# Breast Cancer Diagnosis Using Artificial Neural Networks


**By**

**Chen Chen, MComp**


A dissertation submitted to the

School of Computing

in partial fulfilment of the requirements for the degree of

**Master of Computing**


**University of Tasmania**


**November 2009**

# Declaration

I, Chen Chen, declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any tertiary institution. To my knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

.........................

# Abstract

Breast Cancer is one of the most dangerous diseases for women. Mammography is an effective method in early detection. However, there are difficulties in accurate analysis of some mammogram images. Therefore, a method of data analysis using artificial neural networks (ANNs) has been developed.

In this thesis, the performances on the Wisconsin breast cancer data (WBCD) of three different neural network models: Multi-layer neural networks (MLPs), Trigonometric Neural Networks (TNNs), and Exponential Neural Networks (ENNs) are examined. These models are based on a back propagation algorithm, with different activation functions. The activation function is one of most factors to influence the performance of ANNs. The purpose of thesis is to test the hypothesis that the performance of TNNs and TNNs on breast cancer dataset is better than MLPs.

The strategic experiments are implemented. The overall performances of three models are evaluated and discussed through an analysis of four aspects of testing results: correctness rate, root mean squared error, training speed and misclassification cost. Moreover, from the testing results, the basis of further work is formed.

# Acknowledgements

Firstly, I would like to thank my supervisor Dr Shuxiang Xu for positive feedback and patience in answering my questions, even if I asked the questions many times.

Moreover, I would like to thank my family for their support which makes me concentrate on my research.

I would also like to thank the School of Computing staff. They provided a guide when I met problems.

Furthermore, I give a big thankyou to two EnglishAssist advisors: Louise Oxley and Morag Porteous in English Language Centre, they helped me to fix problems of English when I was writing this thesis. I would especially like to thank Morag for additional assistance.

I would like to thank my classmates Sheng and Ming, they gave me advice when I met problems.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

Breast cancer is the major cause of death among all cancers for women (Liao, Wang& Yi 2005, p.790). Early detection and treatment are helpful to save patient's life, where screening mammography is considered to be the most effective tool (American Cancer Society 2003). With the rapidly increasing amount of relevant data, computer-aided diagnosis (CAD) systems have emerged as useful in assisting doctors (Vouros & Panayiotopoulos 2004, p.268). In addition, the data analysis method is becoming popular in many areas, which are Artificial Neural Networks (ANNs) (Ciampi & Zhang 2002). Therefore, the CAD systems using Artificial Neural Networks has been a focus in medicine field (Vouros & Panayiotopoulos 2004, p.268).

Artificial Neural Networks (ANNs) provide insight into how biological neural processing by retaining enough of the structure observed in the brain (Dayhoff 1990, p.1). It can be used to solve lots of problems (Tipton, Krause & Corporation, 2008). Nowadays, many types of ANNs have been used to deal with a variety of problems such as Hopfield networks, Radial basis function (RBF) networks and Multi-layer neural networks (MLPs) that are focused on in my research. MLP are the common type of neural network employed in process modeling (Goel, Saxena & Bhanot 2006). Traditional perceptrons are feed-forward neural network with a supervised learning algorithm. Because the limitation of the simple perception is that it can only be classify linearly separable pattern (Suzuki et al. 2000. p132). Therefore, the development of multilayer perceptrons and the back error propagation algorithm conquer the limitation of a simple perception (Suzuki et al. 2000. p132). MLPs offer an arrangement for ANNs implementation and have been successfully applied for adaptive identification and control of variety of nonlinear processes (Goel, Saxena &

Bhanot 2006).

There exist two NN models: Trigonometric Neural Networks (TNNs) and Exponential Neural Networks (ENNs). The two models are based on a back propagation algorithm. TNNs and ENNs differ from MLPs in the use of the neuron activation functions. The Trigonometric Neural Networks (TNNs) use a trigonometric function as the neuron activation function. Exponential Neural Networks (ENNs) use an exponential function as the neuron activation function.

Neuron activation functions have significant impact on the performance of ANNs. It is the important feature of the logical operations within the architecture (Tipton, Krause & Corporation, 2008).

## 1.1 Thesis Hypothesis

It has been reported that Trigonometric Neural Networks (TNNs) and Exponential Neural Networks (ENNs) outperformed Multilayer perceptrons (MLPs) in handling financial data with high order non-linearity. Therefore, in this research, one of function of TNNs and ENNs which has been shown to have good performance in other areas will be chosen as a activation function to be compared with MLPs using traditional function (sigmoid function) that is also called back propagation neural networks (BPNN) in this thesis , in testing using different neural networks architectures. The data to be tested will be based on breast cancer dataset.

The purpose of this thesis is to explore the advantages and disadvantages of these types of ANNs in terms of training speed, architectures, and performance capabilities in breast cancer diagnosis.

In the thesis, there are six chapters. The first two chapters include the introduction and background information on work related to diagnosis breast cancer, an overview of artificial neural networks, and a review of work using different types of artificial neural networks. Moreover, the methodology will be presented. The next chapter analyzes and discusses the results obtained from the experiments. Finally, the last two chapters summarize the results and discuss further work that could be done in this area.

# 2 Background

## 2.1 Overview of breast cancer diagnosis

Breast cancer is a common and dangerous disease for women. Therefore, the search for preventive methods for the disease has never stopped until now. With the development of science technology, the increasing number of areas in society is trying to make use of computer technology to solve their problems, and the field of medicine. Hence, many IT experts and engineers joined the group to find effective preventive methods (Adam & Omar 2008). By their knowledge and skills, they attempt to design and build systems using hardware or software to assist the medical practitioners or doctors (Adam & Omar 2008).

However, there is still no clear and single dominant reason in the etiologies of breast cancer (Ioanna et al. 2000). Therefore, the preventive method is still unknown, As far as it goes, the most effective method is early detection, which means if the cancerous cells are detected before spreading to other organs, the number of death is decreased amongst breast cancer patients, the survival rate for patients is more than 97% (Adam & Omar 2008) .

Mammography is a very useful tool in early detection for breast cancer diagnosis (Adam & Omar 2008). It was introduced in 1969, which is the first step in digital detecting cancerous cells (Adam & Omar 2008). From the images, early signs of having high possibility of cancerous cells could be detected such as (Figure2.1) microcalcifications, mass, architectural distortion and breast asymmetries (Adam & Omar 2008). With this information, the patient could be cured by surgical excision at the initial stage of breast cancer (Radiol, 2006).

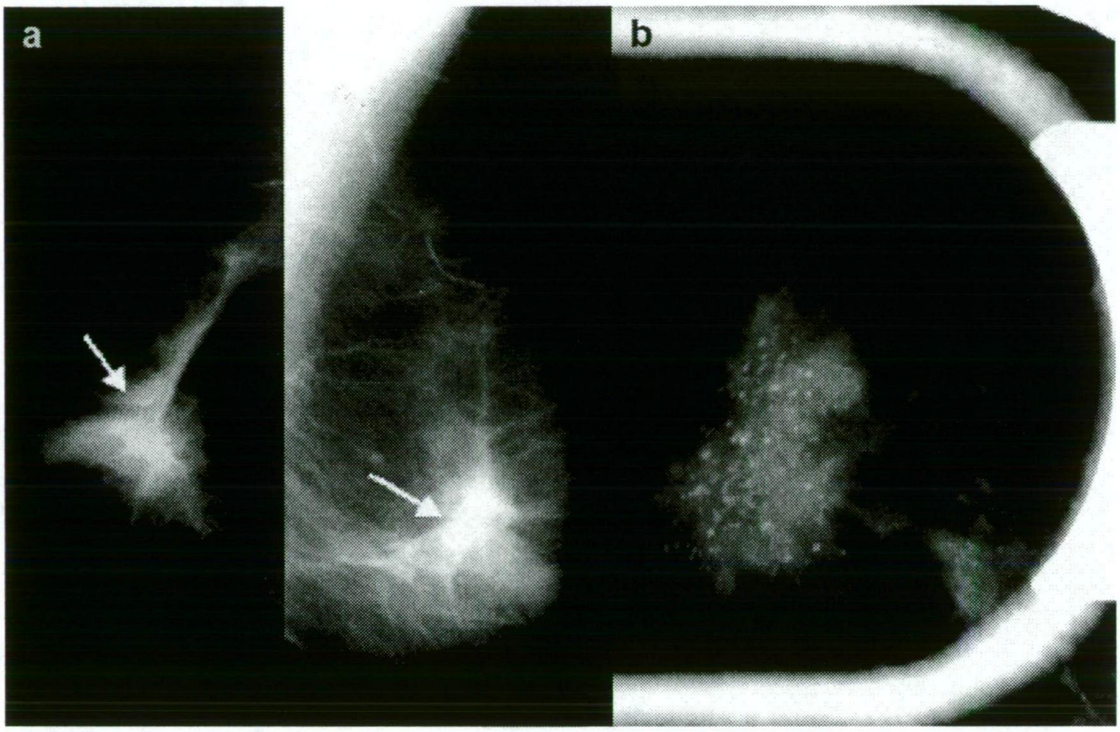



**Figure 2.1 - Mammogram displaying (a) mass (b) microcalcification (Radiol 2006)**

Therefore, it is important to analyze the difference between images of malignant and benign masses accurately, because it influences patient management and choice of treatment (Radiol 2006). However, 40-50% cases can not be analyzed easily, because the features of the calcifications and their cluster are classified as indeterminate or equivocal (Radiol 2006). Hence, a method is desired for enhancing the accuracy of analysis of the Mammogram images.

In past years, artificial neural networks have been popular tool for data analysis in a variety of fields of science and technology (Ciampi & Zhang 2002). Recently, they have been utilized to analyze data from a variety of human clinical studies. The data of breast cancer was also tried to analyze using artificial neural networks (Ciampi & Zhang 2002).

In the next section, firstly, the concept of arterial neural networks will be described. Moreover, some types of artificial neural networks and related works for analyzing breast cancer will be presented. Especially, the back propagation (BP) neural networks are focused on in the thesis. Finally, the key points of this thesis will be discussed.

## 2.2 Overview of Artificial Neural Networks

### 2.2.1 Definition

Artificial neural networks are a computer algorithm which can learn impartment relationships from a set of data, and apply this knowledge to evaluate new cases (Radiol 2006).

Artificial neural networks are based on the observation that biological learning systems are made up of very complex webs of interconnected neurons (Mitchell 1997). Artificial neural network consists of a densely interconnected set of simple units, and every unit has a number of real-valued inputs and generates a single real-valued output (Mitchell 1997).

### 2.2.2 History of neural networks

In 1943, Warren McCulloch and Walter Pitts first provided Hypothesis about the function of neurons in their famous paper. The theory showed that neurons could be structured as a computing device with the ability to perform logic function (Stein & Ludik 1998). Hence, the concept of a neural network has changed from carrier of energy to carrier of information (Stein & Ludik 1998). However, the problem for McCulloch and Pitts was the understanding of how the mind works (Stein & Ludik 1998). In 1949, Hebb proposed a mechanism by which the association of neurons could be influenced by experience. It was a method of neurons learning (Stein & Ludik 1998). In 1961, a "learning" algorithm was presented by Edurado Caianiello, which was called mnemonic equation, and it combines in a simple way the basic concept of Hebb's learning theory (Müller 1995).

Around 1960, a specific type of neural network was developed by Frank Rosenblatt and coworkers, which was called the perceptron (Müller 1995). It was a simple model of the biological mechanisms for processing of sensory information. The perceptron contains two separate layers, an input layer and an output layer (Müller 1995).

After the 1980s, a variety of models of neural networks were appearing, such as Hopfield networks and Back-propagation networks and so on (Müller 1995). With the development of neural networks, more problems can be solved by this method.

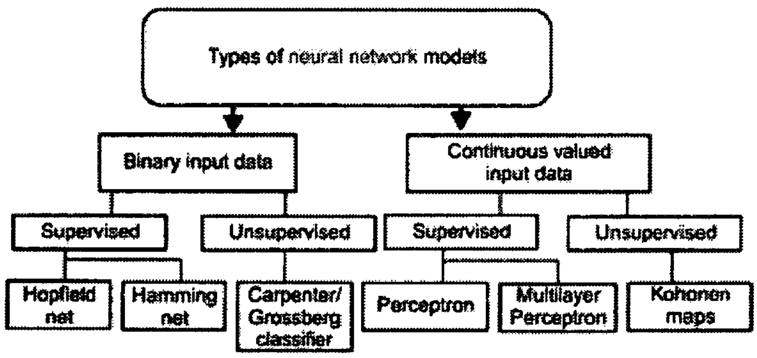



**Figure 2.2 - Classification of neural networks models**

### 2.2.3 The Biological Neuron

As mentioned above, artificial neural networks are motivated by models of a biological neural network, which are made up of billions of neurons. A single neuron may be connected with an amount of other neurons (Regan 2008). Therefore, there is a huge and complex network in a human brain (Regan 2008). For a neuron(figure 2.3), there are three parts: the output area of the neuron is called the axon, the input area of the neuron is a set of branching fibers called dendrites, and the connecting point is the synapse between an axon and a dendrite (Dayhoff, 1990) .

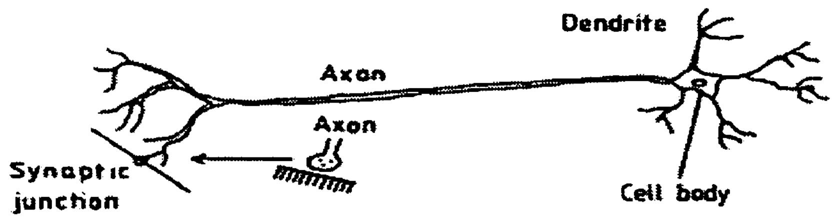



**Figure 2.3 - A biological neuron (Regan 2008)**

In a biological neural network, the electrical signals are passed to each other from the neurons. The connections are composed of axons and dendrites. Connections have different strength which could be influenced by the change of one of its neighbors from very weak to very strong (Regan 2008). Every connection has its own weight with its strength. Neurons send the output value to other neuron, and the neuron collecting connections calculates the weight through a sum of input signals, which could influence activation of other neurons to fire (Regan 2008). The output is decided by activation function (Regan 2008). The learning method for networks is based on the adjusting weights of the connections (Regan 2008). The learning process in brains basically is from the change of those connection strengths. Activities in brain are associated with the special pattern of firing activity in the networks of neurons (Regan 2008).

### 2.2.4 Single Layer Neural Networks

In 1958, a computational model of neurons was constructed by the American psychologist Frank Rosenblatt, and he called the perceptron (Mange& Tomassini, 1998). The perceptron pattern-mapping architecture learns to classify patterns through supervised learning (Dayhoff, 1990). The model is defined as follow:

y(x)=1 if $\sum_{i=1}^{n} WiXi > \theta$

y(x)=0 if $\sum_{i=1}^{n} WiXi < \theta$

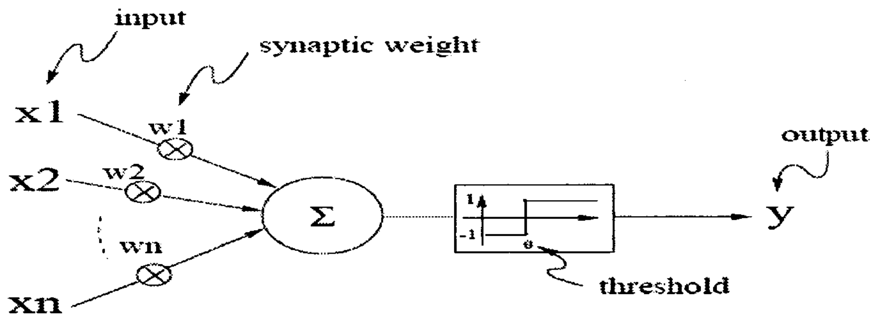



**Figure 2.4 - Rosenblatt's perceptron model of a neuron (Mange& Tomassini, 1998)**

In the model, y is the output of the perceptron, $w_i$ is the weight of input xi, and ↕ is the threshold (Mange& Tomassini, 1998). The inputs (x1; x2:::; xn) and weights (w1; w2:::;wn) are real values. If the presence of a value xi in a given input pattern tends to re the perceptron, the corresponding weight $w_i$ will be positive (Mange& Tomassini, 1998). If the value xi inhibits the perceptron, the weight $w_i$ will be negative (Mange& Tomassini, 1998) .The activation threshold can be implemented as an additional weight making it furthermore adjustable (Mange& Tomassini, 1998).

The perceptron does a weighted summation of its inputs and then outputs one or zero as a function of the weighted sum and the threshold (Mange& Tomassini, 1998) .It can be shown in the equation:

$$y(x)=g\,(\sum_{i=0}^{n} WiXi),$$

In the equation, g(wixi) is called the activation function (Mange& Tomassini, 1998) . The use of continuous activation functions instead of the simple threshold function is a straightforward way of generalizing the perceptron model to real-valued outputs. There are common activation functions such as the linear, the sigmoid and so on (Mange& Tomassini, 1998).

Basically, a single perceptron can be used to symbolize many Boolean functions such as AND, OR, NAND and NOR, which are all linearly separable (Mitchell 1997).

However, non-linear function can not be described by a single perceptron, for example, XOR function (Mitchell 1997).

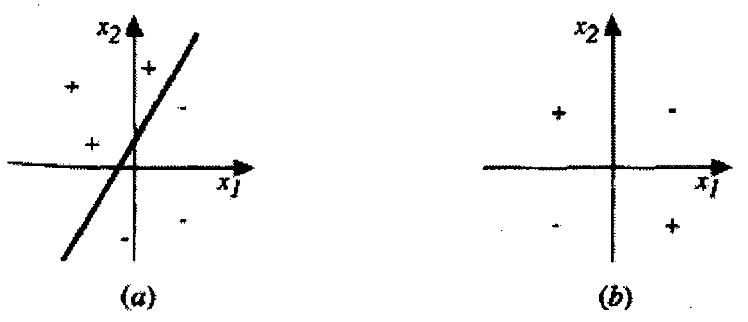



**Figure 2.5 - The perceptron as a decision surface to classify the input.**
**(a) Linearly separable decision surface which can be represented correctly**
**(b) Non-linearly separable decision surface can not be classified correctly**
**(Mitchell 1997)**

The single-layer network is the simplest form that an input layer of source nodes to an out layer (Cho 2007, p.31). The main advantage of a single layer perceptron is that can be easier train to perform in the expected way (Maurer, 1991, p.1). However, the single-layer network may not be able to make complex decisions (Jagadish & Iyengar, 2005, p.231).

### 2.2.5 Multilayer Neural Networks

Multi-layer neural networks are widely used artificial neural networks architecture (Mastinu & Miano 2006, p.107). A multilayer perceptron (MLP) contains two or more layers, each layer being separated by a layer of nodes called hidden nodes (Husmeier, Dybowski & Roberts 2005, p.62).

In an MLP, the first layer of nodes that receives the inputs is called the input layer. The layer of nodes producing the outputs is called the output layer. The hidden layers are the layers that are between the input and output layers (Husmeier, Dybowski & Roberts 2005, p.62).

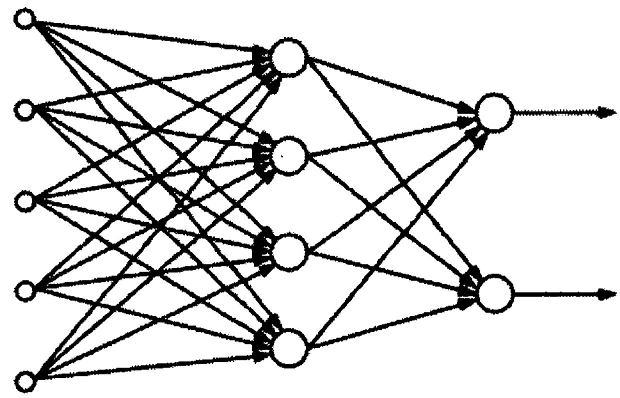



**Figure 2.6 - Architecture of a multilayer network (Liu 2001)**

This network has a feedforward architecture, which means the number of input neurons defines the dimensionality of the input space being mapped by the network and the number of output neurons defines the dimensionality of the output space into which the input is mapped (Liu 2001, p.5) .

For a feedforward neural network, the overall mapping is completed through intermediate mappings from one layer to another (Liu 2001, p.5). There are two factors which affect the intermediate mappings (Liu 2001, p.5). The first is connection mapping that transforms the output of the lower-layer neurons to an input to the neuron of interest, and the second is the activation function of the neuron (Liu 2001, p.5).

The multilayer perceptron networks are most useful because of the hidden layers that enable to form a complex non-linear function with a good degree of accuracy to acquire a global perspective (Cho 2007, p.31). Moreover, the multilayer perceptron networks are flexible, general-purpose, non-linear models, and can be used when the user has little knowledge about the form of the relationship between the independent and dependent variables. Therefore, the multilayer perceptron neural networks are valuable for designers to vary the complexity of the model easily ( Mastinu & Miano 2006,p.107) .

### 2.2.6 Hopfield Networks

In the 1982, Join Hopfield introduced the network architecture, Hopfield network is a feedback network that is called if the network is not feedforward network, In Hopfield networks, an associative memory can be implemented, and they can be used to solve optimization problem (MacKay 2003).

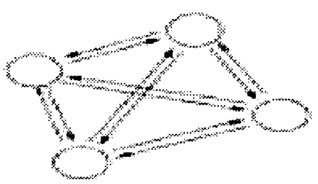


**Figure 2.7 - Feedback network (MacKay 2003)**

The basic structure of Hopfield Networks is, the neurons are represented by amplifiers which are like the old diode tubes of the Geiger counter, have two states-OFF and ON. An amplifier produces an output voltage, V, as a neuron that is firing at a rate, V/Vmax, relative to its peak rate where Vmax is the greatest output voltage that the amplifier can produce (Coughlin & Baran 1995). The strength of the connection between amplifier i and amplifier j is inversely proportional to the magnitude of the coupling resistance:

$Ti \infty 1/Rij$

For inhibitory coupling, each amplifier is paired with an invertor so that either the positive or the negative voltage is taken (Coughlin & Baran 1995).

Hopfield Networks are one of the most used neural network models for Medical image processing (Shi et al. 2009). In medical processing, there are two categories, which are image reconstruction and image restoration, the major advantage of using

Hopfield neural networks for medical image reconstruction is that medical image reconstruction can be taken as an optimization problem, which can be resolved by Hopfield Networks easily without pre-experimental knowledge (Shi et al. 2009).

### 2.2.7 Radial basis function (RBF) networks

The radial basis function (RBF) network is a very simple and powerful network structure (Arbib 2002). Broomhead and Lowe first were to develop the use of radial basis functions in the design of neural networks and to present how RBF model nonlinear relationships and implement generalization or interpolation between data points (Arbib 2002).

A radial basis function (RBF) neural network consists of three layers, the input layer, the hidden layer and the output layer. The activation functions that are the exponential functions are placed in the hidden layer. The input layer of the network is directly connected with the hidden layer of the network, therefore, only the connections between the hidden layer and the output layer are weighted. It is totally different from backpropagation networks (Palit & Popović 2005).

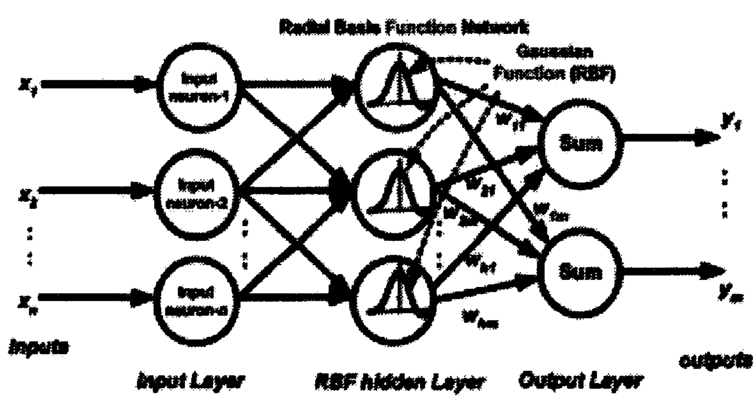


**Figure 2.8 - Diagram of RBF network (Palit & Popović 2005)**

There are some differences between RBF and multi-layer perceptron (MLP) networks, first of all, In RBF neural networks, there is a single hidden layer, which carries out nonlinear reflection

$$\Re^d \Rightarrow \Re^m \ , \ y = \sum_i h_i \phi(w_i, x),$$

This is being linear combination of basic functions. It is different from MLP where those functions rely on projection to a set of hyperplanes $\sigma(wx)$,the function in RBF neural network based on distances to basic centers(often Gaussian) are used: (Malyshevska 2009)

$y = \sum_i h_i \phi_i(|w_i - x|)$, Activation function: $\phi_i(z) = e^{-z^2/\sigma_i^2}$

For MLPs, there are one or more hidden layers. Moreover, nodes share a common neural model in MLPs. However, the hidden and output nodes of an RBF network are functionally distinct. Furthermore, MLP networks construct "global" functions approximations, whereas RBFs construct "local" function approximations using exponentially decaying localized non-linearites ( Fogarty 1996) .

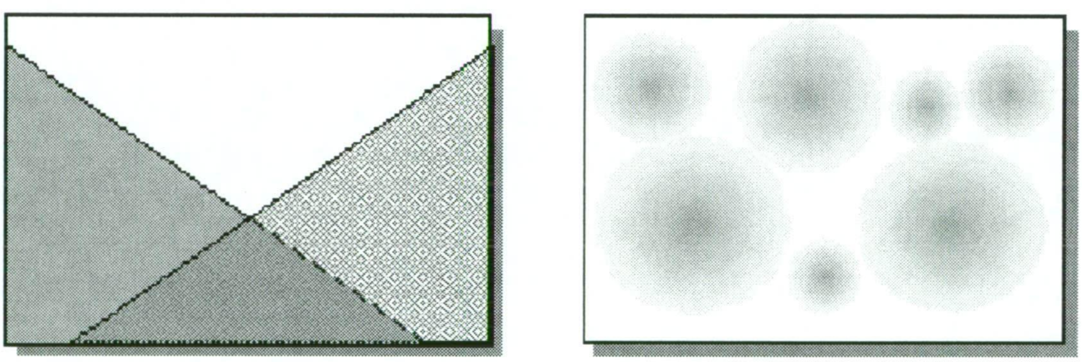


**Figure 2.9 - Global (perceptron) and local (RBF NN) methods of approximation (Fogarty 1996)**

To explain further, For Multi-layer Perceptrons, all neurons in hidden layer join in approximation in a field of any point, however, for RBF networks, only the nearest neurons join in approximation in a field of any point (Malyshevska 2009). In a RBF neural network, the number of the support function which is needed for approximating with set accuracy grows rapidly with dimension of space, which is the primary disadvantage of RBF neural networks (Malyshevska 2009).

The advantage of RBF networks is that unlike Multi-layer Perceptrons, they can avoid problems due to local minima (Malyshevska 2009). The reason is due to a hidden layer, the only parameters are changed in the learning process are the linear mapping from the hidden layer to the output layer (Malyshevska 2009). It is easy to find a single minimum by linearity. For regression problems this can be found in one matrix operation (Malyshevska 2009).

### 2.2.8 Back-propagation neural networks

In this section, back-propagation neural networks which are a kind of multilayer networks will be described particularly, because the neural network architecture will be used in the research.

The back propagation algorithm is the most widely used algorithms for calculating the neural network weights, and it was presented to a wide readership by Rumelhart and McClelland (Dayhoff 1990, p.58). It is has used in applications studies in a broad range of fields that include character recognition, sonar target recognition, image classification, signal encoding, knowledge processing and a variety of other pattern-analysis problems (Dayhoff 1990, p.58) . Moreover, It is one of easiest neural network to learning, the basic concept of back propagation is if the network gives the wrong answer, then the weights are corrected so the error is lessened and as a result future responses of the network and more likely to be correct (Dayhoff 1990, p.58) .

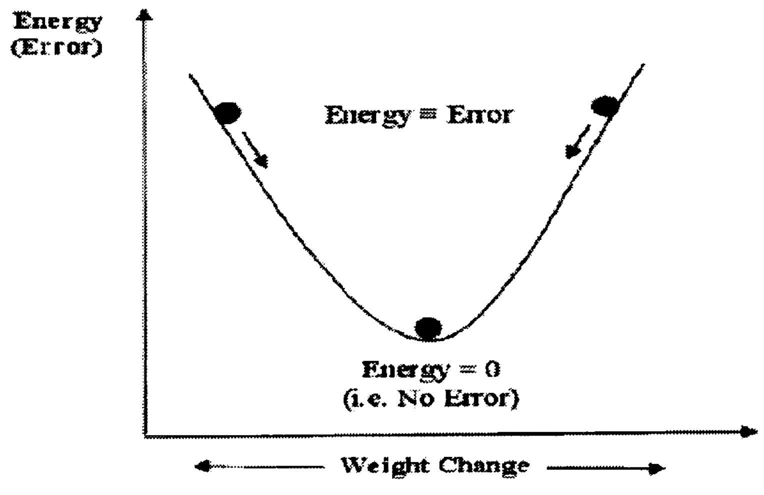



**Figure 2.10 - Energy analogy of gradient decent**

In the other word, the algorithm computes the weights for multilayer network, and then a set of unite are fixed and interconnections. Gradient decent is used to try to minimize the squared error between the network output values and the aim values for those outputs.

$$E(\vec{w}) = \frac{1}{2}\sum_{d \in D} \sum_{k \in outputs} (t_{kd} - o_{kd})^2$$

Where $E$ is all error values, *outputs* is the set of output units, $t_{kd}$ and $o_{kd}$ are the target value and output value of kth unit (Mitchell 1997).

Moreover, a unit is necessary for building multiplayer networks, which have two features, output is a nonlinear function of its inputs, and output is also a differentiable function of its inputs (Mitchell 1997). In standard back-propagation, there are feed forward networks with linear-sigmoid nodes in hidden layers and hidden weights and output weights that are learnt by gradient descent on error (Glyn & Humphreys 1999, p.123). The linear-sigmoid function is activated as a generalized or smoothed version of the linear-threshold function in neuron perceptron (Glyn & Humphreys 1999, p.124).

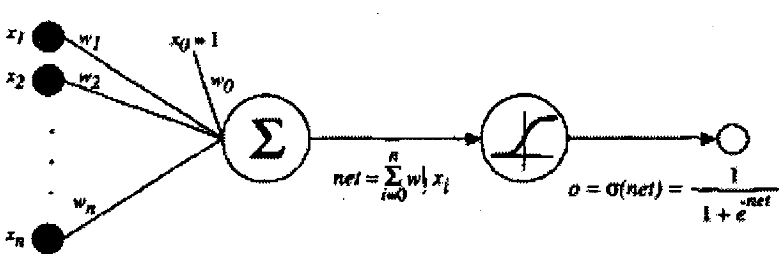
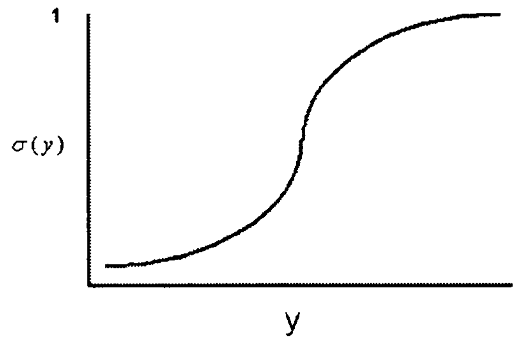


**Figure 2.11 - A sigmoid unit (Mitchell 1997)**

The sigmoid unit, firstly it calculates a linear combination of its inputs, and then provides a threshold to the result. The output from the sigmoid unit is

$$o = \sigma(\vec{w}\vec{x})$$

Where




$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

$\sigma$ is sigmoid function or the logistic function. The range of the output is between 0 and 1 (Mitchell 1997). The function can map a huge input domain to a small range of outputs, (Mitchell 1997) so it is also called the squashing function. Another benefit of using sigmoid function is that its derivative is easily described by its output:

$$\frac{d\sigma(y)}{dy} = \sigma(y) \cdot (1 - \sigma(y)),$$

This is used in Gradient descent rule (Mitchell 1997).

In the following figure, the basic learning process for back propagation will be given, where the important steps are presented in the feed-forward network with two layers of sigmoid units.

BACKPROPAGATION(*training_examples*, $\eta$, $n_{in}$, $n_{out}$, $n_{hidden}$)

*Each training example is a pair of the form* $\langle \vec{x}, \vec{t} \rangle$, *where* $\vec{x}$ *is the vector of network input values, and* $\vec{t}$ *is the vector of target network output values.*

$\eta$ *is the learning rate (e.g., .05).* $n_{in}$ *is the number of network inputs,* $n_{hidden}$ *the number of units in the hidden layer, and* $n_{out}$ *the number of output units.*

*The input from unit i into unit j is denoted* $x_{ji}$, *and the weight from unit i to unit j is denoted* $w_{ji}$.

- Create a feed-forward network with $n_{in}$ inputs, $n_{hidden}$ hidden units, and $n_{out}$ output units.
- Initialize all network weights to small random numbers (e.g., between −.05 and .05).
- Until the termination condition is met, Do
    - For each $\langle \vec{x}, \vec{t} \rangle$ in *training_examples*, Do

      *Propagate the input forward through the network:*

      1. Input the instance $\vec{x}$ to the network and compute the output $o_u$ of every unit $u$ in the network.

      *Propagate the errors backward through the network:*

      2. For each network output unit $k$, calculate its error term $\delta_k$

      $$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

      3. For each hidden unit $h$, calculate its error term $\delta_h$

      $$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh}\delta_k$$

      4. Update each network weight $w_{ji}$

      $$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

      where

      $$\Delta w_{ji} = \eta \delta_j x_{ji}$$

**Figure 2.12 - The Back propagation algorithm for feedforward network includes two layers of sigmoid units (Mitchell 1997)**

The problem of learning for back-propagation algorithm is to find a large hypothesis space which is defined by all weight values for all the units in the network (Mitchell 1997). However, in multilayer networks, the error surface can have many local minima, so gradient descent can be trapped in one of them (Mitchell 1997). Although the global minimum error could not be found by back-propagation, it provides ability to converge toward some local minimum in error surface (Mitchell 1997).

As result of the poor understanding of complex errors, there are only common heuristics to reduce the problem of local minima such as adding a momentum term to the weight-update, using stochastic gradient descent rather than true gradient descent and training multiple networks using the same data with different random weights in the initialization of each network (Mitchell 1997).

Moreover, another problem for back-propagation is overfitting problems. As training proceeds, the growing of some weights reduce the errors for training data, meanwhile the complexity of a learned decision surface is increasing. If the iterations of weight-update are too many, the overly complex decision surface could be appear, which can not be representative for training data (Mitchell 1997).

In the past, back-propagation neural network is one of most used type of artificial neural networks in the family of artificial neural networks (Ciampi & Zhang 2002). It has been proved to achieve a high accuracy level in detecting breast cancer ( Karlik & Unlu 2008) .

### 2.2.9 Trigonometric neural networks

In trigonometric neural networks, where the trigonometric function is as a nonlinear activation function in the feedforward neural networks, the function in particular sines and cosines, may be employed in the hidden neural units in order to the resulting networks satisfy the conditions of the stone-Weierstrass theorem (Gupta, Jin & Homma 2003, p260) .

The functions in the basic trigonometric system are:

1, cos(x),sin(x),(cos2x),sin(2x),cos(nx),sin(nx)

All these functions have the common period $2\pi$.

A two-layered trigonometric network with a single hidden layer is presented by the following input-output transfer function.

$$y = \sum_{i=1}^{N} u_i \phi_i \left( \sum_{j=1}^{n} w_{ij} x_j + \theta_i \right)$$

This is instead of the sigmoid function with a trigonometric function $\phi(x)$ in the conventional two-layered neural network. The trigonometric activation function $\phi i$ may be chosen as (Gupta, Jin & Homma 2003, p.260)

1) All $\phi_i(x)=\cos(x)$ (cosine network);

2) All $\phi_i(x)=\sin(x)$ (sine network);

3) $\phi_i(x)=\cos(x)$ or $\sin(x)$ (trigonometric network)

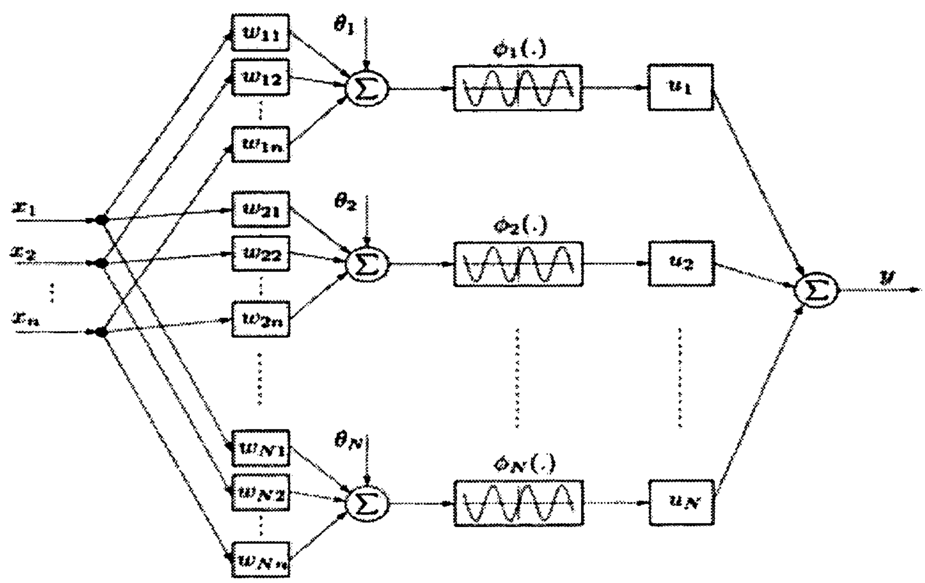


**Figure 2.13 - Diagram of the trigonometric network (Gupta, Jin & Homma 2003)**

Trigonometric networks are a typical class of feedforward neural networks with nonsigmoidal functions. Because of the classical trigonometric series expansion and the network expression of a continuous function, the advantage of the trigonometric network is more flexible and useful for many application and the coefficients of the trigonometric series have to be solved analytically using the function to be approximated while the weights of the network can be determined through a learning process (Gupta, Jin & Homma 2003).In a past research, the TNN with the sine function has been used in analysis of multi-frequency signal and the good performance has been shown (Hara & Nakayama 1994).

### 2.2.10 Exponential neural networks

Exponential neural networks, in which the Exponential function is as a nonlinear activation function in the feedforward neural networks, the simple exponential function is negative exponential function.

$$y = e^{-x}$$

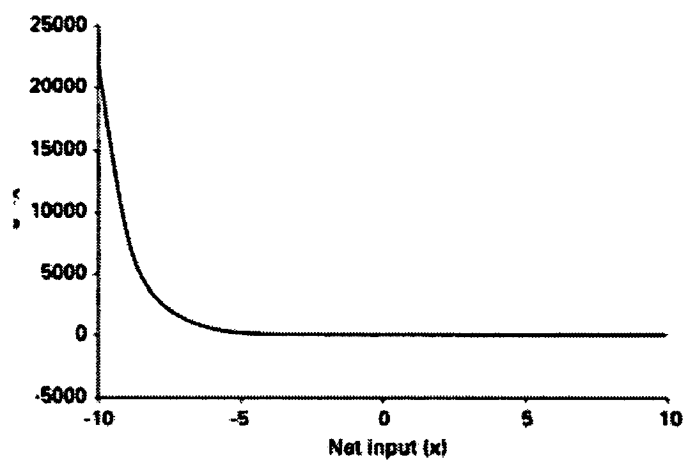



**Figure 2.14 - negative exponential Function (Shultz 2003)**

The negative exponential does just the reverse of the exponential, it begins very high with a negative x and then decreases with the increasing value of x. when x=0, the value of the negative exponential function, and as x increases, the value approaches a value of 0 (Shultz 2003).

In a past study of learning library of Chinese Character, the training speed using the BP neural network with negative exponential function was faster than the BP neural networks with sigmoid function by 30% (Yang & Han 1999).

## 2.3 Summary

In the field of neural networks, the presentation of new developments and new ideas from many researchers since the concept of neural network was presented. ANNs have been applied success in a variety of areas. However, researchers expect to utilize ANNs to solve more complex problems. Therefore, they are still exploring and improving the models of ANNs.

For a variety of models of ANNs, the activation function is one of the most important parts of an artificial neuron. Sigmoid activation function may be the most popular choice of the activation function (Laganà 2004). However, this function may not be appropriate in all situations. Therefore, in the paper, there are some analyses of the advantages and disadvantages of other activation functions such as Trigonometric and Exponential function.

In conclusion, the purpose of this thesis is to explore the structure of ANNs, and prove that trigonometric and Exponential function as activation are more valuable and better than MLPs in breast cancer diagnosis, Moreover , the learning speed and generalization capabilities of the ANNs will be discussed.

# 3 Methodology

In the previous chapters, the background and overview of some types of neural networks was described; especially different models of back propagation neural networks using different functions as activation functions. Moreover, earlier works about using neural network to diagnose breast cancer was introduced briefly. The aim was to present relevant methods and defines which methods will be used in the research.

In the introduction, the purpose of thesis was mentioned, which was to explore the performances of different functions as activation function in back propagation neural networks. Those types of back propagation neural networks are trigonometric networks, exponential networks. In order to achieve this, two steps should be done, firstly, the trigonometric function or exponential function as well as their derivatives in calculation of error value, was used instead of sigmoid function and its derivative in back propagation neural networks. Moreover, the performances of those neural networks were compared by testing them on the breast cancer dataset.

In this chapter, the preparation of the dataset, the selected NN program, neural network architecture and training process, will be descried. Due to the effects of different neural networks architecture on performance of neural network, the changing the architecture of back propagation neural networks was considered including the different number of hidden layers or numbers of neurons in a hidden layer.

## 3.1 Data set Preparation

In my experiments, the data set is a Breast Cancer Wisconsin dataset, which is available from UCI website, and the format file of *Weka* (ARFF) is available from the website: http://www.cs.waikato.ac.nz/~ml/weka/index_datasets.html, The brief description of the dataset follows.

Breast Cancer Wisconsin dataset includes 699 patterns with 9 integer-valued attributes. Each attributes range from 1 to 10, which are viz clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. These attributes present the external appearance and internal chromosome changes in nine different levels. Moreover, the output class involves two classes, benign (non-cancerous) and malignant (cancerous) (Witten & Frank 2005).

In the paper, the experiments of each neural network were based on the WBCD database. The classification results obtained from the Trigonometric neural network, Exponential neural network were used to compare with those obtained from traditional back propagation neural network with sigmoid activation function.

## 3.2 Neural Networks selection

There are three main types of back propagation neural networks were concerned in the experiment, which are a trigonometric network, a Exponential network and a traditional neural network with sigmoid activation function.

Trigonometric network

According to the literature review, one Trigonometric function was chosen to be an activation function in back propagation network, which is:

$$f(x) = \sin(x)$$

Exponential network

For a exponential network, the activation function which was tested is

$$f(x) = e^{-x}$$

Moreover, the activation function of the traditional neural networks is

$$f(x) = \frac{1}{1 + e^{-x}}$$

Those activation functions were used in back propagation neural network to classify the WBCD data.

## 3.3 Neural network program

The NN programs are modified based on *Weka*. *Weka* was developed at the University of waikato in New Zealand (Witten & Frank 2005). It is one popular and open source framework which involves many well-known implementations of machine leaning algorithms that you can easily apply to your dataset (Witten & Frank 2005).

In *Weka*, there is a program called MultilayerPerceptron which is a neural network that trains using back propagation with the common activation function (sigmoid function). In my research, I modified the program with the different activation functions which I mentioned before.

Moreover, *Weka* provides some options to configure the structure of the networks

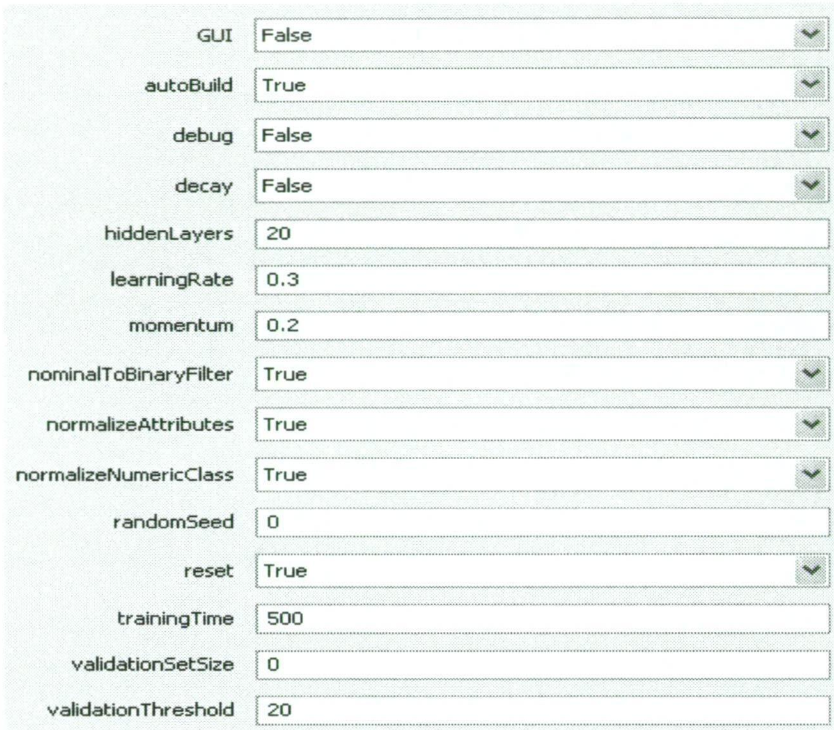



**Figure 3.1 - Default options in Weka**

Basically, the default options were chosen in my experiments, such as learning rate (0.3), momentum (0.2) and training Time (500). The numbers of neurons and hidden layers which are key areas in my experiments will be adjusted in testing. The options of nominalToBinaryfilter, normalizeAttributes and normailizeNumericClass are selected, which is to improve the performance of the network.

In the output of the program, some values are reported, which includes the correctness rate in testing, training time, root mean squared error, misclassification costs and the information about the structure of network.

## 3.4 Training method

10- Fold cross validation was selected to test the different neural networks in my experiments, which is a major method in measuring the success of machine learning algorithms (Kohavi 1995). The techniques could be used in classification, regression, clustering, and feature selection (Celis & Musicant 2002).

The general technique for testing how well a machine learning algorithm works is k-fold cross validation (Kohavi 1995). The basic concept of cross-validation is the matrix A, and the equivalent elements of y, are divided into k groups for each of k iterations, a distinct fold of the data is held out from the processes as a test set, and the left data is used as a training set. The machine learning algorithm is implemented on the training set, and validated via the test set. It generates k estimates of the generalizability of the algorithm, and then the overall test set accuracy is generated from averaged value (Kohavi 1995).

The K-fold cross validation method is one of the most successful methods for conquering the overfitting problem, especially, for dealing with small training sets (Mitchell 1997). Because k-fold cross-validation approach is performed k different times, a different part of the data is used for training data and for validation data. Then the results are averaged (Mitchell 1997).The approach not only avoids the overfitting problem but also improves the accuracy of finding the lowest error for the validation set (Mitchell 1997) .

## 3.5 Neural network Architecture

In multiplayer neural network, there are three parts including input layers, hidden layers and output layers. The number of neurons in hidden layers, and the number of hidden layers are important factors for the performance of neural networks. However, in past research and testing, there were no specific rules about the number of neurons and hidden layers. Basically, the appropriate number of hidden neurons and hidden layers are determined by constant testing.

In the current experiments, the strategy focuses on testing the three types of BP neural networks with different hidden nodes or hidden layers. The performances of the BP neural networks in those situations were compared including the correctness rate in testing, root mean squared errors, training speed, misclassification costs.

## 3.6 Training process

In the training process, firstly, the strategy for selecting the number of hidden nodes and hidden layers was designed .Due to the effect of the change of number of neurons and hidden layers on the performance of neural networks, the number of hidden layers were changed from 1 to 2, and different number of neurons set in those layers. Generally, for the number of neurons in hidden layers, it is less than N/d, where N is the total number of instances (training pairs), and d is the number of attributes (i.e. the number of neurons in the input layer) (Barron 1994). Therefore, with this strategy and the feature of the WBCD data which contains 699 instances and 9 attributes, consequently, the maximum number of the neurons in a hidden layer is set at 60. The rule of the change of numbers of neurons was made as following:

1. When the hidden layer is 1, the number of neurons was adjusted as follows:

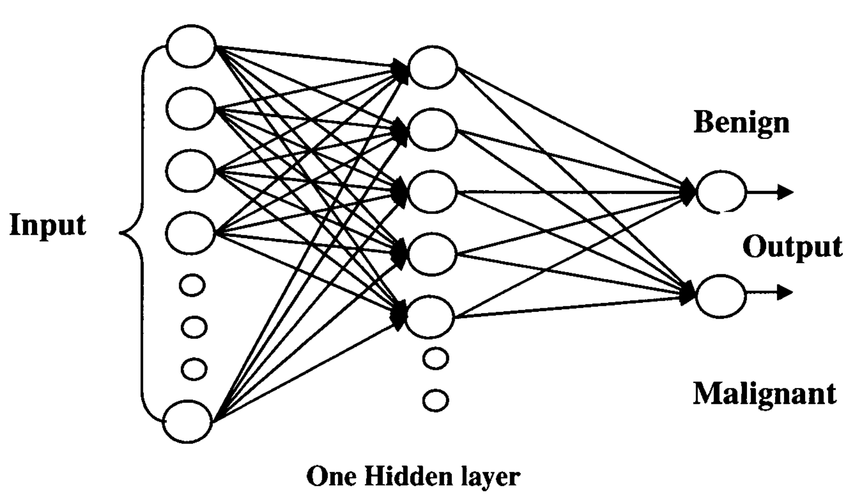



**Figure 3.2 - Neural network structure (One Hidden layer)**

With the neural network structure (figure 3.2) which contains one hidden layer, the number of neurons was set to 5, 10,15,20,30 and 60.

When the number of hidden layers is 2, the number of neurons is adjusted as follows:

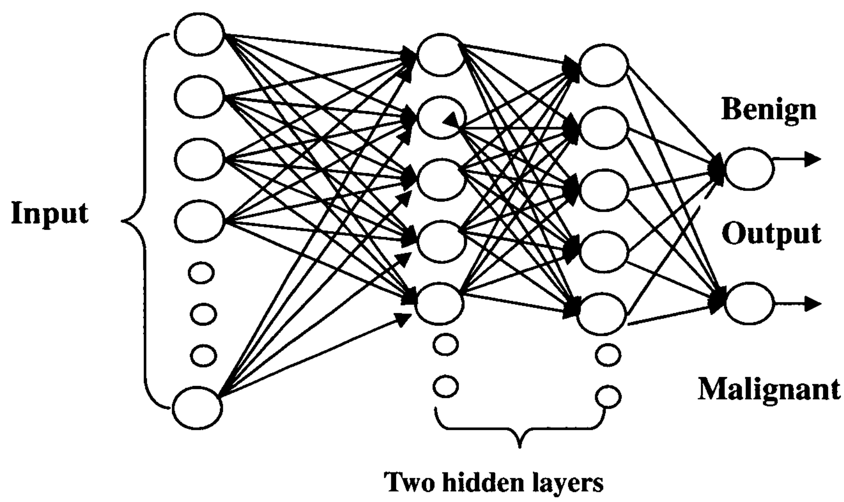



**Figure 3.3 - Neural network structure (Two Hidden layers)**

With the neural network structure (figure 3.3) which includes two hidden layers, when the number of neurons in the first layer was 5, tests were conducted with 5,10,15,20,30 and then 60 neurons in the second layer. A similar adjustment was made to the second layer for first layers with 10,15,20,30 and then 60 neurons.

There are two reasons that cause testing to finish. First of all, the number of neurons reached the maximum value that was set up at the beginning. Secondly, the operation of increasing the number of neurons in hidden layers could be stopped when the correctness rates was getting worse because of the overfitting problems.

When the training was finished, the results were saved to a file, which was including the correctness rates, the root mean squared errors (RMSE), detailed accuracy and Confusion Matrix. They are important information relevant to the performance of neural networks.

# 4 Results and Discussion

Basically, the aim of the project is to explore different models of back propagation neural networks which use different functions as activation functions and to try to show that their performance was better than the model of back propagation neural networks with sigmoid function in analyzing breast cancer data. In the previous chapter, the methodology was described including the two modified models of back propagation neural networks: trigonometric networks as well as exponential networks, the training software platform, and the strategy of training and some problems that appeared in training process.

Those models of back propagation neural networks will be examined through experiments which were followed the mentioned strategy. The collected results from the will be analyzed. The experiment is expected to provide some useful information or evidence to support the project achieving the goal.

In this chapter, there are four parts. Three models will be analyzed in each part separately, and in the final part, those models will be compared and discussed. The cover contains the correctness rate, the root mean error, training speed and so on.

All the models will be tested in *Weka*, which implemented on a laptop with Intel core 2 CPU T7200 2.00GHZ and 1.5GB of RAM

In the test, there are four kinds of result values should be focus on, first of all , the correctly classified rate, which reflects directly whether the performance is good or not, moreover, the root mean squared error (RMSE) is important reliance for the judgment to the network topology ,basically, if the RMSE is closer to zero, the simulated image from ANNs is more similar to real image. In addition, misclassification costs show the correctness rates for each class. Finally, training speed depends on the complexity of networks, and it can not be easily judged whether long or short training speed is better. For example, in an experiment, there are two networks, one containing 30 hidden units and another one 3 hidden units. Although the accuracy rate of the former higher than the rate of the latter by two percent, the network with 30 hidden units network spent approximate 1 hour on a workstation, compared to approximately 5 minutes for a network with 3 hidden networks (Mitchell 1997). Therefore, balancing accuracy with training speed should be considered in building networks. In the following part, the changes in those results will be described particularly.

**BP neural network with sigmoid activation function**

| number of hidden neurons | Correctly Classified rate | RMSE | Training speed(seconds) |
|---|---|---|---|
| 5 | 95.28% | 0.197 | 23.75 |
| 10 | 95.57% | 0.1911 | 26.03 |
| 15 | 95.99% | 0.1859 | 33.33 |
| 20 | 95.85% | 0.185 | 39.67 |
| 30 | 95.71% | 0.1916 | 42.02 |
| 45 | 95.57% | 0.1875 | 75.2 |
| 60 | 95.85% | 0.1864 | 90.27 |

**Table 4.1 BPN with sigmoid function projection results**

In building the model, the number of hidden layers is one, and the initial number of hidden neurons was chosen for test including 5, 10, 15,20,30,45 and 60. The training rate is 0.3, momentum is 0.2, and training times is 500.

The prediction results of the BP networks with different hidden neurons are presented in Table 4.1. From Table 4.1, with the increasing number of hidden neurons, the correctness rates and the RMSE of the BPN models are close. Only the training speed is increased and no significant changes in values. Moreover, the topology {9-20-2}, which had 95.85% correctness rate and 0.185 RMSE, gave the best result. The Confusion Matrix of the result is presented in Table 4.2.

| === Confusion Matrix === | | |
|---|---|---|
| a | b | <-- classified as |
| 440 | 18 | a = benign |
| 11 | 230 | b = malignant |

**Table 4.2 Confusion Matrix of the BPN with 20 hidden neurons**

In Table4.2, 18 benign patients were misclassified as malignant, and 11 malignant patients were misclassified as benign.

The diagnostic results for breast cancer data using BPN with sigmoid model are summarized in Table 4.3. From the results in Table 4.3, the average correctness rate is 95.69% and the average RMSE 0.1892. Thus, this model achieves high average accuracy level in analyzing breast cancer data.

| Average correctness rate | Average RMSE | Average training speed |
|---|---|---|
| 95.69% | 0.1892 | 47. 1814 |

**Table 4.3 Average correctness rate and RMSE**

As following the planed training strategy, the number of hidden layer will be changed from 1 to 2, and the models are rebuilt with different hidden neurons and all other parameters are unchanged.

| Number of hidden neurons in 1st hidden layer | Number of hidden neurons in 2nd hidden layer | Correctly Classified rate | RMSE | Training speed(seconds) |
|---|---|---|---|---|
| 5 | 5 | 95.42% | 0.2016 | 21.98 |
| 5 | 10 | 95.14% | 0.2051 | 16.3 |
| 5 | 15 | 95.71% | 0.2013 | 20.64 |
| 5 | 20 | 95.99% | 0.1926 | 25.74 |
| 5 | 30 | 95.99% | 0.188 | 34.52 |
| 5 | 45 | 95.42% | 0.2049 | 49.19 |
| 5 | 60 | 96.14% | 0.1866 | 64.91 |

**Table 4.4 BPN with sigmoid function projection results (2 hidden layers) and the number of hidden neurons is 5 in the first layer**

The prediction results of the BP networks with different numbers of hidden neurons in different hidden layers are presented in Table 4.4. (These are only part of the results, and other numbers of hidden neurons were also tested with no remarkable results.) As shown in Table 4.4, the BP network contains 2 hidden layers where the number of hidden neurons is 5 in the first layer, and the number of hidden neurons in the second layer is from 5 to 60. There was a slight change of correctness rate and RMSE with the number of hidden neurons and layers. The best result appears in the topology {9-5-60-2}, which achieved 96.14% correctness and 0.1866 RMSE. The Confusion Matrix of the result is presented in Table 4.5.

| === Confusion Matrix === | | |
|---|---|---|
| a | b | <-- classified as |
| 441 | 17 | a = benign |
| 10 | 231 | b = malignant |

**Table 4.5 Confusion Matrix of the BPN with 20 hidden neurons**

As shown in Table 4.5, 17 benign patients were misclassified as malignant, and 10 malignant patients were misclassified as benign.

| Number of hidden neurons in 1st hidden layer | Number of hidden neurons in 2nd hidden layer | Correctly Classified rate | RMSE | Training speed(seconds) |
|---|---|---|---|---|
| 60 | 5 | 95.57% | 0.1879 | 107.97 |
| 60 | 10 | 95.14% | 0.1995 | 145.41 |
| 60 | 15 | 95.42% | 0.1996 | 188 |
| 60 | 20 | 95.71% | 0.1954 | 210.17 |
| 60 | 30 | 94.71% | 0.2029 | 283.39 |
| 60 | 45 | 95.85% | 0.1881 | 385.84 |
| 60 | 60 | 95.57% | 0.1907 | 500.66 |

**Table 4.6 BPN with sigmoid function projection results (2 hidden layers) and the number of hidden neurons is 5 in the first layer**

In Table 4.6 shows the results from the BPN with 60 hidden neurons in the first layer. There was a good result with the topology {9-60-45-2} where the correctness rate is 95.85% and the RMSE is 0.1881. However, the training speed is lower than other topologies which have less complexity.

The diagnostic results for breast cancer data using BPN with sigmoid model with two hidden layers are summarized in Table 4.7. From the results in Table 4.7, the average correctness rate is 95.69% and the average RMSE 0.1892. Thus, this model achieves high average accuracy level in analyzing breast cancer data.

| Average correctness rate | Average RMSE | Average training speed |
|---|---|---|
| 95.47 | 0.1993 | 122. 644 |

**Table 4.7 Average correctness rate and RMSE**

In summary, the BPN with sigmoid model has good performance in analyzing breast cancer data. Comparing the results of networks with one hidden layer to the results of networks with two hidden layers, the networks with one hidden layer has better performance and faster training speed.

**Trigonometric neural network (TNN)**

**(BP neural network with sine function)**

| number of hidden neurons | Correctly Classified rate | RMSE | Training speed(seconds) |
|---|---|---|---|
| 5 | 90.41% | 0.3096 | 9.27 |
| 10 | 79.97% | 0.4475 | 15.73 |
| 15 | 84.41% | 0.3934 | 22.88 |
| 20 | 81.26% | 0.4275 | 30.47 |
| 30 | 85.69% | 0.3597 | 45.08 |
| 45 | 76.97% | 0.4356 | 67.63 |
| 60 | 84.12% | 0.3955 | 92.86 |

**Table 4.8 Trigonometric neural network projection results**

In building the model, the number of hidden layer is one, and the initial number of hidden neurons was chosen for test including 5, 10, 15,20,30,45 and 60. The training rate is 0.3, momentum is 0.2, and training times is 500. (All testing parameters are same as the BP network with sigmoid in order to achieve comparable results)

The prediction results of TNN with different hidden neurons are presented in Table 4.8. Table 4.8 shows that, with the increasing number of hidden neurons, the correctness rates of the TNN models vary a lot. Basically, the tendency of the correctness rate reduces with increased numbers of hidden neurons. The best result appears at the beginning, when the topology is {9-5-2} with 90.41% correctness rate and 0.3096 RMSE. The lowest one {9-45-2} just achieved 76.97% correctness and 0.4356 RMSE. Furthermore, the situation that some instances were unclassified happened. With the increasing number of hidden neurons, the reason could be overfitting problems from increasing complexity of neural networks. The Confusion Matrix of the result is presented in Table 4.9.

| === Confusion Matrix === | | |
|---|---|---|
| a | b | <-- classified as |
| 435 | 23 | a = benign |
| 44 | 197 | b = malignant |

**Table 4.9 Confusion Matrix of the TNN with 5 hidden neurons**

In Table4.9, 23 benign patients were misclassified as malignant, and 44 malignant patients were misclassified as benign.

The diagnostic results for breast cancer data using TNN are summarized in Table 4.10. From the results in Table 4.10, the average correctness rate is 83.26% and the average RMSE 0.3955. According to the correctness rate and confusion matrix, the performance of ENN with one hidden layer is satisfactory

| Average correctness rate | Average RMSE | Average training speed(secs) |
|---|---|---|
| 83.26% | 0.3955 | 40. 56 |

**Table 4.10 Average correctness rate and RMSE**

In the next section, the number of hidden layers will be changed from 1 to 2, and the models are rebuilt with different hidden neurons and all other parameters are unchanged.

| Number of hidden neurons in 1st hidden layer | Number of hidden neurons in 2nd hidden layer | Correctly Classified rate | RMSE | Training speed(seconds) |
|---|---|---|---|---|
| 5 | 5 | 90.70% | 0.3049 | 12.95 |
| 10 | 5 | 92.13% | 0.2805 | 23.17 |
| 15 | 5 | 87.41% | 0.3548 | 30.25 |
| 20 | 5 | 85.12% | 0.3857 | 41.44 |
| 30 | 5 | 89.13% | 0.3297 | 58.72 |
| 45 | 5 | 86.41% | 0.3687 | 84.16 |
| 60 | 5 | 91.13% | 0.2978 | 113.67 |

**Table 4.11 TNN projection results (2 hidden layers) and the number of hidden neurons is 5 in the second layer**

The prediction results of TNN with different numbers of hidden neurons in different hidden layers are presented in Table 4.11. (These are only part of the results, and other numbers of hidden neurons were also tested with no remarkable results.) As shown in Table 4.11, TNN topologies containing 2 hidden layers, where the number of hidden neurons is 5 in the second layer, and the number of hidden neurons is from 5 to 60 in first layer, were focused on. Those network topologies with few neurons in second layer all have good correctness rates, which can represent overall performance in the experiment. The maximum correctness rate of TNN is 92.13% when the topology is {9-10-5-2}, and its RMSE is 0.2805. The Confusion Matrix of the result is presented in Table 4.12.

| === Confusion Matrix === | | |
|---|---|---|
| a | b | <-- classified as |
| 452 | 6 | a = benign |
| 49 | 192 | b = malignant |

**Table 4.12 Confusion Matrix of the TNN with 10 hidden neurons**

In Table 4.12, 93 benign patients were misclassified as malignant, and 166 malignant patients were misclassified as benign.

The diagnostic results for breast cancer data using TNN with two hidden layers are summarized in Table 4.13. From the results in Table 4.13, the average correctness rate is 58.48% and the average RMSE 0.5544.

| Average correctness rate | Average RMSE | Average training speed |
|---|---|---|
| 88.86% | 0.3317 | 115. 5269 |

**Table 4.13 Average correctness rate and RMSE**

In summary, the performance of TNN in analyzing breast cancer data is good, its average correctness rate is around 90% and high accuracy level often appeared in the neural works with small numbers of neurons in hidden layers. With increasing numbers of neurons in hidden layers, the performance of TNNS networks reduced quickly and some instances were unclassified, which is considered as result of overfitting problems.

**Exponential neural network**

**(BP neural network with negative Exponential function)**

| number of hidden neurons | Correctly Classified rate | RMSE | Training speed(seconds) |
|---|---|---|---|
| 5 | 67.24% | 0.4695 | 7.97 |
| 10 | 61.52% | 0.4739 | 14.77 |
| 15 | 61.95% | 0.4892 | 22.17 |
| 20 | 61.95% | 0.5028 | 28.67 |
| 30 | 65.52% | 0.4811 | 44.56 |
| 45 | 67.24% | 0.4781 | 63.02 |
| 60 | 62.80% | 0.4868 | 82.72 |

**Table 4.14 Exponential neural network projection results**

In building the model, the number of hidden layers is one, and the initial number of hidden neurons was chosen for test including 5, 10, 15,20,30,45 and 60. The training rate is 0.3, momentum is 0.2, and training times is 500. (All testing parameters are same as the BP network with sigmoid in order to achieve comparable results)

The prediction results of ENN with different numbers of hidden neurons are presented in Table 4.14. From Table 4.14, with increasing number of hidden neurons, the correctness rates and the RMSE of the ENN models are steady, only the training speed are increased and no remarkable change appears. Moreover, the best result from the topology {9-5-2} is 67.24% correctness rate and 0.4685 RMSE. The Confusion Matrix of the result is presented in Table 4.15.

| === Confusion Matrix === | | |
|---|---|---|
| a | b | <-- classified as |
| 458 | 0 | a = benign |
| 229 | 12 | b = malignant |

**Table 4.15 Confusion Matrix of the ENN with 5 hidden neurons**

In Table 4.15, all benign patients were classified correctly, but 229 malignant patients were misclassified as benign.

The diagnostic results for breast cancer data using ENN are summarized in Table 4.16. From the results in Table 4.16, the average correctness rate is 64.03% and the average RMSE 0.4831. Although ENN correctly classified all benign patients, the high incorrect classification of malignant patients affects the performance of ENN.

| Average correctness rate | Average RMSE | Average training speed |
|---|---|---|
| 64.03% | 0.4831 | 37. 6971 |

**Table 4.16 Average correctness rate and RMSE**

In the next section, the number of hidden layers will be changed from 1 to 2, and the models are rebuilt with different hidden neurons and all other parameters are unchanged.

| Number of hidden neurons in 1st hidden layer | Number of hidden neurons in 2nd hidden layer | Correctly Classified rate | RMSE | Training speed(seconds) |
|---|---|---|---|---|
| 30 | 5 | 62.52% | 0.587 | 54.09 |
| 30 | 10 | 65.24% | 0.5208 | 72.67 |
| 30 | 15 | 57.08% | 0.5112 | 89.45 |
| 30 | 20 | 56.22% | 0.5231 | 4.89 |
| 30 | 30 | 60.94% | 0.5194 | 142.58 |
| 30 | 45 | 56.51% | 0.4919 | 200.05 |
| 30 | 60 | 66.81% | 0.4828 | 251.5 |

**Table 4.17 ENN projection results (2 hidden layers) and the number of hidden neurons is 30 in the first layer**

The prediction results of ENN with different numbers of hidden neurons in different hidden layers are presented in Table 4.17 (These are only part of the results, and other numbers of hidden neurons were also tested with no remarkable results.) From Table 4.17, the BP network contains 2 hidden layers where the number of hidden neurons is 30 in the first layer, and from 5 to 60 in the second layer. As shown in the table, the maximum correctness rate of ENN is 66.81% which was achieved with the topology {9-30-60-2} and its RMSE is 0.4828. The change in correctness rate of ENN with two hidden layers is from around 40% to 66.81%. The Confusion Matrix of the result is presented in Table 4.18.

| === Confusion Matrix === | | |
|---|---|---|
| a | b | <-- classified as |
| 457 | 1 | a = benign |
| 231 | 10 | b = malignant |

**Table 4.18 Confusion Matrix of the ENN with 10 hidden neurons**

In Table 4.18, 1 benign patient was misclassified as malignant, and 231 malignant patients were misclassified as benign.

The diagnostic results for breast cancer data using ENN with two hidden layers are summarized in Table 4.19. From the results in Table 4.19, the average correctness rate is 58.48% and the average RMSE 0.5544.

| Average correctness rate | Average RMSE | Average training speed |
|---|---|---|
| 58.49% | 0.5547 | 115. 427 |

**Table 4.19 Average correctness rate and RMSE**

In summary, the performance of ENN in analyzing breast cancer data is unsatisfactory in the experiment. Its average correctness rate is only around 60%. The major problem is the misclassification of malignant patients.

**The comparison of three BP models**

In order to evaluate the overall performance of these models, four aspects of the results will be considered including the correctness rate, the root mean squared errors, training speed and the misclassification costs.

**Correctness rates and RMSE**

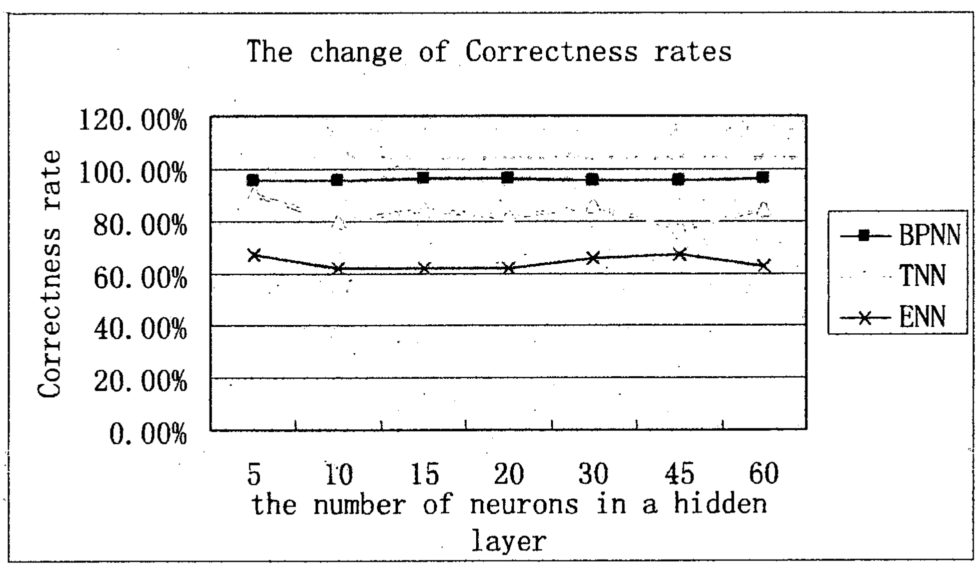



**Figure 4.1 - The change of Correctness rates with the increasing number of neurons in a hidden layer**

For the correctness rates and RMSE, the example of the models with a hidden layer was taken because the situation of the models with two hidden layers is similar to the models with a hidden layer. From Figure 4.1, the BPNN achieved the highest accuracy level, and the values are still stable when the hidden neurons and layers are adjusted. For TNNs, the best performance is slightly less then the performance of BPNN, however the values varied significantly with the increasing number of neurons and layers. For ENN, the performance is the lowest of these models in the experiment. Those best results are summarized in Table 4.20.

| Types of NNs | correctness rate | RMSE |
|---|---|---|
| BPNN | 95.85% | 0.1864 |
| TNN | 92.13% | 0.2805 |
| ENN | 67.24% | 0.4695 |

**Table 4.20 the Best results of those NN models**

**Training speed**

| Types of NNs | Average speed(seconds) |
|---|---|
| BPNN | 47. 1814 |
| TNN | 40. 56 |
| ENN | 37. 6971 |

**Table 4.20 the average training speed of those models with a hidden layer**

The average training speed of those models with a hidden layer is summarized in Table 4.20. The BPNN had the lowest training speed among three models, and the TNN was faster then the BPNN, the ENN was fastest. Finally, the classification of the best results of the three models is described in Table 4.21.

**Misclassification Cost**

| Types of NNs | Correctly Benign | Correctly malignant |
|---|---|---|
| BPNN | 96. 3% | 95. 9% |
| TNN | 98. 7% | 78. 7% |
| ENN | 99. 8% | 4% |

**Table 4.21 the classification of the best results of the three models**

In classifying benign patients, all models have good performance; however there is only good performance from BPNN in classifying the malignant patients, and in this experiment, the ENN basically can not classify that.

All in all, the BPNN still presents a good and stable performance. For TNN, although the performance is slightly worse than the performance of BPNN, the fast training speed should be noticed. For ENN, the performance is not good; however, the training speed is better.

# 5 Conclusion and Further work

There is no doubt that breast cancer is a dangerous disease for women throughout the world, therefore the effective solution is desired. Nowadays, the early detection is common and effective method to cure and save patients, which is generally using mammography. However, there exists the lack of analysis of the mammogram images, the low accuracy of classifying images of benign or malignant often affects the practitioner make more reasonable choices and manage patients. Hence, the problems of breast cancer diagnosis are in the scope of the more general and widely discussed data classification problems.

The artificial neural networks have been used for solving this kind of problems in a variety of areas and the performance is satisfying. Nevertheless, different neural network models have different effects on the same dataset. Therefore, it is necessary to try different models of neural networks and find a better one for improving the accuracy of analyzing the expected dataset, which is the purpose of thesis.

In the research, three different types of neural network models for classifying Breast Cancer Wisconsin dataset were used: MLP, trigonometric network (TNN) and exponential network (ENN) and the testing platform is *Weka*. All of them are based on the back propagation algorithm. The difference is activation function, which is one of the important factors that influence the performance of neural networks. The strategic experiments were carried out, and the three neural networks were tested by adjusting the number of hidden layers and the numbers of neural networks. The results showed that, the MLPs have good and stable performance, although the performance of the TNNs is slightly worse performance than the MLPs, the training speed is faster than the MLPs, which can save lots of implementation time on the computer and make decision on time. For ENN, the performance are not good in the experiment, however, the training speed is better.

## 5.1 Further work

In order to extend and explore further the performance of the two NN models: trigonometric networks and exponential networks. For analysis of the breast cancer dataset, further testing methods should be tried.

Firstly, the different training rates could be considered in the further experiments. Training rate is also one important factor for the performance of neural networks. According to Rumelhart et al. (1986), if learning rates are lower, the network tends to provide better network results. Therefore, experiments with lower training rates can be undertaken, and the changes in results will be observed and analyzed

Moreover, for the TNN, a smaller range of numbers of neurons in each hidden layer can be focused on. From the results in the previous experiments, the better results are often obtained from a small number of neurons in each hidden layer. Therefore, a small range (e.g. from 1 to 10) can be considered in further experiments.

In addition, for the ENN, although the training speed was faster than the other models, the correctness rates and RMSE were worse. Therefore, instead of the negative exponential function, another activation function could be chosen .The best candidate is Gaussian function, the formula for which is

$$y = e^{-x^2}$$

This is often used in the RBF networks which were mentioned in the literature review, in a past study, this function was selected as the activation function in BP neural networks in analysis of multi- frequency signals, and the performance of this neural network was better than BPNN with sigmoid function (Hara & Nakayama 1994).

# 6 References


1. American Cancer Society: Cancer Facts And Figures 2003: Technical Report, American Cancer Society, Atlanta, GA (2003).
2. Arbib, MA 2002, *The handbook of brain theory and neural networks,* MIT Press, US.
3. Adam, A & Omar, K 2008, 'Computerized Breast Cancer Diagnosis with Genetic Algorithms and Neural Network', *University Kebangsaan Malaysia.*
4. Barron, A. R. 1994, 'Approximation and Estimation Bounds for Artificial Neural Networks', *Machine Learning*, vol.14, no.1, pp.115-133
5. Ciampi, A & Zhang, F 2002, 'A new approach to training back-propagation artificial neural networks: empirical evaluation on ten data sets from clinical studies', *Statistics in medicine*, vol. 21, no. 9.
6. Cho, SY 2007, *Neural Networks and Computing: Learning Algorithms and Applications* Imperial College Press, US.
7. Coughlin, JP & Baran, RH 1995, N*eural Computation in Hopfield Networks and Boltzmann Machines,* University of Delaware Press, US.
8. Dayhoff, JE 1990, *Neural Network Architectures An Introduction*, Van Nostrand Reinhold, United States.
9. Fogarty, TC 1996, *Evolutionary computing,* Springer, Berlin.
10. Goel, AK, Saxena, SC & Bhanot ,S 2006, 'Modified Functional Link Artificial Neural Network', *International Journal of Computer Science*,vol.1,no.1,p.22.
11. Glyn, RE & Humphreys, W 1999, *Connectionist psychology,* Psychology Press, England.
12. Gupta, MM, Jin,L & Homma, N 2003, *Static and dynamic neural networks*, Wiley-IEEE,US.
13. Hawkes, PW 2003, *Advances in Imaging and Electron Physics,* Academic Press,

UK.

14. Husmeier, D, Dybowski, R & Roberts, S 2005, *Probabilistic modeling in bioinformatics and medical informatics*, Springer, Berlin.
15. Hara, K & Nakayama, K 1994, 'Comparison of activation functions in multilayer neural network for pattern classification', *Kanazawa University*.
16. Karlik, B & Unlu, S 2008, 'Practical artificial neural networks software for breast cancer diagnosis', *Department of Computer Engineering.*
17. Liao, XF, Wang, J & Yi, Z 2005, *Advances in Neural Networks,* Springer, Berlin.
18. Jagadish, K.S. & Iyengar, R.N 2005, *Recent Advances in Structural Engineering*, Orient Blackswan, US.
19. Liu, GP 2001, *Nonlinear identification and control,* Springer, Berlin.
20. Laganà, A 2004, *Computational science and its applications*, Springer, Berlin.
21. Ioanna, C., Evalgelos, D., George, K.2000. Fast Detection of Masses in Computer aided Mammography, IEEE Signal Processing Magazine: January 2000: 54-64.
22. MacKay, JCD 2003, *Information Theory, Inference, and Learning Algorithms,* Cambridge University Press, England.
23. Malyshevska, K 2009 'the usage of neural networks for the medical diagnosis' *XVth International Conference "Knowledge-Dialogue-Solution"* ,Varna, Bulgaria.
24. Mastinu, G & Miano, C 2006, *Optimal Design of Complex Mechanical Systems: With Applications to Vehicle Engineering,* Springer, Berlin.
25. Mange, D & Tomassini, M 1998, *Bio-Inspired Computing Machines: Toward Novel Computational Architectures.* PPUR Press, pp. 289-316.
26. Matignon, R 2005, *Neural Network Modeling Using SAS Enterprise Miner,* AuthorHouse, Bloomington.
27. Müller, B, Reinhardt, J & Strickland, MT 1995, *Neural networks: an introduction*, Springer Verlag.
28. Maurer, HA 1991, *New results and new trends in computer science,* Springer, Berlin.
29. Palit, AK & Popović, D 2005, *Computational intelligence in time series forecasting*, Birkhäuser.

30. Regan, GO 2008, *A brief history of computing,* Springer, Berlin.
31. Radiol, IJ 2006, 'Neural Network Analysis of Breast Cancer from Mammographic Evaluation' *Tarbiat Modarres University.*
32. R.Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In C.S. Mellish.editor, *Proceedings of the 14the International Joint Conference on Artificial Intelligence.* Morgan Kaufmann, 1995.
33. Rumelhart, D. E., Hinton, D. E., &Williams, R. J. (1986) (Vol. 1). Learning internal representations by error propagation in parallel distributed processing, Cambridge, MA: MIT Press, pp. 318–362.
34. Mitchell, T 1997, *Machine Learning*, McGraw-Hill Science, New York.
35. Shultz, TR 2003, *Computational developmental psychology*, The MIT Press.
36. Stein, DJ & Ludik, J 1998, *neural networks and psychopathology: connectionist models in practice and research*, Cambridge University Press.
37. Shi, Zh, He, LF, Nakamura1,T, Suzuki, K & Itoh, H 2009, ' Survey on Neural Networks Used for Medical Image Processing', *International Journal of Computational Science,* Vol. 3, No.1, pp.86-100.
38. Suzuki, Y, Ovaska, SJ, Furuhashi, T, Roy, R & Dote, Y 2000, *Soft Computing in Industrial Applications*, Springer, Berlin.
39. Tipton, HF, Krause, M & Corporation, E 2008, *Information security management handbook,* CRC Press, UK.
40. Vouros, GA & Panayiotopoulos, T 2004, *Methods and applications of artificial intelligence,* Springer, Berlin.
41. Ukil, A 2007, *Intelligent Systems and Signal Processing in Power Engineering*, Springer, Berlin.
42. Whitten, IH & Frank, E 2005, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann., Los Altos, CA.
43. Yang, W & Han, CC 1999, ' Improving the Fnction of BP Neural Network's Node and Apply the Improved Algorithm to Learn Library of Chinese Character', *Journal of Changchun University of Science and Technology,* Vol.22, No.3, pp 28-33.

# 7 Appendices

---

## Electronic Submission

**All test results**

Original results – these are all outputs from *Weka* in the experiments, which include detail information for each tested neural network topologies.

Excel results – the Excels contains results which are used in evaluation.

**Programs**

Modified_weka.jar – this is an executable program where the three tested models called MultilayerPerceptron, TNN and ENN in 'functions' option.

modifed _main_parts – this structure of folder is same as in *Weka*, and the folder contains main modified programs as follows:

/ modifed _main_parts/weka/classifiers/functions

TNN .java – this is main program for TNN.

ENN .java –this is main program for ENN.

/ modifed _main_parts/weka/classifiers/functions/neural

Derivative.java – this is to gain the input values for Derivative calculation.

TrigonometricUnit.java – this contains functions about calculation of output using sine activation function and error value.

ExponentialUnit.java - this contains functions about calculation of output using negative exponential activation function and error value.

/ modifed _main_parts/weka/classifiers/functions/neural/common

Utils.java – this is to normalize output values.

modeifed_weka_src.rar – all resource codes

**Dataset**

breast-w.arff – this is the format of *Weka* for breast cancer wisconsin data.

thesis.pdf – an electronic copy of this thesis