بِسمِ اللهِ الرَّحمَنِ الرّحيمِ

(اللهُ نُورُ السَّمَواتِ وَالأرضِ مَثَلُ نُورِهِ كَمِشكَاةٍ فِيهَا مِصبَاحٌ المِصبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوكَبٌ دُرِّيٌّ يُوقَدُ مِن شَجَرَةٍ مُّبارَكَةٍ زَيتُونَةٍ لاشَرقِيَّةٍ وَلاغَربِيَّةٍ يَكَادُ زَيتُهَا يُضِيءُ وَلَو لَم تَمسَسهُ نَارُ نُّورٌ عَلَى نُورٍ يَهدِي اللهُ لِنُورِهِ مَن يَشَاءُ وَيَضرِبُ اللهُ الأَمثَال لِلنَّاسِ وَاللهُ بِكُلِّ شَيءٍ عَلِيمُ)(٣٥)

صَدَقَ اللهُ العَليِّ العَظِيم

سورة النور

الخلاصة

ان تقنية تحليل البيانات لها تطبيقات عدةوفمنها تطبيق تصنيف البيانات و تطبيق التمايز و تطبيق التعرف على الانماط او الانظمة ، وظيفة التمايز او التصنيف تحدث في مجال واسع ضمن فعاليات الانسان الحياتية، فهي في اشمل وصف، يمكن ان تغطي اي سياق يتم فيه اتخاذ القرار اعتمادا على معلومات متوفرة، فطريقة التصنيف او التمايز هي طريقة مألوفة لطريقة تكرارية في اتخاذ القرارات في مواقف جديدة.

المشروع يتضمن ألية لدائرة تصنيف الشبكات العصبية نوع (perceptron)، وبطبقة خفية واحدة ان أستخدام شبكة عصبية في التصنيف او تمييز الاشارات(الاشارات الدماغية EEG Signls) .

تم استخدام برنامج الماتلاب (MATLAB)، نسخة[(R2010a) 7.10] في تحليل و تصنيف الاشارات. تنفذ عملية صنع القرار في مراحلتين: عملية تطبيع الاشارة و مرحلة الشبكات العصبية الصناعية لغرض التصنيف للبيانات الداخلة،النتائج توضح بان الاداء بأستخدام الشبكة العصبية الصناعية (75.07630%)، وهذا الاداء الافضل قد تم بأستخدام زمن تنفيذ (epochs) اقل.

APPENDIX A

A) The updating eq. for the synaptic weight (w_{ij}) is as follow:

$$w_{ij,k} = w_{ij,k-1} + \Delta w_{ij} \qquad \text{(A1)}$$

$$\Delta w_{ij} = -\eta \frac{\partial E_k}{\partial w_{ij}}$$

$$\Delta w_{ij} = \eta(d_i - O_i)\, f'_i(net_i) y_j^T \qquad \text{(A2)}$$

Substitute eq (A2) in eq.(A1) yields:

$$w_{ij,k} = w_{ij,k-1} + \eta(d_i - O_i)\, f'_i(net_i) y_j^T \qquad \text{(A3)}$$

Where $w_{ij,k-1}$ & $w_{ij,k}$ are the synaptic weights before and after the updating respectively.For sigmoid activation function derivative can take following forms:

1) *Unipolar continuous sigmoid activation function*,the derivative can be written as:

$$f'(net_i) = O_i(1 - O_i) \qquad \text{(A4)}$$

<u>Proof:</u>

$$f(net_{i)} = \frac{1}{1+e^{(net_i)}} \xrightarrow{yields} f'(net_i) = \frac{\partial[f(net_i)]}{\partial net_i}$$

$$f'(net_i) = \frac{\left(1 + e^{(-net_i)}\right)(0) - (1)\left(e^{(-net_i)}\right)(-1)}{(1 + e^{(-net_i)})^2} = \frac{e^{(-net_i)}}{(1 + e^{(-net_i)})^2}$$

$$= \frac{1}{1+e^{(-net_i)}} \left[\frac{e^{(-net_i)}}{1+e^{(-net_i)}}\right] = \frac{1}{1+e^{(-net_i)}} \left[\frac{e^{(-net_i)}+1-1}{1+e^{(-net_i)}}\right]$$

$$= \left(\frac{1}{1+e^{(-net_i)}}\right)\left[\left(\frac{1+e^{(-net_i)}}{1+e^{(-net_i)}}\right) - \left(\frac{1}{1+e^{(-net_i)}}\right)\right] = \left(\frac{1}{1+e^{(-net_i)}}\right)\left[1 - \left(\frac{1}{1+e^{(-net_i)}}\right)\right]$$

As we know:

$$O_i = \frac{1}{1 + e^{(-net_i)}} \qquad \text{(A5)}$$

$$f'(net_i) = O_i\,(1 - O_i) \qquad \text{(A6)}$$

2) Bipolar continuous sigmoid activation function, derivative can be written as:

$$f'(net_i) = \frac{1}{2}\left(1 - {O_i}^2\right) \qquad \text{(A7)}$$

Proof:

The bipolar continuous sigmoid activation function is

$$f(net_i) = \frac{2}{1+e^{(-net_i)}} - 1 \text{ , (bipolar continous activation function)} \qquad \text{(A8)}$$

$$f'(net_i) = \frac{\partial[f(net_i)]}{\partial net_i} = \frac{\left[\left(1+e^{(-net_i)}\right)(0) - (2)\left(-e^{(-net_i)}\right)\right]}{(1+e^{(net_i)})^2} * \left[\frac{2}{2}\right]$$

$$= \frac{1}{2}\left[\frac{4e^{(-net_i)}}{\left(1+e^{(-net_i)}\right)^2}\right] = \frac{1}{2}\left[\frac{\left(4e^{(-net_i)}+1-1\right)}{\left(1+e^{(-net_i)}\right)^2}\right]$$

$$= \frac{1}{2}\left[\frac{\left(1+4e^{(-net_i)} - 1 + e^{(-2net_i)} - e^{(-2net_i)}\right)}{(1+e^{(-net_i)})^2}\right]$$

$$= \frac{1}{2}\left[\frac{\left(1+2e^{(-net_i)}+e^{(-2net_i)}-1+2e^{(-net_i)}-e^{(-2net_i)}\right)}{\left(1+e^{(-net_i)}\right)^2}\right]$$

$$= \frac{1}{2}\left[\frac{\left(\left(1+2e^{(-net_i)}+e^{(-2net_i)}\right) - \left(1-2e^{(-net_i)}+e^{(-2net_i)}\right)\right)}{(1+e^{(-net_i)})^2}\right]$$

$$= \frac{1}{2}\left[\frac{\left(1+e^{(-net_i)}\right)^2 - \left(1-e^{(-net_i)}\right)^2}{(1+e^{(-net_i)})^2}\right]$$

$$= \frac{1}{2}\left[\left(\frac{1+e^{(-net_i)}}{1+e^{(-net_i)}}\right)^2 - \left(\frac{1-e^{(-net_i)}}{1+e^{(-net_i)}}\right)^2\right] = \frac{1}{2}\left[1 - \left(\frac{1-e^{(-net_i)}}{1+e^{(-net_i)}}\right)^2\right]$$

$$= \frac{1}{2}\left[1 - \left(\frac{1-e^{(-net_i)}+1-1}{1+e^{(-net_i)}}\right)^2\right] = \frac{1}{2}\left[1 - \left(\frac{2-\left(1+e^{(-net_i)}\right)}{1+e^{(-net_i)}}\right)^2\right]$$

$$= \frac{1}{2}\left[1 - \left(\frac{2}{1+e^{(-net_i)}} - \left(\frac{1+e^{(-net_i)}}{1+e^{(-net_i)}}\right)\right)^2\right] = \frac{1}{2}\left[1 - \left(\frac{2}{1+e^{(-net_i)}} - 1\right)^2\right]$$

But :

$$O_i = \frac{2}{1+e^{(-net_i)}} - 1$$

For bipolar continuous sigmoid activation function the derivative is:

$$f'(net_i) = \frac{1}{2}\left(1 - {O_i}^2\right) \qquad \text{(A9)}$$

$$\Delta w_{ij} = \eta(d_i - O_i)\, f'_i(net_i) y_j^T$$

a) Then for unipolar continuous sigmoid activation eq.(A2) can be written as:

$$\Delta w_{ij} = \eta(d_i - O_i)\ O_i(1 - O_i)y_j^T \tag{A10}$$

The updating equation (eq.(A1)) for the synaptic weight (w_{ij}) as follow:

$$w_{ij,k} = w_{ij,k-1} + \eta(d_i - O_i)\ O_i(1 - O_i)\ y_j^T \tag{A11}$$

b) But for bipolar continuous sigmoid activation eq.(A2) can be written as:

$$\Delta w_{ij} = \frac{\eta}{2}\ (d_i - O_i)\,.\left(1 - {O_i}^2\right)y_j^T \tag{A12}$$

The updating equation (eq.(A1)) for the synaptic weight (w_{ij}) as follow:

$$w_{ij,k} = w_{ij,k-1} + \frac{\eta}{2}\ (d_i - O_i)\,.\left(1 - {O_i}^2\right)y_j^T \tag{A13}$$

B) Changes in weights of input-to- hidden units (Δv_{jl}), synaptic weights that connect j^{th} hidden unit by the l^{th} input from k^{th} input feature vector, are:

$$\Delta v_{jl} = -\eta \frac{\partial E_k}{\partial v_{jl}} \tag{A14}$$

For $j = 1,2,3 \ldots .n_h$, and $l = 1,2,3 \ldots ..n_i$

$$\frac{\partial E_k}{\partial v_{jl}} = \frac{\partial E_k}{\partial(net_j)} \cdot \frac{\partial(net_j)}{\partial v_{jl}} = \delta_j^y \cdot \frac{\partial(net_j)}{\partial v_{jl}} \tag{A15}$$

$$\delta_j^y \triangleq -\frac{\partial E_k}{\partial(net_j)} = -\frac{\partial E_k}{\partial y_j} \cdot \frac{\partial y_j}{\partial(net_j)} \tag{A16}$$

Where δ_j^y is referring to the error of j^{th} hidden unit from k^{th} input feature vector that needs to be back-propagated to correct the synaptic weights v_{jl}.

$$\frac{\partial E_k}{\partial y_j} = \frac{\partial\left(\frac{1}{2}\sum_{i=1}^{n_o}\left(d_i - f\left(net_i(y_j)\right)\right)^2\right)}{\partial y_j}$$

$$\frac{\partial E_k}{\partial y_j} = -\sum_{i=1}^{n_o}\left[\left(d_i - f(net_i)\right)\frac{\partial}{\partial y_j}\left(f\left(net_i(y_j)\right)\right)\right] \tag{A17}$$

But using the chain rule on $\left[\frac{\partial\left(f\left(net_i(y_j)\right)\right)}{\partial y_j}\right]$ to another form yields

$$\frac{\partial f(\mathrm{net_i})}{\partial y_j} = \frac{\partial f(\mathrm{net_i})}{\partial(\mathrm{net_i})} \cdot \frac{\partial(\mathrm{net_i})}{\partial y_j} \quad \text{(A18)}$$

$$\frac{\partial f(\mathrm{net_i})}{\partial y_j} = f'(\mathrm{net_i}) \cdot \frac{\partial(\mathrm{net_i})}{\partial y_j} \quad \text{(A19)}$$

$$\mathrm{net_i} = \sum_{j=1}^{n_h} w_{ij}\ y_j \quad \text{(A20)}$$

$$\frac{\partial(\mathrm{net_i})}{\partial y_j} = \frac{\partial}{\partial y_j}\left(\sum_{j=1}^{n_h} w_{ij}\ y_j\right)$$

$$\frac{\partial(\mathrm{net_i})}{\partial y_j} = \sum_{j=1}^{n_h} \frac{\partial y_j}{\partial y_j} \cdot w_{ij} = w_{ij} \quad \text{(A21)}$$

Substitute eq.(A21) in eq.(A19) then put the resulted form in eq.(A15) yields:

$$\frac{\partial E_k}{\partial y_j} = -\sum_{i=1}^{n_o} (d_i - O_i)\ f'(\mathrm{net_i}) \cdot w_{ij} \quad \text{(A22)}$$

$$\frac{\partial E_k}{\partial y_j} = -\sum_{i=1}^{n_o} \delta_i^o w_{ij} \quad \text{(A23)}$$

$y_j = f(\mathrm{net_j})$, then

$$\frac{\partial y_j}{\partial(\mathrm{net_j})} = f'(\mathrm{net_j}) \quad \text{(A24)}$$

Substitute both eq.(A23) & eq.(A24) in eq.(A16) yields :

$$\delta_j^y = \sum_{i=1}^{n_o} \delta_i^0 w_{ij}\ f'(\mathrm{net_j}) \quad \text{(A25)}$$

Or eq (A25) can be written as follow:

$$\delta_j^y = f'(\mathrm{net_j}) \sum_{i=1}^{n_o} \delta_i^0 w_{ij} \quad \text{(A26)}$$

$$\mathrm{net_j} = \sum_{l=1}^{n_i} v_{jl}\, x_l \quad \text{(A27)}$$

$$\frac{\partial(\mathrm{net_j})}{\partial v_{jl}} = x_l \quad \text{(A28)}$$

Substitute eq.(A26) & eq.(A28) in eq.(A15) yields:

$$\frac{\partial E_k}{\partial v_{jl}} = f'(net_j) \sum_{i=1}^{n_o} \delta_i^0 \, w_{ij} \cdot X \tag{A29}$$

Substitute eq.(A29) in eq.(A14) yields:

$$\Delta v_{ji} = \eta \, f'(net_j) \sum_{i=1}^{n_o} \delta_i^0 w_{ij} \cdot X \tag{A30}$$

a) for unipolar continuous activation over all nodes in the hidden layer

$$f'(net_j) = y_j(1 - y_j) \tag{A31}$$

Then eq(A30) can be written as:

$$\Delta v_{jl} = \eta \cdot ((y_j(1 - y_j)) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{A32}$$

$$v_{jl,k} = v_{jl,k-1} + \Delta v_{jl} \tag{A33}$$

$$v_{jl,k} = v_{jl,k-1} + \eta \cdot ((y_j(1 - y_j)) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{A34}$$

b) For bipolar continuous activation over all nodes in the hidden layer :

$$v_{jl,k} = v_{jl,k-1} + \Delta v_{jl}$$

$$\Delta v_{ji} = \eta \, f'(net_j) \sum_{i=1}^{n_o} \delta_i^0 w_{ij} \cdot X$$

The first derivative for bipolar continuous sigmoid activation function is:

$$f'(net_j) = \frac{1}{2}(1 - y_j^2) \tag{A35}$$

$$\Delta v_{jl} = \frac{\eta}{2} \cdot ((1 - y_j^2) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{A36}$$

$$v_{jl,k} = v_{jl,k-1} + \Delta v_{jl}$$

$$v_{jl,k} = v_{jl,k-1} + \frac{\eta}{2} \cdot ((1 - y_j^2) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{A37}$$

Appendix B

Step by Step Execution of

The ANN Classifier Algorithm

- The 1st step in the ANN classifier algorithm, all the parameters is initialized.
- Randomly select α the learning parameter & nitialize the Weights (W & V)

1) The feedforwarde phase of ANN

A) Update The Synaptic Weights***:***

For k = 1,2,.....,m (m: is the number of feature vectors)

For j = 1,2,....., n_h (n_h:is the number of hidden units)

%To find the response of hidden units

$$x_{0,k} = 1, \forall k$$

$$y_{j,k} = \sum_{l=0}^{n_i} v_{jl}\ x_{l,k}$$

$$\bar{y}_{j,k} = \bar{y}_{j,k} + \mathrm{sgm}\big(\ y_{j,k}\big)\ ,(\text{response of } j^{th} \text{ hidden unit})$$

For i=1,2,.....,n_o

$$\bar{y}_{0,k}\ = 1, \forall k$$

$$O_{i,k} = \sum_{j=0}^{n_h} w_{jl}\ x_{l,k}$$

$$\bar{O}_{i,k} = \bar{O}_{i,k} + \mathrm{sgm}\big(\ O_{i,k}\big)\ ,(\text{response of } j^{th} \text{ hidden unit})$$

End for

2) The ***Backward propagation phase*** of ANN

For i = 1,2,.....,n_o

%Calculate the output error

$$e^{o}_{i,k} = e^{o}_{i,k} + \left(d_{i,k} - \overline{O}_{i,k}\right)\overline{O}_{i,k}\,(1 - \overline{O}_{i,k})$$

End for

End for

%To Calculate the backpropagated errors error

For $j = 1,2,.....,n_h$

For $i = 1,2,.....,n_o$

$$w_{ij} = w_{ij} + w_{ij} e^{o}_{i,k}\,\bar{y}_{j,k}$$

End for

End for

End for

For $j = 1,2,.....,n_h$

$$e^{h}_{j,k} = e^{h}_{i,k} + \bar{y}_{j,k}\,(1 - \bar{y}_{j,k}) \sum_{i=0}^{n_i} e^{o}_{i,k}\; w_{ij}$$

End for

For $j = 1,2,.....,n_h$

For $l = 1,2,.....,n_i$

$$v_{jl} = v_{jl} + \alpha\, e^{h}_{j,k}\, x_{l,k}$$

End for

End for

End for

CHAPTER ONE

INTRODUCTION

1.1 General consideration

Data analysis is the process of gathering, modeling, and transforming of data with the goal of highlighting useful information ,suggesting conclusions, and supporting decision making[1]. The development of the science world, the amount of the information has been increased surprising such that clustering the data (or generally data analysis) has become very important concept to be discussed and developed.

Data analysis has several applications, such as classification, recognition, and system identification. As an example of **classification**, one can consider the problem of finding the neighbor countries for the certain neighborhood radius. Also, **system identification** significantly finding a practical model of unknown system, but all of these applications have a unique concept of ***clustering***, which is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense each other [1, 2]. This is done some with clustering algorithms which are called classifiers. A classifier divides a set of data into subsets such that for each subset, its members have similar structure features. Each classification algorithm uses certain method for training its initial structure to decrease the error gradually [2].

Classification is a form of pattern recognition and it has been the subject of many researches, both theoretical and experimental, in the last few decades. It is essentially an old problem, but one of the new techniques is constantly being used for an existing ones refined with the aim of extracting better form increasingly complex data, classification systems often begin with some kind of preprocessing to remove noise and redundancy in the measurements sharpening and smoothing by ensuring an effective and efficient signal description [3].

1.2 Motivation

Artificial Neural Networks (ANNs) are the most important and useful classifier in many data analysis applications, because of their memory and intelligibility, also they respond in practically a few times and get acceptable accurate results. ANN classifier is a the process of gathering, modeling, and transforming of data with the goal of highlighting useful information ,suggesting conclusions, and supporting decision making, The development of the science world, the amount of the information has been increased surprising such that clustering the data (or generally data analysis) has become very important concept to be discussed and developed.

1.3 Literature Survey

In 2003, D. Garrett, et al. [4] made a comparison of linear and nonlinear methods for EEG signal classification. They depend on the fact of the reliable operation of brain-computer interface based on EEG signal which requires accurate classification of multichannel.

In 2003, M. Usken and P. Stagge [5] presented a paper which focused on recurrent neural networks, their applications for time series applications.

In 2005, A. Saxena and A. Saad [6] presented a paper which introduces a certain type of artificial neural networks, which evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems.

In 2008, M. Mazurowski, et al. [7] presented a paper which focused on training neural network classifiers for medical decision making through study of the effects of imbalanced dataset on classification performance .

In 2011, A. R. Muslim [3] presented a three-dimensional data structure neural network for image recognition and classification, Training neural network classifier for images decision making through the study of the effects of dataset on classification performance.

1.4 Aims of project

- To use a multilayer Artificial Neural Network into classifier system based on sigmoid transfer function in hidden layer (using their crisp membership function)and and to show structure of ANN classifier.
- To show the enhancement in performance of ANN classifier through improve the ability to perform classification task using Iris data set as classification signal

1.5 project Organization

Chapter one serves as a background for the reader as well as an introduction to the subject of this work. A brief introduction to ANN and show the advantages and reasons to use ANN classifiers ccts.

Chapter two a background for Artificial Neural Network, ANNs is important for classification, provides like a signal with primal classification.

Chapter three focused on ANN as a classifier system, showing three main subsystems for any classification system using Iris data set signal, also, show the structure of ANN classifier, and the learning algorithm to learn such classifier.

Chapter four this chapter includes and exhibits results of ANN classifier experimental classifications process and their discussion.

Chapter five presents conclusions and the suggestions for future works.

CHAPTER TWO

ARTIFICIAL NEURAL NETWORKS

2.1 Introduction

Artificial Neural Networks (ANNs) are the most important and useful classifier in many data analysis applications, because of their memory and intelligibility, also they respond in practically a few times and get acceptable accurate results [2]. The utility of ANN lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical. The wide field of the applications on neural networks encourages the designers to develop neural network based algorithms. On the other hand, the great flexibility in merging with many other algorithms is another reason for neural networks to have high popularity among various classifiers [8].

Preprocessing aim to change the representation of a signal by use of a mathematical operation. It is possible also to decompose a complex problem into simpler ones for obtaining simpler solutions, also they play important role in different signal processing applications like filtering, pattern recognition, restoration, spectrum estimation, signal enhancement, localization and compression, the performance of each application depends on several factors, and hence, each application may need a different transforms technique for a better solution [3].

2.2 Natural and Artificial Neurons

Biologically, a simple neuron has four basic components: dendrites, soma (cell body), axon, and synapses [as shown in Fig. 2.1]. Dendrites are assumed as input channels which receive inputs (signals) from the synapses of the other neurons and send them to the soma. The *soma* processes the signals over time to transform the processed value as output and send it to other neurons through the axon and the synapses [2]

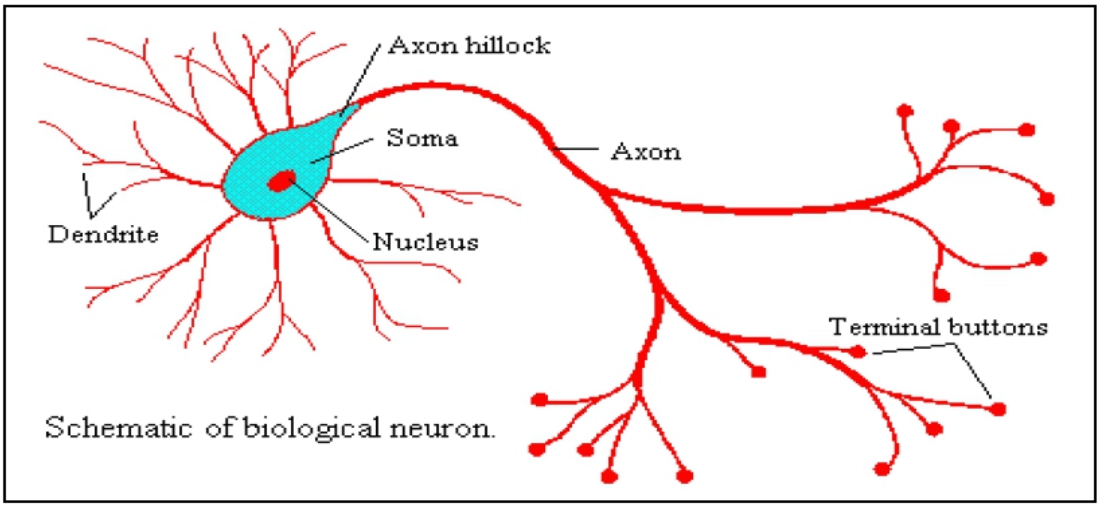

Fig 2.1: Illustrates the four parts of biological neuron [9]

In fact, the main objective of the ANNs researchers is to focus on understanding the neurons capabilities and how they can generate solutions depending on their behavior to solve complex problems which has not been solved by traditional computing or other methods [9].

A neural network is a massively parallel distributed processor that has natural propensity for storing knowledge and making it available for use. It resembles the brain in two respects:

1- Knowledge is acquired by the network through a learning process
2- Interconnection strengths between neurons, known as synaptic weights or weights, are used to store knowledge.

ANNs consist of a number of basic units called *neurons*. The artificial unit is often called artificial neuron perceptron, node, unit, or processing element. It is designed in a method to simulate the four natural neuron functions. As shown in Figure (2.2), the unit receives inputs from other units or external inputs X_i. These inputs are connected with other processing elements by weights W_{ji} . Each input is associated with different synaptic weights. The inputs are multiplied by the connection weights and the products are summed to obtain the ***net*** input. They are feed through a transform function (activation function), thus generating the neuron's output Y at the output layer [3].

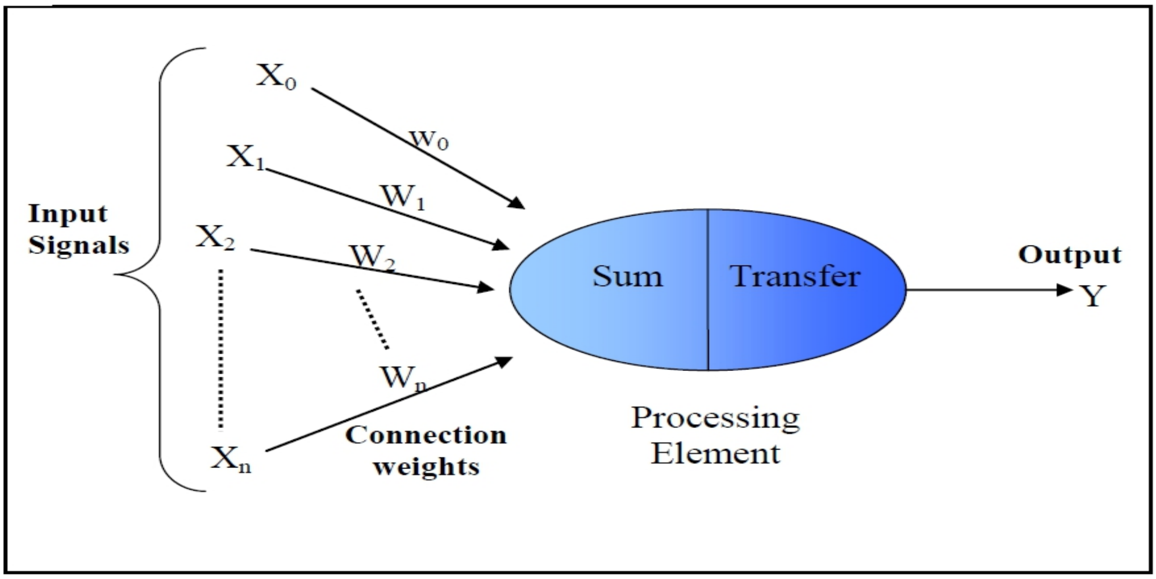

Fig 2.2: A network of single artificial neuron or perceptron [9]

It is worth noting that the hidden nodes and output units are considered as processing elements, while the input nodes are used to provide the data or signals to the neural network. From this point of view, it appears that hidden and output units use a “bias” or “threshold” value in computing the neurons output. A bias term can be considered as a connection weight with a constant input. Since the bias term has the ability to learn as other weights, it is connected as a single input to the hidden or output nodes, which require this term. [9].

Weights are the storing elements of a neural arrangement and contribute with the input to the output production. Therefore, the two basic operations of a NN are the learning process and the process of information retrieval with the use of the existing weights [2].

2.2.1 The Learning Process

Neural network can be classified according to learning is the process by which it acquires ability to carry out certain tasks by adjusting its internal parameter. It learns by repeatedly trying to match input data to the corresponding target value. After a sufficient number of iterations, the network creates an internal model that can be used to predict for new input condition. Learning can be performed in two modes [10]:-

2.2.1.1 Supervised Learning: which incorporates an external teacher, each output unit is told what its desired response to input signals ought to be required, and Figure (2.3) presents the supervised learning [10].

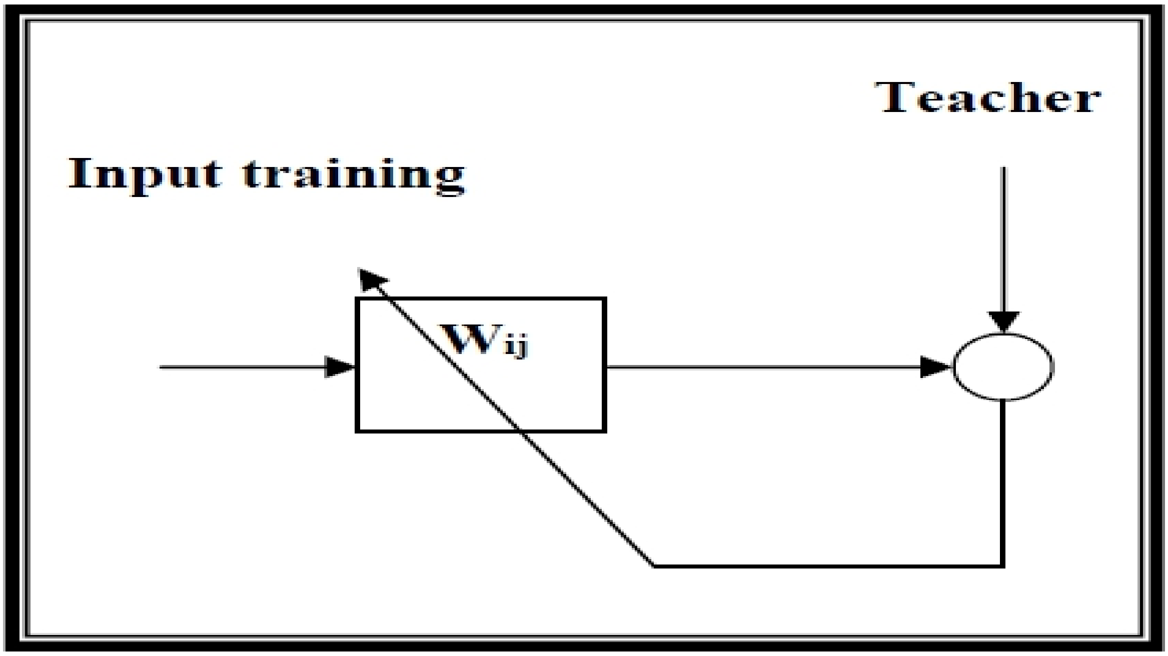

Fig2.3: Supervised Learning [10]

2.2.1.2 Unsupervised Learning: no external teacher and based upon only local information. It is also referred to as self-organization,(see Figure2.4)[10].

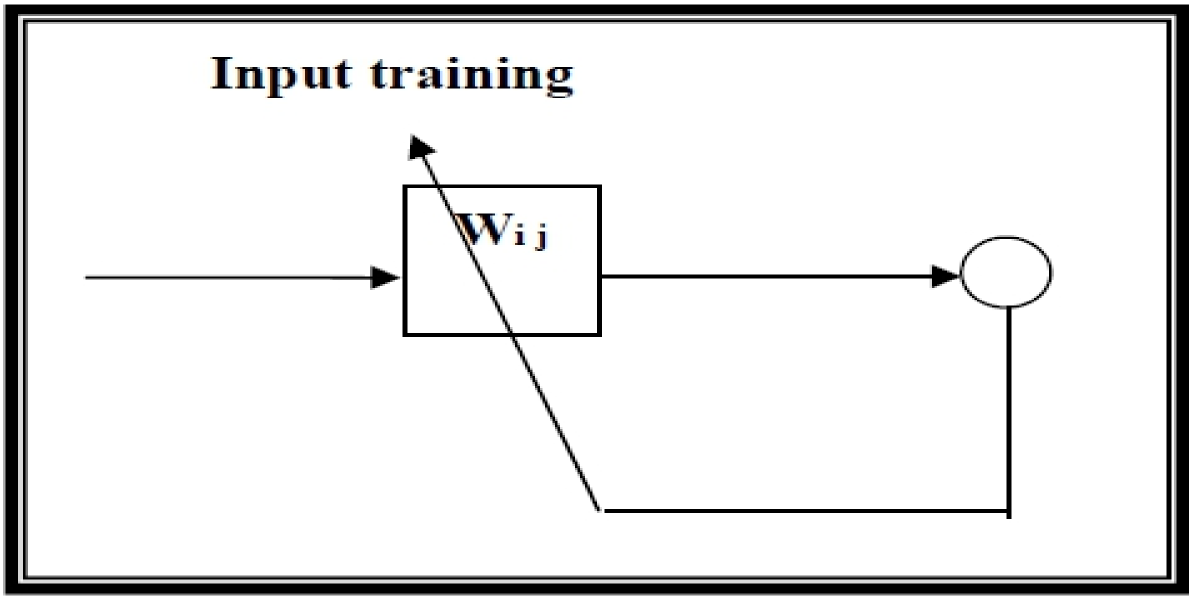

Fig2.4: Unsupervised Learning [10]

2.2.2 Models of Artificial Neural Networks

Neural networks have different types and these types have the simple clustering of the Artificial Neurons (ANs). Neurons grouped in one or many layers to form clusters and the layers are connected to each other. Their models have various methods of the interconnection between network layers. The ANs can be connected in different ways. There are two main types of neural network architectures shown as follow: [2, 11].

2.2.2.1 Feed-Forward Neural networks (FFNNs)

Feed-forward ANNs allow signals to travel one way only, from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Such neural class includes one or more layers of hidden units between the input and the output layer. The perceptron is a neuron with a transfer function and a weight adaptive mechanism (learning) by comparing the actual and the desired output for any input or stimulus.

Multilayer perceptrons are a type of FFNNs, which consists of a number of neurons grouped in layers. It has three types of layers which are: input layer, which sent the input signal (data) to the other layer in a forward direction, hidden layer transmits the signal (data) from the input nodes to the output nodes (also it enables the network to learn complex tasks) and output layer, which provides the actual response of ANN [9]. Figure (2.5) shows the structure of the MLP with one hidden layer.

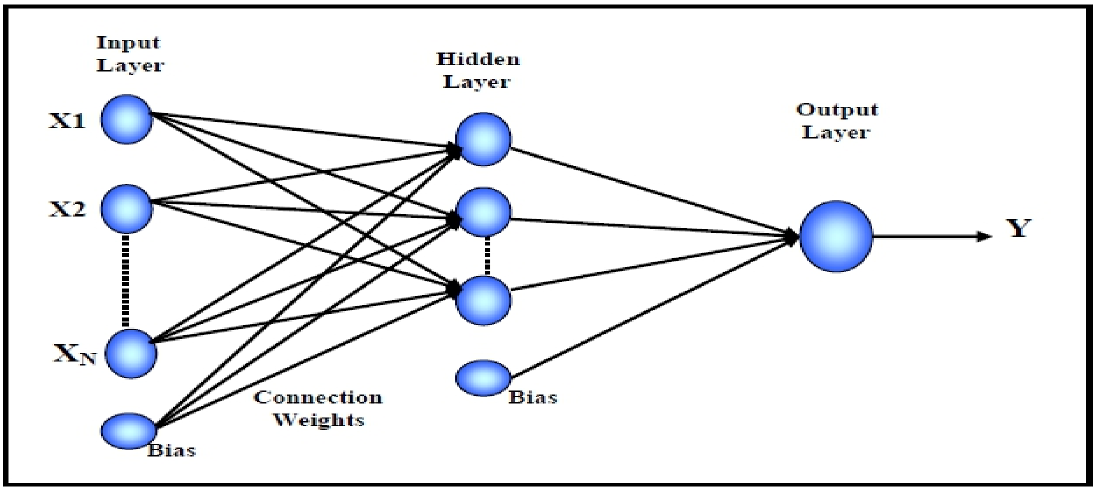

Fig2.5: Multilayer feed forward neural network [9].

The operation of the MLP network can be divided into two phases: training phase in which the MLP trained using training algorithms and retrieval phase that generates output by using the previously trained MLP networks. MLP networks are training using supervised algorithm with one hidden layer and sufficient number of hidden nodes (having the nonlinear transfer function) can produce feasible function to any desired rate of accuracy [10].

The output of the MLP network is calculated as follows: [5]

$$O_i = f\left(\sum_{j=1}^{n_h} w_{ij}\ f\left(\sum_{l=1}^{n_i} v_{jl} X_l + v_{j0}\right) + w_{i0}\right) \qquad (2.1)$$

Where (f) is a sigmoid transfer function, x_l is the input value, (v_{jl}) is the weights from the input layer to the hidden layer, (W_{ij}) are the weights from the hidden layer to the output layer, v_{j0} and w_{i0} are bias for hidden node and output node respectively, and O_i denotes the output of the i^{th} output unit . MLP network is considered as a fully connected network since every node is connected to all nodes in the next layer.

In this research work, MLP network with one hidden layer is considered.

2.2.2.2 **Recurrent Neural Networks (RNNs)**

Recurrent neural Networks (RNNs) or feed-back networks can have signals traveling in both directions by introducing loops in the network. Such networks are very powerful and can get extremely complicated, RNNs are dynamic; their state is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feed-back classes are also referred to as interactive or recurrent, although the latter term is often used to denote feed-back connections in single layer organization [2].

To overcome the limitation of feed forward networks, feedback connections in N have been introduced. In RNNs the past inputs can be cycled back and can impact the processing of future inputs, this allows the network to have information of the past behavior. RNN is utilized to enable the architecture of NN to learn a representation of time in data.

a) **Back Propagation Learning Algorithm**

Multi-layer networks have the ability to learn a suitable mapping from data set. Furthermore they have been used to solve complex problems using a supervised training with a selected algorithm which is the error back-propagation algorithm. Hence, the learning process that performed with the error back propagation algorithm is named back-propagation learning [30]. Learning iteration (epoch) of the neural network can be summarized into two phases:

1. **Feed forward**: the input vector is applied from input layer to the output layer. Thus, the actual output is produced. Consequently, all synaptic weights of the network are fixed through this phase [9].

2. **Backward propagation**: the synaptic weights of the network are adjusted. Then error propagated backward from the output layer to the input layer through the network, so that, this algorithm called "Error Back-Propagation" (EBP), the reason behind adjusting the synaptic weights to reduce the error between the desired and actual outputs [10].

b) **Training by Back-Propagation algorithm [11, 12]**

Firstly, the output is calculated and compared to the desired output (as shown in fig 2.5) for Multilayer perceptrons .The input vector to ANN (X_k) from k$^{\mathrm{th}}$ input feature vector:

$$\mathrm{X_k} = \left[\mathrm{x_{1,k}} \quad \mathrm{x_{2,k}} \quad \cdots \mathrm{x_{n_i,k}}\right]^{\mathrm{T}} \tag{2.2}$$

The responses at the output nodes from k$^{\mathrm{th}}$ input feature vector:

$$\mathrm{O_k} = \left[\mathrm{o_{1,k}} \;\; \mathrm{o_{2,k}} \dots\dots \mathrm{o_{n_o,k}}\right]^{\mathrm{T}} \tag{2.3}$$

The weight matrix between input and the hidden layer (V) is:-

$$\mathrm{V} = \begin{bmatrix} \mathrm{v_{1,1}} & \cdots & \mathrm{v_{1,n_i}} \\ \vdots & \ddots & \vdots \\ \mathrm{v_{n_h,1}} & \cdots & \mathrm{v_{n_h,n_i}} \end{bmatrix} \tag{2.4}$$

The weight matrix between hidden layer and output (W) is:-

$$W = \begin{bmatrix} w_{1,1} & \cdots & w_{1,n_h} \\ \vdots & \ddots & \vdots \\ w_{n_o,1} & \cdots & w_{n_o,n_h} \end{bmatrix} \quad (2.5)$$

The target or desired ($\mathbf{d_k}$) of the ANN from k^{th} input feature vector:

$$\mathbf{d}_k = [d_{1,k} \quad d_{2,k} \cdots\cdots \quad d_{n_o,k}]^T \quad (2.6)$$

Where:-

n_i: Total number of inputs.

n_h: Total number of hidden nodes at the hidden layer.

n_o: Total number of nodes at the output layer.

$$\mathbf{net_k} = V\,X_K \quad (2.7)$$

The responses of nodes of hidden layer from k^{th} input feature vector:

$$Y_K = \left[y_{1,k} \quad y_{2,k} \ldots\ldots y_{n_h,k}\right]^T \quad (2.8)$$

$$Y_K = f(\mathbf{net_k}) \quad (2.9)$$

$f(\,.\,)$: activation function that achieved by particular unit, many types of activation function such as linear and sigmoidal, depending on their processing effect of that neuron on inputted activation vector ($\mathbf{net_k}$).

$$y_{j,k} = \mathrm{sgm}(\mathbf{net_k}) \quad (2.10)$$

1) The transfer function for unipolar sigmoid activation function is:

$$f(\mathbf{net_k}) = \frac{1}{1 + e^{(-\mathbf{net_k})}} \quad (2.11)$$

2) The transfer function for bipolar sigmoid activation function is:

$$f(\mathbf{net_k}) = \frac{2}{1 + e^{(-\mathbf{net_k})}} - 1 \quad (2.12)$$

Then the response of j^{th} hidden neuron from k^{th} input feature vector as:

For unipolar sigmoid activation function is:

$$y_{j,k} = f(net_j) = \frac{1}{1+e^{(-net_j)}} \tag{2.13}$$

$$\mathbf{net_k} = W\,Y \tag{2.14}$$

The activation value at of the i^{th}output neuron of the output layer is:

$$net_i = w_{i,j} * y_j \tag{2.15}$$

The response at i^{th} output node from k^{th} input feature vector if a unipolar sigmoid activation function is used:

$$o_i = f(net_i) = \frac{1}{1+e^{(-net_i)}} \tag{2.16}$$

$$e_i = d_i - O_i \tag{2.17}$$

Where d_i is desired output, O_i is actual output, and e_i is error of the network. The objective function (training error) which is popular to use for the Sum of Squares Error (SSE) that measures the error for each pattern given by training set. This error from k^{th} input feature vector is:

$$E_k = \frac{1}{2}\sum_{i=1}^{n_o}\left(d_{i,k} - O_{i,k}\right)^2 \tag{2.18}$$

Note: (E_k) of eq. (2.18) some time is called the objective function is:

$$E_k = \frac{1}{2}\left\|d_{i,k} - o_{i,k}\right\|^2 \tag{2.19}$$

Back-propagation algorithm applies a weight corrections Δw and Δv to update synaptic weights (both W & V) so as to minimize sum of the squared error (sse).Since the gradient descent is used for minimizing the training error, weights correction from k^{th} input feature vector is determined by minimizing (derivation) of the quadratic error function (objective function) with respect to particular synaptic weight as below:

$$\Delta w_{ij} = -\eta \frac{\partial E_k}{\partial w_{ij}} \qquad (2.20)$$

Where η is the learning rate, which used to control learning step and merely represents relative size of the weights change. For a three layer network, it is clearly that the error dependent on w_{ij} as follows:

$$net_i = \sum_{j=1}^{n_h} w_{ij}\, y_j \qquad (2.21)$$

$$O_i = f(net_i) \qquad (2.22)$$

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial net_i} \cdot \frac{\partial net_i}{\partial w_{ij}} \qquad (2.23)$$

$$\delta_i^o = -\frac{\partial E_k}{\partial net_i} \qquad (2.24)$$

Where δ_i^o is referring to output error of i^{th} output from k^{th} input feature vector that needs to be back-propagated. From eq. (2.15), then

$$\frac{\partial net_i}{\partial w_{ij}} = y_j \qquad (2.25)$$

Substitute both eq.(2. 24) and eq.(2. 25) in eq.(2.23) yields:

$$\frac{\partial E_k}{\partial w_{ij}} = -\delta_i^o y_j \qquad (2.26)$$

Substitute eq. (2.26) in eq.(2.20) yields:

$$\Delta w_{ij} = \eta\, \delta_i^o\, y_j \qquad (2.27)$$

From above eq. and by using chain rule we can say:

$$\delta_i^o = -\frac{\partial E_k}{\partial net_i} = -\frac{\partial E_k}{\partial O_i} \cdot \frac{\partial O_i}{\partial net_i} \quad (2.28)$$

$$\frac{\partial O_i}{\partial net_i} = \frac{\partial f(net_i)}{\partial net_i} = f'(net_i) \quad (2.29)$$

$f'(net_i)$: is the derivative of output response with respect to net_i,

$$\frac{\partial E_k}{\partial O_i} = -(d_i - O_i) \quad (2.30)$$

Substitute both eq.(2. 29) and eq.(2. 30) in eq.(2.28) yields:

$$\delta_i^o = (d_i - O_i)\, f'(net_i) \quad (2.31)$$

Substitute eq. (2.31) in eq. (2.27) yields:

$$\Delta w_{ij} = \eta(d_i - O_i)\; f'(net_i) y_j \quad (2.32)$$

The updating eq. for the synaptic weight (w_{ij}) is as follow:

$$w_{ij,k} = w_{ij,k-1} + \Delta w_{ij} \quad (2.33)$$

Substitute eq (2. 32) in eq. (2.33) yields:

$$w_{ij,k} = w_{ij,k-1} + \eta(d_i - O_i)\, f'_i(net_i) y_j \quad (2.34)$$

A) The formulas for updating equation for the synaptic weight (w_{ij}) can be derived as follow: [for more details see Appendix A]

I) Then for unipolar continuous sigmoid activation eq.(2.32) is as:

$$\Delta w_{ij} = \eta(d_i - O_i)\; O_i(1 - O_i) y_j \quad (2.35)$$

$$w_{ij,k} = w_{ij,k-1} + \eta(d_i - O_i)\, O_i(1 - O_i)\, y_j \quad (2.36)$$

II) for bipolar continuous sigmoid activation eq.(2.32) is as:

$$\Delta w_{ij} = \frac{\eta}{2}\,(d_i - O_i)\,.\left(1 - O_i^{\,2}\right) y_j \quad (2.37)$$

$$w_{ij,k} = w_{ij,k-1} + \frac{\eta}{2}\,(d_i - O_i)\,.\left(1 - O_i^{\,2}\right) y_j \quad (2.38)$$

Where$w_{ij,k-1}$ & $w_{ij,k}$ are weights before and after updating respectively.

B) while the changes in the weights of the input-to- hidden units (Δv_{jl}) [for more details see Appendix A], are computed as follow:

I) for unipolar continuous activation over all nodes in hidden layer

$$f'(net_j) = y_j(1 - y_j) \tag{2.39}$$

$$\Delta v_{ji} = \eta\, f'(net_j) \sum_{i=1}^{n_o} \delta_i^0 w_{ij} \cdot X \tag{2.40}$$

$$f'(net_j) = y_j(1 - y_j) \tag{2.41}$$

Then eq(2.40) can be written as:

$$\Delta v_{jl} = \eta \cdot ((y_j(1 - y_j)) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{2.42}$$

The updating eq. for the synaptic weight (V_{jl}) is as follow:

$$v_{jl,k} = v_{jl,k-1} + \Delta v_{jl} \tag{2.43}$$

$$v_{jl,k} = v_{jl,k-1} + \eta \cdot ((y_j(1 - y_j)) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{2.44}$$

II) For bipolar continuous activation over all nodes in the hidden layer:

$$v_{jl,k} = v_{jl,k-1} + \Delta v_{jl}$$

$$\Delta v_{ji} = \eta\, f'(net_j) \sum_{i=1}^{n_o} \delta_i^0 w_{ij} \cdot X$$

$$f'(net_j) = \frac{1}{2}(1 - y_j^2) \tag{2.45}$$

$$\Delta v_{jl} = \frac{\eta}{2} \cdot ((1 - y_j^2) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{2.46}$$

$$v_{jl,k} = v_{jl,k-1} + \frac{\eta}{2} \cdot ((1 - y_j^2) \sum_{i=1}^{n_o} \delta_i^0 w_{ij}) \cdot X \tag{2.47}$$

Where $v_{ij,k-1}$ & $v_{ij,k}$ are the synaptic weights before and after the updating respectively.

CHAPTER THREE

ARTIFICIAL NEURAL NETWORK SIGNAL CLASSIFIER SYSTEM

3.1 Introduction

The aim of this chapter is to design a system for classification of signals, classification is a form of pattern recognition and it has been the subject of much research, both theoretical and experimental. Classification systems often begin with some kind of preprocessing to remove noise and redundancy in the measurements sharpening and smoothing by ensuring an effective and efficient signal description [3].

ANN signal classifier system is here to prove the classification ability of ANN for signals, therefore, here EEg signal database will be used as recorded signals to be used for classification will be applied on it.

3.2 Methodlogy for Signal Classification System

FFNNs must use the sample information as a mere reference for creating the internal representations. Thus, it should not encode the sample information accurately into the internal representations. Such an exact or faithful encoding of the sample information results in the FFNN memorizing the “crispness” in the training data set.

3.3 Structure of ANN Signal Classifier System

The overall block diagram that shows the structure of ANN as signal classifier system is shown in (Figure 3.1). Every single recorded the electroencephalograph (EEG) is as one of the first and still very useful ways of non-invasively observing human brain activity. It continues to be an essential part in studying patients with seizures and those suspected of having seizures. It is also used in evaluating the cerebral effects of many systemic metabolic diseases, in the study of sleep and in the operating room to monitor cerebral activity in anesthetized patients [55](see appendex D).

The structure of ANN signal classifier system can be shown by the following principal steps:

1) **Normalization**: Feature vectors (patterns) in the data set should be normalized at first.

Raw EEG data is generally a mixture of several things: brain activity, blinking eye, muscle activity, environmental noise, etc. After that data was collected, the intended step is processed to remove all the noise is necessary and artifacts of the signal, keeping only the part interestingly, of the signal, the activity of the brain. In order to get satisfying results from the system, the preprocessing step should:

● Remove all noise and artifacts from the signal, but preserve all the characteristics of the original signal.

● Clean the signal from the influence of the reference electrode, if one is used. The requirements for feature extraction are:

● Reduction of the size of the data by selecting the appropriate features.

● Selected redundant minimum features, should expect results that depend on reducing the maximum of these features.

● Keeping all the information of the signal if there is a need for classification. All the above steps can perform by using different methods and algorithms, but the independent component analysis (ICA) is the best way to address the EEG signals. ICA is used to remove artifacts from different recorded EEG signals

2) **Training or Classification step**: they are applied to the network for training or classification purposes.

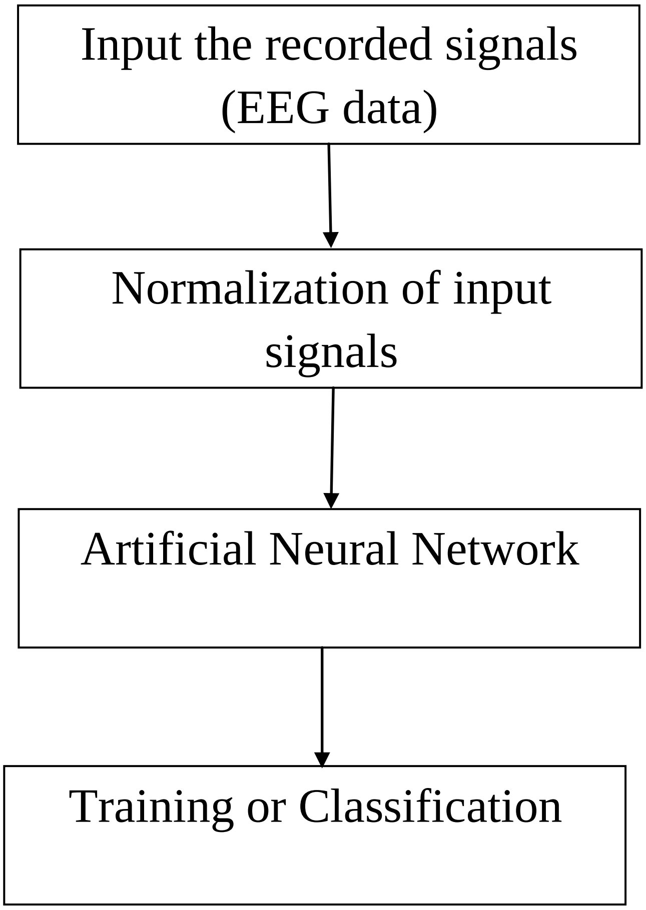

Fig3.1: Block diagram of the methodlogy for signal classification system

3.4 Architecture of Artificial Neural Network

The architecture of the ANN classifier is shown in figure 3.2 .The network is composed of three layers which are made up of a number of neurons. All neurons in the same layer are identical in their functions, but neurons may have different function in different layers.

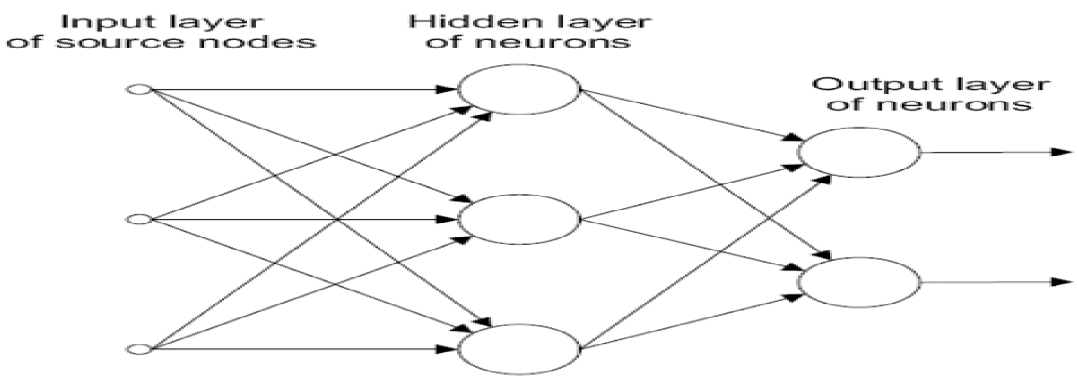

Fig 3.2: Structure of Artificial Neural Network (ANN).

3.5 Gradient-Descent-Based Learning Algorithm for ANN

Learning algorithm for parameters of ANN is considered in single phases: synaptic weights (both v_{jl} which connects input layer by all hidden units and w_{ij} connect hidden units and the output layer) need to be updated at first in order to train ANN to consistently partition feature space of given data set simultaneously.

NOTE: End forms for updating formulas of synaptic weights [v_{jl} &w_{ij}] but here deeply rederivation them until getting end formulas in order to understand process of learning.

3.6 Algorithm of Training for the ANN

ANN is trained in a sequence of adaptation cycles; each adaptation cycle involves adaptation of all the internal parameters of the network, that is, the synaptic weights (both of v_{jl} and w_{ij}) The learning time or the convergence rate of training of ANN is the number of training epochs that required for sum-squared error to reach a certain minimum error value during training of ANN using gradient-descent learning algorithm.

The training of the ANN is as in algorithm is as follow:

Algorithm: Back propagation algorithm

Given m training pairs

$$\{x_1, d_1, x_2, d_2, \dots, x_m, d_m\}$$

Where x_l^k is $(n_i,1)$, d_i^k is $(n_o,1)$, and $k = 1,2,\dots,m$. The hidden layer outputs y is $(n_h,1)$ and z is $(n_o,1)$.

Step1: weights V and W are initialized at small random values; V is

(n_i, n_h), W is (n_h, n_o)

Step 2: Training step starts her, input is presented and the layers outputs computed

$$y_j = f(\sum_l x_l \ v_{lj})$$

$$z_i = f(\sum_j y_j \ w_{ji})$$

f : Activation function.

y_j : Actual output of hidden neuron.

x_l : Input signal.

v_{lj} : Synaptic weight between input and hidden layer.

z_i : Actual output of the output neuron i.

w_{ji} : Synaptic weight between hidden neuron and output neuron.

Step 3: Error value is computed:

$$E = \frac{1}{2}\sum_{i=1}^{n_o}(d_i - z_i)^2$$

Step 4: Error signal vectors δ_i and δ_j of both layers are computed. Vectors δ_i is matrix of $(n_o,1)$, δ_j is matrix of $(n_h,1)$

The error signal terms of the output layer in this step are

$$\delta_i = (d_i - z_i) \ f'(\sum_j y_j w_{ji}) \quad \text{for } i = 1,2,\dots,n_o$$

f': The derivative of the activation function with respect time.

d_i : The desired output of neuron i.

The error signal terms of the hidden layer in this step are

$$\delta_j = \sum_i \delta_i \, w_{ji} \;\; f'(\sum_l x_l \, v_{lj}) \quad \text{for } j = 1,2,\dots,n_h$$

Step 5: Output layer weights are adjusted:

$$w_{ji} = w_{ji} + \alpha\, \delta_i y_j \quad \text{for } i = 1,2,\dots,n_o \;\; \text{and} \;\; j = 1,2,\dots,n_h$$

α : The learning rate.

Step 6: Hidden layer weights are adjusted:

$$v_{lj} = v_{lj} + \alpha\, \delta_j x_l \quad \text{for } j = 1,2,\dots,n_h \;\; \text{and} \;\; l = 1,2,\dots,n_i$$

Step 7: The training cycle is completed.

For $E < E_{max}$ terminate the training session. Output weights W, V, and E.

If $E > E_{max}$, then $E \rightarrow 0, m \rightarrow 1$, and initiate the new training cycle by going to Step 2.

The BP process continues until both all weights are adjusted, the adjusted synaptic weights (v_{jl} & w_{ij}) and testing data as a information is fed forward through network and errors at output layer computed .So, the process continue until either the performance index reaches an acceptable low value, a maximum iteration count (number of epochs) has been exceeded, or a training- time period has been exceeded.(see figure 3.3)

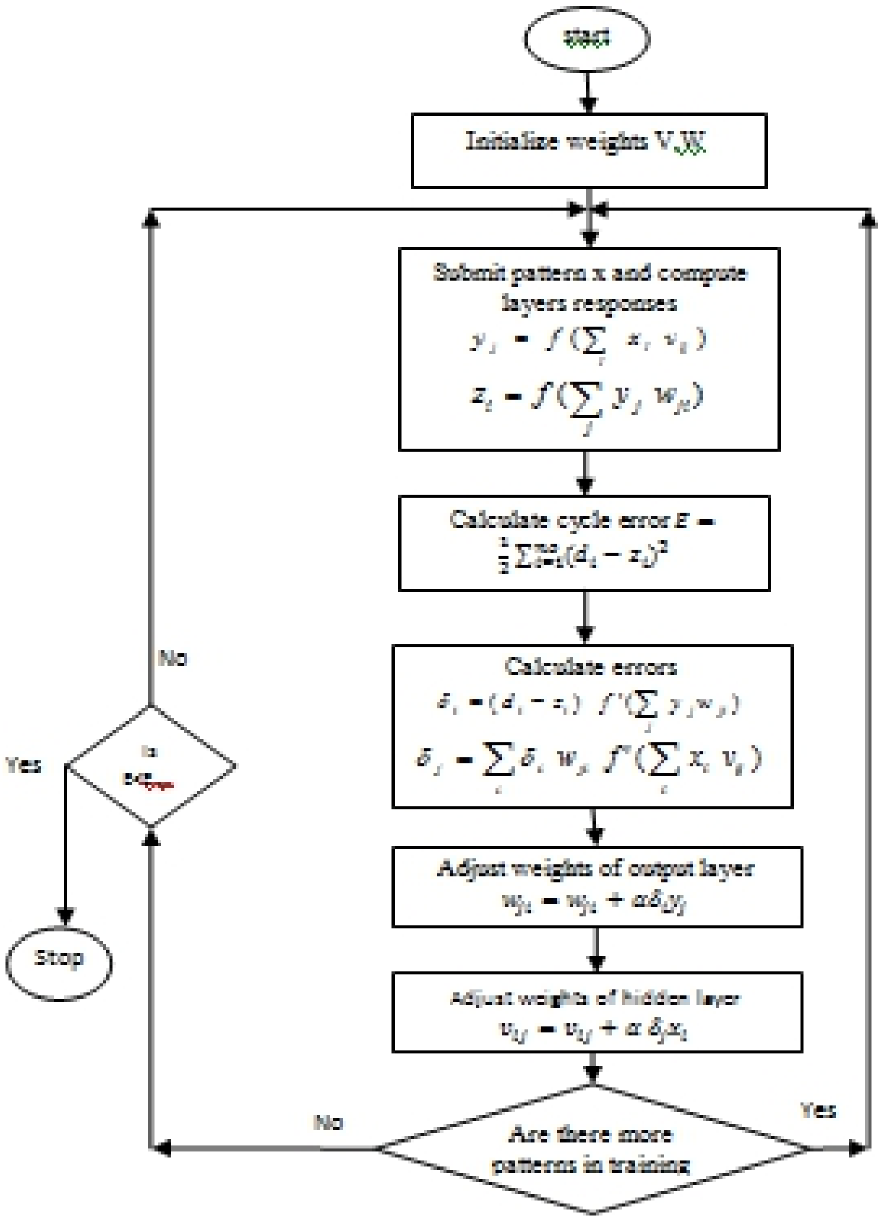

Figure 3.3. Flow chart of error back propagation algorithm.

The learning time or convergence rates are mentioned as the number of training epochs required for the sum-squared error to reach a certain minimum value, these convergence rates are comparable to those of any other gradient-descent-based algorithm applied to similar problems.

CHAPTER FOUR

RESULTS AND DISCUSSION OF SIGNALS' CLASSIFICATION

4.1Classification Experiment

ANN signal classifier system here to enhance the classification ability for signals. For this purpose the data signals database is selected as a recorded signals for classification.

From the EEG data the whole database consists of five EEG data set (denoted A-E), each containing 100 single channel EEG signals of 23.6 seconds from five separate classes. Sets A and B consisted of signals taken from surface EEG recordings of five healthy volunteers with one eye open and one eye closed, respectively. Signals in set C and set D were recorded in seizure-free intervals from five epileptic patients from the hippocampal function formation of the opposite hemisphere of the brain and from within the epileptic zone, respectively. Set E contains the records of five epileptic patients during seizure activity.

All EEG recordings were made with the same 128-channel amplifier system, using an average common reference. The recorded data were digitized at173.61 samples per second using 12-bit resolution. Band-pass filter setting were 0.53-40 Hz. The amplitude of EEG recordings is given in micro volt [60].

THE DEFINITION OF EEG SIGNALS SETS

SET	The details
A and B	consisted of signals taken from surface EEG recordings of five healthy volunteers with one eye open and one eye closed, respectively
C and D	recorded in seizure-free intervals from five epileptic patients from the hippocampal function formation of the opposite hemisphere of the brain and from within the epileptic zone, respectively
E	contains the records of five epileptic patients during seizure activity

The number of elements of each pairs divided into two phases:

- **Training Phase:** Is used for training network (ANN) and the input feature vectors at this stage called input training feature vectors or called training samples. These samples usually take 60%, 65%, or 70% from data. The data sample is sent through the network and calculated the error and accuracy of training data by Error Back Propagation (EBP) and then optimizing is used, the most commonly optimization algorithm for training neural network which is the Gradient Descent (GD)optimization.
- **Testing Phase*:*** Is used for testing ANN classifiers and the input feature vectors called input testing feature vectors or called testing samples. These samples usually take 40%, 35%, or 30% from data. The error and the accuracy are calculated for ANN classifiers.

4.3 Results of using ANN Classifier

For the FFNN classifier, the network structure is 32-128-2 namely the input vector is 32, the hidden layer node number is 128 and the output vector is 2.
The learning ratio of w_{ji} , v_{lj} is 0.35 training by MATLAB programming test. The number of hidden units is where is the number of elements in each $4*n_i$, where n_i the number of number of element in each feature vector (it was chosen by trial and error), the input weight is (32,128) and output weight is (128,2) updated after each iteration and the number of iterations was set to 200.

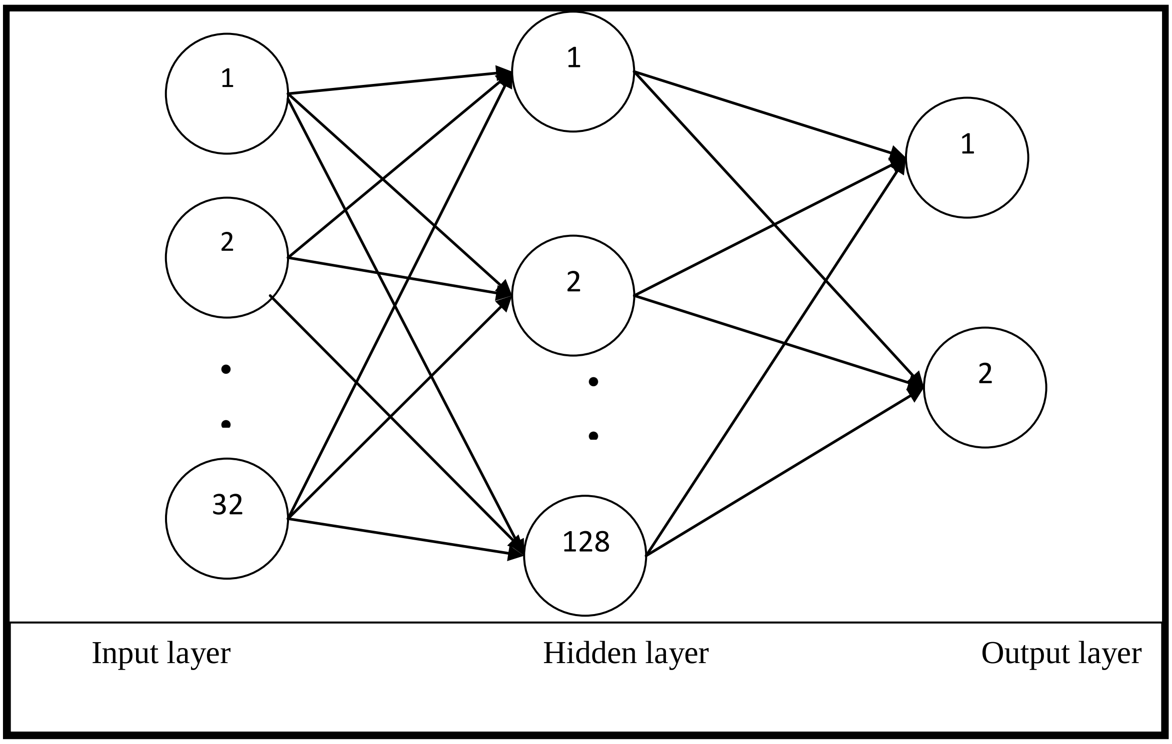

Fig 4.1 Structure of the Artificial Neural Network (ANN) classifier

Three layers ANN was employed as classifier of Iris data signal, randomly select input feature vectors to achieve the uncertainty principle. Randomly selected 70% samples as training samples and 30% for testing samples. Learning rate (η) is chosen by trial and error for weigh adjusting (W_{ij}, V_{jl}) is set to (0.01) MATLAB programming test, the number of iteration (epochs) is set to 200, with following observations:

1) The trial and error method (done by MATLAB[Appendix **B** shows step by step execution of the ANN classifier algorithm]) for randomly selection of feature vectors (to apply the uncertainty principle the selected features vectors) should be guaranteed at any moment the inputted features vectors are with input sequence completely different from preceding ones and certainly also different from those which come after this moment .

In each experiment, a pair of two-class of EEG signals are used, which had 3200 vectors with 32 dimensions taking 1600 vectors for each class. The best classification result was obtained for Sets A and E which was 74.3303%, Sets B and E was 78.8392%, Sets C and E was 79.4628%, Sets A and D was 68.5714% and Sets D and E was 74.0178% . The average classification accuracy rate of FFNN was 75.0763% and the average error was 0.1755 and 0.3612 for training and testing patterns, respectively.

Table (4.1)Performance of the FFNN for different pairs of two-class EEG signal from the epileptic EEG data

pairs	*MSE for training set*	*MSE for testing set*	Accuracy
Set A & E	0.1896	0.3765	74.33
Set B& E	0.1578	0.3125	78.83
Set C& E	0.1374	0.2968	79.64
Set A & D	0.2178	0.4325	68.75
Set D & E	0.1774	0.3875	74.0174
average	0.1765	0.3612	75.07

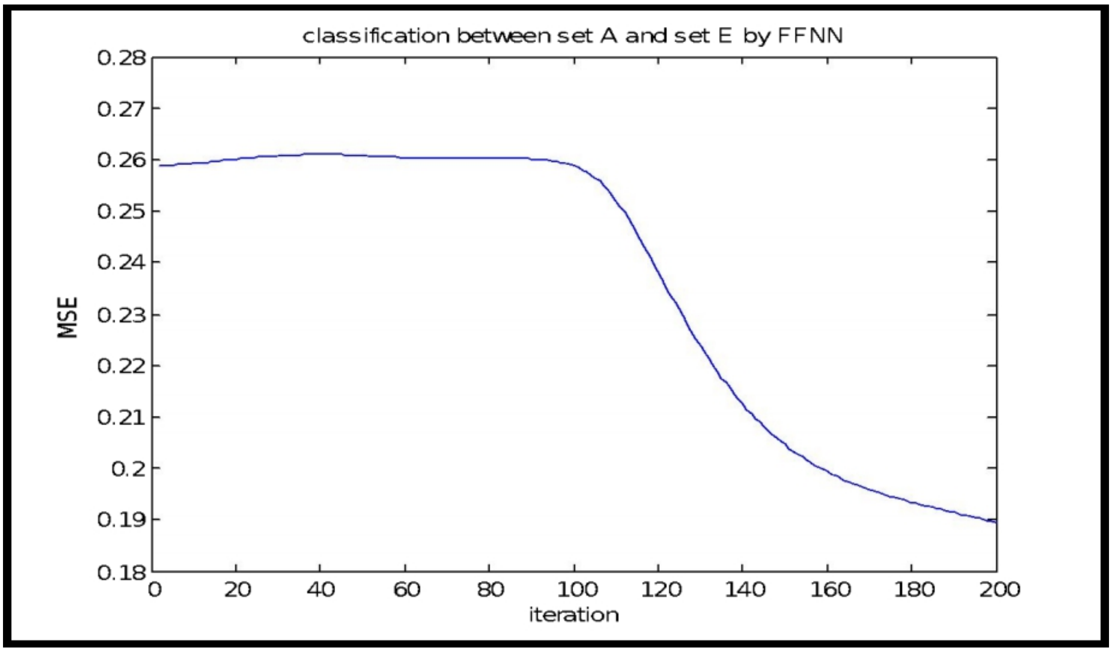

Figure 4.6a . Classification between Set A and Set E by FFNN.

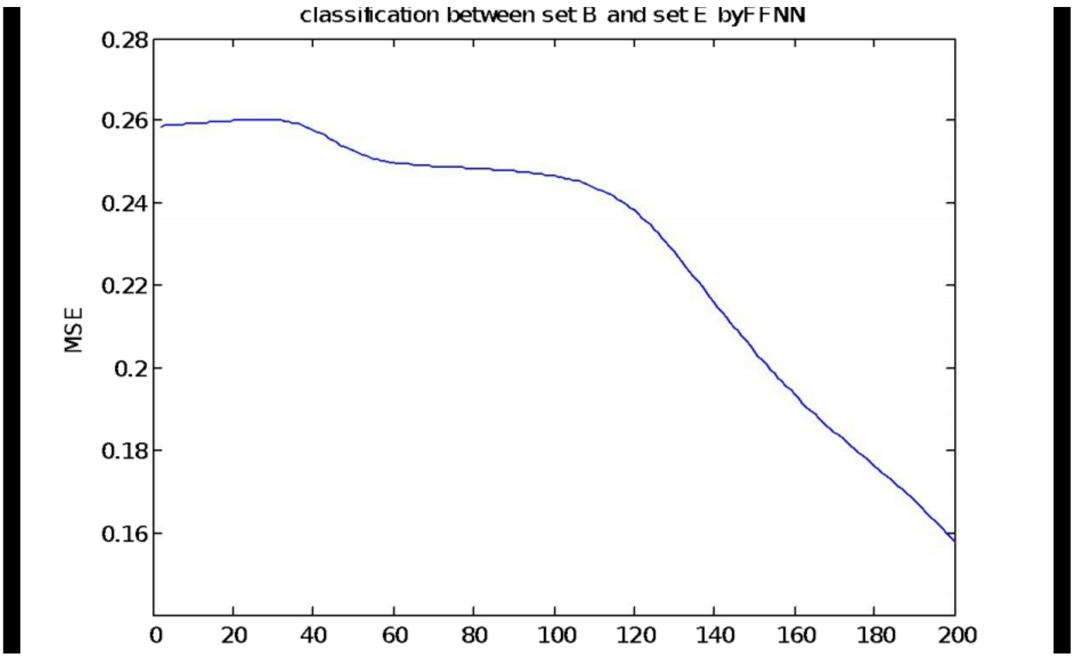

Figure 4.6b . Classification between Set B and Set E by FFNN.

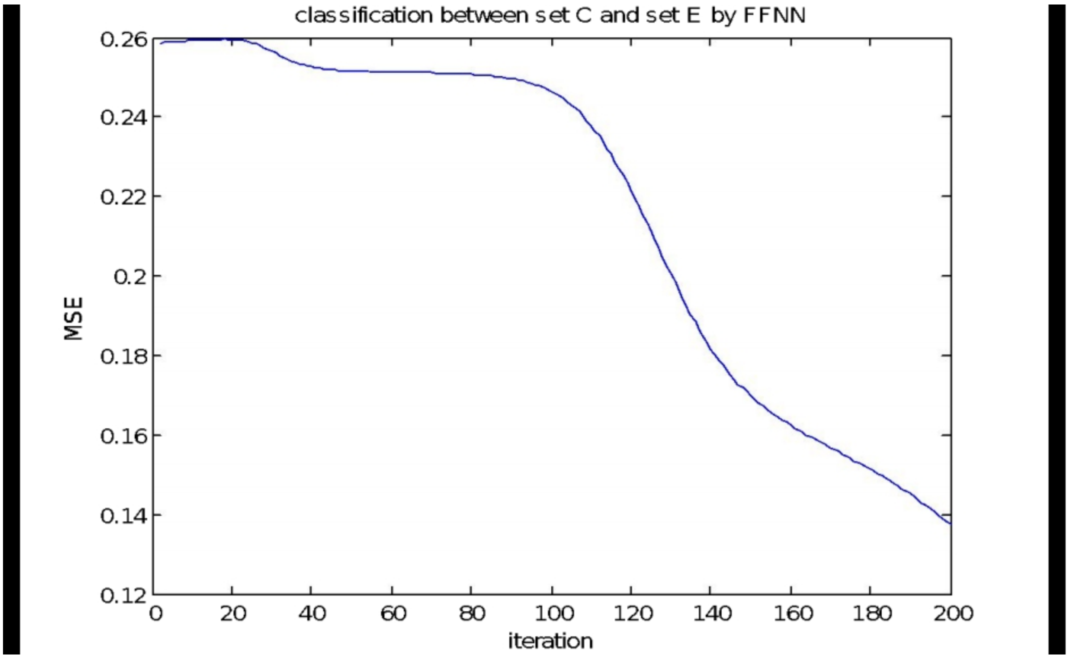

Figure 4.6c . Classification between Set C and Set E by FFNN.

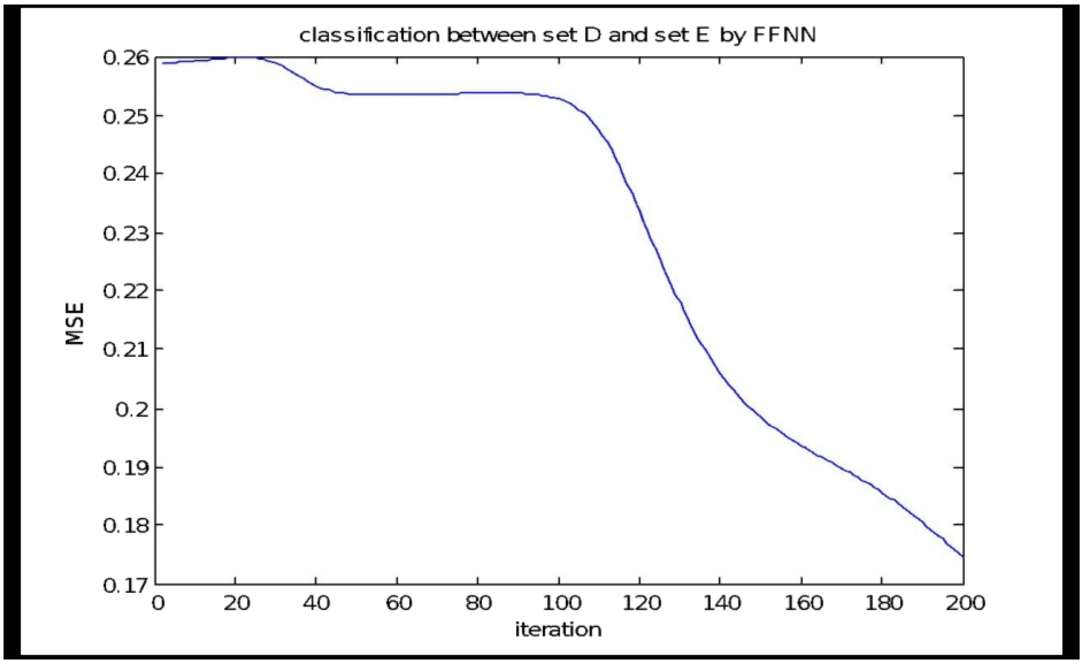

Figure 4.6d.Classification between Set D and Set E by FFNN.

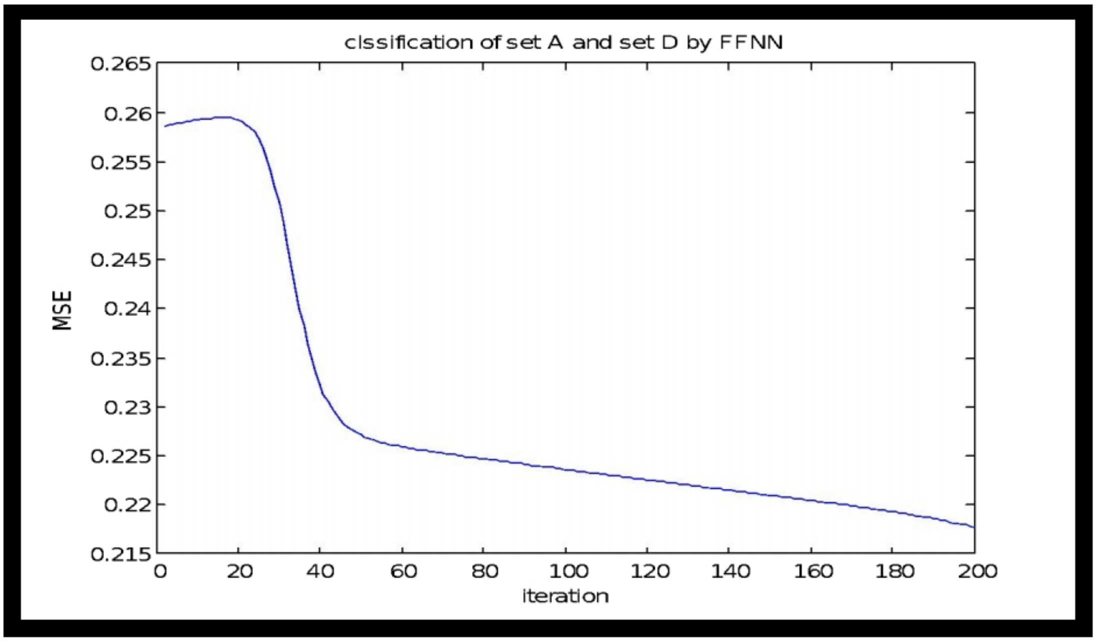

Figure 4.6e. Classification between Set A and Set D by FFNN.

Chapter Five

Conclusions and Suggestions for Future Work

5.1 Conclusions

Structure of ANN has been presented and applied in classification field to show enhancements, Iris data is used as a classification signals for classification task by both classifiers.

From previous experiment the following results can be concluded:

- Ensuring classification capabilities, thus making network clearer, simple, and greatly reduces network size, improve learning speed.
- ANN classifier shows that number epochs were used is less, i.e. ANN takes shorter time.
- Hidden units' responses to input segments from training and testing sets indicated that trained ANN produced a more structured internal representation of input samples. This experimental result was justified by illustrating the ability of trained ANN to implement classification tasks.

5.2 Suggestions for Future Work

- Genetic Algorithms (GA) and/or Particle Swarm Optimization (PSO) could be used as learning algorithms which may result in faster convergence and better generalization of classifiers.
- Different activation function types could be used rather than sigmoid activation such as radial basis function to give radial neural network (RNN) classifiers. Smaller number of adjusted parameters (less epochs) with fewer computations and may result in faster convergence and more accurate results.
- Using larger databases to study performance of ANN classifiers in a more complex situation, such as the biological signals (ECG).

REFERENCES

[1] http://en.wikipedia.org/wiki/*Cluster_analysis*.

[2] J. Zurada, "*Introduction to Artificial Neural Systems*," ISBN 0-3 14-93391-3, West publishing company, USA, 1992.

[3] A. R. Musslim, "*A Proposed Three-dimensional Data Structure Neural Network for Image Recognition*," M.Sc. thesis, Department of Electrical Engineering, College of Engineering, University of Baghdad, 2011.

[4] D. Garrett, G. Brassad, I. Chuang, S. Lioyed, and C. Monroe, "Comparison of Linear and Nonlinear Methods for EEG signal classification", IEEE Transaction Neural System. Vol.11, No.2, pp.4-141, 2003

[5] M. Husken and P .Stagge, *"Recurrent Neural Networks for Time Series classification ,"* Neurocomputing,Vol 50 , PP. 223-235, 2003.

[6] A. Saxena and A. Saad, *"Evolving an Artificial Neural Network Classifier for Condition Monitoring of Rotating Mechanical Systems,"* Journal of Applied Soft Computing, Elsevier Publishers, ISSN:1568-4946, 2005.

[7] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, G. D. Tourassib, *" Training Neural Network Classifiers for Medical Decision Making: The Effects of Imbalanced Datasets on Classification Performance,"* Neural Networks, Elsevier Ltd., Vol 21, pp. 427-436, 2008.

[8] http://en.wikipedia.org/wiki/*Neural_Networks*.

[9] A. A.H. Mahdi, "*The Application of Neural Network in Financial Time Series Analysis and Prediction Using Immune System*," M.Sc. thesis, School of Computing and Mathematical Sciences, Liverpool John Moores university, 2010.

[10] M. N. ALaubaidy, *"Simulation of the Shortest Path Routing Computation Using Neural Network,"* M.Sc. thesis, Department of Computer Science, College of Science, University of Baghdad, 2004.

[11] Y. H. Hu and J. N. Hwanga, "Handbook of Neural Network Signal Processing," ISBN 0-8493-2359-2, CRC Press LLC, USA, 2002.

[12] S. A. Hajani, "*A Neural Network-Based Signature Recognition System*," M.Sc. thesis, Department of Computer Engineering, College of Engineering, University of Nahrain, February 2007.

[13] http:// *www.cormactech.com/neunet/* Download EEG Data*.html*.

Ministry of Higher Education and Scientific Research
University of Diyala
College of Engineering
Electronic Engineering Department

EEG Signal Classification Using FFNN

A project
Submitted to the department of electronic University of Diyala-College of Engineering in Partial Fulfillment of the Requirements for the Degree of Bachelor in Electronic Engineering.

By
Salam Muneer
Sarah Tarik

Supervised By
MSC.
Hassan Saadallah

June / 2016

وزارة التعليم العالي والبحث العلمي

جامعة ديالى

كلية الهندسة

قسم الهندسة الالكترونية

تصنيف الاشارات الدماغية بأستخدام الشبكات العصبية الصناعية

المشروع

قدم إلى قسم الهندسة الإلكترونية في جامعة ديالى كلية الهندسة كجزء من متطلبات الحصول على درجة البكالوريوس في الهندسة الإلكترونية.

من قبل

سلام منير

ساره طارق

بإشراف

م.م. حسن سعدالله ناجي

حزيران ٢٠١٦

Supervisor Certification

I certify this project entitled (**EEG Signal Classification Using FFNN)** was prepared under my supervision at the Electronic department of Engineering College University of Diyala, by **Salam Muneer & Sarah Tarik**, as partial fulfillment of the requirement for the Degree of Bachelor in Electronic Engineering.

Signature:

Asst. Lect.: MSC. Hassan Saadallah Naji

Date: /   / 2016

Certification of the Examination Committee

We certify that we have read the project entitled " **EEG Signal Classification Using FFNN"** and as an examining committee, we examined the student (**Salam muneer & Srah Tarik**) in its contents and that in our opinion it meets the standards of a project for the degree of B.SC of science in "Electronic Engineering".

Signature:

Name:

Title:

Date: /   / 2016

EEG Signals

The electroencephalograph (EEG) is as one of the first and still very useful ways of non-invasively observing human brain activity. It continues to be an essential part in studying patients with seizures and those suspected of having seizures. It is also used in evaluating the cerebral effects of many systemic metabolic diseases, in the study of sleep and in the operating room to monitor cerebral activity in anesthetized patients [55].

The electrical activity of the brain (EEG) is easily recorded from electrodes placed on scalp. These electrodes pick up electric signals naturally produced by the brain and send them to galvanometers which detect and measure small electric currents that are in turn hooked up to pens, under which graph paper moves continuously. The pens trace the signals onto the graph paper [56]. The amplitude and frequency of EEG signals vary according to human state (asleep or awake), age, health etc. There are five major brain waves distinguished by their frequency ranges . These are alpha (α), theta (θ), beta (β), delta (δ) and gamma (γ) [57].

An EEG can show if a person is asleep, awake, and anesthetized because the properties patterns of current differ for each of these situations . One important use of EEGs has been to show how long it takes the brain to process various incentives. A major disadvantage of EEGs is that they cannot show us the structures and anatomy of the brain or really tell us which specific areas of the brain do what [56].

What are EEG Signals?

Electroencephalography (EEG) is the recording of spontaneous electrical activity of the brain that is obtained by firing neurons within the brain. EEG signals are recorded in a short time, usually for 20-40 minutes. The recordings are obtained by putting the electrodes at various positions on the scalp. The wrong belief that the EEG signals only represents the brain signals but the right believe that the EEG signals represent the brain signals and the status of the whole body [57].

Electrode locations are determined by dividing these parameters into 10% and 20% intervals. This results in a total of 19 electrodes which adequately cover the scalp. The “10” and “20” refer to the fact that the actual distance between electrodes is either 10% or 20% of the total front-back or right-left distance of the skull. Even numbers (2,4,6,8) refer to electrode position on the right hemisphere, where an odd numbers (1,3,5,7) refer to those on the left hemisphere [58] as shown in Figure (4.1).

The EEG consists of a set of multi-channel signals. The pattern of changes in signals reflects large-scale brain activities. In addition, the EEG also reflects the activation of the head musculature, eye movements, interference from nearby electrical devices and changing conductivity in the electrodes due to the movements of the subject. All of these activities are not directly related to the current cognitive processing of the subject is collective referred to as background activities [55].

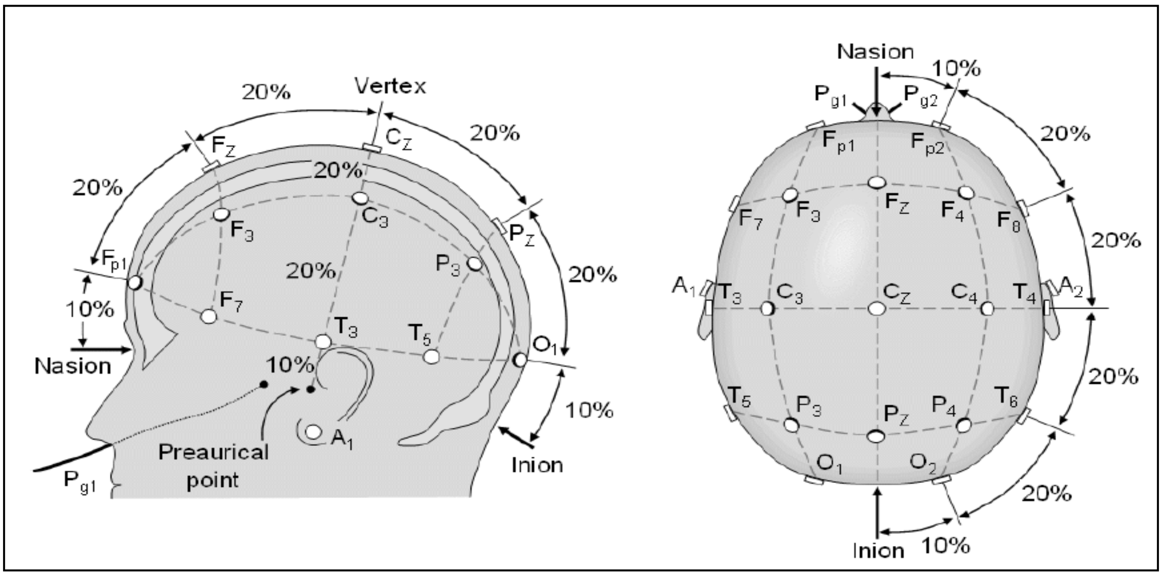

Figure 2.11.a The International 10/20 International System seen from left [56].

Figure 2.11.b The 10/20 System

Brainwaves to Study the State of the Brain

The brain is an electrochemical organ. Electrical activity emanating from the brain is displayed in the form of brainwaves. There are five categories of these brainwaves, ranging from the last activity to the most activity [56].

Delta waves are within the range 0.5-4Hz. They are associated with deep sleep and may be present in waking state [57].

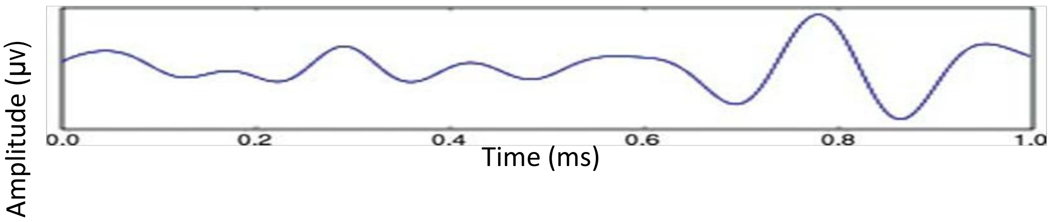

Figure 2.12.a Delta waves [58].

The theta band consists of frequencies between 4Hz and 7Hz. This activity can be observed with drowsiness or meditation [59].

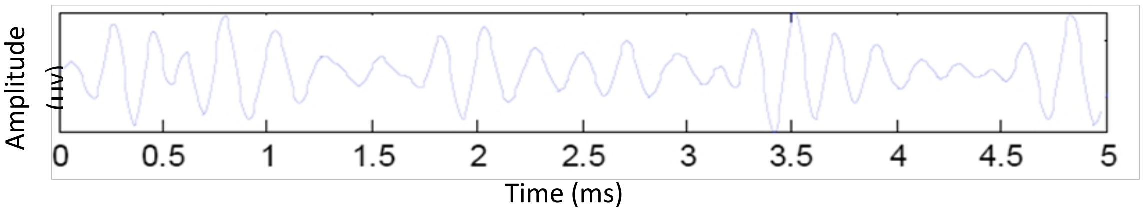

Figure 2.12.b Theta waves [58].

Alpha wave frequencies lie within the range of 8-13 Hz. They can be detected in the posterior lobes of the brain. They are detected in a normal person when he is in a relaxed state without any attention or concentration [57].

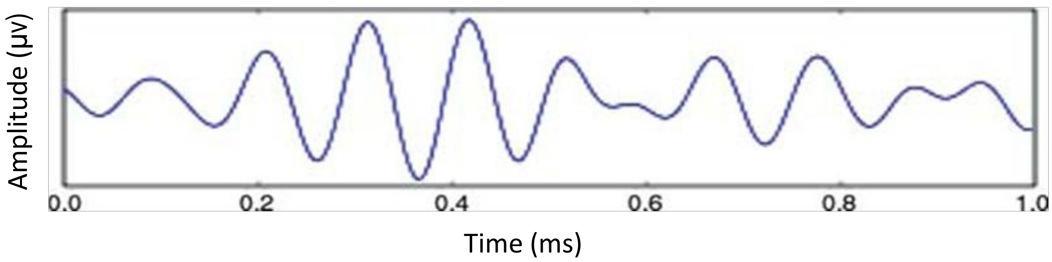

Figure 2.12.c Alpha waves [56]

A beta wave lies in the range 14-26 Hz. It is chiefly encountered in frontal and central regions. It is the usual waking rhythm of the brain associated with active concentration, active thinking, and problem solving and focusing on things [58].

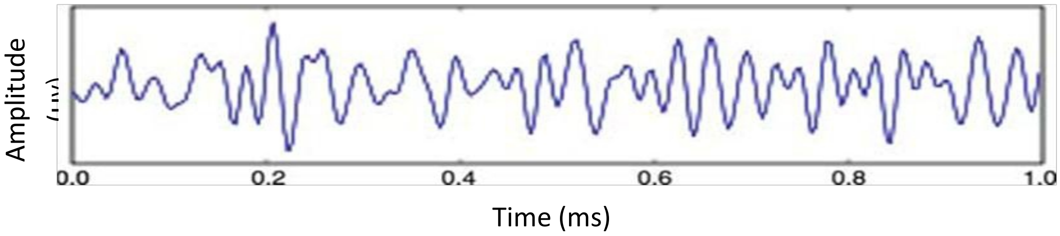

Figure 2.12.d. Beta wave [56]

Gamma waves also called fast beta waves and they have frequencies above 30Hz. The amplitude of these waves is very low and they have rare occurrence. They are associated with certain cognitive and motor function [58].

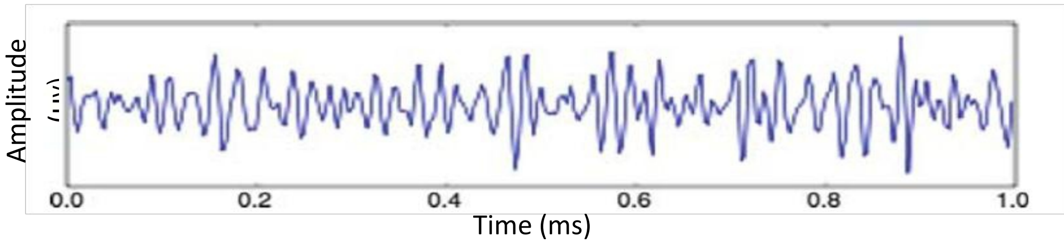

Figure 2.12.e Gamma waves [57].

The amplitude of EEG signals is very low, varying between 5 and 100 mV. Five main types of EEG waves are distinguished, as shown in Table 4.1 [60].

Table (2.1): Amplitude of Five types of EEG waves [60]

Type of EEG	**Frequency (Hz)**	**Amplitude (μV)**
Delta (δ)	0.5-4	10-300
Theta (θ)	4-7	< 50
Alpha (α)	8-13	~ 50
Beta (β)	14-26	< 30
Gamma (γ)	Above 30	Amplitude is very low.