Technische Universität Dresden
Fakultät Elektrotechnik und Informationstechnik
Institut für Nachrichtentechnik
Juniorprofessur für Integrierte Photonische Bauelemente

Hauptseminar

Simulation of a Perceptron Based on Optical Devices

Yunbin Shen, Ming Ke
China

Betreuer: M.Sc. Mircea Catuneanu
Integrated Photonic Devices group

Ort, Datum Dresden, 26.07.2018

Simulation of a Perceptron Based on Optical Devices

Require

The goal of this project is to create a framework for the analysis of all-optical ANN. This architecture will theoretically, not be limited by electronic clock rates and ohmic losses. The framework will help in understanding and simulating future photonic chips. The student(s) will coordinate with the Integrated Photonic Devices team in simulating existing theoretical models – for example the circuit of a perceptron, which is a simple version of a neuron. The simulated models will be used in a simple classification problem for proofing and performance analysis.

The implementation should be realized in Python preferably in the Anaconda distribution.

Abstract

Our project is to create a framework for the analysis of optical ANN. The framework will help in understanding and simulating future photonic chips. We realize the simulation of a neuron based on the optical architecture by Python-programming. Nonlinear photonic circuit models based on existing theories are used to build a perceptron. This simulated perceptron is applied to solve a simple problem for proofing and performance analysis.

Keywords: optical information processing, integrated photonics, perceptron, Python

Contents

List of Abbreviations

ANN	Artificial Neural Networks
ONN	Optical neural network
MRR	Microring resonator
OPO	Optical parametric oscillator
NOPO	Nondegenerate optical parametric oscillator
MZI	Mach-Zehnder interferometer

1 Introduction

Artificial Neural Networks (ANN) are a great combination of our available processors and models inspired by deep research in the biologic neural networks. In the modern world Deep Learning is applied in a wide range of fields to make everything intelligent. ANNs play an important role in Deep Learning and they are usually used to improve the performance of learning tasks such as speech and image recognition. Compared to an all-electronic implementation of such network, an all-optical one would greatly increase power efficiency. Also, hybrid opto-electronic or even all-optical ones have better performance (speed-wise), so in our project we focus on it.

The performance of neural networks built with electronic components is limited by electronic clock rates, ohmic losses, problems of electromagnetic compatibility (EMC), etc. Traditional central processing units (CPUs) are not optimal for realizing the neural network, because they are not totally suitable for the implemented algorithms that need more parallel computing. Therefore, special neuromorphic architectures have been developed, such as Graphical processing units (GPUs), application-specific integrated circuits (ASICs), and field-programmable gate arrays (FPGAs). They have higher speed and energy efficiency for learning tasks. But compared to optical architectures, they still need much more energy and have higher latency. But, once an optical neural network(ONN) is trained, the architecture can be passive, and computation on the optical signals will be performed without additional energy input. These features could enable ONNs that are substantially more energy-efficient and faster than their electronic counterparts.1

The perceptron built with nonlinear photonic circuit consists of four main components, programmable amplifiers, a static mixing element, a quadrature filter and a thresholder.2 In our program we modularize the perceptron circuit. And it is divided into couplers (beamsplitters), microring resonators (MRRs), an optical parametric oscillator (OPO). These elements build up a programmable gain amplifier and a thresholder. Chapter 2 presents the components above. The static mixing element and the quadrature filter are not in the scope of our discussion. Chapter 3 presents results of our simulation and conclusions.

2 The Simulated Perceptron

In this chapter we discuss the basic perceptron learning algorithm and describe main optical components.

2.1 The perceptron algorithm

The perceptron is a machine learning algorithm. The structure of one perceptron is visualized in Figure 2.1.1. The input data x_n is multiplied by the weights w_n. Then the summary of these results will be substituted into a nonlinear function f (2.1.1) to get the result.

$$t_{result} = f\left(\sum_{i=1}^{n} a_i \cdot w_i\right) \tag{2.1.1}$$

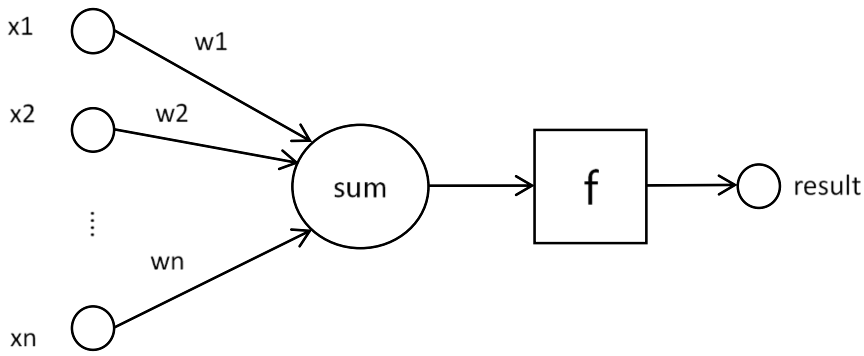


Fig2.1.1

At first, the weights w_n are randomly selected, therefore the output value t compared to the expected value usually have a big deviation. So, we should change the value of w_n to make this deviation smaller. We defined this deviation as e_{error} and can be derived from the equation 2.2.

$$e_{error} = t_{expected} - t_{result} \tag{2.1.2}$$

If the expected value is bigger than the result, we need to increase the weights, if it's smaller, we need to decrease them. The changed weights $w_n{}'$ can be derived from the equation (2.1.3).

$$w_n' = w_n + e_{error} \cdot x_n \cdot \eta \qquad \eta \in \mathrm{R} \tag{2.1.3}$$

with η is the learning rate, which is manually given and will also influence the precision of the result. Form the equation 2.3 the weights can be changed rapidly or gently according to the value of the e_{error} to improve the result in the next iteration.

In our project, we will try to use all optical instruments to realize these mathematical equations. In the section 2.2 and section 2.3 present the coupler and micro ring resonator (MRR), which are the basic components of an optical amplifier. Then, the section 2.4 introduce the optical amplifier, which can enlarge the inputted power with a large bias power. The next section 2.5 present the programmable amplifier, which is based on the optical amplifiers. Finally, the section 2.6 talk about the optical thresholder, which is a special type of optical amplifier.

2.2 Coupler

The coupler that is also called beam-splitter is a part of an optical amplifier circuit. Figure 2.2.1 demonstrates the structure of the coupler. The coupler mixes two input fields and has two output fields. The relationship between both fields are parametrized by a matrix about a mixing angle θ,

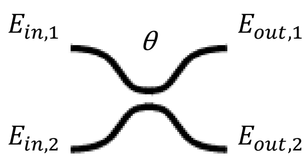


Figure 2.2.1

$$\begin{pmatrix} E_{out,1} \\ E_{out,2} \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} E_{in,1} \\ E_{in,2} \end{pmatrix} \tag{2.2.1}$$

2.3 Microring Resonator

In our project all-pass ring resonators are used and their input is one output of a coupler (see

Figure 2.3.1).

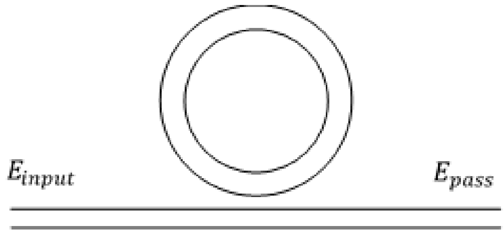


Figure 2.3.1

Two main parameters, the circulating power buildup factor B and single pass phase shift ϕ, are needed to describe the feature of the resonator. The resonance relationship of a linear MRR with the self-coupling coefficient r and the power attenuation coefficient α [1/cm] ($a^2 = \exp(-\alpha L)$, L is the round-trip length) can be expressed

$$B(\phi) = \frac{a^2(1-r^2)}{1-2ar\cos\phi+a^2r^2} \ . \tag{2.3.1}$$

In a cavity of MRR, the phase shift from the original ϕ_0 of the small signal depends on BP. The resonance relationship can be expressed

$$B(\phi) = \frac{\phi-\phi_0}{P}\frac{\lambda A}{2\pi L n_2} \tag{2.3.2}$$

where A is the power coupling ratio, n_2 is an intensity-dependent refractive index, λ is the free space wavelength, and the round-trip length L is 2πR for a MRR of a radius R.

A transcendental equation of ϕ can be acquired from the equations (2.3.1) and (2.3.2),

$$B(\phi) = \frac{\phi-\phi_0}{P}\frac{\lambda A}{2\pi L n_2} = \frac{a^2(1-r^2)}{1-2ar\cos\phi+a^2r^2} \approx \frac{a^2(1-r^2)}{1-2ar\left(1-\frac{1}{2}\phi^2\right)+a^2r^2} \tag{2.3.3}$$

In equation (2.3.3) there is a Lorentzian approximation because of the small value of $\cos\phi$ .3 Then a cubic equation can be obtained

$$[ar]\phi^3 + [-ar\phi_0]\phi^2 + [(1-ar)^2]\phi. \ . \ .$$
$$+[-(1-ar)^2\phi_0 - \frac{2\pi L n_2}{\lambda A}a^2(1-r^2)P/A_{eff}] = 0 \tag{2.3.4}$$

From this equation can we get at least one real root, sometimes three real roots. When there are three real roots, for the nonlinear resonator values of P are regions of bistability; the medial root being an unstable solution. The one or two stable solutions for $\phi(P)$ are substituted into

$$t(\phi(P)) = \frac{E_{pass}}{E_{input}} = e^{i(\pi+\phi)} \frac{a - re^{-i\phi}}{1 - are^{i\phi}}. \tag{2.3.5}$$

Here E_{pass} is the electric field intensity E_{input} after the complex transmission of MRR.

2.4 Amplifier

An optical amplifier circuit consists of the above two parts, Figure 2.4.1. There are two MRRs between two couplers. The outputs of the left coupler are the inputs of the MRRs. And the outputs of the MRRs are the inputs of the right coupler. MRR can magnify the passing signal. Here the important parameter is the mixing angle θ at the couplers. Through the matrix with the θ, $E_{input,2}$ can influence the output, so make the amplifier tunable.

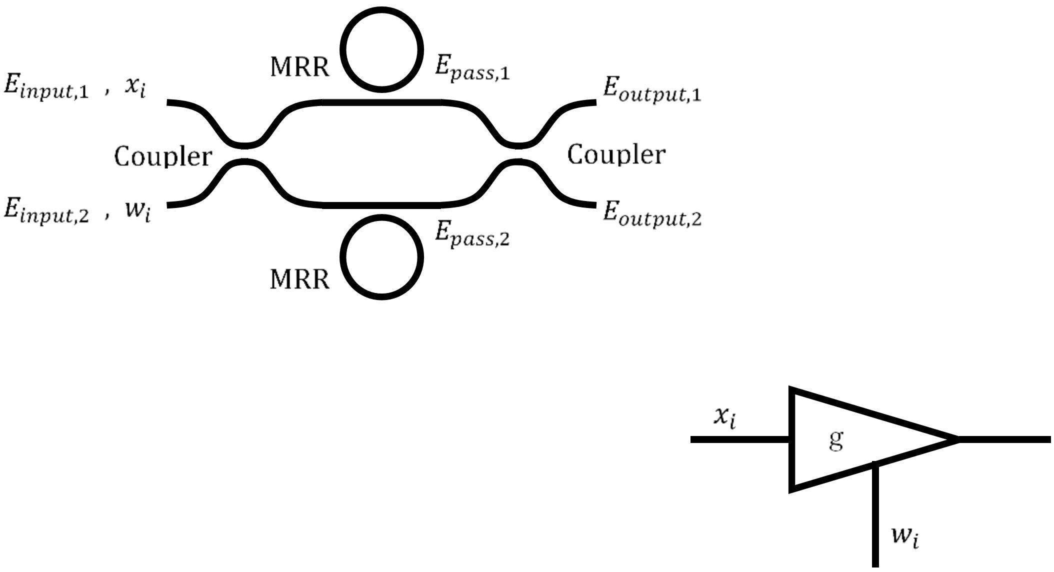


Figure 2.4.1

In our program the amplifiers are built with modular parts. This method is simpler and more efficient. As a tunable gain amplifier, the value of its output is changed with the bias $E_{input,2}$. In the perceptron algorithm, weight w_i is inputted as this bias. The $E_{output,1}$ is what we need,

$$E_{pass,1} = (cos\theta \quad -sin\theta) \begin{pmatrix} E_{input,1} \\ E_{input,2} \end{pmatrix} t(\phi(P)), \tag{2.4.1}$$

$$E_{pass,2} = (sin\theta \quad \cos\theta)\begin{pmatrix} E_{input,1} \\ E_{input,2} \end{pmatrix} t\big(\phi(P)\big), \tag{2.4.2}$$

$$E_{output,1} = (cos\theta \quad -sin\theta)\begin{pmatrix} E_{pass,1} \\ E_{pass,2} \end{pmatrix}. \tag{2.4.3}$$

2.5 Programmable Amplifier

This programmable amplifier is an important and senior component in the optical perceptron circuit. Its programmable function is realized by combining characteristics of nondegenerate optical parametric oscillators(NOPOs) and amplifiers. Firstly, we introduce briefly the NOPO. The signal through the NOPO adds a small vector with unchanged amplitude but different phase, so the module of output has a bias compared to the input signal. Through the following amplifier the signal with different bias has a different gain. In this way, its process is programmable.

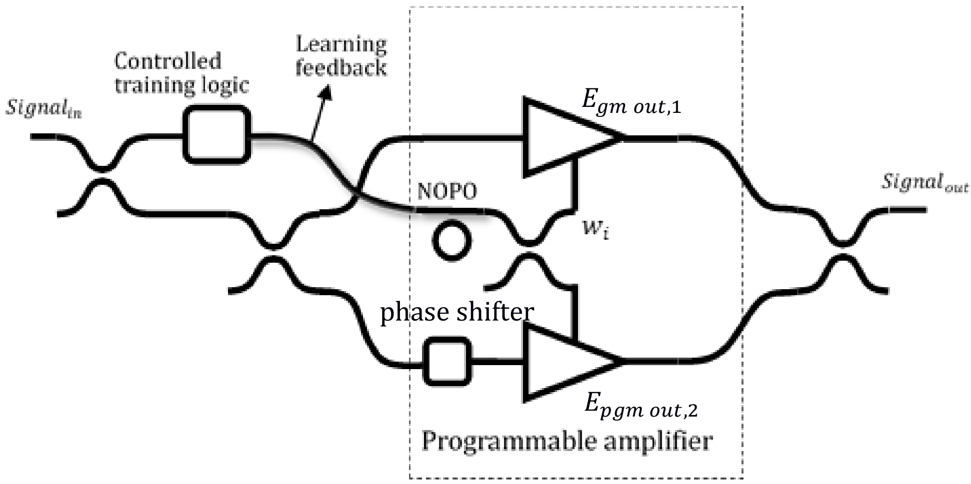


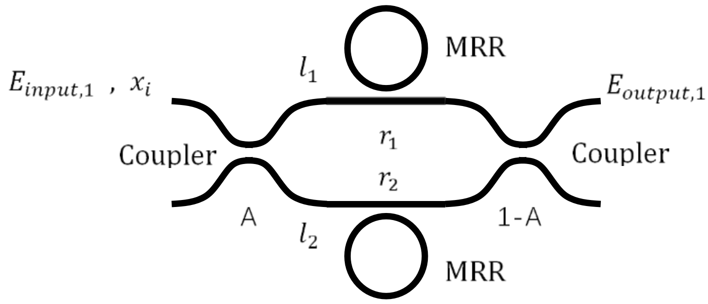
Figure 2.5.1

$$E_{Signal_{out}} = (cos\theta \quad -sin\theta)\begin{pmatrix} E_{pgm\ out,1} \\ E_{pgm\ out,2} \end{pmatrix}. \tag{2.5.1}$$

The amplifier blow in Figure 2.5.1 is behind a phase shifter. Learning feedback is obtained after comparing the results with the real ones, then it is inputted into the controlled logic component and changes the weight w. In our program, the process in the controlled logic component and NOPO is simplified.

2.6 Thresholder

Thresholder is special mode of amplifier. Its structure is very similar to the amplifier, it is also based on a Mach-Zehnder interferometer (MZI) and there is one MRR on each arm. The difference is that this MZI is asymmetric.



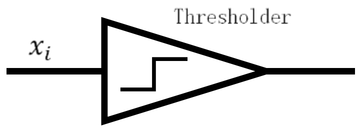


Fig2.6.1

The structure is visualized in Fig2.6.1. The length of two arms of the MZI are different, and the value of r_1, r_2, what is defined as amplitude self-coupling coefficient (i.e., reflectance) of the directional coupler to the MRR in the top or bottom arms, are different. Then, the power coupling ratio A of the coupler is about 0.483, which means that the input power is not averagely divided into the top or bottom arms. The thresholder is inputted by one power in the top port. And according to the above, the transmission of the electric field in two MRRs are not equal because of the difference of the powers, which are inputted into the MRRs. The transmissions of the electric fields of the top or bottom MRRs are defined as t_1 and t_2. Finally, the output electric field is

$$E_{output} = \sqrt[2]{A(1-A)} \cdot E_{input} \cdot (t_1 - e^{i\emptyset_b} t_2) \tag{2.6.1}$$

And the transform of the power in thresholder is

$$P_{output} = A(1-A) \cdot P_{input} \cdot \left| t_1 - e^{i\emptyset_b} t_2 \right|^2 \tag{2.6.2}$$

Where the $\emptyset_b$ is the fixed phase difference between two arms of the MZI. To get maximum destructive interference for small signal inputs, which are below the threshold, the phase difference $\emptyset_b$ is be defined below.

$$\emptyset_b = \arctan\left(\frac{r_2 \sin\emptyset_0}{a - r_2\cos\emptyset_0}\right) + arctan\left(\frac{ar_2 \sin\emptyset_0}{1 - ar_2\cos\emptyset_0}\right) \dots$$
$$-\left[\arctan\left(\frac{r_1 \sin\emptyset_0}{a - r_1\cos\emptyset_0}\right) + arctan\left(\frac{ar_1 \sin\emptyset_0}{1 - ar_1\cos\emptyset_0}\right)\right]. \quad (2.6.3)$$

where $\emptyset_b$ is the single-pass phase offset.

2.7 The structure of one all optical perceptron

The structure of one all optical perceptron is shown in Fig2.7.1 below. The N input data x_i are coherent powers or electric fields. At first these data are multiplied by the coherent weights w_i through an optical programmable amplifier. Then, they are coherently added with the help of an optical power combiner. Finally, the sum is sent to the optical thresholder to generate the estimated outcome t_{result}. In the training phase, the estimated outcome is compared to the expected value $t_{expected}$ to get the value of error e_{error}, which will change the value of the weights w_i in the next step.

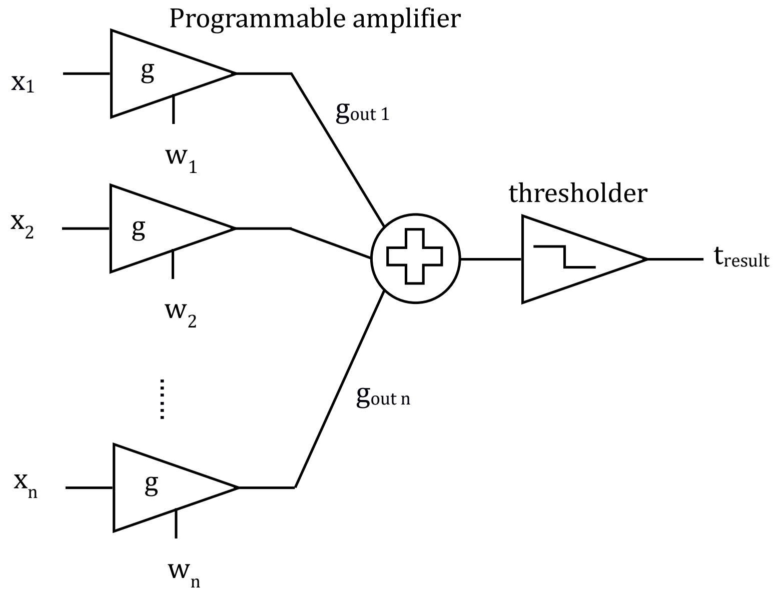


Fig2.7.1

The error is calculated by thresholder's output and expected output of perceptron.

$$e_{error} = t_{expected} - t_{result} \quad (2.7.1)$$

The change of weights w_i are depending on the error, the output of the amplifier and the value of the weights in the last learning process.

$$w_i{}' \ \ (next\ step) = w_i + e_{error,\ i} \cdot g_{out,\ i} \cdot \eta \qquad \eta \in \mathrm{R} \tag{2.7.2}$$

Where η is the learning rate of the perceptron.

In our program the input x_i are discrete, the value of x_i is specified of 0mW or 100mW, which defined in digital as “0” or “1”. The expected output of the thresholder is also discrete with the value of 0mW and 15mW, which represent “0” or “1” of the output. However, the input and output can also be continuous, when the training data are continuous.

3 Results and Conclusions

3.1 Results

In the above introduction, the whole process is described with electric field intensity. In the neuron, we just need the power of outputs. High level power means "1" in the binary system and low-level power means "0" in the binary system. The transformation between both two parameters is

$$P = C \cdot A_{eff} \cdot |E|^2, \tag{3.1.1}$$

where C is a coefficient and A_{eff} is the effective sectional area of the photonic circuit.

According to the equation above, we have written several Python programs to simulate these optical devices. At first is the thresholder, which is at the end of the optical perceptron. Device parameters are A = 0.483, r_1 = 0.970, r_2 = 0.989, $\emptyset_0$ = −0.0481 and $\emptyset_b$ = 0.667. The material parameters respond to an SOI strip waveguide platform. Thus, the refractive index n = 3.45, loss α = 2.4dB/cm, intensity-dependent refractive index n_2 = 7.5*10e-14 cm^2/w, and mode cross section A_{eff} = 1.2*10e-10cm^2, the MRR radius is 4μm and excitation wavelength very near 1.5μm. And the result is visualized in Fig 3.1.1 below.

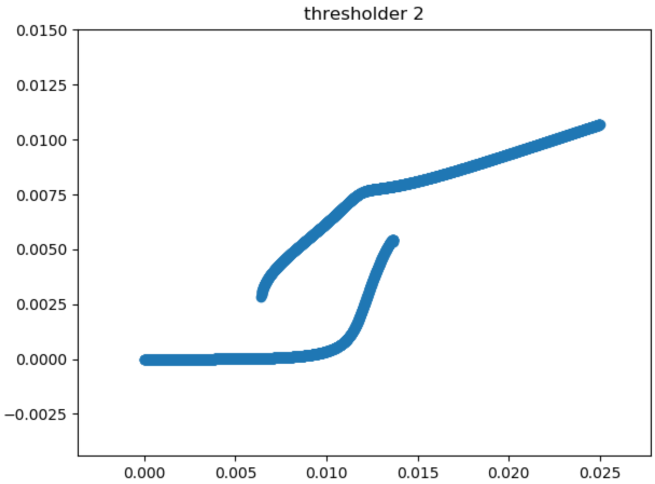


Fig 3.1.1

The curve shows that when the input power is between 6mW to 14mW, a optical bistability exits in the thresholder. And the output power is very near to 0 when the input is less than 6mW, while the high level of the outcome is more than 8mW. The optical bistability is that the output of thresholder is unstable and has the same input-output attributes of the last inputted power, i.e. use the upper or lower curve like the last step. Only when the input is less than 6mW, it will change to use the lower curve, and when the input is more than 10mW, it will change to use the upper curve. Because of the optical bistability, it will bring errors to the following calculations.

Secondly, the design of the programmable amplifier should refer to character of optical thresholder, because the threshold should the range of the output power of the programmable amplifier. Only after that can we get two different levels of the power on the output port. Thus, the parameters are r = 0.926, loss α = 3dB/cm, the radius of the MRR is 6um, and the rest parameters are same as amplifier, i.e. $\emptyset_0$ = −0.0481, intensity-dependent refractive index n_2 = 7.5*10e-14cm^2/w, and mode cross section A_{eff} = 1.2*10e-10cm^2. The amplifier has two inputs x, w, and the character is shown in Fig 3.1.2 below.

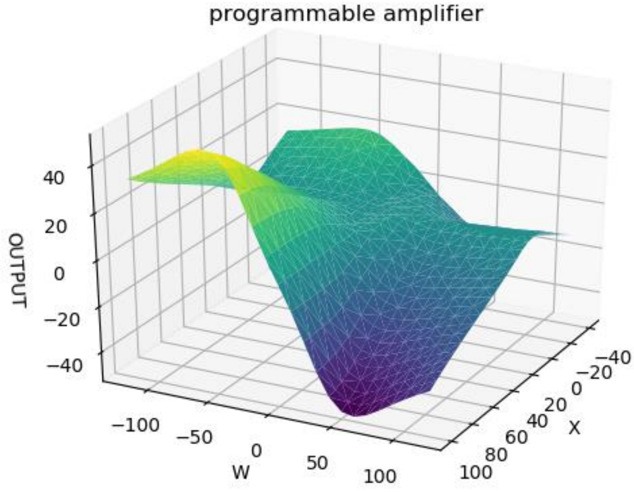


Fig 3.1.2

The output power is proportional to the product of x and w. The range of the output is about 0-40mw, when the input x is between 0-100mW while w is between 0-50mW.

Finally, it is the design of the perceptron. The input x (signal power) is 100mw or 0mw, which represent 1 and 0. The input w (bias power) is between 0mW and 50mW. The expected outcome is much smaller, i.e. 15mW defined as "1" and 0mW defined as "0". Both the inputs and outcome are analog. The outcomes are compared to the expected outcome to change the input w. We use this optical perceptron to model a OR function. The training data is like in Table3.1.3.

Training Data of the OR function

A		**B**		**A OR B**	
(digital)	(analog)	(digital)	(analog)	(digital)	(analog)
0	0mW	0	0mW	0	0mW
0	0mW	1	100mW	1	15mW
1	100mW	0	0mW	1	15mW
1	100mW	1	100mW	1	15mW

Table3.1.3

The variation of the analog error is shown in Fig3.1.4. And the learning rate is 0.2.

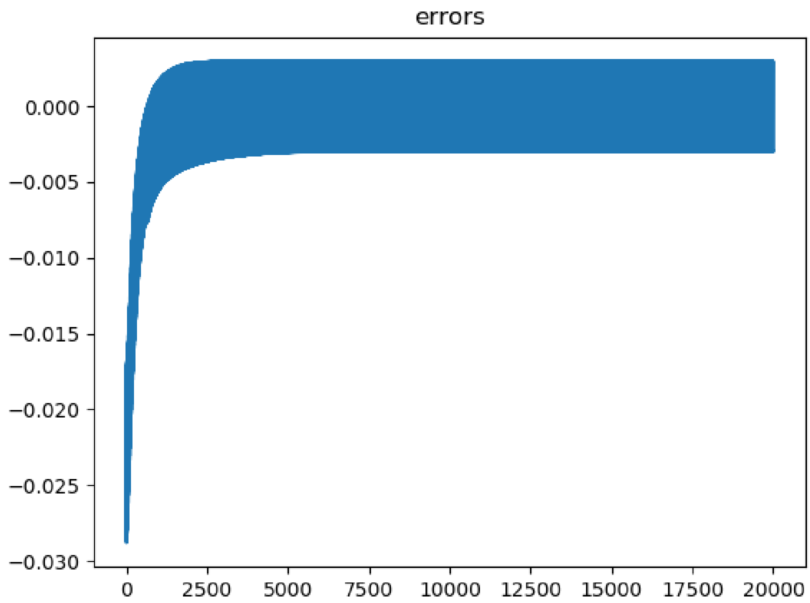


Fig3.1.4

The change of the weights w_i during the training is showed in Fig3.1.5. And the weights are changed differently according to the output errors and the output of the amplifier.

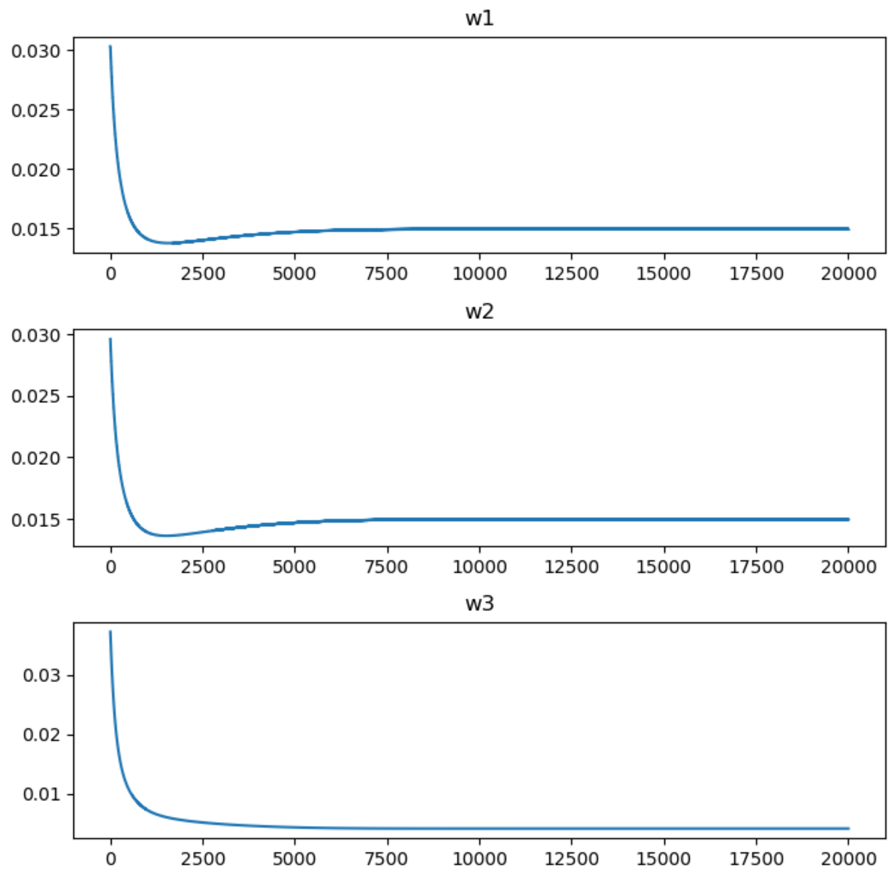


Fig3.1.5

The output of the perceptron in analog and digital is like in Table3.1.6

The input and output of the optical perceptron

Input A		**Input B**		**Output**	
(digital)	(analog)	(digital)	(analog)	(digital)	(analog)
0	0mW	0	0mW	0	2.9mW
0	0mW	1	100mW	1	12.0mW
1	100mW	0	0mW	1	12.0mW
1	100mW	1	100mW	1	17.9mW

Table3.1.6

For the digital output we used an analog-to-digital conversion function (3.1.1).

$$D_{output} = \begin{cases} 0, & P_{ouput} < \frac{1}{3} \cdot 15mW \\ 1, & P_{ouput} > \frac{2}{3} \cdot 15mW \end{cases} \qquad (3.1.2)$$

3.2 Compare to the results with an ideal thresholder

To analyze the performance of the optical perceptron, we designed an almost same perceptron with an ideal thresholder as a contrast. The ideal thresholder is described by the following formula:

$$t_{result} = \begin{cases} 0, & if\ t < 0.006 \\ 0.015, & if\ t \geq 0.006 \end{cases} \qquad (3.2.1)$$

Here 0.006 is threshold value. The parameter t is the input of the thresholder. 0.015 is defined as the high level power, the unit is W. Except of the difference of the thresholders in these two perceptrons, other parts of them are the same. The OR function is also learned by this perceptron. The variation of the error is shown in Fig3.2.1. As we can see, the speed of learning here is 3 times faster as the learning with the simulated optical thresholder.

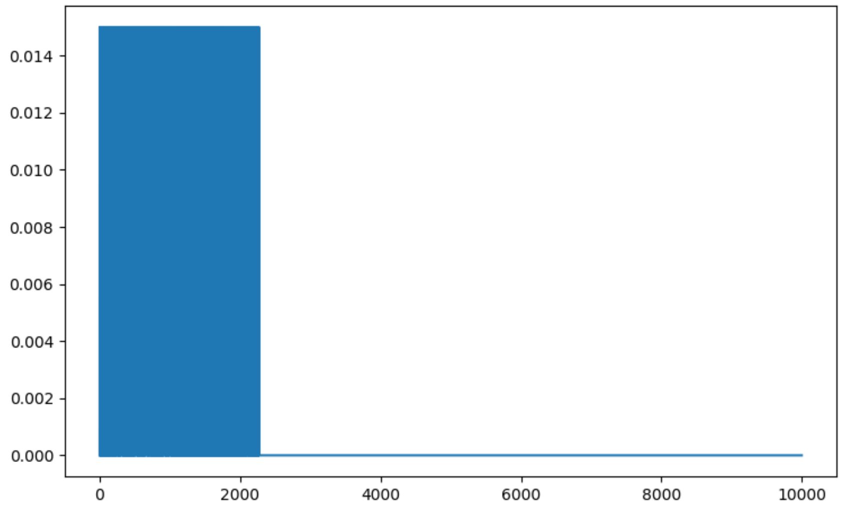


Figure 3.2.1

The implemented perceptron learning with the ideal thresholder has better results. The output can be exact 0 or 15 mW.

Input A		**Input B**		**Output**	
(digital)	(analog)	(digital)	(analog)	(digital)	(analog)
0	0mW	0	0mW	0	0mW
0	0mW	1	100mW	1	15mW
1	100mW	0	0mW	1	15mW
1	100mW	1	100mW	1	15mW

Table3.2.2

The outputs are the same as the training data. After three implements, three groups of weights are following:

[$w_1 = 0.05318839$ $w_2 = 0.08909262$ $w_3 = 0.02424924$]

[$w_1 = 0.04927762$ $w_2 = 0.12899748$ $w_3 = 0.03307816$]

[$w_1 = 0.10755929$ $w_2 = 0.02159535$ $w_3 = 0.05026437$]

Because the weights begin with the random initial values, so results of weights are not the same.

One group of them is showed in Figure 3.2.3:

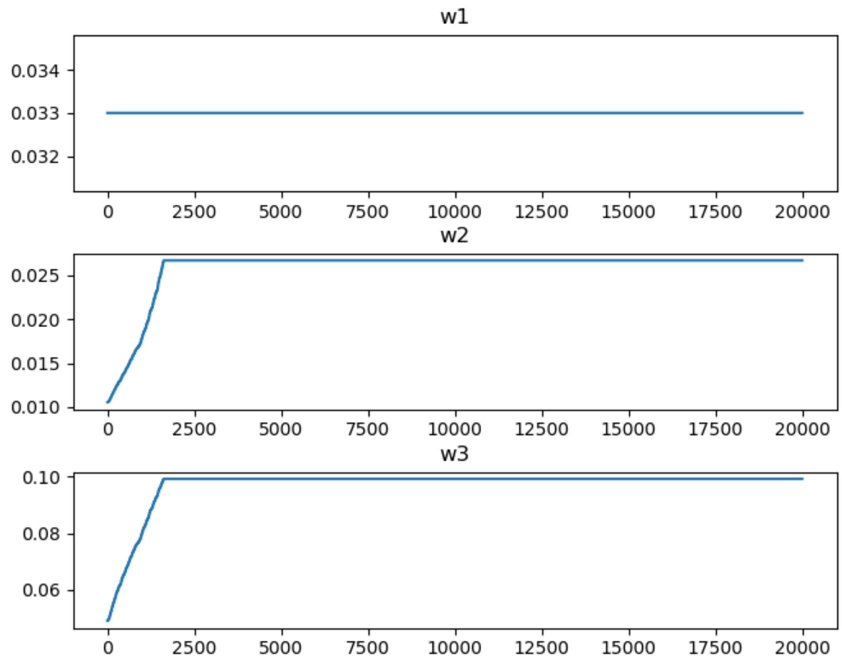


Figure 3.2.3

After three implements with the simulated optical thresholder, three groups of weights are following:

[w_1 =0.01495796886 w_2 =0.01496176252 w_3 =0.00417525552]

[w_1 =0.01495796886 w_2 =0.01496176252 w_3 =0.00417525552]

[w_1 =0.01495796886 w_2 =0.01496176252 w_3 =0.00417525552]

These weights here are similar size to the third group of weights with the ideal thresholder and are invariant because of the same initial weights. It indicates that the results converge in the same way and the process is stable.

3.3 Conclusions

Comparing to literature implementations, we have constructed a perceptron capable of performing a simple logical operation (in our case OR between 2 input binary signals). When the ideal thresholder is applied, we can obtain the better results and error is exact zero. Under the same condition, the learning with the simulated optical thresholder is 3 times slower than it with the ideal, and the error is close to zero, because of the nonlinearity of the optical thresholder. If η is smaller, the error will become smaller but learning times more than before. In a word, they both can realize the OR function, so our simulated model is an available perceptron that can do some learning.

In our project, the important idea is modularizing the optical components. Additionally, it is easy to analyze the whole process and build the model. Through analyzing the results, the simulation is not perfect but meaningful. The framework can be applied in analyzing and simulating future optical circuit. Based on our work, the ANN can be also further built.

Literature

[1] Shen, Yichen, et al. "Deep learning with coherent nanophotonic circuits." Nature Photonics 11.7 (2017): 441.

[2] Tezak, Nikolas, and Hideo Mabuchi. "A coherent perceptron for all-optical learning." EPJ Quantum Technology 2.1 (2015): 10.

[3] Tait, Alexander N., et al. "The DREAM: an integrated photonic thresholder." Journal of Lightwave Technology 31.8 (2013): 1263-1272.

Bogaerts, Wim, et al. "Silicon microring resonators." Laser & Photonics Reviews 6.1 (2012): 47-73.