CHAPTER II

THEORIES ON NEURAL NETWORKS

2.1 Introduction

Computer technology is so pervasive that people have come to accept computers as an integral part of everyday lives. Modern computer technology is inexpensive, reliable, and extremely fast. However there are many types of problems that conventional computer system simply cannot address. These problems can be classified in several different ways. Some are nonpolynomial problems since the amount of time required to solve them cannot be expressed in polynomial function of their sizes. Some are intractable, i.e. they are difficult to be identified in terms of the conventional. Examples of such problems are vision and image interpretation, weather forecasting, handwritten character recognition and so forth.

People perform these difficult tasks without being consciously aware that they are solving the problems. These neurons execute at a maximum rate of about 1,000 times per second while computers can work more than a billion times per second [18]. It is a suggestion of benchmarking for speed that human brain operates at about 400-500 hertz (Hz) [20]. Modern digital computer typically operate at clock speeds between 100 and 200 megahertz (MHz), which means that they have a very large speed advantage over the brain. However, this advantage is dramatically reduced because digital computers operate in a serial mode whereas the brain operates in a parallel mode. Here is an underlying reason why artificial neural network has been addressed in a number of various applications.

2.2 Definitions of Neural Network

A neural network is a computing system that imitates intelligent behavior [6]. It is made up of a number of simple, highly connected processing elements and processes information by its dynamic state response to external inputs.

Neural network is defined as an information processing technology inspired by studies of the brain and nervous system [12]. An artificial neural network (ANN) is a model that emulates a biological neural network.

A definition is given in general that a neural network is a collection of simple, analog signal processors, connected through links called connections [18].

Artificial neural networks is defined as "a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain" [20].

Artificial neural networks are able to improve the performance of several existing technologies. They have been used in a broad range of applications including pattern classification, pattern completion, function approximation, optimization, prediction, and automatic control. They performs either impressively or disappointingly. Examples of the areas where neural networks work best are classifying data, modeling and forecasting, and signal processing.

Wherever the technologies such linear and polynomial regression techniques, auto-regressive (integrated) moving average (ARMA & ARIMA), and Box-Jenkins are introduced, neural networks should also be considered [11].

Examples of forecasting are currency exchange rate forecasting [16]. The main problem with the traditional approach is that the full knowledge of the system rules are not available, especially macroeconomics problems.

The performance of neural models often depends on the specific nature of the data set. They are capable of learning relationships from data and represent a class of robust, non-linear models.

2.3 Characterization of Neural Networks

Neural networks are characterized in three ways: 1) architecture, 2) transfer function and 3) learning paradigm used for training the network. Details of the characteristics are described in 2.4 through 2.9.

2.4 Network Structure

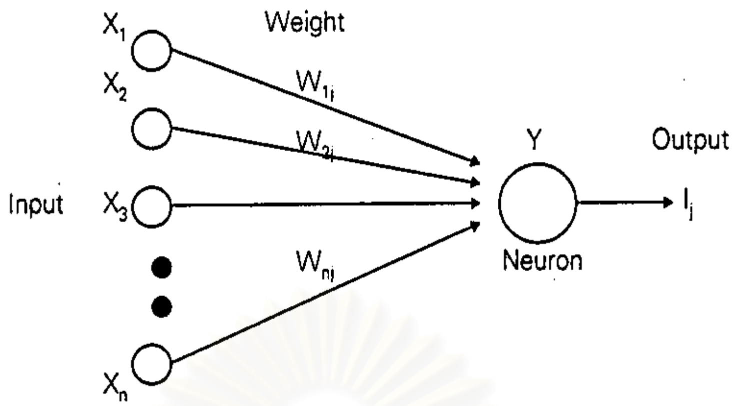


Figure 2.1 - A Single Artificial Neuron

Figure 2.1 shows a simple artificial neuron consisting of a node (Y), called neurode, and its associated links. The value of the node, Y, is the sum of all the weighted input signals. This value is compared with the node's threshold activation level. When the value meets the threshold level, the node transmits a signal to its neighbors.

Each input connection to the unit has a weighting value associated with it. Weights (W_1 to W_n) are numeric estimates of connection strengths. They are assigned to the links between nodes. Positive weights indicate reinforcement; negative weights are associated with inhibition. Connection weights are "learned" by the network through a training process. (See 2.5)

Sum of weighted inputs described above is demonstrated by the equation

$$I_j = x_1w_1 + x_2w_2 + \ldots + x_nw_n = \sum_{j=1}^{n} x_jw_i \quad \textbf{(2.1)}$$

An activation function as an algorithm for computing the activation value of a neurode as a function of its net input [6]. An activation function is the unit which produces an output signal that is related to its activation by a transfer function [18]. An activation function is described as "a squashing function" or "a transfer function" or "a gain function" which indicates that this function squashes or limits the values of the output of an artificial neuron to values between the two asymptotes [20].

Activation function (Φ) is demonstrated by the equation

$$Y = \Phi(I) \tag{2.2}$$

There are several popular and versatile activation functions including the linear unit, binary-threshold units, sigmoidal units, competitive units and Gaussian units. A benefit of activation functions is to remove the nonlinearity from the artificial neuron when linear activation function is introduced. These functions are shown in Table 2.1.

Table 2.1 - Activation Functions

Activation Functions	**Equations**	**Range**
The Linear Unit	$f^{l}_{i}(I) = I$	any value
Binary-Threshold Unit	$f^{bt}_{i}(I) = 1$ if $I > \theta$ 0 otherwise	1 or 0
Sigmoidal Units	$f^{S}_{i}(I) = \dfrac{1}{1 + e^{-(I-\theta)/\tau}}$	between 0 and 1
Competitive Units	$f^{C}_{i}(I) = 1$ if $I = \max\{I\}$, $1 \le j \le n$ 0 otherwise	1 or 0
Gaussian Units	$f^{g}_{i}(I) = e^{(I-1)/\sigma 2}$	between -1 and 1

Remark: θ is the point of transition between the two states.
τ is the shape of the curve.
σ is smoothing parameter.

The processing elements are usually grouped together into a layered structure , when element is known as a layer. The top layer is the output layer which presents the output response to a given input. A typical neural network is “fully connected”, which means that there is a connection between each of the neurons in any given layer with each of the neurons in the next layer as shown in Figure 2.2 and 2.3. There are many different neural network models nowadays [12]. Associative memory, hidden layer and double layer are representative architectures as shown in Figure 2.2.

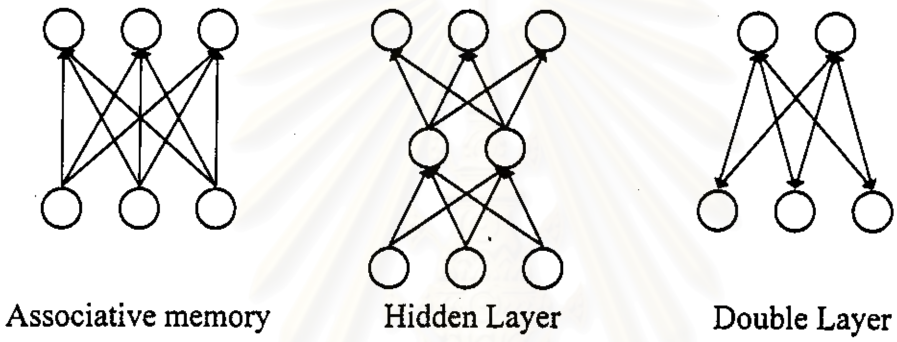


Figure 2.2 - Neural Network Structures

2.4.1 Associative Memory

Associative memory is the ability to recall complete situations from partial information. It can detect similarities between new input and stored patterns. An example of associative memory is Hopfield network, (See 2.9), which uses the collective properties of the network and minimizing of an energy function to classify input patterns.

2.4.2 Hidden Layer

The layer between input layer and output layer is called hidden layer as shown in Figure 2.3. It is called the hidden or intermediate layer because it usually has no connections to the outside world. Backpropagation learning algorithm is introduced for many multilayer networks.

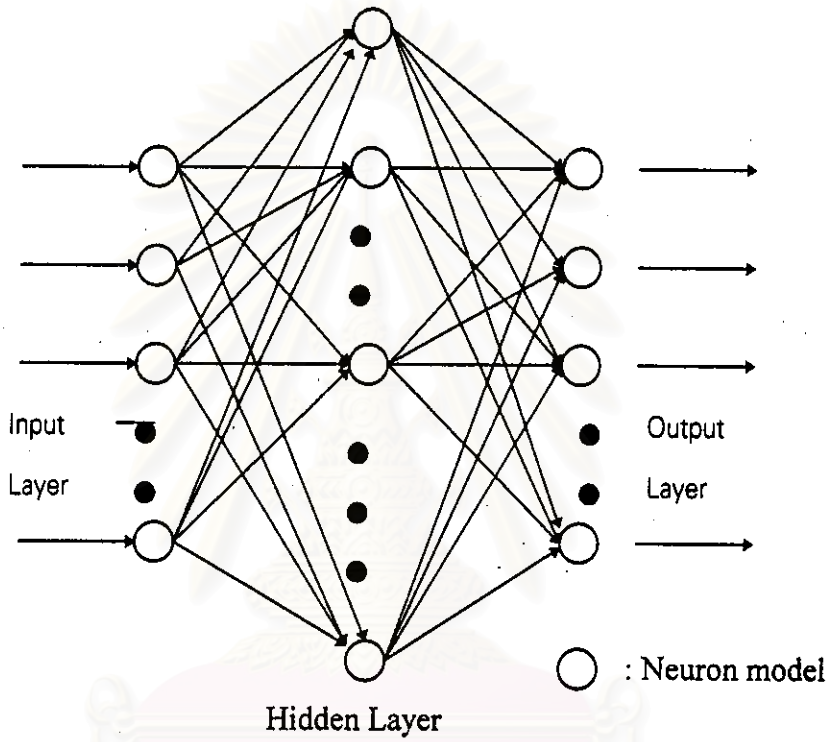


Figure 2.3 - Three-Layer Network

2.4.3 Double-Layer Structure

A double-layer structure, exemplified by the adaptive resonance theory (ART) approach. There is no requirement for the knowledge of a precise number of classes in the training data. It uses feed-forward and feed-backward to adjust parameters as data are analyzed to establish arbitrary numbers of categories that represent data presented to the system.

2.5 Learning Process

Learning is the process of adapting the connection weights in an artificial neural network to produce the desired output vector in response to a stimulus vector presented to the input layer. Neural networks learn the application data patterns by modifying a set of weight values contained in its internal structure. The knowledge contained by the network is modified. Learning methods may be generally grouped as supervised, unsupervised and reinforced. The appropriate choice of learning method depends on data available.

Supervised learning is sometimes called learning with a teacher; the teacher tells the network what the right answers are. The network is composed of a training set having input and target vectors. An input vector is paired with a desired output or target vector. Neural networks'objective for supervised training is to discover the pattern underlying the relationship between the input and the target output. Examples of supervised learning paradigms are perceptron and backpropagation. (See 2.6 and 2.8)

In unsupervised learning, there is no teacher and no right and wrong answers. It requires only input vectors to train the network; no target vector exists. The objective is to detect and identify patterns in the input. The system self-organizes until a consistent output is produced.

Reinforcement training is a compromise between supervised and unsupervised training. It requires an input and only a grade or reward signal as the desired input [20].

Training is accomplished by adjusting the network weights in order to minimize the error which is the difference between the desired and the actual network outputs. Generally the process of learning or training involves three tasks: 1) compute outputs, 2) compare outputs with desired answers and 3) adjust the weights and repeat the process, as demonstrated in Figure 2.4.

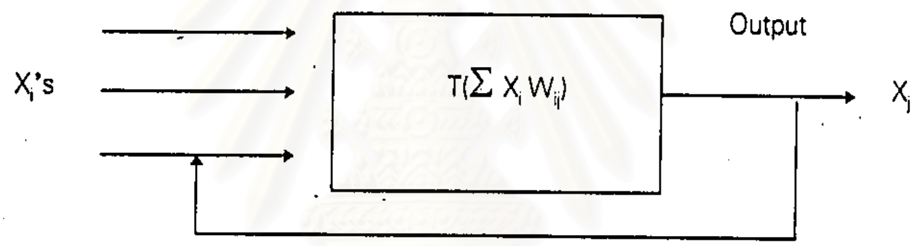


Figure 2.4 - A Neural Processing Element

2.6 Perceptron

Perceptron was created by Rosenblatt in 1958, during the Age of Camelot. (See 2.16 Neural Network Development History) It was criticized by Minskey and Papert in 1969 until neural network got into the Dark Age.

2.6.1 General Principle

Perceptron is trained by supervising. The training technique is called "perceptron learning rule". It is suitable for simple problems in pattern classification. There is only one layer due to its limited capability. This rule is applicable only for binary neuron response. Therefore it is proper for simple problems in pattern classification.

The algorithm of perceptron learning rule is executed by comparing the output with the actual. If the output equals to the actual, the training is stopped and neuron bias remain unchanged. Otherwise the neuron bias will changed by either plus 1 or minus 1 (these values can be defined). The rule is also applied with neuron weight. This can be demonstrated by two equations as follows:

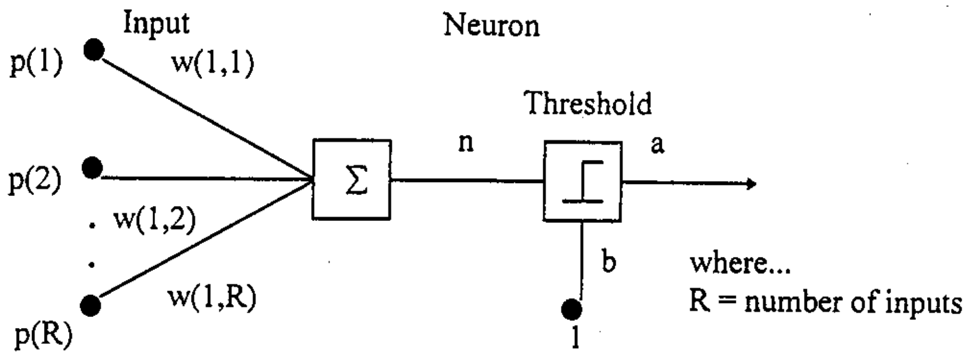
$$W(i,j)_{new} = W(i,j)_{old} + [T(i)-A(i)] \times P(j) \tag{2.3}$$

$$B(i)_{new} = B(i,j)_{old} + [T(i)-A(i)] \tag{2.4}$$

where W = weight, B = bias, T = target vector,

P = input vector, and A = network output vector

A hard limit transfer function is introduced for the perceptron neuron. Refer to Figure 2.5, a is the output of the function "hardlim" which can be either 0 or 1. The variables - w, p and b are weight, input and bias respectively. This function may be called "Binary-Threshold Unit" which is mentioned earlier in Table 2.1. The output is shown in Figure 2.6.



a = hardlim(w*p+b)

Figure 2.5 - Perceptron Neuron

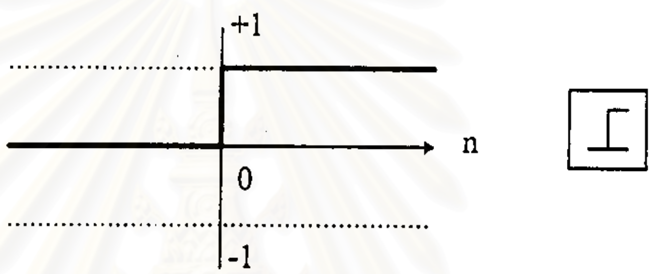


Figure 2.6 - Hard Limit Function

Architecture of perceptron network is a single layer having R inputs and S neurons with a set of weight w(i,j) as shown in Figure 2.7.

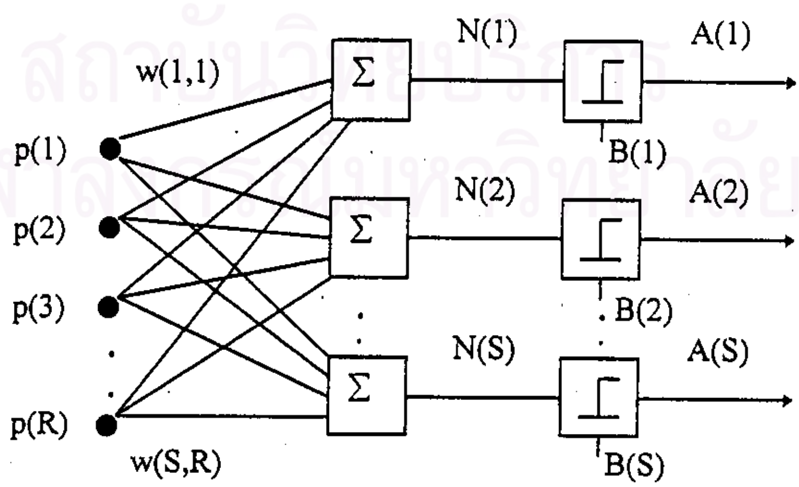


Figure 2.7 - Perceptron Architecture (Mutioutput Perceptron)

2.6.2 Limitations

Since the perceptron learning rule is competent to train only a single layer, hence the network can have only one layer. The capabilities of perceptron are limited in only simple classification problems as mentioned before. The limitations of perceptron can be conclude as followings:

1. Perceptrons can only classify linearly separable sets of vectors (Figure 2.8 (a)). An example of linear nonseparable problem is exclusive-or problem as demonstrated in Figure 2.8 (b). This problem is defined as in Table 2.2.

Table 2.2 - XOR Problem

X	Y	Output
0	0	1
0	1	0
1	0	0
1	1	1

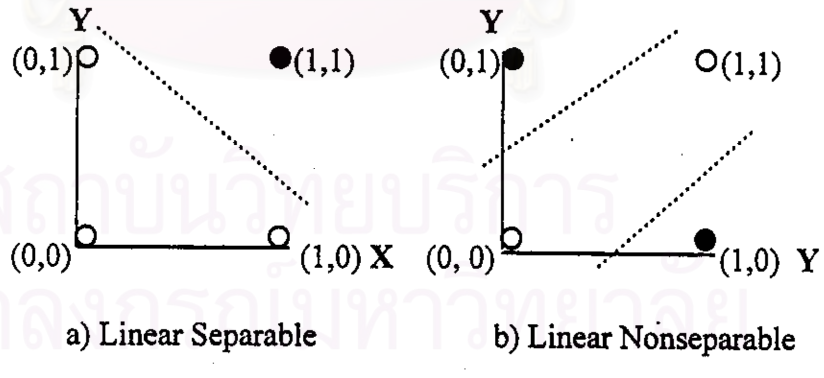


Figure 2.8 - Linear Separable Compared to Linear Nonseparable

2. Output values of a perceptron can take on only one of two values (0 or 1) due to the hard limit transfer function.

Another basic concern of neural network is training period. Training time depends on input vectors mainly. If there is an outlier input, it will consume more time to train to find a solution. If the solution is not possible to be solved, the result will never be correct.

2.7 Widrow-Hoff Alogorithm

The Widrow-Hoff algorithm yields learning that is faster and more accurate. It is a form of supervised learning that adjusts the weights (gains) according to the size of the error on the output of the summer. Another name for Widrow-Hoff method is the least mean square (LMS).

Widrow-Hoff algorithm was developed during the Age of Camelot. (See 2.16) Bernard Widrow and Marcian Hoff introduced a device called an "adaline" which consists of a single neurode with an arbitrary number of input elements that can take on values of plus or minus one and a bias element that is always plus one.

2.7.1 General Principle

An adaline or adaptive linear element is used in Widrow-Hoff learning rule. From Figure 2.9 and 2.11, the neuron model includes a linear transfer function (Figure 2.10) which can take any value of outputs and that makes this learning rule different from perceptron. Steepest descent technique is introduced here to find the minimum sum-squared error. The learning rule can be demonstated by these two equations.

$$W(i,j) = W(i,j) + lr[T(i) - A(i)]P(j) \tag{2.5}$$

$$B(i) = B(i) + lr[T(i) - A(i)] \tag{2.6}$$

where W = weight, B = bias, lr = learning rate, T = target vector,

P = input vector, and A = network output vector

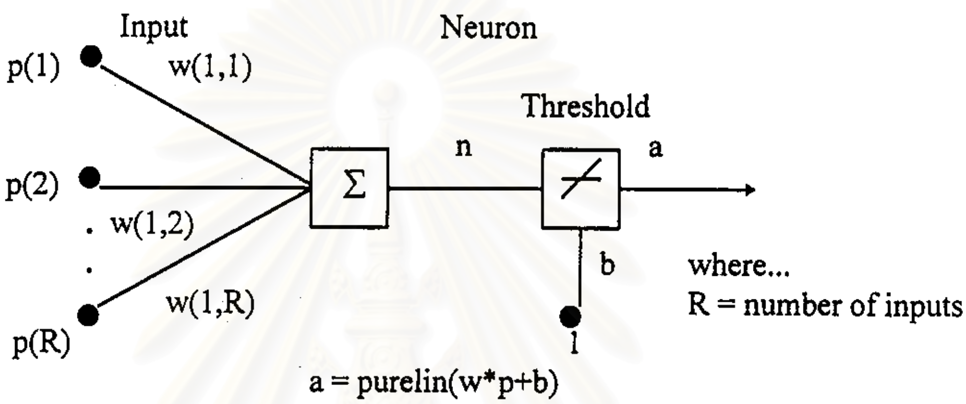


Figure 2.9 - Widrow-Hoff Neuron

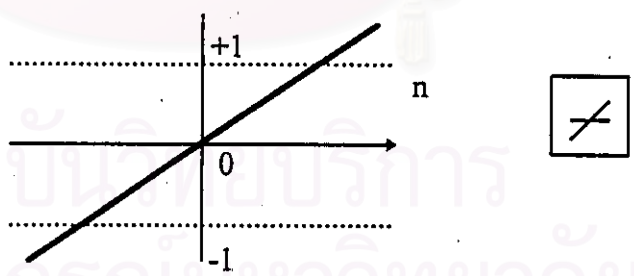


Figure 2.10 - Linear Transfer Function

Adaline (adaptive linear element) is a neural network that adapts a system to minimize the error signal using supervised learning, as shown in Figure 2.11.

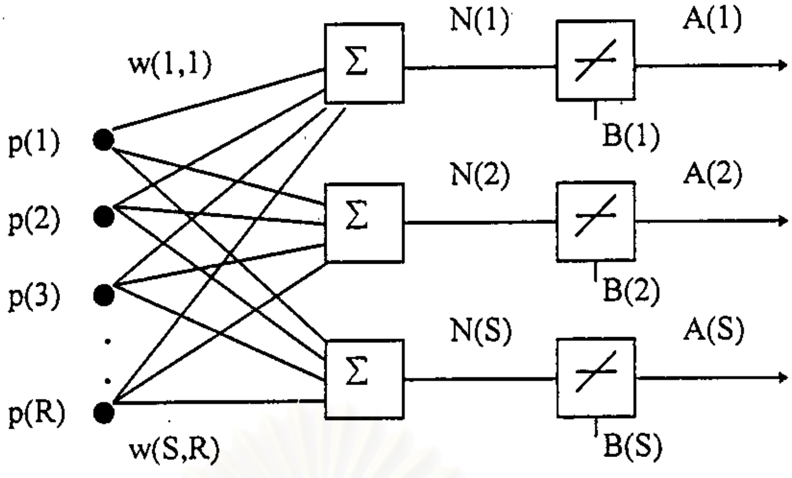


Figure 2.11 - ADALINE Network

Minimizing the sum-squared error involves an error reduction method called "gradient descent" or "steepest descent". The sum-squared error is obtained by measuring the error for each pattern presented to the adaline.

2.7.2 Problems and Limitations

The possible problems are overdetermined and underdetermined. The first problem "overdetermined" means that although Widrow-Hoff can minimize the error, error is too much high. The second problem "underdetermined" is caused by having more variables than constrains (the number of input/target vector pair).

The first limitation is similar to perceptron, the Widrow-Hoff rule can only train single layer linear networks. The second one is that ADALINE may only learn linear relationship between input and output vectors.

An extension of the Widrow-Hoff learning algorithm is used in back-propagation networks.

2.8 Backpropagation

Backpropagation was first developed by Werbos in 1974 as part of his Ph.D. dissertation at Harvard University, then Parker in 1982 and Rumelhart, Hinton and Williams in 1986. It has almost universally become the standard network paradigm for modeling, forecasting, and classification.

Eighty percentages of all applications utilize this backpropagation algorithm [20]. Backpropagation is the most general purpose, and commonly used neural-network paradigm [18]. Sometimes it is called as a multilayer perceptron.

It is designed to train a feedforward network, overcomes some of perceptron's limitations by making it possible to train a multiple-layer network. It is an effective learning techniques that is capable of exploiting the regularities and exceptions in the training sample.

2.8.1 General Principle

Nowadays backpropagation is the most widely used neural network training method. The backpropagation algorithm consists of two phases: forward-propagation and backward-propagation. Gradient descent is its fundamental operating principle which is analogous to an error-minimization process. It minimizes the mean squared error of the system by moving down the gradient of the error curve. The error surface is multidimensional and may have many local minima.

It is used to adjust the weights and biases of networks in order to minimize the sum squared error of the network. Changes that are made to weight and bias are calculated from these following equations.

$$\Delta W(i,j) = lrD(i)P(j) \tag{2.7}$$

$$\Delta B(i) = lrD(i) \tag{2.8}$$

where W = weight, B = bias, lr = learning rate,

D = delta vector, and P = input vector.

The generalized delta rule is used with the back-propagation of error to transfer values from internal nodes. It is used to determine the error for the current pattern contributed by every unit in the network. The sigmoidal function is the activation function specified in neural networks and is used to adjust weights associated with each input node.

Backpropagation will minimize the least squared error if the model does not get trapped in a local minima and there are an adequate number of nodes in the hidden layer.

A transfer function is included in the architecture of backpropagation using F as the symbol in Figure 2.12. The output is calculated from F(W*P+b). There are three types of transfer functions which are log-sigmoid function, tan-sigmoid function, and linear function as shown in Figure 2.13.

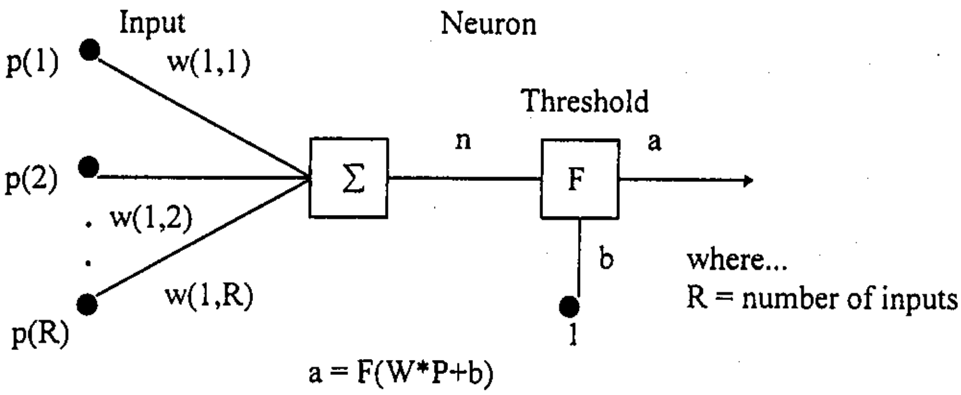
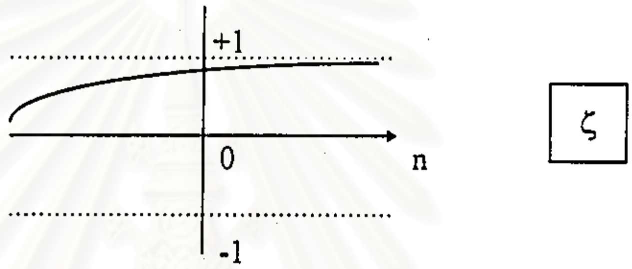
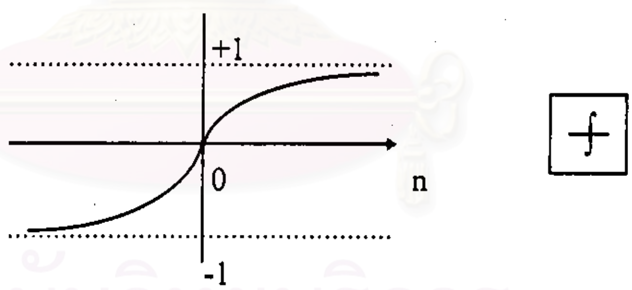


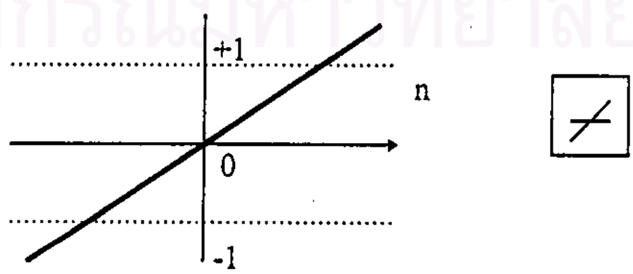
Figure 2.12 - Backpropagation Neuron



a) Log-Sigmoid Function



b) Tan-Sigmoid Function



c) Linear Function

Figure 2.13 - Transfer Functions

The most commonly used transfer function is log-sigmoid function or logsig function. The logsig function gives the outputs between 0 and 1 where the inputs are from negative to positive infinity. The outputs for tan-sigmoid and linear functions are -1 or 1, and any value respectively.

From Figure 2.14, the log-sigmoid function which is a nonlinear transfer function is chosen to be the transfer function here. It allows the network to learn nonlinear and linear relationships between input and output vectors.

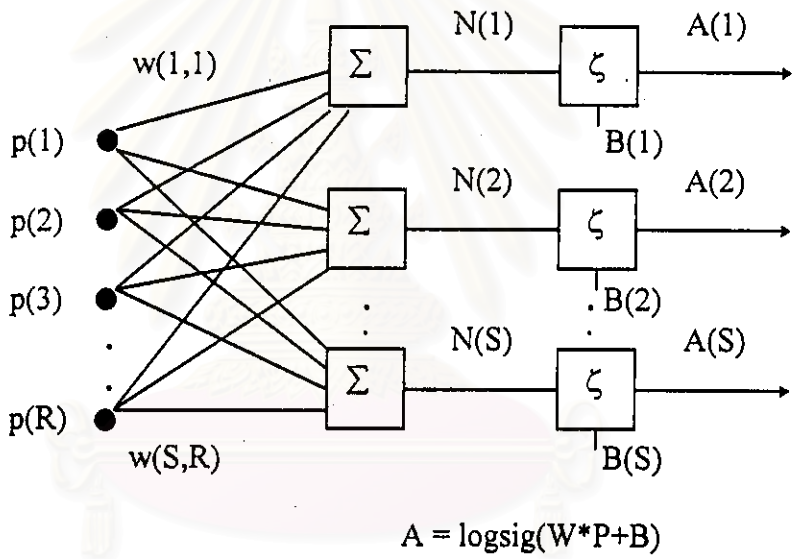


Figure 2.14 - Backpropagation Architecture

2.8.2 Chracteristics of Backpropagation

The characteristics of backpropagation are classified as follows [22]:

1) Learning

The network is capable of learning the most complicated relationships between the training data and their correct classifications. Arbitrary nonlinear decision boundaries in a space of any dimensionality can be extracted solely from the training set without human intervention.

2. Generalization

Inputs that are similar, but not identical to those in the training set will, within limits, be correctly classified. Also, erroneous, noisy, or incomplete training or data inputs do not have a disproportionate effect on the classification accuracy.

3. Concurrency

The inherently parallel nature of the algorithms allows efficient use of the multiprocessor systems. Since the classification time is almost exactly inversely proportional to the number of processors, arbitrary speed improvements can be achieved by adding hardware.

There are numerous variations on the standard back-propagation model. Each of them is slightly different. Examples are the Elman network, the Jordan network, the functional-link network and the probabilistic neural network.

2.8.3 Problems and Limitations

There are two problems associated with backpropagation. They are as follows:

1) Overfitting

This problem arises when the number of hidden units is relatively large with respect to the size of training sample.

2) Local Minima

The major problem of backpropagation is local minima. It is very easy for the training process to get trapped in a local minimum. There is no guarantee of the result from long training periods due to local minima. Since backpropagation employs a form of gradient descent, it is assumed that the error surface slope is always negative and hence constantly adjusts weights toward the minimum.

There are two possible solutions. One solution is changing all weights by specific or random amounts. If it fails, re-randomize the weights and start the training. Another alternative is simulated annealing which is a technique used to search for global minima in a search surface.

3) Training period takes long time. Iterative training of a backpropagation network may take days or weeks.

4) Any modification in backpropagation requires a repetition of the entire training process.

5) A backpropagation system may provide no confidence indication.

6) A problem is called paralysis. When choosing inappropriate parameters, it can cause slow convergence or network paralysis

2.9 The Hopfield Memory

The Hopfield memory is a unique implementation of a recurrent network. A recurrent network is one in which feedback connections are implemented between layers. It is feedback connections only from the output layer directly back to the input layer. Its structure is similar to that of single-layer network with only the interconnections between each unit and itself eliminated.

2.10 Strengths and Weaknesses of Neural Networks

The followings are some strengths of neural networks:

1) No need for knowledge

The primary advantage over conventional modeling techniques are the ability to model complex non-linear processes without assuming any prior knowledge about the underlying data generating process [1,17]. Neural networks do not require knowledge to be formalized. Thus, they are appropriate to domains in which knowledge is inadequate.

2) Ability to handle highly non-linear problems

Neural networks can represent complex nonlinear relationships. They can develop input/output map boundaries that are highly non-linear. Some types of problems have advantage from this capability.

3) Highly correlated data

Neural networks have no problem with highly correlated data while others do, such as expert system. Expert system has difficulties of developing rules from historical data when the inputs are highly correlated.

4) Adaptability

Since the network learns in new environments, training can occur continuously over its useful life, and occur concurrently with the deployment of the network. This allows the model to respond swiftly to changes in the real world.

5) Generalization

When a neural network is presented with noisy, incomplete, or previously unseen input, it generates a reasonable response. Neural networks can accommodate variations in inputs. They are good at filtering out noise and isolating useful input information.

Neural networks can work with noisy and incomplete inputs and produce the correct output by making use of context and generalizing or "filling in the gaps" in incomplete information. This ability to generalize is based on the adaptive structure of the neural net system, rather than on complex programming. They are very good at classification of phenomena into preselected categories used in the training process.

6) Fault tolerance

Since there are many processing nodes, each with primarily local connections, damage to a few nodes or links does not bring the system to a halt.

7) Faster processing time

Time to process each problem of neural networks can be faster than that of complex conventional systems.

8) Robustness

Neural networks do not assume any probability distribution or equal dispersion. Also there is no rigid restriction on the use of input/output functions other than that they be continuous and differentiable.

9) Multimodal distribution

The nonlinear discriminant function represented by a neural network provides a better approximation of the sample distribution, especially when the latter is multimodal. According to Shepenski's report, human judgments are better approximated by a nonlinear function [19].

Weaknesses of Neural Networks are as follow:

1) Difficult to interpret

Justification for results are difficult to obtain because the connection weights do not usually have obvious interpretations.

2) Tedious training

Training period can not be defined exactly. It also demands more computation time than other methods. The training process may be very slow. The time required for proper training a neural network using one of the variations of backpropagation training can be substantial (sometimes hours or days). It is possible to overtrain or undertrain a neural network, resulting in poor performance in real world recognition applications [2].

Neural computing usually requires large amounts of data which produce lengthy training times. The best way to represent input data and the choice of architecture is still mostly subject to trial and error.

3) No guarantee

Most neural networks cannot guarantee on optimal solution or best configuration to a problem. There is no guarantee that the network can be trained in a finite amount of time. Also lacking of interpretability of neural network parameters is reported in a number of applications.

4) Ad hoc

The usage of neural networks is ad hoc. There is no formal method to derive a network classification for a given classification task.

5) Limited precision

The precision of the outputs is sometimes limited because the variables are effectively treated as analog variables, and minimization of least squares errors does not mean zero error.

2.11 Problems and Limitations of Neural Networks

There are several basic problems. It requires guidelines to deal with the huge number of choices and decisions [13]. Without guidelines, the procedure for selecting the structure of a neural network will continue to be essentially a trial and error process. The selection of the training parameters also remains a trial error.

Applying an artificial neural network requires experience, judgment, and patience [22]. The user may not be confident of the results from a given network, training algorithm, and training set. A number of trial and error is inevitable.

The development and interpretation of neural network models requires more expertise from the user than traditional statistical models.

Problems with neural networks are concluded as the followings [9]:

1. Artificial neural networks contain more parameters to estimate than do most statistical forecasting models.

2. Artificial neural networks require more computer time than statistical models

3. Artificial neural network models are harder to interpret, and to give physical meaning, than are many forecasting models.

4. Neural network methodology and modeling techniques are rapidly changing whereas many statistical modeling techniques are stable and well developed.

5. While the software is readily available for statistical techniques, commercial artificial neural network software often lags behind developments in the field.

2.12 Comparison of Neural Network to Regression Analysis

Regression analysis is ranked as one of the most popular quantitative methods used in business and finance. There are comparisons between regression analysis and neural networks that regression analysis provides the parameters of a given functional form but not the correct functional form while neural networks can determine the functional form as well as parameters by tuning both to fit the learning examples [5]. This allows a more general framework for discovering relationships existing in data.

"Artificial neural network (ANN) models provide a viable alternative to classical regression models. According to Wasserman, these models can learn from experience, can generalize and "see through" noise and distortion, and can abstract essential characteristics in the presence of irrelevant data. According to Lippman, these models provide a high degree of robustness and fault tolerance. In addition, artificial neural network models can find the right transformations for variables, detect weak linear relationships, and deal with outliers" [13].

The regression analysis is straightforward but the neural network estimation was more complex. The result from Marquez et al.'s study is that neural network models possess considerable potential as an alternative to regression models [13]. The result of predicting bond ratings is that neural networks consistently outperformed regression methods [5].

The result from Lapedes & Farber at Los Alamos National Laboratories is that neural network approaches considerably outperformed all existing techniques for forecasting chaotic time series [11].

Statistical techniques always require the assumption of a certain functional form for relating dependent variables to independent variables.

However, regression models are useful in determining the right set of independent variables, which determine the dependent variable to the largest extent. It is difficult for neural network with hidden layers since the inputs do not influence directly to the output.

2.13 Development of Neural Network Applications

Network decisions concern transfer functions, learning rules, topology and learning rates. Topology involves number of hidden layers and number of processing elements per layer. Other parameters are the momentum, and the number of hidden nodes - needed to be varied by us to enhance the learning and the generalization performance of the neural network model. Table 2.3 shows the parameters for various network decisions using backpropagation algorithm.

Table 2.3 - Parameters for Back-Propagation Networks

Parameters	
Network Decisions:	
Transfer Functions:	Sigmoid
	Hyperbolic
	Sine
Learning Rules:	Delta Rule
	Cumulative Delta Rule
	Normalized Cumulative Delta Rule
Topology:	Number of Hidden Layers
	Number of Processing Elements
	(PEs) per Layer
Learning Rates:	Connection to Prior Layers
	Learning Rates for each layer
Problem Specific	
Parameters:	Number of Inputs to Network
Number of Input PEs:	Number of Outputs from the
Number of Output PEs:	Network
Min-Max Table:	Required to Normalize Data

Source: Klimasaukas C C, "Applying Neural Networks." *Neural Networks*, Probus, 1993, pp.65

A process of experimentation is used to determine the number of hidden layers, as well as the number of nodes in the hidden layer.

Selection of an appropriate representation of the input data has been shown to be crucial, and still largely an art [21]. Human creativity and understanding of the problem can often discern the essential features, ignore irrelevant details, and greatly improve network performance.

The developer has to make decisions for 1) Learning algorithms, 2) Topology - number of processing elements and their configurations (inputs, layers, outputs), 3) Learning rate for each layer, 4) Size of training data and test data and 5) Select diagnostic and validation tools. Black-box testing (comparing test results to actual historical results) is the primary approach to verify that inputs produce appropriate outputs.

The steps of neural-network development: 1) define the data sources ,2) select a network, 3) define the input pattern, 4) define the output, 5) collect the training exemplars, and 6) train the network [18]. Selecting a network is determined by identifying the characteristics of the application that will influence the paradigm selection.

The first four steps from Skapura are compiled before starting Klimasaukas's process. Klimsauskas's basic process of developing a neural network are as the follows:

1) Gather all the data in one place.

All information relevant to the problem must be brought together. The sources for errors in data collections can be data formatting errors, temporal inconsistencies, and conflicting exemplars.

2) Separate the data into training and testing sets.

As a general rule, the more training examples available, the better the network will ultimately perform. Selecting appropriate test and training sets is an essential step in developing a neural network application. The ideal training set is equally distributed among each of the possible outcomes. The ideal test set is one that is representative of the data as a whole.

3) Transform the data into network-appropriate inputs.

Neural networks accept only numeric inputs. The process of transforming numeric and symbolic inputs is called preprocessing. Back-propagation requires input patterns to be represented as vectors composed of elements that range in magnitude between zero and one.

4) Select, train, and test the network.

Picking the right network configuration can have a substantial impact on the performance of the resulting system. The network performance peaks for a certain epoch size. The testing phase examines the performance of the network using the derived weights by measuring the ability of the network to classify the test data correctly.

5) Repeat steps 1,2,3,4 as required.

6) Deploy the developed network in the application.

2.14 Tools for Developing Artificial Neural Networks

Several tools are available for developing artificial neural networks. In 1990, an application "Analyzing financial health", created by Don Barker, is developed by integrating neural networks and expert systems using KnowledgePro, NeuroShell and dBase III Plus. KnowledgePro is an expert system shell. NeuroShell is used to construct the neural network portion of the program. dBase III Plus was chosen to create the data files for the program.

Other tools are NNetSheet (based on spreadsheet), NEURALWORKS PROFESSIONAL II/PLUS "Back-Prop Builder", ADAM developed and used at Chase Manhattan Bank, ANSIM (Artificial Neural Network Simulator), NeuralWare, NeuralWorks Explorer, NetSet II, Plexi, N-NET 210.

NET-talk and DECtalk are tools that learn the correct pronunciation of words from written text (ASCII characters).

2.15 Applications of Neural Networks

Neural networks have been introduced to multiple areas. They can be used effectively to automate both routine and ad-hoc financial analysis tasks. Some authors mention the similar areas that neural network work best. Three tasks based on pattern recognition have been applied effectively: associative memory, classification, and clustering [7]. The areas where neural network work best are modeling and forecasting, and signal processing [11]. Four areas are classified for which neural networks are generally considered to be best suited: classification, associative memory, clustering and generation of structured sequences or modeling [6].

1) Classification

Classifying data is currently one of the most widely used capabilities of neural networks. It involves the assignment of input vectors to predefined groups or classes based on patterns that exist in the input information.

The examples of data classification are distinguishing sounds of a musician playing an instrument from noisy environment (in a busy subway station), the recognition of handwritten characters, targeted marketing (deciding whom to send mail-order catalogs to), picking winning football teams, predicting future job performance, diagnosing problems with automobile engines, prostate-cancer detection, radar-signature classifier and so forth.

2) Associative memory

An associative memory is any device that associates a set of predefined output patterns with specific input patterns. Neural networks can perform as associative memory or content addressable memory. There are three basic types of associative memories, i.e. hetero-, interpolative, and autoassociative.

Examples of applications of neural networks in associative memory are reconstruction of complete fingerprint from smudged fingerprint and obtaining the complete version of a pattern at the output of the network by providing a partial version at the input.

3) Clustering

For clustering or compression task, neural networks are used to for compressing or filtering input data without losing important information. It is considered a form of encoding. An example is about speech recognition which reduce significantly the dimensionality of an input.

4) Modeling and Forecasting

Modeling and forecasting are concerned with developing mathematical relationships between several continuous input variables and one or more output variables. In forecasting, the input variables consist of samples of the data to predict at several points back in time.

An example of modeling: if a network is trained to reproduce a certain style of musical sequence, then it is possible for the network to compose "original" versions of that type of music.

The predictive ability neural networks falls into forecasting area. The method used for neural network prediction is called generalization [14]. Business applications of neural networks for forecasting include the ratings of corporate bonds, emulating mortgage underwriting judgments, financial and economic forecasting, and prediction of default and bankruptcy.

The most exciting aspect of neural network technology is that it represents a fundamental breakthrough in the ability to approximate complex mathematic mappings.

Neural networks have been applied in many areas. Some interesting applications are described below:

From the study of [3], "Comparing the predictive performance of a neural network model with some traditional market response models", the study compares the performance of two statistical market response models (a logistic regression model and a discriminant analysis model) to that of a back propagation neural network model.

It is found that in the area of classification and forecasting the neural network model performs better than a logistic regression model and a discriminant analysis model in terms of their ability to identify market segments. But the level of performance is not significantly higher than those of the other models.

According to the study, "Artificial neural network models for forecasting and decision making", it compares artificial neural networks and statistical models, particularly in regression-based forecasting, time-series forecasting, and decision making [9].

Another interesting study is "Comparative study of artificial neural network and statistical models for predicting student grade point averages" [10]. The study compares linear regression; stepwise polynomial regression; and fully-connected, single middle layer artificial neural network models with an index used by an admissions committee for predicting student GPAs in professional school. There are two interpretations of the results. One is that there is no underlying structure in the data set to be detected. This means the performance of artificial neural network is not better than that of the simpler models. Another interpretation is that there is an underlying structure in the data but all methodologies were not used in full capacity.

In the area of pattern recognition such as [2], "A Hierarchical neural network architecture for written numeral recognition", it indicates the effectiveness of the proposed architecture which outperforms the backpropagation learning neural network. Another achievement in this area is the study "Off-line signature verification based on geometric feature extraction and neural network classification." The study shows that 90% correct classification rate can be achieved on a database of over 3000 signature.

2.16 Neural Network Development History

The history of neural network development is divided into four segments, which are called ages [6]. The first age, "the Age of Camelot", started with William James from about 1890 to 1969, ended with the publication of Minskey and Papert's book on perceptrons. A significant development during this period is Widrow-Hoff algorithm. The second age, "the Dark Age", started from 1969 to 1982, ended with Hopfield's landmark paper on neural networks and physical systems. The third age, "the Renaissance", started from 1982 to 1986, ended with the publication of Parallel Distributed Processing, Volume 1 and 2, by Rumelhart and McClelland. The fourth age, "the Age of Neoconnectionism", started with Cowan and Sharp's review article on neural nets and artificial intelligence from 1987 to present.

2.17 Discussion

Neural network has been developed for about a hundred years. Comparing to other technologies, neural network is very young. It has been demonstrated for its potential to be introduced in many types of applications. Some applications have been successful with neural network and some have not. It depends on the nature of data of each application which attracts the novice and expert who use conventional statistics to discover and try this new methodology with the hope that it will outperform the existing technologies.

Nowadays there are a number of neural network algorithms. The outstanding ones are perceptron, Widrow-Hoff, backpropagation and so on. The simple one-layer perceptron model is used.for classifying only linear separable data. The result could be either 1 or 0, black or white, on or off which is limitedly used in only some certain areas. The Widrow-Hoff algorithm use gradient descent for minimizing the sum squared error while the output can be any value. Backpropagation is the most accepted methodology of neural network even it consumes much time and effort.