STUDY OF NEURAL NETWORK FOR FLY HEIGHT FAILURE PATTERN CLASSIFICATION IN HARD DISK DRIVE

THANAPONG THANASARN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN DATA STORAGE TECHNOLOGY
INTERNATIONAL COLLEGE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2014

KMITL-2014-IC-M-005-001

STUDY OF NEURAL NETWORK FOR FLY HEIGHT FAILURE PATTERN CLASSIFICATION IN HARD DISK DRIVE

THANAPONG THANASARN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN DATA STORAGE TECHNOLOGY
INTERNATIONAL COLLEGE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2014
KMITL-2014

หัวข้อวิทยานิพนธ์	การศึกษาโครงข่ายประสาทเทียมเพื่อการคัดแยกรูปแบบความเสียหายที่เกิดจากการหาระยะบินในฮาร์ดดิสก์ไดร์ฟ
นักศึกษา	นาย ธนพงศ์ ธนะศาล
รหัสนักศึกษา	53600622
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีการบันทึกข้อมูล
พ.ศ.	2557
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ. ดร. ชานนท์ วริสาร

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้เป็นการศึกษาโครงข่ายประสาทเทียม ซึ่งประกอบไปด้วยแบบจำลอง 3 ชนิด คือ การเรียนรู้เวกเตอร์ควอนไทเซชัน (Learning Vector Quantization: LVQ), การเรียนรู้แบบแพร่กลับ (Back Propagation: BP) และโครงข่ายประสาทเทียมแบบความน่าจะเป็น (Probabilistic neural network: PNN) เพื่อการคัดแยกรูปแบบของความเสียหายที่เกิดจากการหาระยะห่างระหว่างหัวอ่านเขียนกับแผ่นดิสก์ในกระบวนการผลิตฮาร์ดดิสก์ไดร์ฟ

โดยในการทดลองได้นำเอาข้อมูลความเสียหายที่เกิดจากการวัดระยะห่างระหว่างหัวอ่านเขียนกับแผ่นดิสก์ทั้งหมด 6 รูปแบบเพื่อใช้ในการจดจำรูปแบบความเสียหายในโครงข่ายประสาทเทียมซึ่งรูปแบบของความเสียหายทั้ง 6 ชนิดนี้เกิดจาก 3 สาเหตุ คือ ความเสียหายทางกายภาพของชุดหัวอ่านเขียน (Head Gimbal Assembly: HGA) ความสกปรกบนหน้า slider Air Bearing Surface (ABS) และความไม่เสถียรของชั้นแม่เหล็กในหัวอ่านเขียน

ผลการทดลองพบว่าความสามารถในการคัดแยกรูปแบบความเสียหายของแบบจำลองโครงข่ายประสาทเทียมแบบความน่าจะเป็นสามารถคัดแยกได้ดีกว่าแบบจำลองการเรียนรู้เวกเตอร์-ควอนไทเซชันและแบบจำลองการเรียนรู้แบบแพร่กลับ โดยที่แบบจำลองโครงข่ายประสาทเทียมแบบความน่าจะเป็นมีความถูกต้องในการคัดแยกสูงถึง 96.2% ในขณะที่แบบจำลองการเรียนรู้เวกเตอร์ควอนไทเซชันสามารถคัดแยกได้ 90.7% และแบบจำลองการเรียนรู้แบบแพร่กลับมีความความถูกต้องในการคัดแยกเท่ากับ 81.8%

Thesis	Study of Neural Network for Fly Height Failure Pattern Classification in Hard Disk Drive
Student	Mr. Thanapong Thanasarn
Student ID.	53600622
Degree	Master of Engineering
Program	Data Storage Technology
Year	2014
Thesis Advisor	Asst.Prof.Dr. Chanon Warisarn

ABSTRACT

For this thesis, three Neural Network (NN) models: Learning Vector Quantization (LVQ), Back Propagation (BP), and Probabilistic neural network (PNN) algorithms were studied and used for Fly Height (FH) failure pattern classification in Hard Disk Drive (HDD) manufacturing process. Six types of failure patterns from FH measurement were recognized, including three causes of failure: physical damage on Head Gimbal Assembly (HGA), contamination on the slider Air Bearing Surface (ABS) and head instability.

Experimental results showed that the classification of performance of the FH failure pattern based on the PNN model was better than the LVQ and BP models. The PNN model achieved the highest accuracy at 96.2%, with LVQ at 90.7%, and BP the worst with 81.8%.

ACKNOWLEDGEMENT

This thesis has been supported by co-scholarship, Seagate Technology (Thailand) Ltd., National Science and Technology Development Agency (NSTDA) and King Mongkut's Institute Technology Ladkrabang (KMITL). I would like to acknowledge with heartfelt thanks to Asst.Prof.Dr. Chanon Warisarn for your kind assistance and guidance for my research. This accomplishment would not have been possible without the encouragement from my family. Thank you my life.

Thanapong Thanasarn

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continue)

Page

TABLE OF CONTENTS (Continue)

Page

LIST OF FIGURES

LIST OF FIGURES (Continue)

LIST OF TABLES

LIST OF TABLES (Continue)

CHAPTER 1

INTRODUCTION

1.1 Background

After assembly, a hard disk drive (HDD) requires functional testing for drive operations, such as the calculation of areal density with Bit Per Inch (BPI)/Track Per Inch (TPI), the measurement of Bit Error Rate (BER) performance, and the test focused on in this study, the Fly Height (FH) measurement.

FH testing is the measurement of the distance spacing between a recording head and a disk by applying power/voltage values into a heater element of the recording head until the recording head contacts the lubricant of the disk and also provides the heater power in each data zone with appropriate value to perform read and write operations. In general, the failure analysis on failed heads from FH measurement must be analyzed using the FH profile to classify the causes of failure. It was observed that the FH profile indicated the cause of each problem. For example, physical damage on the Head Gimbal Assembly (HGA), contamination on the slider Air Bearing Surface (ABS), or head instability problem.

Currently, FH profile failure is investigated by engineer takes a long time period. Therefore, methods to reduce the time required for this using failure analysis, are necessary for the HDD manufacturing process to become more efficient. Recently, many researchers have applied neural network (NN) models to generate pattern recognition in several fields. This thesis applies three classification methods based on Learning Vector Quantization (LVQ), Back Propagation (BP), and Probabilistic neural network (PNN) to speed up the time required for failure analysis.

1.2 Objectives

To apply Neural Network models for FH failure pattern classification in the manufacturing process.

1.3 Scope of work

1) To study three models of neural network, Learning Vector Quantization (LVQ), Back Propagation (BP), and Probabilistic neural network (PNN).

2) To use failure samples from manufacturing database to simulate with computer programming.

1.4 Benefits

1) To develop an automatic analysis tool instead of waiting for results from manual analysis.

2) To minimize the time required to verify failure pattern.

1.5 Research outline

This thesis is organized into five chapters as follows:

1) Introduction: presents the background, objectives, scope of work and benefits of the thesis.

2) Theory: explains the related theory, including the concept of FH measurement in HDD manufacturing, Neural Network architectures and a literature review.

3) Research methodology: presents the description of the research methodology. The first topic is data and sample collection, which has two sub-topics: appearance of FH failure pattern, and failure analysis results of each FH pattern. The

second topic is classification system, which has three sub-topics: data preprocessing, network training, and network testing.

4) Experiment and results: presents the evaluation and performance comparison of FH failure pattern classification for each Neural Network model. The topics include: network verification, comparison results between known and unknown dataset, evaluation number of training iterations, evaluation number of training samples, evaluation number of hidden neurons, evaluation on smoothing parameter, performance comparison of three Neural Network models, and a training time comparison.

5) Conclusions: presents the conclusions and recommendations for the model as suitable to classify FH failure pattern in HDD manufacturing.

CHAPTER 2

THEORY

This chapter first presents the concept of Fly Height measurement in Hard Disk Drive manufacturing, then looks at Neural Network architectures which include three models: Learning Vector Quantization (LVQ), Back Propagation (BP), and Probabilistic neural network (PNN), and ends with a literature review.

2.1 Fly height measurement

Fly Height (FH) measurement in the Hard Disk Drive (HDD) manufacturing process uses Thermal Fly-height Control (TFC) technology to measure the distance spacing between the recording head and the disk by applying power to the heater element of the recording head until it contacts the lubricant of the disk. The objective of FH measurement is to determine the most suitable heater power in each data zone to perform read and write operations. The clearance target requirement is less than three nanometers from the disk, so FH measurement is an important drive testing function, as one of the critical methods for increasing Areal Density Capacity (ADC) is FH reduction. Figure 2.1 presents a schematic diagram of the head–disk interface. In (a) the recording head is included in the slider with the Air Bearing Surface (ABS), while flying on the disk. In (b) the head is shown in side view and cross section, illustrating the change in thermal protrusion when power is applied to the heater.

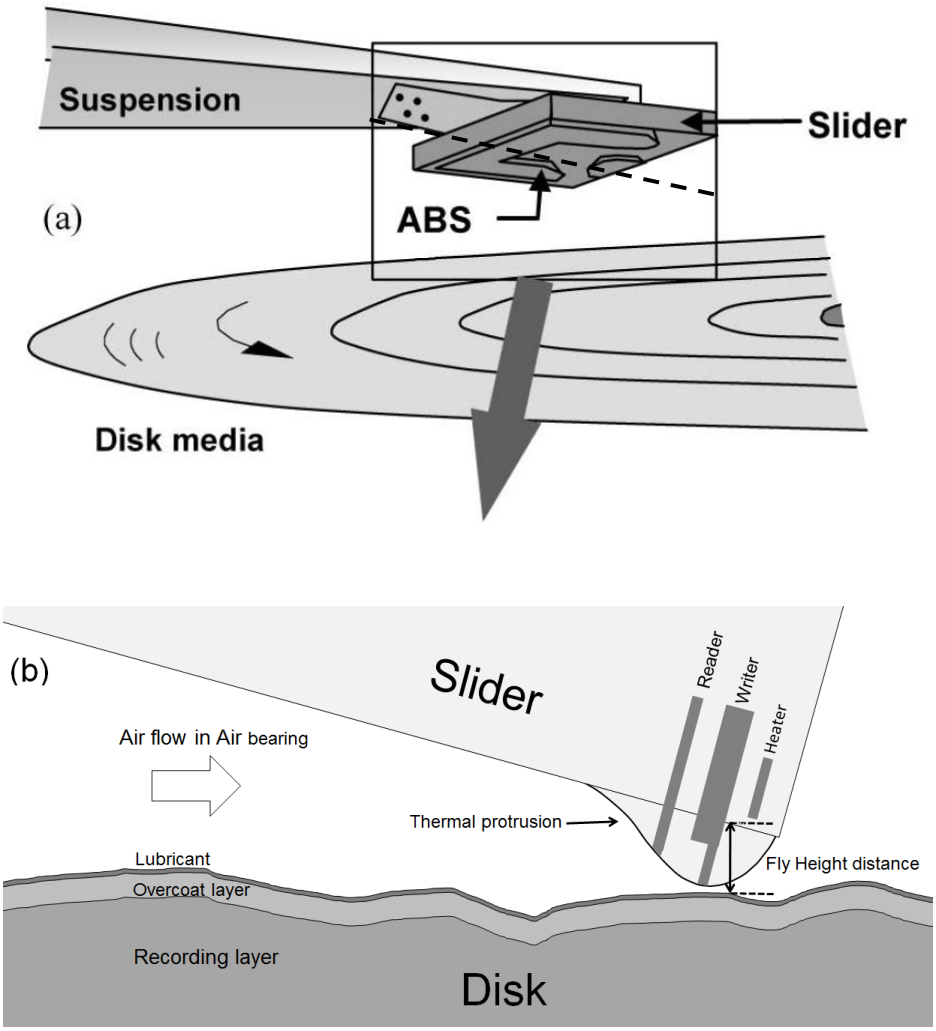


Figure 2.1: Schematic diagram of the head–disk interface

(a) The recording head is included in the slider with ABS, while flying on the disk positioned by its suspension [4].

(b) The cross section of the head and media, illustrates thermal protrusion change when power is applied to the heater.

2.1.1 Touchdown detection

FH measurement requires specific head and data zone testing. Several attributes and designs, the ABS pattern, the mechanical assembly, including the head gimbal assembly (HGA)/suspension and the lubricant type on the disk all contribute to different FH characteristics.

A widely used technique for touchdown detection on the drive level is applying the servo system with a Position Error Signal (PES) [1]-[3]. The servo system controls the position of the recording head, and moves the Head Stack Assembly (HSA) from the current to another desired track. The embedded servo shown in Figure 2.2, has a magnetic pattern (servo pattern) to generate the signal when the reader passes over it. Each of the servo sectors are written by a specific pattern of magnetization, with a unique number for each track. Thus, the embedded servo information identifies which track the recording head is on. A servo sector consists of a DC-gap field, Automatic Gain Control (AGC) field, Servo Timing Mark (STM) field, grey coded track ID field, and PES burst pattern field.

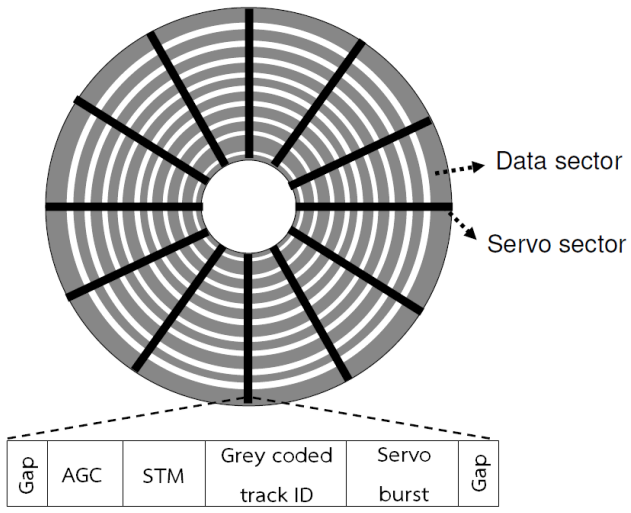


Figure 2.2: Embedded servo on a disk [6]

The PES is generated using utilization from the grey coded track ID field and PES burst pattern field. The grey coded track ID field contains information of the track number, while the PES burst pattern is used to measure the off-track position of the reader from the track center. A simple example for servo burst patterns is shown in Figure 2.3. For a setting using A/B burst patterns, when the reader passes over the burst pattern, a readback signal will be generated [6]. For example, when the reader stays on position number 0 (desired track), the amplitude difference between A burst and B burst equals zero. On the other hand, at the center of position number 1, the readback amplitude from B burst is a negative maximum value. The readback amplitude is proportional to the length of the part of reader when it flies over the burst.

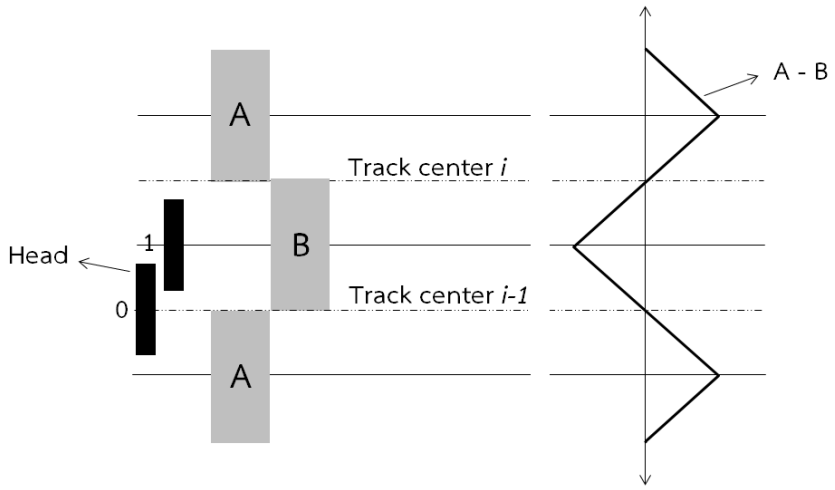


Figure 2.3: Servo patterns with A/B burst to generate off-track error

PES is computed by the ratio of the difference between the signal amplitude coming from A and B bursts as shown Equation (2.1)

$$PES = \frac{A - B}{A + B} \tag{2.1}$$

where *A* and *B* represent the Track Average Amplitude (TAA) of the A and B bursts, respectively [3].

The FH measurement utilizes PES to detect a touchdown incident. When applying heater power, the recording head will be protruded. Every step of the increasing voltage bias to the heater is monitored by PES, using the Fast Fourier Transform (FFT) algorithm. When the recording head touches the lubricant of the disk, this produces a force between the head and the disk that generates an off-track error. The change in PES is detected by the FFT algorithm, and identifies the passive FH distance in Digital to Analog Converter (DAC) units between the recording head and the disk. For the DAC, it is the 8-bit value that preamp controls the voltage of the heater. It can also transform the power in milliwatt unit, using the formula from the preamp vendor's datasheet. However, this thesis considers only data pattern, so DAC from log-file will be not transformed to milliwatt unit, because DAC is directly proportional to milliwatt unit.

2.1.2 Fly height measurement results

In this thesis, the value of touchdown power represents the FH distance. Figure 2.4 shows the FH distance of a recording head, where the y axis is the touchdown heater power value, and the x axis is the data zone of a disk in order of the outer diameter towards the inner diameter, composed of ten data points for each read/write operation. The FH distance for the write operation is always lower than the read operation, because the thermal power from the writer element during the write operation causes more protrusion.

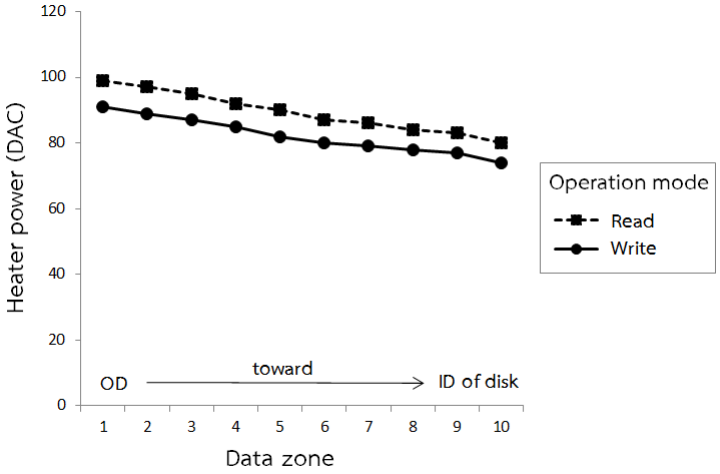


Figure 2.4: FH distance of a recording head

After completing FH measurements, the head-disk spacing change at various levels of heater power was computed from the basics of the Wallace spacing loss equation, as shown in Equation (2.2) [4], [5]

$$\Delta FH = \frac{\lambda}{2\pi} \ln\left(\frac{A_1}{A_2}\right) \quad , \tag{2.2}$$

where ΔFH is the different in FH between the heater power and the reference, λ is the written wavelength, $\ln$ is the natural logarithm, A_1 is the amplitude of the

reference read-back signal, A_2 is the amplitude of the heater power, and A_1 / A_2 is the amplitude ratio of the read-back signal.

Therefore, each step of applied heater power is proportional to the change in head-disk spacing. The result of the relation between Δ FH from the reference (delta FH = 0) on the y axis, versus the various levels of heater power on the x axis is illustrated in Figure 2.5.

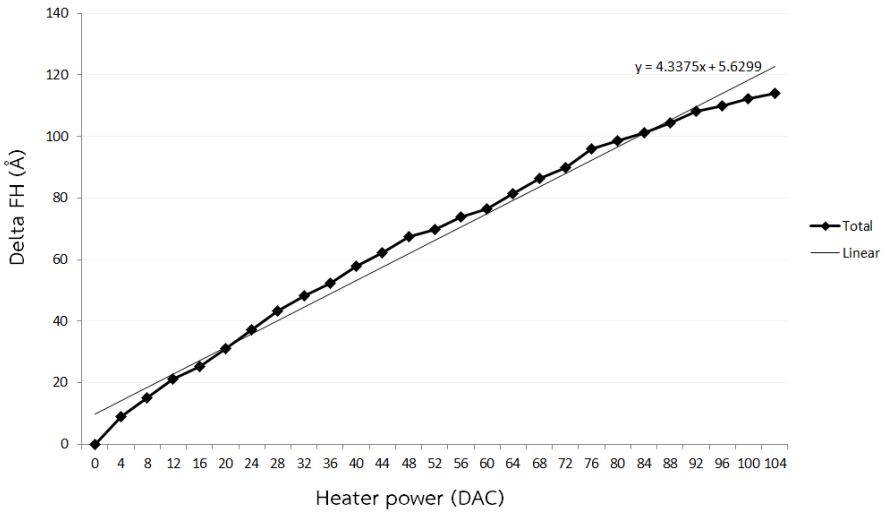


Figure 2.5: Delta FH at various levels of heater power

Finally, the touchdown distance is calculated by computing the active FH (working) distance from the formula in Equation (2.3)

Active FH distance = Touchdown distance – FH Target distance . (2.3)

Where the FH target distance is the appropriate spacing from the disk to the recording head to perform read and write operations. The target distance depends on product design. For example, 20 angstroms from the disk will achieve the areal

density to reach the product's requirement. Figure 2.6 shows a schematic diagram of the passive FH (touchdown) distance, the active FH distance, and the FH target distance.

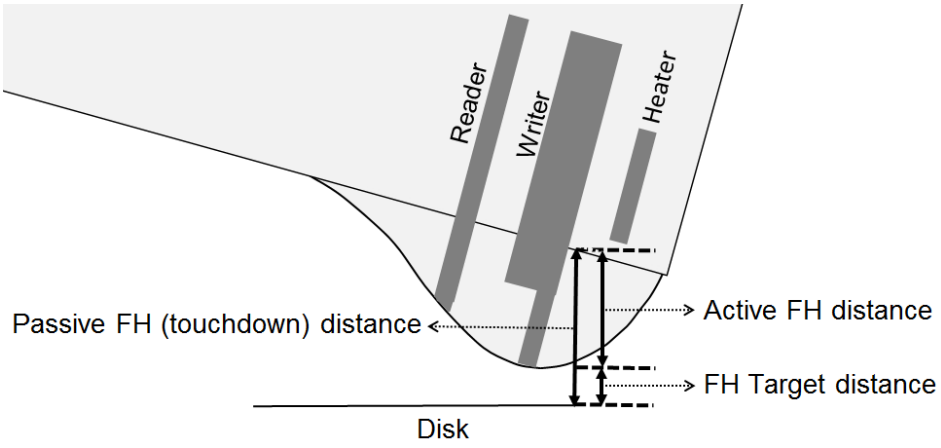


Figure 2.6: Schematic diagram of the passive FH (touchdown) distance, the active FH distance, and the FH target distance

2.2 Neural Network architectures

2.2.1 Introduction of Artificial Neural Network

Artificial Neural Network (ANN) is a mathematical model for an information-processing system [8]. It has an inspiration and desire to duplicate the function of biological neural networks in the nervous system of the human brain. An ANN has the ability to learn and respond like a human brain. McCulloch and Pitts published the first systematic study of ANNs in the 1940s. Since then, there has been continued exploration and development of ANNs. Famous pioneer researchers and their inventions include Hebb (Hebb learning); Rosenblatt, Minsky & Papert (Perceptrons); Widrow & Hoff (ADALINE); and Kohonen (self-organizing feature maps).

Nowadays, neural networks are used in many applications and fields including pattern recognition, classification, prediction, control systems, speech recognition, manufacturing, medicine, business, traffic, and predictions [15]-[25].

2.2.2 Biological Neural Networks

Biological neural networks are the foundation of ANNs. The human nervous system is a very complex neural network, with the brain as the central element, consisting of 10^{10} biological neurons connected to each other through sub-networks.

A biological neuron is an electrically excitable cell that processes and transmits information through electrical and chemical signals. It has three components soma, axon, and dendrites (Figure 2.7). The dendrites receive signals from other neurons. A chemical process creates signals of electric impulses that are transmitted across a synaptic gap. The action of the chemical transmitter modifies the incoming signal, similar to the action of the weights in an ANN.

When sufficient input is received, the cell fires; it transmits a signal over its axon to other cells. It is often supposed that a cell either fires or does not at any

instant of time, so that transmitted signals can be treated as binary. However, the frequency of firing varies and can be viewed as a signal of either greater or lesser magnitude. This corresponds to looking at discrete time steps and totaling all activity (signals received or signals sent) at a particular point in time.

The soma (or cell body) totals all the incoming signals from the dendrites as well as the signals from numerous connections or synapses on its surface. A particular neuron will transmit or fire an impulse signal to its axon if sufficient input signals are received to stimulate the neuron above its threshold activation level. However, if the inputs do not reach the required threshold level, they will quickly decay and not generate any activity.

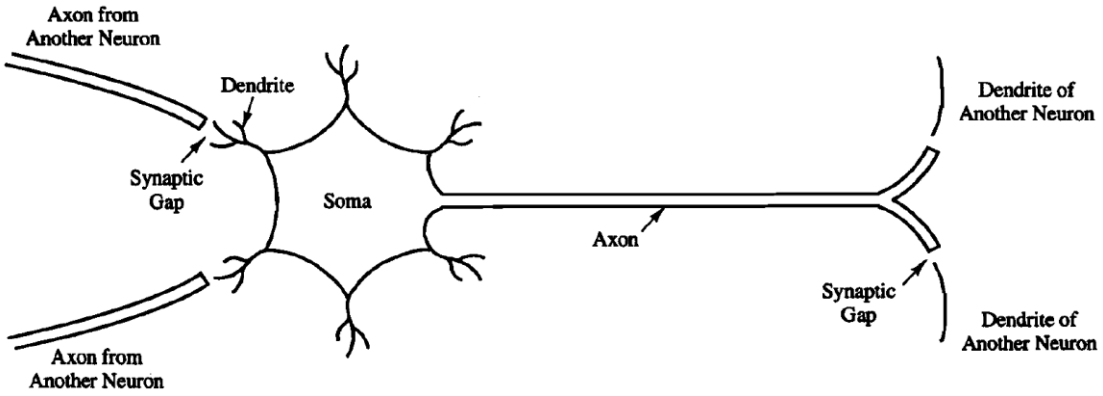


Figure 2.7: Biological neuron [8]

2.2.3 Neural Network classification

A neural network is characterized by three components (1) Architecture (the pattern of connections between the neurons), (2) Learning algorithm (or Training algorithm; the method of determining the weights on the connections), and (3) Activation function (or Transfer function which performs a mathematical operation on the output).

A neural network consists of a large number of simple processing elements called neurons. Each neuron is connected to other neurons for communication, and is also associated by weight. The weights represent information being used by the network to solve a problem. Each neuron has an activation, which is a function of the inputs it has received. Typically, a neuron sends its activation as a signal to several other neurons. A neuron can send only one signal at a time, although that signal is transmitted to several other neurons. ANN can be compared to biological NN, each function that an ANN duplicates from a biological NN is shown in Table 2.1. The weight duplicates synapse, the summation and activation function duplicates the soma, and the output duplicates the signals from the axons.

Table 2.1: Comparison between Biological NN and ANN

Biological NN	ANN
Synapse	Weight
Soma (or cell body)	Summation and Activation function
Signal from Axon	Output

For example, an ANN functions as a simple artificial neuron with a single R-element input vector (Figure 2.8), that receives inputs from neurons p_1, p_1,..., p_R. Then the inputs are multiplied with the weights on the connections from p_1, p_1,...,

p_R to neuron a which are $w_{1,1}$, $w_{1,2}$,..., $w_{1,R}$, together, respectively. The neuron has a bias b, which is summed with the weighted inputs to form the net input n,

$$n = w_{1,1}p_1 + w_{1,2}p_1 + ... + w_{1,R}p_R + b \quad , \tag{2.4}$$

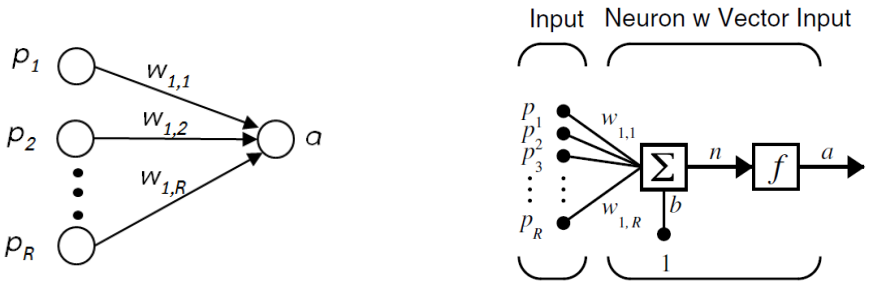
this expression (2.4) can be written as

$$n = Wp + b \quad . \tag{2.5}$$

The activation function net input n, is the sum of the weighted input Wp and the bias b. This sum is the argument of the activation function f.

The output of neuron a can therefore be written as

$$a = f(Wp + b) \quad . \tag{2.6}$$



(a) (b)

Figure 2.8: (a) Basic structure and (b) Network architecture of simple artificial neuron [10]

2.2.3.1 Network architecture

To determine a neural network in terms of network architecture (the arrangement of neurons into layers and the connection patterns within and between layers), the network can be distinguished into two types, single-layer and multi-layer.

1) Single-layer network: A single-layer network has one layer of connection weights. The units can be distinguished as input units (which receive signals from the outside), and output units (which are the response of the network). Figure 2.9, shows a typical single-layer network; the input units are fully connected to the output units, but are not connected to other input units, and the output units are not connected to other output units.

2) Multilayer network: A multilayer network is a network with more layers of nodes (or hidden units) between the input units and the output units. Typically, there is a layer of weights between two adjacent levels of units (input, hidden, or output) as shown in Figure 2.10. Furthermore, most multilayer networks can solve more complicated problems than single-layer networks, but they require more difficult training.

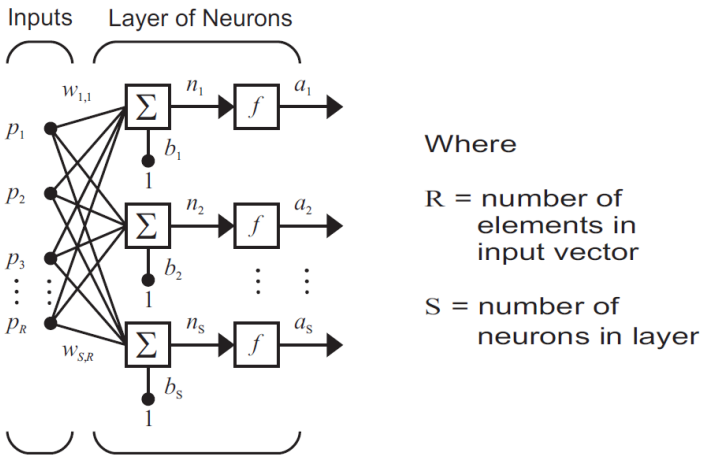


Figure 2.9: A single-layer neural network [10]

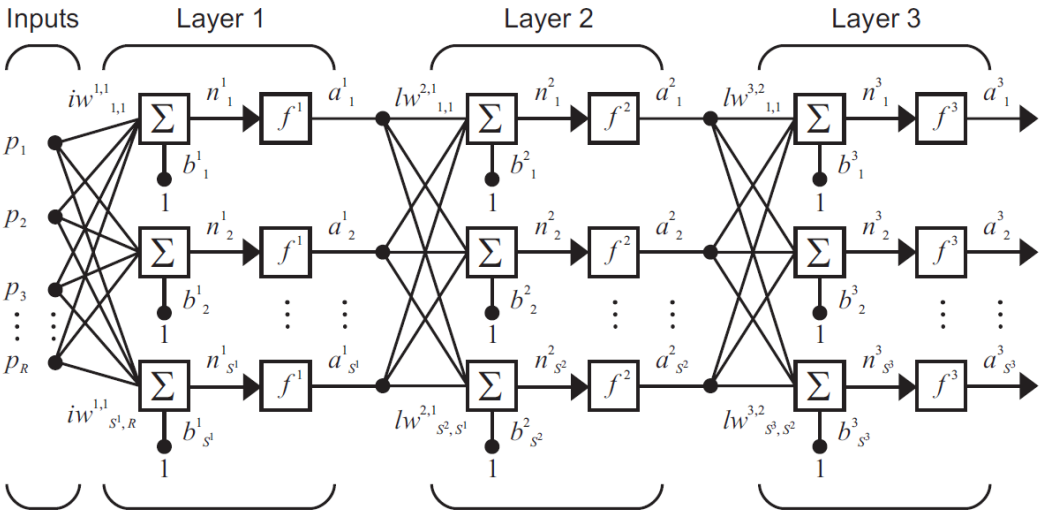


Figure 2.10: A multilayer neural network [10]

2.2.3.2 Learning algorithm

To determine a neural network in terms of a learning algorithm, it can be separated into two types of learning rules, supervised and unsupervised learning.

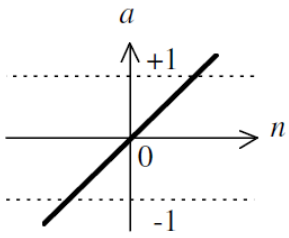
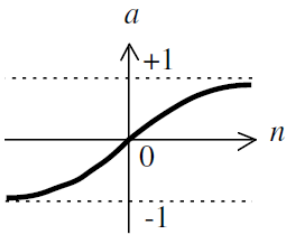
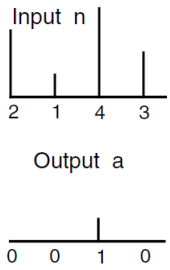
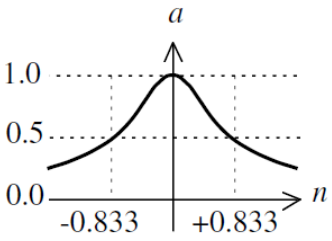
1) Supervised learning: Supervised learning requires a teacher to control the learning rule. The teacher provides the input training set and the requirement target to network. For example, $\{p_1,t_1\},\{p_2,t_2\},...,\{p_k,t_k\}$, where ' p_k ' is an input and ' t_k ' is the corresponding target. After the inputs are applied to the network, the outputs are compared to the targets. The learning rule will adjust the weights of the network, to move the network outputs closer to the targets. Examples of supervised learning are the backpropagation algorithm, and a radial basis function network.

2) Unsupervised learning: Unsupervised learning has no teacher for training. The system must organize itself, and adjust the weights and biases in response to the network inputs only. The network can classify the input patterns into different groups. Sometimes unsupervised learning is referred to as self-organizing learning. An example of unsupervised learning is the Kohonen network.

2.2.3.3 Activation function

An activation function performs a mathematical operation on the output signal. Activation functions can also be utilized depending on the types of problems to be solved by the network. Examples of four types of activation functions used in this thesis are shown in Table 2.2. They are Linear function, Hyperbolic tangent sigmoid, Competitive, and Radial basis function.

Table 2.2: Description of four types of activation functions [10]

Activation function:	Linear	
Function on Matlab:	*a = purelin(n)*	
Icon on Matlab:		
Input/Output relation:	$a = n$	
Activation function:	Hyperbolic tangent sigmoid	
Function on Matlab:	*a = tansig(n)*	
Icon on Matlab:		
Input/Output relation:	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
Activation function:	Competitive	
Function on Matlab:	*a = compet(n)*	
Icon on Matlab:		
Input/Output relation:	***a = 1*** for a maximum of n ***a = 0*** for other n	
Activation function:	Radial basis function	
Function on Matlab:	*a = radbas(n)*	
Icon on Matlab:		
Input/Output relation:	$a = e^{-n^2}$	

2.3 Learning Vector Quantization Neural Network

Learning Vector Quantization (LVQ) is a kind of supervised algorithm for a training network. LVQ was invented by Kohonen in 1990. It consists of three layers of neurons: input layer, competitive layer, and output layer as shown in Figure 2.11. The input layer connects with each neuron in the competitive layer, but the neurons of the competitive layer are only connected with a neuron in the output layer [9].

In the network architecture of LVQ in Figure 2.12, each neuron in the competitive layer is assigned to a class, with several neurons often assigned to the same class. Each class is then assigned to one neuron in the linear layer. Therefore, the number of neurons in the competitive layer (S^1), will always be at least as large as the number of neurons in the linear layer (S^2).

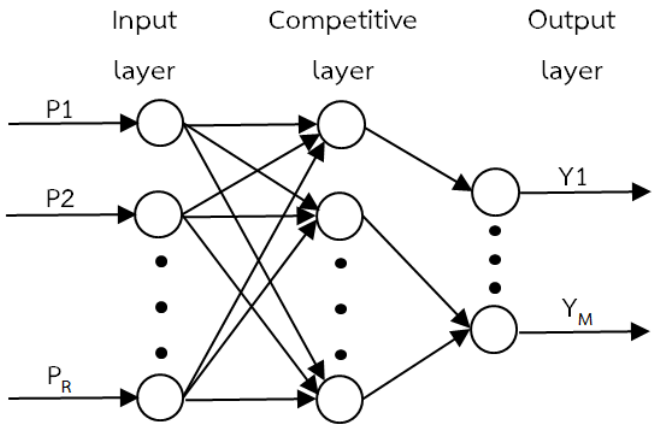


Figure 2.11: Basic structure of LVQ neural network

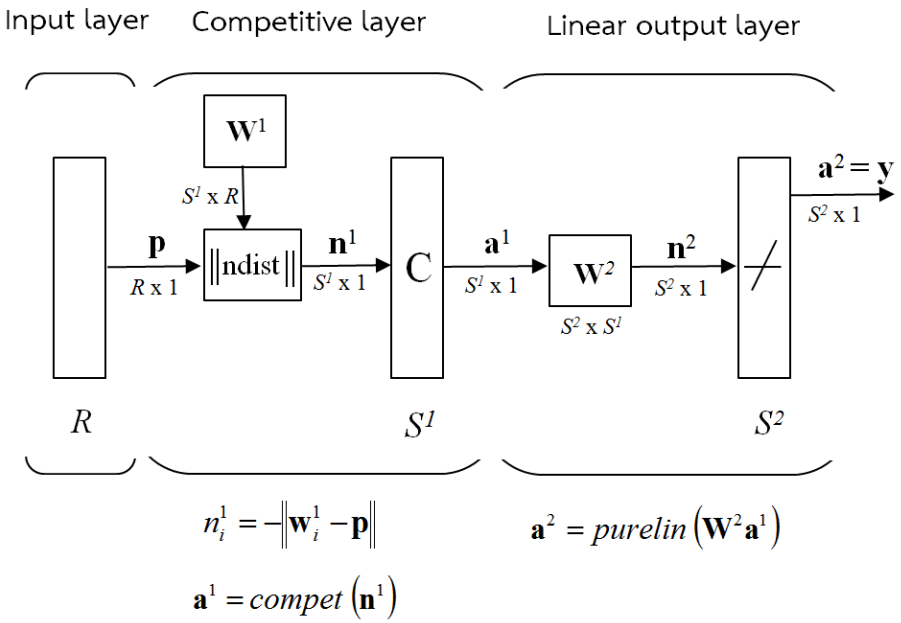


Figure 2.12: The network architecture of LVQ

2.3.1 LVQ algorithm

When an input vector ($\mathbf{p}$) is fed into the competitive network, each neuron learns by using the Euclidean distance equation (Equation (2.7), to find the closest distance between the input vector and any of the weight vectors. The competitive layer weights ($\mathbf{W}^1$) are initially random values.

The *i*-th element of net input of the competitive layer will be

$$n_i^1 = -\sqrt{\sum_{j=1}^{R}\left(w_{i,j}^1 - p_j\right)^2}\ , \tag{2.7}$$

where R is the number of inputs. Equation (2.7) can be written as

$$n_i^1 = -\left\|\mathbf{w}_i^1 - \mathbf{p}\right\|\ . \tag{2.8}$$

The total net input ($\mathbf{n}^1$) can be expressed as

$$\mathbf{n}^1 = -\begin{bmatrix} \left\| w_1^1 - \mathbf{p} \right\| \\ \left\| w_2^1 - \mathbf{p} \right\| \\ \vdots \\ \left\| w_{S^1}^1 - \mathbf{p} \right\| \end{bmatrix} \tag{2.9}$$

where S^1 is the number of neurons in the first layer.

After calculating the net input ($\mathbf{n}^1$), the competitive activation function gives the output value for the winner neuron (i^*) whose weight vector closest to the negative distance with the input vector is a '1', and the other neurons are '0'.

Therefore, the output of the competitive layer is

$$\mathbf{a}^1 = compet\ \left(\mathbf{n}^1\right). \tag{2.10}$$

It follows that the linear output layer transforms the classes of the competitive layer into the specified target categories ($\mathbf{t}$). These are determined by the designer, so the linear layer weights $\mathbf{W}^2$ are set values, and do not change. The $\mathbf{W}^2$ matrix is used to integrate the subclasses into a single class. The columns of $\mathbf{W}^2$ represent the subclasses and the rows represent the classes. Each column of $\mathbf{W}^2$ has a single '1' that indicates that their subclass belongs to the appropriate class, and the other elements are defined as '0'. For example, a $\mathbf{W}^2$ matrix is shown in Figure 2.13.

$$\mathbf{W}^2 = \begin{array}{c} \begin{matrix} \text{Subclass 1} & \text{Subclass 2} & \text{Subclass 3} & \text{Subclass 4} & \text{Subclass 5} & \text{Subclass 6} \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array} \begin{matrix} \text{Class 1} \\ \text{Class 2} \\ \text{Class 3} \end{matrix}$$

Figure 2.13: Example of $\mathbf{W}^2$ matrix

Where:

- subclasses 1 and 3 belong to class 1,
- subclass 4 belongs to class 2, and
- subclasses 2, 5 and 6 belong to class 3.

Finally, the output of the linear layer ($\mathbf{a}^2$) from $\mathbf{a}^1$ is multiplied by $\mathbf{W}^2$. The linear activation function is then applied as in Equation (2.11). This also has only '1' element, which indicates that $\mathbf{p}$ is being assigned to class k^*,

$$\mathbf{a}^2 = purelin\ \left(\mathbf{W}^2\mathbf{a}^1\right). \tag{2.11}$$

The output results will be compared to the target (t_{k*}). The Kohonen learning rule (Equation (2.12), and (2.13)) is used to adjust the competitive layer weights ($\mathbf{W}^1$): if the input pattern is classified correctly, then the winner weight (w_{i*}^1) will be moved toward the input vector ($\mathbf{p}$). However, if the input pattern is classified incorrectly, then the winner weight will be moved far away from the input vector.

If the answer is correct ($a_{k*}^2 = t_{k*}$),

$$\mathbf{w}_{i*}^1(q) = \mathbf{w}_{i*}^1(q-1) + \alpha\left(\mathbf{p} - \mathbf{w}_{i*}^1(q-1)\right). \tag{2.12}$$

If the answer is incorrect ($a_{k*}^2 \neq t_{k*}$),

$$\mathbf{w}_{i*}^1(q) = \mathbf{w}_{i*}^1(q-1) - \alpha\left(\mathbf{p} - \mathbf{w}_{i*}^1(q-1)\right), \tag{2.13}$$

where: q is the order of q^{th} iteration (or epoch) of training (with q = 1,2,3,...,N), and

α is the learning rate.

This is the finished first iteration of the training phase. The updated weight will be used to train the network in the next iteration. In this case, the network will stop training until the iteration reaches the determined setting value [9].

2.4 Back Propagation Neural Network

Back Propagation (BP) neural network is one of the feed forward neural network with supervised learning, and is a popular model used in pattern recognition field. The basic structure of BP neural network is shown in Figure 2.14; it consists of input layer, hidden layer, and output layer. The input layer connects with each neuron of the hidden layer and each neuron of hidden layer also connects with each neuron of the output layer [8].

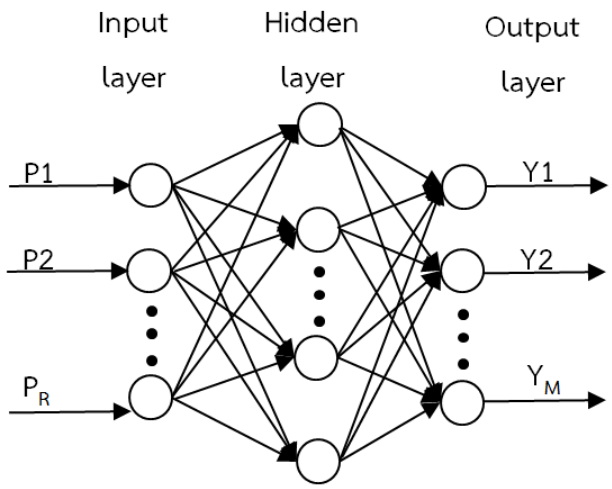


Figure 2.14: Basic structure of BP neural network

2.4.1 BP algorithm

BP training algorithm has two calculation steps. The first is the feed-forward phase and the second is the backpropagation phase. In the process of feed-forward phase, the input data from input layer pass through the hidden layer processing and transmit to output layer, then get the result. In the next step: backpropagation phase, if output layer get the incorrect result from the target, it will be transferred back propagation and re-calculate the weights of each neuron both output layer and hidden layer, respectively. Thereafter update the weight and bias of each neuron, the network will get the output that in consistency with the target.

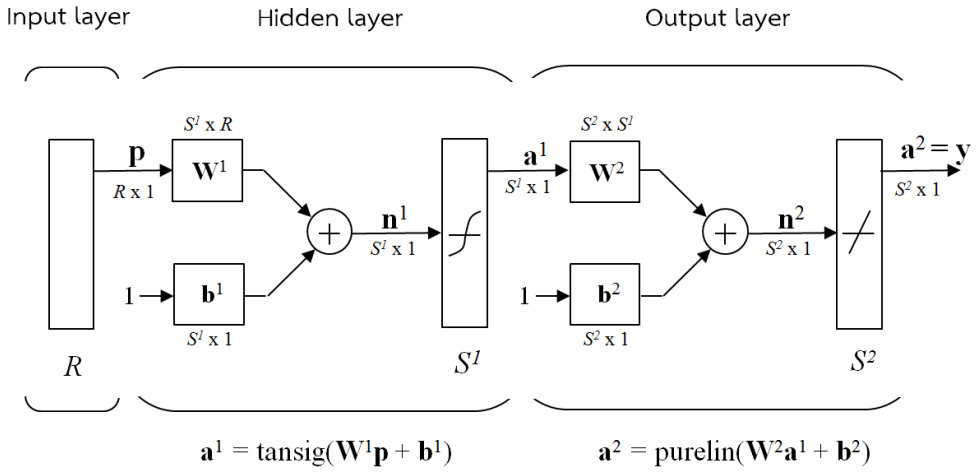


Figure 2.15: The network architecture of BP

The Figure 2.15 shows the network architecture of BP. The algorithm can be divided in two main steps which are feed-forward phase, and backpropagation phase. The basic BP algorithm has the following calculation steps.

2.4.1.1 Feed-forward phase

Feed-forward computation has two steps which are: first step is to get the values of the hidden layer neurons and second step is using those values from hidden layer to compute the values of output layer.

1) First step: when input vector ($\mathbf{p}$) are fed into the hidden layer. They are multiplied with weights of connecting neurons of hidden layer ($\mathbf{W}^1$) in Equation (2.14). The weights ($\mathbf{W}^1$) are initially random values. The *j*-th element of net input of the hidden layer is

$$n_j^1 = \sum_{i=1}^{R} w_{ij}^1 p_i + b_j^1 \ , \qquad (2.14)$$

where b_j^1 and R are bias on hidden neuron j and number of input, respectively. Then apply activation function to compute the output of hidden layer (a_j^1)

$$a_j^1 = f(n_j^1) \tag{2.15}$$

2) Second step: transfer output of hidden layer (a_j^1) to compute at output layer. a_j^1 are multiplied with weights of connecting neurons of output layer ($\mathbf{W}^2$) as Equation (2.16), which is the weights $\mathbf{W}^2$ also initially random values.

Thus, the net of the output layer is

$$n_k^2 = \sum_{j=1}^{S_1} w_{jk}^2 a_j^1 + b_k^2 \quad , \tag{2.16}$$

where b_k^2 and S_1 are bias on output neuron k and number of neural in hidden layer, respectively.

Finally, we apply activation function to compute the output result (a_k^2) of network

$$a_k^2 = f(n_k^2) \quad . \tag{2.17}$$

2.4.1.2 Backpropagation phase

Backpropagation computation has two steps which are: first step is "backpropagation to the output layer" and second is "backpropagation to the hidden layer", respectively.

1) Backpropagation to the output layer

After getting the output result (a_k^2) of network. Next step is to compute the error. Each output neuron (a_k^2, k = 1,..., S_2) receive a target pattern corresponding to the input training pattern. So it compute error information of output layer (δ_k) as Equation (2.18)

$$\delta_k = (t_k - a_k^2) f'(n_k^2) \quad , \tag{2.18}$$

where δ_k is portion of error correction weight adjustment for w_{jk}^2 which due to an error at output neuron (a_k^2); also, the information about the error at neuron a_k^2 is propagated back to the hidden neurons that feed into neuron a_k^2, t_k is output target, $f'(.)$ is derivative of activation function.

Then find the weight correction of output layer (Δw_{jk}^2),

$$\Delta w_{jk}^2 = \alpha \delta_k a_j^1 \quad , \tag{2.19}$$

also find the bias correction of output layer (Δb_k^2),

$$\Delta b_k^2 = \alpha \delta_k \quad , \tag{2.20}$$

where α is learning rate,

both weight correction Δw_{jk}^2 and bias correction Δb_k^2 of output layer will be used to update weight w_{jk}^2 and bias b_k^2 later.

2) Backpropagation to the hidden layer

It uses the error information from output layer to calculate error at hidden layer. Sum of delta inputs of hidden layer is

$$\delta\_in_j = \sum_{k=1}^{S_2} \delta_k w_{jk}^2 \quad . \tag{2.21}$$

Therefore, the error information at hidden layer (δ_j) is

$$\delta_j = \delta\_in_j f'(n_j^1) \quad , \tag{2.22}$$

where δ_j is portion of error correction weight adjustment for w_{ij}^1 that is due to the backpropagation of error information from the output layer to the hidden neuron a_j^1.

Then find the weight correction of hidden layer (Δw_{ij}^1),

$$\Delta w_{ij}^{1} = \alpha \delta_j p_i \quad , \tag{2.23}$$

also find the bias correction of hidden layer (Δb_j^1),

$$\Delta b_j^{1} = \alpha \delta_j \quad , \tag{2.24}$$

both weight correction Δw_{ij}^1 and bias correction Δb_j^1 of hidden layer will be used to update weight w_{ij}^1 and bias b_j^1 later.

Finally, updating weight and bias:

Weight and bias adjustment are the last step of BP training algorithm. It needs to update weight and bias on both output neurons and hidden neurons.

The formula for updating weight of output neurons (w_{jk}^2) is

$$w_{jk}^{2}(new) = w_{jk}^{2}(old) + \Delta w_{jk}^{2} \quad . \tag{2.25}$$

The formula for updating bias of output neurons (b_k^2) is

$$b_k^{2}(new) = b_k^{2}(old) + \Delta b_k^{2} \quad . \tag{2.26}$$

The formula for updating weight of hidden neurons (w_{ij}^1) is

$$w_{ij}^{1}(new) = w_{ij}^{1}(old) + \Delta w_{ij}^{1} \quad . \tag{2.27}$$

The formula for updating bias of hidden neurons (b_j^1) is

$$b_j^{1}(new) = b_j^{1}(old) + \Delta b_j^{1} \quad . \tag{2.28}$$

After weight and bias adjustment process, this is the completed first iteration of training network. The updated weight/bias will be used to train the network in next iteration. In this thesis, network will stop training until iteration reaches to the determined setting value [8], [10]-[12].

2.5 Probabilistic Neural Network

Probabilistic Neural Network (PNN) is a kind of feed forward neural networks with supervised learning. PNN was introduced by D.F. Specht [14] in the 1990s. It is developed based on Radial Basis network and Bayesian classification. PNN has some difference in training process compared to the other supervised neural network: It has no random initial network weights and no iteration process [15]. The Figure 2.16 shows the basic structure of PNN, it has four layers which are Input layer, Pattern layer, Summation layer, and Output layer, whereas the function of each layer is:

1. Input layer - Feed the value of input neuron to each neuron in the pattern layer.
2. Pattern layer - Calculate Euclidean distance between input neuron and hidden neuron from the neuron's center point and then applies the Radial Basis activation function. Thereafter, the resulting value is passed to the summation layer.
3. Summation layer - In this layer, the one pattern neuron represent each category of the target. The target class of each training pattern is reserved with each hidden neuron; the weighted value comes out of a hidden neuron is fed only to the pattern neuron that corresponds to its category. The neurons of the summation layer add the output of the pattern layer for the class they represent.
4. Output layer – Take a final decision on classifying the sample as belong to the specific category. It compare the output for each target category from the summation layer and select the largest output to predict the target category [13], [16].

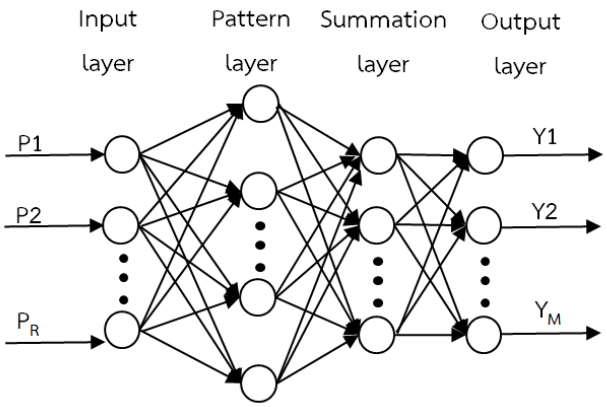


Figure 2.16: Basic structure of PNN

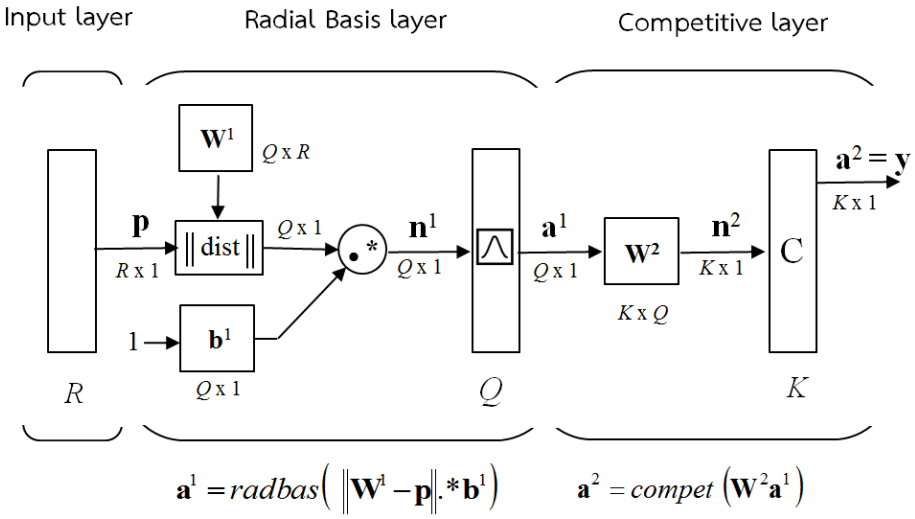


Figure 2.17: The network architecture of PNN

2.5.1 PNN algorithm

In terms of PNN architecture based on Matlab Neural Network tool on Figure 2.17, It is transformed to three layers which are: Input layer, Radial Basis layer, and Competitive layer. The algorithm of PNN can be explained as follows [10], [17].

When input vector ($\mathbf{p}$) from input layer is fed into the Radial Basis layer. Euclidean distance between input neuron and hidden neuron (weight matrix $\mathbf{W}^1$) will be computed. It produces a vector whose elements which indicate how close the input is to the vectors of the training set. The initial $\mathbf{W}^1$ is set to the transpose of the matrix formed from the Q training pairs. Then, the bias vector $\mathbf{b}^1$ is combined with the Euclidean distance of $\|\mathbf{W}^1 - \mathbf{p}\|$ by an element-by-element multiplication, represented by '.*'. Also, Q = number of input/target pairs is equal to the number of neurons in Radial Basis layer, and K = number of classes of input data is equal to the number of neurons in competitive layer.

Thus, the net input of Radial Basis layer is

$$n_i^1 = \|\mathbf{w}_i^1 - \mathbf{p}\| .* b_i^1 \quad , \tag{2.29}$$

where $\mathbf{w}_i^1$ is the vector made of the *i*-th row of $\mathbf{W}^1$,

b_i^1 is the *i*-th element of bias vector $\mathbf{b}^1$.

Next we apply Radial Basis activation function on the net input n_i^1,
so the output Radial Basis layer (a_i^1) is

$$a_i^1 = radbas\ (n_i^1) \quad , \tag{2.30}$$

where
$$radbas\ (n) = e^{-\frac{n}{2\sigma^2}} \tag{2.31}$$

the Radial Basis function on Equation (2.31), based on Gaussian distribution and σ is "smoothing parameter" of the probability density function [15].

The next step, competitive layer receives output of Radial Basis layer (a_i^1), then multiplication with weight matrix ($\mathbf{W}^2$). $\mathbf{W}^2$ is set by corresponding to the target vectors. Each vector has a '1' only in the row associated with that particular class of input, and other is '0'.

Therefore, the net input of competitive layer is

$$n_i^2 = \mathbf{W}^2 a_i^1 \quad , \tag{2.32}$$

where n_i^2 is the *i*-th element of the net input $\mathbf{n}^2$

Then apply competitive activation function on the net input $\mathbf{n}^2$. Competitive layer makes a final decision by producing a '1' to the largest element of $\mathbf{n}^2$, and other is '0'. Thus, the network classifies the input vector into a specific K class because that class has the maximum probability of being correct [10]

$$\mathbf{a}^2 = compet\ \left(\mathbf{n}^2\right) \quad . \tag{2.33}$$

Finally, updating weight and bias:

This is the last step of training part. Weight and bias could be adjusted using any of the standard methods for neural networks. The updated weight and bias will be used in testing network.

2.6 Literature review

2.6.1 Recognition of ECG patterns using Artificial Neural Network

L. He, W. Hou, X. Zhen and C. Peng [18] presented the recognition of Electrocardiogram (ECG) patterns using three types of Artificial Neural Network models which were Self-Organization Map (SOM), Back Propagation (BP) and Learning Vector Quantization (LVQ).

The electrocardiogram examination has been widely used in clinical diagnosis of cardiovascular diseases. The automatic ECG pattern recognition techniques help to reduce physician's work and also improve diagnosis efficiency. A typical ECG waveform contains P wave, QRS complex and T wave in each heart beat as shown in Figure 2.18.

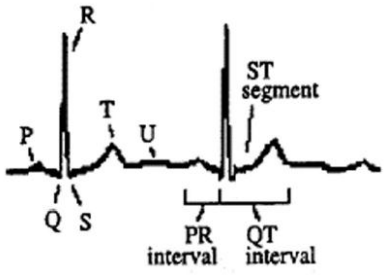


Figure 2.18: A typical ECG waveform [18]

In their experiment, the ECG data from database contained 48 half-hour recordings that were digitized at 360 samples per second with 11-bit resolution. The features being frequently used for ECG analysis in time domain included the wave shape, amplitude, duration, areas, and R-R intervals. They chose four types of ECG patterns from the database, which were respectively annotated as the normal sinus rhythm (N), premature ventricular contraction (PVC), atrial premature beat (A) and left bundle branch block beat (L) as shown in Figure 2.19.

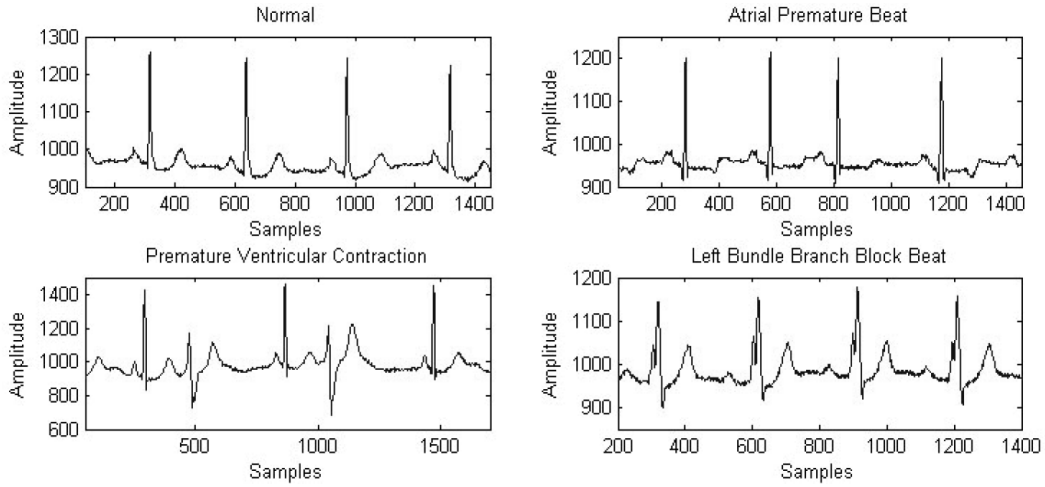


Figure 2.19: Four different ECG patterns to be recognized [18]

The system structure for ECG pattern recognition using artificial neural network method in Figure 2.20 consisted of ECG preprocessing, feature extraction step, feature normalization, learning of the neural network and pattern recognition by neural network, respectively.

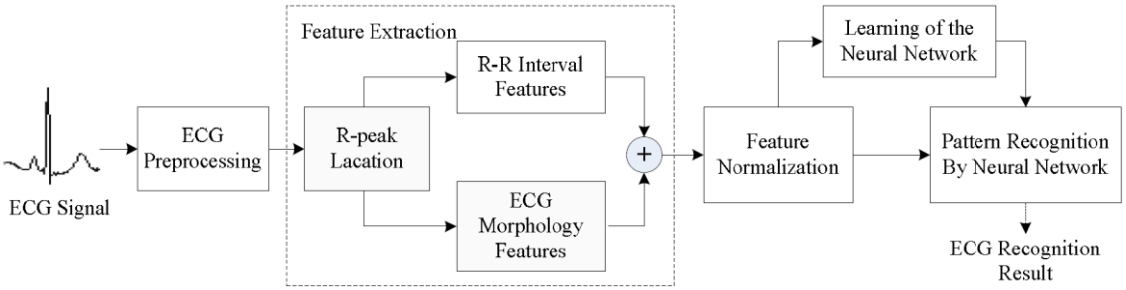


Figure 2.20: System block diagram of neural network for ECG pattern recognition [18]

In the first step, ECG preprocessing was to eliminate the baseline wander and high frequency interference by applying a band-pass filtering to the ECG signals.

The second step was feature extraction. In this step, R-R interval features and ECG morphology features were extracted. For R-R interval features, after the location of R peaks, R-R interval features were calculated. They used two R-R interval features, namely RR1, the R-R interval between the given heartbeat and the previous heartbeat, and RR2, the R-R interval between the given heartbeat and the next heartbeat. For ECG morphology features, features used to represent the morphology of original ECG signals, the amplitude values, were extracted from the ECG sequence 50ms before and 100ms after the R point. Thus only the QRS morphology, with no P or T wave, was included. They took one amplitude value every two samples, which mean a total number of 27 features for each QRS complex were extracted.

The next step was feature normalization. The characteristic vector of 29 dimensions was generated by the R-R interval features in additional with the ECG morphology features. Before being fed to the neural network models, the features were normalized between -1 and 1.

The last step was artificial neural networks recognition, which separated in two phases; one was the training phase, and the other was the test phase. In the training phase, the connection weights were automatically adjusted to map the input to the corresponding output.

The ECG records of 4 different types of patterns were obtained from 11 patients. Two hundred of ECG segments were chosen for each of the pattern, which produced a dataset with a total number of 800 ECG records, each containing a QRS complex. The four types of patterns were respectively designated as N, A, L and V. The whole dataset was divided into four groups of equal size with equal number of

the 4 patterns in each group, and every neural network model was tested by a different data group, while the other three was used for training.

The performance of the neural networks was evaluated by the recognition sensitivities, the overall recognition accuracy and the neurons number needed. The recognition sensitivity to a particular pattern was defined as the ratio of beats of that pattern that were correctly recognized to the total beats of that pattern. The overall accuracy was defined as the ratio of the total number of beats recognized correctly to the total number of beats in the test phase.

For their recognition results, the performance of the SOM was better than BP and LVQ networks as shown in Table 2.3.

Table 2.3: Performance of recognition of ECG patterns using different neural network models [18]

Results	Recognition Sensitivities								Overall Accuracy	Neurons needed
	N		V		A		L			
	Learning	Testing	Learning	Testing	Learning	Testing	Learning	Testing		
SOM	93.3%	94.0%	95.3%	96.0%	91.3%	92.0%	98.7%	100%	95.5%	129
BP	98.0%	90.0%	100%	100%	98%	84.0%	100%	96.0%	92.5%	41
LVQ	81.5%	72.7%	100%	98.3%	100%	99.3%	100%	96%	91.5%	43

2.6.2 Expert system for speaker identification using lip features with PCA

Anuj Mehra et al. [19] presented the application for speaker Identification using Lip features by PCA (Principal Component Analysis) along with neural networks. They used different neural network classifiers which included Back Propagation (BP), Radial Basis Function (RBF) and Learning Vector Quantization (LVQ).

In the first step of their experiment, the database was chosen which consisted of 7 subjects saying the first 4 digits in English and the digits were repeated twice by each speaker. Each digit spoken constituted of 6 instances and the size of each image was 100 by 75 pixels. The 7 subjects of database with 24 images in each class was split into two datasets, 168 images for training and 168 images for testing dataset. In the next step, PCA technique had been used for feature extraction from the six geometric lip features which were height of the outer corners of the mouth (H1), width of the outer, corners of the mouth (W1), height of the inner corners of the mouth (H2), width of the inner corners of the mouth (W2), height of the upper lip (H3), and height of the lower lip (H4) as shown in figure 2.21.

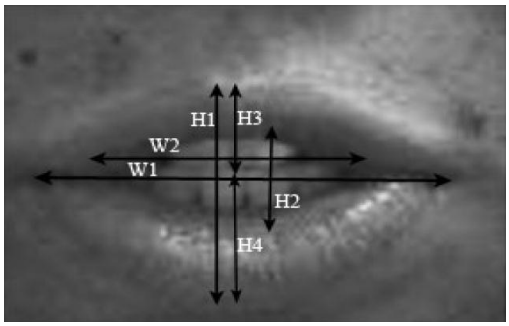


Figure 2.21: Six geometry of lip features [19]

After feature extraction process, data was fed into the neural network classifiers (BP, RBF, and LVQ). An output matrix was achieved which consisted of the reduced set of images. A target matrix was formed and testing images were trained

using PCA followed by the simulation of the neural networks resulting in the calculation of the accuracy (number of images of training and testing dataset matched) in the last step. The block diagram of speaker identification process is shown in Figure 2.22.

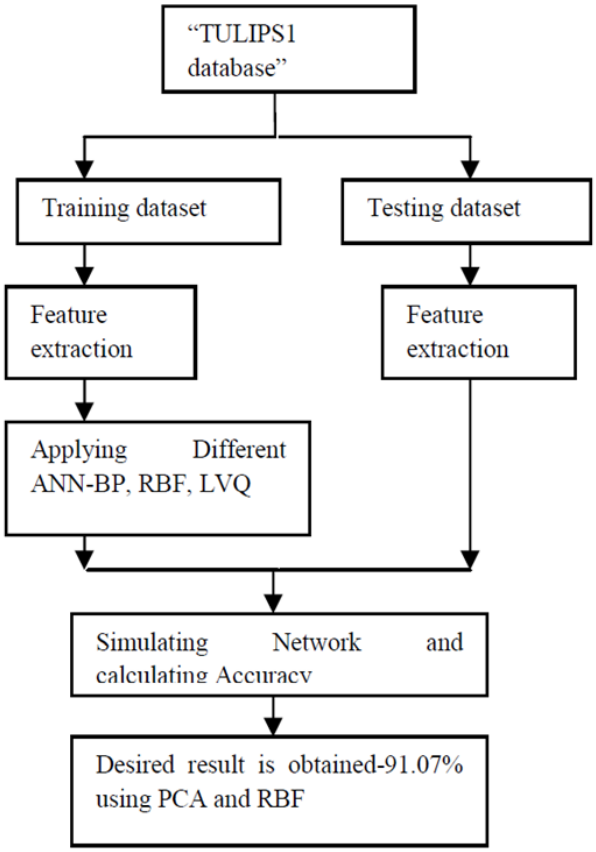


Figure 2.22: Block Diagram for Speaker Identification [19]

The setting of each neural network classifier (BP, RBF, and LVQ) are shown Table 2.4, 2.5, and 2.6, respectively.

Table 2.4: Statistical data of BP [19]

	For PCA
Input Vector Nodes	6
Number of Hidden Layers	2
Number of neurons (hidden1 ,hidden2 & output)	20,25,7
Transfer functions (input, hidden & output)	Tansigmoid, tansigmoid, linear
Network Learning rate	.01
Epochs	1000

Table 2.5: Statistical data of RBF [19]

	For PCA
Number of Radial Basis Layers	1
Number of neurons (input ,radial basis & output)	6,125,7
Spread	25
Epochs	50

Table 2.6: Statistical data of LVQ [19]

	For PCA
Number of competitive Layers	1
Number of neurons (input ,competitive & output)	6,40,7
Transfer function	Lvq1.0
Network Learning rate	.001
Epochs	1000

The result of their experiment in Table 2.7, showed that the recognition performance for PCA with RBF was better than the other methods and also BP network achieved better accuracy than LVQ.

Table 2.7: Recognition rate of different neural network models [19]

Methods	**BP**	**RBF**	**LVQ**
Recognition Rate	89.88% (151/168)	91.07% (153/168)	87.5% (147/168)

2.6.3 Analyzing financial distress of listed companies using Neural Network

Q. Wang, Q. Yang, and M. Zhang [20] presented Back Propagation (BP) and Probabilistic neural network (PNN) models to predicate financial distress. The sample consisted of 276 companies listed on the Shanghai Stock Exchange and Shenzhen Stock Exchange over the period 2001–2010. The companies was separated in two groups, which were: (1.) specially treated (ST) by China Securities Supervision and Management Committee (CSSMC) were considered as companies in financial distress, and (2.) those never specially treated were regarded as healthy finance.

In experiment, they used the financial statement data of three years prior to the financial distress to predict the financial status. The non-ST companies group was matched with ST companies group by industry, fiscal year and size. Finally, the sample was composed of 138 ST companies and 138 non-ST companies. The selection of variables to be used as candidates for participation in the input vector was based upon prior other researcher's work. Therefore, they selected 49 financial indicators and 3 non-financial indicators which had great influence on financial distress.

The 49 financial indicators in Table 2.8, cover 7 features which are per share indicators, profitability, solvency, growth ability, operating capacity, capital structure, cash flow. The non-financial indicators are the proportion of independent directors, the first largest shareholding ratio and the proportion of the top ten shareholders.

In the factor analysis step, the variable data of 138 ST companies and 138 non-ST companies, each of them had 52 ratios which included 49 financial indicators and 3 non-financial indicators for the predictive financial distress model factors. With the purpose of making the factor structures easier and simpler to explicate, they used the SPSS statistical software to run factor analysis and PCA with varimax for rotation (VARIMAX) on the financial indicator.

Table 2.8: The 49 financial indicators [20]

Per share indicators	X1:Earnings per share
	X2:Net assets per share
	X3:Operating income per share
	X4:Operating profit per share
	X5:Cash flow from operating activities per share
	X6:Net cash flow per share
Profitability indicators	X7:Return on net assets
	X8:Return on assets
	X9:Assets net margin
	X10:Returns on invested capital
	X11:Sales margin
	X12:Cost to sales ratio
	X13:Net profit / Total operating income
	X14:Operating profit / Total operating income
	X15:Profit to cost ratio
Solvency indicators	X16:Current ratio
	X17:Quick Ratio
	X18:Equity ratio
	X19:Shareholders' equity / Total liabilities
	X20:Net operating cash flow / total liabilities
	X21:Net operating cash flow / current liabilities
	X22:Long-term debt / working capital
	X23:Cash /current liabilities
Growth ability indicators	X24:The growth rate of earnings per share
	X25:The growth rate of operating income
	X26:The growth rate of total profit
	X27:The growth rate of Operating profit
	X28The growth rate of net profit
	X29:The growth rate of return of net assets
	X30:The growth rate of net assets
Operating capacity indicators	X31:Inventory turnover ratio
	X32:Accounts receivable turnover ratio
	X33:Accounts payable turnover ratio
	X34:Current Assets turnover ratio
	X35:Fixed Assets turnover ratio
	X36:Total asset turnover ratio
	X37:The turnover rate of the shareholders' equity
Capital structure indicators	X38:Asset-liability ratio
	X39:Long-term liabilities / equity
	X40:Current assets / total assets
	X41:The equity / total invested capital
	X42:Suitable rate of long-term assets
	X43:The ratio of fixed assets
	X44:Current liabilities / total liabilities
Cash flow indicators	X45:Cash of The sales of commodity and labor income / operating income
	X46:Sales Cash Ratio
	X47:Net operating cash flow / Net Operating Income
	X48:Net profit cash content
	X49:Operating income cash content

The result of factor analysis in Table 2.9, shows the factor loadings, the eigenvalues and the explained variance information for each variable. In addition, the total explained variance was 88.9%. Consequently, they selected 15 variables presented high communality value to be the input vector of Neural Network.

Table 2.9: Factor analysis results [20]

factors	Variables	Factor loading	eigenvalues	Explained variance
1	X1	0.800	10.335	20.670%
	X4	0.774		
	X8	0.878		
	X7	0.911		
	X9	0.907		
	X10	0.893		
	X11	0.641		
2	X14	0.631	7.945	15.890%
	X15	0.729		
	X16	0.921		
	X17	0.915		
	X19	0.909		
3	X3	0.843	5.553	11.065%
	X34	0.712		
	X35	0.712		
	X37	0.858		
	X36	0.894		
4	X5	0.660	3.720	7.440%
	X21	0.920		
	X23	0.920		
	X46	0.867		
	X44	0.818		
5	X18	0.770	3.203	6.407%
	X38	0.678		
	X41	-0.714		
	X44	-0.796		
	X39	0.853		
6	X24	0.892	2.436	4.871%
	X25	0.449		
	X27	0.888		
	X26	0.926		
	X28	0.898		
7	X31	0.496	2.224	4.448%
	X33	0.493		
	X40	-0.775		
	X43	0.765		
8	X6	0.832	1.736	3.472%
	X30	0.862		
	X42	0.638		
9	X45	0.925	1.265	2.730%
	X49	0.925		
10	X47	0.877	1.194	2.387%
	X48	0.876		
11	X2	0.845	1.12	2.239%
12	X20	0.904	1.02	2.041%
13	X22	0.910	0.942	1.884%
14	X29	0.872	0.875	1.749%
15	X12	0.634	0.819	1.637%
	X32	0.596		
Total explained variance				88.932%

The next step is neural network for the financial distress prediction models by MATLAB programming. The 15 comprehensive financial variables and 3 non-financial variables were loaded as input nodes of neural network and output nodes were two types:" healthy company" and "distressed company". Two kinds of sample: firstly 100 ST companies and 100 non-ST companies were randomly chosen as training samples, the rest of 76 companies as test samples. Secondly 69 ST companies and 69 non-ST companies were randomly chosen as training samples, the rest of 138 companies as test samples. The accuracy results of two neural network models: BP and PNN are shown in Table 2.10 and Table 2.11, respectively.

Table 2.10: Results of BP model for the financial distress prediction [20]

	Training samples		**Test samples**	
	Sample size	***Accuracy***	***Sample size***	***Accuracy***
ST	100	68	38	27
Non-ST	100	78	38	24
Average accuracy	73%		67%	
ST	69	46	69	47
Non-ST	69	57	69	42
Average accuracy	75%		64.5%	

Table 2.11: Results of PNN model for the financial distress prediction [20]

	Training sample		**Test sample**	
	Sample size	***Accuracy***	***Sample size***	***Accuracy***
ST	100	100	38	28
Non-ST	100	100	38	26
Average Accuracy	100%		71%	
ST	69	69	69	56
Non-ST	69	69	69	40
Average Accuracy	100%		70%	

They concluded that, the results of BP model with determine the best of the number of hidden layer nodes is 9. When the training sample size was 100, accuracy rates of training sample and test sample, were respective 73%, 67%. When the training sample size was half of the all sample size, that was 138, accuracy rates of training sample and test sample, were respective 75%, 64.5%.

Meanwhile the results of PNN model with hidden node was set by default of MATLAB programing. In the first case, the accuracy rates of training sample and test sample, were respective 100%, 71%. In the second case, accuracy rates of training sample and test sample were respective 100%, 70%. Therefore the predication ability of PNN was much higher than that of BP model.

In literature review, we presented three researches that related with pattern recognition by using neural network in medicine, speaker Identification and financial fields, respectively. These applications are an inspiration to our thesis. The next chapter is research methodology. We will apply neural network to classify Fly Height failure pattern from Hard Disk Drive manufacturing.

CHAPTER 3

RESEARCH METHODOLOGY

There will be two major research methodologies in this chapter. The first topic is data and sample collection which has two sub-topics: appearance of fly height failure pattern and failure analysis result of each fly height pattern. The second will be classification system which describes three sub-topics: preprocessing data, network training, and network testing.

3.1 Data and sample collection

After the Fly Height (FH) measurement in functional testing process is done, and failed drives were detected. Normally, these drives will be sent to scrap, or sampling for Failure Analysis (FA). In this thesis, data and sample are from 2.5-inch HDD Seagate product, as summarized in Table 3.1. The FH measurement results of failed drives are collected from log file database.

Table 3.1: HDD sample specifications

HDD size	2.5 inch
Capacity	1 TB
Number of head	8 heads
Number of disk	4 disks
Spin speed	7200 RPM
Interface	SAS/SATA
HDD technology	Perpendicular recording with TGMR head

3.1.1 Appearance of Fly Height failure pattern

The FH profile is the touchdown power that represents the FH distance in each data zone during both write and read operations. However, to reduce the complexity of input dataset, only FH profile from write operation was chosen for pattern classification. Then, FH profile contains 10 data points for each head.

The six types of failure patterns found on failed heads, as shown in Figure 3.1. On FH profile, y-axis is the heater power and x-axis is the data zone of a disk where the outer diameter toward the inner diameter of disk. In this thesis, dataset are collected from FA results by six types of failure patterns, a 1,000 samples per type are collected. Then, there are 6,000 failure samples to supervise network and test classification performance of network.

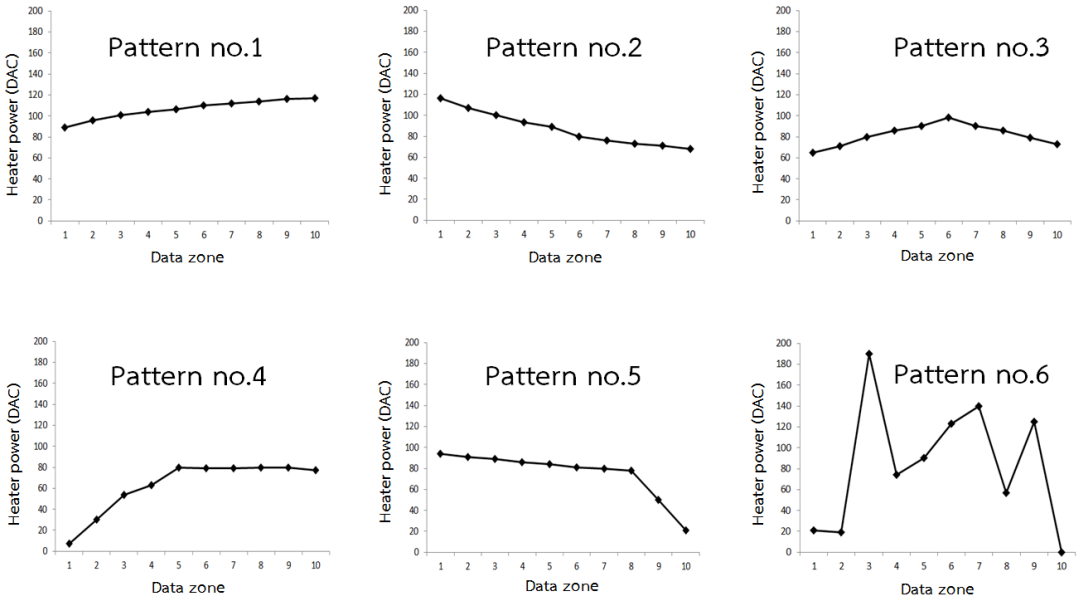


Figure 3.1: Six types of failure patterns from FH measurement

3.1.2 Failure Analysis result of each Fly Height pattern

Based on experience of FA results, each FH pattern can indicate three causes of failure which are: pattern no.1 and no.2 indicate the failure are from physical damage on Head Gimbal Assembly (HGA). Pattern no.3, 4 and no.5 relate to contamination on slider Air Bearing Surface (ABS), and pattern no.6 is head instability. The example of failure analysis results are shown in Figure 3.2, 3.3, and 3.4, respectively.

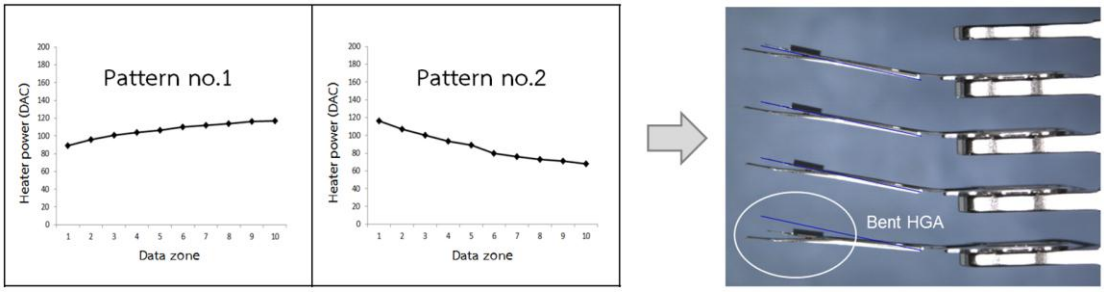


Figure 3.2: FA result of pattern no.1 and no.2

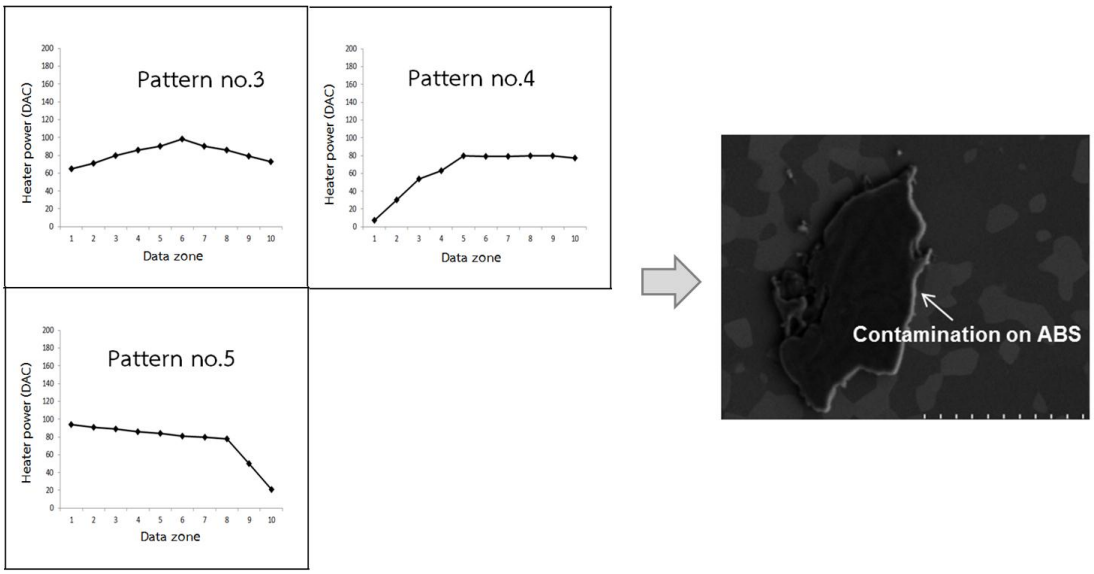


Figure 3.3: FA result of pattern no.3, 4, and 5

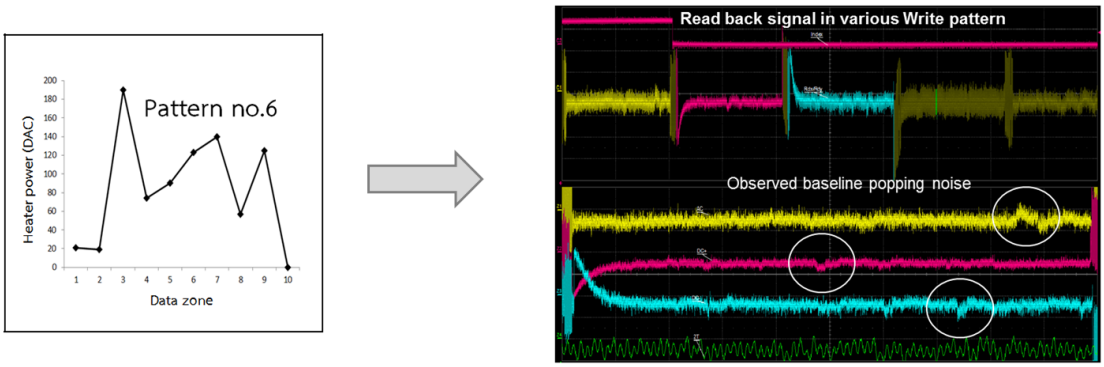


Figure 3.4: FA result of pattern no.6

The first FA results of pattern no.1 and no.2, in Figure 3.2, show the side view of Head Stack Assembly (HSA) where up tab was removed. The bent suspension is obviously seen as a root cause of failure. The second FA result of pattern no.3, 4, and 5, in Figure 3.3, illustrated by FESEM (Field Emission Scanning Electron Microscopy) image. There is a contamination on the face of slider Air Bearing Surface (ABS). The last FA result of pattern no.6 in Figure 3.4, shows the readback signal from media with various write pattern. This is monitored by oscilloscope, which is observed baseline popping noise. These abnormal cases are screened by FH testing before shipment to customers.

3.2 Classification system

Classification system with Neural Network (NN) in this thesis operates in three parts: first is data preprocessing, second is the training part, and the last is the testing part as presented in Figure 3.5.

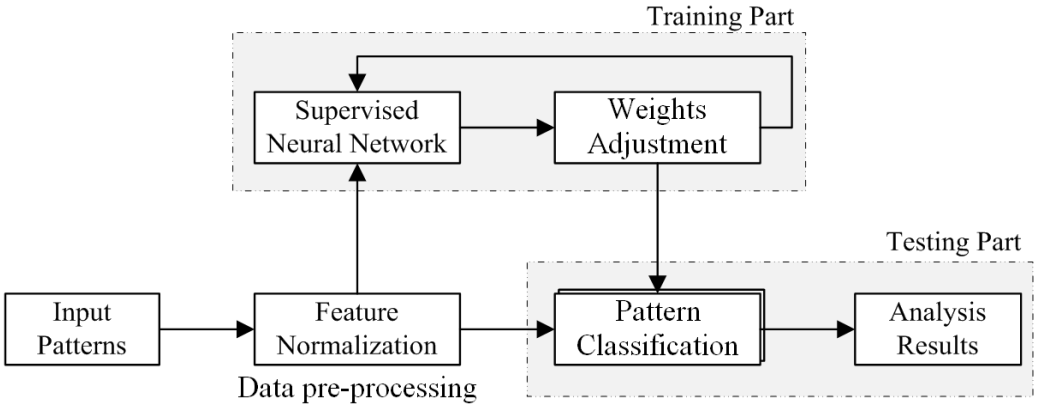


Figure 3.5: System block diagram for the classification of FH failure pattern using neural network

The system for the classification of FH failure pattern using neural network, it has operation steps as follows.

3.2.1 Data preprocessing

In order to deal with appropriate data, in the preprocessing step, before training and testing the network, all of the input data are normalized into a range of 0 and 1 by Equation (3.1)

$$pn_R = \frac{(p_R - \min p)}{(\max p - \min p)} \quad , \tag{3.1}$$

where:

pn_R is the R^{th} data point that is normalized between 0 and 1,

p_R is the R^{th} data point,

$\min p$ is the minimum value in a pattern,

$\max p$ is the maximum value in a pattern.

3.2.2 Network training

This part is the supervised network. After feature normalization, Neural Network is learned by computing relation between input sample, actual output, and target output. The connection weights are automatically adjusted to require the actual output that corresponds with target output. This thesis determines the network will stop training loop until iteration number reaches the setting.

3.2.3 Network testing

After finishing network training, the next step is network testing. The network performance is evaluated by the accuracy rate. The accuracy rate is the number of correctly classified patterns output, divided by the total number of testing patterns fed into the network, and then multiplied with 100 as shown in Equation (3.2)

$$\text{Accuracy rate}(\%) = \frac{\text{The number of correctly classified patterns}}{\text{The total number of testing patterns}} \times 100 \quad . \qquad (3.2)$$

CHAPTER 4

EXPERIMENT AND RESULTS

This chapter presents the evaluation and performance comparison of Fly Height (FH) failure pattern classification, which is associated to Neural Network (NN) model. The topics include: (1) network verification, (2) comparison results between known and unknown dataset, (3) evaluation number of training iterations, (4) evaluation number of training samples, (5) evaluation number of hidden neurons, (6) evaluation on smoothing parameter, (7) performance comparison of three neural network models and the last topic is (8) training time comparison.

4.1 Network verification

The whole data from the 1,000 samples for each type of failure patterns are separated into two groups with an equal number of six patterns. A first half of dataset (500 samples each pattern) will be used for network training, whereas the other group (also 500 samples each pattern) will be used for network testing. Then, the input data with 10×3000 array of six patterns will be fed into the network for training and network testing, respectively.

First of all, we selected the first half of dataset (first 500 samples each pattern) to verify network. We used these samples to train and also test the network. The determined simulation condition of each NN model for network verification is shown in Table 4.1.

Table 4.1: Simulation conditions for network verification

Simulation conditions	LVQ	BP	PNN
Number of hidden neurons:	10	10	10
Training iterations:	50	50	-
Learning rate:	0.01	0.01	-
Smoothing parameter:	-	-	0.1
Activation function on hidden layer:	Competitive	Tansig	Radbas
Activation function on output layer:	Linear	Linear	Competitive

Table 4.2: Network verification results

Average accuracy		
LVQ	BP	PNN
88.4%	84.1%	99.2%

The network verification results on Table 4.2 shows that the Probabilistic neural network (PNN) provides the best performance with an average accuracy at 99.2%. The Learning Vector Quantization (LVQ) and Back Propagation (BP) yield the average accuracy at 88.4% and 84.1%, respectively.

4.2 Comparison results between known and unknown dataset

After we verified network with the same training/testing dataset, we test network with the unknown dataset (network has never seen it before). The simulation conditions of each NN model are still the same as Table 4.1. However, unknown dataset from the remaining group (second 500 samples each pattern) will be used for testing network.

Table 4.3: Comparison results between known and unknown dataset

NN model	Average accuracy		Difference
	Known dataset	Unknown dataset	
LVQ	88.4%	88.2%	-0.2%
BP	84.1%	81.6%	-2.5%
PNN	99.2%	96.2%	-3.0%

The comparison results between known and unknown dataset are shown in Table 4.3. The result shows that LVQ is no significant difference between known and unknown dataset (only -0.2%). While PNN has a maximum difference, (decrease -3%). For this comparison, unknown dataset has no effect on the average accuracy of LVQ.

4.3 Evaluation number of training iterations

This experiment evaluates number of training iterations. It can supervise network with various number of training iterations on two models, which are LVQ and BP. The simulation conditions are determined as shown in Table 4.4. The network will be trained by a first half of dataset. The remaining group used for testing. The step of various number of training iterations are 5, 10, 20, 30, 40, 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 (total 15 settings).

Table 4.4: Simulation conditions of various number of training iterations

Simulation conditions	LVQ	BP
Various number of training iterations:	5,10,20,30,40,50,100,150,200,250,300,350,400,450,500	
Number of hidden neurons:	10	10
Learning rate:	0.01	0.01
Activation function on hidden layer:	Competitive	Tansig
Activation function on output layer:	Linear	Linear

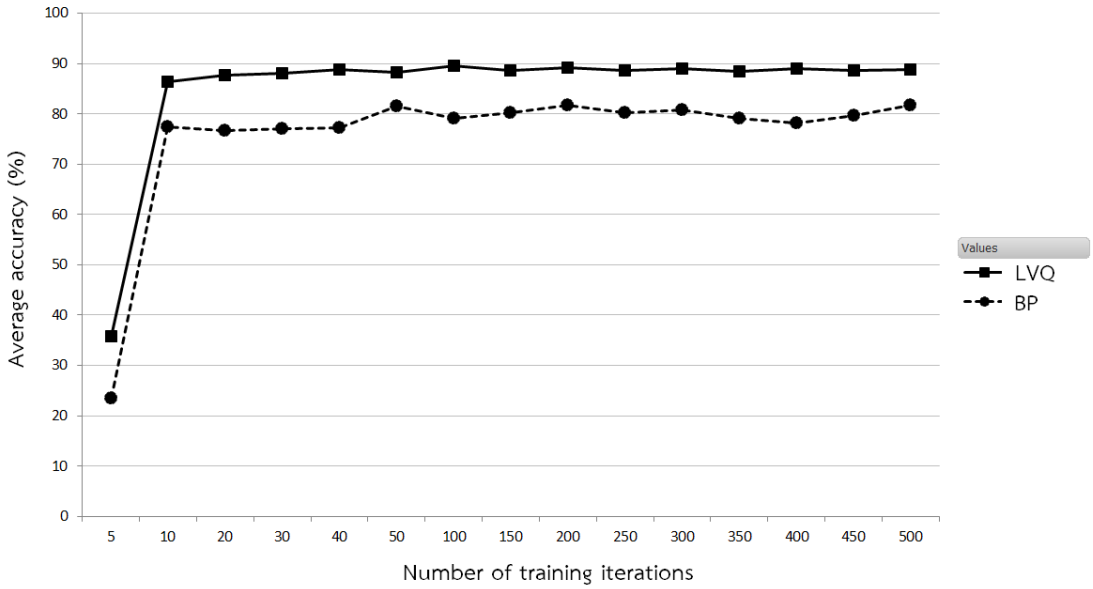


Figure 4.1: Average accuracy at various number of training iterations between LVQ and BP.

Figure 4.2: Difference of accuracy between LVQ and BP at various number of training iterations.

Table 4.5: Simulation results at various number of training iterations between LVQ and BP.

Number of training iterations	Average accuracy		Difference
	LVQ	BP	
5	35.9%	23.5%	12.4%
10	86.4%	77.3%	9.1%
20	87.7%	76.7%	11.0%
30	88.1%	77.0%	11.1%
40	88.8%	77.3%	11.5%
50	88.2%	81.6%	6.7%
100	89.6%	79.1%	10.4%
150	88.6%	80.2%	8.4%
200	89.2%	81.7%	7.5%
250	88.5%	80.1%	8.4%
300	88.9%	80.8%	8.1%
350	88.4%	79.1%	9.3%
400	89.1%	78.2%	10.9%
450	88.7%	79.6%	9.1%
500	88.8%	81.8%	7.0%

Figure 4.1 and Table 4.5 present average accuracy at various number of training iterations between LVQ and BP. The simulation results show that both LVQ and BP show extremely high accuracy when the number of training iterations are between 5 and 10. The accuracy saturates when the number of iteration is higher than 10. LVQ has accuracy of 87.7%-89.6%, while BP fluctuates around 76.7%-84.2%. Furthermore, accuracy of LVQ is higher than BP for all number of training iterations.

4.4 Evaluation number of training samples

This experiment evaluates the number of training samples. The network will be trained by a first half of dataset with the various number of training samples, which are 5, 10, 20, 30, 40, 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 (total 15 settings). The remaining group will be used for testing the whole number of training samples. The simulation conditions are determined as shown in Table 4.6.

Table 4.6: Simulation conditions of various number of training samples.

Simulation conditions	LVQ	BP	PNN
Various number of training samples:	5,10,20,30,40,50,100,150,200,250,300,350,400,450,500		
Number of hidden neurons:	10	10	10
Training iterations:	500	500	-
Learning rate:	0.01	0.01	-
Smoothing parameter:	-	-	0.1
Activation function on hidden layer:	Competitive	Tansig	Radbas
Activation function on output layer:	Linear	Linear	Competitive

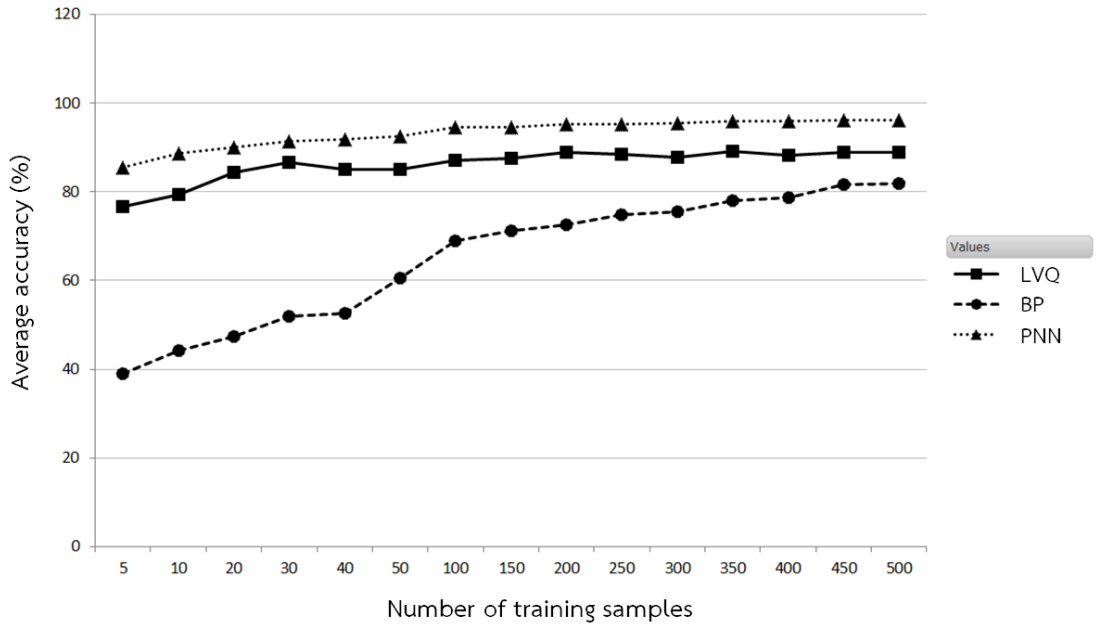


Figure 4.3: Average accuracy at various number of training samples among LVQ, BP, and PNN

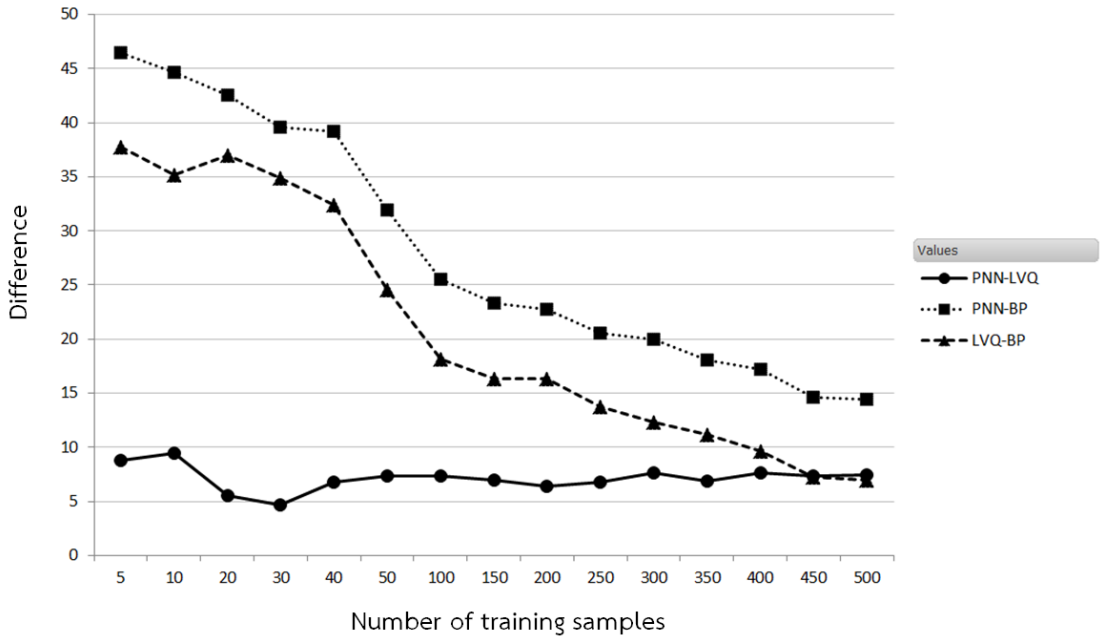


Figure 4.4: Difference of accuracy among LVQ, BP, and PNN at various number of training samples

Table 4.7: Simulation results at various number of training samples among LVQ, BP, and PNN

Number of training samples	Average accuracy			Difference		
	LVQ	BP	PNN	PNN-LVQ	PNN-BP	LVQ-BP
5	76.7%	39.0%	85.5%	8.8%	46.5%	37.7%
10	79.3%	44.1%	88.7%	9.4%	44.6%	35.2%
20	84.4%	47.5%	90.0%	5.5%	42.5%	37.0%
30	86.7%	51.8%	91.4%	4.7%	39.6%	34.9%
40	85.0%	52.6%	91.8%	6.8%	39.2%	32.4%
50	85.1%	60.5%	92.4%	7.3%	31.9%	24.6%
100	87.1%	69.0%	94.5%	7.4%	25.5%	18.1%
150	87.6%	71.2%	94.6%	7.0%	23.3%	16.4%
200	88.9%	72.6%	95.3%	6.4%	22.7%	16.3%
250	88.5%	74.7%	95.3%	6.8%	20.6%	13.8%
300	87.9%	75.5%	95.5%	7.6%	20.0%	12.3%
350	89.1%	77.9%	95.9%	6.9%	18.0%	11.2%
400	88.3%	78.7%	96.0%	7.6%	17.2%	9.6%
450	88.8%	81.5%	96.1%	7.3%	14.6%	7.3%
500	88.8%	81.8%	96.2%	7.4%	14.4%	7.0%

Figure 4.3 and Table 4.7 present the average accuracy at various number of training samples among LVQ, BP and PNN. The simulation results are observed that number of training samples corresponds with accuracy rate, when increase number of training samples. The accuracy rate will improve especially on BP. Moreover,

accuracy on both PNN and LVQ are always higher than BP and also PNN is higher than LVQ.

4.5 Evaluation number of hidden neurons

This experiment evaluates number of hidden neurons. It can simulate with various number of hidden neurons on two models, which are LVQ and BP. On the previous experiment that found the number of training samples corresponds with accuracy rate. So we selected a first half of dataset in each pattern to teach the network. The remaining group still will be used for testing. Also the number of training 500 iterations is chosen. The various number of hidden neurons that will be simulated are 10, 20, 30, 40 and 50. The simulation conditions are summarized as follows Table 4.8.

Table 4.8: Simulation conditions of various number of hidden neurons

Simulation conditions	LVQ	BP
Various number of hidden neurons:	10, 20, 30, 40, 50	
Training samples:	500	500
Training iterations:	500	500
Learning rate:	0.01	0.01
Activation function on hidden layer:	Competitive	Tansig
Activation function on output layer:	Linear	Linear

Table 4.9: Simulation results at various number of hidden neurons between LVQ and BP

Number of hidden neurons	Average accuracy		Difference
	LVQ	BP	
10	88.8%	81.8%	7.0%
20	89.6%	79.8%	9.8%
30	87.8%	79.1%	8.7%
40	89.0%	74.9%	14.1%
50	90.7%	78.5%	12.2%

The simulation results on Table 4.9 are observed that the accuracy of LVQ is improved when number of hidden neurons increased. LVQ yields the best accuracy of 90.7% with 50 hidden neurons. This means that LVQ require the maximum number of hidden neurons in the experiment for the highest precision possible. On the other hand, BP is lower accuracy when number of hidden neurons increased. Therefore, BP needs only 10 hidden neurons to be suitable for best precision (81.8%). Based on S. Karsoliya's study [26], he explained about the approximating number of hidden neurons in hidden Layer that BP required. If the number of neurons are less as compared to the complexity of the problem data then "Under-fitting" may occur. The under-fitting occurs when there are too few neurons in the hidden layers to enough detect the signals in a complex data set. On the other hand, if unnecessary more neurons are used in the network then "Over-fitting" may occur.

4.6 Evaluation on smoothing parameter

This experiment evaluates smoothing parameter (σ) on PNN. Smoothing parameter is the parameter on the probability density function as shown in Equation (2.31) in chapter 2. The network should select an appropriate smoothing parameter with the problem. Dataset for training/testing network are same as in the previous experiment. The various number of smoothing parameters will be simulated are 0.05, 0.1, 0.2, 0.3, 0.4, and 0.5. The simulation conditions are summarized as shown in Table 4.10.

Table 4.10: Simulation conditions of various smoothing parameters

Simulation conditions	PNN
Various smoothing parameters:	0.05, 0.1, 0.2, 0.3, 0.4, 0.5
Number of hidden neurons:	10
Number of training samples:	500
Activation function on hidden layer:	Radbas
Activation function on output layer:	Competitive

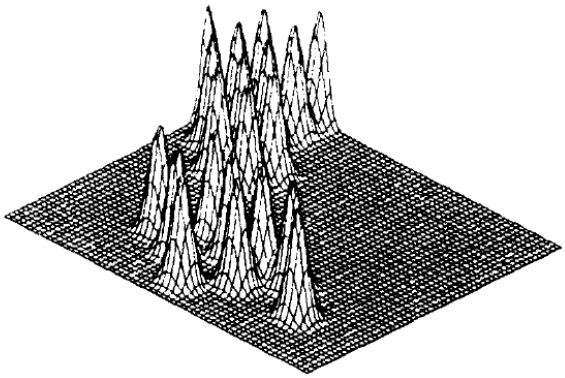
Table 4.11: Simulation results at various smoothing parameters with PNN

Smoothing parameter	Average accuracy on PNN
0.05	96.0%
0.1	96.2%
0.2	95.0%
0.3	92.4%
0.4	89.6%
0.5	87.2%

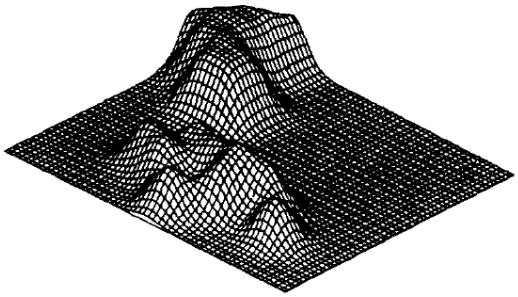
The simulation results on Table 4.11 shows that the different smoothing parameter affects to the accuracy rate. Based on D. F. SPECHT's work [15], he

explained the effect of different smoothing parameter to the probability density function of the random vector X. He estimated from a set of samples taken from category A $\left(f_A(X)\right)$ in two-dimensional. If smoothing parameter is small, the estimated parent density function to have distinct modes corresponding to the locations of the training samples as shown in Figure 4.5 (a). If smoothing parameter is larger, as shown in Figure 4.5 (b), it produces a greater degree of interpolation between points. The values of X that are close to the training samples are estimated to have about the same probability of occurrence as the given samples. If smoothing parameter is an even larger value, as shown in Figure 4.5 (c), it produces a greater degree of interpolation. A very large value of smoothing parameter would cause the estimated density to be Gaussian regardless of the true underlying distribution.

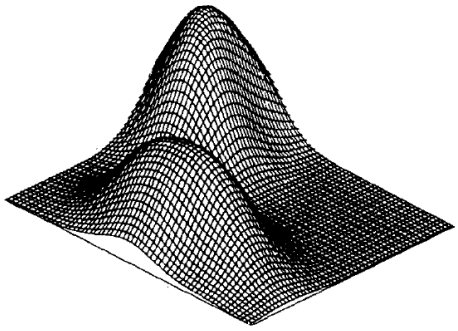
Thus, in this work, 0.1 is the suitable smoothing parameter that obtains the best performance on PNN with the accuracy rate of 96.2%.



(a) A small value of σ



(b) A larger value of σ



(c) An even larger value of σ

Figure 4.5: The smoothing effect of different values of smoothing parameter on a probability density function estimated from samples [15].

4.7 Performance comparison of three Neural Network models

In order to compare the classification accuracy among three NN models, the same training and testing dataset will be considered. The network has been trained by a first half of dataset and the remaining group has been used for testing. The performance comparison of three NN models is illustrated as follows.

Table 4.12: Performance comparison of three NN models

NN model	The highest of Average accuracy	Number of hidden neurons	Number of training iterations	Smoothing parameter
LVQ	90.7%	50	500	-
BP	81.8%	10	500	-
PNN	96.2%	10	-	0.1

Table 4.13: Performance comparison in each pattern of three NN models

Pattern	Accuracy			Accuracy difference		
	LVQ	BP	PNN	PNN-LVQ	PNN-BP	LVQ-BP
1	91.4%	81.2%	94.4%	3.0%	13.2%	10.2%
2	98.4%	94.2%	98.4%	0.0%	4.2%	4.2%
3	87.4%	80.0%	92.4%	5.0%	12.4%	7.4%
4	82.4%	83.6%	95.6%	13.2%	12.0%	-1.2%
5	97.8%	93.2%	99.4%	1.6%	6.2%	4.6%
6	86.8%	58.4%	97.0%	10.2%	38.6%	28.4%
Average	**90.7%**	**81.8%**	**96.2%**	**5.5%**	**14.4%**	**8.9%**

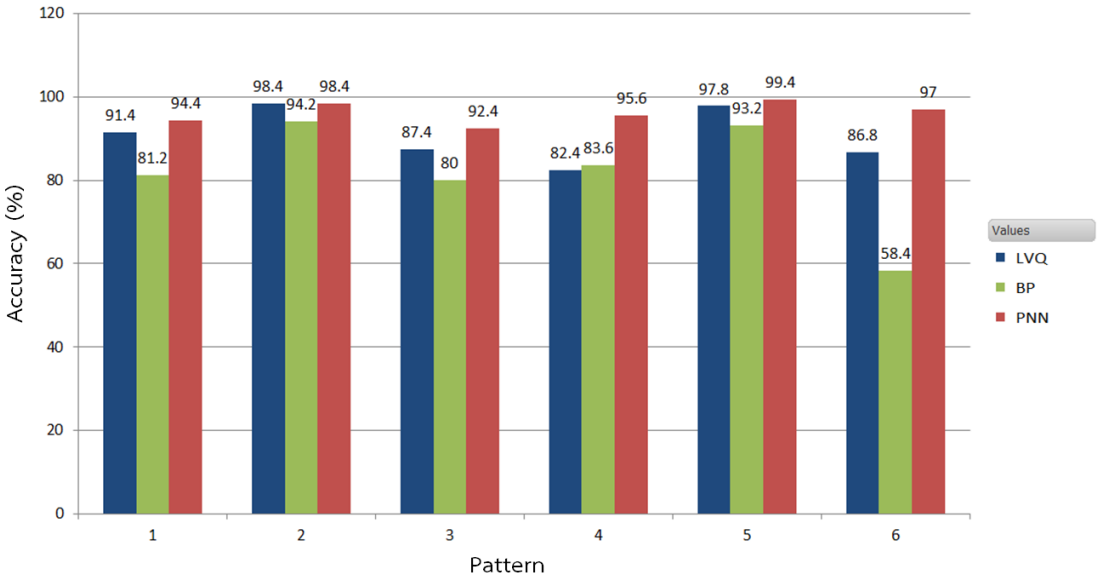


Figure 4.6: Accuracy in each pattern on three NN models

As shown in Table 4.12, LVQ shows the highest average accuracy at 90.7% with 50 hidden neurons and 500 training iterations. While BP has the highest at 81.8% with 10 hidden neurons and 500 training iterations. Also PNN has the highest at 96.2% with smoothing parameter 0.1.

Performance of classification in each pattern are compared and presents in Table 4.13 and Figure 4.6, respectively. The results show that the LVQ has a maximum accuracy on pattern no.2 with 98.4%, but a minimum accuracy on pattern no.4 with 82.4%. While BP also shows a maximum accuracy on pattern no. 2 with 94.2% and a minimum accuracy is pattern no. 6 with 58.4%. The last model, PNN has a maximum accuracy on pattern no.5 with 99.4% and has a minimum accuracy on pattern no.3 with 92.4%. The studied results show that each NN model has a different pattern. The maximum accuracy can be found in pattern no.2 on both LVQ and BP.

Furthermore, when comparing the highest difference of accuracy in each pattern among three NN models, results show that: pattern no.4 is the highest different accuracy between PNN and LVQ with 13.2%. On the other hand, pattern no.6 is the highest different accuracy on both PNN versus BP with 38.6% and LVQ versus BP with 28.4%.

4.8 Training time comparison

This experiment, we compare training time of each NN model based on the best accuracy. Results on Table 4.14 show that, LVQ uses very long time with 120 minutes. BP uses 3 minutes for training time while PNN uses only 2 minutes. PNN is the fastest time because PNN algorithm does not need training iteration.

Table 4.14: Training time comparison

Training time (minutes)		
LVQ	**BP**	**PNN**
120	3	2

Based on whole of simulation experiment on this thesis, it can be concluded that the FH failure pattern classification on PNN yields the best performance with an average accuracy of 96.2%. The PNN best performance is higher than LVQ 5.5% and BP 14.4%, while LVQ with an average accuracy of 90.7% that is higher than BP 8.9%. Where BP yields the worst performance with average accuracy only 81.8%.

In terms of training time, PNN is faster than BP, while LVQ uses the longest time. Therefore, we propose PNN model which has the best accuracy and fastest training time to be applied for pre-analyzing failure from FH measurement.

Chapter 5

CONCLUSIONS

In this thesis, we apply the Neural Network (NN) based on three models for Fly Height (FH) failure pattern classification in Hard Disk Drive (HDD) manufacturing process. Three NN models that widely used in pattern recognition field, it consists of Learning Vector Quantization (LVQ), Back Propagation (BP), and Probabilistic neural network (PNN) algorithms. There are six types of failure patterns from FH measurement were chosen for classification. Based on experience of Failure Analysis (FA) results, each failure pattern can indicate three causes of failure which are physical damage on Head Gimbal Assembly (HGA), contamination on slider Air Bearing Surface (ABS), and head instability.

In experiment, we evaluated the performance of these NN models with the various parameters that affect to classification performance of the network, which are: number of training iterations, training samples, hidden neurons, and smoothing parameters. Based on the simulation experiment, it shows that the classification accuracy of PNN is the best when compared to LVQ and BP algorithm. We found that the PNN can achieve the highest an average accuracy at 96.2% while LVQ reaches at 90.7%. Otherwise, BP has the worst performance with the highest average accuracy only 81.8%. In terms of training time, PNN is the fastest. Therefore, we propose PNN model which has the best accuracy and fastest training time to be applied for pre-analyzing failure from FH measurement. FA engineers can take the advantage from this thesis to minimize time to analysis FH failure pattern in HDD manufacturing.

REFERENCES

[1] H. Dakroub, M. C. Rao and A. Gay Sam, "Fly height measurement for a disc drive," *US Patent: US6898034, Assignee: Seagate Technology LLC*, Publication date on May 24, 2005.

[2] B. C. Schardt, E. Schreck and R. Sonnenfeld, "Flying Height Measurement while Seeking in Hard Disk Drives," *IEEE transactions on Magnetics*, vol.34, no.4, July 1998.

[3] U. Boettcher, C. A. Lacey, H. Li, K. Amemiya, R. A. de Callafon and F. E. Talke, "Servo signal data processing for flying height control in hard disk drives," *Microsystem Technologies Journal*, vol. 17, Issue 5-7 , pp 937-944, Jun 2012.

[4] B. D. Strom, S. C. Lee, G. W. Tyndall and A. Khurshudov, "Hard Disk Drive Reliability Modeling and Failure Prediction," *IEEE transactions on Magnetics*, vol. 43, no. 9, Sep 2007.

[5] J. Y. Juang, T. Nakamura, B. Knigge, Y. Luo, W. C. Hsiao, K. Kuroki, F. Y. Huang and P. Baumgart, "Numerical and Experiment Analyses of Nanometer-Scale Flying Height Control of Magnetic Head with Heating Element," *IEEE transactions on Magnetics*, vol. 44, no. 11, Nov 2008.

[6] Nie Jianbin, "Control Design and Implementation of Hard Disk Drive Servos," *University of California, Berkeley*, 2011.

[7] A. Zilouchian and M. Jamshidi, "Intelligent Control Systems Using Soft Computing Methodologies," *ISBN: 0-8493-1875-0*, 2001.

[8] Laurene Fausett, "Fundamentals of Neural Networks: Architectures, Algorithms And Applications," *ISBN: 978-0133341867*, 1993.

[9] M. T. Hagan, H. B. Demuth and M. H. Beale, “Neural Network Design,” *ISBN: 978-0971732100*, 2002.

[10] H. B. Demuth, M. H. Beale and M. T. Hagan, “Neural Network Toolbox 6 User’s Guide,” *MathWorks, Inc*, 2008.

[11] Raul Rojas, “Neural Networks A Systematic Introduction,” *ISBN: 978-3642610684*, 1996.

[12] Mirza Cilimkovic, “Neural Networks and Back Propagation Algorithm,” *www.dataminingmasters.com*

[13] Al-Sakib Khan Pathan, “The State of the Art in Intrusion Prevention and Detection,” *ISBN: 978-1482203516*, 2014.

[14] Donald F. Specht, “Probabilistic Neural Networks”, *Neural Networks*, vol. 3. pp. 109-118, 1990.

[15] S. J. Chuang, S. R. Zeng and Y. L. Chou, “Neural Networks for the Recognition of Traditional Chinese Handwriting,” *14th IEEE International Conference on Computational Science and Engineering (CSE2011)*, Aug 2011.

[16] N. B. A. Mustafa, K. Arumugam, S. K. Ahmed and Z. A. M. Sharrif, “Classification of Fruits using Probabilistic Neural Networks - Improvement using Color Features,” *2011 IEEE Region Ten Conference (TENCON 2011)*, Nov 2011.

[17] M. F. Othman and M. A. M. Basri, “Probabilistic Neural Network For Brain Tumor Classification,” *2011 International Conference on Intelligent Systems, Modelling and Simulation (ISMS 2011)*, Jan 2011.

[18] L. He, W. Hou, X. Zhen and C. Peng, “Recognition of ECG Patterns Using Artificial Neural Network,” *6th International Conference on Intelligent Systems Design and Applications (ISDA 2006)*, Oct 2006.

[19] A. Mehra, M. Kumawat, R. Ranjan, B. Pandey, S. Ranjan, A. Shukla and R. Tiwari "Expert System for Speaker Identification using Lip features with PCA," *2nd International Workshop on Intelligent Systems and Applications (ISA 2010)*, May 2010.

[20] Q. Wang, Q. Yang and M. Zhang, "Analyzing financial distress of listed companies using Neural Network," *3rd International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM 2012)*, Oct 2012.

[21] N. Qiakai, G. Chao and Y. Jing, "Research of Face Image Recognition Based on Probabilistic Neural Networks," *24th Chinese Control and Decision Conference (CCDC 2012)*, May 2012.

[22] J. A. Calderon, G. Z. Madrigal and D.A.O. Carranza, "Comparative Analysis between Models of Neural Networks for the Classification of Faults in Electrical Systems," *4th Congress of Electronics, Robotics and Automotive Mechanics (CERMA 2007)*, Sep 2007.

[23] F. Yan, W. Liu and L. Tian, "A New Neural Network Approach for Fault Location of Distribution Network," *International Conference on Mechatronic Science, Electric Engineering and Computer (MEC 2011)*, Aug 2011.

[24] X. Shen and J. Chen, "Study on Prediction of Traffic Congestion Based on LVQ Neural Network," *International Conference on Measuring Technology and Mechatronics Automation (ICMTMA 2009)*, Apr 2009.

[25] T. G. Amaral, O. P. Dias, A. Wolczowski and V. F. Pires, "Neural Network Based Identification of Hand Movements Using Biomedical Signals," *IEEE 16th International Conference on Intelligent Engineering Systems (INES 2012)*, Jun 2012.

[26] S. Karsoliya, “Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture,” *International Journal of Engineering Trends and Technology*, Volume 3, Issue 6, 2012.

APPENDIX A

PUBLICATION

This research has been published and presented in the International Conference on Engineering, Applied Sciences, and Technology (ICEAST 2013) at the Sukosol, Bangkok, Thailand during August 21-24, 2013.

ICEAST 2013 Technical Program

Thursday August 22, 2013					
9:20-9:50	Opening Ceremony Room: Kamolthip 2&3				
9:50-10:35	Keynote Speech 1: Some Consideration on Applied Superconductivity to Innovative Energy Devices Prof. Dr. Yasuyuki Shirai, Kyoto University Room: Kamolthip 2&3				
10:35-11:00	Coffee Break				
11:00-11:45	Keynote Speech 2: Toward VLSI Reliability Enhancement by Reconfigurable Architecture Prof. Dr. Takao Onoye, Osaka University Room: Kamolthip 2&3				
11:45-13:00	Lunch				
13:00-14:40	MS-1 Room: Duangkamol Materials for Energy Conversion I Chair: Wisanu Pecharapa ID: 139, 39, 122, 74, 126	MS-2 Room: Kamolruedi Nano Materials/Synthesis and Application Chair: Naratip Vittayakorn ID: 28, 104, 137, 75, 135	ES-1 Room: Kamolthip 2 Smart Energy Device Chair: Chanin Bunlaksananusorn ID: 148, 71, 147, 161, 76	IS-1 Room: Kornkamol 1 ICT Applications 1 Chair: Attasit Lasakul ID: 21, 27, 58, 61, 88	IS-2 Room: Kornkamol 2 ICT Applications 2 Chair: Pipat Prommee ID: 89, 92, 94, 105, 107
15:40-15:00	Coffee Break				
15:00-17:00	MS-3 Room: Duangkamol Environmental Friendly Materials/Biomaterials Chair: Supanit Porntheeraphat ID: 63, 99, 116, 48, 81, 106	MS-4 Room: Kamolruedi Materials Simulation/Characterization Chair: Tossawat Seetawan ID: 120, 78, 95, 40, 158, 141	ES-2 Room: Kamolthip 2 Smart Renewable Energy Chair: Worawat Nakawiro ID: 162, 166, 170, 150, 31, 152	IS-3 Room: Kornkamol 1 ICT Applications 3 Chair: Taweepol Suesut ID: 22, 35, 38, 98, 101, 102	AS-1 Room: Kornkamol 2 Agricultural and Food Science Technology Chair: Panmanas Sirisomboon ID: 47, 77, 165, 167, 25, 103
18:30-21:00	Banquet Room: Kamolthip 2&3				

Friday August 23, 2013					
9:00-9:40	Keynote Speech 3: Photocatalytic Performance of ZnO-SnO2 Nanocomposites Synthesized by Mechanochemical Routes Prof. Dr. Keiichi Ishihara, Kyoto University Room: Duangkamol				
9:40-10:20	Keynote Speech 4: Robots that Can Attend People Prof. Dr. Jun Miura, Toyohashi University of Technology Room: Duangkamol				
10:20-10:40	Coffee Break				
10:40-12:00	MS-5 Room: Duangkamol Materials for Energy Conversion II Chair: Anek Charoenphakdee ID: 129, 29, 66, 123	MS-6 Room: Kamolruedi Magnetic/Electroceramic Materials I Chair: Chesta Ruttanapun ID: 59, 117, 93, 46	MS-7 Room: Kingkamol Material Processing Chair: Sorapong Pavasupree ID: 79, 108, 112, 114, 97	ES-3 Room: Kornkamol 1 Applied Energy Chair: Pongjet Promvonge ID: 140, 154, 155, 70, 164	IS-4 Room: Kornkamol 2 ICT Applications 4 Chair: Worawat Choensawat ID: 51, 62, 144, 145, 171
12:00-13:00	Lunch				
13:00-14:40	MS-8 Room: Kamolruedi Composite and Functional Materials Chair: Keiichi Ishihara ID: 156, 110, 118, 33, 67	MS-10 Room: Duangkamol Magnetic/Electroceramic Materials II Chair: Worawut Makcharoen ID: 32,124, 30, 125, 80	MS-9 Room: Kingkamol Thin Films Technology Chair: Wisanu Pecharapa ID: 60, 42, 130, 83, 34	IS-5 Room: Kornkamol 1 ICT Applications 5 Chair: Pakorn Watanachaturaporn ID: 54, 55, 56, 82, 84	IS-6 Room: Kornkamol 2 ICT Applications 6 Chair: Don Isarakorn ID: 24, 41, 50, 149, 169
14:40-15:00	Coffee Break				
15:00-17:00	MS-11 Room: Duangkamol Materials Characterization Chair: Winadda Wongwiriyapan ID: 160, 49,121, 138, 109, 159	MS-12 Room: Kamolruedi Materials Technology and Device Chair: Sirapat Pratontep ID: 64, 68, 86, 113, 128, 142	-	IS-7 Room: Kornkamol 1 ICT Applications 7 Chair: Worapong Tangsrirat ID: 53, 69, 96, 132, 136, 151	IS-8 Room: Kornkamol 2 ICT Applications 8 Chair: Don Isarakorn ID: 65, 91,119, 153, 168

International Conference on Engineering, Applied Sciences, and Technology
August 21-24, 2013, The Sukosol, Bangkok, Thailand

PID : 00089

Comparative Analysis between BP and LVQ Neural Networks for the Classification of Fly Height Failure Patterns in HDD Manufacturing Process

Thanapong Thanasarn
King Mongkutâ€™s Institute of Technology Ladkrabang, Thailand
Chanon Warisarn
King Mongkutâ€™s Institute of Technology Ladkrabang, Thailand

Keywords:
LVQ Neural Network, BP Neural Network, Hard Disk Drive, Fly Height.

Abstract:

In this paper, Learning Vector Quantization neural network (LVQ) and Back Propagation neural network (BP) algorithms were used for fly height failure pattern classification in the manufacturing process of Hard Disk Drive (HDD). Six types of failure patterns were chosen from fly height measurement to be recognized, which including three causes of failure are the poor mechanic on Head Gimbal Assembly (HGA), the contamination on slider Air Bearing Surface (ABS), and head instability. In experiment, data from fly height failure are used to learn the pattern recognition for both neural network algorithms. The simulation results show that the classification performance of fly height failure patterns based on LVQ neural network is better than BP neural networks, LVQ can achieve the best of overall accuracy at 90.7% while BP reaches only 81.8%.

International Conference on Engineering, Applied Sciences, and Technology
August 21-24, 2013, The Sukosol, Bangkok, Thailand

Comparative Analysis between BP and LVQ Neural Networks for the Classification of Fly Height Failure Patterns in HDD Manufacturing Process

Thanapong Thanasarn
International College
King Mongkut's Institute of Technology Ladkrabang
(KMITL), Bangkok, Thailand
thanapong.thanasarn@seagate.com

Chanon Warisarn
College of Data Storage Innovation
King Mongkut's Institute of Technology Ladkrabang
(KMITL), Bangkok, Thailand
kwchanon@kmitl.ac.th

***Abstract*— In this paper, Learning Vector Quantization neural network (LVQ) and Back Propagation neural network (BP) algorithms were used for fly height failure pattern classification in the manufacturing process of Hard Disk Drive (HDD). Six types of failure patterns were chosen from fly height measurement, to be recognized, including three causes of failure: poor mechanics on Head Gimbal Assembly (HGA), contamination on slider Air Bearing Surface (ABS), and head instability. In the experiment, data from fly height failure was used to learn the pattern recognition for both neural network algorithms. The simulation results show that the classification performance of fly height failure patterns based on LVQ neural network is better than BP neural networks; LVQ can achieve the best of overall accuracy at 90.7% while BP reaches only 81.8%.**

Keywords - Learning Vector Quantization Neural Network; Back Propagation Neural Network; Hard Disk Drive; Fly Height.

I. INTRODUCTION

In the manufacturing process of Hard Disk Drive (HDD), after assembly process, HDD requires functional testing to calculate all of the drive operations, such as the calculation of areal density with bit per inch (BPI) and track per inch (TPI), the measurement of bit error rate (BER) performance, the MR bias calculation, and the fly height measurement, etc.

Fly height testing is the measurement of the spacing distance between a recording head and a media by applying voltage values into a heater element of the recording head until the recording head touchdown on the media and also provides the heater voltage values in each data zone that are appropriate to perform read and write operations [1], [2]. In general, the failure analysis on failed heads from fly height measurement must be analyzed using the fly height profile to classify the causes of failure. It was observed that each fly height profile can indicate the cause of each problem. For example, poor mechanics on Head Gimbal Assembly (HGA), found contamination on the slider Air Bearing Surface (ABS), and a head instability problem. Currently, however, fly height profile failure is investigated by failure analysis engineers over a long period of time. Therefore, ways to reduce the engineer's work with failure analysis are needed for the HDD manufacturing process to be more efficient.

Recently, many researchers have applied neural network models for pattern recognition in several fields, such as: face image recognition, the speaker identification, the classification of faults in electrical systems, and the recognition of electrocardiogram (ECG) patterns [3]-[12]. However, neural network models have never been applied with fly height failure pattern classification in the manufacturing process. Therefore, this paper proposes two classification methods by using Learning Vector Quantization neural network (LVQ) and Back Propagation neural network (BP) to support failure analysis.

The paper is organized as follows: After presentation of the system structure for the classification of fly height failure patterns of LVQ and BP model in Section-II, Simulation results are given in Section-III. Finally, Section-IV is a conclusion of the results.

II. METHODOLOGY

In this work, the spacing distance between recording head and media are considered in terms of the voltage values that are applied into the heater element of the recording head. It affects the writer and reader element protrude until the contact area touchdown on the media, which is detected by the servo detector system. Data is received from the fly height measure for this study, including heater voltage values in the Digital to Analog Converter (DAC) unit for both read and write operations in each location of the data zone on the media.

Nevertheless, to reduce the complexity of input datasets, only data from write operation for pattern classification were chosen. The six types of failure pattern that are found on failed heads are shown in Figure 1. The x-axis is the heater voltage (in DAC unit) and the y-axis is the data zone of the media in order of the outer disk towards the inner disk, which are composed of 10 data points for each head. In this study, data was collected from failure analysis results, which have been distinguished by type of pattern for 1,000 samples per type. Then, there are 6,000 failure samples to supervise networks and test classification accuracy between LVQ and BP neural networks.

International Conference on Engineering, Applied Sciences, and Technology
August 21-24, 2013, The Sukosol, Bangkok, Thailand

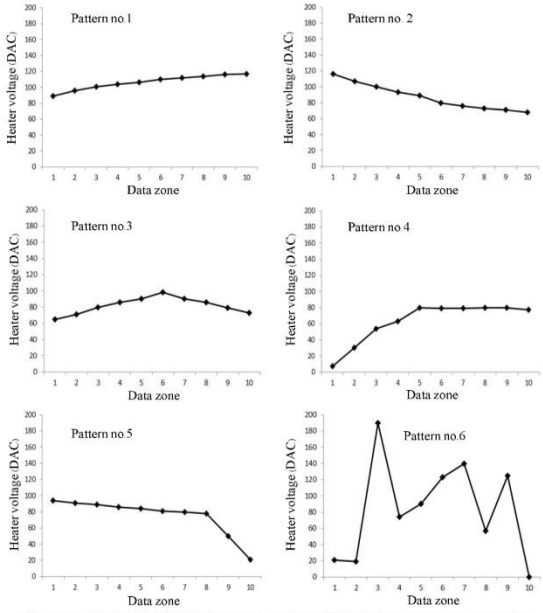


Figure 1. Six types of failure patterns from fly height measurement to be recognized

Artificial neural networks work in two parts as presented in Figure 2: one is the training part and another is the testing part. In the training part, the connection weights are automatically adjusted to map the input to the corresponding output. Two different neural network models, which are LVQ and BP networks, were operated to recognize the fly height failure patterns.

A. Feature normalization

In order to deal with appropriate data, in the preprocessing step, before training and testing the network, all of the input data are normalized into a range of 0 and 1 with formula (1)

$$pn_k = (p_k - \min p) / (\max p - \min p), \quad (1)$$

where pn_k is the k^{th} data point that is normalized between 0 and 1, p_k is the k^{th} data point, min p is the minimum value in a pattern, and max p is the maximum value in a pattern.

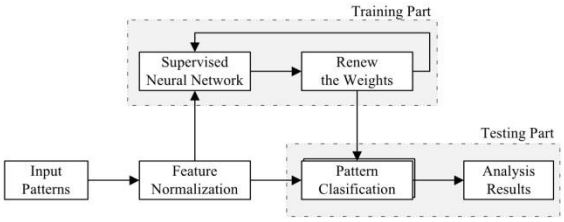


Figure 2. System block diagram for the classification of fly height failure patterns using neural networks

B. LVQ neural network model

LVQ neural network is a kind of supervised feed forward neural network. It is composed of three layers of neurons: input layer, competitive layer, and linear output layer as shown in Figure 3. The linear output layer transforms the classes of the competitive layer into the specified target classifications that are defined by the user. The input layer connects with each neuron of the competitive layer, but the neurons of the competitive layer are connected with a neuron of the linear output layer and the neurons in the competitive layer and the linear output layer produce the binary output values which are 0 and 1.

When the input patterns are entered into the network, all of the units at the input layer are connected to all of the neurons at the linear output layer with connection weight (W_{ij}), which means the connection weight from the unit j in the input layer to unit i in the linear output layer. The linear output layer will generate only one winner neuron which is the neuron with the minimum distance between an input vector and its connection weight vector. The winner neuron is allowed to produce '1' which indicates the class of input pattern and other neurons are constrained to produce '0' [5]-[8]. The model of the LVQ neural network that is used in this study is shown in Figure 4, where R is the number of input vector, S^1 is the number of neurons in the competitive layer, and S^2 is the number of neurons in the linear output layer.

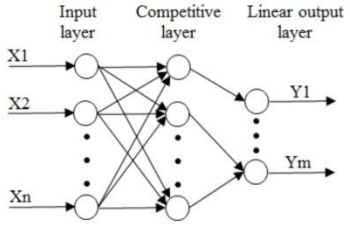


Figure 3. Basic structure of LVQ neural network

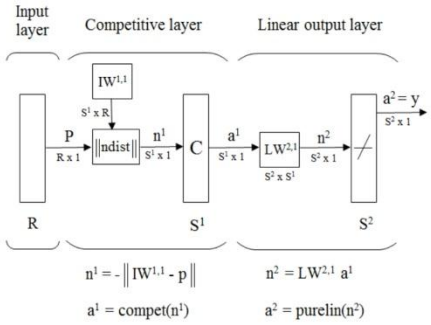


Figure 4. The model of LVQ neural network

International Conference on Engineering, Applied Sciences, and Technology
August 21-24, 2013, The Sukosol, Bangkok, Thailand

C. *BP neural network model*

BP neural network is one of the supervised multilayer feed forward neural networks, and is a popular model used in pattern recognition. The basic structure of the BP neural network is shown in Figure 5; it consists of input layer, hidden layer, and output layer.

Overview of BP algorithm as shown in Figure 6, it includes two calculations. The first is the forward calculation and the second is the error back calculation. In the process of forward calculation, the input data from input layer pass through the hidden layer processing and transmit to output layer. If output layer get incorrect output result from the target, it will be transferred back propagation and re-calculate the weights of each neuron. After correct the weights of neurons, the network get the output in consistency with the transfer form again, the actual output and expected error will lead to a new weight correction with the minimum error [7],[12].

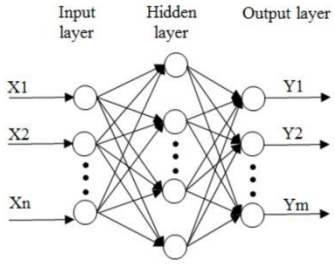


Figure 5. Basic structure of BP neural network

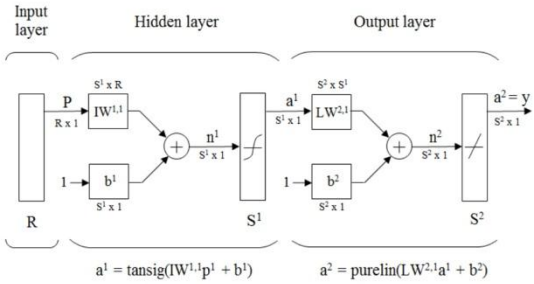


Figure 6. The model of BP neural network

III. EXPERIMENT AND RESULTS

A. *Pattern classification by LVQ neural network*

The whole of the data from the 1,000 samples per one type of failure pattern were separated into two groups with an equal number of six patterns. A first half dataset of 500 in each pattern is used for network training and the last group of 500 for network testing. Therefore, the input data with 10×3000 array of six patterns will be fed into the network for training in the first step and network testing in the second step. Finally, the setting conditions are determined for experimentation of the LVQ neural network as follows:

- Study on various number of hidden neurons:
  - 10, 20, 30, 40, 50
- Study on various number of learning iteration (epochs):
  - 500, 1000
- Learning rate: 0.01
- Transfer function for the hidden layer:
  - Competitive transfer function
- Transfer function for the output layer:
  - Linear transfer function

TABLE I.
PERFORMANCE OF CLASSIFICATION BY LVQ NEURAL NETWORK

Number of hidden neurons	Overall accuracy	
	Epochs 500	Epochs 1000
10	88.8%	89.0%
20	89.6%	89.5%
30	87.8%	88.1%
40	89.0%	88.0%
50	90.7%	90.2%

TABLE II.
PERFORMANCE OF CLASSIFICATION BY BP NEURAL NETWORK

Number of hidden neurons	Overall accuracy	
	Epochs 500	Epochs 1000
10	81.8%	80.1%
20	79.8%	80.8%
30	79.1%	78.2%
40	74.9%	78.3%
50	78.5%	77.9%

The simulation results from the LVQ neural network are shown in Table I. The best overall accuracy is 90.7% with 50 hidden neurons and 500 epochs. It means that LVQ requires the maximum number of hidden neurons in this experiment for the highest precision possible.

B. *Pattern classification by BP neural network*

In order to compare the classification accuracy between both the LVQ and the BP neural networks, the same training and testing sample sets, the number of hidden neurons, the learning rate and the number of learning iteration from experiment of LVQ are selected for this experiment and the linear transfer function is also used for the output layer. Except the transfer function for hidden layer is Tan-sigmoid transfer function which differences from LVQ setting. The simulation results from the BP neural network are shown in Table II. The best overall accuracy is 81.8% with 10 hidden neurons and 500 epochs. Therefore, the BP neural network needs only 10 hidden neurons to be suitable for best precision.

C. *Comparison performance of pattern classification between LVQ and BP neural networks*

As shown in table III. the LVQ neural network shows the best overall accuracy at 90.7% with 50 hidden neurons and 500 epochs, where the precision has higher accuracy than the BP neural network, which yields its best overall accuracy at 81.8% with 10 hidden neurons and 500 epochs.

TABLE III.
PERFORMANCE OF PATTERN CLASSIFICATION BETWEEN LVQ AND BP NEURAL NETWORK

Type of neural network	The best of overall accuracy	Number of hidden neurons	Number of epochs
LVQ	90.7%	50	500
BP	81.8%	10	500

TABLE IV.
PERFORMANCE OF PATTERNS CLASSIFICATION BETWEEN LVQ AND BP NEURAL NETWORKS IN EACH PATTERN

Pattern	Accuracy		Delta
	LVQ	BP	
1	91.4%	81.2%	10.2%
2	98.4%	94.2%	4.2%
3	87.4%	80.0%	7.4%
4	82.4%	83.6%	-1.2%
5	97.8%	93.2%	4.6%
6	86.8%	58.4%	28.4%
Overall	90.7%	81.8%	8.9%

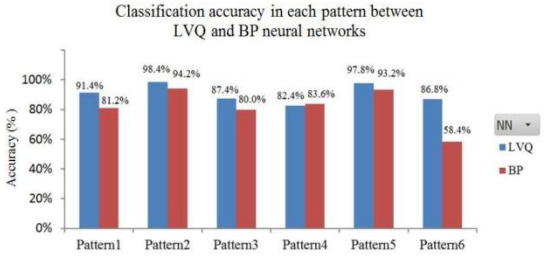


Figure 7. Classification accuracy in each pattern between BP and LVQ

Performance of classification in each pattern was also compared and is shown in Table IV and Figure 7. The results show that the LVQ neural network yields its maximum accuracy on pattern no.2 of 98.4% and has the minimum accuracy on pattern no.4 of 82.4%, while the BP neural network also shows its maximum accuracy on pattern no. 2 of 94.2%. However, the minimum accuracy is pattern no. 6 with 58.4%, which is a different type of failure pattern when compared with LVQ.

Furthermore, when comparing the difference in accuracy of levels in each pattern between LVQ and BP neural networks, results show that the most different accuracy level is 28.4% on pattern no.6 where the LVQ is better than the BP neural network. On the other hand, only the classification of pattern no.4 on BP is slightly more precise than the LVQ neural network of 1.2% as shown in Figure 7.

IV. CONCLUSIONS

Learning Vector Quantization neural network (LVQ) algorithm is introduced for fly height failure pattern classification in the manufacturing process of Hard Disk Drives (HDD). Through the simulation experiment, it is shown that the classification accuracy of LVQ is the best when compared to Back Propagation neural network (BP) algorithm. We found that LVQ can achieve the best overall accuracy at 90.7% while BP reaches its best overall accuracy at only 81.8%. Moreover, this model can also be applied to pre-analyze failed drives from fly height measurement in HDD manufacturing for time reduction in terms of pattern verification. However, it is the hope that through future research, the other models of neural network with optimization in various parameters, as well as other features will be investigated for an improvement in classification accuracy.

ACKNOWLEDGMENT

We would like to thank Seagate Technology, NSTDA, and KMITL for supporting the Master scholarship and providing data for this study.

REFERENCES

[1] H. Dakroub, M. C. Rao and A. Gay Sam, "Fly height measurement for a disc drive," *US Patent: US6898034, Assignee: Seagate Technology LLC*, Publication date on May 24, 2005.

[2] B. C. Schardt, E. Schreck and R. Sonnenfeld, "Flying Height Measurement while Seeking in Hard Disk Drives," *IEEE transactions on Magnetics*, vol.34, no.4, July 1998.

[3] N. Qiakai, G. Chao and Y. Jing, "Research of Face Image Recognition Based on Probabilistic Neural Networks," *24th Chinese Control and Decision Conference (CCDC 2012)*, May 2012.

[4] Anuj Mehra et al., "Expert System for Speaker Identification using Lip features with PCA," *2nd International Workshop on Intelligent Systems and Applications (ISA 2010)*, May 2010.

[5] J. A. Calderon, G. Z. Madrigal and D.A.O. Carranza, "Comparative Analysis between Models of Neural Networks for the Classification of Faults in Electrical Systems," *4th Congress of Electronics, Robotics and Automotive Mechanics (CERMA 2007)*, September 2007.

[6] L. He, W. Hou, X. Zhen and C. Peng, "Recognition of ECG Patterns Using Artificial Neural Network," *6th International Conference on Intelligent Systems Design and Applications (ISDA 2006)*, Oct. 2006.

[7] F. Yan, W. Liu and L. Tian, "A New Neural Network Approach for Fault Location of Distribution Network," *International Conference on Mechatronic Science, Electric Engineering and Computer (MEC 2011)*, August 2011.

[8] X. Shen and J. Chen, "Study on Prediction of Traffic Congestion Based on LVQ Neural Network," *International Conference on Measuring Technology and Mechatronics Automation (ICMTMA 2009)*, April 2009.

[9] T. G. Amaral, O. P. Dias, A. Wolczowski and V. F. Pires, "Neural Network Based Identification of Hand Movements Using Biomedical Signals," *IEEE 16th International Conference on Intelligent Engineering Systems (INES 2012)*, June 2012.

[10] J. Wang and Y. Wen, "Application of Genetic LVQ Neural Network in Credit Analysis of Power Customer," *4th International Conference on Natural Computation (ICNC 2008)*, Oct. 2008.

[11] M. Zheng and Q. Liu, "Application of LVQ Neural Network to Car License Plate Recognition," *International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2010)*, Nov. 2010.

[12] X. Ren and X. Hou, "The Misalignment Fault Model Building for Rotating Machinery Rotor Based on BP Network," *8th International Conference on Natural Computation (ICNC 2012)*, May 2012.

AUTHOR BIOGRAPHY

Name-Surname: Mr. Thanapong Thanasarn

Date of Birth: February 25th, 1986

Present Address: Seagate Technology (Thailand) Ltd. 90 Mittraphap road, Sungnoen, Nakhonratchasima, Thailand 30170

Education: 2004-2007: Bachelor's degree in Science, Applied Physics , King Mongkut's Institute of Technology Ladkrabang

Scholarship: 2010-2011: Scholarship for study in Master of Engineering in Data Storage Technology (English program) by NSTDA, KMITL and Seagate Technology (Thailand) Ltd.

Publication: Thanapong T., Chanon W., "Comparative Analysis between BP and LVQ Neural Networks for the Classification of Fly Height Failure Patterns in HDD Manufacturing Process", ICEAST 2013 (International Conference on Engineering, Applied Sciences, and Technology) at the Sukosol, Bangkok, Thailand during August 21-24, 2013.

Experience:

2008 - Present Seagate Technology (Thailand) Ltd.

- Product Engineer