CHAPTER 2

THEORY OF THE ARTIFICIAL NEURAL NETWORK

2.1 Introduction

This chapter covers the basic theory of the artificial neural network which are its definition, network structure, transfer function and applications.

Artificial neural network is one branch of the artificial intelligence (AI). Now it is widely used for solving problems that are difficult for computing with conventional computers or human. Artificial neural network is inspired by biological systems so it imitates human's brain. An artificial neural network is comprised of nodes or elements and their connections. We can train an artificial neural network to perform a particular function by adjusting the values of the connections or the weights between elements.

The network will be adjusted based on the comparison of output and target until the output meets the target. Learning process of the artificial neural network is shown in Figure 2.1.

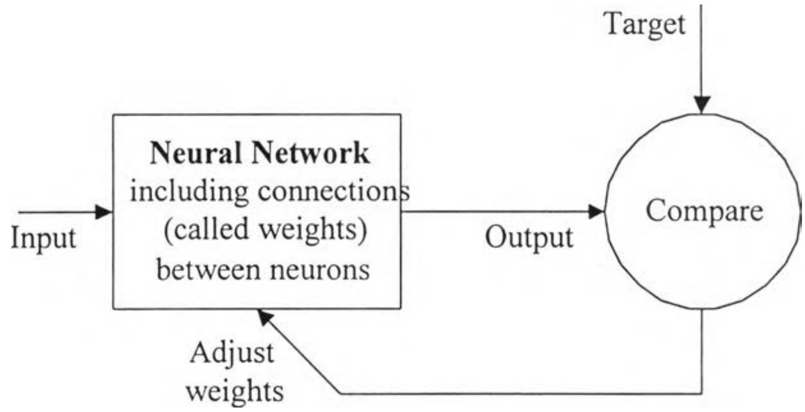

***Figure 2.1**: Learning Process of the Artificial Neural Network*

2.2 Definition

Since artificial neural network has been developed for many decades, many people have defined it.

Medsker, Turban and Trippi (1993) defined artificial neural network as an information-process system that imitates biological neural network.

Skapura (1995) described that a neural network is collection of simple, analog signal processor, connected through links called connection.

2.3 Applications of neural network

Neural network have been found that it is widely used in many fields such as Aerospace, Automotive, Banking, Defense, Electronics, Entertainment, Financial, Industrial, Insurance, Manufacturing, Medical, Oil & Gas, Robotics, Speech Recognition, Securities, Telecommunications and Transportation.

2.4 Neuron Model

2.4.1 A simple neuron

One neural network is comprised of several neurons. A model of a simple neuron that explains the computation of a neuron is illustrated in Figure 2.2.

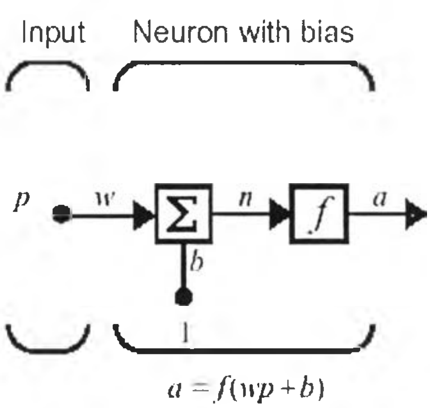

***Figure 2.2**: A simple neuron model*

The scalar output "a" of a neuron is demonstrated by the equation

$$a = f(wp+b) \qquad \textbf{(2.1)}$$

Where "p" is a scalar input,

"w" is a weight,

"b" is a bias,

"f" is a transfer function

and "a" is an output of the network.

The sum of weighted input and bias will be compared with the threshold activation value by the transfer function. If the sum meets the threshold value, this neuron will transfer an output to its neighbor.

2.4.2 A neuron with vector input

A model of a neuron with vector input is shown in Figure 2.3

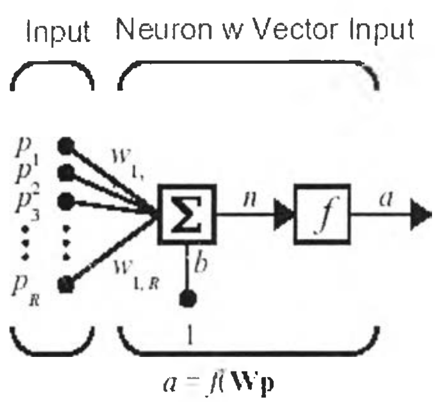

***Figure 2.3**: A Neuron with Vector Input*

The scalar output "a" of a neuron is demonstrated by the equation

$$a = f(n) \qquad \textbf{(2.2)}$$

$$n = w_1p_1 + w_2p_2 + w_3p_3 + \ldots + w_Rp_R + b \qquad \textbf{(2.3)}$$

Where "p" is a vector input,

"w" is a matrix of weight of each input,

"b" is a bias,

"n" is a sum of weighted input added with bias, and "*f*" is the transfer function.

A neuron with vector input, like a simple neuron, the sum of weighted input and bias will be compared with the threshold activation value by the transfer function. When the sum meets the threshold value, this neuron will transfer an output to its neighbors.

2.5 Characterization

An artificial neural network is characterized by three characteristics, which are as follows: Network Architecture, Transfer Function, and Learning Process. The details of network architecture, transfer function and learning process are described in sections 2.5, 2.6 and 2.7 respectively.

2.6 Network Architecture

The architecture of a network concerns with the number of layers in the network, layer's transfer function and number of neurons per layer.

2.6.1 A layer of neurons

A layer of neurons is comprised of several neurons positioned in parallel. Its model can be described as shown in Figure 2.4. Note that now the output "a" is a vector output.

2.6.2 Multiple-layer network

A neural network can contain many layers of neurons. A network shown in Figure 2.5 is an example of multiple-layer network. Figure 2.5 presents a network with two layers of neurons. The first layer of neurons is called the hidden layer because its position is between the input layer and the output layer so that it has no connections to the outside.

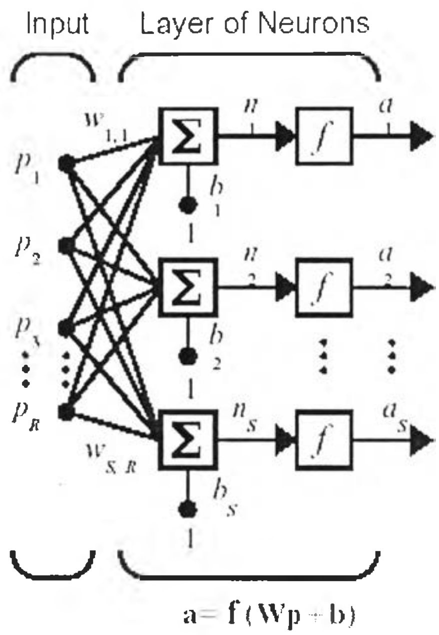

Figure 2.4: A layer of neurons

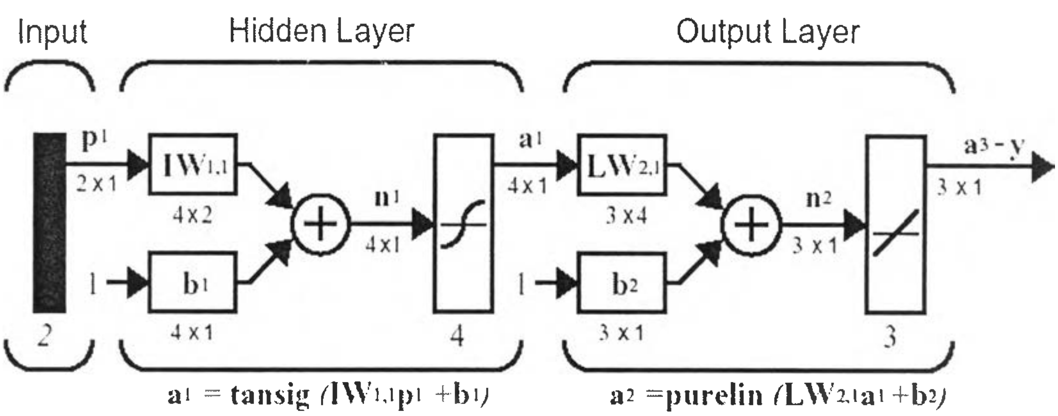

Figure 2.5: An example of a two-layer network

2.7 Transfer functions

There are many transfer function or activated function. Mathematics functions that are commonly used as transfer function are described as follows:

2.7.1. Hard-Limit Transfer Function

The hard limit transfer function as shown in Figure 2.6 takes the input and limits the output to be either 0 if the value of "n" is less than 0; or 1 when "n" is more than or equal to 0.

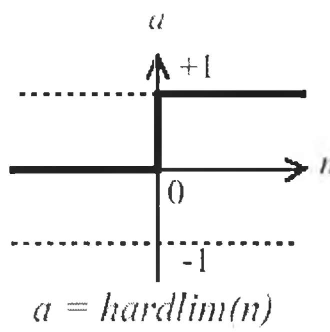

Figure 2.6: *Hard Limit Transfer Function*

2.7.2 Linear Transfer Function

Linear Transfer Function as shown in Figure 2.7 takes the input and produces output as its input. The range of input is any value from minus of infinity to infinity and the range of output also can be any value.

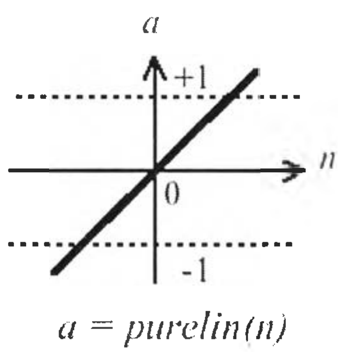

Figure 2.7: *Linear Transfer Function*

2.7.3. Log-Sigmoid Transfer Function

Log-Sigmoid Transfer Function as shown in Figure 2.8 takes the input that can be any value and generates the output that ranges from 0 to 1.

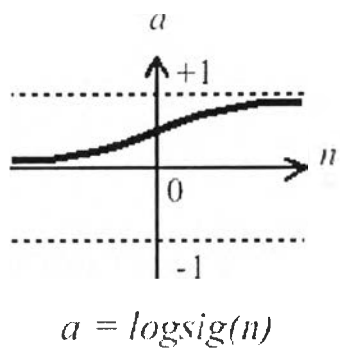

Figure 2.8: Log-Sigmoid Transfer Function

2.7.4. Tan-Sigmoid Transfer Function

Tan-Sigmoid Transfer Function as shown in Figure 2.9 takes the input that can be any value and generates the output that ranges from -1 to 1.

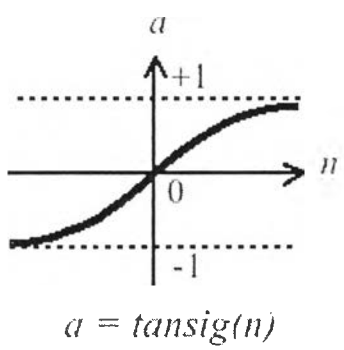

Figure 2.9: Tan-Sigmoid Transfer Function

2.8 Learning process

Learning process of the neural network can be divided into two styles, which are batch training and incremental training. In batch training the weights and biases are adjusted for the entire set of inputs and targets. In incremental training, training the weights and biases are adjusted for each input that is presented to the network.

2.9 Backpropagation

It can be said backpropagation is the most commonly used network for problems solving in the fields of artificial neural network. Its ability to learn the complicated relationships between the training input and its targets makes it become a standard network for forecasting.

2.9.1 Principle

The operation principle of backpropagation is gradient descent that is used to update the weights and biases. Demuth and Beale (2000) have described gradient descent as the process of making changes to weights and biases where the changes are proportional to the derivatives of network error with respect to those weights and biases.

2.9.2 Architecture

Backpropagation network is a multi-layer network that mostly consists of the input layer, hidden layers and the output layer. The number of inputs to the network is the number of variables that is considered to be the input of the problem needed to be solved. The number of neurons in the output layer is determined by the number of outputs desired by the problem. The number of hidden layers and the number of neurons in the hidden layer are up to network designer.

2.9.3 Transfer Function

Log-Sigmoid, Tan-Sigmoid and Linear Transfer Functions are most commonly used in backpropagation network. The selection of transfer function depends on the desired output of the problems.

2.9.4 Algorithm

For the algorithm of backpropagation, weights and biases are updated in the directions of the negative gradient, which can be demonstrated by the equation

$$x_{k+1} = x_k - \alpha_k g_k \tag{2.4}$$

where x_k = a vector of current weights and biases

g_k = the current gradient and

α_k = learning rate

Learning algorithms that will be used in this experiment are batch gradient descent and batch gradient descent with momentum.

1. Batch gradient descent

Batch means that weights and biases will be updated when all of the inputs are applied to the network. So in batch gradient descent learning style, the update of weights and biases based on negative gradient will be made when all entire set of inputs are presented to the network.

2. Batch gradient descent with momentum

Batch Gradient Descent with Momentum is the addition of batch gradient descent. Momentum helps a network to response to the recent trend in the error surface not just to the local gradient. The important benefit of momentum is to prevent network from settling in the local minimum that is one major problem of backpropagation algorithm. Batch gradient descent with momentum also provides faster than training that batch gradient descent does.

Momentum constant ranges from 0 to 1 where O represents the update of weights based on just the gradient and 1 represents the update of weights based on just the recent trend. Universally, the momentum constant is set to 0.9.

2.9.5 Problems

Although the backpropagation is the general-purpose network for many problems, users could find some problems on its use. Problems that should be mentioned to network designer are:

1. Overfitting

Overfitting occurs when there are too many neurons in the hidden layers that can result in high fluctuations at the fitting curves. In contrast, if there are too few neurons in the hidden layer, underfitting can occur.

2. Local Minimum

Some networks can generate wrong results because of local minimum. Since there can be more than one error surface, in some cases, a network is stuck at local minimum not the global minimum, users have to initialize and train network many times to be assured that the global minimum is found.

2.10 Development Process of Neural Network Model

Development process of neural network model concerns data selection and preparation, network design, training and testing the network.

2.10.1 Data selection and preparation

1. Consider variables to be input and output.
2. Collect Data and divide it into two groups, which are training data and testing data.
3. Transform all data to appropriate format.

2.10.2 Network design

1. Specify the number of input to the network and the number of outputs required from the network.
2. Specify number of hidden layers, number of neurons and transfer function of each hidden layer

2.10.3 Training and testing

1. Specify training parameters and then train the network.
2. Test the network.

Because users cannot know which neural network model will be the best for a given problems, development process of neural network model should be repeated by adjusting parameters in order to obtain the model that gives the minimum error of the testing data.

2.11 Literature Survey

Related studies on water demand forecasting and the application of Artificial neural network are summarized as follows:

Ounvichit (1996) developed neural network models for crude oil price time series prediction. The developed neural network model is able to learn the taught crude oil price information and accurately predict future price. Standard back propagation neural network models are used in this study. The result shows that the designed neural networks can learn crude oil price pattern and can predict the price accurately. The study indicates that the number of neurons in input layer should not be too large or too small as it takes longer to learn and shows no gain in accuracy in prediction.

Rurkhamet (1997) developed neural network models for forecasting the requirements of new issued banknotes. Widrow-Hoff and backpropagation techniques are used to forecast new issued banknotes of the year 1993-1996 using the preceding historical data of 12-15 years to train and test the models. In this study, backpropagation technique provides the best forecasting results. In comparison with the traditional regression technique used by the Bank of Thailand, Backpropagation can give more accurate results.

Vasinpongvanit (1999) examined the variables that have effects on water demand to find out how they are related. From Multiple regression analysis, three independent variables that have important effects on water demand are Gross Provincial Product (GPP), water price and population per household.

2.12 Conclusion

Theory on artificial neural network presents itself, as a new effective technique for problem solving but the performance of it is what the users themselves need to prove. Existing applications has shown that the artificial neural network can be used as the alternatives to the existing method.