國立交通大學

光電工程研究所

碩士論文

類神經網路在光學量測方面之應用

Application of Artificial Neuron Network in Optical Metrology: Some Case Studies

研 究 生：陳立志

指導教授：黃中垚　教授

中華民國九十九年一月

類神經網路在光學量測方面之應用

Some Case Studies of the Application of Artificial Neuron Network in Optical Metrology

研究生：陳立志　　Student：Li-Chih Chen

指導教授：黃中垚　　Advisor：Jung Y. Huang

國立交通大學

光電工程研究所

碩士論文

A Thesis

Submitted to Institute of Electro-Optical Engineering

College of Electronic Engineering and Computer Science

National Chiao Tung University

In Partial Fulfillment of the Requirements

for the Degree of Master

in Electro-Optical Engineering

January, 2010

Hsinchu, Taiwan, the Republic of China

中華民國九十九年一月

類神經網路在光學量測方面之應用

研究生：陳立志　　　　　指導教授：黃中垚

國立交通大學光電工程研究所

摘 要

本論文提出兩種新的類神經網路應用於雷射脈衝特性診斷, 以及點光源的定位。利用類神經網路學習以及函數逼近的功能, 提出一個方式由二倍頻光譜去反推出原光場在頻域上的相位, 並且提出可能實現此量測方式的實驗架構。另一方面由類神經網路學習, 從一個包含點光源的圖像中告知點光源所在的位置, 並以此方式結合搜索做到多數點光源的定位。

Some Case Studies of the Application of Artificial Neuron Network in Optical Metrology

Student：Li-Chih Chen Advisor：Jung Y. Huang

Institute of Electro-Optical Engineering, National Chiao Tung University

Abstract

In this thesis we report two new application cases for the Artificial Neuron Network they are focus on the laser plus characterization and the locating of light spot. By using the learning ability and the function approximation, we report a method to retrace the phase in the frequency domain of the origin light field by the second harmonic generate spectrum, and also provide a possible experimental setup for realizing this measurement. On the other hand, by training Artificial Neuron Network to learn the position of a light spot within an image and combining the search method, the location of multiple light spot can be realized.

誌謝

多年來的求學生涯在此終於告一段落, 感謝黃中垚老師給我的指導與鼓勵，提供良好的學習環境。

其次非常感謝我的父母及家人, 因為他們的支持才能無後顧之憂完成學業。

再者非常感謝實驗室的同學, 學長姐, 學弟妹的幫忙, 大家的參與讓我擁有這段多采多姿的日子。

CONTENTS

LIST OF FIGURES

Chapter 1 General Introduction

1.1 Fundamental Principle of Optical Metrology

In the field of optical metrology, two fundamental issues encountered are: how to characterize the light to be used, and how to measure the optical signal to extract the information of the measurement. As shown in Fig. 1.1, by analyzing the difference between the intrinsic properties of the light source used and the optical signal from the object under test, the information of the object can be revealed. By analyzing the data we can further construct a model to explain the behavior of the object. After this model is refined with improved knowledge from the data analysis, we can use to model to predict the object behaviors which are still unknown to researchers. In a more aggressive attitude, researchers or engineers can invoke the knowledge accumulated to control or produce the desired response of a physical system by applying a proper waveform of stimulant on the system.

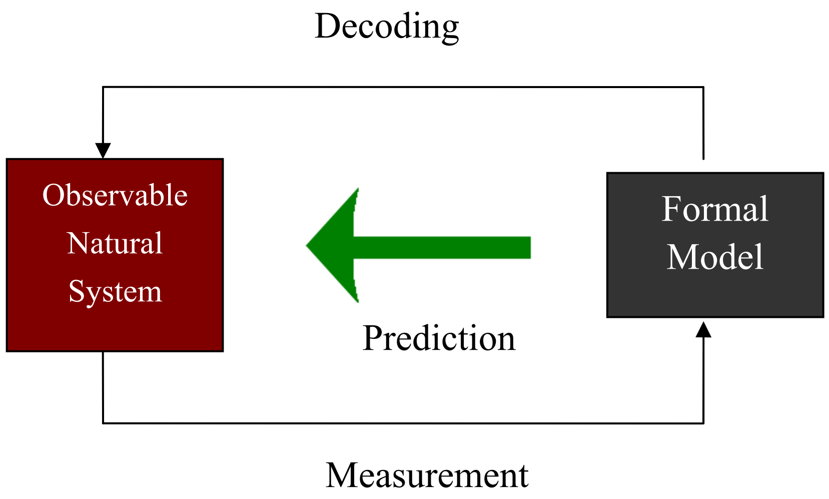
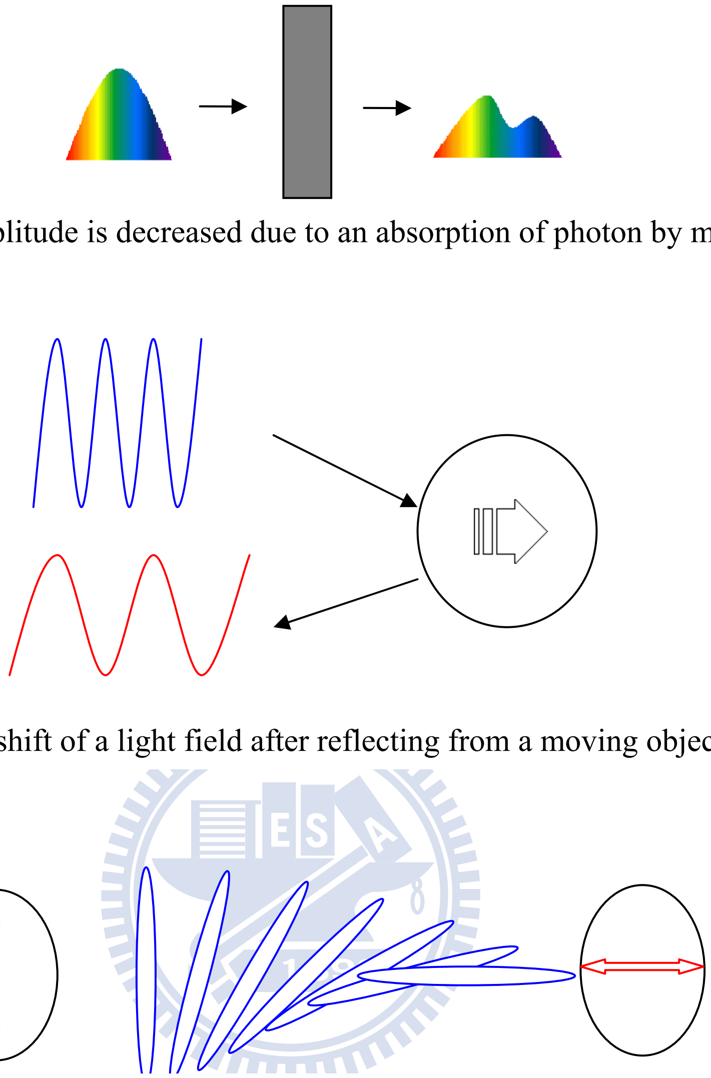


Fig. 1.1. The schematic showing the relationship of measurement, model construction, and the control of a physical system.

The fundamental properties of a light field including the amplitude, phase (revealing via frequency or wavelength), and the state of polarization can be invoked in optical metrology. For examples, as illustrated in Figure 1.2, the amplitude of an optical field can be modified by a material and the information can be used to reveal how much light energy is absorbed as the light beam passes through the medium, which relates to the band structure of material. Furthermore, by detecting the frequency shift of a light field as it reflects from a moving object, we can deduce the information about the velocity of the object, which is known as the Doppler Effect. The direction and the state of polarization of a light field after passing through a transparent medium such as a thin film of liquid crystal will be changed and the polarization variation carries the information about the orientational profile of the liquid crystal molecules in the film.



Light amplitude is decreased due to an absorption of photon by matter.

Frequency shift of a light field after reflecting from a moving object

Polarization direction of a light field is changed by a liquid crystal film

Fig. 1.2. Examples showing the information of physical properties of a medium can be embedded in the three fundamental characteristics of a light field.

Among a variety of the light-matter interaction processes such as optical absorption by crystalline silicon, the absorption of an incident photon with energy larger than the band gap of the medium can readily create an electron-hole pair and results in photocurrent. As the strength of the optical field increases, numerous nonlinear optical effects can be detected and be employed to form a variety of measurement techniques. If one can tightly focus the light beam on a medium with a field-concentrating tip, the signal to be detected will mainly originate from the illuminated area under the tip. By using such a tip, we can achieve an extremely high spatial resolution beyond the optical diffraction limit. On the other hand as shown in Fig. 1.3, if one can employ optical pulses with duration down to a few femto-seconds to probe a medium, the deep insight into the ultrafast dynamics of the excited medium may be yielded. When optical pulses with ultra short duration are used for probing materials, dynamic studies of materials, such as the dynamic processes of a chemical reaction, photosynthesis, interaction between proteins and substrates, photo-excited electron-hole relaxation, shall become possible. However, to realize the potential of optical metrology at the femtosecond scales, techniques that can be employed fully characterize the ultrafast light field must be developed.

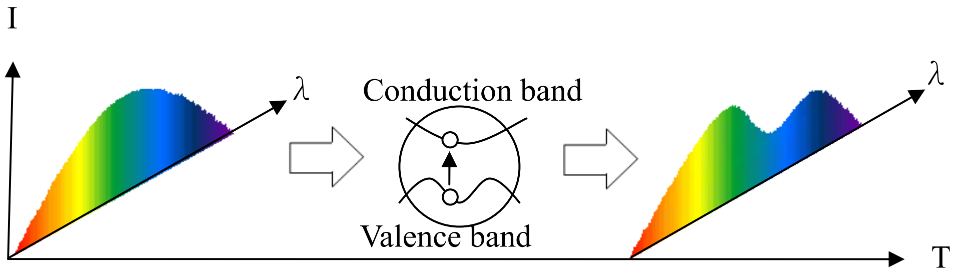


Fig. 1.3. Schematic showing an optical absorption process of broadband light by a medium. The information about the band structure of the medium and the dynamics of light-matter interaction can be embedded into the dynamic spectrum of the light after passing through the medium.

A proper characterization of the light field to be used in ultrafast optical metrology is not a simple task. In the past few decades, several methods had been developed in order to solve the issue. Optical intensity autocorrelation (AC) [1] is the first technique to deduce the temporal profile of a laser pulse. Although the autocorrelation can yield pulse information in time domain, the profile deduced is not the real field profile of the optical pulse. Furthermore, the biggest disadvantage of the technique is that it carries no information about the phase of the optical pulse. To fully characterize an optical pulse field, several two-dimensional methodology such as frequency-resolved optical gating (FROG) [2] and spectral phase interferometry for direct electric-field reconstruction (SPIDER) [3] had been developed in the past two decades. The schematic setups of FROG and SPIDER are illustrated in Figs. 1.4.

FROG is similar to the autocorrelation except that it detects the transient spectra of an optical pulse instead of intensity. We use FROG to acquire the spectra with different time delays and assemble a time-frequency distribution of the pulse. We then retrieve the phase information from the time-frequency distribution by using an iteration algorithm. SPIDER is based on the concept of spectral-shearing interferometer; the optical pulse to be measured is split into two parts, with the time delay and phase delay to be adjusted separately. And then the two parts of the pulse can be recombined to generate a set of fringe patterns. The major advantage of SPIDER is that the set of fringe patterns can be used to retrieve the spectral phase of the pulse field in a direct way without invoking any iteration procedure. Therefore, the information retrieving speed from data of SPIDER can be very fast.

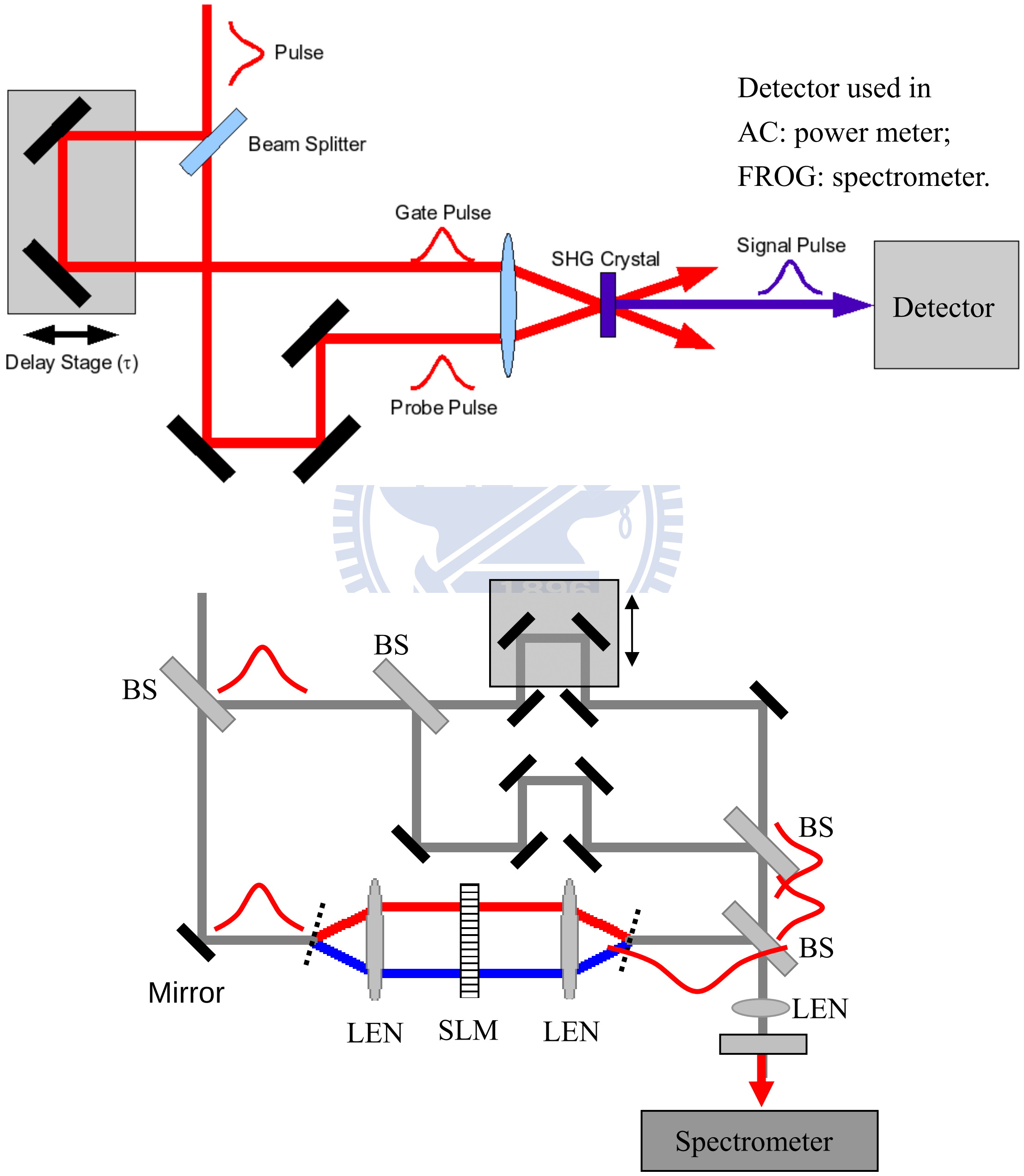


Fig. 1.4. Typical experimental setup for frequency resolved optical gating (FROG, top) and spectral phase interferometry for direct electric-field reconstruction (SPIDER, bottom).

To characterize the structure of a heterogeneous material such as the organization and distribution of molecules or subcellular objects in a cell raises the need to precisely localize these objects. The position and velocity of a nano object can be determined by localizing the corresponding light spots in a 2D photo-detector in an optical microscope. By tracing the particle in real time, important mechanisms in a live cell had been revealed. The light spots can be originated from either light scattering or fluorescent emission from nanoparticles. The technique that can accurately localize nano particles is also an important tool in the material structural determination, especially for nanostructured materials.

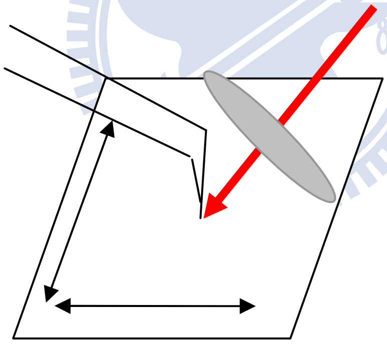


Fig. 1.5. Schematic shows the concept of concentrating the light field on a medium with a tip. The signal to be detected will mainly originate from the illuminated area under the tip, yielding the possibility to localize particles with high spatial resolution.

To localize nanoparticles with the nanometer accuracy, we first have to excite the particles at very low level such that two light spots from neighboring excited particles are separated on CCD camera at least by one full width of the point spread function of the optical microscope used. The central positions of the light spots are then fitted to 2D Gaussian profiles [4] or with the center-of-mass method. The fitting of a light spot to a 2D Gaussian function in principle can determine the sub-pixel center of the light spot. Unfortunately, the localization accuracy with the method is time consuming and sensitive to the signal-to-noise ratio of the data.

For the center-of-mass method, the center-of-mass of a light spot can be calculated as the weighted position of all pixels involved in the spot. For the case shown in Fig. 1.6, the mean value can be (1*1+2*2+3*2.5+4*1)/(1+2+2.5+1)=2.5385.

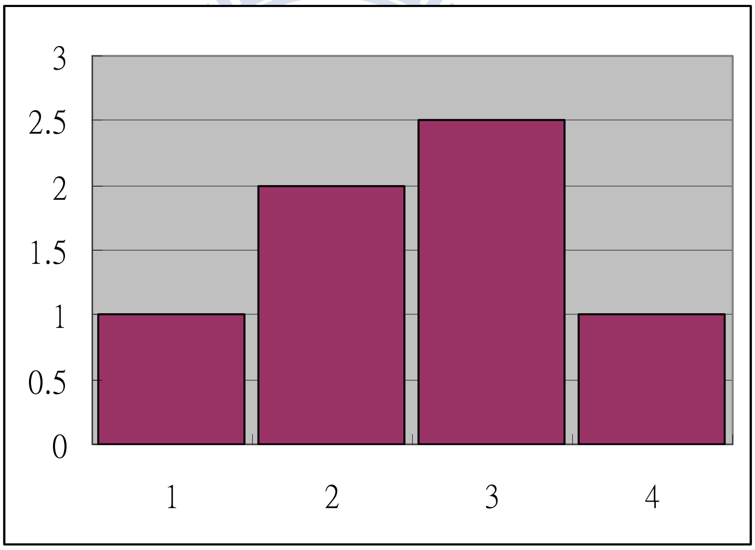


Fig. 1.6. A 1D example of the center-of-mass method

1.2 Overview of Artificial Neural Network

An Artificial Neural Network (ANN) [5] is an algorithm designed to simulate the capabilities of learning and data processing of the neuron network in our brain. ANN simulates our brain on two aspects; firstly, the knowledge learning from the data can be updated and stored in the weighting parameters of ANN. Second, the set of optimal weighting parameters can be deduced through a learning process. The schematic of the overall process is depicted in Fig. 1.7. The input data is first converted into an output, and then the calculated output is compared with the desired output to generate an error, which can be feedback to adjust the weighting parameters of ANN in order to further reduce the error. By this way, the output can approach the desired performance.

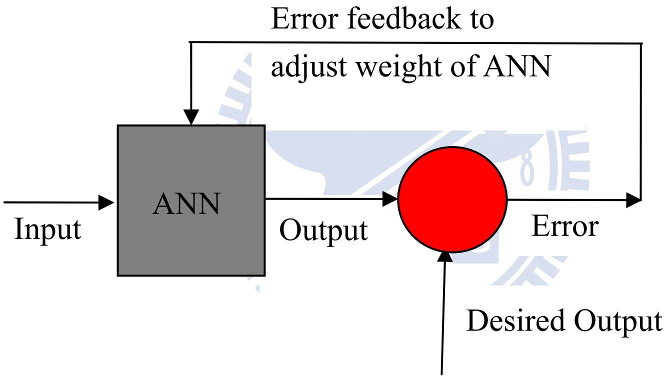


Fig. 1.7. Flow chart of the learning process of an ANN.

ANN can be invoked to offer several useful functionalities, including data classification, functional approximation, and series prediction, *etc*. [6-8]. Usually, one can start with data analysis, and then based on the results to construct a preliminary model. After testing and verification involved in the learning process, the trained ANN can be used to predict the future behaviors of the system under study. Especially, when the data is complex and cannot yield sufficient information to expose the underlying structures, ANN could be very valuable in this case.

1.3 Motivation and Outline of this Thesis

In this thesis, we will apply ANN for producing an intelligent learning system to improve the measurement accuracy or generating new functionality of an apparatus in optical metrology, namely the complete characterization of ultrashort laser pulses and nanometer localization in real time for an optical microscope. The techniques needed to implement artificial neuron network for the two applications will be developed.

We also like to implement the learning ability of artificial neuron network into an optical apparatus to accumulate the user experiences and improving the prediction accuracy of the ANN as more data are taken. To achieve this goal, this thesis is organized as follows:

In chapter 2, we will first review the general concepts of ANN and introduce the skills needed for the applications. We will introduce the most useful functionalities of ANN, some illustrative application examples are prepared too.

In chapter 3, we will combine ANN with a second-harmonic generation spectroscopy to retrieve the spectral phase profile of ultrashort optical pulse field and yield the complete field information of the pulse. Feasibility of this application is demonstrated and difficulties encountered are discussed.

In chapter 4, we will develop an ANN scheme to search and locate the position of particles with an optical microscope. The feasibility of artificial neuron network to find the central position of a light spot was revealed. Higher accuracy, faster localization process and more immune to the noise than that does by the 2D Gaussian fitting were demonstrated.

Finally, conclusions and future works of this thesis research are presented.

Chapter 2 Introduction to Artificial Neural Network

In Section 2.1, we review the development history and some background of artificial neural network. We will describe how does the artificial neural network work in Sections 2.2 to 2.4 and then present application examples to illustrate the important functionalities of ANN, including classification, functional approximation and time series prediction. A brief summary will be given in Section 2.5.

2.1 Overview of Artificial Neural Network

2.1.1 Historical Development of ANN

Inspired by the structure of neuron and the connection topology in brain, the first mathematical model of neural network was reported in 1943 by Warren McCulloch and Walter Pitts [9]. In 1958, Frank Rosenblatt developed the first practical network with related learning rules, which is now named as the artificial neural network structure of perceptron [10]. From 1967-1982, Marvin Minsky and Seymour Papert discovered some limitations of existing neural networks [11], such as that ANN cannot perform some complex logical operations for example XOR, and new learning algorithms were not put forward causing researches to suspect the limited functionality of ANN. However, the development stagnation of ANN finally broke in 1980s. Several new architectures of

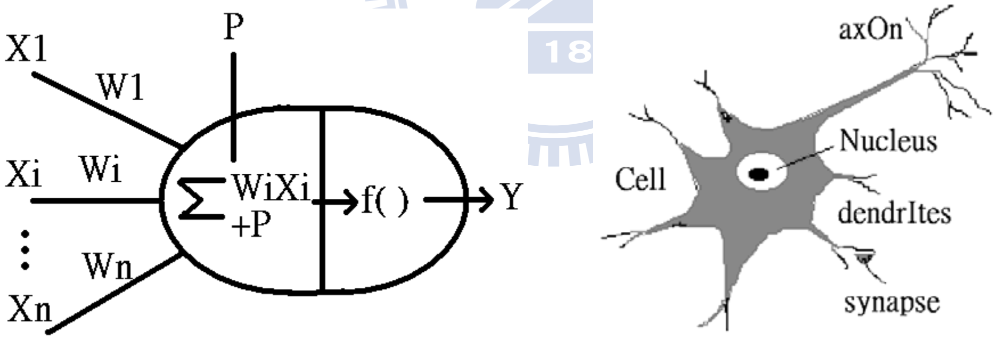
ANN and theorems had been discovered, including the self-organizing map structure by Teuvo Kohonen in 1982 [12], the Hopfield neural network by John J. Hopfield in 1983 [13], and David Rumelhart's group reported back-propagation neuronal network in 1986 [14]. Furthermore, in 1991 Stephen Grossberg developed the adaptive resonance theory [15] that can significantly improve the learning capability of an artificial neural network. Thanks to these progresses, artificial neural network has been widely used in the pattern recognition, identification, and data classification.

Hossein reported in 1989 [16] that if the weights of a backward-propagating ANN are initially set to be high values, the learning performance of the ANN will be improved and the best initial values depend on the problem. Kruschke [17] inserted a gain into a backward propagation network and found a regularization effect on the weights, which improves the performance of backward ANN. Sarker [18] found by tuning the weights the oscillation occurring in ANN can be reduced. Kubat [19] used a determination tree to guiding the construction of a backward-propagating ANN. Wu [20] developed a method of optimizing the hidden layers' outputs (OHLO) to isolate each neural layer and then to modify the weights and inputs. Leung [21] combined the weight evolution method with a new generalized back-propagation method to accelerate the convergence and avoid the problem of trapping into a local minimum.

2.1.2 Technical Background of ANN

Artificial neural network is an algorithm that simulates the function of brain. In order to make an artificial neural network function properly, the algorithm must be divided into two major steps: the first step is called a learning phase and the second step is a retrieving phase. In the learning phase, we need to adjust the system parameters of an artificial neural network such as weights or bias. In the retrieving phase, we can use the artificial neural network to predict the result based on new input data.

The neuron in an artificial neural network functions similarly as a neuron in a brain. The schematic showing the analog can be found in Figure 2.1.



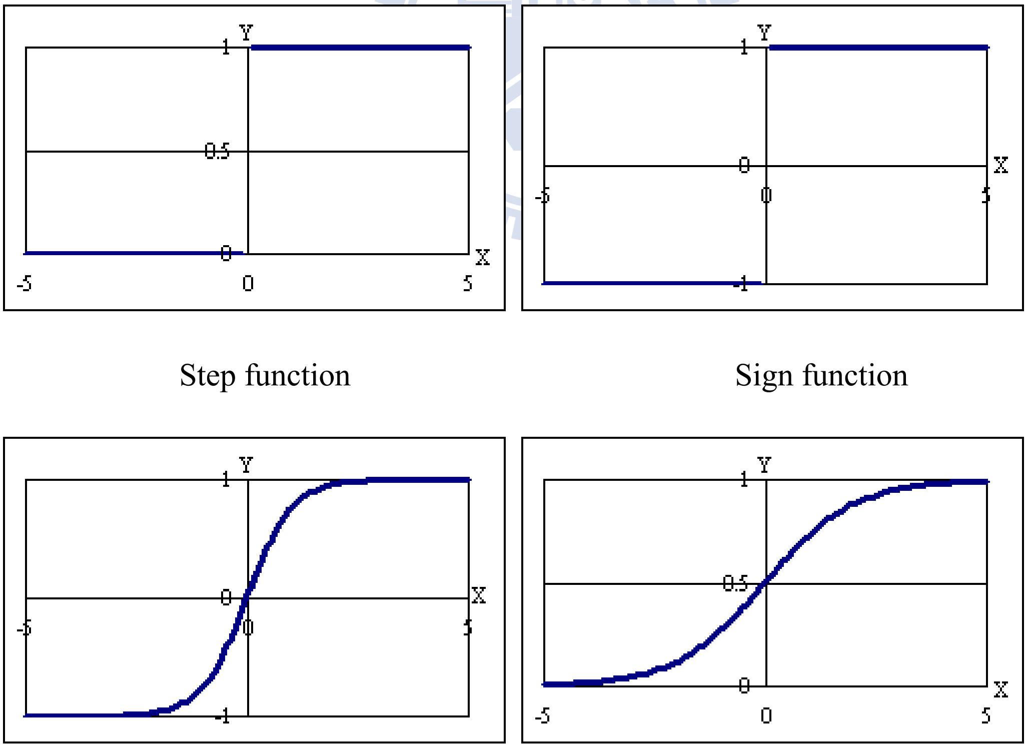
Xi is the input, i=1… n.
Wi is the weights, i=1… n.
P is the bias.
f ( ) is the active function.
Y is the output

A typical structure of neuron
Dendrites: input from other neuron
Axon: output connect to other neuron
Synapse: the gap between the neurons
Cell, Nucleus: body of neuron

Fig. 2.1. Schematic shows the analog of artificial neuron and brain neuron structure.

As shown in Fig. 2.1, the {Xi}, i=1,…,n denotes an excitation from other neuron, and {Wi}, i=1,…,n represents a set of weighting coefficients denoting the connection strength between neurons, P denotes a threshold level beyond which a neuron will fire. And Y is the excitation to other neurons. Depending on the performance desired, a variety of active functions have been adopted in different artificial neural networks. Several useful forms of active function are step function, sign function, hyperbolic tangent function, or sigmoid function, which is defined as

$$Sigmoid = \frac{1}{1+e^{-x+b}}, \quad \text{Hyper tangent} = \tanh(x-b). \quad \text{b: the intercept on x axis.} \qquad (2\text{-}1)$$



Step function

Sign function

Hyper tangent

Sigmoid

Fig. 2.2. Some useful forms of activation function for ANN

2.2 Typical Structures of ANN and the Corresponding Application Examples

2.2.1 The Perceptron Architecture

Perceptron is the first practical ANN, typically used for data classification. Figure 2.3 shows the structure of a perceptron with a structure of input layer and output layer.

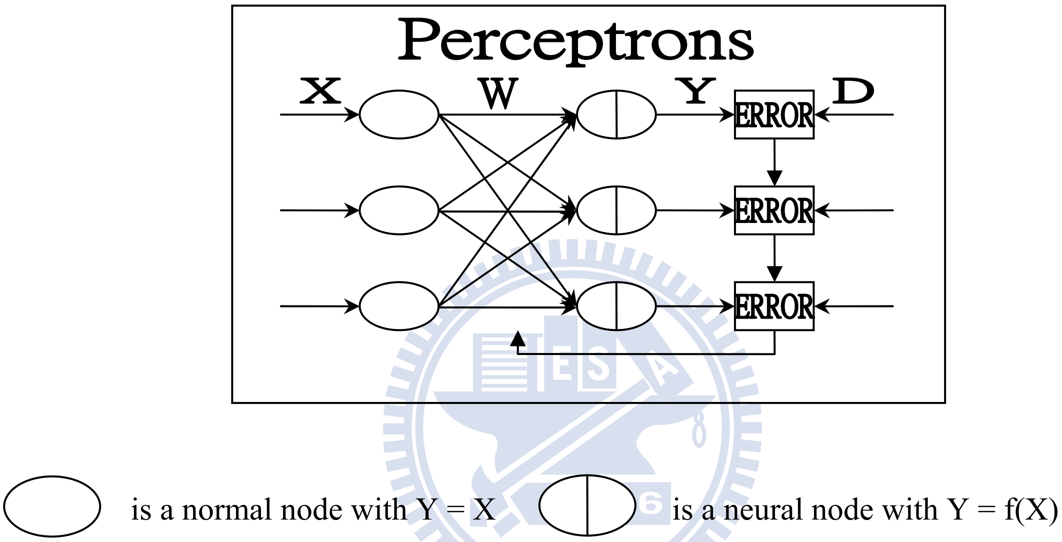

Perceptrons
X
W
Y
D
ERROR
ERROR
ERROR
◯ is a normal node with Y = X
⦶ is a neural node with Y = f(X)

Fig. 2.3. A typical perceptron structure

A perceptron ANN usually uses either a step function or a sign function as its active function depending on the training data set used. Denoting X_i to be the input of the *i*-th input node, W_{ij} the weighting coefficient the *i*-th input node and the *j*-th neural node, then the input of the *j*-th neural node can be expressed as $Y_j = \sum_{i=1}^{N} X_i W_{ij} + P_j$. Here N is the total number of input nodes, P_j is the bias of the neural node, and Y_j is the output of the *j*-th neural node. Let D_j be the desired output at the *j*-th node, E_j is the difference between Y_j and D_j. We can use E to adjust the ANN system parameters W_{ij} and P_j by following the learning rule shown below:

$$W_{ij}(t+1) = W_{ij}(t) + \eta E_j X_i \tag{2-2}$$

$$P_j(t+1) = P_j(t) + \eta E_j, \tag{2-3}$$

where η denotes the learning rate with a typical magnitude ranging between 0 and 1, $E_j = D_j - Y_j$, and t is the iterations.

The training process can be set up by dividing the process into the following steps:

1. First, random numbers between zero and one are used to initialize the weighting coefficients and biases.
2. An input data pattern is inserted at the input, and the output and related errors E are calculated.
3. Using the errors and the learning rules to update the weighting coefficients and biases.
4. Repeat the steps 2 and 3 until the desired result is achieved with a satisfactorily small error.

2.2.2 Application Example of Perceptron

In this section, we will apply a perceptron for data classification to illustrate how this network works. The application example is to classify the 26 English alphabets. The

26 English alphabets were presented in a figure format of 12x12 pixels figure at the input of the perceptron. Figure 2.4 presents some of the letter figures. The output of the ANN is shown by an array of twenty six digits such that the output of alphabet A is (100000…0), B is (010000…0), C is (001000…0)…*etc*.

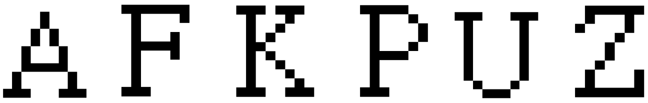


Fig. 2.4. Some English alphabet figures used in this example.

We converted each alphabet figures (from first to the last pixel) into an integer array of 144 elements and used the array for the input data. If the pixel is black, the corresponding integer is one. Otherwise it is zero. We therefore have 26 input data sets with each data being one dimensional array of 144 components of ones or zeros. For the output, we use a 26x26 identity matrix with each row corresponding to the desired output, resulting in a perceptron of 144 input neurons and 26 output neurons. The learning rate is set to be 0.5. The weight coefficients are set to be 0.25 by adjusting the bias. As the result, the learning process can converge with a learning curve shown in Fig. 2.5, indicating that the mean square error (MSE) can approach zero. As shown in Fig. 2.6, the ANN can successfully recognize the alphabet letters with an extremely low failure, revealing with a diagonal form of confusion matrix.

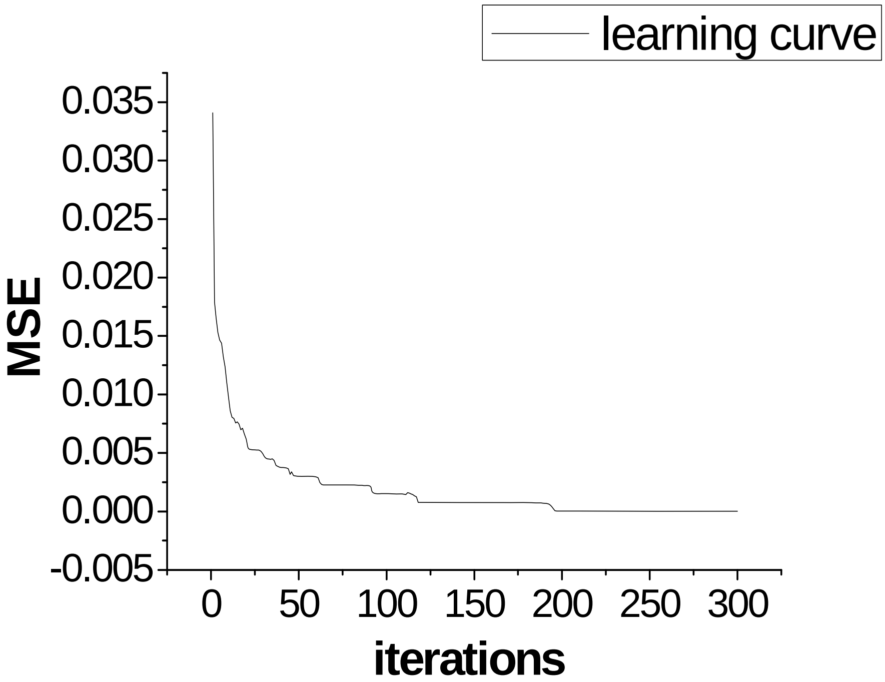
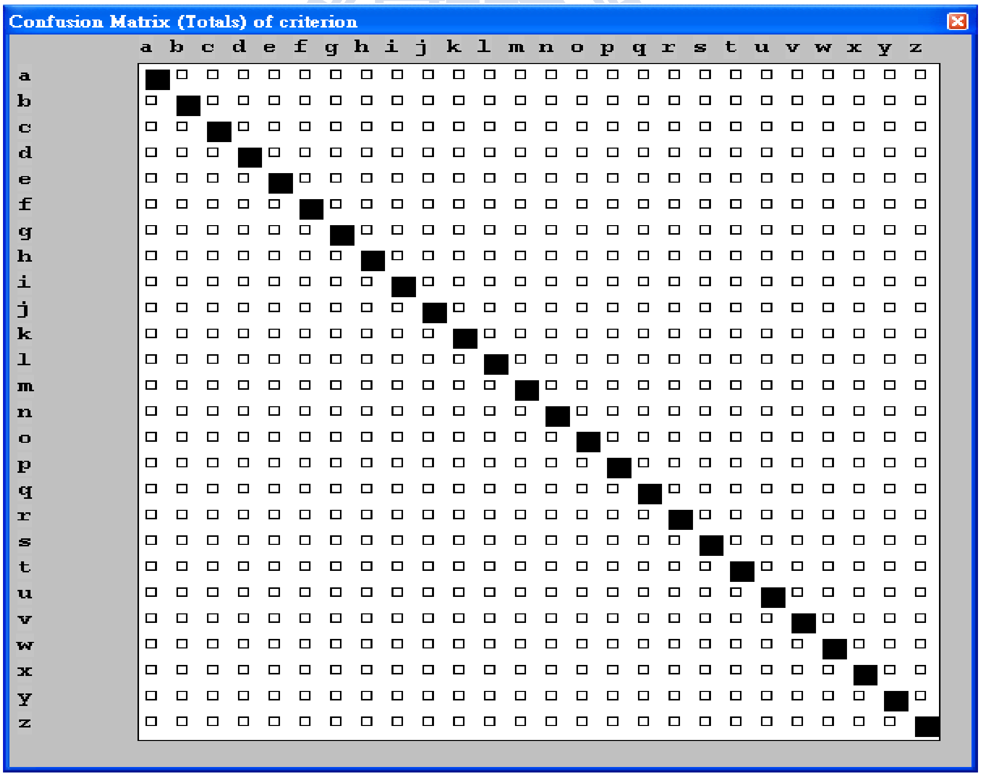


Fig. 2.5. The learning curve of the perceptron ANN for the 26 alphabet letters


Confusion Matrix (Totals) of criterion

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
b	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
c	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
d	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
e	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
f	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
g	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
h	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
i	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
j	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
k	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
l	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□	□
m	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□	□
n	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□	□
o	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□	□
p	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□	□
q	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□	□
r	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□	□
s	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□	□
t	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□	□
u	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□	□
v	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□	□
w	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□	□
x	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□
y	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■	□
z	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	■

Fig. 2.6. The Confusion matrix (X: Output Y: Desired Output)

The learning curve is different for each training cycle of the alphabet letters because the initial weighting coefficients are generated by random number generator and they are of course different for each training cycle. Nevertheless, the final results with different initial set of weighting coefficients are similar.

The weighting coefficients and biases can be represented by matrices with the dimensions of 144x26 and 26x1, respectively. Fig. 2.7 presents the original matrix form of weighting coefficients. Fig. 2.8 shows the reshaped form of the weighting matrix with dimension of 12x312 with the weighting coefficients at the same pixel position in each alphabet letter figures. It was found that the weighting coefficients have large magnitudes at the positions corresponding to the positions of the black pixels in the alphabet letter figures. The weighting coefficients with larger magnitudes reveal those important pixels in the alphabet letter figures. The ANN after training may invoke those pixels to recognize the characteristic features of the alphabet letters. We found that those important pixels are clustered in the central region.

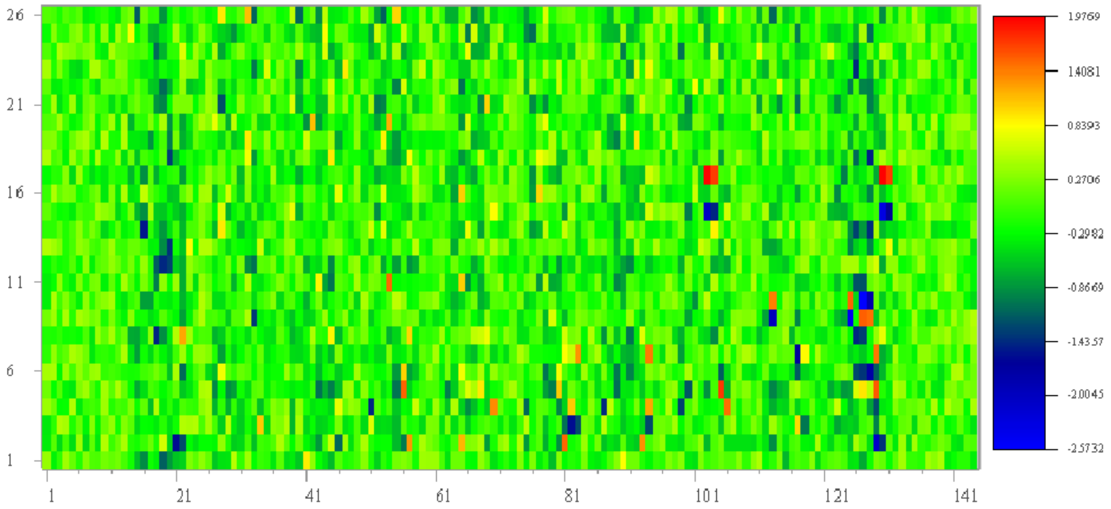


Fig. 2.7. The matrix of weighting coefficients (in its origin form).

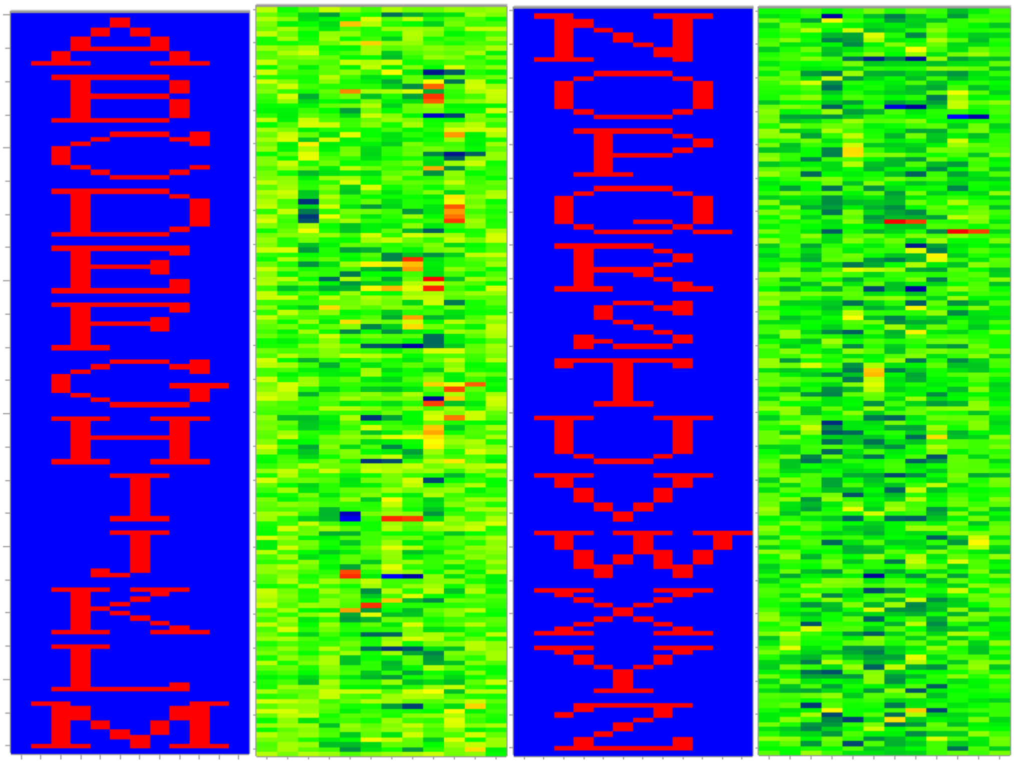


Fig. 2.8. English alphabets and the corresponding reshaped weighting matrices

For example, if an ANN tries to distinguish the alphabets O from X. It may firstly examine the pixels locating at the central region of the letters to find some important indicators, leading to the large weighting coefficients in the central region. But this may not be the only way to distinguish. There may have several possible forms of weighting matrix to meet our goal. The weighting matrix, which yields a power of recognition, is a solution to an equation that connects the desire output vector to the multiplication of the input vector and the weighting matrix. Since there are 26 alphabet letters, we have a set of linear equations with 144x26 variables and 26x26 constraints. It is not surprising that many solutions may exist. In the case that the equations do not have a solution, the training process would fail and no satisfied weighting matrix could be produced. Although a perceptron structure is suited for data classification and recognition, it cannot be used for function approximation. Researchers in the past had developed another type of ANN structure, called back-propagation neuron network to expand the applicability of ANN.

2.2.3 Back-Propagation Artificial Neuron Network

Back-propagation artificial neuron network is useful in many applications of ANN. The typical structure of a back-propagation ANN is illustrated in Figure 2.9, which has a multilayer configuration. Here *W* is the matrix of weighting coefficients of the ANN. For using an input data *X*, the ANN can generate an output *Y* with the desired output *D*. Back-propagation ANN typically possesses a structure of three to four layers, including one input layer, one output layer and one or two hidden layers. The active function implemented in back-propagation ANN is sigmoid function in both hidden layers and output layer to enhance the performance in the learning phase with nonlinear data structure.

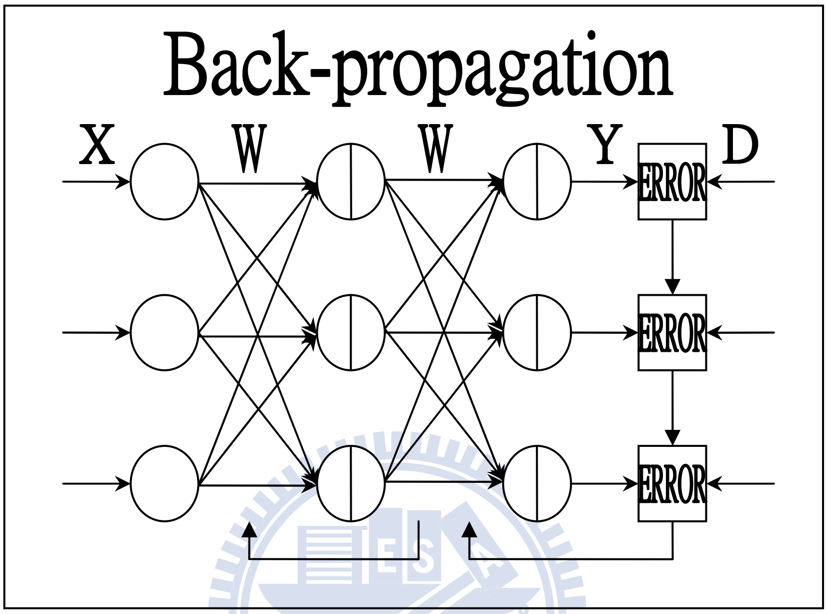


Fig. 2.9. Typical configuration of a back-propagation artificial neuron network

The learning procedure of a back-propagation ANN is summarized as follows: Firstly, the input data and the corresponding desired output are sent to the ANN. The input data propagates forward layer by layer from the input layer, hidden layers to the output layer. The resulting error, which is defined to be the difference between the output and the desired response, is calculated. The weighting coefficients and biases of the ANN are then adjusted in order to minimize the error. Because the ANN has a multi-layer structure, we need to invoke chain rule to calculate the gradient of the system parameters used in the error function.

The learning rule of back-propagation ANN is described as follows. We denote W_{jk} to be the weighting coefficient between the *j*-th hidden neuron and the *k*-th output neuron. The bias of the *j*-th hidden neuron and the *k*-th output neuron is denoted as P_{j} and P_{k}, respectively. Y_{k} is the output from the *k*-th output neuron with D_{k} the desired output, and H_{j} is the output from the *j*-th hidden neuron. Therefore, Y_{k} and H_{j} can be given by

$$H_j = F(\sum_{i=1}^{n} X_i W_{ij} + P_j) = F(net_j) \qquad (2\text{-}4)$$

$$Y_k = F(\sum_{j=1}^{m} H_j W_{jk} + P_k) = F(net_k) \quad . \qquad (2\text{-}5)$$

Here *F* is the active function. The sum of the variables in the active function is defined as a new variable “net”, and *n* and *m* are the number of input neuron and the number of hidden neuron, respectively. Error function will be defined as

$$E = \frac{1}{2}\sum_{k=1}^{l}(D_k - Y_k)^2 \, , \qquad (2\text{-}6)$$

where *l* is the number of output neuron.

We used the gradient descent algorithm to minimize the error function, leading to the following equations to be used for adjusting the weighting coefficients and bias parameters.

$$W_{jk}(t+1) = W_{jk}(t) - \eta \frac{\partial E}{\partial W_{jk}} \, , \qquad (2\text{-}7)$$

$$P_k(t+1) = P_k(t) - \eta \frac{\partial E}{\partial P_k} \, , \qquad (2\text{-}8)$$

where *t* is the iterations and η is the rate constant of learning process.

By using the chain rule, we can calculate the partial derivatives in Eqs. 2.7 and 2.8, which lead to (detail in appendix I)

$$W_{jk}(t+1) = W_{jk}(t) + \eta \delta_k H_j \,, \tag{2-9}$$

$$P_k(t+1) = P_k(t) + \eta \delta_k \,, \tag{2-10}$$

where $\delta_k = (D_k - Y_k) F'(net_k)$, let δ_k be the error of the *k*-th output neuron and we can use δ_k to adjust the bias at the nodes in the hidden layer and the weighting coefficients between input layer and hidden layer by

$$W_{ij}(t+1) = W_{ij}(t) + \eta \delta_j X_i \,, \tag{2-11}$$

$$P_j(t+1) = P_j(t) + \eta \delta_j \,, \tag{2-12}$$

where $\delta j = F'(net_j) \sum_{k=1}^{l} \delta_k W_{jk}$. Because the active function used is a sigmoid function, we can simplify $F'(net_j)$ to a multiplication of sigmoid functions as shown below

$$\frac{d}{dx} F(x) = \frac{d}{dx} \frac{1}{1+e^{-x}} = (\frac{1}{1+e^{-x}})(1 - \frac{1}{1+e^{-x}}) = F(x)(1 - F(x)) \,. \tag{2-13}$$

Eqs. 2-10 and 2-11 form the basis of learning needed to train the ANN.

2.2.4 Application of Back-Propagation Artificial Neuron Network for Function Approximation

The function approximation is the most useful feature of ANN. This functionality of ANN can be use to approximate the relationship between input and output of a physical system. Here we consider a simple application example of back-propagation ANN to approximate the relationship between the critical angles of the interface between two media, which is known as the Snell's law. The critical angle of light passing through an interface of two media is known to be

$$\theta = \sin^{-1}(n_1 / n_2), \tag{2-13}$$

where n_1, n_2 are the refractive indices of the two medium. For this case, we construct a back-propagation ANN with two input nodes, four hidden nodes and one output node. The learning rate is set to be 0.1 for all weighting and bias parameters. As shown in Fig. 2.10, the convergence of the learning process is poor if the number of hidden nodes is less than three, whereas no further improvement in the convergence can be yielded when the number of hidden nodes is more than 4.

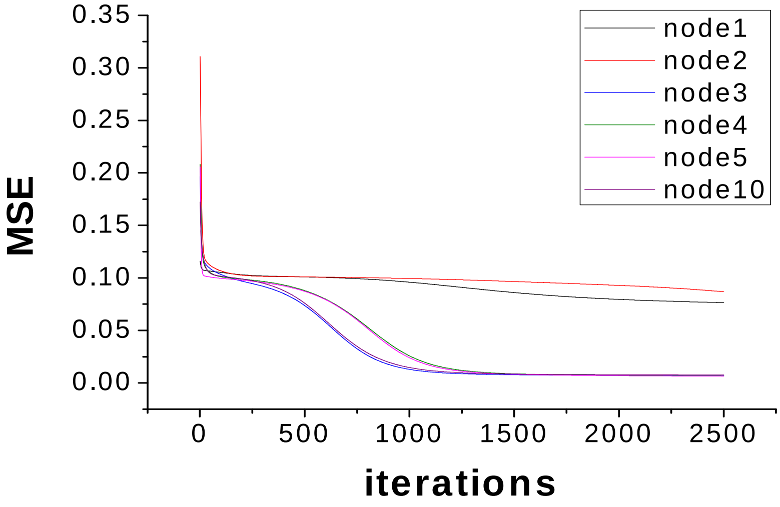


Fig. 2.10. Learning curve of a back-propagation ANN with different number of hidden nodes.

After training, the ANN can simulate the training data and predict the test data as shown in Figure 2.11. The predicted values of ANN agree very well with the theoretic curve.

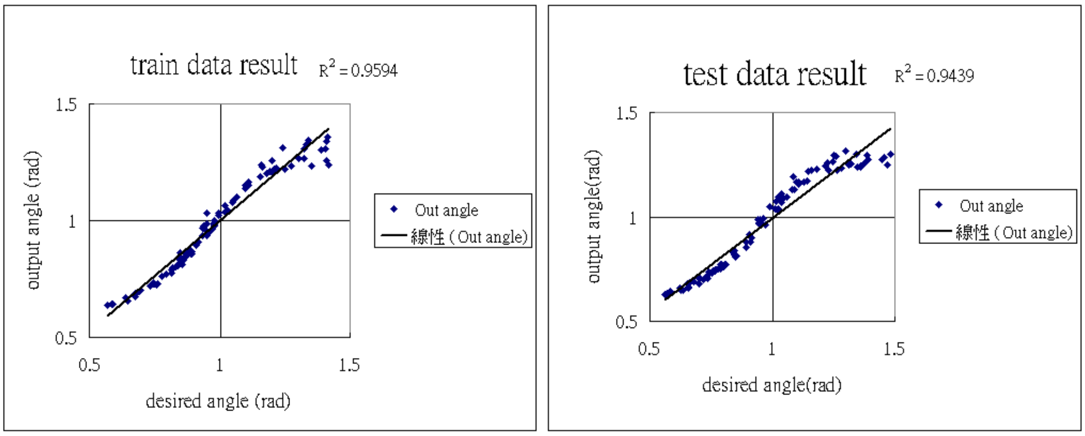


Fig. 2.11. The input-to-output characteristic curve of the back-propagation ANN with (a)the training data and (b) the test data.

2.2.5 Application of Back-Propagation Artificial Neuron Network for Time Series Prediction

Time series prediction is useful for the prediction of weather temperature, the periods of sunspot, and chaos, etc. Time series prediction is very similar to functional approximation. Here we will focus on the issue of how to use ANN to predict the behavior of a chaotic series. We created a chaotic series with the follow formulas:

$$\begin{aligned} &Y_0 = 0.01 \\ &Y_{t+1} = 4Y_t(1-Y_t)\text{,} \end{aligned} \tag{2-14}$$

which is results in a logistic map with *r*=4.

For time series prediction, the input to an ANN is the data we have prepared. For example, we can use Y_1 to Y_3 from Eq. 2-14 as the input data and Y_4 as the desired output of the ANN. Similarly, we can shift one position and take Y_2 to Y_4 as the input and Y_5 as the desired output. In this way, we can prepare numerous dataset to train the ANN. We chose the learning rate to be 0.01 for all weighting and bias parameters. The learning curves with different sets of system parameters are presented in Figure 2.12. After training, we can use the ANN to predict the data remaining in the time series data. The predicted values of the test data and training data by the ANN are plotted in Fig. 2.13, indicating the performance of the ANN is excellent with the predicted values almost identical to the real values in the chaotic series.

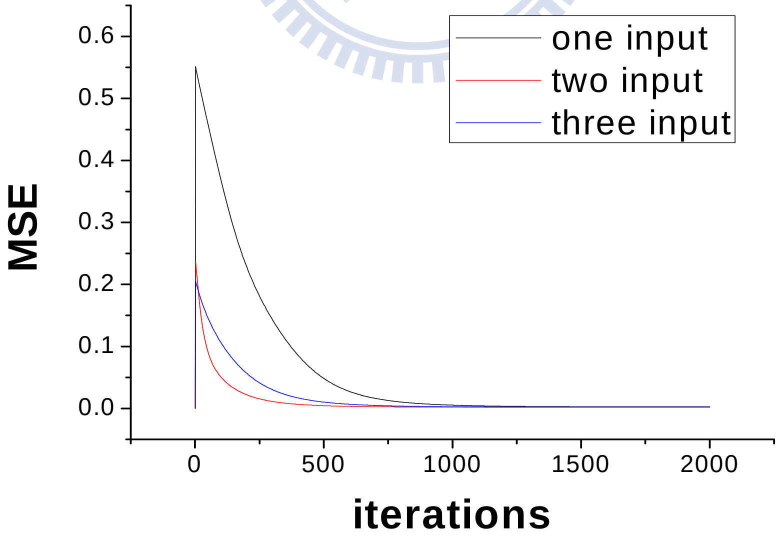


Fig. 2.12(a) The learning curves of an ANN used for time series prediction. During the learning phase, the data prepared for training the ANN is single, two serial numbers, three serial numbers, respectively.

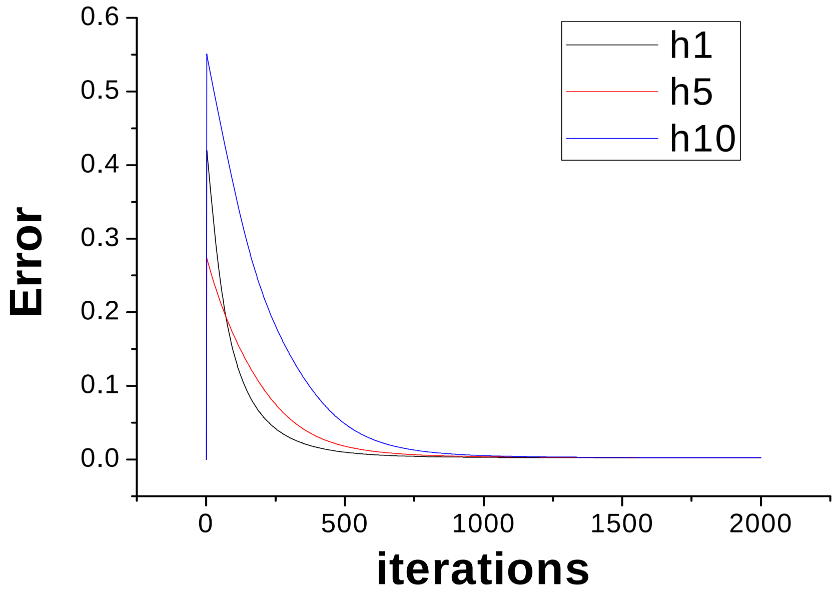


Fig 2.12(b) The learning curves of an ANN with different number of hidden nodes.

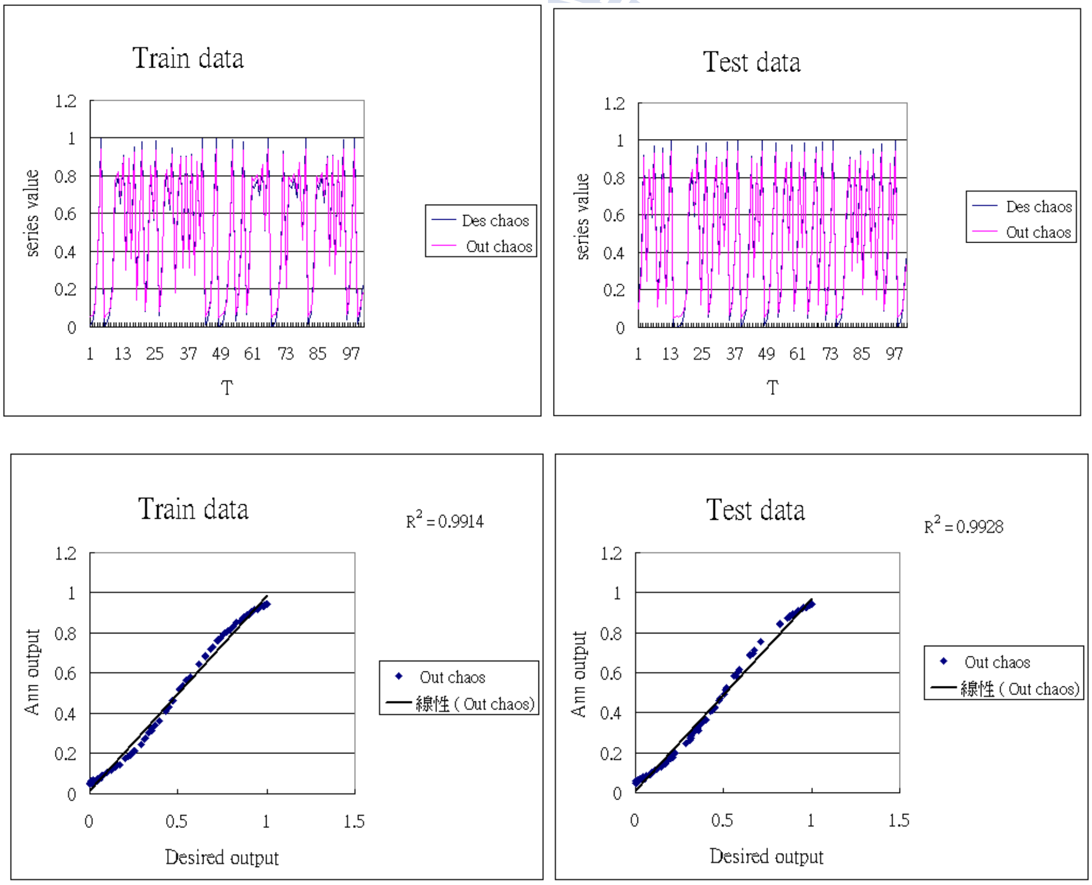


Fig. 2.13. The time series (top) and the input-to-output characteristic curves (bottom) of a back-propagation ANN with the training data (left) and the test data (right).

2.3 Summary

From the case studies shown in this chapter, we found that ANN is useful for numerous applications. Most of the applications of ANN use the back-propagation structure. Function approximation is useful to simulate the behavior of a physical system, which the underlying processes inside the system are unclear. We can build an ANN to simulate the relationship between the input and output of a physical system. The most sensitive issue of ANN relates to the training process. The training process requires much CPU time and may yield poor performances if an inappropriate ANN structure is implemented and is trained with inappropriate data sets.

Chapter 3 Complete Characterization of Ultrashort Coherent Optical Pulses with SHG Spectral Measurement

3.1 Introduction

As explained in Chapter 1, the complete field characterization of coherent optical pulses is the first step to invoke these optical pulses for optical metrology. Several techniques had been developed to offer the complete characterization of coherent optical phases, such as frequency-resolved optical gating (FROG) first reported by D. Kane and R. Trebino [2], and spectral-phase interferometry for direct electric field reconstruction (SPIDER) developed by T. Tanabe, et al. [3].

The basic concept of FROG is quite similar to the autocorrelation measurement but FROG measures the spectrums at different time delays instead of optical intensity only. Retrieving the spectral phases and then yielding a complete-field information of the coherent pulse under study is via an iteration algorithm. SPIDER can directly measure the spectral phase of a coherent pulse with a spectral-shearing interferometer, which separates the incoming coherent pulse into two parts and sent one part through a linear spectral phase modulator, and the other through a linear temporal phase modulator. And then by superpose these two parts together to yield a spectral-shearing interferogram.

The spectral phase and therefore the complete field information of the coherent pulse can be deduced directly from the interferogram without involving any further iterative calculation.

Along the development of complete coherent pulse characterization, Dorrer, *et al*. had invoked a self-referencing device based on the concept of shearing interferometry in the space and frequency domains to perform the spatio-temporal characterization of ultrashort light pulses [22]. Weiner *et al*. [23] had demonstrated the spreading of femtosecond optical pulses into picosecond-duration pseudo-noise bursts. In this case, pulse spreading was accomplished by encoding pseudorandom binary phase codes onto the optical frequency spectrum. Subsequently, decoding of the spectral phases restores the original pulse. Shelton *et al*. have generated a coherently synthesized optical pulse from two independent mode-locked femtosecond lasers, providing a route to extend the coherent bandwidth available for ultrafast science [24]. Applications of coherent light pulse characterization techniques in femto-chemistry had been well reviewed in [25].

Another attractive approach to characterize coherent laser pulse is to use an adaptive feedback-controlled apparatus to tailor the spectral phase of a coherent pulse to achieve the maximum second harmonic generation output from a nonlinear optical crystal [26]. In this way, the compensating spectral phases carry the spectral phase information about the coherent pulse under study.

Control the quantum evolution of a complex system is an important advance in optical metrology. The technique has now been coined as coherent or quantum control. Adaptive coherent pulse control [27-30] is the most successful scheme to be used for quantum control. Several algorithms have been developed to tailor a coherent optical field for a specific target on the basis of fitness information [31-36]. In this regard, a freezing phase concept had been proposed for adaptive coherent control with a femtosecond pulse shaper [26].

Our main goal of this study is to develop an artificial neuron network (ANN) model which can be used to retrieve the spectral phase of a coherent pulse directly from the spectrum of the second harmonic generation (SHG) with a nonlinear optical crystal. The SHG spectrum is affected by both the SHG process and the spectrum of the incident light pulse. In this chapter, we will develop an ANN to help us retrieving the spectral phase and therefore the complete-field information of a coherent pulse (phase and spectrum) with the measured spectrum of second harmonic generation.

Assuming the temporal profile of a coherent pulse is known, therefore we only need to adjust the spectral phase of the input pulse to generate the maximum SHG output from a nonlinear crystal. From the measured SHG spectrum, we retrieve the spectral phase of the input coherent pulse with an artificial neuron network. If the approach is successful, we can simply retrieve the complete field information of a coherent pulse in real time directly from a measured SHG spectrum without time-consuming computation. The apparatus needs only NLO (Nonlinear Optics) crystal and a spectroscope.

3.2 Theory

Considering an incident coherent optical pulse $E(\omega) = A(\omega)\,e^{i\varphi(\omega)}$ with a spectrum of $A(\omega)$ and spectral phase distribution of $\varphi(\omega)$. The second harmonic generation spectrum can be expressed as

$$I_{SHG}(2\omega)\alpha\left|\int E(\omega+\Delta)^{*}E(\omega-\Delta)\,d\Delta\right|^{2}. \qquad (3\text{-}1)$$

Assuming the spectrum of the coherent pulse to be Gaussian, and the spectral phase profile can be properly depicted with a polynomial of order 6, usually factor the phase of a high order is much small than the low order we cut it off at the order six.

$$A(\omega) = \exp(-\omega^{2}/\omega_{0}^{2}), \qquad (3\text{-}2)$$

$$\varphi(\omega) = \sum_{n=0}^{6} a_{n}\omega^{n} \quad . \qquad (3\text{-}3)$$

In general, the phase terms of order zero and one do not have any effect on SHG. The spectral phase profile can be further simplified by including terms from two to six only.

Note that from the point of view of theory, it shall be impossible to retrieve the spectral phase of a coherent pulse directly from the SHG spectrum of a coherent pulse. Therefore, in the following we will conduct some simulations to test the feasibility of the concept.

3.3 Simulation 1

3.3.1 Preparation of the training data set

To prepare the training process of ANN, we sampled the spectrum and phase of a coherent Gaussian pulse to generate 64 data points. The second harmonic spectrum is presented with a data array of 127 data points because the second harmonic spectrum was calculated via a convolution operation.

The schematic of the training process is detailed in Fig 3.1. The input into a back-propagation artificial neuron network is the data of the second harmonic generation pulse comprising a spectral profile array and a spectral phase array.

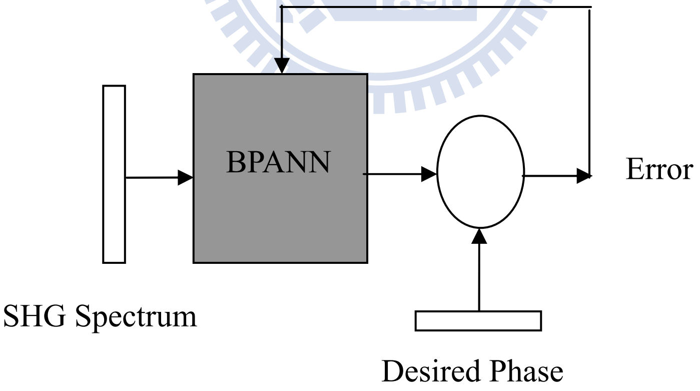


Fig. 3.1. The schematic showing the training process of a back-propagation artificial neuron network. The input data to the ANN is prepared from the SHG Spectrum generated by a coherent pulse with a Gaussian amplitude profile and a desired phase profile.

3.3.2 Creation of a Backward Propagation Artificial Neuron Network

The input layer of the BP ANN was designed to accommodate the 127 inputs of the second harmonic generation spectrum. The output layer generates the retrieved spectral phase profile for the coherent pulse under study. A typical training data for the coherent pulse under study is shown in Fig. 3.2. The resulting SHG spectrum obtained from the training data is presented in Figure 3.3. We had investigated BP ANN with different numbers of hidden nodes and different learning parameters to find out the best learning performance of the artificial neuron network. The results will be discussed in the following section.

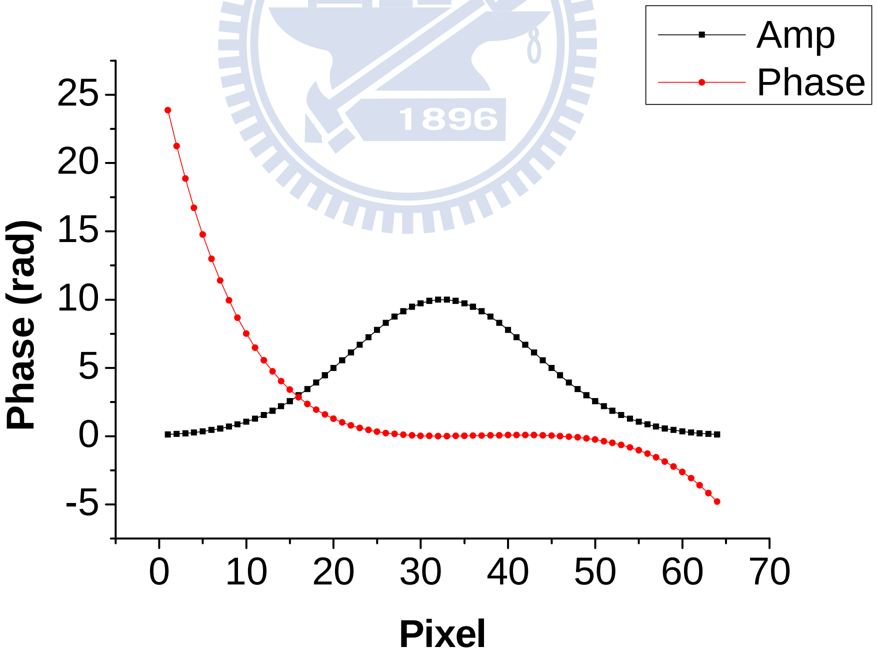


Fig. 3.2. Typical training data prepared for the BP ANN. The data comprises the spectrum and the spectral phase of the coherent pulse under study.

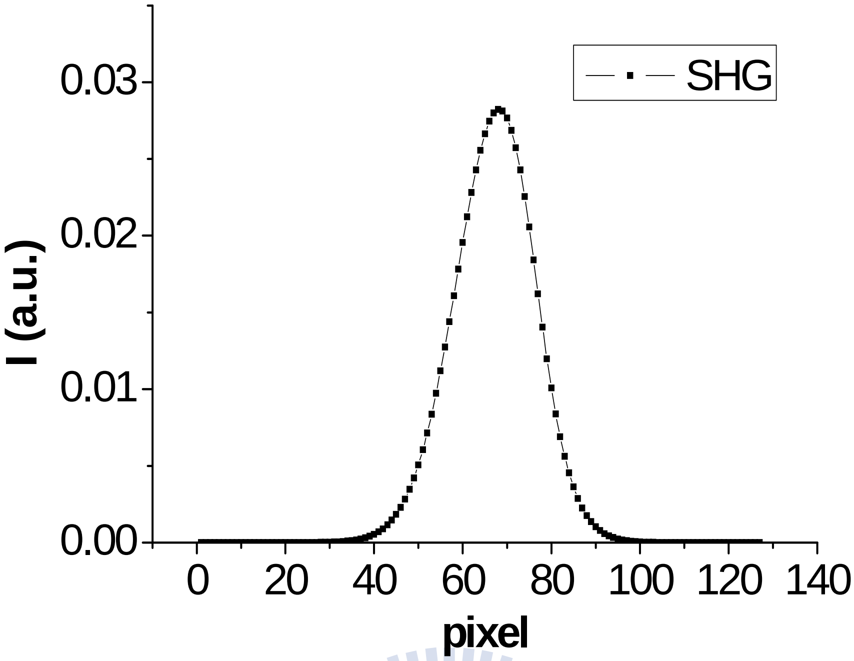


Fig. 3.3. The resulting SHG spectrum obtained from the training data shown in Fig. 3.2.

3.3.3 Results and Discussion of Simulation 1

In this simulation, we used a two-layer BP ANN to find out how many hidden nodes are needed to yield a satisfactory learning performance. The active function of each node was chosen to be the sigmoid function and the rate constant of learning rate was set to be 0.1 for the first and the second weighting layers. In the learning phase, we trained the network by 1000 epochs. To evaluate the performance of the training, we used a correlation coefficient *r*, which is define as

$$r = \frac{S_{xy}}{S_x S_y} \ . \tag{3-4}$$

Here $S_{xy} = \frac{1}{n-1}\sum_{i,j=1}^{n}(x_i - \overline{x})(y_j - \overline{y})$, and S_x, S_y are the standard deviations of x, y. The correlation coefficient can reveal the underlying relation between two sets of data. It value lies between -1 and 1 with value one implies a prefect linear dependence. In our case, r=1 means the output of the ANN is same as the desired output.

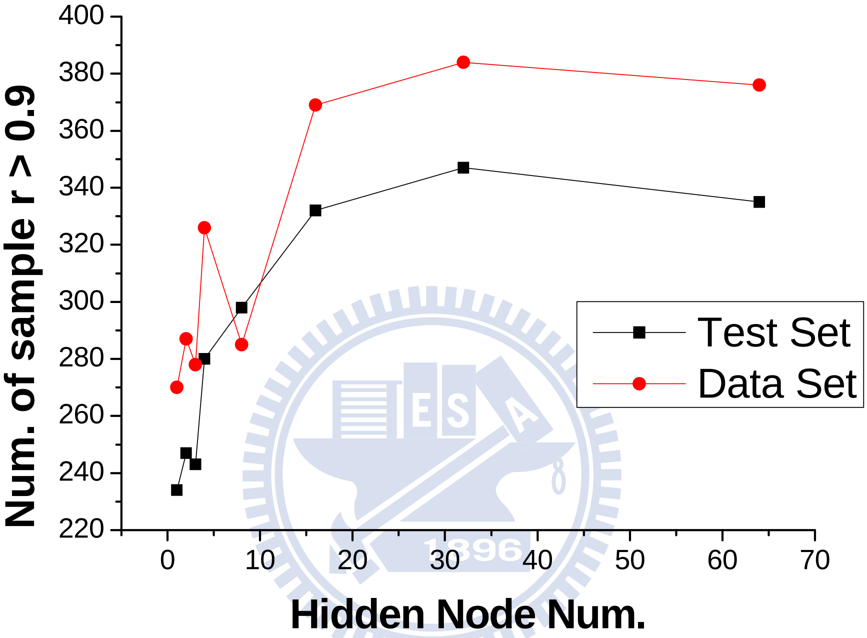


Fig. 3.4. The curves showing the relation of the population number with $r > 0.9$ in 1000 data points with the number of hidden nodes in the BP ANN.

Figure 3.4 presents a plot of the population number with $r > 0.9$ in 1000 data points as a function of the number of hidden nodes used in the BP ANN. We can find that by increasing the number of hidden nodes the number of data points with the correlation coefficient higher than 0.9 increases, implying that the predicted values with ANN can approach to the target phase. As the number of hidden nodes is larger than 20, the increasing tendency becomes stagnated. Therefore, the best choice is a BP ANN with each hidden layer containing about 20 hidden nodes.

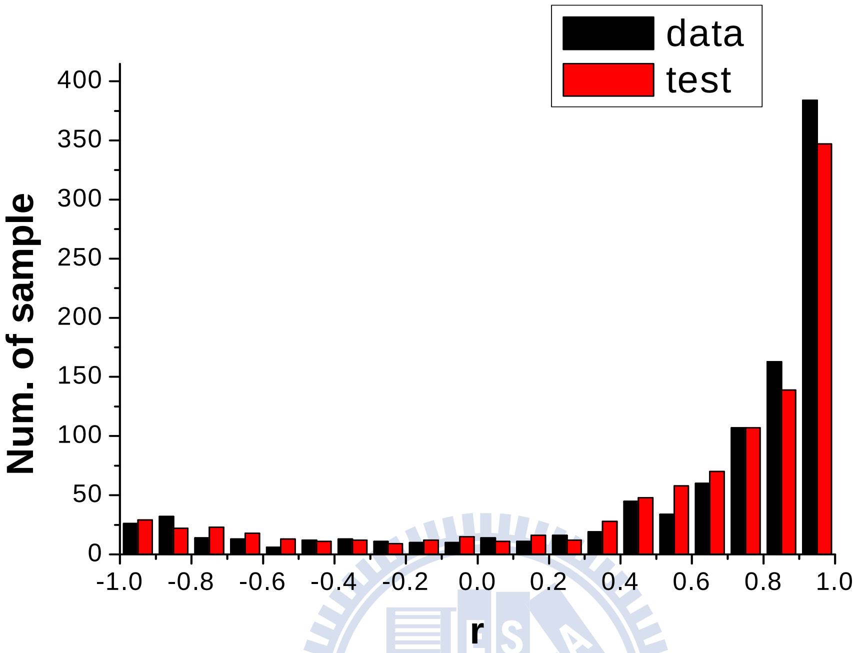


Fig. 3.5. The resulting distribution of *r* with 1000 test data points , and 1000 test data samples. The number of hidden nodes used is 32.

From the distribution of *r* presented in Fig 3.5, we found that only about 35% of the predicted values of spectral phase has a correlation coefficient higher than 0.9. Apparently, the learning performance of this SP ANN is not satisfactory. In view that the spectral phases used are expressed in terms of polynomials, we may be able to solve the problem with an increase of the information content in the training data by including more orthogonal phase profiles. Therefore, in the next section, we will try to express the spectral phase profile in terms of Legendre polynomials of order 2 to 6, which are shown in Figure 3.6.

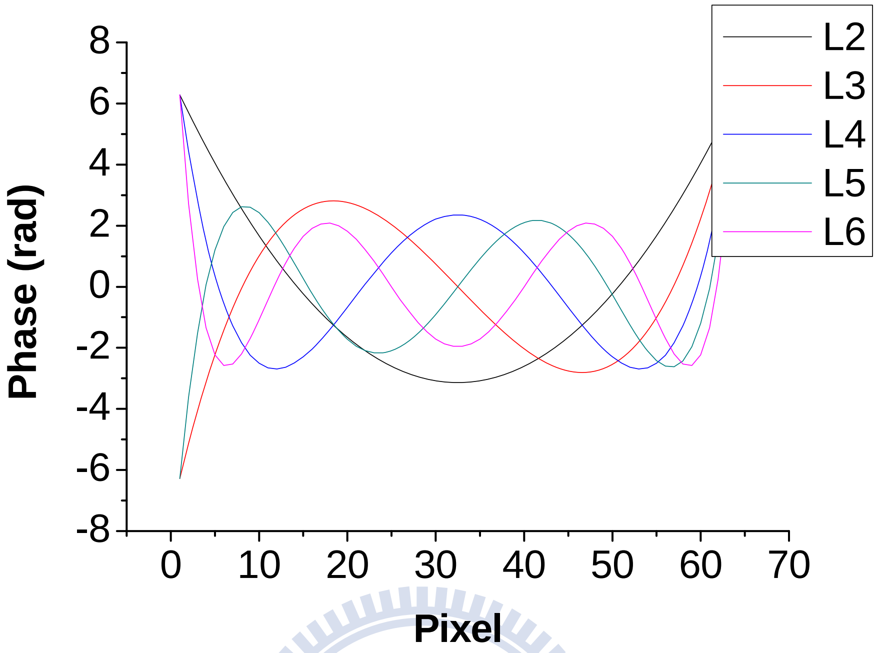


Fig. 3.6. L2, L3, L4, L5, L6: the Legendre polynomials of order 2 to 6.

3.4 Simulation 2

3.4.1 Preparation of the Training Data Set

Legendre polynomials form a complete set of orthogonal basis for a continuous function. By expanding the spectral phase profile of a coherent pulse into Legendre polynomials, we can significantly increase the information content with a minimum number of Legendre polynomials. Indeed as shown in Fig. 3.7, by including Legendre polynomials of order 2 to 6 in the spectral phase profile $\varphi(\omega)$, more complicated SHG spectrum can be synthesized. The method significantly increases the degrees of freedom in the phase retrieval procedure. Six SHG spectra are prepared for the training of BP ANN by including more Legendre polynomials in the spectral phase are shown in Figure 3.7.

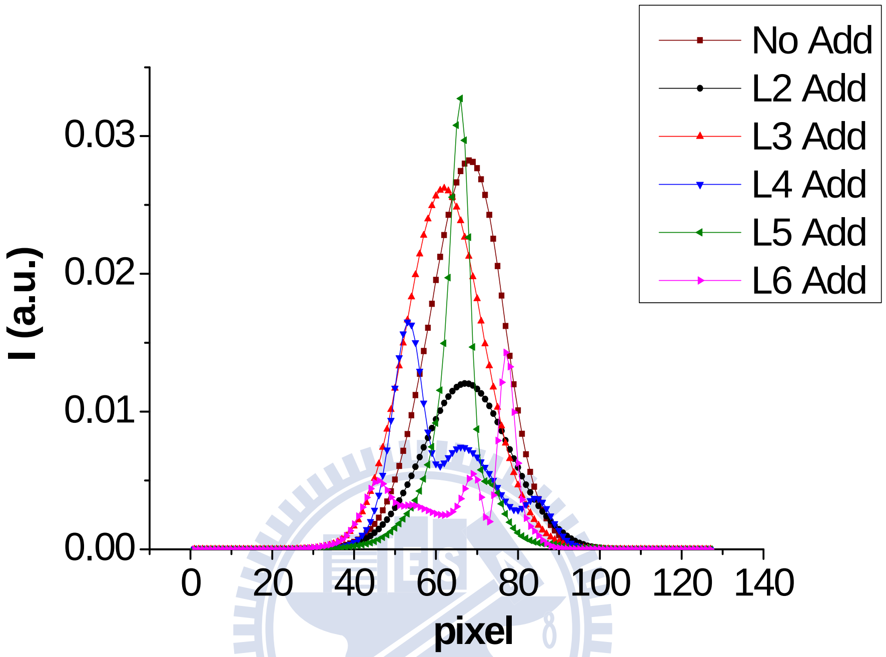


Fig 3.7. Six SHG spectra are prepared by including more Legendre polynomials in the spectral phase.

3.4.2 Creation of the Backward Propagation Artificial Neuron Network

For this study, we built another BP ANN which contains 762 input nodes and 64 output nodes. The 762 (6x127=762) input nodes are designated for the six SHG spectrums and 64 output nodes are for the retrieval spectral phase of the coherent pulse under study. The learning rate is set to 0.1 for both the first weighting layer and the second weighting layer. The data-flow schematic for the training is shown as Fig 3.8.

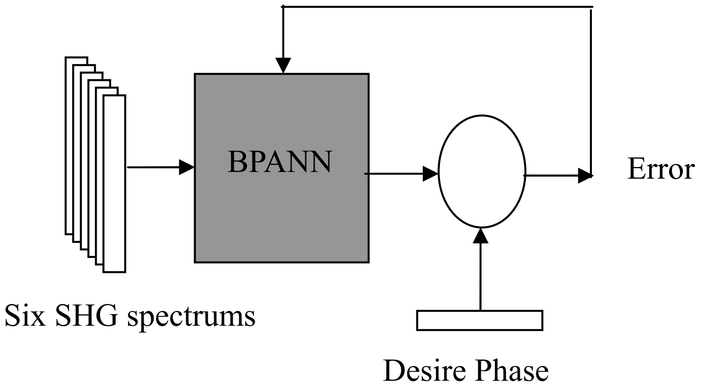


Fig 3.8. The data-flow schematic for the BP ANN training. Six SHG Spectra as shown in Fig. 3.7 were used.

3.4.3 Results and Discussion of Simulation 2

The simulation results showing the relation of the population number with $r > 0.9$ in 1000 data points with respect to the number of hidden nodes in the BP ANN used are presented in Fig. 3.9. The results are quite encouraging in view that the number of the predicted phases with a correlation coefficient $r> 0.9$ can reach more than 90% of the test samples.

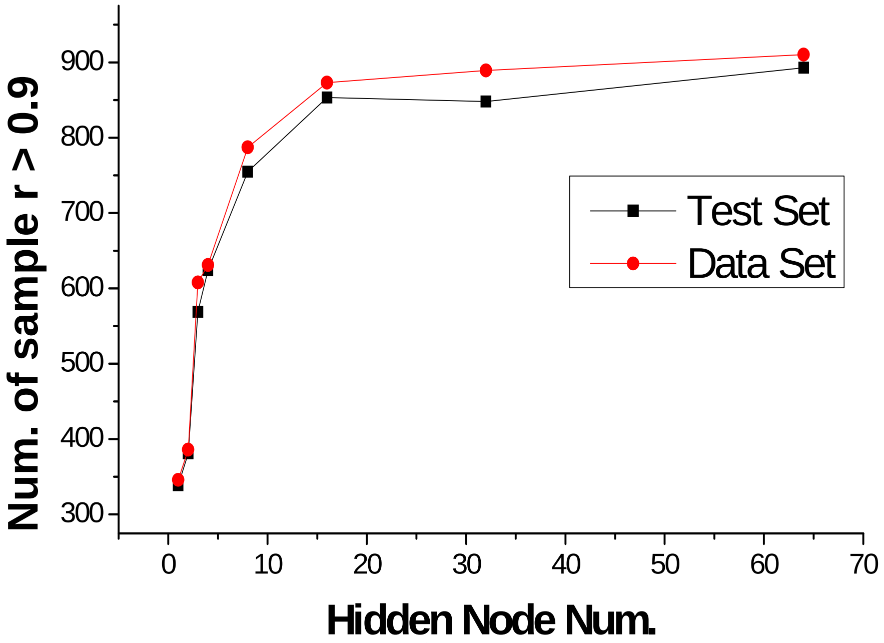


Fig 3.9. The curves showing the plot of the population number with *r* > 0.9 in 1000 data points as a function of the number of hidden nodes in the BP ANN.

We presented in Fig. 3.10 some representative profiles with *r*>0.9 to give some hints of how well the BP ANN performs. From this Figure, we can see that the phase profile retrieved by our ANN agrees very well with the target profile.

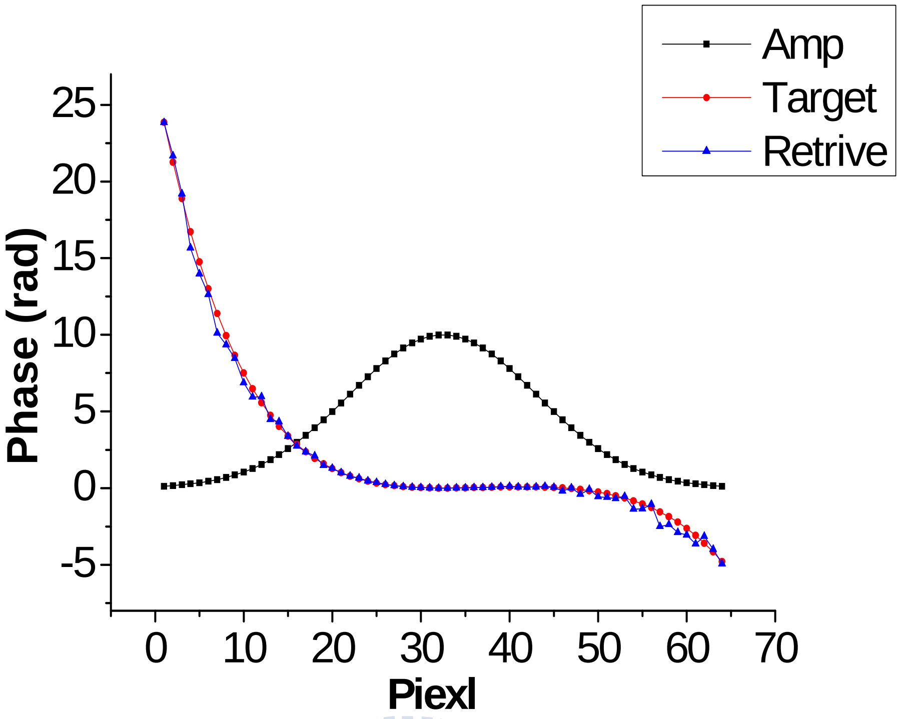


Fig 3.10. A representative phase profile retrieved from the BP ANN with *r*>0.9 is plotted to reveal the close agreement with the target phase profile.

3.5 The Proposed Experimental Setup

For an experimental realization of the technique, one may concern how to conduct the training of BP ANN and then how use the trained ANN to perform the complete field characterization of a coherent pulse experimentally. We proposed an advanced apparatus with a pulse shaper to yield an adaptive feedback control loop as depicted in Fig. 3.11. By using this apparatus, we can first measure the spectrum of the laser pulse under study. We can produce many possible phase distorted versions of the coherent pulse by combining the measured spectrum with a variety of spectral phases expressed as a series of Legendre polynomials. A variety of SHG spectra are then synthesized to train the BP

ANN. After training, the trained ANN can be invoked to retrieve the spectral phase of the coherent pulse with the measured SHG spectrum. The apparatus offers a possibility to perform a complete-field characterization of a coherent excitation and quantum control of a physical system in a single setup.

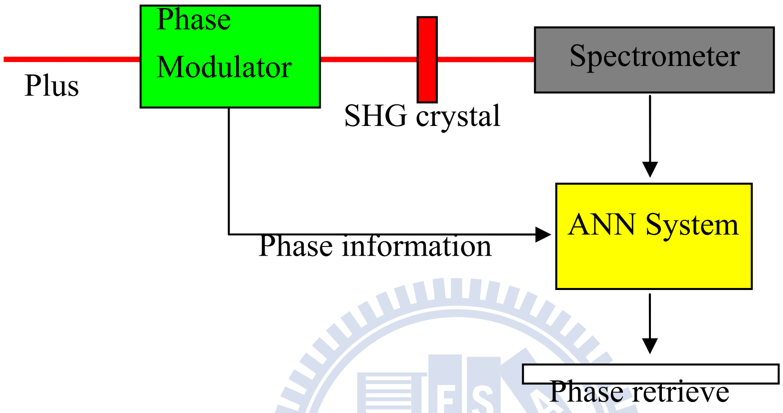


Fig. 3.11. Proposed Experimental setup.

3.6 Conclusions

We developed a BP ANN which can be invoked to retrieve the spectral phase of a coherent pulse from the measured SHG spectrum. We proposed a setup to be used for the experimental realization of the concept. By using this apparatus, we only need to measure the spectrum of the coherent pulse under study and then combine the spectrum with a variety of spectral phase profiles to prepare the SHG spectra for training the BP ANN. The trained BP ANN can be invoked to retrieve the spectral phase profile for the complete field characterization of a coherent pulse. From a simulation study, we found the predicted phases can achieve the target profile with more than 90% confidence. Thanks to the computation efficiency of BP ANN, the technique developed in this study offers a possibility to perform a complete-field characterization of a coherent excitation to a physical system and quantum control of the system in a single setup.

Chapter 4 Real-Time Localization of Nano Objects at the Nanometer Scales

4.1 Introduction

An isolated fluorescent molecule or nano object will be observed like a light spot under an optical microscope. The spatial profile of the light spot simply reveals the point spread function of the optical microscope used. The peak position of the light spot can be determined with an accuracy of 1 nm if the signal-to-noise ratio of the detection is high enough. This impressive feature of nanometer localization with optical microscopy had recently inspired many applications including Fluorescence Imaging with One Nanometer Accuracy (FIONA) [37], sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM) [38], and fluorescence photoactivation localization microscopy (FPALM) [39], *etc*. Important biophysical mechanisms at the subcelluar scales had been discovered [40].

In the historical point of view, we noticed that a modified Hough transformation had been developed to detect a circular object [41] for recognizing and classifying interesting features present in a phase-contrast (PC) cytological image. Fillard, *et al*. had invoked the frequency dependence of the argument of Fourier transform to analyze an in-focus two-dimensional Airy disk [42]. Alexander, *et al*. proposed a method to eliminate the systematic error in centroid estimation and achieved a subpixel accuracy [43]. It is interesting to know that a diffractive optical element (DOE) [44] had been applied to effectively locate a laser spot on a projection screen. The method could be invoked to achieve nanometer localization for optical microscopy. Anderson had presented an algebraic solution to the problem of localizing single fluorescent particle with sub-diffraction-limit accuracy [45]. Qu, *et al*. [46] had demonstrated nanometer localization of multiple single-molecules (NALMS) by using fluorescent microscopy and photobleaching properties of fluorophores. Cui, *et al*. [47] had devised an optimized algorithm useful for localizing light spots in high noise background. A method [48] combining the radial basis network with anisotropic Gaussian basis function had been used to detect the position of a fluorescent protein. Fillard [49] relied on the Fourier phase frequency dependence to achieve sub-pixel localization accuracy of a light spot. Enderlein [50] had proposed a method useful for tracking single fluorescent molecules diffusing in a two-dimensional membrane by invoking a rotating laser focus to track the position of the molecule. More information about single-molecular imaging and spectroscopy can be found in [51].

Based on the technical review, we found the major issue in the localization and tracking of nano objects is how to localize these objects accurately and rapidly with minimum invasiveness. To achieve the goal, many algorithms had been developed.

Fitting the light spot to a 2D Gaussian function is the most popular technique in this field. Another useful technique is to retrieve the peak position of a light spot via a center-of-mass approach [4]. However, to invoke these two techniques to localize a light spot with large size often fails to yield an accurate result. Therefore, in this chapter we will develop an ANN model to rapidly localize multiple light spots with high accuracy. We invoked the feature of function approximation of ANN. We expect that the localization accuracy of ANN can be further improved when more data are accumulated. Comparing to the 2D Gaussian fitting method, our ANN localization method is also less sensitive to noise influence.

4.2 Data Preparation for Training the Artificial Neuron Network Localization Model

To train and test the performance of an ANN localization model, we prepared an image of 10 bright spots with 256x256 pixels as shown in Figure 4.1. The brightness profile of the light spots is Gaussian. The main target of this study is to construct a trained ANN model which can be invoked to yield the peak positions with a localization error less than one pixel.

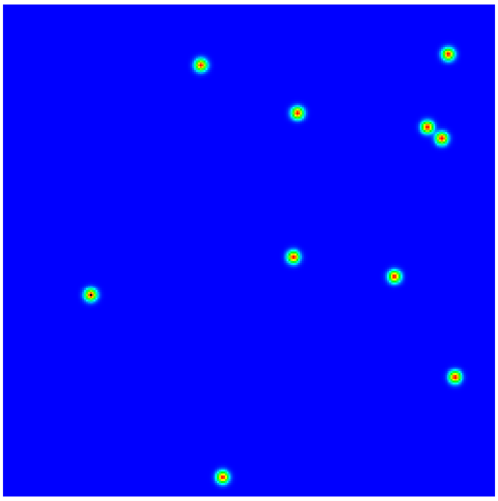


Fig 4.1. An image of 10 bright spots prepared for training the artificial neuron network localization model.

4.3 First Test Run of the Artificial Neuron Network Localization Model

In this study, we constructed a BP ANN for localizing bright spots in an observing region. To serve this purpose, we began at an image of one spot with 25x25=625 pixels shown in Fig. 4.2. We input this image into the BP ANN. The output is the coordinates (*x*, *y*) of the spot. Therefore, the BP ANN possesses a total of 625 input nodes, 2 output nodes, and a hidden layer of 30 nodes. The activation function of the hidden layer is chosen to be hyperbolic tangent (tanh), while the activation function of the output layer is sigmoid function. We set the rate constant of learning to be 1.0 and 0.1 for the first and the second weighting layer, respectively.

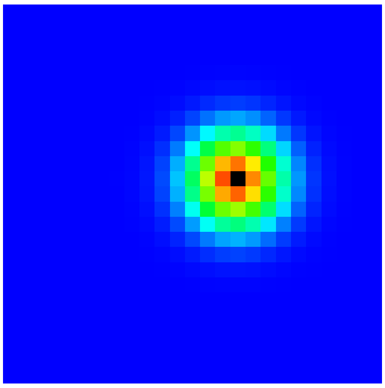


Fig. 4.2. One of the images with 25x25=625 pixels and containing single bright spot, used for training the artificial neuron network localization model.

The performance of the BP ANN trained by 10000 epochs with 1000 samples is shown in Figure 4.3. The distribution of the localization error deduced from the results with a training set of 1000 images or 1000 test images reveals that the localization error can be smaller than one pixel. However, in this example, we did not take into account the noise influence and only single bright spot is included.

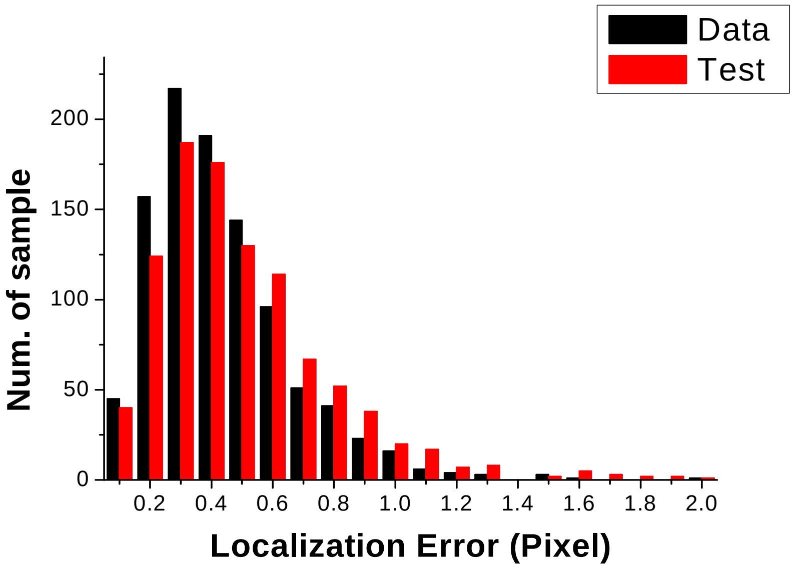


Fig. 4.3. The distribution of the localization error deduced from a run with either 1000 training images or 1000 test images.

4.3.1 Difficulty Encountered by the Artificial Neuron Network to Localize Multiple Light Spots

In the previous section, we found that the BP ANN can be used to successfully localize a single bright spot with a localization error less than one pixel. However, it is difficult to use single BP ANN to searching an area with multiple light spots. In fact, in a real situation, we do not know how many spots are involved in the beginning. Therefore, we try to combine two BP ANNs with the first ANN to determine how many spots are involved in a large region, and divide the region into a series of smaller domains with each containing single spot only. And then the second ANN will take over to localize each spot with high accuracy. The second BP ANN can be the same as the one described in the previous section. Thus, we separate the ANN localization procedure into the ANN searching step and the ANN localization step. We will detail each steps in the following sections.

4.3.2 The ANN Searching Step: Searching Over an Entire Region to Deduce the Number of Light Spots

In order to detect how many spots involved and their rough locations over an image area, we design a BP ANN to search over the entire region to find out the pixels with an intensity value exceeding a threshold value. Based on this searching result, we shall be able to deduce the number of light spots and roughly locate their positions.

4.3.3 The ANN Localization Step: Localize a Smaller Region to Yield the Coordinates with High Accuracy

We had demonstrated in Section 4.3 that a light spot in a region of 25x25 pixels can be localized to subpixel accuracy. For a quantitative analysis, we draw all the coordinates deduced in the learning phase over the image area of 25x25 pixels in Figure 4.4. The average inter-spot distance is about one pixel; the localization error is 0.5 pixel.

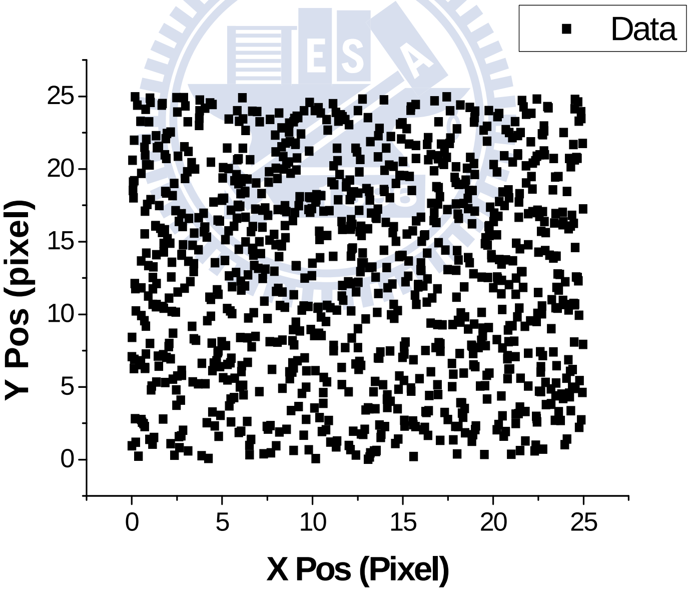


Fig 4.4. Distribution of the peak position of light spot taken from 1000 training data over an image area of 25x25 pixels.

4.4 Training ANN over a Small Image Region with Higher Data Density

In this section, we aim to reduce the localization error to less than 0.1 pixel. To achieve this goal, we build a BP ANN and apply it to analyze images with 11x11 pixels as shown in Figure 4.5. Furthermore, to reduce the localization error, we increase the number of training data to 10000.

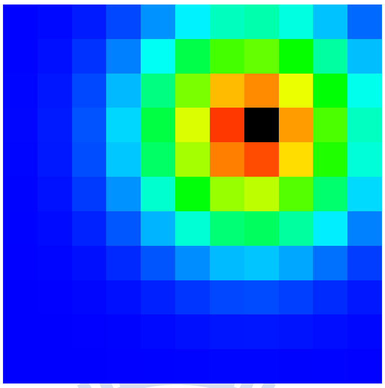


Fig. 4.5. One of the training images with 11x11 pixels and single bright spot.

The BP ANN we created contains 121 input nodes and 2 output nodes. The activation function of the input layer is chosen to be hyperbolic tangent (tanh) and that of the output layer is sigmoid function. The learning rates are 1.0 and 0.1 for the input and output weighting layers, respectively.

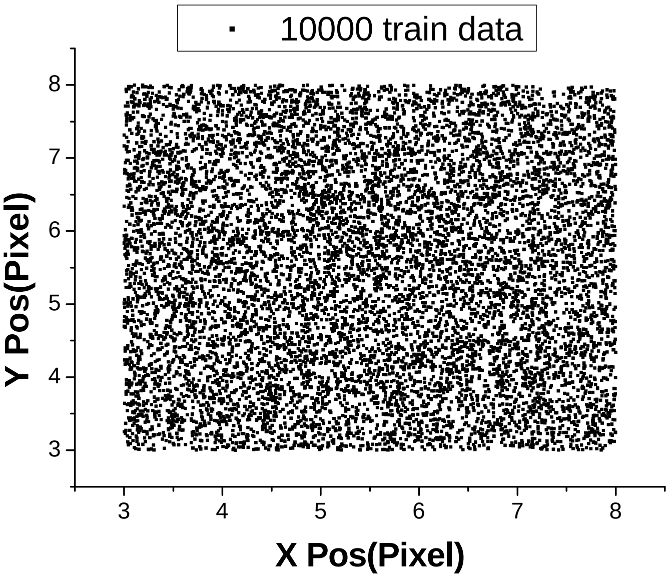


Fig. 4.6. Distribution of the peak position of light spot taken from 10000 training data over an image area of 11x11 pixels.

The peak positions of the 10000 training data are presented in Figure 4.6. The average inter-spot distance is much smaller than one pixel. After training 7500 epochs, the result of localization error is shown in Figure 4.7.

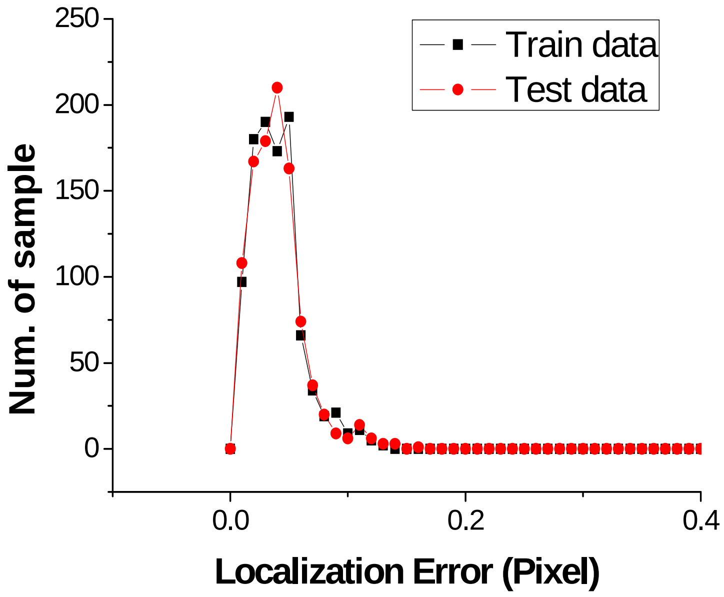


Fig. 4.7. The distribution of localization error taken from either 1000 training data or 1000 test data after the ANN has been trained for 7500 epochs.

From this figure, we can conclude that the spots in an image area of 11x11 pixels can be localized by the ANN to within an error of 0.1 pixel.

4.5 Comparison between ANN Localization Model and 2D Gaussian Profile Fitting Method

In the single molecular research with nanometer localization and tracking technique, the two-dimensional Gaussian fitting technique used is often limited to an image size of 11x11 pixels due to the compromise between speed and localization accuracy. Therefore, it is interesting to study the performance comparison between our ANN localization model and the 2D Gaussian fitting method in the same image size of

11x11 pixels.

The typical localization error of the 2D Gaussian fitting method is zero and it takes about 0.03 sec to complete the fitting process for each spot. On the other hand, the average localization error of our ANN is about 0.05 pixel but the localization time for each spot is negligible. The 2D Gaussian fitting method can be rendered into a linear algebraic problem and the computation time will increase rapidly as the number of the unknown parameters increases. Therefore, as show in Fig 4.8, we can find that the difference of the computation time between the two methods becomes significant as the image size and therefore the number of spots involved increases.

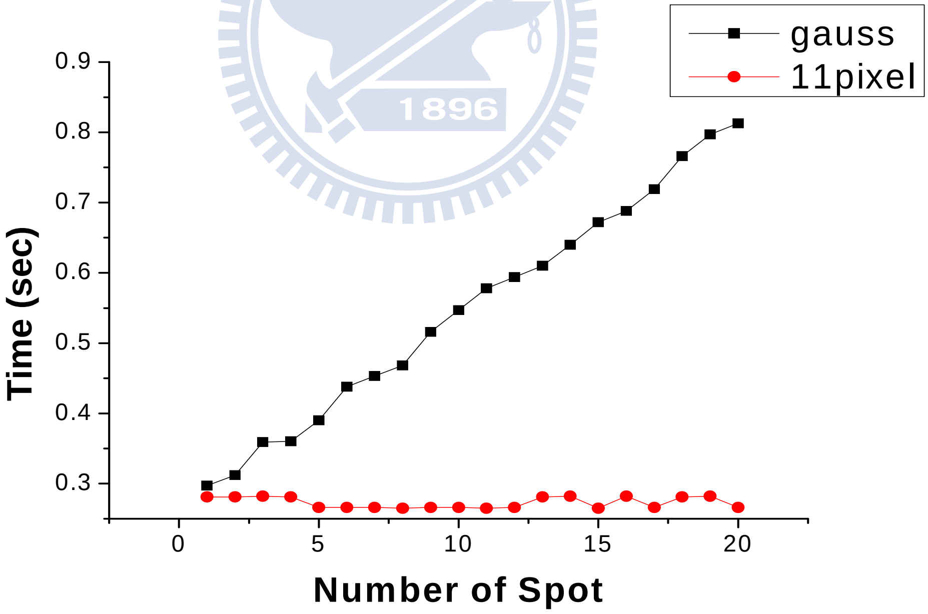


Fig. 4.8. Comparison showing the localization time of our ANN localization model and the 2D Gaussian profile fitting method as a function of the number of spots.

4.6 Noise Influence

For a real application with ANN, we shall consider the influence of noise that is unavoidable in experimentally measured data. In this section, we will study the influence of noise on the performance in more detail.

We prepared a set of test images by adding into the images with a noise level from 10 % to 50 % of the peak height of the light spot. The resulting images after adding noise are shown in Figure 4.9.

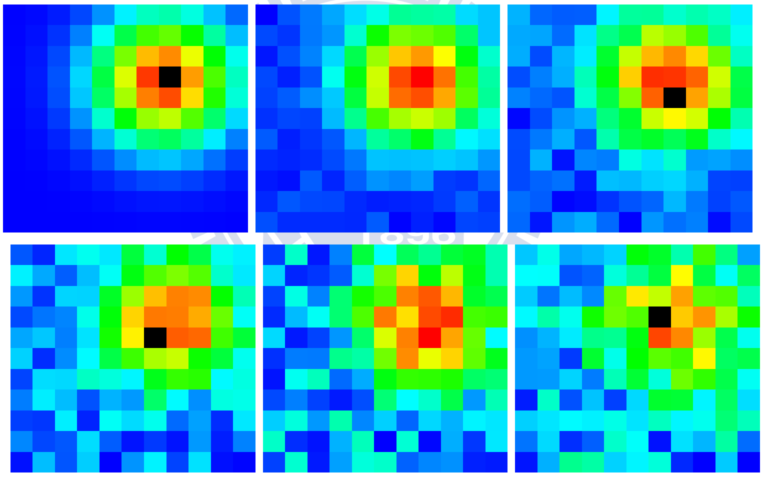


Fig. 4.9. Images showing the influences of noise on the training data. The noise level is set to be 0%, 10 %, 20 % , 30%, 40% and 50 % of the peak height (from top left to right bottom), respectively.

We used these images as a template and varies the spot position to generate a training set of 10000 images. We trained an ANN with structure the same as that to be described in Section 4.7. After training, the learning performances are shown in Figure 4.10.

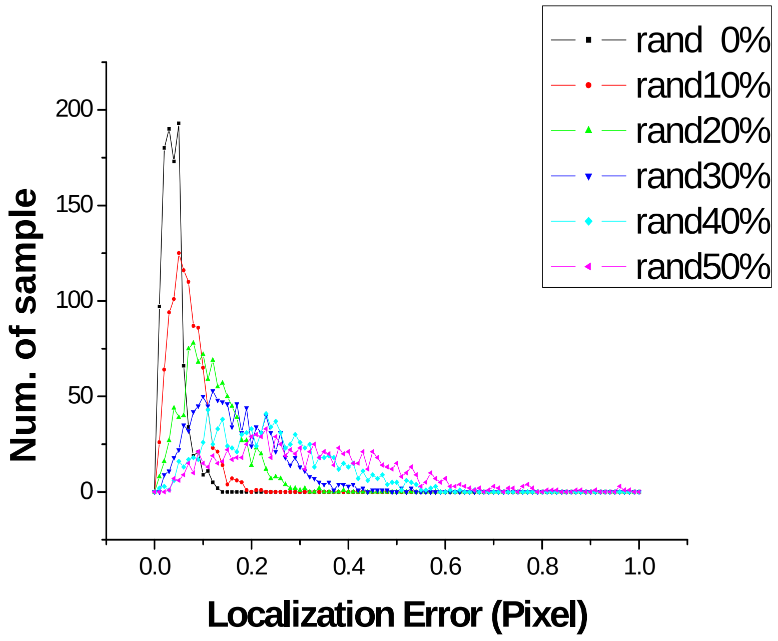


Fig. 4.10. The distribution of localization error taken from 1000 most populated test data among the set of 10000 data used in our ANN model. The test data are affected by noise with 10%, 20%, 30%, 40%, and 50% of peak height.

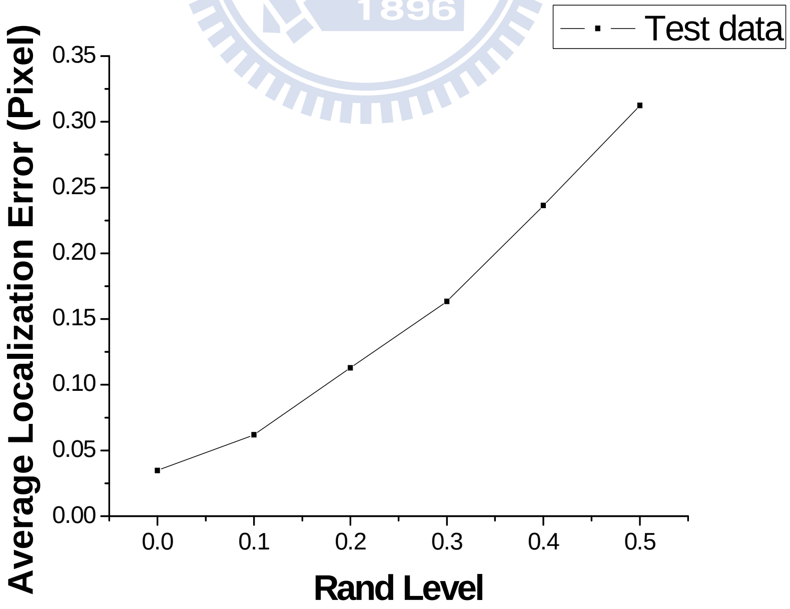


Fig. 4.11. The average localization error as a function of noise level in the test data.

We can find that the localization error increases as the noise level of the test images is increases. However, as shown in Fig. 4.11, even the noise level has been raised to 50% the average localization error can still be kept below 1 pixel.

For a fair comparison, we will examine the influence of noise on the 2D Gaussian fitting method by using the same test data set. The result is presented in Figure 4.12. Because our test spot was generated from 2D Gaussian profile, the resulting localization error with the 2D Gaussian fitting method is zero when noise is negligible. The localization error increases as noise level is increased. If we focus on the cases with the localization error below 1 pixel, the performance with the 2D Gaussian fitting method is almost identical to the ANN. However, as the noise level is above 30%, the probability that the 2D Gaussian fitting method fails to yield a peak position is larger than 20%, leading to that the average localization error becomes larger than 1 pixel as shown in Figure 4.13.

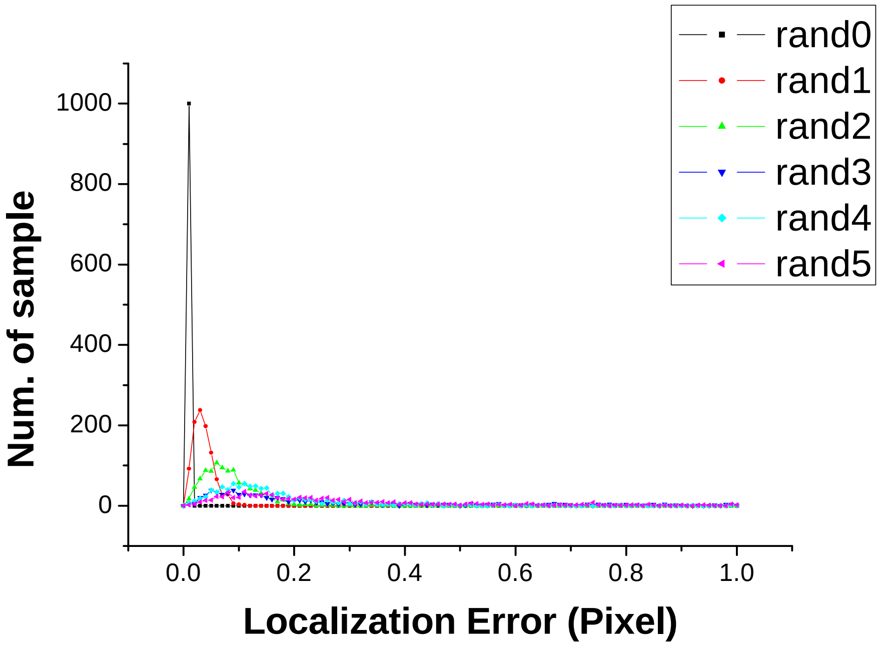


Fig 4.12. The distribution of localization error taken from 1000 most populated test data among the set of 10000 data used in the 2D Gaussian fitting method. The test data are affected by noise with 0%, 10%, 20%, 30%, 40%, and 50% of peak height.

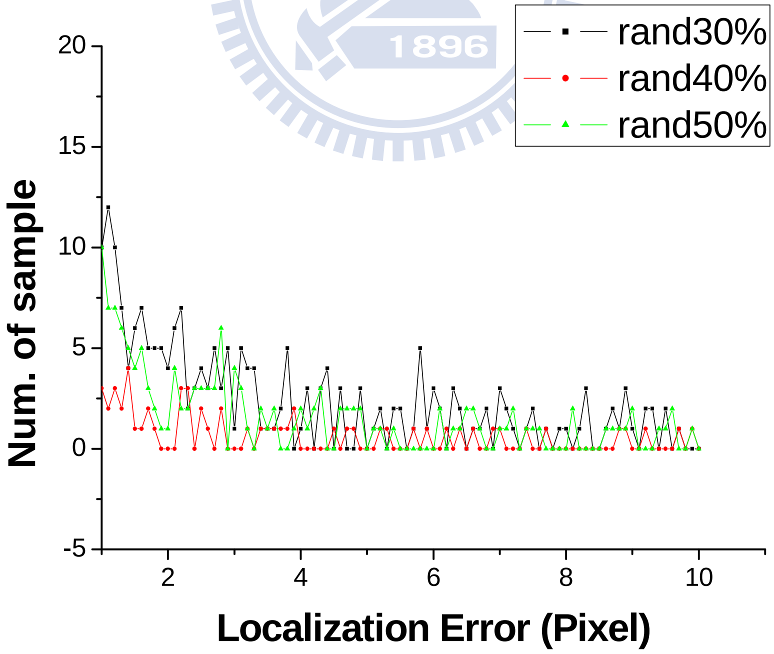


Fig 4.13. The distribution of localization error taken from 1000 most populated test data among the set of 10000 data used in the 2D Gaussian fitting method. The test data are affected by noise with 30%, 40%, and 50% of peak height.

4.7 The ANN Localization Model Suited for Large Area Searching

As mentioned in Section 4.3.1, we divide our ANN localization process into two steps. In the first stage, we use the first ANN to determine how many spots are involved in the entire region. For an image as shown in Figure 4.14, depending on the hardware, the complete localization (shown in Fig. 4.15) of all light spots may take less than 0.3 sec.

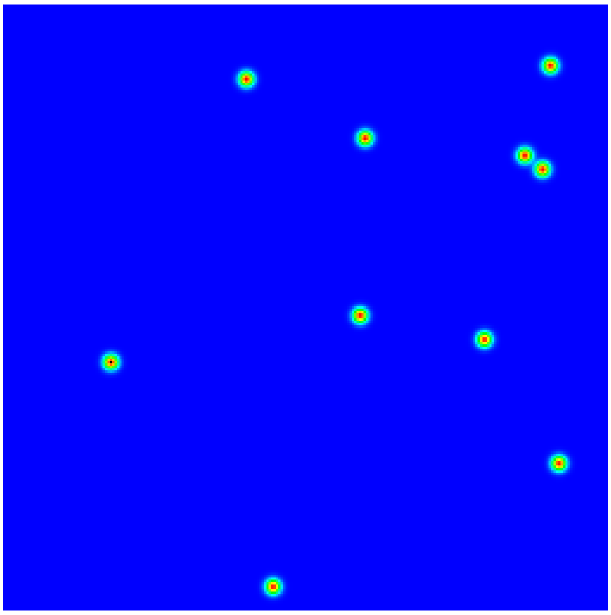


Fig. 4.14. An image of 10 bright spots in a size of 256x256 pixels prepared for testing the artificial neuron network localization model.

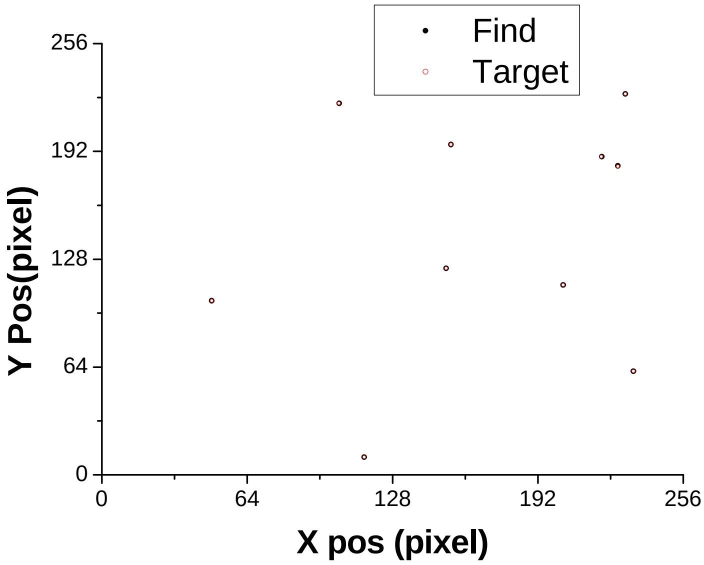


Fig. 4.15. The target positions (red circles) and the positions (black dots) retrieved with our ANN localization model.

In the second stage, we isolate a squared region around each spots within which the pixel value is larger than 90% of the peak value. The images of these isolated domains with each occupying 11x11 pixels are sent to the ANN to locate the peak position of each light spot with high accuracy.

In the following, we will analyze the influence of noise on the performance to assess the potential for a real application. The test image is added with a noise level of 10 % and 20 % of the peak height, respectively. The trained ANN model is used to localize the light spots. The results are shown in Figure 4.16.

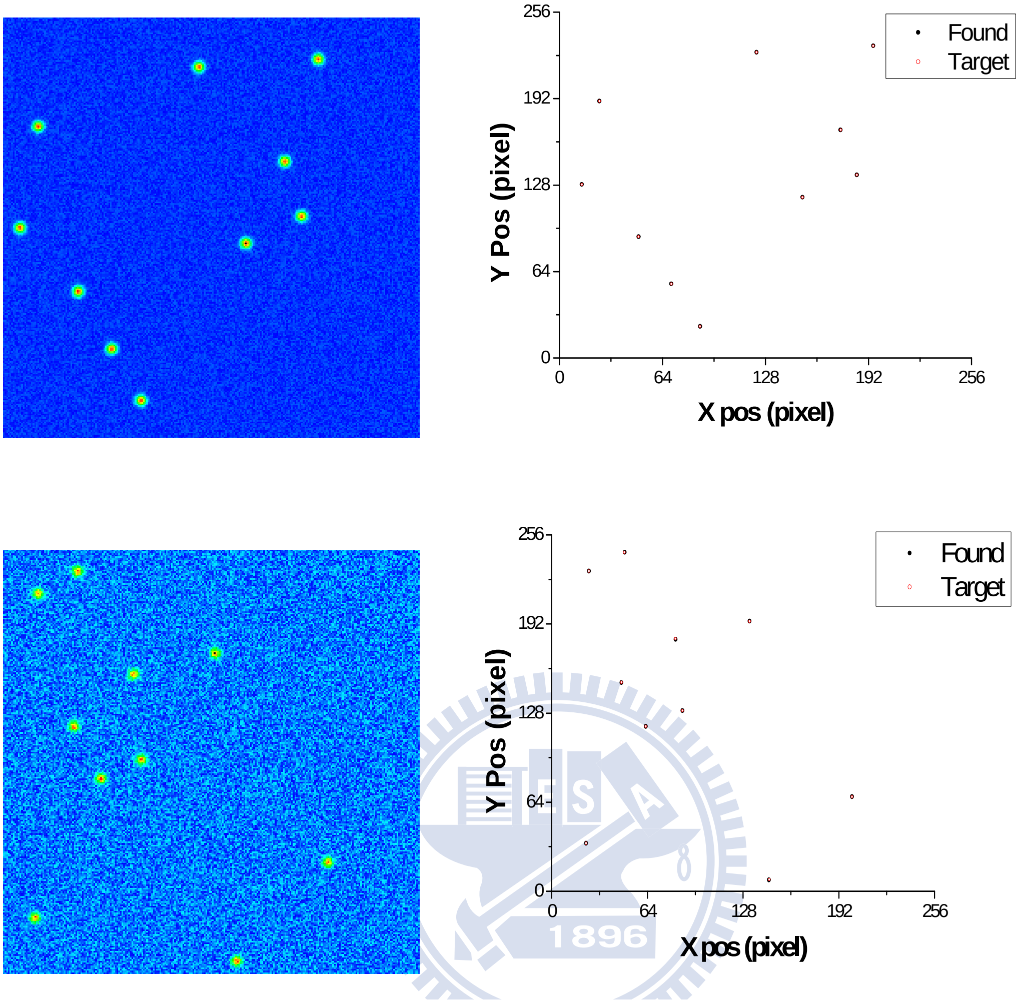


Fig. 4.16. The same run as Fig. 4.15 except that the test images are affected with different noise levels: (a) 10%; (b) 30%.

4.8 Conclusions

From these case studies reported in this chapter, we can conclude that localization of light spots with ANN is feasible with a performance better than that with the 2D Gaussian fitting method. This is true, especially when high noise level is involved in the images. The ANN also requires less computation effort in comparison with the Gaussian fitting method. Although the training procedure of an ANN takes a lot of computation time, this procedure only needs to be done once before its use.

Chapter 5 Conclusions and Future Work of This Thesis

5.1 Conclusion of This Thesis

In this thesis, we aim to implement the learning ability of artificial neuron network into an optical apparatus to accumulate the user experiences and improve the prediction accuracy of the artificial neuron network as more data are collected. We demonstrated that artificial neuron network can improve the measurement accuracy or generate new functionality of an apparatus. We focus on two case studies in optical metrology, including the complete characterization of ultrashort laser pulses and nanometer localization in real time for an optical microscope. From the first case study, we conclude that a trained back-propagation artificially neuron network can be invoked to retrieve the spectral phase profile of a coherent pulse from a measured SHG spectrum. The technique developed in this study offers a possibility to perform a complete-field characterization of a coherent excitation to a physical system and quantum control of the system in a single setup.

From the second case study, we developed a two-stage ANN strategy by first determining how many spots involved in the entire image. And then we invoked the second ANN to further process a set of squared domains around each spots to localize each spot. We successfully used the two-stage ANN strategy to localize multiple light spots in a large area with high accuracy. We concluded that localization of light spots with ANN is feasible with a performance better than that with the 2D Gaussian fitting method. This is true, especially when high noise level is involved in the images. In addition, the ANN localization model also requires less computation effort in comparison with the Gaussian fitting method.

5.2 Future Work

Although we have successfully applied ANN in two cases in optical metrology, the following issues related to the current work remain unsolved and worth further study:

(1) How many hidden nodes should one use in an ANN to reach the best performance for a specific application?

(2) Since the ANN is powerful for modeling a nonlinear system, it is possible to combine ANN with other solvers for nonlinear equations such as Landsweber iteration method to enhance the iteration performance.

(3) The initial guesses of the weighting parameters and biases may influence the learning speed and thereby the convergence. How to set up a better initial guess worth further study.

(4) In this thesis, we applied ANN to localize multiple bright spots in an optical microscopic image. It is also very interesting to develop an ANN to recognize a

complex object in an image. A related issue is how to distinguish the object from its environment (*i.e.*, how to determine the boundary of the object?) Or how to derive a criterion to guide a computer to determine the boundary of an object automatically?

References

[1] E. M. Hempe, P. C. J., A. Podlipensky, "Measuring ultrashort laser pulses Intensity Autocorrelation", project for the "Elitestudiengang Physik Regensburg-Erlangen", (2006).

[2] D. J. Kane, R. Trebino, "Characterization of Arbitrary Femtosecond Pulses Using Frequency-Resolved Optical Gating", IEEE Journal of Quantum Electronic, Vol.29, No.2, P.571 (1993).

[3] T. Tanabe, H. Tanabe, Y. Teramura, and F. Kannari, "Spatiotemporal measurements based on spatial spectral interferometry for ultrashort optical pulses shaped by a Fourier pulse shaper", Opt. Soc, Vol.19, No.11, P.2795, (2002).

[4] M. R. Brady, "Subpixel Resolution Schemes for Multiphase Flows", Virginia Polytechnic Institute and State University, Master of Science, (2006).

[5] 蘇木春, 張孝德, "機器學習:類神經網路、模糊系統以及基因演算法則", 全華科技圖書股份有限公司, (2006).

[6] V. V. Zharkova, V. Schetinin, "A Neural-Network Technique for Recognition of Filaments in Solar Images", Lecture Notes in Computer Science, Vol.2773, P.148, (2003).

[7] H. A. Rowley, S. Baluja, T. Kanade, "Neural Network-Based Face Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.20, No.1, P.23, (1998)

[8] J. Korczak, P. Roger, "Stock timing using genetic algorithms", Applied Stochastic Models in Business and Industry, Vol.18, Issue.2, P.121, (2002).

[9] W. S. McCulloch,W. H. Pitts, "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics, Vol. 5, P. 115, (1943)

[10] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain", Psychological Review, Vol.65, No.6, P.386, (1958).

[11] Marvin Minsky, Seymour Papert, "Perceptrons: An Introduction to Computational Geometry", MIT Press, (1969).

[12] T. Kohonen, "Self-organized formation of topologically correct feature maps", Biological Cybernetics, Vol.43, P.59, (1982).

[13] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", Proc. Natl. Acad. Sci. USA, Vol.79, P.2554, (1982) .

[14] D. E. Rumelhart, J. L. McClelland, "Parallel Distributed Processing", MIT Press, (1986).

[15] G. A. Carpenter,S. Grossberg, D. B. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", Neural Networks, Vol.4, Issue.6, P.759, (1991).

[16] L.-N. Hossein, M. Nasiruddin, T. Samad, "Effect of Initial Weights on Back-Propagation and It’s Variations", Systems, Man and Cybernetics, IEEE International Conference, Vol.1, P.218, (1989).

[17] J. K. Kruschke, J. R. Movellan, "Benefits of Gain: Speed Learning and Minimal Hidden Layers in Back-propagation Networks", IEEE Transactions on Systems, Man, and Cybernetics, Vol.21, No.1, P.273, (1991).

[18] Y. M. Pirez, D. Sarkar, "Back Propagation Algorithm with Controlled Oscillation of Weights", Neural Networks: IEEE International Conference, Vol.1, P.21, (1993).

[19] I. Ivanova, M. Kubat, "Initialization of neural networks by means of decision trees", Knowledge-Based Systems, Vol.8, No.6, P.333, (1995).

[20] M. Zhao, Y. Wu, X. Ding, "Speeding up the training process of the MFNN by optimizing the hidden layers' outputs", Neurocomputing, Vol.11, Issue.1, P.89, (1996).

[21] S. C. Ng, S. H. Leung, A. Luk, "The Generalized Back-Propagation Algorithm with Convergence Analysis", Proceedings of the IEEE International Symposium on Circuits and Systems, Vol. 5, P.612, (1999).

[22] C. Dorrer, E. M. Kosik, I. A. Walmsley, "Direct space–time characterization of the electric fields of ultrashort optical pulses", Opt. Lett., Vol.27 No.7, P.548, (2002).

[23] A. M. Weiner, J. P. Heritage, and J. A. Salehi, "Encoding and decoding of femtosecond pulses", Opt. Lett., Vol.13, No.4, P.300, (1988).

[24] R. K. Shelton, L. S. Ma, H. C. Kapteyn, M. M. Murnane, J. L. Hall, Jun Ye "Active synchronization and carrier phase locking of two separate mode-locked femtosecond lasers", J. Modern Optics, Vol.49, Issue3, P.401, (2002).

[25] A. H. Zewail, "Femtochemistry, Recent progess in studies of dynamics and control of reactions and their transition states", J. Phys. Chem., Vol.100, P.12701, (1996).

[26] M. C. Chen, J. Y. Huang, L. J. Chen, "Coherent control multiphoton processes in semiconductor saturable Bragg reflector with freezing phase algorithm", Appl. Phys. B, Vol.80, No.3, P.333, (2005).

[27] R.S. Judson, H. Rabitz, "Teaching lasers to control molecules", Phys. Rev. Lett., Vol.68, Issue 10, P.1500, (1992).

[28] N. Dudovich, B. Orion, Y. Silberberg, "Single-pulse coherently controlled nonlinear Raman spectroscopy and microscopy", Nature, Vol.418, P.512, (2002).

[29] T. Brixner, N. H. Damrauer, P. Niklaus, G. Gerber, "Photoselective adaptive femtosecond quantum control in the liquid phase", Nature, Vol.414, P.57 (2001).

[30] I. Pastirk, J. M. Dela Cruz, K.A. Walowicz, V. V. Lozovoy, M. Dantus, "Selective two-photon microscopy with shaped femtosecond pulses", Opt. Express, Vol.11, Issue.14, P.1695 (2003).

[31] D. Yelin, D. Meshulach, Y. Silberberg, "Adaptive femtosecond pulse compression",

Opt. Lett. Vol.22, P.1793, (1997).

[32] D. Meshulach, D. Yelin, Y. Silberberg, "Adaptive real-time femtosecond pulse shaping", J. Opt. Soc. Am. B, Vol.15, P.1615, (1998).

[33] T. Baumert, T. Brixner, V. Seyfried, M. Strehle, G. Gerber, "Femtosecond polarization pulse shaping", Appl. Phys. B, Vol.65, P.779 (1997).

[34] E. Zeek, R. Bartels, M.M.Murnane, H. C. Kapteyn, S. Backus, "Adaptive pulse compression for transform-limited 15-fs high-energy pulse generation", Opt. Lett., Vol.25, P.587, (2000).

[35] A. Efimov, M.D. Moores, N.M. Beach, J.L. Krause, D.H. Reitze, "Adaptive control of pulse phase in a chirped-pulse amplifier", Opt. Lett., Vol.23, P.1915, (1998).

[36] J. Kunde, B. Baumann, S. Arlt, F. Morier-Genoud, U. Siegner, U. Keller, "Adaptive feedback control of ultrafast semiconductor nonlinearities", Appl. Phys. Lett., Vol.77, P.924, (2000).

[37] A. YILDIZ, P. R. SELVIN, "Fluorescence Imaging with One Nanometer Accuracy: Application to Molecular Motors", Acc. Chem. Res, Vol.38, No.7, P.574, (2005).

[38] M. J. Rust, M. Bates, X. Zhuang, "Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM)", Nature Methods Advance Online Publication, Vol.3, P.793, (2006).

[39] T. J. Gould, V. V. Verkhusha, S. T. Hess, "Imaging biological structures with fluorescence photoactivation localization microscopy", Nature Protocols, Vol.4, No.3, P.291, (2009).

[40] R. D. Kaufman, M. D., "Biophysical Mechanisms of Anesthetic Action: Historical Perspective and Review of Current Concepts", Anesthesiology, Vol.46, Issue.1, P.49, (1977).

[41] M. Smereka, I. Duleba, "CIRCULAR OBJECT DETECTION USING A MODIFIED HOUGH TRANSFORM", Int. J. Appl. Math. Comput. Sci., Vol.18, No.1, P.85, (2008).

[42] J. P. Fillard, H. M'timet, J. M. Lussert, M. Castagne , "Computer simulation of super-resolution point source image detection", Optical Engineering, Vol.32, No.11, P.2936, (1993).

[43] B. F. Alexander, K. C. Ng, Optical Engineering, Vol.30, No.9, P.1320, (1991).

[44] W.-K. Kuo, D.-T. Tang, "Laser spot location method for a laser pointer interaction application using a diffractive optical element", Optics & Laser Technology, Vol.39, P.1288, (2007).

[45] S. B. Andersson, "Localization of a fluorescent source without numerical fitting", Optics Express, Vol.16, No.23, P.18714, (2008).

[46] X. Qu, D. Wu, L. Mets, N. F. Scherer, "Nanometer-localized multiple single-molecule fluorescence microscopy", PNAS (Biophysics), Vol.101 No.31, P.

11298 (2004).

[47] J. W. Cui, J. B. Tan, L. Ao, W. J. Kang, "Optimized algorithm of laser spot center location in strong noise", , 7th International Symposium on Measurement Technology and Intelligent Instruments, Journal of Physics, Conference Series 13, P.312, (2005).

[48] S. Arturus, "New Algorithm of protein spot center detection in two dimensional electrophoresis gel images", IX international PhD workshop, P.61, (2007).

[49] J. P. Fillard, "Subpixel Accuracy Location Estimation from Digital Signals", Optical Engineering, Vol.31 No.11, P.2465, (1992).

[50] J. Enderlein, "Tracking of fluorescent molecules diffusing within membranes", Appl. Phys. B, Vol.71, P.773, (2000).

[51] T. Basche, W. E. Moerner, M. Orrit, U. P. Wild, "Single-Molecule Optical Detection, Imaging and Spectroscopy", Weinheim, 1997.

Appendix I: Technical Details of BP ANN

To help understanding the implementation of a back-propagation artificial neuron network (BP ANN) reported in this thesis, some useful information is presented in this Appendix.

From the configuration of BP ANN shown in Fig. 2.9, the output from *j*th hidden node can be expressed as

$$H_j = F(\sum_{i=1}^{n} X_i W_{ij} + P_j) = F(net_j) \quad . \tag{2-4}$$

The output of the *k*th output node becomes

$$Y_k = F(\sum_{j=1}^{m} H_j W_{jk} + P_k) = F(net_k) \quad . \tag{2-5}$$

The deviation of the real output *Y* from the desired output *D* can be found

$$E = \frac{1}{2}\sum_{k=1}^{l}(D_k - Y_k)^2 \ . \tag{2-6}$$

We update the weighting parameters of BP ANN by using the gradient descent method to minimize *E*, yielding

$$W_{jk}(t+1) = W_{jk}(t) - \eta \frac{\partial E}{\partial W_{jk}} , \tag{2-7}$$

$$P_k(t+1) = P_k(t) - \eta \frac{\partial E}{\partial P_k} . \tag{2-8}$$

To estimate $\frac{\partial E}{\partial W_{jk}}$, we invoke the chain rule and in

Case 1: the weighting parameters between the hidden and output layer are found to be

$$\frac{\partial E}{\partial W_{jk}} = \frac{\partial E}{\partial Y_k}\frac{\partial Y_k}{\partial net_k}\frac{\partial net_k}{\partial W_{jk}} = -(D_k - Y_k) * F^{-1}(net_k) * H_j \quad .$$

In **Case 2**: we can derive the weighting parameters between the input and hidden layer as

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial H_j}\frac{\partial H_j}{\partial net_j}\frac{\partial net_j}{\partial W_{ij}} = \frac{\partial E}{\partial H_j} * F^{-1}(net_j) * X_i \text{ , where}$$

$$\frac{\partial E}{\partial H_j} = \sum_k \frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial H_j} = \sum_k \frac{\partial E}{\partial Y_k}\frac{\partial Y_k}{\partial net_k} W_{jk} = \sum_k -(D_k - Y_k) * F^{-1}(net_k) W_{jk} \ .$$

Define $\delta_k = (D_k - Y_k)\ F^{-1}(net_k)$ and $\delta_j = F^{-1}(net_j)\ \sum_k \delta_k W_{jk}$, we obtain

Eq. (2-9) and Eq. (2-11)

$$W_{jk}(t+1) = W_{jk}(t) + \eta\delta_k H_j \ , \qquad (2\text{-}9)$$

$$W_{ij}(t+1) = W_{ij}(t) + \eta\delta_j X_i \ \ . \qquad (2\text{-}11)$$

Similar procedure can be employed to obtain Eq. (2-10) and Eq. (2-12) by simply replacing $\frac{\partial net_k}{\partial W_{jk}}$ and $\frac{\partial net_j}{\partial W_{ij}}$ with $\frac{\partial net_k}{\partial P_k}$ and $\frac{\partial net_j}{\partial P_j}$.

Since both of the derivatives are equal to one, therefore

$$P_k(t+1) = P_k(t) + \eta\delta_k \ , \qquad (2\text{-}10)$$

$$P_j(t+1) = P_j(t) + \eta\delta_j \ . \qquad (2\text{-}12)$$

During the training stage, Eq. (2-4) and Eq. (2-5) are used to obtain the output of BPANN. Eqs. (2-9) to (2-12) are then employed to update the weighting parameters of the BP ANN until the error *E* is small enough or the limit number of training loop is arrived. During the application stage of BP ANN, only Eq. (2-4) and Eq. (2-5) are used.

The detailed steps are summarized in the following:

1. Initialize the weighting parameters *W*s and *P*s of BP ANN with random numbers.
2. Input a training data set *X* and *D* and invoke Eq. (2-4) and Eq. (2-5) to derive *Y*.
3. Calculate the value of error with Eq. (2-6). If the value is small enough stop the

training process, otherwise continue to the next step.

4. Calculate the error for the output layer δ_k.

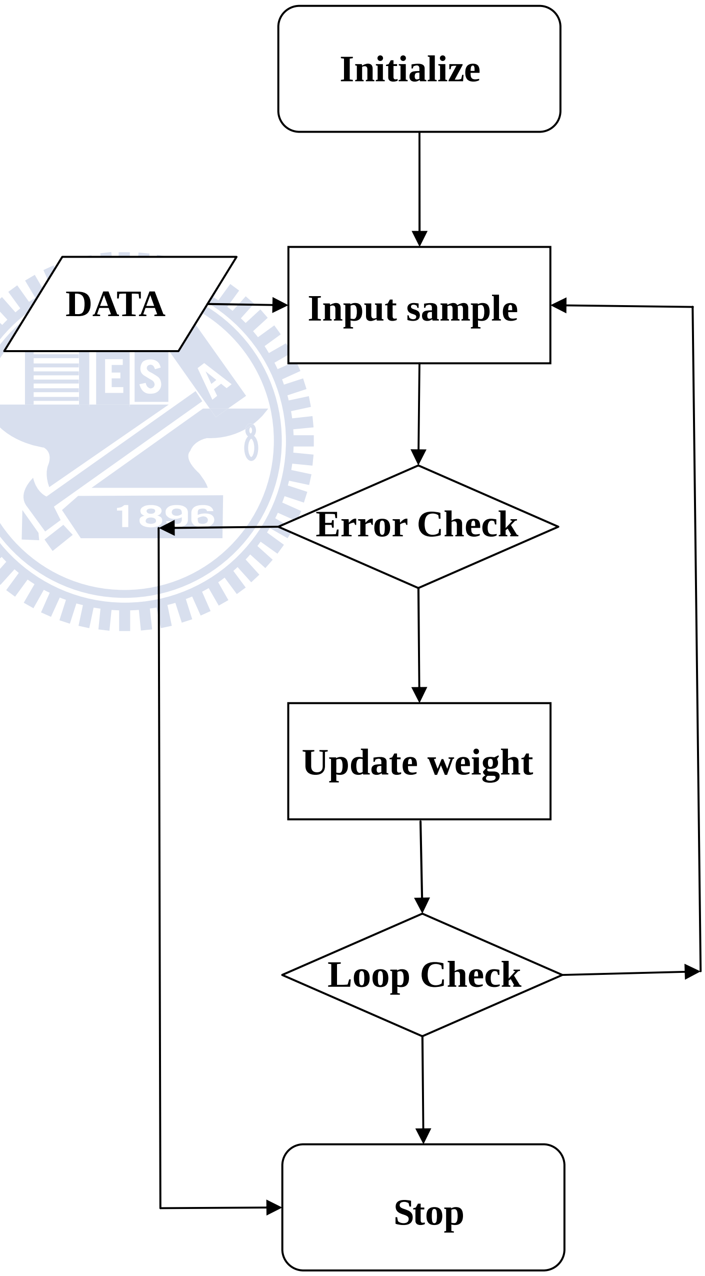
5. Update the weighting parameters of the BP ANN for W_{jk} and P_k with Eq. (2-9) and Eq. (2-10).

6. Calculate the error for the hidden layer δ_j.

7. Update the weighting parameters of the BP ANN for W_{jk} and P_k with Eq. (2-11) and Eq. (2-12).

8. Go to step 2 for next training data set or stop the training process if the limit number of training loop is arrived.

The Flow Chart of the Training Process of a BP ANN



Appendix II: The Code for Preparing the Data Used in This Thesis

To retrieve the phase information of a coherent optical pulse from its SHG spectral

Intensity, we follow the following procedure:

Step1: Generate a Gaussian spectral profile with the code of

```
for w=1:64
     A(w)=exp(-((w-32.5)/15)^2);
end
```

Step2: Prepare the polynomial form of phase with

```
mask=[1 1 1 1 1;        1 1 1 1 -1;       1 1 1 -1 1;       1 1 -1 1 1;
        1 -1 1 1 1;       -1 1 1 1 1;       1 1 1 -1 -1;       1 1 -1 1 -1;
        1 -1 1 1 -1;       -1 1 1 1 -1;       1 1 -1 -1 1;       1 -1 1 -1 1;
        -1 1 1 -1 1;       1 -1 -1 1 1;       -1 1 -1 1 1;       -1 -1 1 1 1;
        -1 -1 -1 -1 -1;     -1 -1 -1 -1 1;      -1 -1 -1 1 -1;      -1 -1 1 -1 -1;
        -1 1 -1 -1 -1;      1 -1 -1 -1 -1;      -1 -1 -1 1 1;       -1 -1 1 -1 1;
        -1 1 -1 -1 1;       1 -1 -1 -1 1;       -1 -1 -1 1 1;       -1 -1 1 -1 1;
        -1 1 -1 -1 1;       1 -1 -1 -1 1;       -1 1 1 -1 -1;       1 -1 1 -1 -1; ];
row=fix(rand(1)*32+1);
paramater=10*mask(row,:).*rand(1,5);
Phase=zeros(1,64);
for i=1:64
     x=(i-32.5)/32;
     Phase(i)=paramater(1)*x^2+paramater(2)*x^3+paramater(3)*x^4 ...
               +paramater(4)*x^5+paramater(5)*x^6;
end
```

Step3: Add the phase some Legendre polynomials and yield the associated SHG spectral intensity with

```
PhaseAdd=[
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
6.2832    5.6943    5.1244    4.5735    4.0416    3.5287    3.0347    2.5598    2.1039
     1.667     1.249     0.85011   0.47017   0.10923   -0.23271  -0.55566  -0.8596
     -1.1446   -1.4105   -1.6575   -1.8854   -2.0944   -2.2844   -2.4553   -2.6073
     -2.7403   -2.8543   -2.9493   -3.0252   -3.0822   -3.1202   -3.1392   -3.1392
     -3.1202   -3.0822   -3.0252   -2.9493   -2.8543   -2.7403   -2.6073   -2.4553
     -2.2844   -2.0944   -1.8854   -1.6575   -1.4105   -1.1446   -0.8596   -0.55566
     -0.23271  0.10923   0.47017   0.85011   1.249     1.667     2.1039    2.5598
     3.0347    3.5287    4.0416    4.5735    5.1244    5.6943    6.2832;
-6.2832   -5.1334   -4.0755   -3.1067   -2.2237   -1.4237   -0.70356  -0.060332
     0.50902   1.0075    1.4382    1.804     2.108     2.3532    2.5426    2.6792
     2.7661    2.8063    2.8027    2.7584    2.6765    2.5598    2.4115    2.2346
     2.032     1.8068    1.562     1.3006    1.0257    0.74015   0.4471    0.14954
     -0.14954  -0.4471   -0.74015  -1.0257   -1.3006   -1.562    -1.8068   -2.032
     -2.2346   -2.4115   -2.5598   -2.6765   -2.7584   -2.8027   -2.8063   -2.7661
     -2.6792   -2.5426   -2.3532   -2.108    -1.804    -1.4382   -1.0075   -0.50902
     0.060332  0.70356   1.4237    2.2237    3.1067    4.0755    5.1334    6.2832;
6.2832    4.4275    2.8361    1.4888    0.36615   -0.55052  -1.2793   -1.8377   -2.2425
     -2.5096   -2.6546   -2.6921   -2.6363   -2.5005   -2.2974   -2.0392   -1.7372
     -1.4022   -1.0441   -0.67247  -0.29589  0.07757   0.44053   0.78629   1.1088
     1.4028    1.6634    1.8868    2.0695    2.2089    2.3029    2.3503    2.3503
     2.3029    2.2089    2.0695    1.8868    1.6634    1.4028    1.1088    0.78629
     0.44053   0.07757   -0.29589  -0.67247  -1.0441   -1.4022   -1.7372   -2.0392
     -2.2974   -2.5005   -2.6363   -2.6921   -2.6546   -2.5096   -2.2425   -1.8377
     -1.2793   -0.55052  0.36615   1.4888    2.8361    4.4275    6.2832;
-6.2832   -3.6098   -1.5204   0.06077   1.2036    1.9726    2.427     2.6211    2.6041
     2.4206    2.1108    1.7104    1.2512    0.7608    0.26336   -0.22069  -0.6742
     -1.0832   -1.4367   -1.7264   -1.9467   -2.0944   -2.1684   -2.1696   -2.1008
     -1.9665   -1.7724   -1.5257   -1.2344   -0.90767  -0.55507  -0.18678  0.18678
     0.55507   0.90767   1.2344    1.5257    1.7724    1.9665    2.1008    2.1696
     2.1684    2.0944    1.9467    1.7264    1.4367    1.0832    0.6742    0.22069
     -0.26336  -0.7608   -1.2512   -1.7104   -2.1108   -2.4206   -2.6041   -2.6211
     -2.427    -1.9726   -1.2036   -0.06077  1.5204    3.6098    6.2832;
6.2832    2.7183    0.24698   -1.3414   -2.2315   -2.5836   -2.5359   -2.2061   -1.693
     -1.0785   -0.42916  0.20267   0.7769    1.2645    1.6463    1.9113    2.0559
     2.0826    1.9989    1.8164    1.5495    1.2153    0.8318    0.41806   -0.0070076
     -0.42505  -0.81884  -1.1727   -1.4731   -1.7087   -1.8706   -1.9531   -1.9531
     -1.8706   -1.7087   -1.4731   -1.1727   -0.81884  -0.42505  -0.0070076     0.41806
```

```
    0.8318   1.2153   1.5495   1.8164    1.9989    2.0826   2.0559   1.9113
    1.6463   1.2645   0.7769   0.20267  -0.42916  -1.0785  -1.693   -2.2061
   -2.5359  -2.5836  -2.2315  -1.3414    0.24698   2.7183   6.2832
];
SHGI=zeros(1,127*6);
for phasecount=1:6
    PhaseMod=Phase+PhaseAdd(phasecount,:);
    SHGI(127*phasecount-126:127*phasecount)=SHG(A,PhaseMod);
end
```

Function SHG()

```
function ShgI=Shg(Ea,Ep)
Ew=Ea.*exp(i.*Ep);
ShgE=conv(Ew,Ew);
ShgI=ShgE.*conj(ShgE);
```

Finally, the training data for ANN are prepared with a pair of input *SHGI(1,127*6)* and output *Phase(1,64).*

To localize the central position of an image of light spot with 11×11 pixels,

Step1: Generate an image of light spot with random central position by

```
Size=11;
X0=rand(1,1)*Size;
Y0=rand(1,1)*Size;
```

Step2: Prepare the image of 11*11 pixels with a 2D Gaussian profile

```
for X=1:Size
    for Y=1:Size
        G(X,Y)=exp(-((X-X0)^2+(Y-Y0)^2)/3^2)+exp(-((X-X1)^2+(Y-Y1)^2)/3^2);
    end
end
```

Step3: Add random noise to the image

```
noiselevel = (a value from 0 to 0.5)
for X=1:Size
        for Y=1:Size
                Gr(X,Y)=G(X,Y)+rand(1,1)* noiselevel- noiselevel/2;
        end
end
```

Finally, the training data for ANN are prepared with a pair of input *Gr(11,11)* and output *X0,Y0.*