Adaptive Structure Neural Networks

with

Applications to EEG Automatic Seizure Detection

Wei Weng

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

June 1994

ISBN 0-315-97663-2

Abstract

Adaptive Structure Neural Networks

with

Applications to EEG Automatic Seizure Detection

Wei Weng

This thesis proposes a novel approach for Back-Propagation (BP) structure level adaptation for artificial neural networks (ANN). Back-propagation is the most commonly used neural network algorithm. Back-propagation allows the training of the weights in a feed-forward neural network of arbitrary structure by following a gradient steepest decent path in weight space. However, BP networks have limitations due to their fixed network structure. This thesis will show how a BP network may be improved by replacing the fixed network structure with an adaptive one.

To improve the standard BP algorithm, a new scheme designated as Adaptive Structure Algorithm (ASA) is proposed to allow a neural network to adjust its structure according to the characteristics of the input data. To overcome the slow convergence of the BP algorithm, a modified Delta Adaptation (DA) algorithm is used in the ASA to speed up the training time. Simulation results are presented to confirm the improvements obtained as a result of utilizing the proposed algorithms.

To demonstrate a practical application of the proposed algorithm, OSLA is applied to automatic seizure detection in Electroencephalogram (EEG) during long-term monitoring of epilepsy. Satisfactory results are obtained, substantiating the effectiveness of the new algorithm.

DEDICATION

To my parents Ms. Yu-Zhen Liu, Mr. Sheng-Bin Weng, my husband Li-Yan Wang and my daughter Xiao Wang with love and reverence.

Acknowledgments

First of all, I would like to extend my sincere appreciation to my supervisor, Professor K.Khrossani, for guiding my understanding of neural network as well as for all the direction guidance and financial support he gave throughout the preparation of this thesis. My special thanks go to Dr. Gotman and Mr. Qu who supplied useful EEG data for my research.

This thesis is dedicated to my family. My parents, Sheng-Bin Weng and Yu-Zhen Liu, have given me their unreserved encouragement. My husband, Li-Yan Wang, has given me his constant love and full support, without which none of this would have been possible. I am also deeply grateful to my daughter, Xiao Wang, for her understanding in allowing me spend time with computers rather than playing with her.

I would like to extend my special appreciation to Mary O' Malley, Dr. Jin and Dr. Gao for all their review of my thesis. I also wish to thank all the members of the ECE Computer support group especially Nicky who has been extremely helpful. I would like to thank my friends and fellow graduate students, who created a warm environment in which ideas developed rapidly. I wish particularly to thank my friends Dong, Shao, Raghu and Khatri for all those helpful conversation at the research Lab.

This research was supported in part by Natural Science andEngineering Research Council of Canada (NSERC) under Grant number OGP0042515.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The field of Artificial Neural Networks (ANN) has been originated from the biological neuron of human brain. It has brought about revolutionary changes in several application areas such as pattern recognition, control, signal processing, etc. No one can deny that new and challenging concepts arise constantly in this emerging field.

1.1 The biological neuron

The basic anatomical unit responsible for the processing of information in the nervous system is a cell called the neuron. As shown in Figure 1.1(a), the structure of a neuron includes dendrites, the cell body and single axon. The dendrites receive inputs from other neurons and the axon provides outputs to other neurons. The neuron itself is imbedded in an aqueous solution of ions, and its selective permeability to these ions establishes a potential gradient responsible for transmitting information. Neurons receive electrochemical input signals from other neurons to which they are connected at sites on their surface, called synapses (see Figure 1.1(b)). The input signals are combined in various ways, triggering the generation of an output signal by a special region near the cell body. However, the particular interest of the neuron-biological phenomenon is the transmitting or pre-synaptic side of the synapse. The triggering of the synaptic pulse releases a neurotransmitter that diffuses across a gap to the receiving side of the synapses. On the post-synaptic or receiving side, the neurotransmitter binds itself to receptor molecules, thereby affecting the ionic channels and changing the electrochemical potential. The magnitude of this change is determined by many factors local to the synapse, e.g., amount of neurotransmitter released, number of post-synaptic receptors, etc. Therefore, neurocomputation, biological self-organization, adaptive learning and other mental phenomena are largely manifested in changing the effectiveness or "strength" of the synapse and their topology.

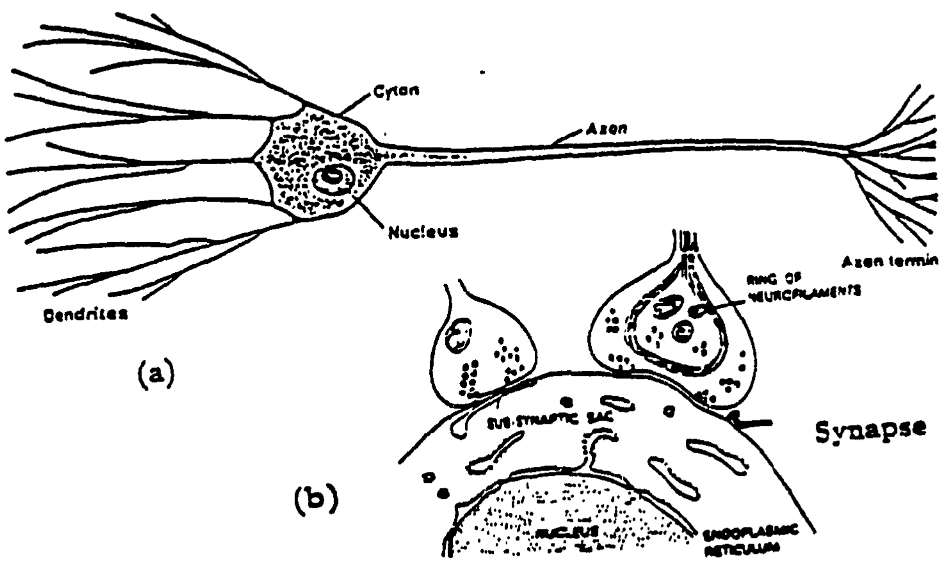

Figure 1.1: (a) Biological neuron

(b) morphology of neuron-to-neuron connection or synapse

1.2 The artificial neuron

The artificial neuron was designed to mimic the first-order characteristics of the biological neuron [1]. As a fundamental building block, the neuron is the basic processor in neural networks. Each neuron receives several inputs over these connection, called synapses. The inputs are the activations of the incoming neurons multiplied by the weights of the synapses. Each neuron has one output, which is generally related to the state of the neuron-its activation- and which may fan out to several other neurons. An abstract model of the neuron is shown in Figure 1.2.

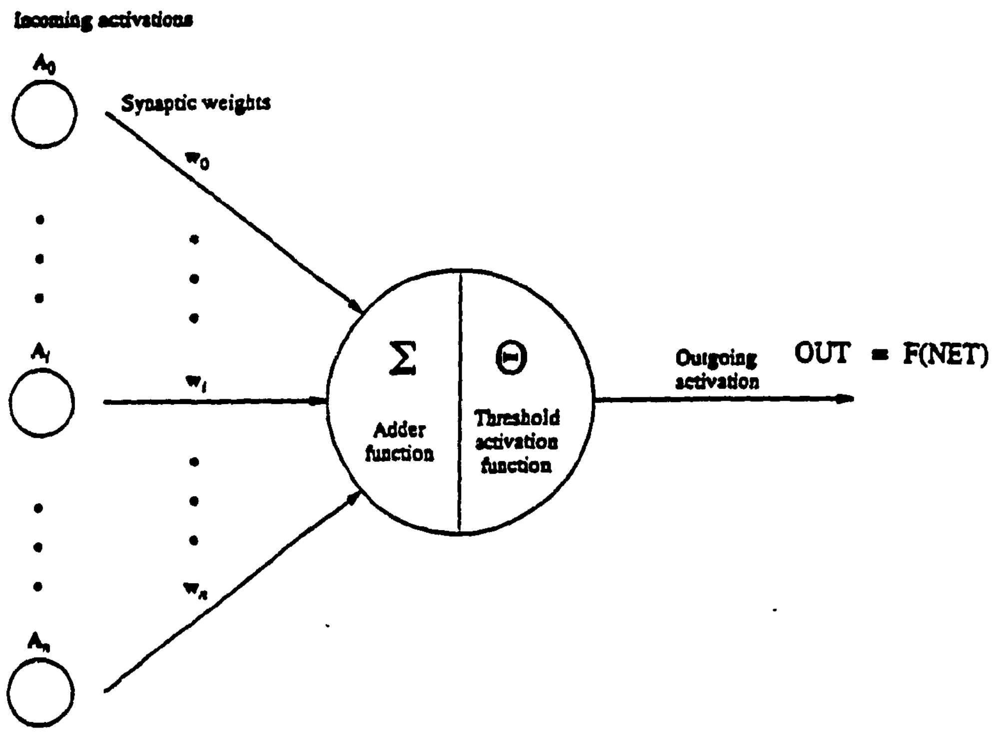

Figure 1.2: Diagram of abstract neuron model

1.2.1 Activation functions

The activation of the neuron is computed by applying a threshold function to this product. This threshold function is produced by a nonlinear activation function F acting on the summed output vector NET as shown in Figure 1.2.

One simple nonlinear function that is appropriate for discrete neural networks is described in Figure 1.3, where x is the summation (over all of the incoming neurons) of the product of the incoming neuron's activation and the synaptic weight of the connection:

$$x = \sum_{i=0}^{n} A_i w_i$$

where n is the number of incoming neurons, A is the vector of incoming neurons, and w is the vector of synaptic weights connecting the incoming neurons to the particular neuron.

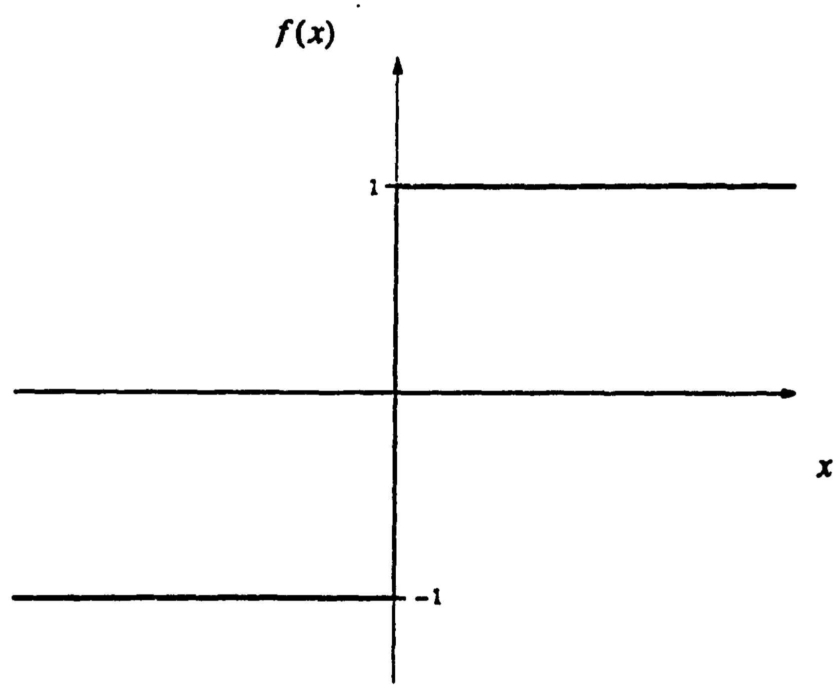

Figure 1.3: Threshold activation function

Another popular function, which is more appropriate for analog networks, is the sigmoid, or squashing function:

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

which is illustrated in Figure 1.4.

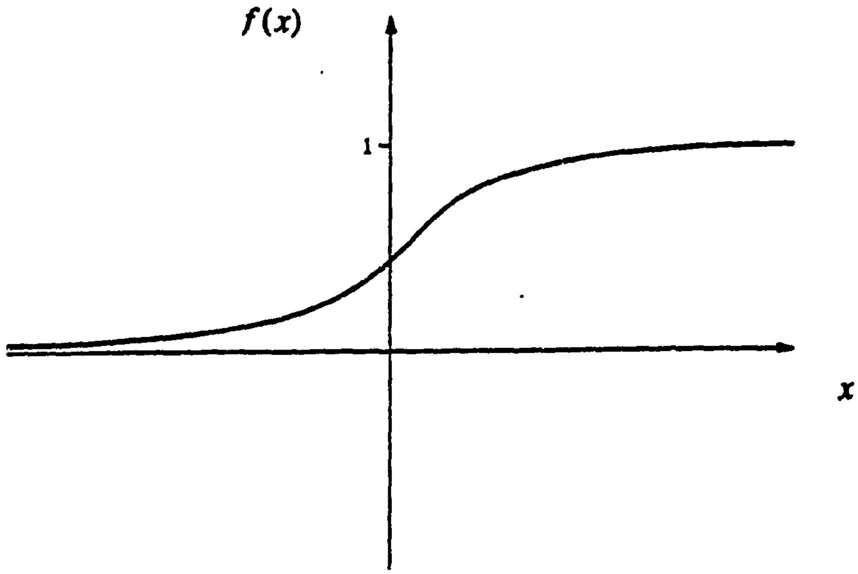

Figure 1.4: Figure of logistic activety function

1.2.2 Learning

Learning is the most important property of ANN. It is defined as a change in connection weight values to capture the information contained in the training data. Learning involves adjusting the weights of the network so that the application of a set of inputs produce the correct outputs. The weight adjustment scheme is known as the learning law. All learning method can be classified into two categories: supervised learning and unsupervised learning.

1. Supervised Learning

Supervised learning involves the association of a target vector representing the desired output values with each input vector. After the output values of the network for a given input vector are computed and compared to the target values, the difference (error) is fed back so that the network weights are adjusted according to an algorithm that tends to minimize the error. The vectors of the learning (training) set are applied sequentially and the learning procedure is repeated until the error for the entire training set is at an acceptable low level.

2. Unsupervised Learning

Unsupervised learning requires no target vectors for the output, and hence no comparison to predetermined output responses. The learning set consists solely of input vectors, and the learning algorithm modifies network weights to produce consistent outputs. The learning process extracts the statistical properties of the learning set and group similar vectors into classes.

1.3 Background and history of ANN

In 1949, Hebb first proposed a learning rule that became the starting point for artificial neural networks training algorithms. Twenty years later, a group of scientists tried to combine the biological and psychological insights as electronic circuits[2],[3], which were later converted to a more flexible medium as computer simulation. Early successes produced a burst of activity and optimism. Marvin Minsky, Frank Rosenblatt, Bernard Widrow, and others developed networks consisting of a single layer of artificial neurons. In 1969, Minsky and Papert published the book Perceptrons [4], which analyzed in detail a single-layer artificial neural network model. Often called perceptrons, single layer networks were applied to such diverse problems as weather prediction, electrocardiogram analysis, and artificial vision. However, because of the limitations of perceptron, neural networks research was not active for the next 10 years.

In 1980s, this field became the center of research focus again. This was partly due to the development of multi-layer learning algorithms which enabled the network to learn (using a more complex structure) difficult problems that perceptrons could not solve.

Back-Propagation (BP) is one of the most well know algorithms which has been widely used in the neural networks field [3]. Numerous successful applications of using BP have been reported in areas such as pattern recognition [5], image processing [6], bio-medical engineering [7],[8], control [9],[10]; etc. However, BP has some limitations. One of the important drawbacks is its fixed network structure. In most practical problems the fixed network structure could either fail to solve complex problems or become too redundant to solve the simple problems. Three different approaches can been used to solve this problem:

1. **Trial and error**: Start with a fixed structure. If the learning process of the selected network does not lead to the desired accuracy, then design a new structure by adding or pruning neurons. Clearly, this is not an efficient approach because no knowledge can be gained from the previous design.

2. **Choose a large number of neurons in the hidden layer**: Two difficulties will arise in using this approach. First, there may be redundant neurons for representing the given function, and thus computational overhead is extensive. Second, more neurons in the hidden layer result in a cost function with additional local minimum points, consequently resulting in a higher probability of getting trapped in a local minimum.

3. **Adaptive structure**: The network has the ability to adapt its structure according to the statistics of the training set. When the inputs of the network change, the network structure adapts to compensate for these variations. This is accomplished provided that the time used for weight adjustment is much faster than the dynamics of the training set. Thus the network structure adaptation provides more flexibility to the input training set.

Structure adaptation is the best approach to overcome BP's limitations. Currently, there are several researchers working on the adaptive structure networks. Tsh-Chang Lee [11], Azimi-Sadjadi [12], Fahlman and Lebiere [13], among others, have proposed several methods to improve the performance of the BP network. A new algorithm for improving the structure of the network adaptively will be presented in this thesis. A practical bio-medical application namely an Electroencephalogram (EEG) automatic seizure detection is included to demonstrate the effectiveness of the proposed network.

1.4 Objectives of the thesis

There are two main objectives in this thesis. First, a new algorithm to realize an adaptive structure neural networks (ASNN) is proposed. Numerical simulations for several classical problems are included to illustrate the superior performance of the new algorithm compared to the BP algorithm. Second, this thesis shows the application of the proposed neural network to seizure detection in EEG and reduction of false seizure detection during long term EEG's monitoring. Furthermore, comparison between ASNN and BP networks are included to demonstrate the superiority of the ASNN.

1.5 Outline of the thesis

The thesis is divided into five chapters. The first chapter introduces the background material for neural networks, objectives and organization of the thesis.

Chapter 2 presents the theoretical foundation. First, the Back-Propagation algorithm is reviewed. Next, an overview of the Lee's method [11] for structure adaptation is given. Finally, ASNN is proposed which includes a new generation rule, Maximum and Minimum Rule (MMR) and Delta Adaptation (DA) Rule.

Chapter 3 applies the new neural networks introduced in chapter 2 to solve several typical benchmark problems.

In Chapter 4, the new algorithm is used to solve a real-world application problem, namely an EEG seizure detection and reduction of false seizure detections. In addition, several comparisons between the new network and the standard BP networks are presented.

Finally, in chapter 5, future research topics and directions for further development in both analytical and practical areas are included.

CHAPTER 2

Methodology and Algorithm

2.1 Literature review: Back-Propagation

The Back-Propagation (BP) network was first proposed by Werbos [14] to recognize computer patterns and perform mapping functions. Later, it was enhanced by Parker [15], Rummelhard and McClelland [16]. The BP network, illustrated generically in Figure 2.1, is designed to operate as a multilayer, feed-forward neural networks, using the supervised mode of learning.

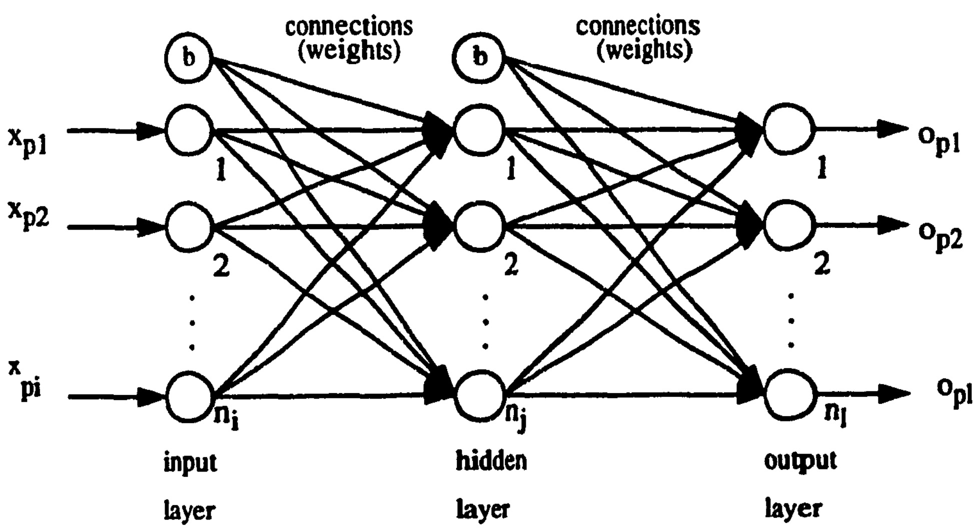

Figure 2.1: Back-Propagation Network Architecture

Back-propagation allows the training of the weights in the feed-forward neural networks of arbitrary structure by following gradient steepest decent path in weight space. The energy surface is usually defined by the mean squared error between desired and actual outputs of the network. Back-propagation has a predetermined network topology. It uses the Generalized Delta Rule (GDR) as the learning algorithm to derive weights for the BP network [1],[17]. Back-propagation iteratively reduces the error in the learning samples by fine turning the weights in the network. Each iteration consists of two stages: feed-forward calculation and error back-propagation. Feed-forward calculation is used in both network training and recall phases. Error back-propagation is used to compute the error derivatives with respect to all the weights in the network. The error derivatives assigned to each weight estimates the effect of each weight on the total error. The error back propagation is applied only during the network training phase.

1. In the feed-forward phase, the output of the network is determined by using the input vector. Let the input vector X_p to the input nodes be denoted by:

$$X_p = \begin{bmatrix} x_{p1} \\ x_{p2} \\ \bullet \\ \bullet \\ x_{pN} \end{bmatrix} \tag{2.1}$$

where subscript p refers to the p-th training vector. The net-input values of the hidden layer can be calculated as:

$$net_{pj}^{h} = \sum_{i=1}^{n} w_{ji}^{h} x_{pi} + \theta_j^h \tag{2.2}$$

where "h" superscript refers to the hidden layer, w_{ji}^h is the weight on the connection from the i-th input node to the j-th hidden node, θ_j^h is the bias term which provide a fictitous input value of 1 on a connection to bias weight. From the input values, the output of the hidden layer is found to be:

$$i_{pj} = f_k^h(net_{pj}^h) \tag{2.3}$$

where f_k^h is a differentiable activation function. From the hidden layer output, the net-input values of the output layer are:

$$net_{pk}^o = \sum_{j=1}^{L} w_{kj}^o i_{pj} + \theta_k^o \tag{2.4}$$

and the output becomes:

$$O_{pk} = f_k^o(net_{pk}^o) \tag{2.5}$$

where "o" superscript refers to quantities on the output layer.

The activation function is usually selected as a sigmoid function:

$$f_k(net_{pk}) = \frac{1}{1 + e^{-net_{pk}}} \tag{2.6}$$

2. In the error back-propagation, on the other hand, connection weights are updated based on errors between the desired and actual outputs.

First, the error terms in the output layer (using a sigmoidal activation function) are determined:

$$\delta_{pk}^{o} = (y_{pk} - o_{pk}) f_{k}^{o'}(net_{pk}^{o}) \quad (2.7)$$

where y_{pk} is the desired output, and o_{pk} is the actual output and

$$f^{o'} = f_k^o (1 - f_k^o) = o_{pk}(1 - o_{pk})$$

Based on the errors, the weights in the output layer are updated according to:

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \eta \delta_{pk}^{o} i_{pj} + \alpha \Delta w_{kj}^{o}(t-1) \quad (2.8)$$

where η is the learning rate parameter and α is the momentum parameter.

Next, the error terms in the hidden layer are calculated as:

$$\delta_{pj}^{h} = (\Sigma \delta_{pk}^{o} w_{kj}) f_{j}^{h'}(net_{pj}^{h}) \quad (2.9)$$

The weights in the hidden layer are now updated as:

$$w_{ji}^{h}(t+1) = w_{ji}^{h}(t) + \eta\delta_{pj}^{h} X_{i} + \alpha\Delta W_{ji}^{h}(t-1) \tag{2.10}$$

At each iteration, the total network error is measured by the following formula:

$$E = \frac{1}{2}\sum_{k=1}^{M} \delta_{pk}^{2} = \frac{1}{2}\sum_{k=1}^{M} (y_{pk} - o_{pk})^{2} \tag{2.11}$$

where o_{pk} is the actual output by calculation, and y_{pk} is the desired output value. When the network error E is smaller than a pre-specified error level, network training is terminated.

Both feed-forward and error back-propagation calculations are used in the training phase and only the feed-forward is used in the recall phase. Reverse propagation is used to compute the error derivative with respect to all the weights in the network. The error derivative assigned to each weight estimates the effect of each weight on the total error. The back-propagation calculations are applied only during the training phase.

The performance of the back-propagation learning algorithm depends on two performance parameters: learning rate and momentum. Learning rate is used to compute the change in weights from the error derivatives, and is infinitesimally small for a true gradient descent. Momentum, on the other hand, is used to reduce oscillations caused by large values of the learning rate. It modifies the weight changes computed using the current derivative which is proportional to the weight change in the previous iteration. Momentum represents the relative importance of the weight change from the previous iteration.

Although BP has played a large part in the resurgence of interest in artificial neural networks, it still experiences some limitations. One problem with BP is its extremely long - potentially infinite - training times. The fixed structure topology of BP is another problem that needs to be improved. If the pre-designed network structure is not sufficiently large, learning may not converge to acceptable solution for complicated practical problems. If the pre-designed network structure is too large, there will be redundant neurons resulting in more training time for simple practical problems. Therefore, research on the improvement of the BP architecture has become a "hot topic" in the neural networks field.

2.2 ANN structure level adaptation

In recent research, structure level adaptation has played a dominant role. Structure level adaptation enables the network to optimize its structure for each specific problem. Many articles have been published in this area of research [11],[12]-[18], among them the following articles reviewed below are particularly interesting.

2.2.1 Lee's approach

Tsh-Chang Lee [11] proposed a general procedure for structure level adaptation for multi-layer feed-forward networks. This structure level adaptation of the network can be achieved through adjusting the number of neurons in the hidden layers. This can be broken down into two steps: neuron generation and neuron annihilation. Lee [11] defined a neuron generation rule as follows:

(a) Neuron i should be split into two neurons when

$$\varepsilon_i = \frac{\partial \varepsilon}{\partial WD_i} \times WD_i > \Theta_G \qquad (2.12)$$

where

$$WD_i[n] = \gamma_w WD_i[n-1] + (1-\gamma_w) Met(\vec{w}_i[n], \vec{w}_i[n-1]) \qquad (2.13)$$

and

ε is the overall system error (or the cost function depending on each specific neural network model)

WD_i is the walking distance for neuron i,

ε_i represents the contribution of neuron i to the overall system error owing to its parameter fluctuation

Θ_G is the threshold value

γ_w is a constant factor, $0 << \gamma_w < 1$

Met is some metric that measures the distance between vectors in a metric space.

(b) When neuron generation is required, the neuron with the highest $FD^{(l)}_i$ which is defined as the Fluctuated Distortion Measure, is split into two neurons. The fluctuated distortion measure is:

$$FD^{(l)}_i[n] = \left|\delta^{(l)}_i[n]\right| \left\|Q^{(l-1)}[n]\right\| WD^{(l)}_i[n] \tag{2.14}$$

where:

$\delta^{(l)}_i$ is the pre-sigmoidal error for the *i*-th neuron in layer l.

$Q^{(l-1)}$ is the receptive field for the neurons in layer l.

$WD^{(l)}_i$ is the walking distance for neuron i in layer l.

The neuron i should be annihilated according to Lee [7], if the following inequality is satisfied:

$$Act_i < \frac{\varepsilon}{M} \tag{2.15}$$

(2.16)

$$Act_i[n] = \gamma_\alpha Act_i[n-1] + (1-\gamma_\alpha)y_i(n)$$

where:

ε is the overall system error

M is the number of neurons in the network

Act is the average output activity for neuron i

γ_α is a constant factor, $0<\gamma_\alpha<1$

$y_i(n)$ is the output level for neuron i

A simple example was tested to demonstrate these ideas in [11]. The test problem is a two class classification problem with 2-D input feature vectors. The results show the proposed network can find the correct decision boundaries for pattern classification.

2.2.2 Other approaches

Azimi-Sadjadi, et. al [12], proposed a new approach for dynamic node creation in multi-layer neural networks. This method uses time and order update formulations in the orthogonal projection algorithm to arrive at a recursive weight updating procedure for training processes. The algorithm may be summarized as follows:

1. Limited Architecture: Construct a neural networks architecture with a small number of hidden layer nodes
2. Time Update - Weight Adaptation: Present the training data sequentially and iterate the standard RLS and update the weights using the order update equations in conjunction with the analog of the back-propagation method. Monitor the Average Mean Squared Error (AMSE) at the output.
3. Order Update - Node Creation: If the rate of change of AMSE is not acceptable, increase the hidden layer nodes by one and update the weights using the order

update equations.

4. Order - Time Updates Interface: To proceed with subsequent weight updating after node creation use a proposed equation and then switch back to the time updating process.

Azimi-Sadjadi, et. al [12] applied this network to microwave data, namely of Nylon and Wood composition, for detection and classification. Improved results were obtained compared to the BP algorithm

.

Another very interesting approach referred to as Cascade - Correlation learning architecture was proposed by Fahlman and Lebiere [13]. Cascade - Correlation begins with a minimum network, then automatically trains and adds new hidden units one by one. Each layer consists of only one unit, thus creating a multi-layer structure. A diagram is shown in Figure 2.2

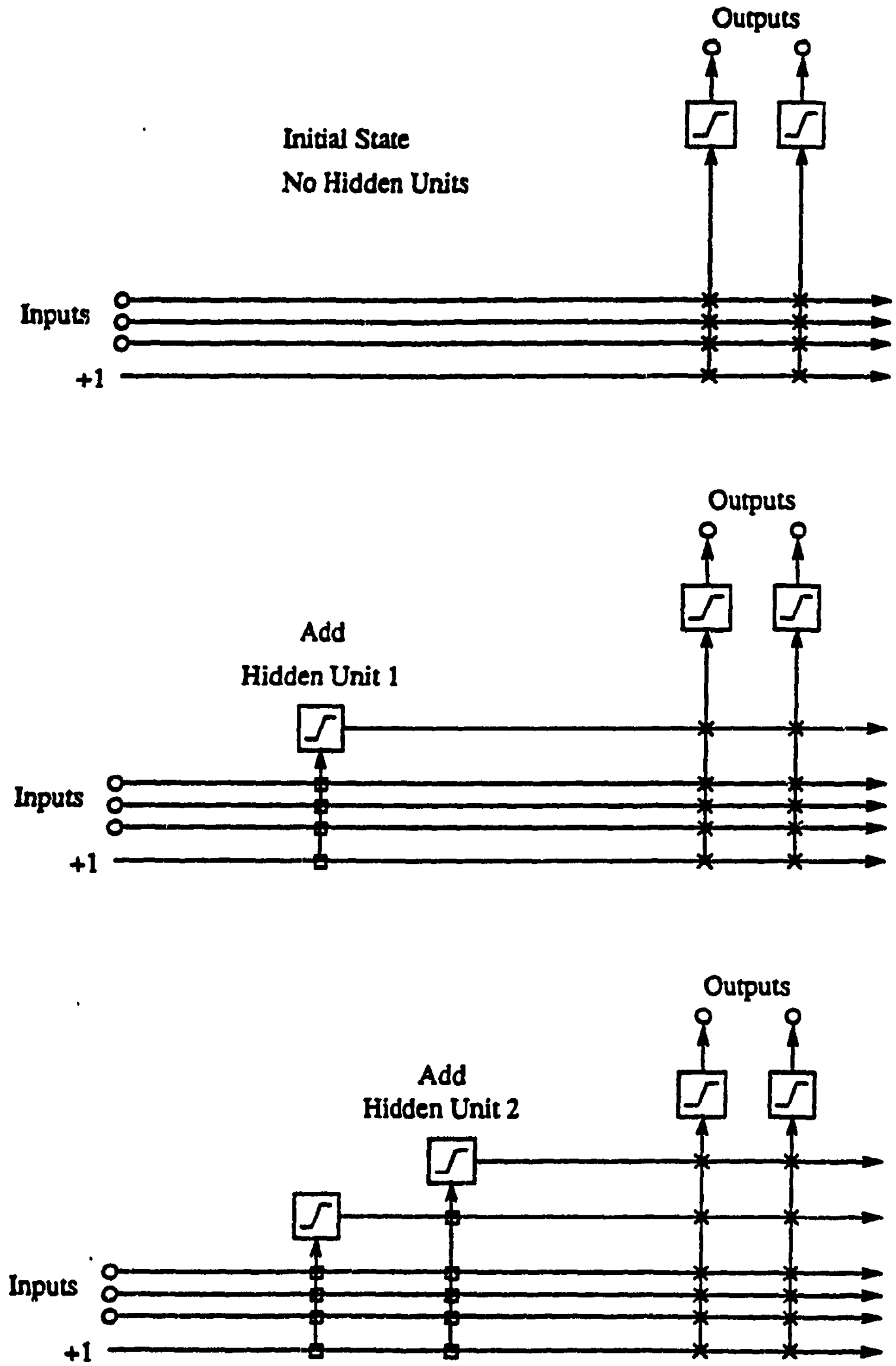

Figure 2.2: The Cascade- Correlation architecture, initial state and after adding two hidden units. The vertical lines sum all incoming activation. Boxed connection are frozen, X connections are trained repeatedly.

Cascade - Correlation also uses the quick-prop algorithm [18] instead of simple, linear gradient descent to update the weights in the network. The Cascade- Correlation architecture was tested on N - input parity, Sonar and Spiral problems and resulted in a tremendous speed up of network convergence.

2.3 The proposed adaptive structure neural networks (ASNN)

2.3.1 Description

In general, neural networks adaptation can be classified into three different levels: functional, parametric and structural levels. Function level adaptation can change the output activities in the neural network according to the input signals. Parameter level adaptation can adapt the weights in the neural networks by changing the learning rate, momentum, etc. specifications. Structure level adaptation can adapt the network structure according to the statistics of the variables in the function and parameter levels. In particular, structure level adaptation allows the network to change the inter-connections between neurons, generate neurons, and annihilate existing neurons. Hence, the research effort will focus on neuron generation techniques for structure level adaptation.

2.3.2 Neuron generation rule

The neuron generation algorithm is based on the principle that if the neural network does not contain enough neurons to learn the specific mapping problem, then the weights and the output of existing neurons will tend to fluctuate and never converge. Network system error is an important criterion in order to determine neuron generation rule. This is the case in the work of Lee [11], Azimi - Sadjadi [12], and Fahlman [13] as well as in this thesis. New neurons should be generated when no significant system error reduction has occurred after a certain number of training cycles. To determine the neuron generation rule, Lee [11] proposed the Walking Distance method. M. R. Azimi - Sadjadi, et. al [12] presented the Recursive Least Squares method. In this thesis, the desired position of the neuron to be split is defined by determining the neuron (1) whose input weight has the biggest fluctuation compared with other neurons in the same hidden layer and (2)whose output fluctuation in the hidden layer is also high. To determine the neuron with the highest fluctuation rate, the current and the most recent iterations are considered because they provide the most crucial historical information. This information includes update of weights in both the output layer and the hidden layer. The formula for the neuron with the Highest Fluctuation Rate $HFR^{(l)}_i$ is as follows:

$$HFR^{(l)}_i = MAX(\,\Delta W^{(l)}_i * \Delta OUT^{(l)}_i\,) \tag{2.17}$$

$$\Delta W^{(l)} = \left\| W_i^{(l)}(n) - W_i^{(l)}(n-1) \right\| \tag{2.18}$$

$$\Delta OUT^{(l)}_i = |\, OUT^{(l)}_i(n) - OUT^{(l)}_i(n-1)\,| \tag{2.19}$$

where:

$HFR^{(l)}_i$ is Highest Fluctuation Rate for the *i*-th neuron in the l-th layer

$\Delta W^{(l)}_i$ is the weight difference for the i-th neuron in the l-th layer

$\Delta OUT^{(l)}_i$ is the output difference for the *i*-th neuron in the l-th layer

2.3.3 Maximum-minimum rule (MMR)

In the dynamic neural networks structure adaptation, researchers focus on splitting one "mother" neuron into two new neurons each time. In other words, only one new neuron is generated each time. This is widely used in the published articles [12],[18]-[21] for the improvement of the network's performance. However, the method of spliting one " mother " neuron each time may still require a long time for the network to converge to the desired error level when applied to most practical problems.

It has been determined that network convergence could be further enhanced by splitting more than one "mother" neuron at a time, particularly during the beginning of the network training period. This improvement will add great benefits for solving complex problems. The key point is to specify the appropriate number of neurons to be split at the right time and place during the network structure adaptation process which is shown in Figure 2.3. The proposed algorithm, designated as Maximum - Minimum Rule (MMR), is explained below.

Consider a set of neurons in the hidden layer. The algorithm is then carried out as follows:

1. Calculate the fluctuation rate ($FR_i^{(l)}$, i = 1,...N_h) for this set of neurons and queue them from high to low
2. Split the neuron with the highest fluctuation rate ($HFR^{(l)}$)
3. For the remaining neurons in the set, check the variation fluctuation:

$$1 - \frac{FR^{(l)}_i}{HFR^{(l)}} < FRTH \qquad (i = 2, \ldots, N_h) \qquad (2.20)$$

where N_h is the number of neurons in the hidden layer set and

$FRTH$ is the threshold of the fluctuation rate

The above formula computes the minimum difference of the fluctuation rate between the *i*-th neuron and the neuron with the highest fluctuation rate. If the inequality is satisfied for the *i*-th neuron, then this "mother" neuron is also split to generate a new neuron. Whenever neurons are split, the weights of the new neurons are the same as the weights of the mother neuron. The threshold *FRTH* is adjusted according to the specifics of the application problem as applied to the ASNN.

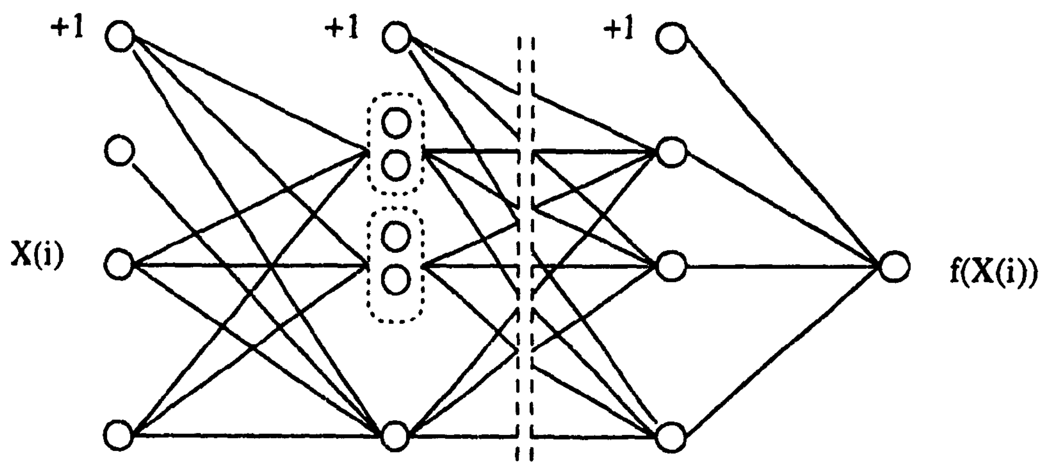

Figure 2.3: Three layer ASNN with two neurons split

2.3.4 Fast learning algorithms

Back-propagation is based on gradient descent on the error surface. In general, it can get stuck in local minimum on the error surface and can fail to discover the best set of weights. To develop a complete methodology for generating neurons in structure level adaptation, a new neuron generation rule was defined in Section 2.4.2 and a new MMR algorithm was proposed in Section 2.4.3. In this section, additional research on speeding up the network convergence will be discussed.

2.3.4.1 The existing approaches

To accelerate the convergence rate of the BP network, different approaches have been developed in recent years. The research can be categorized into the following areas:

1. Initial weights improvement

Back-propagation and its applications usually initiate training procedures by randomizing a set of initial weights within small range of real values. These random initial weights will cause different local minima during network training process. Therefore the solution for a specific problem depends on the set of initial weights. If the initial weights are set very close to network global minimum, then the network training procedure will converge quickly. Hasanat and Eduardo [22] proposed an extrapolative methods to predict the weights in the BP network. Chih-Liang and Roys [23] developed a Forward Estimation Algorithm and Recurrent Estimation Algorithm for the optimal estimation of initial weights. Kim and Ra [24] suggested a method for using the minimum bound of weights based on dynamics of decision boundaries; which is derived from the generalized delta rule. Simulation results showed that all the above research have accelerated the convergence rate of the BP network.

2. Modification of the gradient and momentum terms

Yoshio and Alex [25] proved that a modified gradient and momentum method can speed up the network convergence. In this paper, the authors used the simple gradient method for updating weights. The convergence time is T1 = C1 / D, where D is the sharpness of sigmoid function and C1 is a constant. When the momentum method was used for updating weights, the converge time is $T_2 = C_2/(\sqrt{D})$, where C_2 is a constant and much smaller than C_1. Both methods have obtained similar improved results to cut down the network training time.

3. New algorithms

Brent [26] published a Fast Training Algorithm (FTA) for multilayer neural network. The key point of FTA is to construct a decision tree and then simulate the decision tree with a neural network (Figure 2.4)

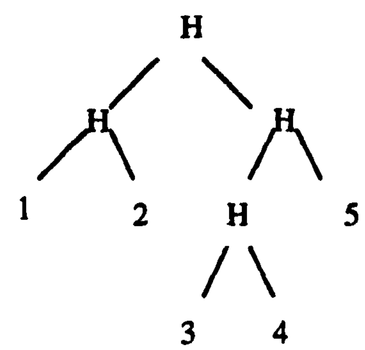

Figure 2.4: A corresponding decision tree for 5 different regions
(H refers to hyperplane)

This new algorithm has been implemented and tested on problems such as the parity problem and speech recognition. Experimental results showed that the network training is much faster than the BP and the accuracy achieved is as good as or better than BP.

4. Other attempts

Some other published methods are also very interesting for the speed up of the BP network convergence. For example, Masafumi [27] proposed an accelerated BP by using unlearning based on Hebb Rule. Wong [28] presented a selective training algorithm for fast error propagation. All these approaches have reported a better results compared to the traditional BP algorithm.

2.3.4.2 The proposed approach using Delta Adaptation (DA)

In this thesis, two different approaches similar to the conventional DA algorithm are explored ,namely, DA vector summation and a DA modification.

(i) DA vector summation

In BP, the Generalized Delta Rule (GDR) plays an important rule to speed up the network convergence. The GDR is used so that the weight is changed by an amount proportional to the product of an error signal δ. Figure. 2.5 is a simple diagram illustrating this property.

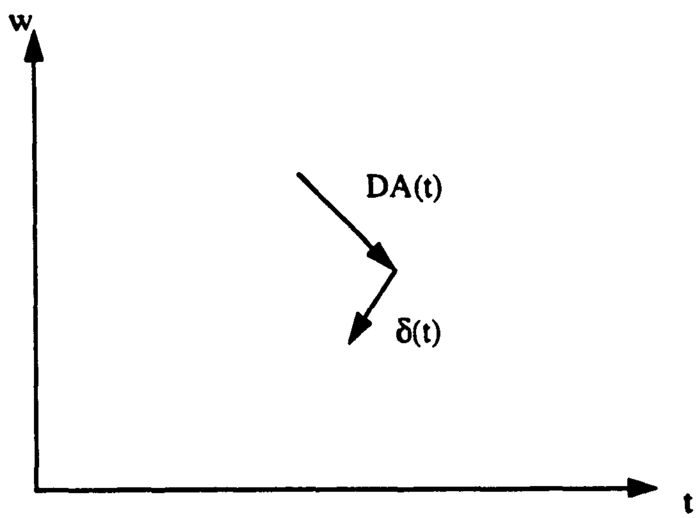

Figure 2.5: The BP generalized delta rule

From Figure 2.5, the GDR is applied to weight update by using a gradient descent method to find the steepest direction and the step size (DA(t)). δ(t) is the error calculated proportional to the applied product.

In an attempt to improve the speed of network convergence, the DA vector summation is expressed as follows:

$$DA(t+1) = \lambda \times \sqrt{DA(t) \times DA(t) + \delta \times \delta - 2 \times DA(t) \times \delta \times \cos(\theta)} \qquad (2.21)$$

where λ is the DA factor and θ is the angle between DA(t) and δ(t).

The resulting graph illustrating the above is shown in Figure. 2.6.

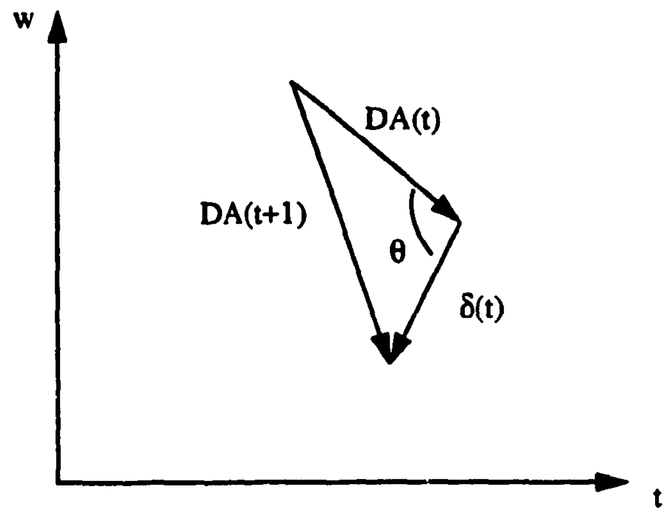

Fig. 2.6 The DA vector summation

Theoretically, the DA vector summation is an ideal approach because an accurate measurement can be performed to determine the DA step size. It is used for the updating weights both in the hidden layer and in the output layer. The sum of any two sides of a triangle is greater than the length of its third side. So if DA(t+1) replaces DA(t) in the conventional method, the speed of network convergence will be improved. In practice, however, determining the angle θ is not trivial, which motivates the next scheme.

(ii) Proposed DA Modification

In the BP algorithm, the GDR is applied to the network learning process. An error term δ, which represents the difference between the desired output and the actual output is calculated using the gradient method. The error δ is then transmitted backwards from the output layer to the hidden layer for weight update in both layers.

For the k-th output node:

$$\delta_{pk}^{o} = (y_{pk} - O_{pk}) f_{k}^{o'}(net_{pk}^{o}) \tag{2.22}$$

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \eta\delta_{pk}^{o} i_{pj} + \alpha\Delta w_{kj}^{o}(t-1) \tag{2.23}$$

For the j-th hidden node:

$$\delta_{pj}^{h} = (\Sigma\delta_{pk}^{o} w_{kj}) f_{j}^{h'}(net_{pj}^{h}) \tag{2.24}$$

$$w_{ji}^{h}(t+1) = w_{ji}^{h}(t) + \eta\delta_{pj}^{h} X_{i} + \alpha\Delta W_{ji}^{h}(t-1) \tag{2.25}$$

The objective of the network learning is to eliminate the difference between the desired output and the actual output. The learning rate parameter η and the momentum parameter α are used to improve the network performance. It is often possible to increase the size of η as learning proceeds. When calculating the weight change, Δw, the momentum σ tends to keep the change of weights follow the same direction.

In many cases, the error term may be small after some iterations, but the network has not reached the desired error level, even when α and η are optimally adjusted. In this case, the performance of the network training will slow down, or fail to reach an optimal solution. One possible reason is due to the non-linear sigmoidal function shown in Figure 2.7.

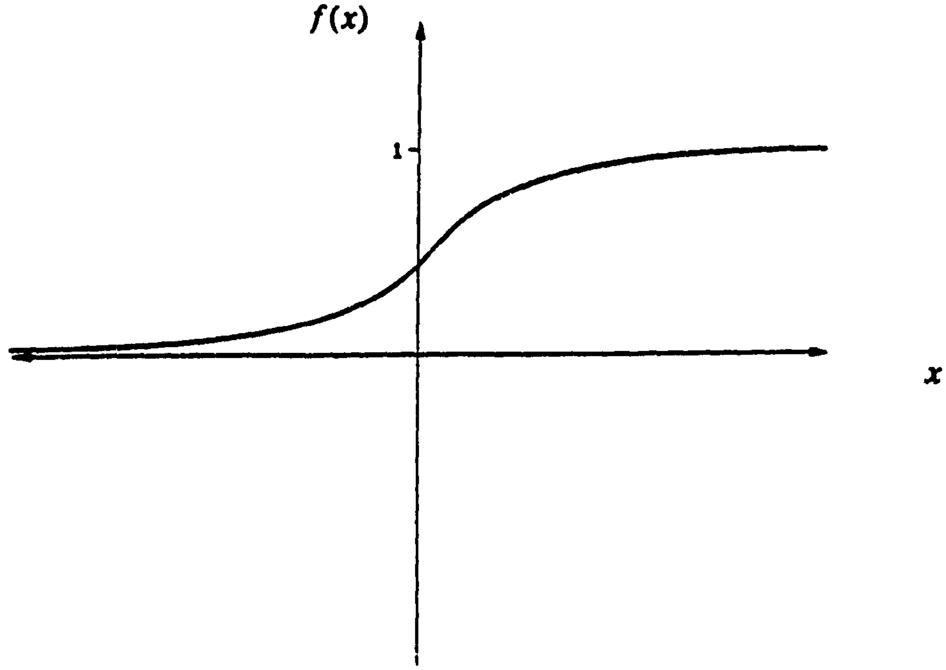

Figure 2.7: This graph shows the S shape characterstic of the sigmoidal function

It is evident that the S shaped curve of sigmoidal function flattens as its inputs increase or decrease. If the error term is calculated within the flat regions, the value will be very small. In order to overcome the shortcoming of the sigmoidal function, it is necessary to adjust the error term to avoid reaching the flat regions. The improved DA algorithm is defined as follows.

When k is an output node:

$$\delta^{o}_{pk} = (y_{pk} - O_{pk}) f^{o'}_{k}(net^{o}_{pk}) \tag{2.26}$$

$$DA^{o}_{pk}(t+1) = \lambda DA^{o}_{pk}(t) + \gamma\delta^{o}_{pk} + \sigma\Delta DA^{o}_{pk}(t-1) \tag{2.27}$$

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta DA^{o}_{pk}(t+1) + \alpha\Delta w^{o}_{kj}(t-1) \tag{2.28}$$

When j is a hidden node:

$$\delta_{pj}^{h} = (\Sigma\delta_{pk}^{o} w_{kj}) f_{j}^{h'}(net_{pj}^{h}) \qquad (2.29)$$

$$DA_{pj}^{o}(n+1) = \lambda DA_{pj}^{h}(n) + \gamma\delta_{pj}^{h} + \sigma\Delta DA_{pj}^{h}(n-1) \qquad (2.30)$$

$$w_{ji}^{h}(n+1) = w_{ji}^{h}(n) + \eta DA_{pj}^{h}(n+1) + \alpha\Delta w_{ji}^{h}(n-1) \qquad (2.31)$$

Where:

- p : the number of patterns
- λ: DA factor
- γ: δ step size
- σ: DA momentum
- $DA^{o}_{pk}(n+1)$: the Delta value at time n+1 for pattern p belonging to node k of the output layer
- $DA^{h}_{pj}(n+1)$: the Delta value at time n+1 for pattern p belonging to node j of the hidden layer

Equations (2.27) and (2.30) are different from those in the BP algorithm. The motivation for these two equations come from the principle of weights update. γ is the step size of δ and σ is the momentum of DA. Momentum tends to keep the weight changes in the same direction. Equations (2.27) and (2.30) are modified weight update formulae using the DA variable to replace the δ variable. One possible plot for δ and DA is drawn in Figure 2.8.

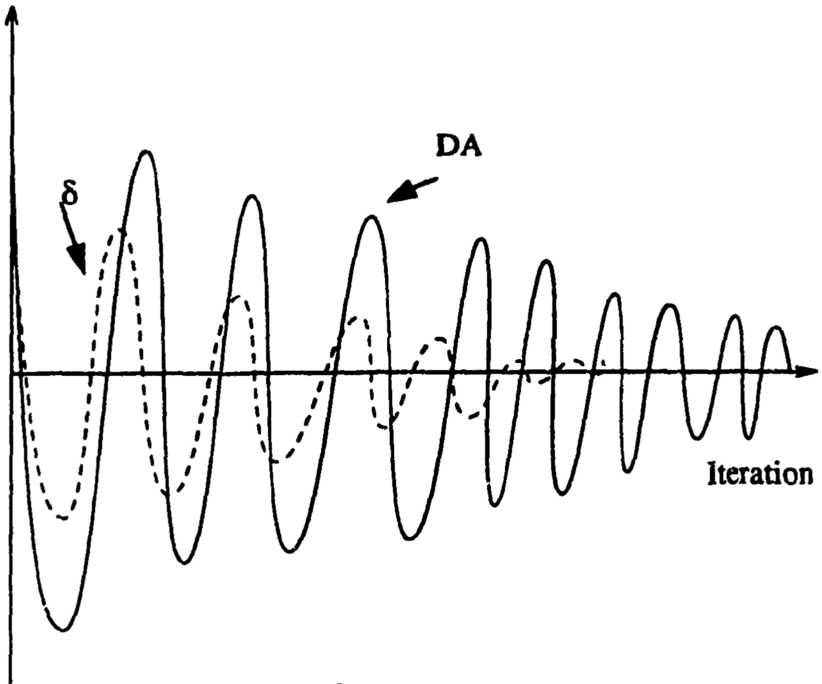

Figure 2.8 : Typical δ and DA process plots

Several issues related to the modified DA are explained below:

(A) The minimization of the cost function

From Figure 2.8, we can clearly see that the difference between the δ and the DA curves is the amplitude. Both the δ and the DA curves have similar decaying features. This means that the DA will tend to eliminate the error term. Because the DA starts with higher error terms compared to δ, it is reasonable to state that DA will have faster convergence speed for reaching to the desired error level before the value of DA falls within the flat regions of the sigmoid function (that is, the DA value falls within the flat regions of sigmoidal function after the δ value).

(B) The convergence properties of the weight vector

The modified weight update equations for both the output and the hidden layers are given by equations (2.27) and (2.30). The difference between the BP weight update and the ASNN weight update is that δ in BP is replaced by DA in ASNN. Large DA values may speed up the network performance when λ, γ and σ parameter are properly selected.

(C) λ, γ and σ parameters

The physical meaning of the parameters γ and σ in ASNN is similar to the parameters η and α in BP. Parameter λ is defined as the DA factor. It represents the strength of DA in the current iteration. The selection of the parameters varies according to application problems.

The value of λ is usually selected between 0.1 to 0.4. The small λ is used to adjust the DA value for its last iteration. If the value of λ is high, the system error may fluctuate and fail to converge. A proper value for δ step size γ can speed up the neural networks convergence. Step size γ is used to compute the change of DA. The range for its value has been found experimentally to be between 0.5 to 1.0. In order to reduce the oscillations, the DA momentum σ is used to control step size γ. It modifies the δ changes and represents the relative importance of δ changes in the previous iteration. The range of σ is set between 0.2 to 0.8.

2.4 Summary

Back-Propagation is the most widely used algorithm developed in neural networks as used in practical applications. It is composed of two algorithms: feed-forward calculation and error back-propagation. Error back-propagation enables the neural networks to learn input-output properties of the data. But the fixed structure of the Back- Propagation network limits its functions. To overcome this limitation, different approaches were tested to find a suitable way for structure level adaptation. Lee [11]. Azimi-Sadjadi [12], Fahlman , et al [13], have developed algorithms for structure level adaptation and their results are encouraging

In this thesis, a new ASNN is proposed. This new algorithm deals with only a neuron generation rule. The Maximum-Minimum Rule (MMR) is developed for spliting more "mother" neurons at a time. MMR has been verified to be an effective improvement over the single "mother" neuron generation method. The modification of the traditional DA method also enhanced the network convergence learning rate. The ASNN will be tested using the benchmark examples in Chapter 3. A more realistic application, namely, an automatic EEG seizure detections will be introduced in Chapter 4.

CHAPTER 3

Simulation Results

3.1 Introduction

As pointed out in Chapter 1, neural networks is a powerful tool for practical applications such as pattern recognition[5], image processing [6], control [13], etc. In Chapter 2, a novel approach for adapting the structure of the network, namely adaptive structure neural network [30]-[32], was proposed. This approach is concerned with mainly neuron generation. A neuron generation rule was defined to determine when and where the new neurons should be added in the hidden layer. To confirm the effectiveness of the proposed algorithm, simulation results on both discrete and analog examples are presented in this chapter. The two discrete examples include character recognition and XOR problem. Character recognition is often used as a test for problems in pattern recognition. XOR is a classic example which is widely used in neural networks research. The two analog functions are 4-leaf-rose and one-spiral. The one spiral problem is also a typical example used by many researchers [13] to test the network performance.

3.2 Architecture of the proposed neural networks

A three layer neural networks is initialized with four neurons in the hidden layer to test the proposed structure adaptation algorithm and to compare it with the standard back-propagation [1] neural networks as well as with Lee's method [11]. The parameters selected for the three networks are:

learning rate = 0.15

momentum = 0.9

initial weight = random

and for the ASNN, the additional parameters are selected as follows:

FRTH = 0.2

$\lambda = 0.2$

$\gamma = 0.8$

$\sigma = 0.2$

3.3 Discrete results

3.3.1 Character recognition problem

Character recognition is widely used as a test for neural networks simulation. Three patterns, "A" ,"C", and "D", are selected for the testing. Each pattern is represented by a 5*7 matrix. The input and output layers consist of 35 neurons. The hidden layer is initialized to 4 neurons. Table 3-1 shows the results using the BP, Lee's and the proposed ASNN algorithms.

Table 3-1: Character simulation results

Desired Error	Pattern	BP	Lee's method		ASNN			
		Iteration	Iteration	Added Neuron	Iteration	Added Neuron	Improved rate (compared to BP)	Improved rate (compared to Lee)
0.0001	A	322	140	5	47	4	85%	66.7%
	C	200	143	5	40	7	80%	72.0%
	D	317	141	5	100	5	67%	29.1%
0.00001	A	6258	2613	8	2042	8	67%	21.9%
	C	6041	2793	8	1375	14	76.8%	50.8%
	D	6106	2658	8	2032	9	66.7%	23.6%

Clearly, ASNN obtained the best results among the three networks. Overall, between 66.7% to 85% of training iterations are saved by using the ASNN method compared to the BP method and between 21.9% to 72.0% of training iterations are saved by using the ASNN method compared to Lee's method. Figures 3.1 to 3.6 show the learning error plots for each of the characters.

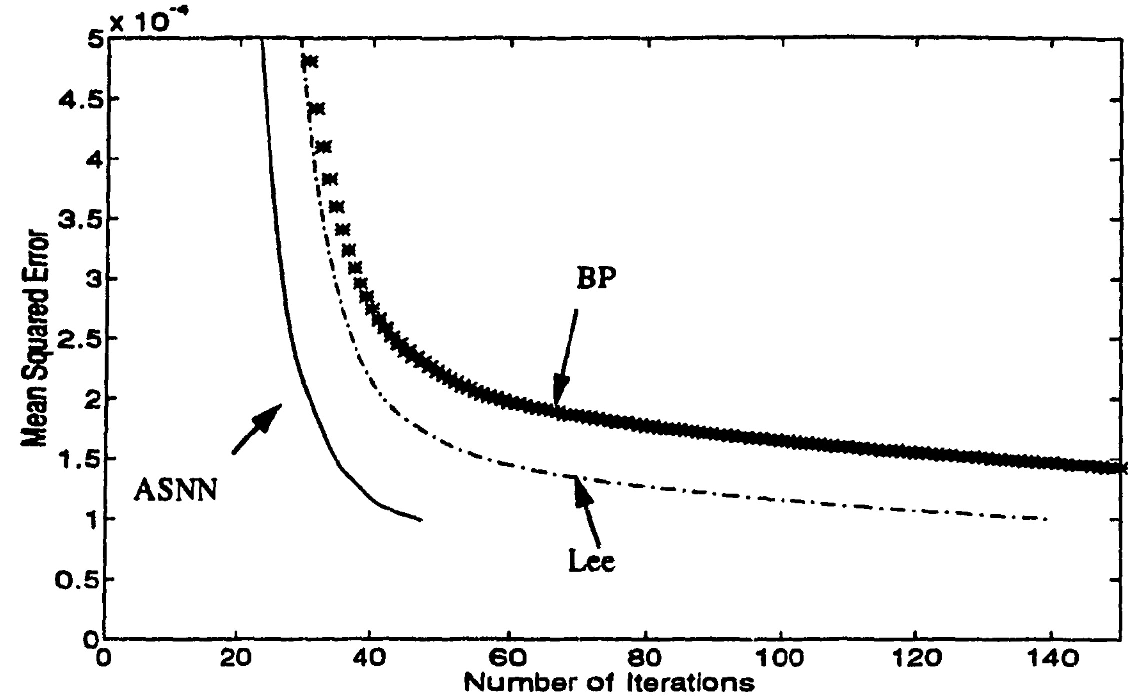

Figure 3.1: The learning curve for character "A" with desired error=0.0001

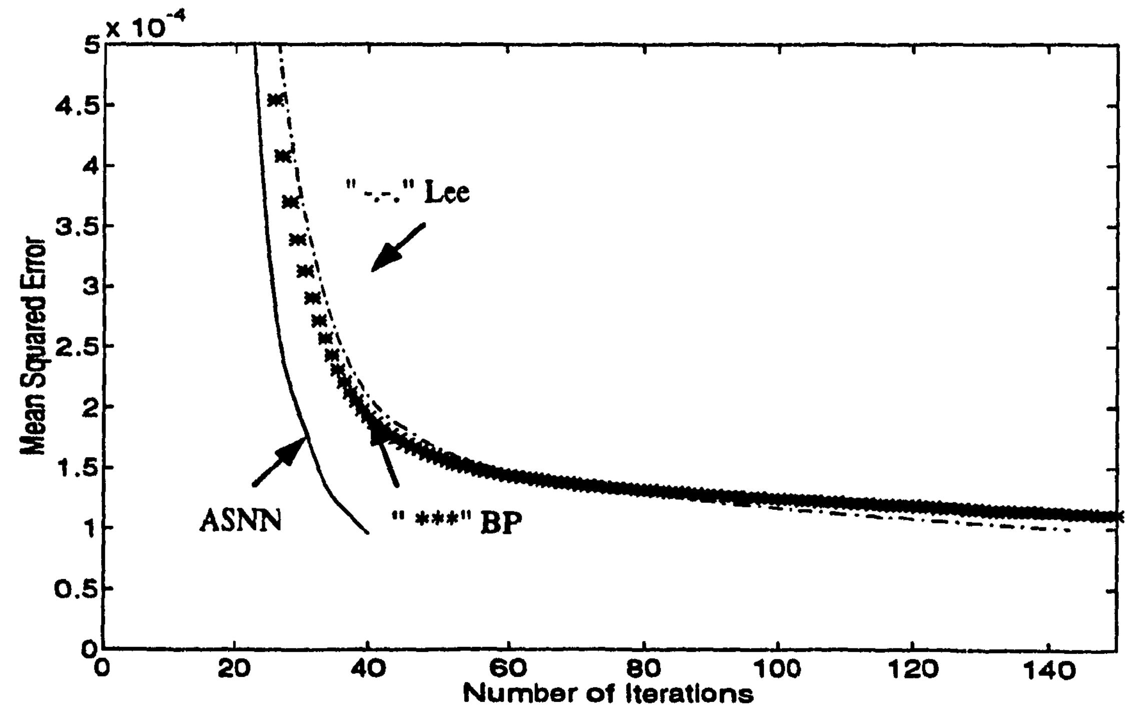

Figure 3.2 : The learning curve for character " C" with desired error = 0.0001

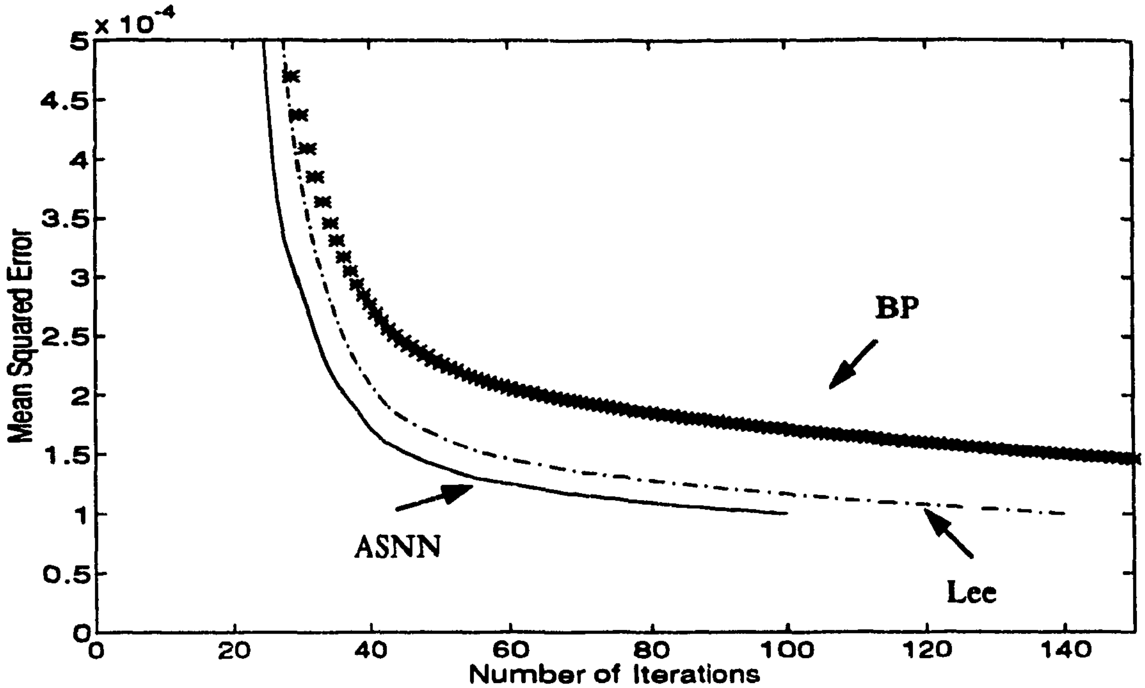

Figure 3.3: The learning curve for character " D " with desired error = 0.0001

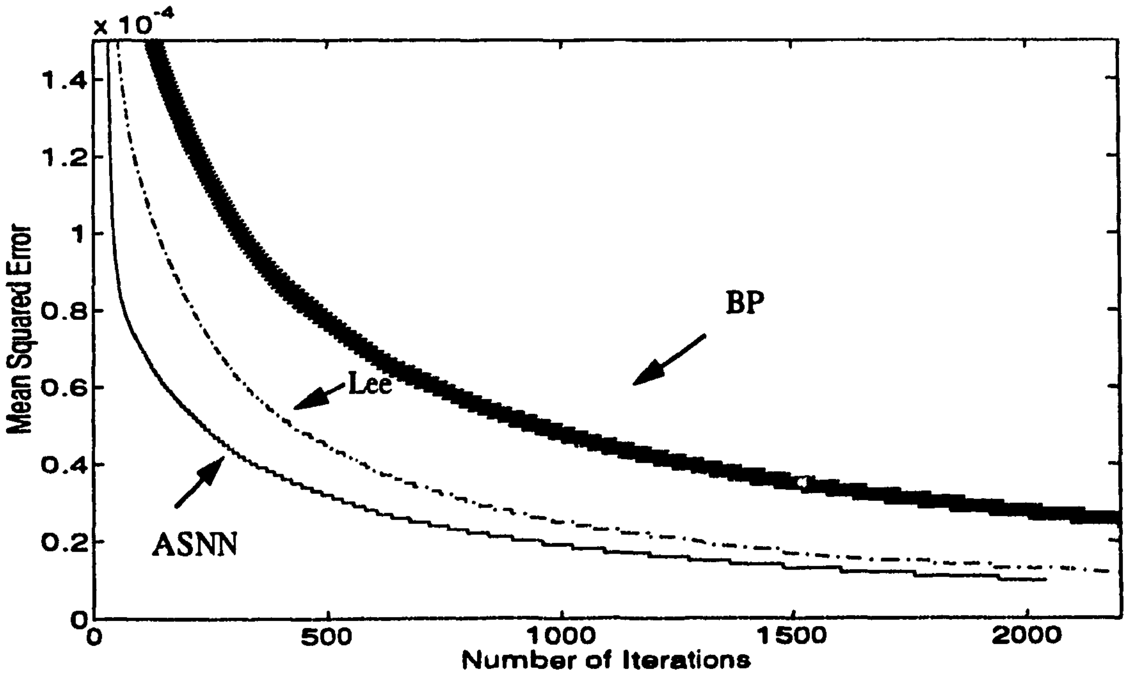

Figure 3.4: The learning curve for character " A " with desired error = 0.00001

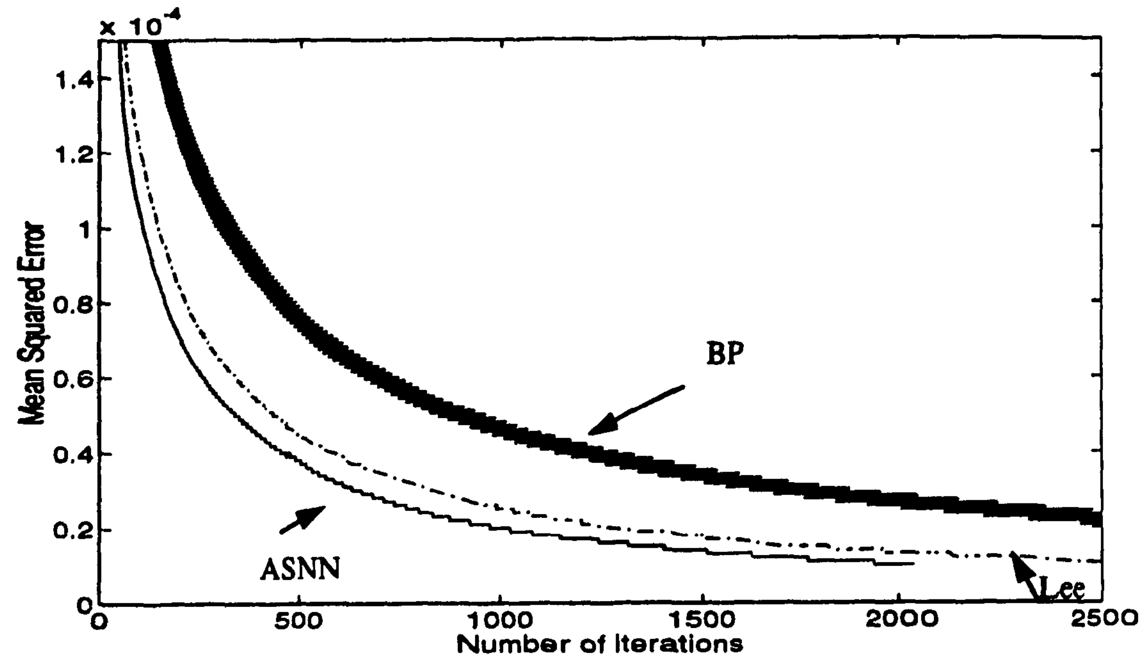

Figure 3.5: The learning curve for character "C" with desired error = 0.00001

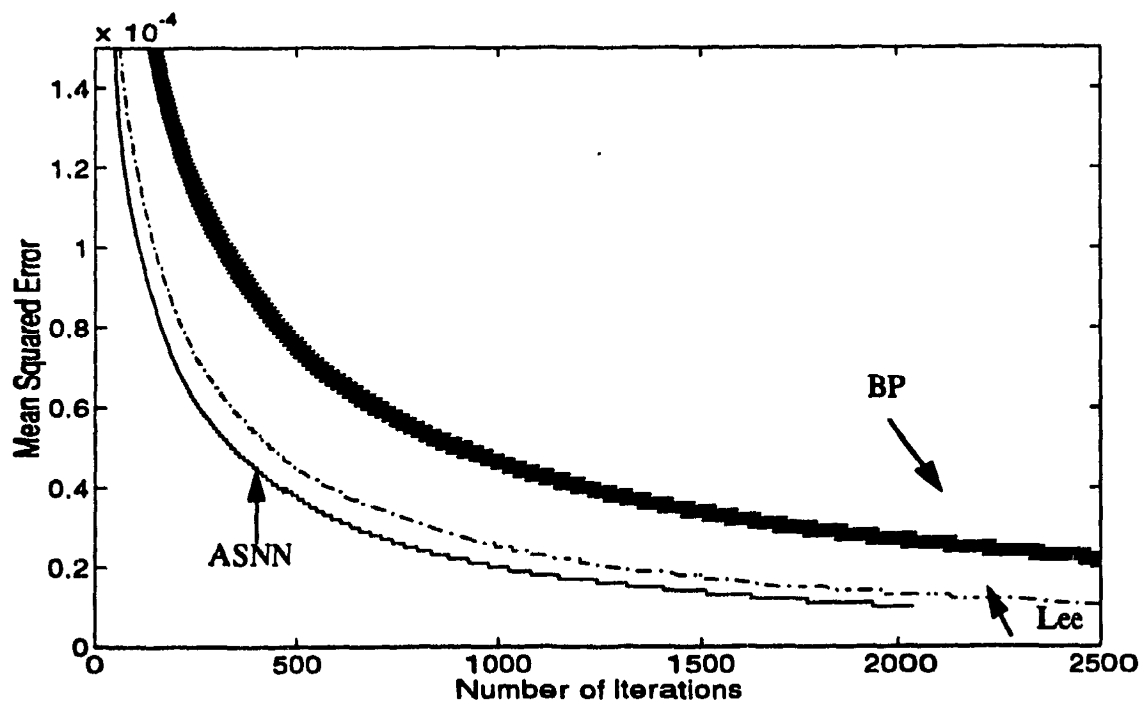

Figure 3.6: The learning curve for character " D " with desired error = 0.00001

From the simulation results, one may observe that for the same value of the desired error the BP or Lee's method required similar number of iterations for "A", "C" and "D" characters, but Lee's method obtained a faster rate of convergence. This is due to the fact that BP is a fixed structure network while Lee's structure level adaptation splits one "mother" neuron each time. With the same value of desired error, the iteration numbers varied when different characters were applied to ASNN. For example, to reach the desired error of 0.0001, 100 iterations were required when character "D" was applied to ASNN , while only 40 iterations were required when character "C" was applied. The reason for this substantial reduction is that more "mother" neurons were split at a time to speed up the network convergence when character "C" was trained with the ASNN. From the relationship between the number of iterations and the neuron generation (shown in Figures 3.7-3.9), for character "C" it is evident that two "mother" neurons were split at iterations 14, 21 and 35. On the other hand, only one "mother" neuron was split each time when character "D" was applied to ASNN. Therefore, ASNN is more effective and flexible in dealing with different patterns.

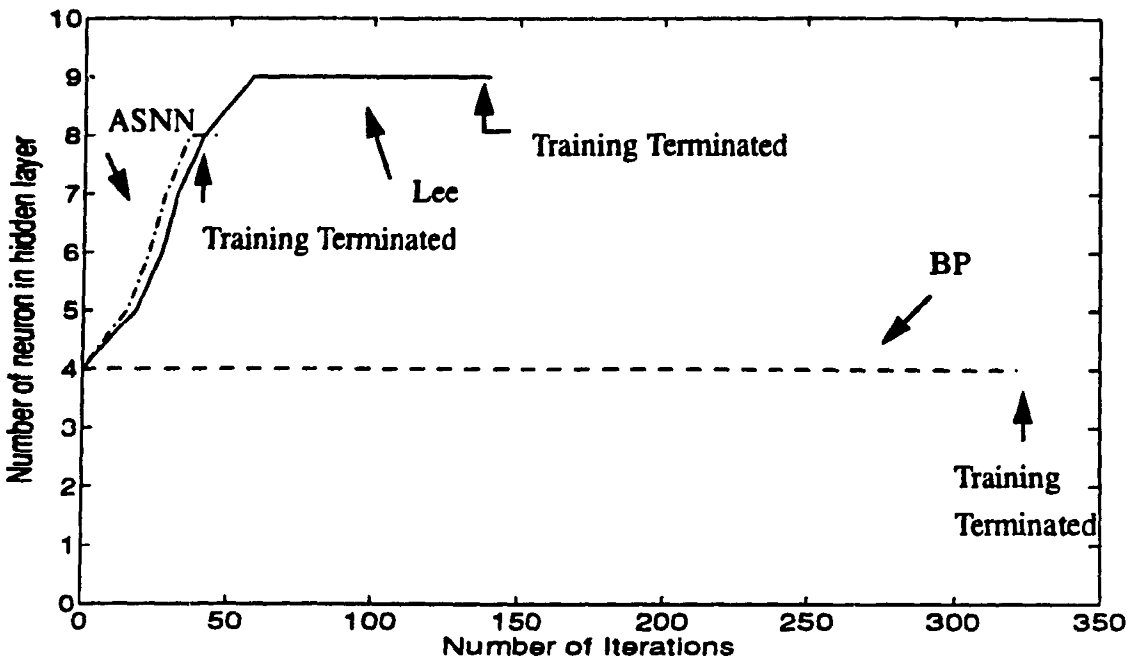

Figure 3.7: The relationship between the number of iterations and the neuron generation (character "A" with desired error = 0.0001)

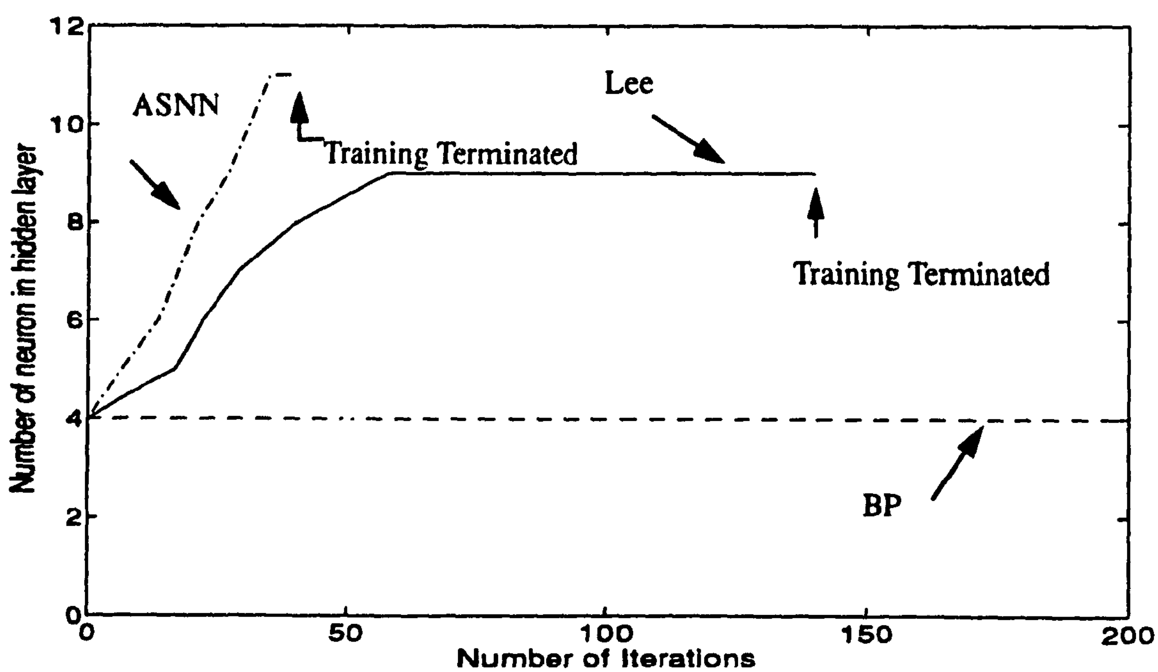

Figure 3.8: The relationship between the number of iteration and the neuron generation (character "C" with desired error = 0.0001)

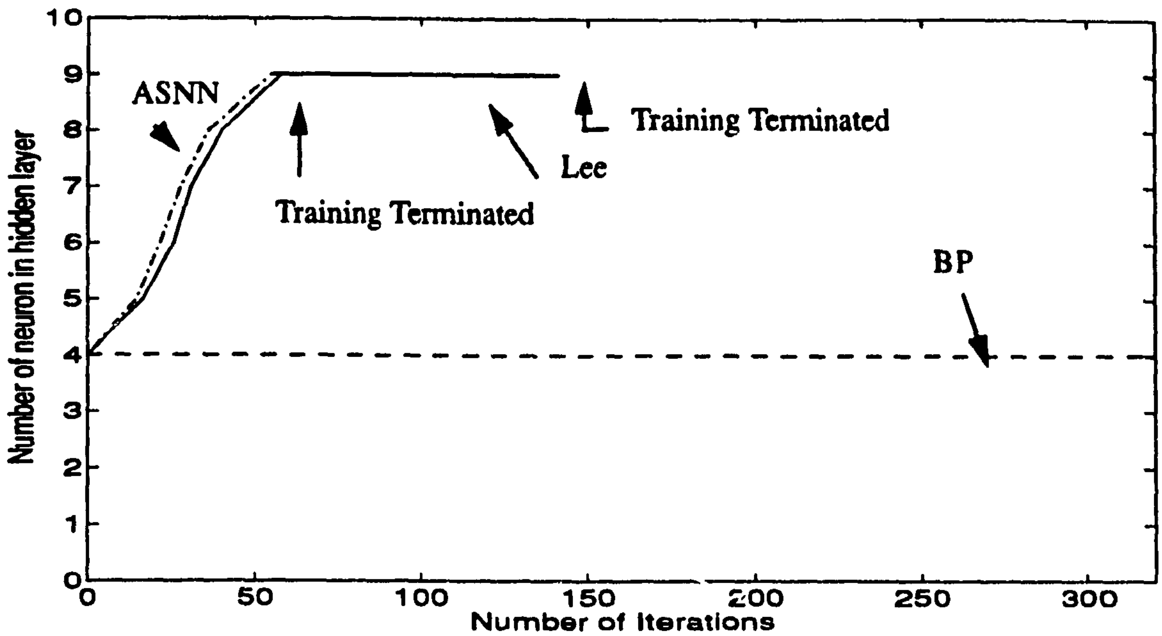

Figure 3.9: The relationship between the number of iteation and the neuron generation (character "D" with desired error = 0.0001)

By comparing the number of new neurons in Lee's algorithm with those in ASNN, it is verified that, for the same value of the desired error the same number of new neurons (see Table 3-1) was generated by using Lee's algorithm, whereas different number of new neurons were created when these characters were trained by ASNN. The neuron splitting method proposed in ASNN is capable of estimating a more appropriate number of neurons for spliting during the network training process and thus can speed up the network convergence.

When new neurons were added to the neural networks, the system error will increase instantaneously because the network structure has been perturbed. This phenomenon is reflected in Figures 3.10 - 3.12. For example, when character "A" is applied to ASNN, a new neuron was added at iteration 16. As shown a sharp peak has appeared on the curve which is then quickly reduced after 4 iterations. The same phenomenon has also occurred in Lee's method, but at iteration 19, whereas BP did not have such a peak because its structure dose not change.

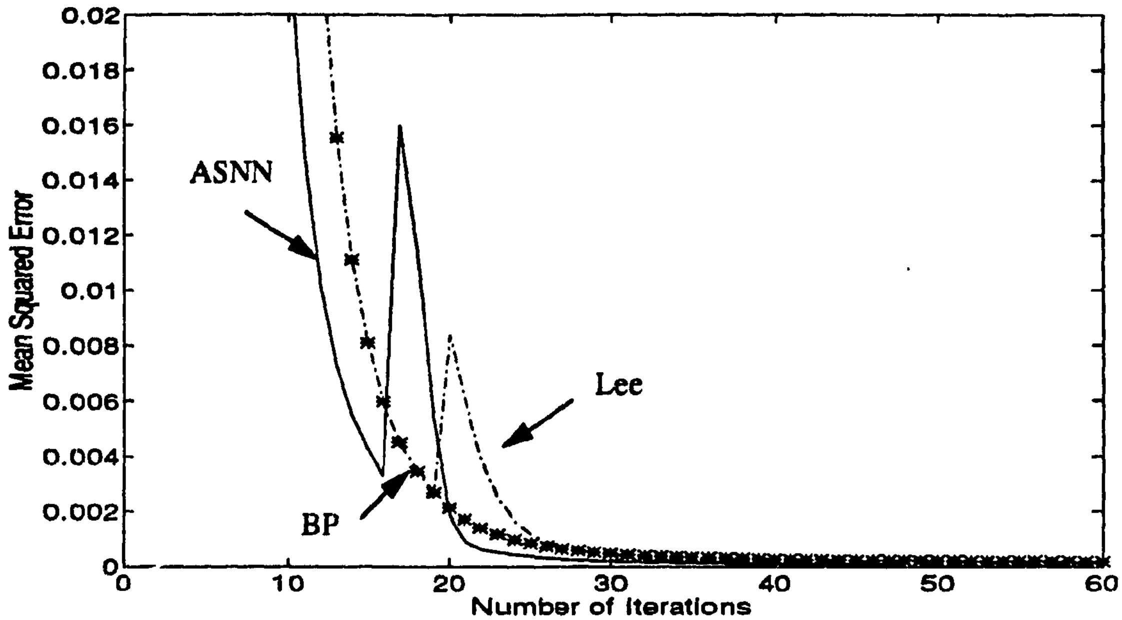

Figure 3.10: The local behavior for learning during the first 60 epochs

(charac..r "A" with desired error =0.0001)

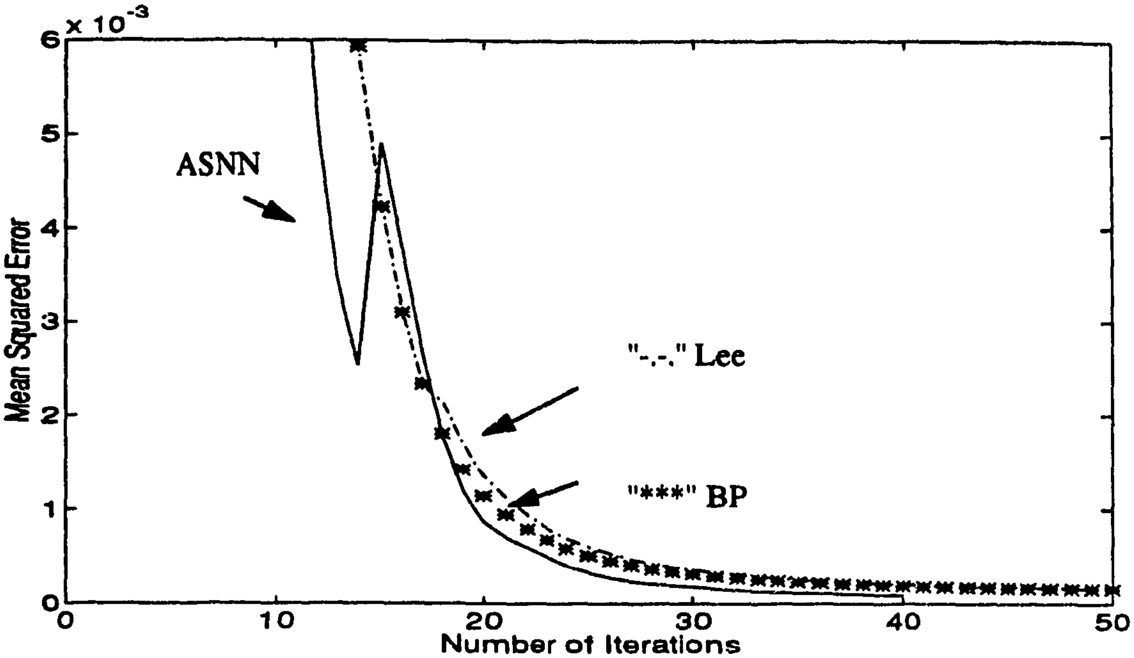

Figure 3.11: The local behavior for learning during the first 60 epochs

(character "C" with desired error = 0.0001)

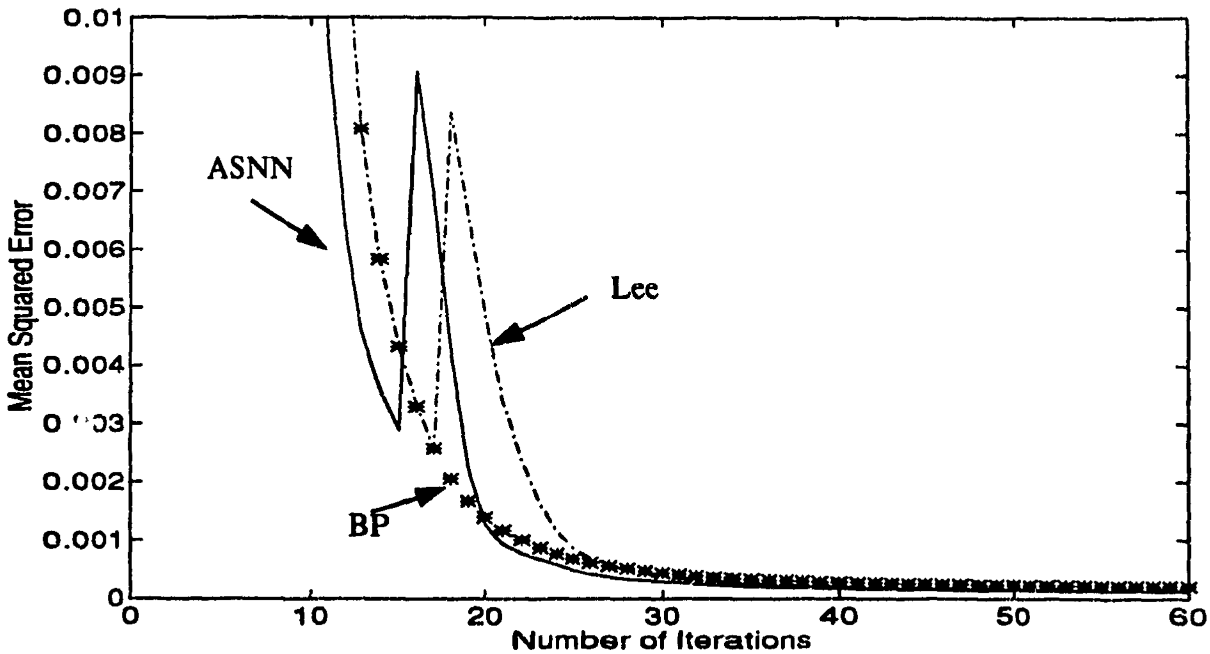

Figure 3.12: The local behavior for learning during the first 60 epochs

(character "D" with desired error = 0.0001)

3.3.2 XOR problem

XOR problem is a classic example often used for the simulation of existing neural networks. To train the network, the input layer is set to 2 neurons, the output layer is set to 1 neuron and the hidden layer starts with 4 neurons. Table 3-2 shows the results demonstrating that the ASNN performs better than the BP and Lee's algorithms. Figures 3.13 - 3.14 depict the error profiles.

Table 3-2: XOR simulation results

Desired Error	BP	Lee's Algorithm		ASNN			
	Iteration	Iteration	Added Neuron	Iteration	Added Neuron	Improved Rate (compared to BP)	Improved Rate (compared to Lee)
0.001	1704	1716	5	1202	5	29.5%	30.0%
0.0001	8486	9143	7	1802	6	78.8%	80.3%
0.00001	66110	78259	10	27468	10	58.5%	64.9%

Similar results as in the character recognition problem were obtained for the XOR problem. Between 29.5% to 78.8% of iterations were reduced by using the ASNN method compared to the BP and between 30.0% to 80.3% iterations were reduced compared to the Lee's method. Surprisingly, although new neurons were created when the Lee's method was applied, the network convergence was not improved significantly.

When XOR problem is applied to both Lee's and ASNN networks it is found that a similar number of new neurons are created (see Table 3-2), but ASNN has an average convergence improvement of 58.4% ((30.0% + 80.3% + 64.9%)/3 = 58.4) compared to Lee's method. The reason is that the split mother neuron and split time are difference between Lee's method and ASNN. This is another example showing that ASNN is a well designed structure level adaptation network.

In the XOR problem when the accuracy of the desired error increases the performance of the ASNN does not change linearly. For example, when the desired error is set to 0.001, 0.0001 and 0.00001, the improvements in network convergence between ASNN and BP are 29.5%, 78.8% and 58.5% respectively. In other words, the best performance of the ASNN is obtained when the desired error is set to 0.0001.

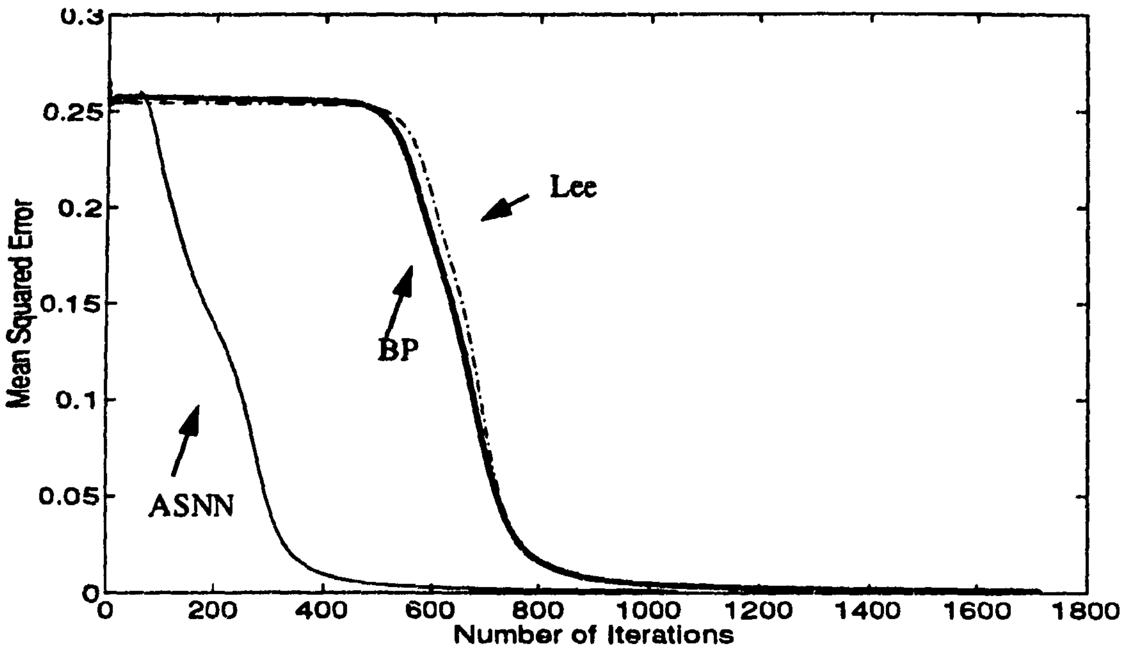

Figure 3 .13: XOR learning curve with desired error = 0.001

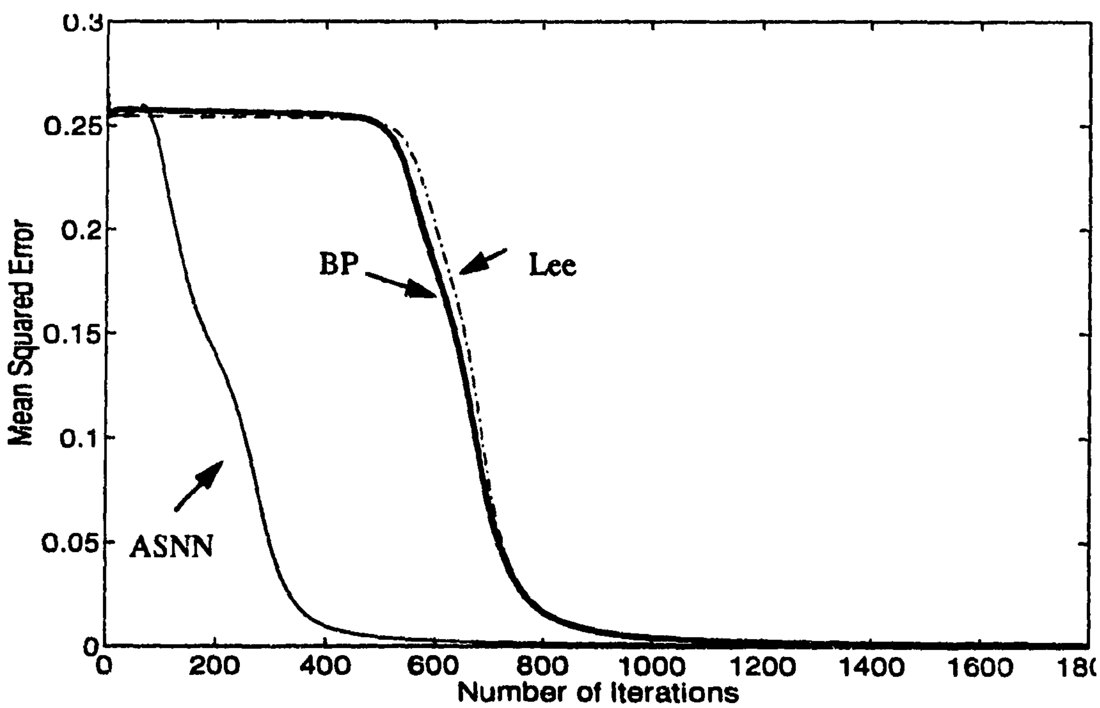

Figure 3 .14: XOR learning curvse with desired error = 0.0001

3.4 Analog results

3.4.1 4-leaf-rose problem

The polar to the rectangular coordinate change is given by:

$$R = \sin(2\Phi) \tag{3.1}$$

$$X = R\cos\Phi \tag{3.2}$$

$$Y = R\sin\Phi \tag{3.3}$$

where X and Y represent the cartesian coordinates and R and Φ represent the polar coordinates. The general shape of this function is a 4-leaf-rose inclined by 45^0 from the x-axis. For the purpose of the simulation the function is limited to only the first quadrant as shown in Figure 3-15. The input and output layers consist of 30 neurons each. The hidden layer starts with 4 neurons. Table 3-3 shows the simulation results comparing the BP, Lee's and the ASNN algorithms.

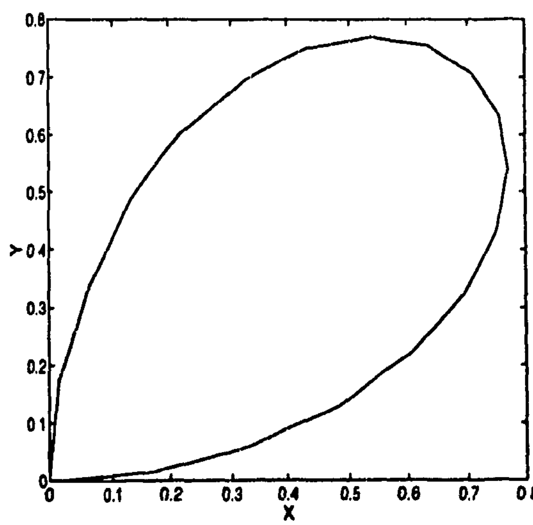

Figure 3.15: The first quadrant of a 4-leaf-rose function

Table 3-3: 4-leaf-rose simulation results

Desired Error	BP	Lee's Algorithm		ASNN			
	Iteration	Iteration	Added Neuron	Iteration	Added Neuron	Improved Rate (compared to BP)	Improved Rate (compared to Lee)
0.00001	466	286	7	169	10	63.7%	40.9%
0.000001	3833	1743	9	1162	13	69.7%	33.3%

Since only the first quadrant of the 4-leaf-rose function is simulated, the network complexity and training iterations are reduced substantially. However, the results still show that the ASNN performs better than the BP and Lee's methods, although more new neurons are created by the ASNN as compared to Lee's method (shown in Figures 3.18-3.19). For example, using Lee's method when the desired error is 0.00001, 286 iterations are required and 7 new neurons are generated. Using the ASNN method, only 169 iterations are required and 10 new neurons are generated. A better representation on new neuron generation for both the ASNN and Lee's network learning process is demonstrated in Figures 3.16 -3.17 and Table 3-4. For example, at iteration 13, the ASNN required another 2 neurons at the "mother" neuron positions of 1 and 2. The learning process continued with 8 more neurons added at different "mother" neurons positions. At iteration 169 the network satisfied the error requirement.

Table 3-4: The comparison for additional new neurons

Method	Number of additional new neurons	The iterations when adding new neuron	The positions of splitting " mother " neuron
Lee	1	16	2
	1	20	0
	1	27	5
	1	43	4
	1	79	1
	1	131	2
	1	251	2
	Total Iteration = 286		
ASNN	2	13	1, 2
	3	17	1, 2, 0
	1	44	4
	1	48	9
	2	66	4, 0
	1	98	4
	Total Iteration = 169		

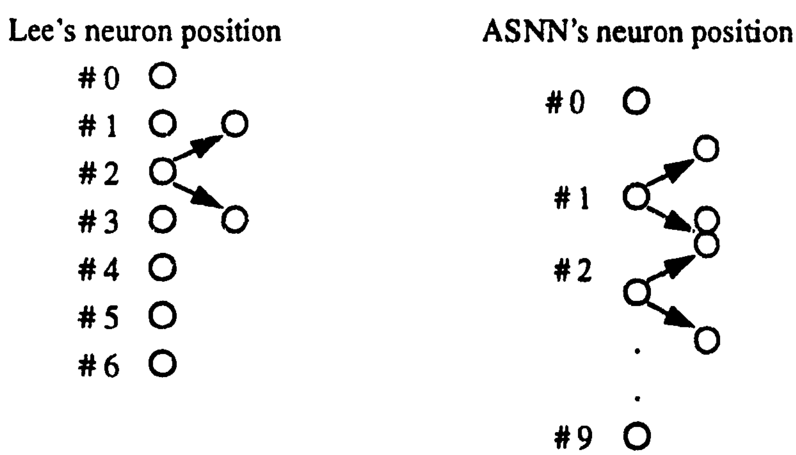

Figure 3.16: The neuron labels for the initial splitting in Table 3-4

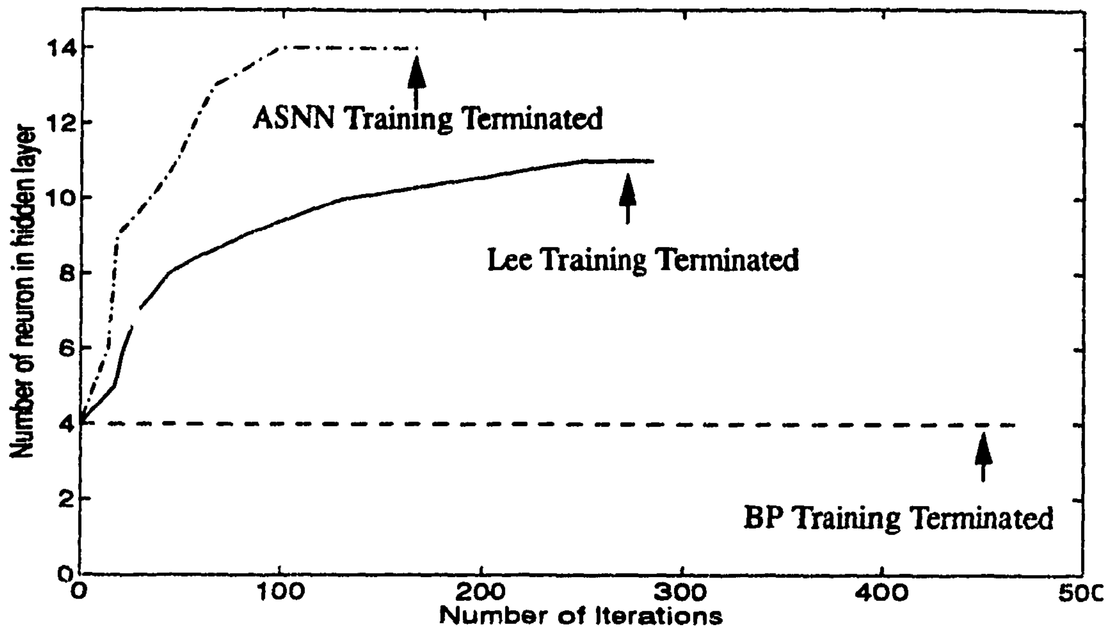

Figure 3.17: The relationship between the number of iterations

and the neuron generation (4-leaf-rose)

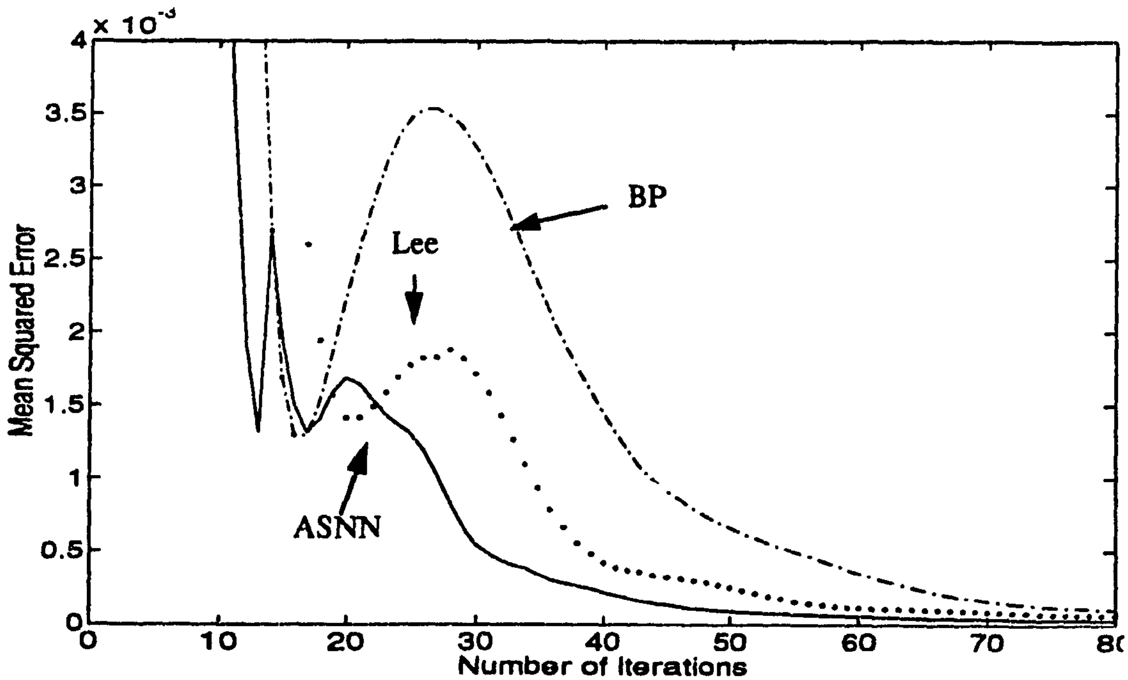

Figure 3.18: The learning curve for 4-leaf rose problem with desired error = 0.00001

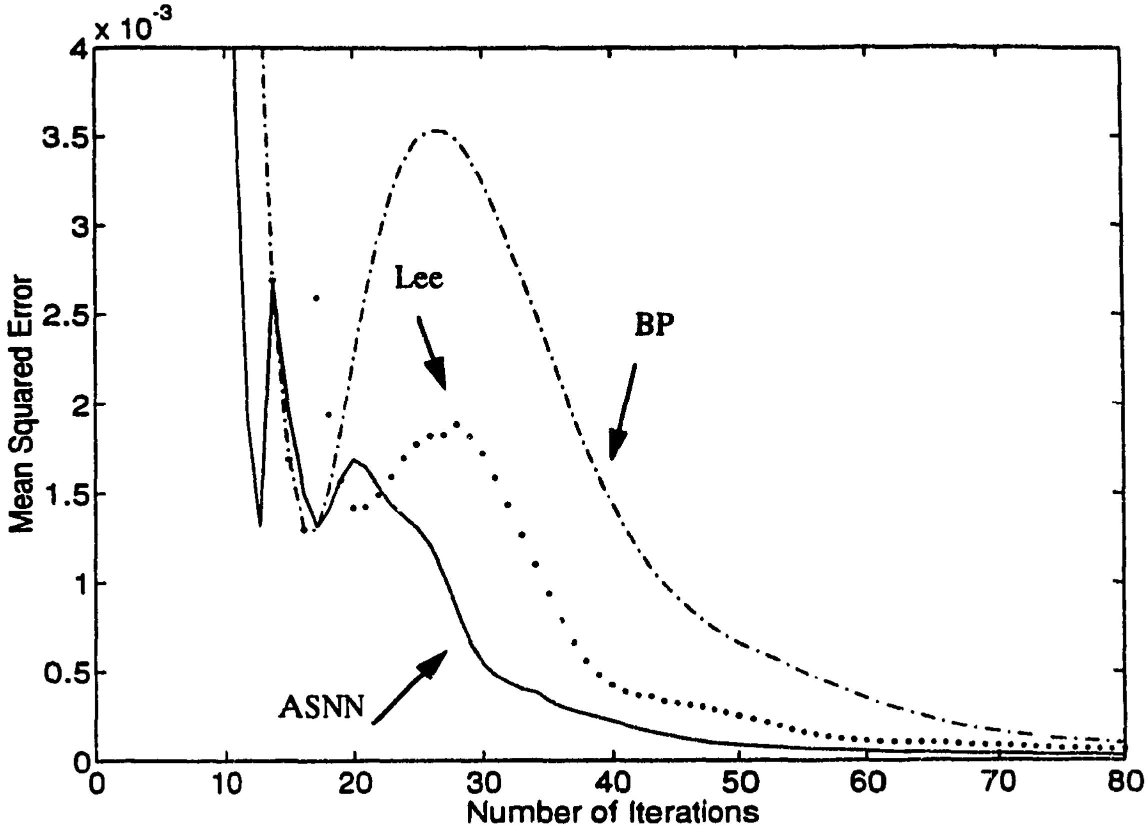

Figure 3 .19: The learning curve for 4-leaf- rose problem with desired error = 0.000001

3.4.2 One spiral problem

The spiral problem is used as another example for analog function approximation. The equations for spiral to rectangular coordinate transformation are as follows:

$$\rho = \alpha\,\Phi \tag{3.4}$$

$$X = \rho\,\cos\Phi \tag{3.5}$$

$$Y = \rho\,\sin\Phi \tag{3.6}$$

In this case the network has two continues-valued inputs and two outputs. The hidden layer starts with 4 neurons. The training set consists of 48 X-Y values.

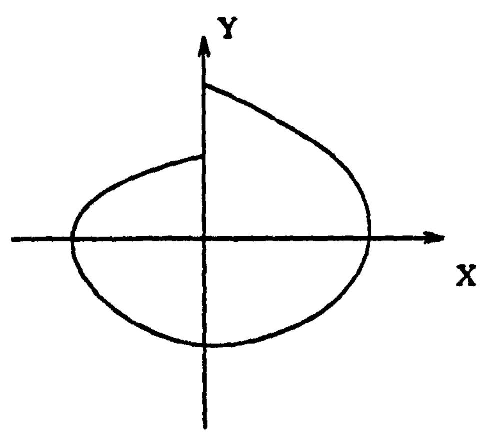

Figure 3-20: One- Spiral Problem

Table 3-4: One spiral simulation results

Desired Error	BP	Lee's algorithm		ASNN			
	Iteration	Iteration	Added Neuron	Iteration	Added Neuron	Improved Rate (compared to BP)	Improved Rate (compared to Lee)
0.001	2877	2304	2	1838	2	36.0%	20.2%
0.0001	20332	17885	5	12630	5	38.0%	29.4%

In order to simplify the learning process, only one spiral is selected for training the network. It is shown that the ASNN method performs best compared to other networks. However, compared to the results of character recognition, XOR problem and 4-leaf-rose function, it is found that the ASNN effectiveness is reduced. For example, to reach the desired error of 0.001, only 36.0% of training iterations were saved compared to the BP and only 20.2% of training iterations were saved compared to the Lee's method. These results are summarized in Table 3-5. It is noticed that only a few new neurons are generated for the one spiral problem. This is due to the fact that the one-spiral problem has many local minima to overcome. The small step size used also adds to the training process. This is easily verified from the slow error convergence plots shown in Figures 3.21-3.25. Since the learning plots for the BP and Lee's methods are combined in Figure 3.21, the comparison of learning curves between the BP and the ASNN as well as the Lee's and the ASNN methods are shown separately in Figure 3.22 to Figure 3.25.

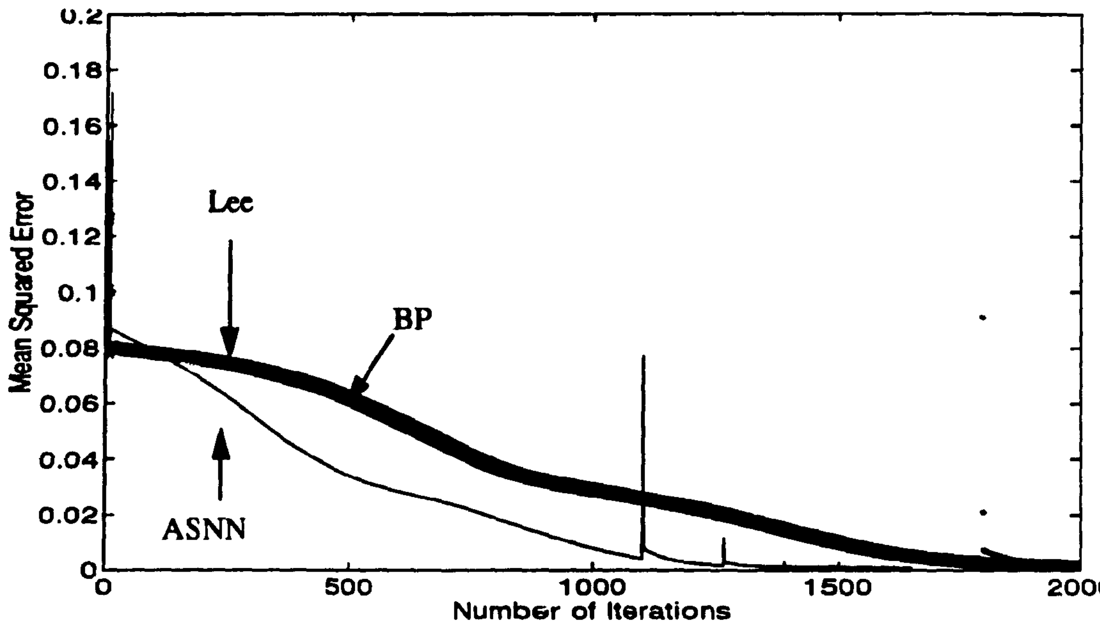

Figure 3 .21: The learning curves for the one-spiral problem using ASNN, Lee and BP algorithms with desired error Of 0.001

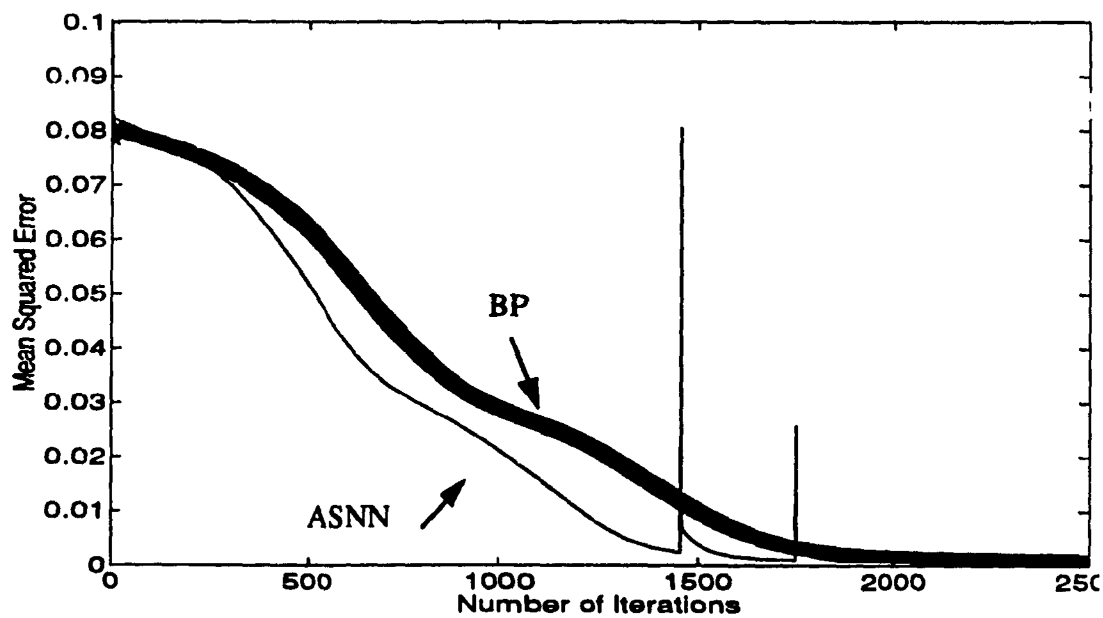

Figure 3 .22: The learning curve for the one-spiral problem using ASNN and BP algorithms with the desired error C.0001

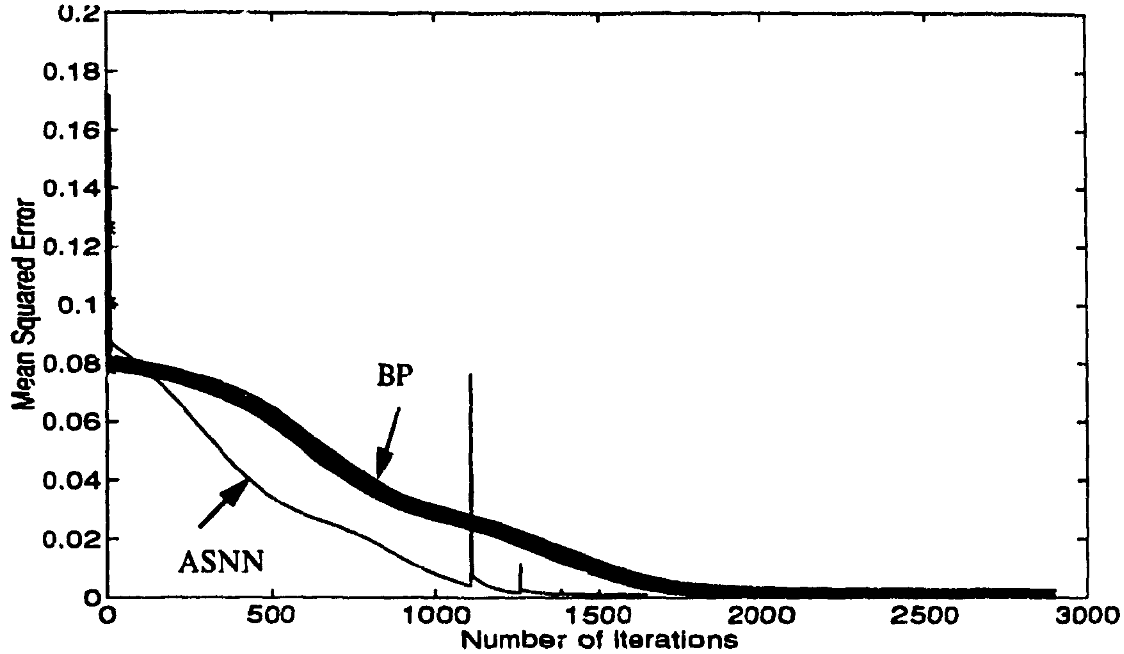

Figure 3 .23: The learning curve for the one- spiral problem using ASNN and BP algorithm with dsired error=0.001

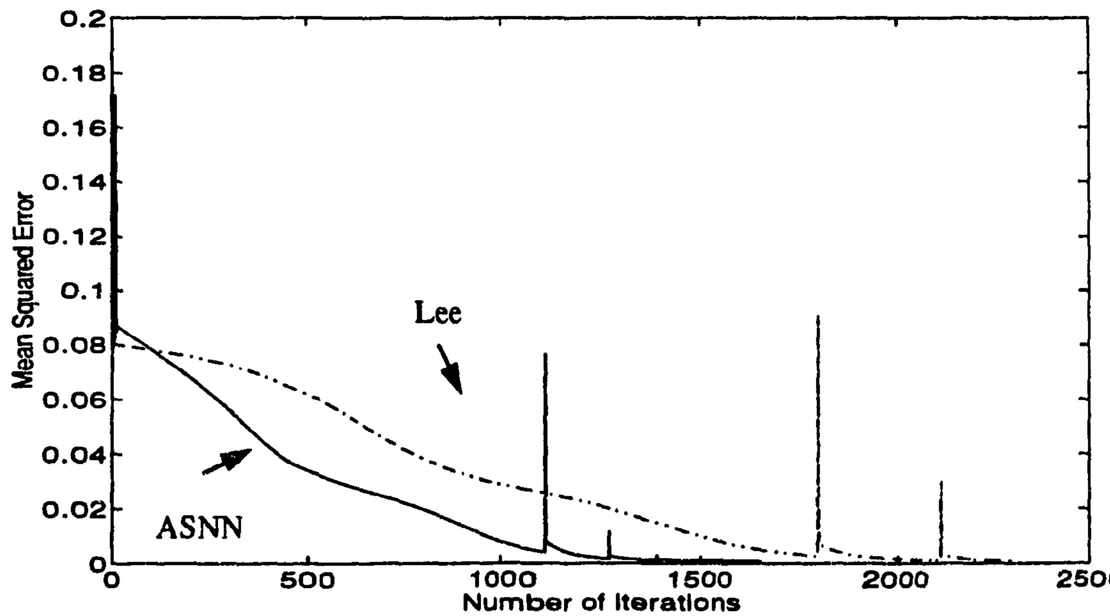

Figure 3.24: The learning curve for the one-spiral problem using the ASNN and Lee's algorithms with desired error of 0.001

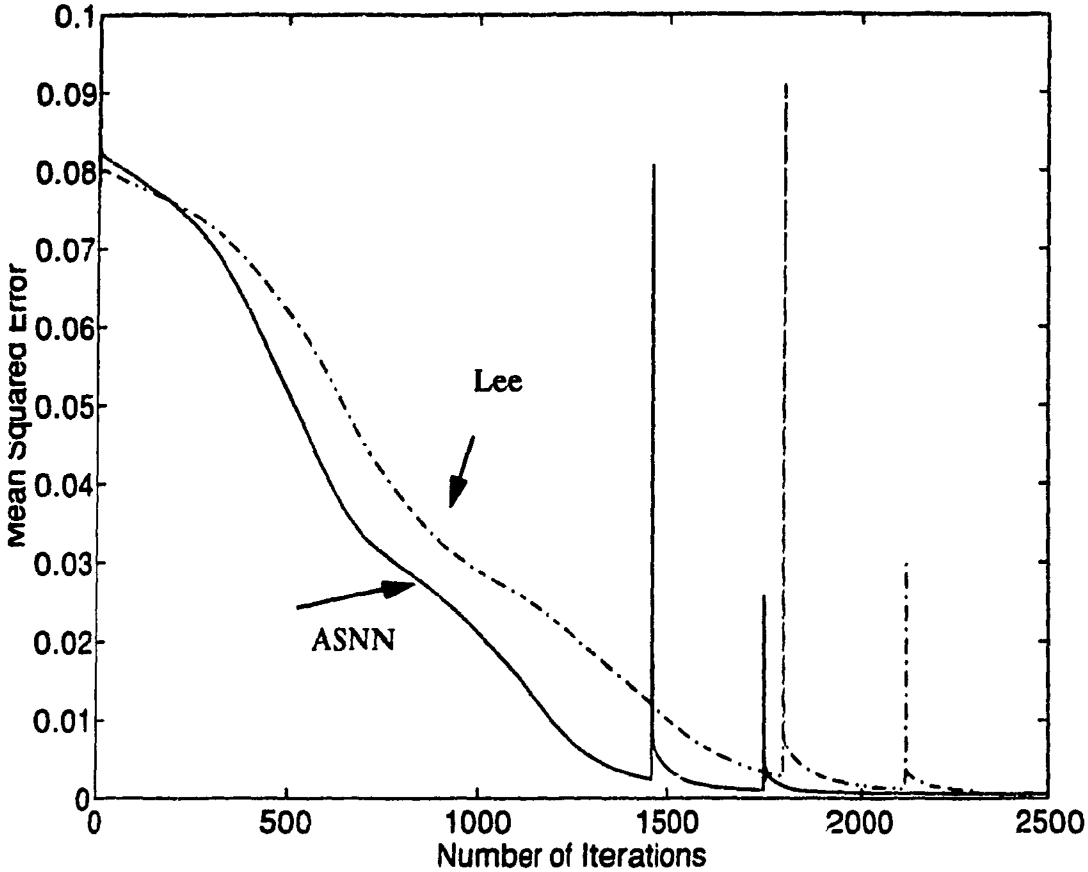

Figure 3.25: The learning curve for the one-spiral problem using the ASNN and Lee's algorithms with desired error of 0.0001

3.5 Summary

In this chapter, the proposed adaptive structure neural networks is applied to several examples. A comparison between the BP, Lee's and the ASNN methods show that the ASNN algorithm is not only superior to the traditional BP network, but also performs better than the Lee's network. Both discrete and analog problems are solved using these three networks.

In the discrete pattern recognition problem by using the ASNN, between 66% to 85% of training iterations are saved compared to the BP algorithm and between 22% to 66% of training iterations are saved compared to the Lee's algorithm. For the XOR problem, ASNN performs quite satisfactory with 29% to 78% fewer number of iterations compared to the BP and 30% to 80% fewer number of iterations compared to the Lee's algorithm. The ASNN algorithm is shown to be an improvement over the BP and the Lee's methods for solving these discrete problems.

For the analog function approximation, the three networks are used to solve both the 4-leaf- rose and the one spiral problems. When the three networks are applied to the 4-leaf-rose problem, the ASNN training iterations is improved from 64% to 70% compared to the BP and from 33% to 40% compared to the Lee's algorithm. The ASNN method performance for the one-spiral problem compared to the BP and the Lee's methods improves from 20% to 38%, respectively.

In all of the four simulation examples, we have shown that the ASNN algorithm may improve the iterations from 30% to 80% compared to the BP and the Lee's methods. In Chapter 4, a real world application problem, namely an Epileptic Seizure Detection algorithm will be used to test the ASNN capabilities.

CHAPTER 4

EEG Applications

4.1 Introduction

Electroencephalogram (EEG) is a signal that records the electrical activity of the outer layer of the cerebral cortex [33]. It is widely used to diagnose the brain disease in the clinical applications. Since Hans Berger invented the EEG in 1924, the EEG has been quickly recognized around the world. In the 1940's, the first EEG Automatic Low Frequency Analyzer was invented [34]. In the 1960's, the invention of the Fast Fourier Transform (FFT) [35] greatly reduced the time needed for electroencephalograph signal analysis. FFT made the on-line process of EEG signal frequency analysis possible and provided significant possibilities for practical clinical applications. Since the1970's, due to the advances in computer technology, other techniques have been applied to EEG research. For example, frequency analysis by using Autoregression (AR) model [36], time domain analysis techniques, and pattern recognition methods have been developed. In recent years, neural networks techniques have also been increasingly used for EEG research, especially for epileptic spike detections [7],[8].

The EEG signals are very complex and their statistical properties depend on both time and space. Generally, EEG signals are very weak and low in frequencies. The amplitudes recorded EEG signals range from 10^1 to $10^3 \mu v$ individually. The frequency rhythms can be divided into four regions, i.e.:

(i) δ: 0.5 ~ 3 HZ

(ii) θ: 4 ~ 7 HZ

(iii) α: 8 ~ 13 HZ

(iv) β: 14 ~ 30HZ

The recorded EEGs are different for adults and children. Normal children's EEG usually appears with higher amplitudes and lower frequency than those of adults. The dominant frequency rhythms also vary when people are in different states (such as awake, quite or sleeping). Analysis of EEG signals always involve questions of quantification, and accurate measurement is difficult because the behavior of the EEG is very weak and unstable. (see Figure 4.1 to Figure 4.4)

An important type of abnormal EEG is called seizure (see Figure 4.5: EEG classification). The beginning point of a seizure is called seizure onset. When a seizure occurs, the amplitude is first decreased and the EEG signal is desynchronized. Then, the EEG shows the appearance of moderate or high amplitude rhythm activity. Next, the presence of high amplitude electromyogram (EMG) obscures the EEG, and irregular paroxysmal activity appears. In some cases, the EEG may not appear to change at all during a small seizure. For clinical purposes, the initial seizure periods (0~20s) is the most important one because it reveals the vital information about epileptic focus.

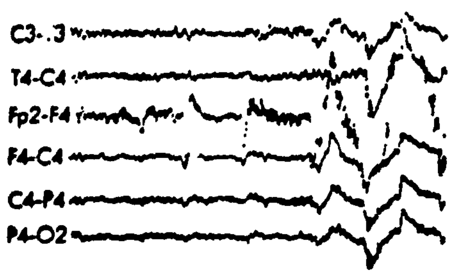

Figure 4.1: Active wakeness

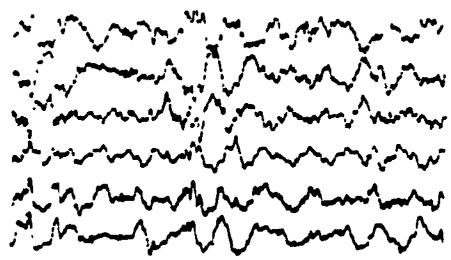

Figure 4.2: Slow EEG

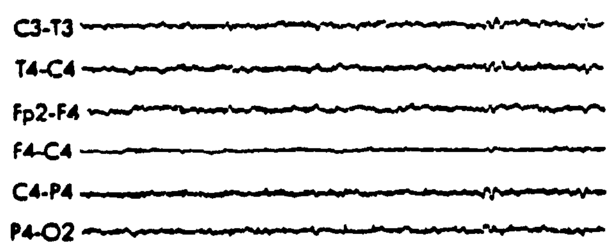

Figure 4:3: Desynchronized EEG

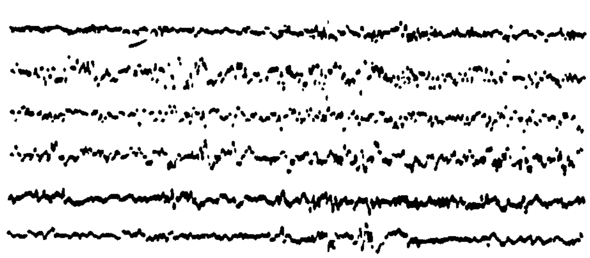

Figure 4.4: Phasic EEG

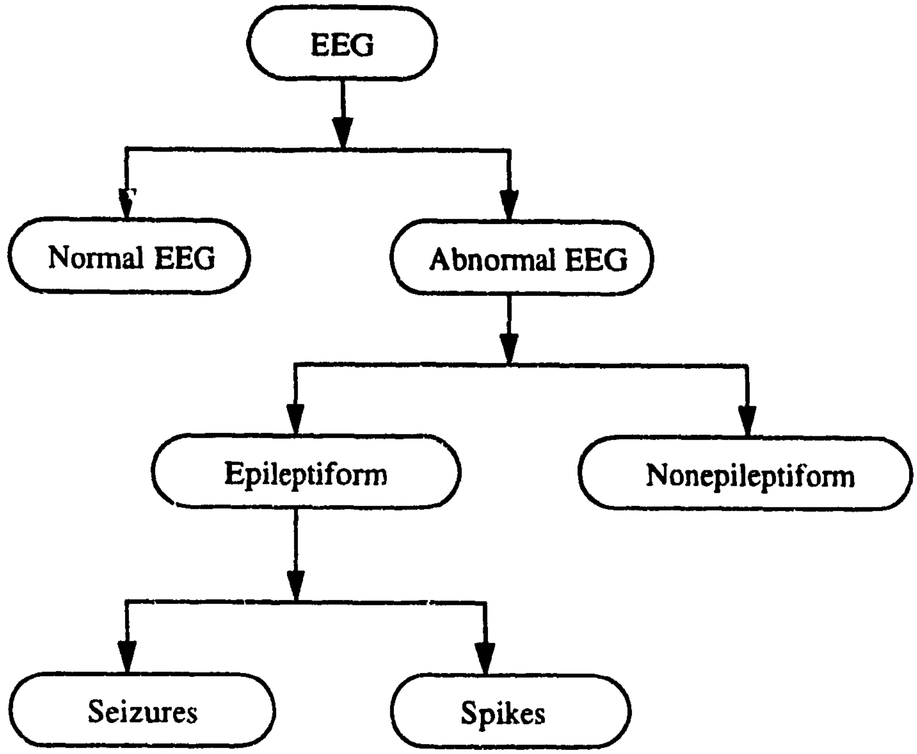

Figure 4.5: Scheme of EEG classification by EEGers

Figure 4.6 shows the normal EEG and Figure 4.7 to Figure 4.10 show four different seizure patterns (in the Figures 4.6-4.10: X-axis represents the time. Y-axis indicates the locations of 16 channels (see Figure 4.11(a)) which are placed on the human scalp). These four seizure patterns are only a small part of the overall seizures, but they clearly present local seizures, generalized seizures, seizures clearly identified from EEG background, and seizures completely mixed up with the EEG background.

EEG activities can be obtained in two ways by spontaneous or inductive methods. The spontaneous EEG is recorded without any external influence, whereas the inductive EEG is recorded using the man-made stimulation such as sound, lights, etc. This thesis focuses on the research on spontaneous EEG.

Figure 4.6: The example of normal EEG

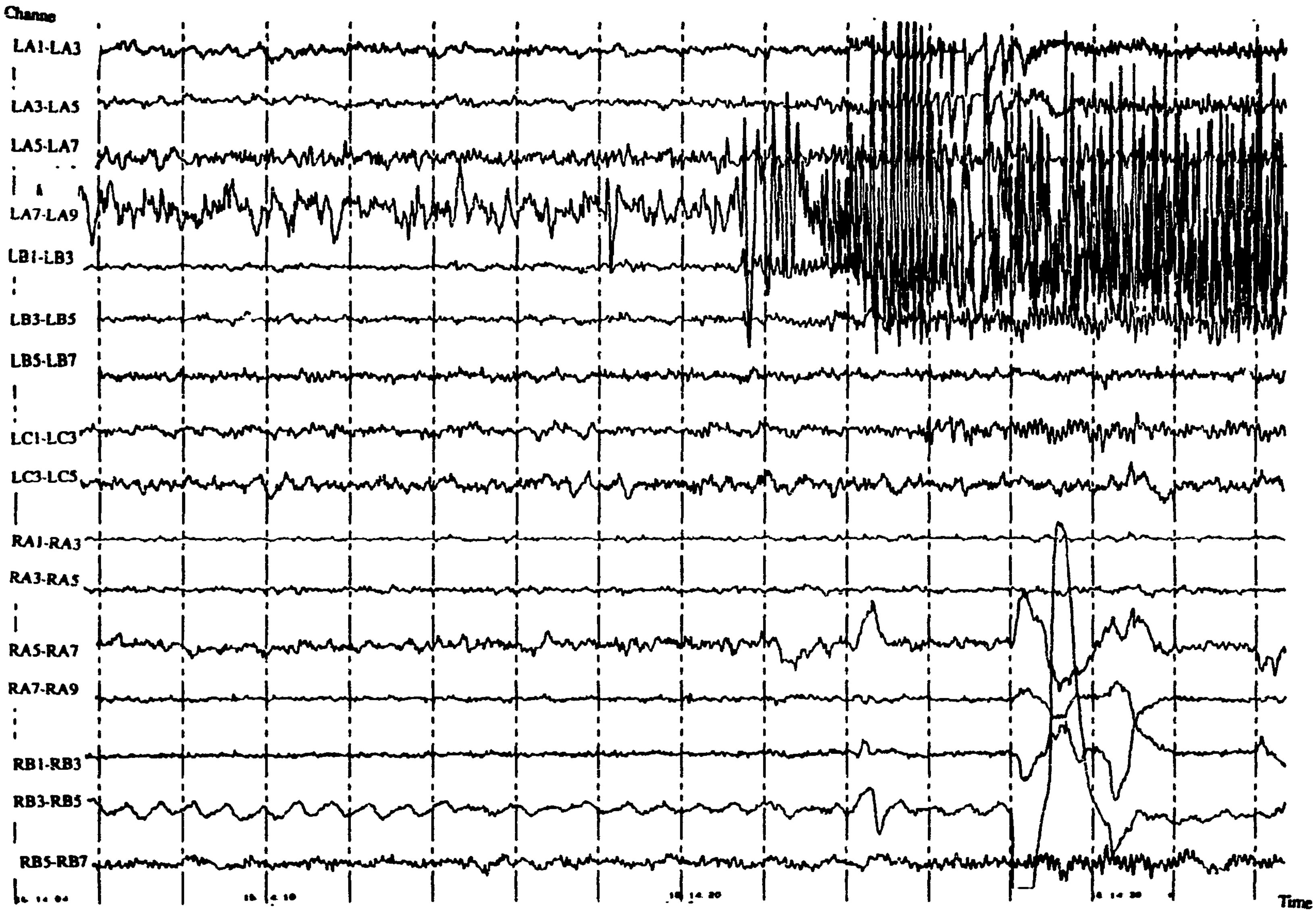

Figure 4.7: Local Seizure

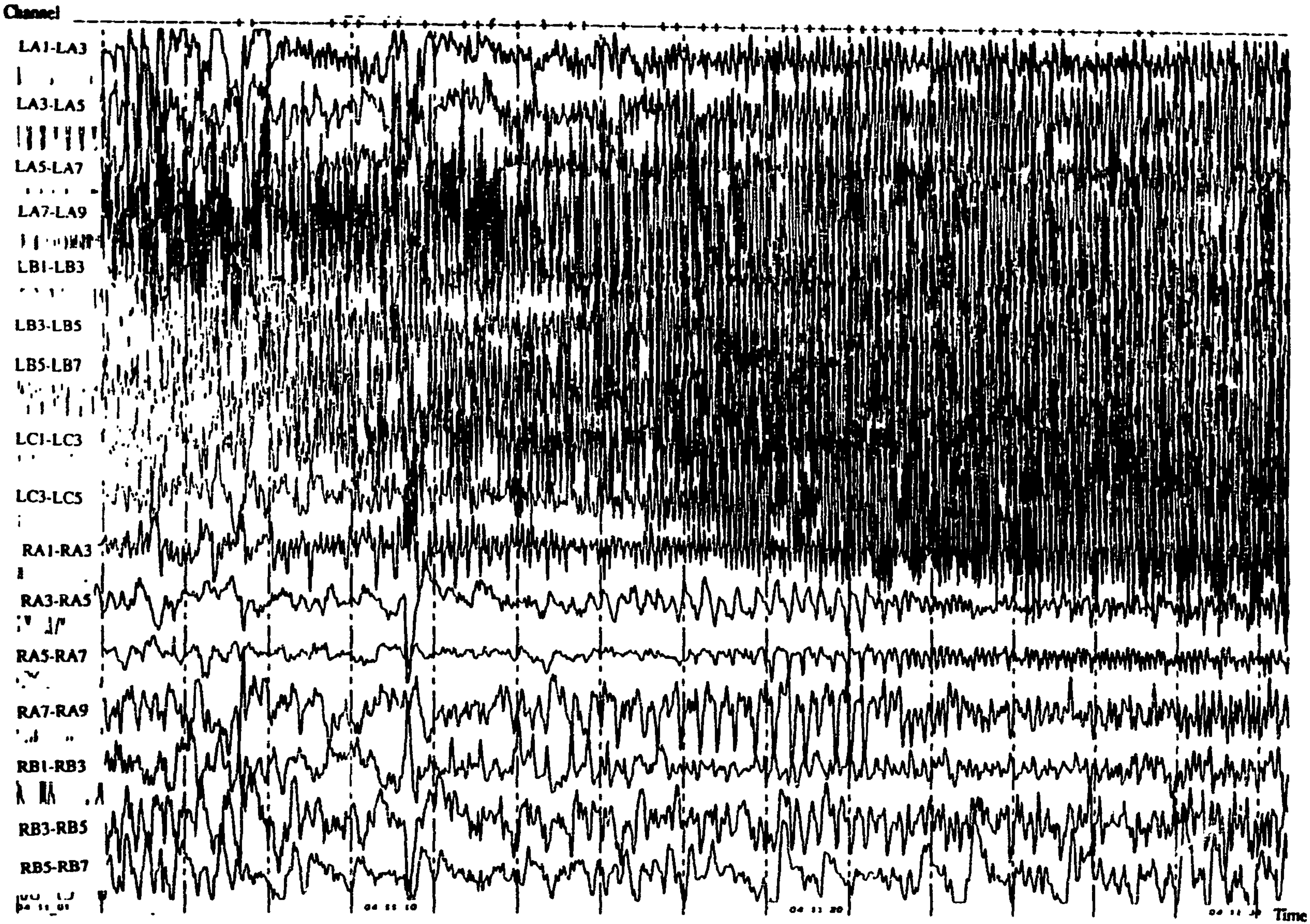

Figure 4.8: Generalized Seizure

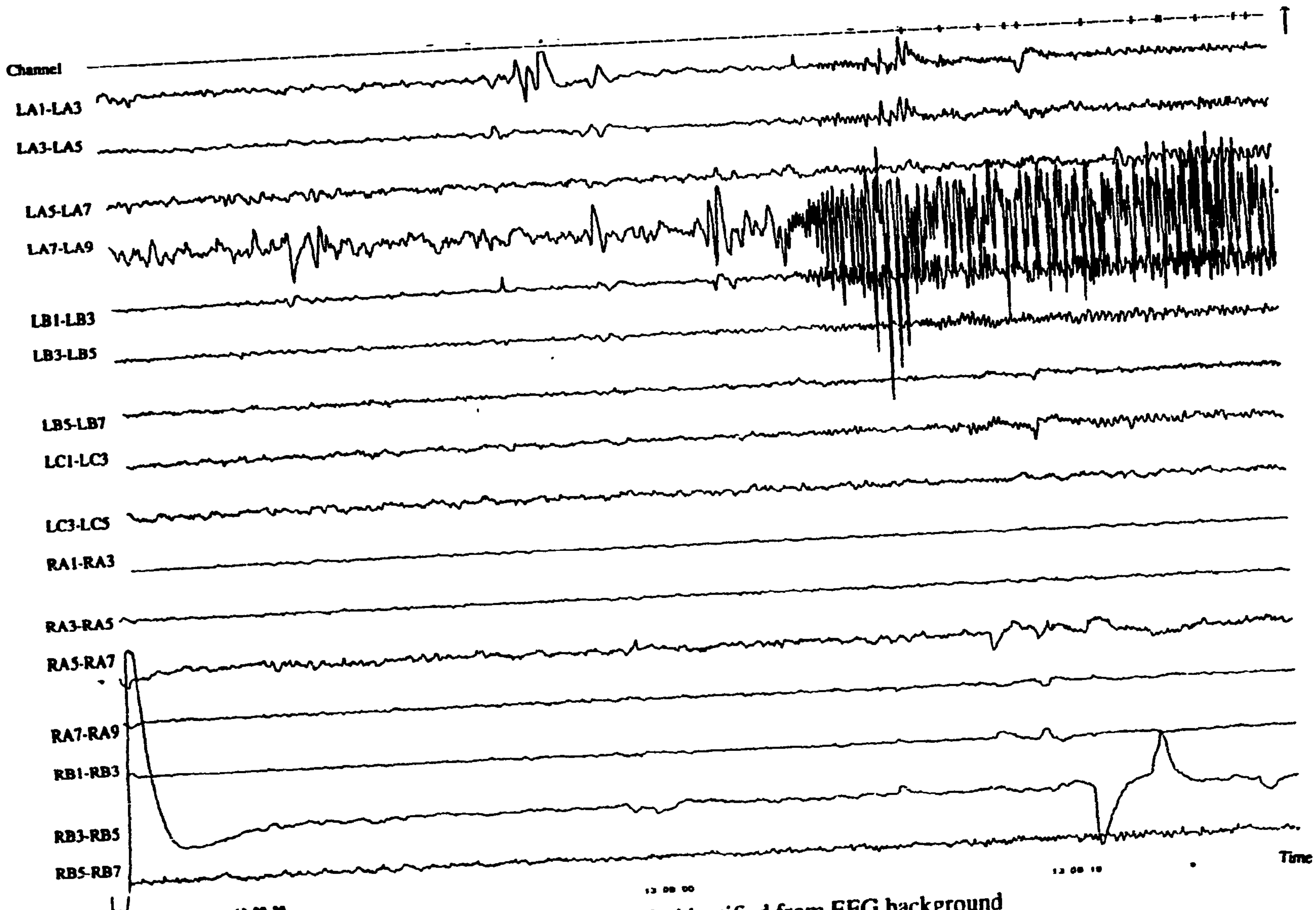

Figure 4.9: Seizure clearly identified from EEG background

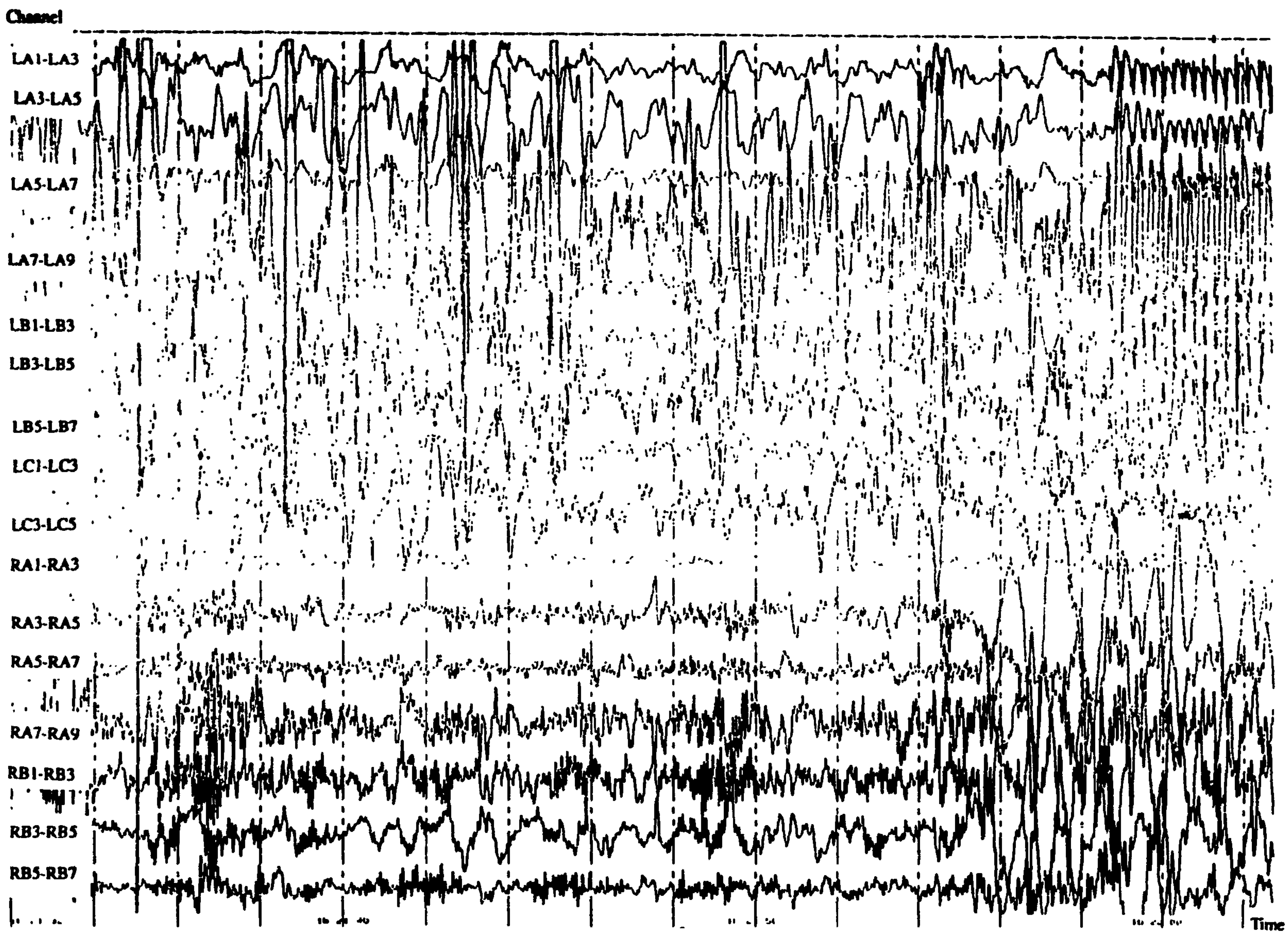

Figure 4.10: Seizure complete mixed up with the EEG background

The spontaneous EEG is obtained by means of electrodes placed on the scalp and is most useful in the investigation and management of patients with suspected epilepsy. The recorded epileptic seizures are particularly helpful to doctors in the treatment of patients. Because seizures usually occur infrequently and unpredictably, automatic detection of seizures during long term EEG monitoring sessions is required. For long term EEG monitoring, distinguishing between seizure and non-seizure is important.

Over the past 20 years, numerous efforts to automate the detection of epileptiform activity have been made and comparatively good results have been obtained [39]-[42]. This thesis is the first approach to real-time automatic epileptic seizure detection by adaptive structure neural networks.

The developed ASNN is based on the standard back-propagation neural networks. A three layer (one input layer, one hidden layer and one output layer) ASNN is used to map the desired precision of EEG seizure recognitions. The EEG testing data is provided by Dr. Jean Gotman and Mr. Qu of the Montreal Neurological Institute (MNI).

In the following sections, several important topics will be discussed. One is the EEG data acquisition and seizure feature extraction. EEG data acquisition enables us to record EEG for later analysis. Seizure feature extraction enables us to use only the most significant seizure features to detect the true seizures. A Seizure Detection Method (SDM) is proposed to classify seizure patterns. The last topic covers the results when different approaches are applied to EEG seizure detections.

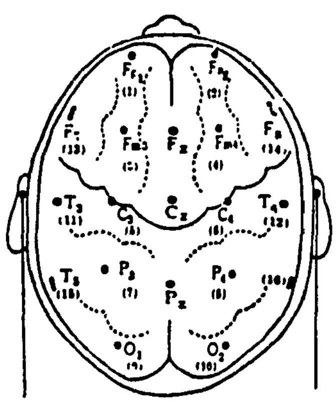

Figure 4.11(a): The location of 16 EEG channels placed on human scalp

4.2 EEG data acquisition and seizure feature extraction

4.2.1 EEG data acquisition

EEGs are recorded by electrodes placed on a human's scalp. The electrodes are made either with sphenoidal or chronic multi-conductors. A 16 (or 32)-channel cable elementary system is used to transmit the EEG. The system includes a 16 (or 32) amplifier and a multiplexer. In order to perform selective recording, the computer first delays the EEG by 2 minutes, so that at every instant the last 2 minutes of the EEG are stored in the computer disk. When a seizure is detected, the computer can then record the 2 minutes of EEG before and 2 minutes of EEG after. The detection of a seizure is obtained by the pressing of a button by either the patient or the observer; it can also be signaled by the computer itself which continuously processes the EEG in an attempt to recognize EEG seizure patterns [39]. At the same time, the patient's behavior is also recorded by a TV camera for later review. The related diagram is shown in Figure 4.11(b).

In this study, sixteen channel EEGs from 5 patients with seizures and non-seizures were recorded at the MNI. These EEGs, digitalized in selected portions with 12 bit resolution at 200 Hz sampling rate, were stored on a computer hard disk. In order to obtain the most tentative EEG, one minute of each EEG was sampled every 30 minutes. The total sampling lasted 2-4 days for each patient. This sampling method is widely accepted for the modelling of long time monitoring because all the possible states of patients can be captured.

4.2. 2 Seizure Feature Extraction

Automatic recognition of seizures is a very difficult problem since the EEG is not well defined morphologically. To apply a neural networks to seizure recognition, it is necessary to extract the most representative seizure features. Recognition of the EEG seizures is based on a seizure's characteristics, property and clinical phenomena. The EEG is broken down into epochs, each epoch lasting 2.56 seconds which are based on the sampling rate of 200 Hz and is easy for FFT calculation (2.56 * 200 = 512). Three time domain variables and two frequency-domain variables are defined and extracted from each EEG epoch in all 16 channels. After these features are extracted, they will be used as the input data either for ASNN to train seizure patterns or to detect seizures. The five features are defined below:

(a) Average EEG amplitude: This is determined in each epoch. HALF-WAVE method [39],[42] is applied to break the EEG down into half-waves. Then the amplitude of each half-wave is measured. The mathematical expression is as follows:

$$\overline{AM}_{hf} = \sum_{i=1}^{M} AM_i \; / M \qquad (4.1)$$

where:

$\overline{AM}_{hf}$ is the average EEG amplitude in one epoch

AM_i is the amplitude measured by one half-wave

M is the number of half-waves in one epoch

(b) Average EEG duration: This is the average duration in one epoch. Similar to the measurement of average EEG amplitude, the Half-Wave Method is used to measure the duration of each half-wave. The mathematical expression is as follows:

$$\overline{DUR} = \sum_{i=1}^{M} DUR_i / M \tag{4.2}$$

where:

$\overline{DUR}$ is average EEG duration in one epoch.

DUR_i is the duration measured by one half epoch

(c) Coefficient of variation: This is the ratio of the standard deviation to the mean value. The feature is used here as a measure of the 'rhythmicity' of the EEG; it is independent of the EEG frequency and it represents the degree of regularity in its duration.The mathematical expression is as follows:

$$COVA = \frac{\sum_{i=1}^{M} (DUR_i - \overline{DUR})^2}{\overline{DUR}^2 \times M} \tag{4.3}$$

where:

COVA is the coefficient of variation

(d) Dominant Frequency: For each epoch of the EEG, the dominant frequency indicates which frequency is the most important one. It can be expressed as a peak in the frequency spectrum. When a seizure happens, rhythmic discharge with large amplitude at a certain frequency is significant. The dominant frequency feature is obtained from power spectrum analysis.

(e) Average Power Spectrum: After one epoch of the EEG is processed by power spectrum analysis, its average power spectrum is calculated. This feature is used to estimate seizure behavior in the spectrum domain.

4.3 ASNN seizure recognition method

After the EEG data is recorded, experienced clinical doctors first review all the EEG to distinguish seizure and non-seizure patterns. The initial seizure patterns from the same patient may vary. On the other hand, the initial seizure patterns from one patient may be very similar to the non-seizure patterns of other patients. These phenomena compounds the difficulty of seizure detection algorithm.

The EEG raw data is broken down into epochs and processed sequentially to extract the 5 features from both time and frequency domains. These extracted features are then sent to the ASNN for either training or recall process.

In order to increase seizure detection rate and the possibility for the clinical use of the proposed method, the following assumptions are required for the studies.

(1) Patients are treated individually and the first one or two day's recording is used to extract different seizure patterns.

(2) **Training set** : A complete EEG training set contains both non-seizure and seizure set. A non-seizure training set is selected as in the following steps:

- The EEG is selected in the interval of one to two minutes before the patients's seizure occurs.
- Random 25.6 second segments of non-seizure EEG (10 epochs) are extracted in the selected time interval.

  In practice, the non-seizure training set is not restricted just before seizure occurs. Since we set the parameter to record the detected seizures two minutes in advance, the selection of non-seizure EEG corresponds to our recorded data.
- The seizure training set is also randomly selected from any recorded seizure patterns. Because the duration of seizures changes in each patient, the length of seizure training set varies from 20 ~ 40 seconds and it usually begins from the seizure onset.

(3) **Recall set**: All the rest of the EEG from the same patient, including the training set and the recording data from later days, form the recall set.

From the detailed research on seizure patterns some common characteristics of seizures can be summarized as following:.

(a) A seizure is the phenomenon of rhythmicity discharge from either a local area or the whole brain of the patient.

(b) Each seizure's individual behavior usually lasts from seconds to minutes.

From the above discussion, a Seizure Detection Method (SDM) is proposed to improve the accuracy of seizure detections in the recall process:

(1) Each 2.56s EEG constitute a detection epoch. Features are then extracted to categorize each epoch as seizure or non-seizure.

(2) A seizure detection threshold for the EEG epochs is set at 0.6. This value reflects the similarity between the training of the EEG and the recall of the EEG and is used to deter-

mine whether an individual detection epoch is a seizure one or not. This value was determined in the following way. During our experiments, the threshold was initially set to 0.8 and it was found that some of the seizures were lost because the 80% similarity between training and testing sets could not be reached. A value of 0.6 can match most of the seizure patterns. Unavoidably, some of the inconsistent non-seizure epochs were also detected as seizure epochs. Most of these inconsistent seizure epochs can be eliminated by the SDM method.

(3) To distinguish the change from non-seizure to seizure or vice versa, 5 subsequent epochs need to be processed. If the first epoch, plus 3 of the remaining 4 epochs, indicate a seizure, then the phase is considered as a seizure phase. This is because when a seizure occurs, it usually lasts from 10 seconds to several minutes. To maximize number of seizure detections, 5 subsequent epochs (5*2.56=12.8 seconds) are selected as the base. The same procedure is used to detect non-seizure phases. This reflects the fact that only a significant section (>=12.8s) is required to detect a change.

4.4 Comparison of different methods for EEG seizure detection

Seizure and non-seizure EEG data of 5 patients were recorded (each patient may have more than one seizure pattern) and applied to ASNN and BP to explore the following three points:

1. Comparison of seizure detections between physician observation and ASNN
2. Comparison of network performance of seizure detections between ASNN and BP.
3. The reduction of False Seizures Detections(FSD) during long term seizure monitoring.

A three layer network is used and initialized with 4 neurons in the hidden layer to test ASNN and for comparison with the BP. The parameters selected for both networks are:

learning rate = 0.15

momentum = 0.9

random initial weights

To test the ASNN, additional parameters are selected:

FRTH = 0.2

$\lambda = 0.2$

$\gamma = 1.0$

$\sigma = 0.2$

To train the network, the input layer is set to 80 neurons (16 EEG channels * 5 features). The output layer is set to one neuron.

4.4.1 Comparison between physician observation and ASNN

In physician observation, experienced doctors review all the recorded EEG's and classify seizure or non-seizure patterns. The ASNN simulates this recognition of patterns by the physicians and thus is very significant for long time seizure monitoring. To evaluate the performance and the practical applicability of the ASNN, the number of seizures detected by the automatic detection method is compared to the number detected by the physicians' observation. From the results of Table 4-A (shown in Appendix I) , we see that all the 38 recorded seizures by physician's observation were detected from the tested patients by using the ASNN network. Furthermore, the duration of seizures detected by the ASNN agrees with the time observed by the physicians. For example, in the case of patient 1, two different seizure patterns were classified by the physicians from the recorded EEG. Three seizures were recorded by the computers for each seizure pattern. All of these 6 seizures are detected by the ASNN. For the recorded EEG for patient 1, the experienced physician noted that the starting time of a seizure is 18:22:40 and the ending time is 18:23:22. Similarly, the ASNN marked the starting time of the detected seizure at 18:22:42.6 and the terminating time at 18:23:23.6.

Table 4-1: Summary of comparison in the seizure detection between physician observation and neural network estimation

Number of Patient	# of detected seizure		The seizure start time detected by the ASNN compared to physician*		
	Physician	Neural Network	Maximum difference	Minimum difference	Average difference
Patient 1	6	6	-3s**	+0.8s	+0.97
Patient 2	7	7	+2.48s***	0s	+0.19
Patient 3	5	5	-1.0s	-0.1s	-0.56
Patient 4	8	8	+53s	-0.2s	-0.84
Patient 5	12	12	+19.8s	0s	+6.08
Notes	* : Assuming the beginning time of seizure detection at 0 (s) by physician				
	** : " - " means the seizure start time by ASNN is later than physician				
	***: "+" means the seizure start time by ASNN is earlier than physician				

Table 4-1 shows the summary of comparison in the seizure detection between physician observation and neural networks estimation. Assuming that the beginning time of seizure marked by the physicians is 0, the results of applying ASNN to patient 1 show the maximum time difference of the detected seizures between ASNN and physician observation is 3 seconds, the minimum time of difference is 0.8 seconds and the average time difference is 0.97 second

Three reasons account for the good results shown in Table 4-1. First, the classification of seizure patterns is clear because each patient is treated individually, and different seizure patterns are defined for each patient. This classification has its clinical meaning in

EEG. It is worth spending one or two days to record the EEG as a training base in order to distinguish seizures or non-seizures while patients are usually monitored for weeks. The second reason is that the ASNN performs well. As mentioned in chapter 2, one of the powerful functions of neural networks is pattern recognition. Seizures can be classified as various patterns. The ASNN not only classifies patterns, but also forms auto-adaptive structures. This make the ASNN more powerful and effective than BP. Features extracted from the EEG, reveal the differences between seizure and non-seizure patterns. The third most important reason is that the proposed seizure detection method (SDM) is properly developed. The SDM simulates the process of physician observation and reflects reality of the EEG's behavior. When a seizure occurs, it usually lasts from seconds to minutes. When a physician reviews the EEG, he / she not only is concerned with the initial seizure onset, but also goes through the whole process of the seizure and watches the patient's behavior on a video, if available.

4.4.2 Performance comparison between ASNN and BP

The BP is a fixed network structure and has its limitations for applications. The other major weakness of the BP is its low convergence speed. Whereas ASNN is an adaptive network structure which not only is suitable to solve complex problems but also has a very fast convergence speed. Table 4-B(a) to Table 4-B(e) (See Appendix II) show the results for seizure detections by using both ASNN and BP. Significant differences are summarized as follows:

4.4. 2.1 Comparison with different number of initial hidden neurons

As shown in Table 4-B(a) to Table 4-B(e), different number of initial hidden neurons (2, 3, 4, 6, 8) were selected to verify the consistency of the ASNN. The highest improvement rate is 85.2% with 3 initial hidden layer neurons obtained from the EEG file1of patient 1 in Table 4-B(b). The lowest improvement rate is 45.4% with 4 initial hidden layer neurons obtained from the EEG file1 of patient 2 in Table 4-B(c). The overall average improvement rate is 71.59%. Table 4-2 shows a summary of the ASNN performance rates. We found that the improvement of the ASNN performance is very stable and consistent (the range is between 69.90% ~ 73.75%) over the BP. On the other hand, when the number of initial hidden layer neuron increased, the performance of both the BP and the ASNN is not suitable for the EEG applications. This is due to the additions in local minima, the differences in the initial weights and the complexity of the EEG data. But in general, the performance of the ASNN is superior to that of the traditional BP.

4.4.2.2 Comparison with the same number of initial hidden neurons

For the EEG application, Table 4-B(a) to Table 4-B(e) show the results for individual fixed initial hidden neurons for both the BP and the ASNN. Each table shows that the number of iterations for the training is greatly reduced by using the ASNN. For example, in file 3 of patient 5 (see Appendix II: Table 4-B(c)), the BP required 19752 iterations to reach a desired error of 0.0001. However, using ASNN to reach the same error, only 2896 iterations are needed when 6 new neurons are added. In other words, by adapting the network structure with 6 additional neurons, the convergence speed of the network has been improved by 83.4%.

4.4.2.3 Comparison of BP and ASNN when BP initial hidden layer neurons are the same number as the ASNN's final hidden layer neurons

The goal of this experiment is to verify the superiority of the ASNN when BP initial hidden layer neurons are the same number as the ASNN's final hidden layer neurons. For example, in File 3 of patient 5 (see Appendix II: Table 4-B(c)), with the 4 initial hidden layer neurons,ASNN required 2896 iterations to reach the desired error of 0.0001. During the process, the ASNN generated 6 new neurons in the hidden layer. So the final number of neurons in the hidden layer is 10. If these 10 neurons are set in the BP as initial hidden layer neurons, the BP still required 17099 iterations to reach the desired error of 0.0001. In other words, starting with 10 hidden layer neurons in BP and 4 hidden layer neurons in ASNN, the convergence speed of ASNN is 83.1% faster than BP. The summary of ASNN improvement rate compared to BP is shown in Table 4-2.

Table 4-2: The summary of ASNN improvement rate compared to the BP

Number of initial neurons in hidden layer	The highest improvement rate	The lowest improvement rate	5 patients Average rate with same initial neurons	5 patients Average rate with same final neurons
2	82.7	57.3	73.11	66.17
3	85.2	60.0	73.75	63.23
4	83.4	45.4	69.80	64.94
6	80.1	56.0	70.62	62.74
8	76.0	64.4	70.67	65.74
Ave. Rate	81.48	56.62	71.59	64.56

4.4.3. Reduction of the false seizure detections (FSD) during long term monitoring

In clinical practice, catching the real seizures is the most significant issue. However, the reduction of the FSD is also an important topic to consider for the new algorithm. Too many FSD's will increase the tedious work for the physicians, and even make long term seizure monitoring unacceptable.

This is a difficult topic and has been studied for many years by different approaches [42],[43]. Our approach to this problem is to focus on both the efficiency of FSD and the feasibility for a practical use. It is comprised of the following two stages: initial FSD reductions and secondary FSD reductions.

4.4.3.1 Initial FSD reductions

This is obtained based on the following conditions:

(i) Doctors score and define different seizure patterns

(ii) 20s~30s seizure EEG's is selected randomly from each different seizure pattern to form the training set

(iii) 25.6s (10 epochs) non-seizure EEG's is also selected randomly from each patient for training set

Table 4-C (Appendeix III) shows the initial results of FSD reduction with the recorded EEG's of 5 patients. From the results of this table we find that the FSD rates vary with different patients. The false detection rate varies between 0.12/hr to 15.2/hr. The average false detection rate is around 7.06/hr.

Two reasons explain why the average false detection rate is high:

(i) Some of the seizures are difficult to distinguish from their EEG background, even for the physicians. For instance, for Patient 2, the background amplitude of this patient is high with large sharp waves. Many fast physical activities also appear on the EEG and their frequencies are similar to this patient's localized seizures. Figure 4.10 shows one of this patients' EEG with a seizure and unclear background. This increases the difficulty of neural networks recognition.

(ii) The non-seizure training set may not be properly selected. The training set includes two parts of both seizure patterns and non-seizure patterns. The initial selection of non-seizure pattern is made by randomly collecting 25.6s of continuous EEG from the same patient. This random selection cannot cover all the non-seizure EEG patterns. Non-seizure EEG patterns include all the stages of the patients such as awake state, quite state, physical state and sleeping state. The question is how to select the limited non-seizure set to represent the generalized non-seizure patterns for training process.

4.4.3.2 Secondary FSD reductions

Although the average FSD level of 7.06 per hour obtained from the initial results is acceptable for clinical purposes, it still has a lot of room for further improvement. In research of initial FSD rates, it was found that the 25.6s non-seizure EEG training set which was selected randomly was not able to cover all kinds of non-seizure patterns. It would be useful to find a self-learning method to create a non-seizure training set for the neural networks. This self-learning method should be able to collect non-seizure patterns which are closer to the border of seizure patterns. To accomplish this, the following four steps are used:

step 1: Perform the initial FSD;

step 2: Select one epoch of EEG from the initial FSD results which has the maximum average value for both amplitude and duration and add this epoch of EEG to the non-seizure training set;

step 3: Re-train the ASNN and perform the reduction of FSD;

step 4: If all the FSDs are eliminated or the FSD rate reaches the tolerant level, then stop FSD; otherwise, repeat step 2.

Table 4-3 shows all the results of FSD reduction for the recorded 5 patients' EEG's. After training the ASNN for the first time, the average recall FSD rate dropped to 3.92/hr. After the ASNN was retrained for the fourth time, the average recall FSD rate dropped to zero. For example, in the recorded EEG from Patient 2, the initial result shows that the FSD was 15.20/hr. Then one epoch of the FSD was selected from the initial results and added to the non-seizure EEG training set. After the network was re-trained to recall the patient's EEG, the FSD rate was quickly reduced to 8.14/hr. The above steps were repeated again and the FSD rate was reduced to zero. This means a perfect recall of EEG FSD was obtained.

However, the FSD of Patient 4 seems a more difficult one. After the initial results of FSD, this patient's recall FSD rate was 8.4/hr. When one epoch of the FSD was selected and added to the non-seizure EEG training set, its recall FSD rate was up to 9.78/hr. After the second re-training of the ASNN, its recall FSD rate was still 8.89/hr. Only at the forth time of re-training did the recall FSD rate drop to zero. This is an exceptional case because not all of the patients' EEGs lend themselves to low FSD. From the viewpoint of clinical application, only true seizures should be detected and recorded by the computer. This is the ideal situation which physicians want to have. Because of the different behavior of patients, some patients' EEG's may require a short period of training for the network to reach on FSD rate of zero, while other patients' EEG may require a long time for training (more patient EEGs are needed to be verified). This problem may easily be controlled in clinical practice by setting up a satisfied threshold level which steps the network training process when necessary. Figure 4.12 to Figure 4.16 show the re-training behavior for the reduction of FSD.

Currently, the selection of epochs for the reduction of FSDs is processed by observers but this work can easily be automated by computer techniques in the future.

Table 4-3: The reduction of the FSD rate after retraining

The number of Patient	Initial FSD	1^{st} Time Retraining	2^{nd} Time Retraining	3^{rd} Time Retraining	4^{th} Time Retraining
#1	3.4	1.7	1.7	1.13	0
#2	15.2	8.14	0	0	0
#3	8.18	0	0	0	0
#4	8.4	9.78	8.89	8.89	0
#5	0.12	0	0	0	0
Average	7.06	3.92	2.12	2.00	0

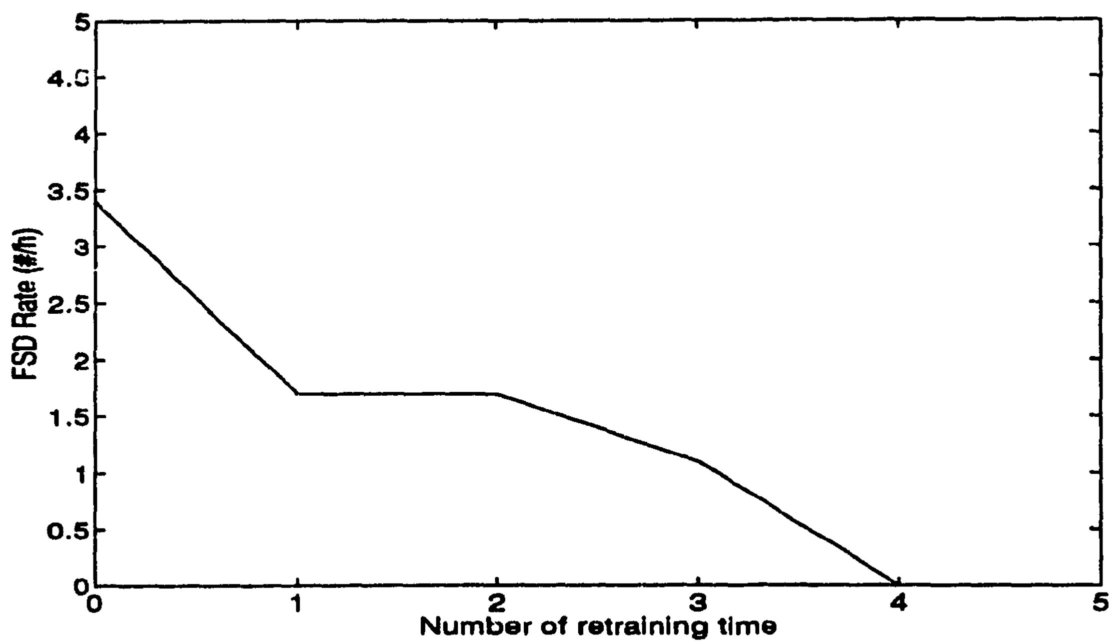

Figure 4.12: The re-training behavior for reduction of FSD (Patient #1)

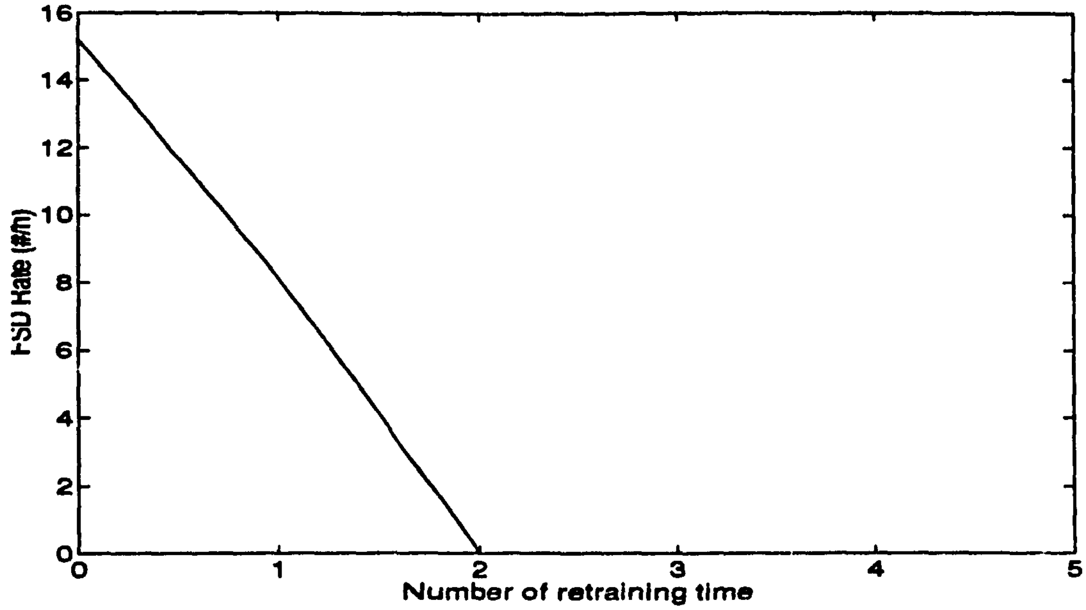

Figure 4.13: The re-training behavior for reduction of FSD (Patient #2)

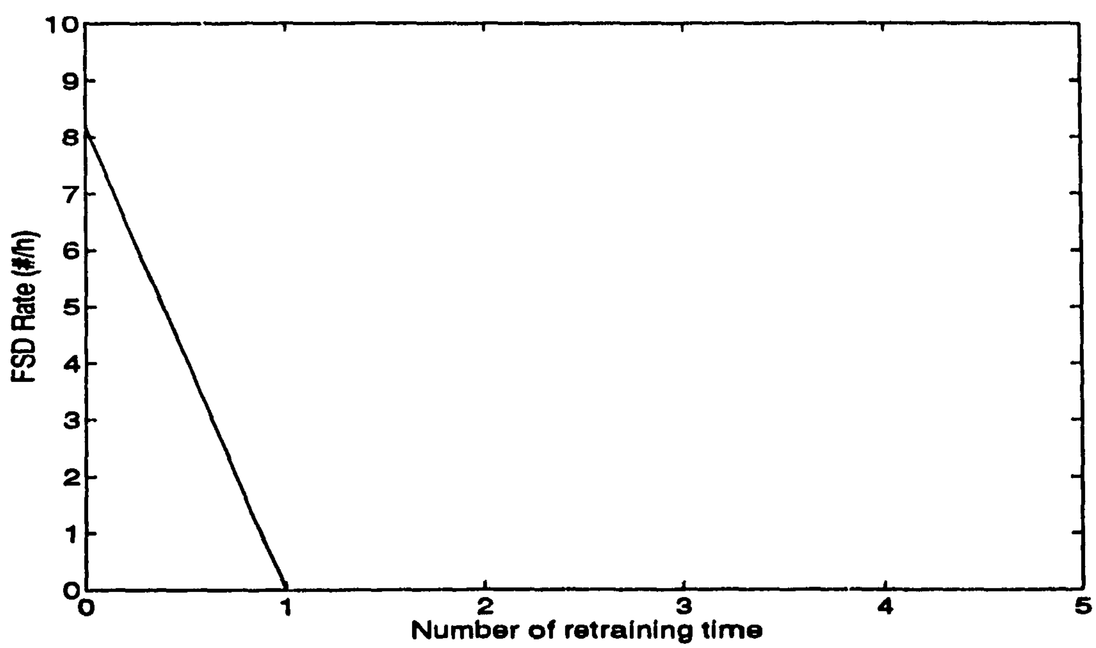

Figure 4.14: The re-training behavior for reduction of FSD (Patient #3)

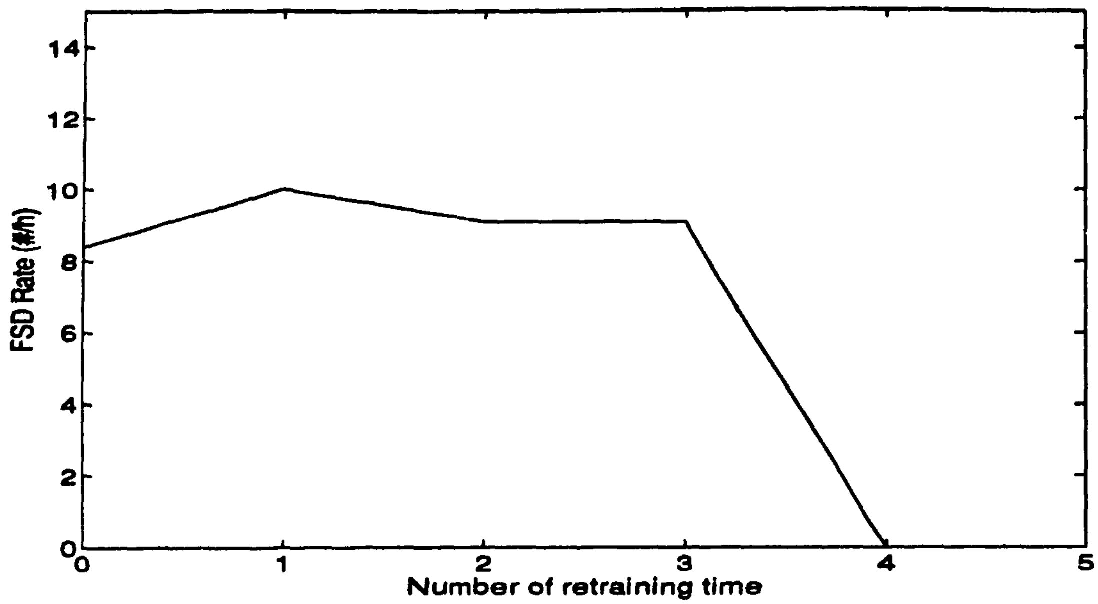

Figure 4.15: The re-training behavior for reduction of FSD (Patient #4)

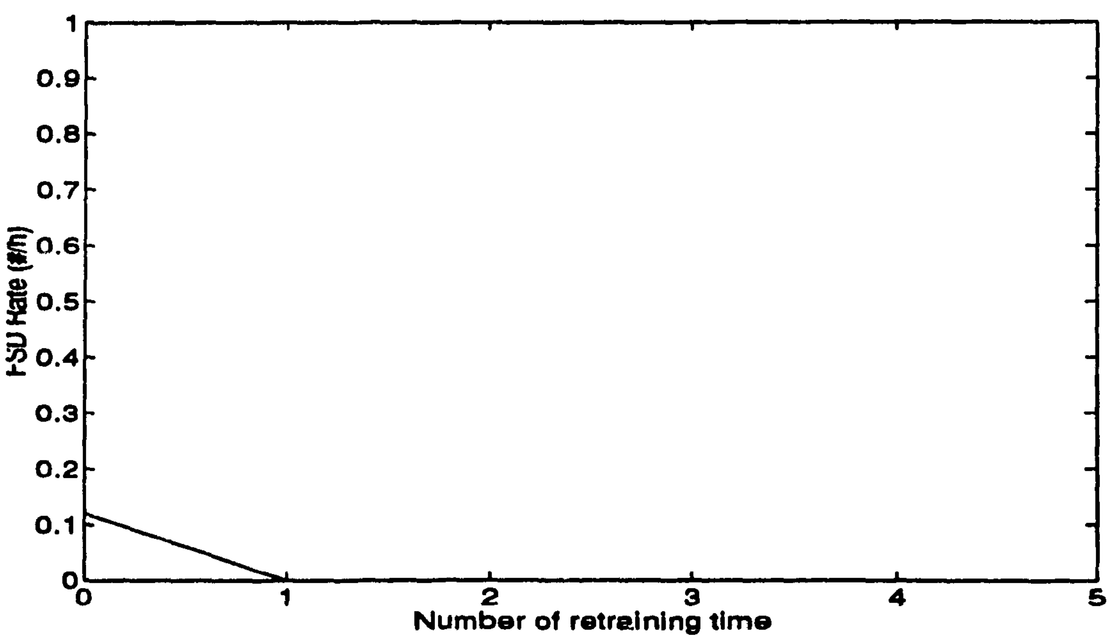

Figure 4.16: The re-training behavior for reduction of FSD (Patient #5)

4.4.3.3 Missing true seizures while the reduction of FSD was performed

This topic is arisen in the reduction of FSD. As a result of the ASNN method and its application to the EEG analysis, a significant improvement for FSD has been achieved. At the same time, we also noticed that the size of non-seizure EEG training set was increasing. This will raise the question as how many true seizures could be missed and whether the missing number of true seizures is beyond our tolerance when the non-seizure training set is changed. From the results of Table 4-4, it is found that 37 out of 39 seizures are still correctly recognized. More accurately, only the EEG file 2 of patient 4 missed two real seizures. The missing rate for true seizures is 5.1%. Figure 4.17 and Figure 4.18 show the EEG with missing seizures. It is not surprising that we find these two missing seizures are completely mixed up with the EEG background. In other words, the more the size of the non-seizure EEG training set, the more risk of missing true seizure. Although the missing rate of true seizures is 5.1%, we have a dramatic deduction of the FSD. This is still worthy in the clinical applications.

Table 4-4: Summary for missing true seizure

Patient Number	The number of true seizures	The number of missing seizures	The missing rate (%)
#1	6	0	0
#2	7	0	0
#3	9	2	22
#4	5	0	0
#5	12	0	0
Total Number	39	2	5.1

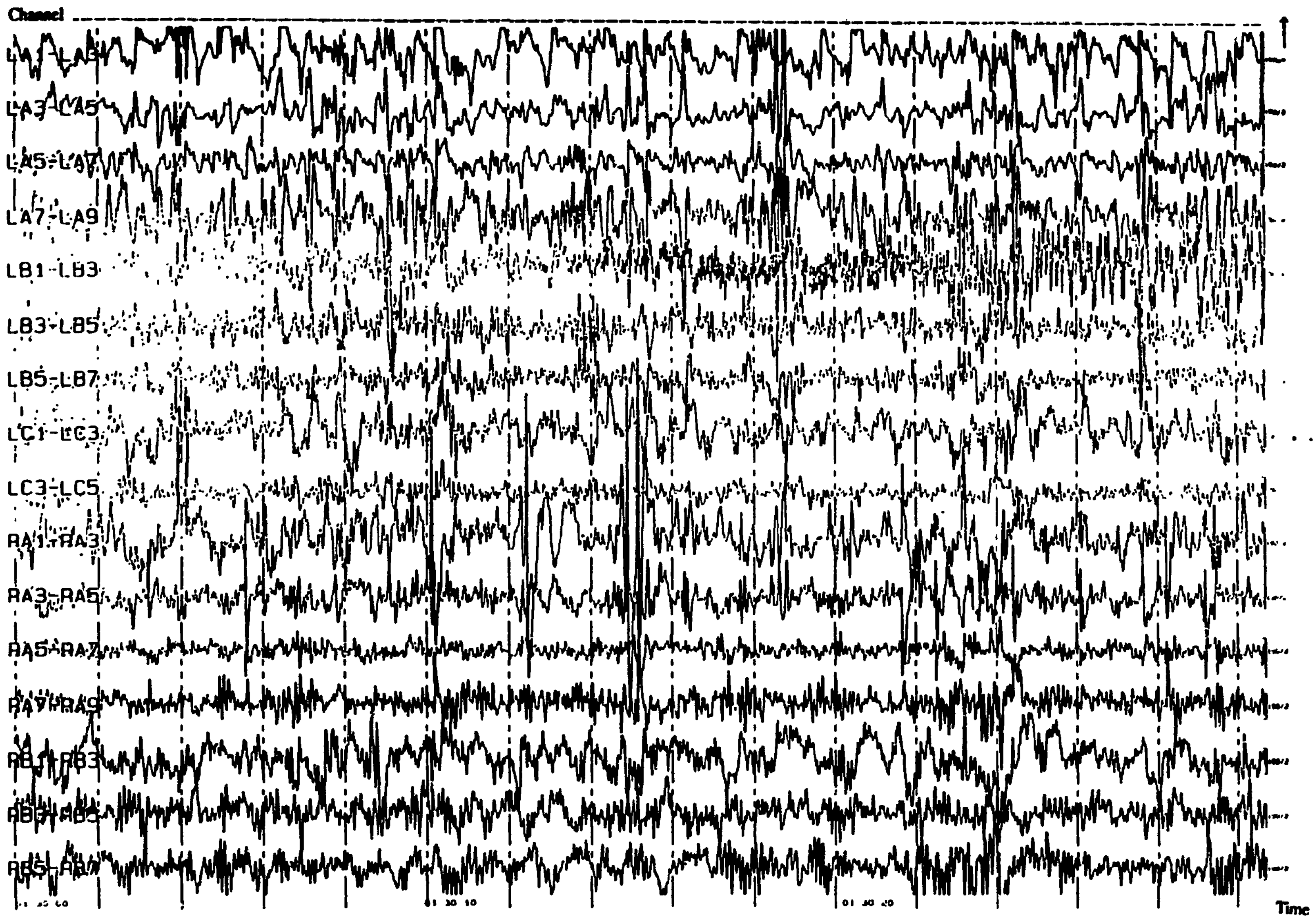

Figure 4.17: The missing one of true seizures (#1)

Figure 4.18: The missing one of true seizures (#2)

4.5 Summary

Long term EEG monitoring of epileptic seizures is significant for clinical purposes, especially to those patients without behavior information. Seizures, whether their behavior is evident or not, are very important to doctors for drug control and surgical operations.

This is the first effort to apply adaptive structure neural networks to the detection of EEG seizures. From the point of view for the clinical use of EEG's, two goals have been achieved. The first is to maximize the detection rate of true seizures. The second is to maximize the reduction rate of FSD and to minimize the missing of true seizures at the same time.

Five patients with thirty-eight recorded seizures from MNI were involved in the research. All the recorded seizures have been correctly recalled by the ASNN. This reveals that the ASNN is capable of addressing such a difficult problem. The sample of thirty-eight seizures is representative of wide variety which include local seizures, more generalized seizures, seizures with clear EEG background and seizures mixed up with EEG background. Since some patients' seizure patterns are mixed up with EEG background, it made our goals harder to reach.

In practice, missing a few clinical minor seizures in exchanges for a large reduction of FSD is still worthy because the majority of true seizures can be caught from the same patient; and it is good enough to help doctors to give a correct judgement. The large reduction of FSD will bring us much more benefits.

For the EEG application, the network performances using the ASNN and the BP algorithms are discussed in this chapter. The effectiveness and consistency of the ASNN were tested by using different initial hidden layer neurons and satisfying results were obtained. It means that the proposed neural networks structure is superior to fixed neural network structures. The ASNN can be used for solving difficult practical problems.

CHAPTER 5

Conclusion and Future Work

Artificial Neural Networks is an advanced technology developed from simulating the central neural system of the human brain. During recent years, various advanced algorithms such as Back-Propagation, Hopfield etc., have been built for practical problems. The Back-propagation algorithm has been widely used in pattern recognition, imaging processing, robotics and control, as well as other application areas.

5.1 Contributions

In the this thesis, the author has focused on research in neural networks structure design; proposed a new algorithm for structure level adaptation of neural networks and applied it to solve real-world application problems.The main contributions of this thesis can be summarized as follows:

1. Defined a systematic neuron generation rule
2. Proposed a multi-neuron generation method
3. Proposed a method called Delta Adaptation developed to speed up training time
4. Verified the ASNN through four simulations
5. Applied the new ASNN successfully to EEG seizure detections and reduce false detection rate

5. 1.1 Defined a systematic neuron generation rule

Several published results [30]-[32] have been discussed in chapter 2 for the automatic neuron generation. In this research, the neuron to be split is defined as the neuron (1) whose input weight has the largest fluctuation compared with other neurons in the same hidden layer and (2) whose output fluctuation in the hidden layer is also high. To determine the neuron with the highest fluctuation rate, the current and the most recent iterations are considered since they contain the most relevant historical information. Two advantages are found in this algorithm compared to other algorithms. First, the neuron generation rule successfully localizes the neuron which needs to be split during the network training. Second, the definition of the neuron generation rule is easily implementable.

5.1.2 Proposed a multi-neuron generation method

Through the results of simulation examples, the author found that additional "mother" neurons can be split at one time to improve the network performance. Hence the Maximum- Minimum Rule (MMR) is proposed to adapt the ASNN. Whenever neurons are split, the weights of the new neurons are the same as the weights of the mother neuron. MMR has been verified to be effective through various simulation results and is applied to automatic EEG seizure detection.

5.1.3 Delta Adaptation improvement

In the traditional BP delta algorithm, the learning rate is fixed. Generally after long network training iterations, the value of system error may become very small but remain above the desired level of error. Thus the network convergence rate will remain low. To overcome this problem, the author proposed a delta adaptation algorithm. A detailed description of this new method is explained in chapter 2. The principle behind the idea is the same as that of the weights update algorithm. To speed up the convergence of the network, δ power factor λ, δ step γ and δ momentum σ were adapted.

5.1.4 Benchmark simulation results

Two discrete and two analog examples were selected to test the proposed ASNN. Character recognition and XOR problems were chosen to represent the discrete examples. One spiral and 4-leafs-rose problems were used for analog examples. In general, using the ASNN, 30% to 80% of training iterations can be saved compared to the BP and Lee's algorithms.

5.1.5 Solving practical problems

The ASNN was applied to the Electroencephalogram (EEG) problem. Five patients' EEG with clinical seizures or without seizures, were recorded for 780 hours and both ASNN and BP were applied to explore the following three points:

1. Comparison of seizure detections between physician observation and ASNN
2. Comparison of seizure detections between ASNN and BP
3. Reduction of False Seizure Detections (FSD) during long term seizure monitoring.

First, all the seizures were correctly detected by using ASNN compared to observation by physicians (Appendix I). Second, when ASNN and BP algorithms were used for EEG seizure detections (Appendix II),60% to 75% of training iterations were saved by using ASNN. Even when the initial hidden layer neurons in the BP network setup were the same as the number of ASNN final hidden layer neurons, the ASNN still improved the training iterations by 30% to 60%. Third, the ASNN is verified as effective for seizure detections during long term monitoring (Appendix III). A retraining method of seizure detections was developed to maximize the reduction rate of False Seizure Detections (FSD) and to minimize the missing of true seizures. The results have shown that the FSD rate can be decreased to zero while the maximum of missing true seizures is only 5.1% These results are quite encouraging.

5.2 Future Work

While these results are encouraging, there is considerable room for improvement. Some further work need to be done for the enhancement of ASNN.

5.2.1 Testing of multi-hidden layers

Currently, ASNN algorithm is only applied to one hidden layer neural networks. As discussed in [3],[4], multi-hidden layers could more efficiently solve practical problems. Since ASNN is developed as a complete neuron generation technique, more complicated network structures could be realized. This is an important generalization of the present scheme.

5.2.2 Research on neuron deletion (pruning)

In this thesis, a neuron generation rule is developed to optimize network structure. In order to avoid redundant neurons, the number of initial hidden layer neurons will be small. This may slow down network performance because one can not predict how large a network structure should be to handle a practical problem. For example, in some cases more than 10 new neurons are added in EEG seizure delectation algorithm. If a neuron deletion rule is developed, then the number of neurons in the hidden layer can start with a larger value. When new neurons are needed for the network structure, the neuron generation rule can take it over automatically; when redundant neurons are found, the deletion rule can handle it automatically. This will further improve the network performance.

5.2.3 Large EEG data testing

The ASNN has been applied to EEG seizure detection and the results obtained are quite satisfactory. Although these 5 patients' EEGs have covered many EEG seizure patterns and EEG non-seizure patterns, more patients' EEGs are still needed to fully confirm the applicability of the proposed research. This work is valuable to physicians for localizing the position of seizures and may even be used in the hospital computerized long term monitoring systems.

5.2.4 Programming selection of non-seizure training set

The selection of EEG epochs to be included in the non-seizure training set for reducing the FSD is currently being processed by an observer. This process should be made automatic by further research on network training.

Bibliography

[1] Philip D. Wasserman, *Neural Computing Theory and Practice*, New York: Van Nostrand Reinhold, 1989.

[2] R. Hecht-Nielsen, *The Backpropagation Neural Network*. Reading, Massachusetts: Addision-Wesley Publishing Company, 1990. Neurocomputing.

[3] J. Anderson and E. Rosenfeld, eds., *Neurocomputing*: Foundations of Research, Cambridge: MIT Press, 1988.

[4] Minisky, Marvin and Papert, Seymour, *Perceptrons*, Cambridge MA: MIT Press, 1969.

[5] Y. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, Massachusetts: Addison-Wesley Publishing Company Inc., 1989.

[6] Imai Katsuji, Gouhara Kazutosh and Uchikawa Yoskiki, "Pattern Extraction and Recognition for Noisy Images Using the Thre -Layered BP Model", Proceedings of International Joint Conference on Neural Networks, pp.262-267, Sigapore,1991.

[7] Adrew J. Gaobor and Masudseyal " Automated interictal EEG spike detection using artificial neural networks", Electroencephalography and clinical Neurophysiology, Vol.83, pp. 271-280, 1992.

[8] B. Apolloni, G. Avanzini, "Diagnosis of Epilepsy via Back-propagation", Proceedings of International Joint Conference on Neural Networks, Vol. II, pp.571-574, Washington Dc, 1990.

[9] V. Zeman, R. V. Patel and K. Khorasani, "Control of a Flexible-Joint Robot Using

Neural Networks", submitted to IEEE Transaction on Neural Networks, 1994.

[10] Stanislaw H. Zak,"Robust Tracking Control of Dynamic Systems with Neural Networks", Proceedings of International Joint Conference on Neural Networks, Vol.II, pp.563-566, Washington Dc, 1990.

[11] Tsu-Chang Lee, *Structure level adaptation for artificial neural networks*, Boston, MA Academic Kluwer Publishers, 1991.

[12] Mahmood R. Azimi-Sadjadi, S.Sheedvash and F.O.Teujillo "Recursive dynamic node creation in multilayer neural networks", IEEE Transactions on Neural Networks, Vol. 4, No. 2, pp.242-256, March 1993.

[13] Scott E.Fahlman and Christian Lebiere "The Cascade-Correlation Learning Architecture", CMU Internal Report, 1991.

[14] P. Werbos Beyond regression "New Tools for prediction and Analysis in the Behavioral Sciences", PhD Thesis, Harvard, Camgridge, NA. August 1974.

[15] D. B. parker "learning Logic", Technical Report TR-47, Center for Computational Research in Economics and management Science, MIT, Camgridge, MA, April 1985.

[16] J. McClelland and D. Rumethart, Parallel Distribution Processing, Volumes 1 and 2. MIT Press, Cambridge, MA, 1986.

[17] James A. Freeman and David M. Skapura, *Neural Networks Algorithm, Applications, and Programming Techniques*, Addison-Wesley Publishing Co. 1991.

[18] Scott E.Fahlman "Faster - Learning Variation on Back Propagation: an Emprical

study" in Proceedings of the 1988 Cone ionist Models Summer School, Morgan Kaufmann

[19] K. J. Cios, and N. Lin, "A machine learning method for generation of a neural network architecture: a continues ID3 algorithm", IEEE Transaction on Neural Networks, vol. 3, No. 2, pp. 280-290, March 1992.

[20] Markus Hoehfeld and Scott E. Fahlman, "learning with limited Numerical Precision Using the Cascade-Correclation Algorithm", IEEE Transaction on Neural Networks, vol. 3, No. 4, pp. 602-611, July 1992.

[21] Shashi Shekhar, Minesh B. Amin, and Prashant Khandelwal, " Generalization Performance of Feed-Forward Neural networks ", Neural Networks: Advances and Applications, 2 E. Gelenbe(Editor), pp. 13-38, Elsevier Science Publishers B.V., 1992.

[22] Hasanat M. Dewan and Eduardo D. Sontag, "Extrapolatory Method for Speeding Up the BP Algorithm", Proceedings of International Joint Conference on Neural Networks, Vol. II, pp. 613-616, Washington Dc, 1990.

[23] Chih-Liang Chen and Roy S. Nutter, "Improving the training Speed of Three-Layer Feedforward Neural Nets by Optimal Estimation of the Initial Weights", Proceedings of International Joint Conference on Neural Networks, pp.2063-2068, Sigapore,1991.

[24] Y. K. Kim and J. B. Ra, " Weight Value Initiation for Improving Training Speed in the Back-propagation Network, "Proceedings of International Joint Conference on Neural Networks, pp.2396-2401, Sigapore,1991.

[25] Yoshon Izui and Alex Pentland, "Speeding Up Back Propagation ", Proceedings of International Joint Conference on Neural Networks, Vol. II, pp.639-642, Washington Dc, 1990.

[26] Richard P. Brent, "Fast Training Algorithms for multilayer Neural Nets", IEEE Transactions on Neural Networks Vol.2, No.3, pp. 346-354, May 1991.

[27] Masafumi Hagiwra, "Accelerated Back-propagation using unlearning based on Hebb Rule",Proceedings of International Joint Conference on Neural Networks, Vol. I, pp.617-620, Washington Dc, 1990.

[28] Wong F. S.,"FASTPROP: A selective training algorithm for fast error propagation", Proceedings of International Joint Conference on Neural Networks, pp.2038-2043, Sigapore,1991.

[29] Watrour R. L., "Learning Algorithms for Connectionist Networks: Applied Gradient Methods of Nonliear Optimization", Proceedings of IEEE First International Conference on Neural networks, vol.2, pp. 619-627, San Diego, California, June 1987.

[30] K. Khorasani and Wei Weng "Structure Adaptations in Feed-Forward Neural Networks", Proceeding of IEEE International Conference on Neural networks, Orlando, June 1994.

[31] Wei Weng, K. Khorasani and L. Y. Wang, "Optimal Structure Adaptation Neural Network Algorithm and Its Applications to EEG Seizure Detection", Proceeding of First International Conference on Electronics and Information Technology, Beijing, August 1994.

[32] Wei Weng and K. Khorasani, "Adaptive Structure Neural Networks with Applications to EEG Automatic Seizure Detection", submitted to IEEE Transactions on

Neural Networks, 1994.

[33] Pei-Chen Lo, Jose and C. Principle, "Dimensional Analysis of EEG Segments: experimental consideration", Proceedings of International Joint Conference on Neural Networks, Vol. I, pp.693-698, Washington Dc, 1989.

[34] Niedermeyer and Fernando Lopes da Silva, *Electroencephalography*, Ernst,1987.

[35] W. G. Water, " Automatic Low Frequency Analyzer ", Electromic Engineering, vol. 16, 1943.

[36] J. W. Cooky and J. W. Tukey " An algorithm for the machine computation of Complex Fourier Series", Math. Comp. Vol.19, pp. 297-230, Apr. 1965.

[37] B. H. Jansen etal, "Identification and labeling of EEG graphic element using autogressive spectral estimates", Comput. Biol. Med. vol. 12, 1982.

[38] Gabor Jando, Ralph M. Siegel, Zsolt Horvath and Gyurgy Buzsaki, "Pattern recognition of Electroencephalogram by artificial neural networks", Electroencephalography and clinical Neurophysiology, Vol. 86, pp. 100-109, 1993.

[39] Gotman, J. "Automatic recognition of epileptic seizure in the EEG" Electroencephalography and clinical Neurophysiology, Vol. 54, pp. 530-540, 1982.

[40] Murro, A.M., King, D. W., Smith, J.R., Gallagher, B.B., Flanigin, H.F. and Meador, K. "Computerized seizure detection of complex partial seizures" Electroencephalography and clinical Neurophysiology", Vol. 79, pp. 330-333, 1991.

[41] J. Gotman and Liyan Wang " State-dependent spike detection: Concepts and prelim inary results", Vol. 79, pp.11-19, 1991.

[42] Hao Qu and Jean Gotman " Improvement in seizure detection performance by automatic adaptation to the EEG of each patient ", Electroencephalography and clinical Neurophysiology, Vol. 86, pp. 79-87, 1993.

[43] Gotman J. "Seizure Recognition and Analysis", In : J. Gotman, J. R. Ives and P. Gloor (Eds), *Long-term Monitoring in Epilepsy*, Electroencephalography and clinical Neurophysiology, suppl. 37. Elsevier, Amsterdam, pp. 133-145, 1985.

Appendix I

Comparison between physician observation and ASNN for EEG seizure detections

Table 4-A

Patient Number	Training set	Recall set	Observe Start time	N.N Start time	Observer End time	N.N End time
Patent #1	File 1-1	File 1-1	13:27:16	13:27:15.2	13:27:50	13:27:20
		File 1-2	18:22:40	18:22:42.6	18:23:22	18:23:23.6
		File 1-3	21:00:52	21:00:55	21:01:44	21:01:41.44
	File 2-1	File 2-1	18:26:28	18:26:27.2	18:27:52	18:26:35
		File 2-2	19:25:23	19:25:23.8	19:26:04	19:26:07.2
		File 2-3	07:19:03	07:19:04	07:19:42	07:19:45
Patient #2	File 1-1	File 1-1	07:26:58	07:26:55.2	07:29:44	07:29:29
		File 1-2	20:50:54	20:51:04	20:52:18	20:52:32.9
		File 1-3	04:10:41	04:10:43.2	04:12:28	04:12:25
	File 2-1	File 2-1	16:21:59	16:21:56	16:22:14	16:22:08.3
		File 2-2	00:35:33	00:35:32.4	00:35:45	00:36:26
		File 2-3	04:32:42	04:32:42	04:32:51	04:33:33.2
		File 2-4	12:28:00	12:27:03	12:28:15	12:28:02.8
Patient #3	File 1-1	File 1-1	16:14:23	16:14:23.4	16:16:04	16:16:35.9
		File 1-2	03:26:48	03:26:48.6	03:28:29	03:28:29.8
		File 1-3	05:29:11	05:29:11.1	05:30:47	05:30:51
		File 1-4	13:09:06	13:09:06.7	13:10:21	13:10:21
		File 1-5	18:51:40	18:51:41	18:52:15	18:52:18.7

Patient Number	Training set	Recall set	Observe Start time	N.N Start time	Observer End time	N.N End time
Patient #4	File 1-1	File 1-1	20:04:24	20:04:24.9	20:04:38	20:04:35.2
		File 1-2	10:37:27	10:37:41	10:38:00	10:38:09.2
		File 1-3	14:12:38	14:12:34.2	14:13:36	14:13:36.2
		File 1-4	16:51:00	16:51:06	16:51:24	16:51:35.4
	File 2-1	File 2-1	05:00:26	05:00:26.2	05:00:40	05:00:40.9
		File 2-2	00:37:58	00:38:32.5	00:38:06	00:38:59.5
		File 2-3	01:30:16	01:30:15.7	01:30:28	01:30:33.6
		File 2-4	02:49:04	02:48:11	02:49:30	02:50:1.9
Patient #5	File 1-1	File 1-1	06:44:43	06:44:48	06:45:27	06:455:14
		File 1-2	08:09:40	08:09:30.6	08:11:10	08:11:24.4
		File 1-3	20:12:30	20:12:12.4	20:13:36	20:13:10
		File 1-4	22:11:10	20:10:58	22:12:12	22:12:09
		File 1-5	11:21:06	11:21:17	11:22:26	11:22:14
	File 2-1	File 2-1	03:10:06	03:09:57.3	03:11:06	03:12:26
		File 2-2	03:32:58	03:32:45	03:33:50	03:33:51.6
		File 2-3	11:24:06	11:23:46.2	11:24:55	11:24:45.3
		File 2-4	10:04:46	10:04:36	10:05:44	10:05:44.6
	File 3-1	File 3-1	14:05:02	14:04:58	14:05:31	14:05:34.9
		File 3-2	16:59:06	16:59:05.6	16:59:40	16:59:41.5
		File 3-3	08:57:08	08:57:08	08:57:46	08:57:48.6

Appendix II

The performance comparison between ASNN and BP networks

Table 4-B(a)

Patient number	Training Set	Recall Set	BP Fixed Hidd N=2	ASNN Start Hidd N=2		BP Fixed N=ASNN' final	Improved rate% Same (initial N)	Improved rate% Same (Final N)
			Iterations	Iterations	Added N	iteration		
#1	File 1-1	File 1	9274	2665	4	11396	71.3	76.6
	File 2-1	File 2	15168	3729	7	12617	75.4	76.6
#2	File 1-1	File 1	11908	5475	4	10982	74.4	50.1
	File 2-1	File 2	19695	6615	4	18545	72.2	64.3
#3	File 1-1	File 1	21405	9147	4	15754	57.3	41.9
#4	File 1-1	File 1	16031	4855	4	14704	69.7	67.0
	File 2-1	File 2	21958	5518	5	16628	74.8	66.8
#5	File 1-1	File 1	15192	4128	5	16315	72.8	74.7
	File 2-1	File 2	19494	3868	4	10750	80.5	64.0
	File 3-1	File 3	20026	3460	5	17060	82.7	79.7

Table 4-B(b)

Patient number	Training Set	Recall Set	BP Fixed Hidd N=3	ASNN Start Hidd N=3		BP Fixed N=ASNN' final	Improved rate% Same (initial N)	Improved rate% Same (Final N)
			Iterations	Iterations	Added N	iteration		
#1	File 1-1	File 1	12325	1818	9	7095	85.2	74.4
	File 2-1	File 2	13793	3424	4	12604	75.2	72.8
#2	File 1-1	File 1	12598	5293	4	8286	60.0	36.1
	File 2-1	File 2	17820	5051	7	12043	71.7	58.1
#3	File 1-1	File 1	16860	6571	9	12892	61.0	49.0
#4	File 1-1	File 1	14704	2788	8	13894	81.0	80.0
	File 2-1	File 2	21598	6736	6	16628	68.8	59.5
#5	File 1-1	File 1	15289	3522	9	10620	80.0	66.8
	File 2-1	File 2	15126	4141	10	8787	72.6	52.9
	File 3-1	File 3	19067	3420	5	19741	82.0	82.7

Table 4-B(c)

Patient number	Training Set	Recall Set	BP Fixed Hidd N=4	ASNN Start Hidd N=4		BP Fixed N=ASNN' final	Improved rate% Same (initial N)	Improved rate% Same (Final N)
			Iterations	Iterations	Added N	iteration		
#1	File 1-1	File 1	9525	3119	4	6605	67.3	52.8
	File 2-1	File 2	14358	3280	8	14533	77.2	77.4
#2	File 1-1	File 1	9714	5303	4	8288	45.4	36.0
	File 2-1	File 2	18141	4829	7	12531	73.4	61.5
#3	File 1-1	File 1	15626	5251	4	15024	66.4	65.1
#4	File 1-1	File 1	15926	4557	4	13737	71.4	66.8
	File 2-1	File 2	19431	7128	4	17538	63.3	66.8
#5	File 1-1	File 1	12704	3496	8	10620	72.5	67.1
	File 2-1	File 2	12647	2826	8	10385	77.7	72.8
	File 3-1	File 3	19752	2896	6	17099	83.4	83.1

Table 4-B(d)

Patient number	Training Set	Recall Set	BP Fixed Hidd N=6	ASNN Start Hidd N=6		BP Fixed N=ASNN' final	Improved rate% Same (initial N)	Improved rate% Same (Final N)
			Iterations	Iterations	Added N	iteration		
#1	File 1-1	File 1	11393	2815	4	7217	75.3	61.0
	File 2-1	File 2	15538	4717	6	14533	69.6	67.5
#2	File 1-1	File 1	10982	4124	5	7744	62.5	46.7
	File 2-1	File 2	18545	5356	4	12043	71.1	55.5
#3	File 1-1	File 1	15754	5344	4	13707	65.2	61.0
#4	File 1-1	File 1	14704	3084	10	13227	79.1	76.7
	File 2-1	File 2	16278	3518	8	15217	78.4	76.9
#5	File 1-1	File 1	18178	3610	6	10620	80.1	66.0
	File 2-1	File 2	10750	3345	7	8787	68.9	61.9
	File 3-1	File 3	17204	7567	5	16507	56.0	54.2

Table 4-B(e)

Patient number	Training Set	Recall Set	BP Fixed Hidd N=8	ASNN Start Hidd N=8		BP Fixed N=ASNN' final	Improved rate% Same (initial N)	Improved rate% Same (Final N)
			Iterations	Iterations	Added N	iteration		
#1	File 1-1	File 1	6605	2106	6	6216	68.1	66.1
	File 2-1	File 2	11613	3639	4	14533	68.7	80.0
#2	File 1-1	File 1	8288	2432	8	7519	76.0	67.7
	File 2-1	File 2	13711	3219	8	12148	76.0	73.5
#3	File 1-1	File 1	15024	5353	7	12539	64.4	57.3
#4	File 1-1	File 1	13737	3935	4	14339	71.4	72.6
	File 2-1	File 2	17538	5091	8	14949	71.0	65.9
#5	File 1-1	File 1	13887	4028	5	9124	71.0	55.9
	File 2-1	File 2	10168	2897	7	8772	72.0	67.0
	File 3-1	File 3	19741	6110	4	12582	69.0	51.4

Appendix III

Initial FSD reduction during long term monitoring

Table 4-C

Patient's Number	Training Pattern	Recall Pattern	FSDs* Rate
Patient #1	File 1-1	File 1-15	3.4 / h
	File 2-1	File 1-16	
		File 1-17	
		File 2-20	
		File 2-21	
		File 2-22	
		File 2-26	
Patient #2	File 1-2	File 1-12	15.2 / h
	File 2-1	File 1-13	
		File 1-14	
		File 1-18	
		File 1-19	
Patient #3	File 1-1	File 0-1	8.18 / h
	File 1-3	File 0-9	
		File 1-7	
		File 2-0	
		File 2-2	
		File 2-3	
		File 2-4	
		File 2-5	
		File 2-8	
		File 2-9	
* FSDs--False Seizure Detections			

Patient's Number	Training Pattern	Recall Pattern	FSDs* Rate
Patient #4	File 2-2	File 0-2	
	File 3-2	File 0-3	
	File 3-3	File 0-4	8.4 / h
		File 0-5	
Patient #5	File 3-1	File 0-71	
	File 3-2	File 0-72	
	File 4-1	File 0-81	0.12 / h
	File 4-2	File 0-82	
	File 5-1	File 1-01	
	File 5-2	File 1-11	
		FIle 1-22	
		File 1-32	
		File 1-42	
		File 2-02	
		File 2-11	
			Total Rate=7.06/h
FSDs--False Seizure Detections			