,2\pi\left(\frac{mu}{M}+\frac{mv}{N}\right)} \tag{5}$$

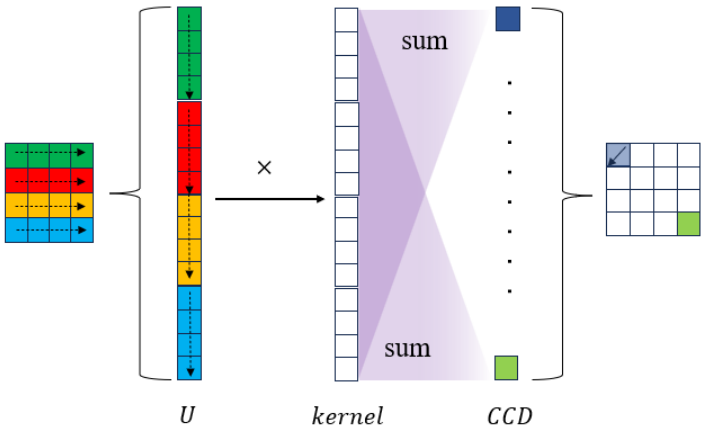

Fig3. Fourier Transform

Let the complex amplitude light field propagating to the CCD plane be denoted as A. The error of an image with the size of $M \times N$ can be expressed as:

$$E = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}(AA^* - I)^2 \tag{6}$$

Calculate the derivative of E with respect to a single element $A_{u,v}$on the recording plane A, which is expressed as:

$$\frac{\partial E}{\partial A_{u,v}} = \frac{2}{MN}\left[\left(A_{u,v}A_{u,v}^* - I_{u,v}\right)A^*(u,v) + \left(A_{u,v}A_{u,v}^* - I_{u,v}\right)A_{u,v}\cdot\frac{\partial A_{u,v}^*}{\partial A_{u,v}}\right] \tag{7}$$

If A and its complex conjugate A^* are expressed as follows:

$$A = x + y\,i \quad , \quad A^* = x - y\,i$$

$$x = \frac{A + A^*}{2} \quad , \quad y = \frac{A - A^*}{2i}$$

$$\frac{\partial A^*}{\partial A} = \frac{\partial A^*}{\partial x}\frac{\partial x}{\partial A} + \frac{\partial A^*}{\partial y}\frac{\partial y}{\partial A} = 1\cdot\frac{1}{2} - i\cdot\frac{1}{2i} = 0 \tag{8}$$

Substituting Equation (8) into Equation (7) and arranging the terms gives:

$$\frac{\partial E}{\partial A_{u,v}} = \frac{2}{MN}\left[\left(A_{u,v}A_{u,v}^* - I_{u,v}\right)A^*(u,v)\right] \tag{9}$$

Each element $A_{u,v}$ on the recording plane is a function of the phase complex amplitudes $U_{m,n}$, denoted as Equation (10). Meanwhile, the error function E is a function of all complex amplitude elements $A_{u,v}$ denoted as Equation (11).

$$A_{u,v} = FFT\left(\dots,\ U_{m,n},\ \dots\right) \tag{10}$$

$$E = f\left(\dots,\ A_{u,v},\ \dots\right) \tag{11}$$

The partial derivative of the recording plane $A_{u,v}$ with respect to a single phase complex amplitude element $U_{m,n}$ is expressed as Equation (12), and the partial derivative of the phase complex amplitude $U_{m,n}$ with respect to the phase element $\phi_{m,n}$is expressed as Equation (13).

$$\frac{\partial A_{u,v}}{\partial U_{m,n}} = e^{-j\,2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)} \tag{12}$$

$$\frac{\partial U_{m,n}}{\partial \phi_{m,n}} = j\,e^{j\,\phi_{m,n}} \tag{13}$$

Therefore, we can derive the derivative of the error E with respect to a single complex amplitude element $U_{m,n}$.

$$\frac{\partial E}{\partial U_{m,n}} = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left\{\frac{2}{MN}\left[\left(A_{u,v}A_{u,v}^{*} - I_{u,v}\right)A^{*}(u,v)\cdot e^{-j\,2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}\right]\right\}$$

$$= \frac{2}{MN}\left\{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left[\left(A_{u,v}A_{u,v}^{*} - I_{u,v}\right)A_{u,v}\cdot e^{j\,2\pi\left(\frac{mu}{M}+\frac{nv}{N}\right)}\right]\right\}^{*}$$

$$= \frac{2}{MN}\left\{FFT^{-1}\left((AA^{*} - I)A\right)\right\}^{*}_{\mathrm{m,n}} \tag{14}$$

Here, the summation process reflects the inverse Fourier transform. Therefore, the summation operation for elements is converted into performing an inverse Fourier transform on the error image matrix, and the value corresponding to (m,n) after the transform is taken as the derivative gradient of the error E with respect to a single element$U_{m,n}$.

By using the chain gradient rule and combining with Equation (13) and (14), the gradient with respect to a single-phase value $\phi_{m,n}$ can be further obtained.

$$\frac{\partial E}{\partial \phi_{m,n}} = \frac{\partial E}{\partial U_{m,n}}\cdot\frac{\partial U_{m,n}}{\partial \phi_{m,n}}$$

$$= \frac{2}{MN}\left\{FFT^{-1}\left((AA^{*} - I)A\right)\right\}^{*}_{\mathrm{m,n}}\cdot j\,e^{j\,\phi_{m,n}}$$

$$= \left[\frac{2}{MN} FFT^{-1}\left((AA^* - I)A\right) \cdot \left(-j\, e^{-j\,\phi_{m,n}}\right)\right]_{m,n}^{*} \tag{15}$$

Since the phase is a real number, updating it requires taking the real part of the gradient. The gradient matrix propagated to the phase plate can ultimately be expressed as:

$$Real\left(\frac{\partial E}{\partial \phi}\right) = Real\left[\frac{2}{MN} FFT^{-1}\left((AA^* - I)A\right) \cdot (-j\, U^*)\right] \tag{16}$$

Now, by replacing the Fourier transform operator with the diffraction propagation operator D that also performs linear operations, and considering the influence of multi-layer propagation, the gradient formula for the k-th layer can be generalized as:

$$Real\left(\frac{\partial E}{\partial \phi_k}\right) = Real\left[\frac{2}{MN} w_k^* D^{-1}\left[\cdots\left[U^*\left[D^{-1}\left((AA^* - I)A\right)\right]\right]\right] \cdot (-j\, U_k^*)\right] \tag{17}$$

According to the average error calculation of the gradient in Section 2.1, substituting it into Equation (4) can complete the update operation.

2.3 Complex-Valued Operation

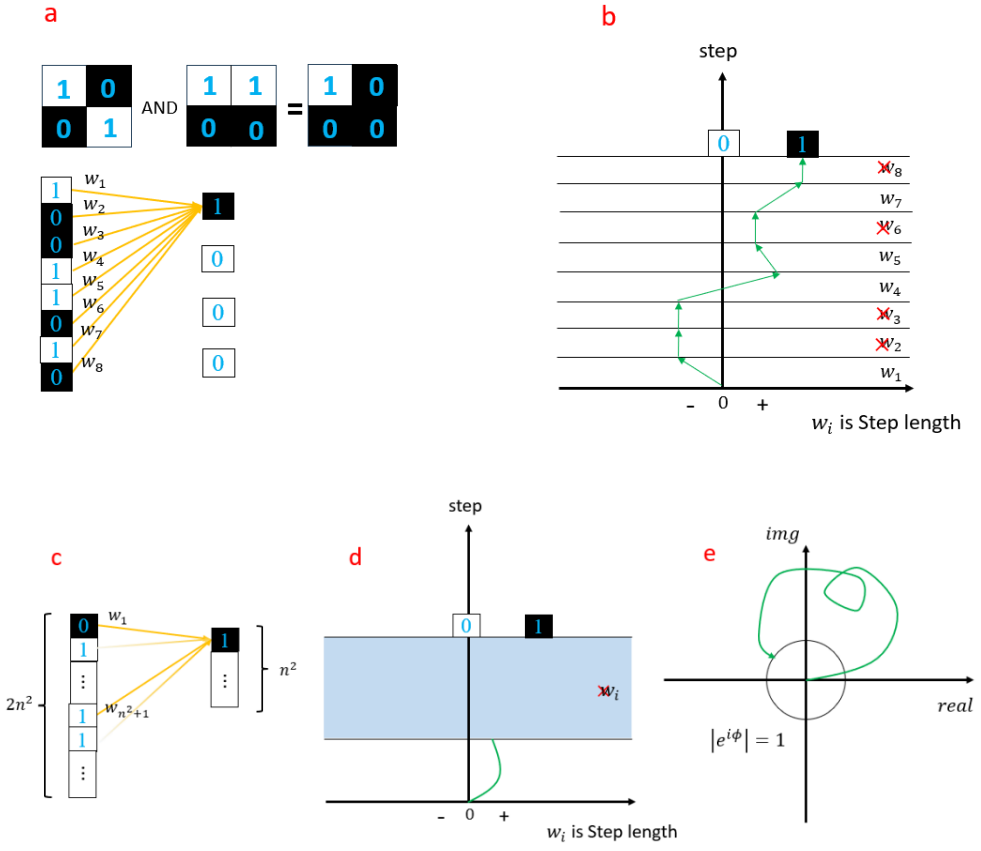

Fig4.complex neural networks. (a)Network model for 8 Boolean operations; (b)

Calculation process; (c) Network model for n×n×2 input image pairs; (d) Calculation procedure; (e) Computational path on the complex plane

In real-valued neural networks, by expanding images into Boolean vectors column-wise and connecting them, basic logical operations can be implemented through network propagation—including the XOR operation, which was previously thought to require nonlinear networks. As shown in Figure (b), the abscissa represents step sizes for left/right movements using weights. Multiplying by a Boolean value of 1 selects the corresponding weight for movement, while 0 indicates no selection, visualized as small vertical lines. After completing 8 summation steps, we successfully reach the AND result (1) for the first pair of elements. When handling more data, as in Figure (c), weights in areas where the input image pairs' common regions equal 0 are not selected during training. We denote these as shadow regions, as shown in Figure (d). Notably, during backpropagation updates, all weights are updated—including shadow regions—suggesting these areas play a key role in the network's generalization performance by providing more stable path choices. This feature is enhanced in complex-valued neural networks: as in Figure (e), weight step sizes become complex numbers, with each computed value corresponding to a path in the complex plane. Since optical diffraction calculations receive intensity information, the goal is achieved if the path endpoints lie on the unit circle (modulus 1), endowing complex-valued networks with stronger generalization and robustness. Adding a nonlinear activation function introduces an acceptable perturbation range for each step: step sizes remain valid as long as they do not exceed the threshold. This, consistent with traditional understanding, enhances the network's generalization ability.

It should be noted that whether XOR operations, other nonlinear operations, or classification can be performed is related to the encoding method of inputs, and there is no absolute distinction. In the preceding text, we merged two sets of Boolean vectors to be calculated, increasing the degrees of freedom of the network to realize the XOR operation. Here, we will discuss the orthogonal decision boundary theory of general complex-valued neural networks to illustrate the differences between real-valued and complex-valued networks in XOR operations. Encode the AND operation inputs of (1, 0, 0, 1) and (1, 0, 1, 0) into four complex inputs (1+i, -1-i, -1+i, 1-i), where 1 corresponds to 1 and 0 corresponds to -1. The number of hidden layers is 0. The output consists of four values.

The input vector can be represented as a complex vector $X = x + yi$ (where x=[x1,x2,...],y=[y1,y2,...]), the weights as $w = u + vi$ (where u=[u1,u2,...],v=[v1,v2,...]), and the output will be written as:

$$Y = [u \quad -v]\begin{bmatrix} x \\ y \end{bmatrix} + [v \quad u]\begin{bmatrix} x \\ y \end{bmatrix} i \ \ (Y \in C^2 | Y = Y_1, Y_2, \dots) \tag{18}$$

At this point, the decision boundaries for the real part and the imaginary part can be expressed by equations (19) and (20), respectively.

$$xu - vy = R^r \tag{19}$$

$$xv + uy = R^i \tag{20}$$

They are orthogonal to each other, referred to as orthogonal decision boundaries. The two orthogonal boundary lines divide the output plane into four sections. If the weights remain unchanged and $u = 0$, $v = 1$, $R^r = R^i = 0$, then the decision division as shown in Figure 5 is obtained.

Input	Output
$X = x + yi$	Y= $xu - vy + (xv + uy)i$
1+i	-1+i
-1-i	1-i
-1+i	-1-i
1-i	1+i

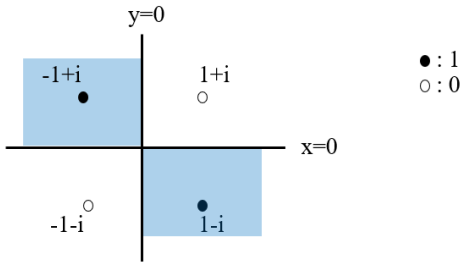

Fig5. orthogonal decision boundaries

The orthogonal decision boundary can still successfully divide the input region into four categories with the maximum number of classifications when the weight change is minimal. By defining the output values of the diagonal regions as equivalent, the correct classification result of the XOR operation can be obtained. However, a two-layer real-valued neural network only has a single linear decision boundary, which cannot distinguish the two sets of elements on the diagonal. It should be noted, though, that a multi-layer real-valued neural network with activation functions can implement the XOR operation.

3. Conclusion

Diffractive neural network optical computing is one of the leading methods for achieving high-speed parallel computing today. The characteristics of its complex neural network enable it to exhibit excellent performance in image recognition and logical operations in optical systems where nonlinear activation functions are difficult to implement. In addition, research on diffractive neural networks has also promoted research on complex neural networks. Mathematicians and engineers have made full use of complex analysis theory to study the properties of complex neural networks, providing a profound theoretical foundation for future research and applications such as optical computing that specifically handle the complex amplitude properties of light fields.

References

[1] Zhu, H.H., Zou, J., Zhang, H. et al. Space-efficient optical computing with an integrated chip diffractive neural network. Nat Commun 13, 1044 (2022).

[2] Bandyopadhyay, S., Sludds, A., Krastanov, S. et al. Single-chip photonic deep neural network with forward-only training. Nat. Photon. 18, 1335–1343 (2024).

[3] Feldmann, J., Youngblood, N., Karpov, M. et al. Parallel convolutional processing using an integrated photonic tensor core. Nature 589, 52–58 (2021).

[4] Xing Lin et al. ,All-optical machine learning using diffractive deep neural networks.Science361,1004-1008(2018).

[5] Liu, C., Ma, Q., Luo, Z.J. et al. A programmable diffractive deep neural network based on a digital-coding metasurface array. Nat Electron 5, 113–122 (2022).

[6] Qian, C., Lin, X., Lin, X. et al. Performing optical logic operations by a diffractive neural network. Light Sci Appl 9, 59 (2020).

[7] Xue, Z., Zhou, T., Xu, Z. et al. Fully forward mode training for optical neural networks. Nature 632, 280–286 (2024).

[8] T. Nitta, “Orthogonality of Decision Boundaries in Complex-Valued Neural Networks,” Neural Computation, Vol. 16, No. 1, 2004, pp. 73-97.

[9] Reichert, D. P. & Serre, T. Neuronal synchrony in complex-valued deep networks. Preprint at https://arxiv.org/abs/1312.6115v5 (2013).

[10] A. Hirose and S. Yoshida, "Generalization Characteristics of Complex-Valued Feedforward Neural Networks in Relation to Signal Coherence," in IEEE Transactions on Neural Networks and Learning Systems, vol. 23, no. 4, pp. 541-551, April 2012.

[11] R. Mashiko, M. Naruse, and R. Horisaki, "Diffraction casting," Adv. Photon. 6, (2024).