Predicting Time Lag between Primary and Secondary Waves for Earthquakes Using Artificial Neural Network (ANN)

Ogbole Collins Inalegwu

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2015
Gazimağusa, North Cyprus.

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Serhan Çiftçioğlu
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Supervisor

Examining Committee

1. Prof. Dr. Omar Ramadan
2. Assoc. Prof. Dr. Muhammed Salamah
3. Asst. Prof. Dr. Gürcü Öz

ABSTRACT

This thesis work investigates the possibility of predicting the arrival time of the secondary seismic earthquake waves. Seismic waves are low frequency acoustic waves that are experienced prior to earthquake, they are basically two types: the primary wave (p-wave) and the destructive secondary wave (s-wave). These waves' approaches a destination at different times, with the p-wave experienced earlier since it travels at higher speed as compared to the s-wave. Knowledge of the time lag between this two waves recorded by a seismometer from previous earthquakes were used together with other parameters suspected to also influence the arrival of the secondary (destructive) earthquake waves which are; the magnitude from the propagating wave, the epicenter distance from the hypocenter, the seismic station's distance from the epicenter and the direction (in azimuths), were used for this prediction. The prediction model was carried out using neural network on MATLAB; the artificial neural network (ANN) design makes it possible to develop the correlation between the various parameters for the study. First, the network was trained with earthquake data of magnitude 6.0-7.0 Richter, validation and testing was carried out to measure the performance of the model. The result gave satisfactory performance, with regression values greater than 0.9, and the root mean square error (RMSE) computed were of the range of 0.1003 to 0.1148 for the most satisfactory network architecture. Secondly, the trained network was also tested with external values of magnitude range outside the values the network was earlier trained with. This network gave results that were not as good as the first case, so it was concluded that it's better to train the network with data from earthquake of all magnitude range. In general, from the experiment we concluded that the design and parameters

considered is possible for predicting the time-lag of these two seismic waveforms using artificial neural networks.

Keywords: Earthquake, Seismic waves, P-wave, S-wave, Seismometer, Atificial Neural Network, Hypocenter, Epicenter, Magnitude.

ÖZ

Bu çalışma, ikincil sismik deprem dalgalarının geliş zamanını tahmin etme olasılığını araştırmaktadır. Sismik dalgalar, depremden önce gerçekleşen düşük frekanslı akustik dalgalardır. Sismik dalgaların iki temel türü mevcuttur: birincil dalga (p-dalgası) ve tahrip edici olan ikincil dalga (s-dalgası). Bu dalgalar hedefe farklı zamanlarda yaklaşır; p-dalgası, s-dalgasından daha hızlı bir şekilde hareket ettiğinden daha önce gerçekleşir. Bir sismograf aracılığıyla kaydedilen iki dalga arasındaki zaman farkına ek olarak, ikincil dalganın gelişini etkilemesi beklenen farklı parametreler de kullanılmıştır. Tahmin için kullanılan parametreler şunlardır: yayılmakta olan dalganın boyutu, iç merkezden merkez üssü uzaklığı, sismik istasyonun merkez üssünden uzaklığı ve yönü (azimuth değerinde). Tahmin modeli MATLAB sinir ağı kullanılarak gerçekleştirilmiştir; sinir ağı (ANN) modeli sayesinde çalışmadaki çeşitli parametreler arasında bir korelasyon geliştirilebilmiştir. Öncelikle ağ, 6.0-7.0 Richter boyutundaki deprem verileri ve doğrulama ile donanmıştır ve ardından modelin performansını ölçümlemek üzere bir test yürütülmüştür. Testin sonunda 0.9 değerinden yüksek regresyon değerleri ile tatmin edici sonuçlar sağlanmıştır. En etkili ağ yapısında ise, işlenen ortalama karekök hatası (RMSE) 0.1003 ve 0.1148 aralığında bulunmuştur. Ardından geliştirilmiş olan ağ, daha önce test edildiği aralık dahilindeki değerlerin dışında olan büyüklük değerleri ile yeniden test edilmiştir. Genel olarak, tasarım ve parametreler ile yürütülen deneyin sonucunda, sinir ağları kullanılarak, iki sismik dalga türü arasındaki zaman farkının önceden tahmin edilebildiği görülmüştür.

Anahtar Sözcükler: Deprem, Sismik dalgalar, P-dalgası, S-dalgası, Sismograf, Yapay Sinir Ağı, İç merkez, Merkez üssü, Büyüklük.

Dedicated to God Almighty

ACKNOWLEDGMENT

I appreciate the love and assistance from my family in the course of this program, most especially my mum Mrs. Comfort Inalegwu, for her profound support to the success of this master's degree. She is indeed God's gift.

I would also love to extend my gratitude to my supervisor Assoc. Prof. (Dr.) Muhammed Salamah, whose deep insight was a guide all through the thesis work. And to all my lecturers and the thesis jury members, Prof. Dr. Omar Ramadan & Asst. Prof. Dr. Gurcu Oz, whose tutelage and constructive criticisms kept me on track.

My gratitude goes to the special friends with whom I enjoyed great times and gained a lot of knowledge; Dan and Daniel Kolo (the Twins), Cyril Ede, Fidel Aimua, Nnaemeka N., Muhammad Bima and the encouragement I receive from Mr. Goddey Samuel and family, Dr. Sebastine Umazua and family & the Navigators for their prayers and care. To all those I forgot to mention I still say a big thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
MAE	Mean Absolute Error
MBE	Mean Bias Error
MLP	Multilayer Perceptron
RMSE	Root Mean Square Error
SES	Seismic Electric Signals
SVM	Support Vector Machine
SOM	Self-Organization Maps
WSN	Wireless Sensor Network
N	The number of hidden neurons,
μ	Momentum constant,
η	Learning rate

Chapter 1

INTRODUCTION

1.1 General Overview

Earthquakes are the result of plate tectonics, and it occurs when a certain level of energy is release in the earth's crust resulting in seismic waves [1]. This energy forcefully tears apart the crust along the fault lines. Faults are cracks in the earth crust; these cracks either may be small and localized or can stretch as far as thousands of kilometers. Most earthquakes are caused by sudden release of stress energy along faults resulting from forces that have been slowly building up, and then eventually become so strong and forces rocks to break underground, releasing energy, which then spreads out in all directions, causing great movement and shaking. Volcanic also causes earthquakes in regions that experience it [2]. Earthquake is believed to be the most destructive of the natural hazards. Its occurrence most often results to massive loss of lives and serious damages in the affected region, depending on its magnitude and the community structure around.

The only mitigation to earthquakes is in its prediction, which unfortunately their occurrence is without reliable preceding events [3]. The prediction could be long term, medium or short term. The short-term prediction study on earthquakes has attracted extensive research lately; this study ranges from hours, to days and weeks. Most earthquake studies in the past were basically on understanding the basics: prediction study on this topic only started about three decades ago [4]. Thus,

earthquake prediction study is presently attracting high interest and its value to humanity is great, as they can save thousands of lives if proper evacuation and sensitization of the community are carried out prior to this event.

The study of the type, frequency and size of earthquake in a region over a period of time is termed its seismicity. Seismologist uses various tools for this analysis, the most important of this is the seismograph machine (or seismometer) which detects and records seismic waves. For a region's seismicity, several factors are considered: Like in [5] air ionization around rock surface are studied, and useful observation indicates some changes prior to earthquakes. Some other factors considered include; geology of the area, location of faults, the earthquake history of the area, the previous earthquake intensities, evidence for recent fault movement and other factors [6]. All these factors are useful for the prediction of an impending earthquake.

The goal of this work is to design an Artificial Neural Network (ANN) model that will give significantly high level of accurate generalization (prediction) for the time lag between the earthquake waves. This wave has two phases, the first phase (primary wave) experienced some minutes before the second (secondary wave) which is the destructive waveform. The neural network is trained with input data collected from seismological stations, and the performance of the model evaluated using statistical measures adopted in this field of study.

1.2 Motivation

This thesis work is driven by the need for a global and dependable prediction model with no false alarms. As many precursory changes are not global observations; like the migratory birds, aquatic animal and the domestic animals (mice, dogs and cats)

unusual behavior, since this animals may either be responding to an entirely different environmental factor or they may not even be found in some of the regions prone to earthquakes. Measuring the arrival time for a traveling earthquake waves will present a more dependable and universal forecast. This method will present a global measure since this pattern is seen in all earthquake wave signals.

1.3 Outline

The second chapter deals with the on-going works in earthquake prediction, the factors considered for various proposed prediction models. These factors are evidence integrated from variety of sources; physical measurements, seismic measurement, geological evidence, statistical information and animal behavior.

Chapter 3 introduces the artificial neural network (ANN) structure. The neural network design and how the various parameters in the structure can influence the training and test results of the input data.

In the fourth chapter, series of simulation results obtained using MATLAB are presented and the results are analyzed. The collected data is trained on several configurations of the ANN network and the best performing architectures are highlighted. Also their performance measure by using statistical tools is presented in this chapter.

Chapter 5 gives the conclusion to the result obtained from this research work and proposed model. The results highlight the accuracy and efficiency of the neural network prediction model. This chapter also states the possible future work on this model.

Chapter 2

LITERATURE REVIEW

Earthquake occurrence varies spatially; its prediction has been a goal of mankind for millennia [7]. Earthquake prediction means the accurate forecasting of the place, size and time of impending earthquakes [8]. Careful measurement of the movement along faults enables forecast for earthquake. These seismic activities are carried by certain physical measurements. The basic begins with measuring the changes in distance (geodetic), also creep-meters which are device to measure movement across a fault are used. In [9], a measure of the change in slope on earth's surface using a tilt-meter is considered. Changes in the properties of physical structures can also be measured; solid rocks are highly resistive but under excessive strain, they develop cracks and shatter thus allowing water to percolate through decreasing the resistivity, this change is monitored and can be used for earthquake prediction [6].

Some studies like in [10], [11] and [12] also consider behavioral activities (seismic-escape response) put up by some animals in response to the precursors to be helpful in earthquake prediction. These animals through natural selection are forced to develop anticipatory mechanism for predicting possible natural disasters, as survivor instincts far outweighs other behaviors like mating, breeding, and maintaining territory. The issue with the belief that certain animal do anticipate earthquakes is that it is poorly supported by evidence [12].

Another method is the "VAN" method that has attracted a very high level of debate. The name VAN is coined from the initials of three Greek scientist, Varotsos, Alexopoulos and Nomicos. They found that seismic electric signals (SES), which are variations in the earth's electric field occurs prior to an earthquake [13]. Depending on this SES's types, the earthquake can be predicted to occur within days to weeks [14], the doubt on this method is distinguishing between similar electric signals from other systems [15]. The researchers in [16] considered data from earthquakes of magnitude 3.5 and greater collected from 1970 to 2008 in Yunnan region (22-28^{o}N, 98 -104^{o}E), and this data were used to predict earthquakes in 1999-2008 and verified using the support vector machine (SVM), this also yielded good results.

For successful prediction of earthquakes, information on the place, time and magnitude are essential. Three different time frame grouping are also considered in earthquake prediction by scientist, there are; long term, intermediate and short term predictions. In the long-term prediction, which spans a period of ten to hundreds of years, seismologist assigns a movement budget, calculated through careful measurement of movement along faults, they find very limited use for public safety. Intermediate prediction spans from few weeks to few years (not up to ten years). In short-term prediction, specific information of the earthquakes time and location is given within minutes, weeks, or months and are therefore very useful for public safety and evacuation [17].

2.1 Earthquake Size and Distribution

The place, time and magnitude of earthquake are all serious consideration in earthquake prediction analysis. The magnitude of an earthquake is measured using the Richter scale, though there is the Mercalli scale, the Richter is the most common

standard used and it measures the magnitude on a logarithmic scale. This means that for every whole number increment on the magnitude scale, the ground motion amplitude as recorded by a seismograph goes up ten times and also 32 times more energy is released [18]. Table 2.1 gives the classification of earthquakes in terms of their magnitude.

Table 2.1: Earthquake magnitude classes (source UPSeis)

Class	Magnitude
Great	8.0 and higher
Major	7.0 - 7.9
Strong	6.0 – 6.9
Moderate	5.0 – 5.9
Light	4.0 – 4.9
Minor	3.0 – 3.9
Very Minor	< 3.0

Fortunately, as the magnitude of an earthquake increases its annual occurrence decreases considerably. Table 2.2 presents their effect and annual frequency of occurrence.

Table 2.2: Earthquake magnitude effect and annual frequency (source UPSeis)

Magnitude	Earthquake Effect	Average Annually
8.0 or more	Can totally destroy communities near the epicenter	One in 5-10 years
7.0 – 7.9	Causes serious damage	20
6.1 – 6.9	May cause a lot of damage in very populated areas	100
5.5 – 6.0	Slight damage to buildings and other structures	500
2.5 – 5.4	Often felt, but only causes minor damage	30,000
2.5 or less	Usually not felt, but can be recorded by a seismograph	900,000

It is also important to know that there exist a rough relationship between earthquake magnitude and the rupture length. Thus if we can predict the part of a fault that would rupture, then we can forecast the magnitude of an impending earthquake as illustrated in Table 2.3 [6].

Table 2.3: Relating earthquake magnitude with rupture length

Magnitude (Richter)	Rupture Length (miles)
5.5	3-6
6.0	6-9
6.5	9-18
7.0	18-36
7.5	36-60
8.0	60-120

2.2 The Earthquake Waves

Seismic waves are low frequency waves that give out acoustic energy. They result from explosion, earthquake or a volcano. Seismograph is used to detect and record

seismic waves and this seismic measurements are the basis for short-term prediction [19]. There are two basic types of seismic waves; the primary wave (P-wave) and secondary wave (S-wave). Though a third wave exists that is called the surface wave (this is the resulting wave formed when the P & S-waves combines at the surface). The point from where the seismic wave originates within the earth's crust is called the hypocenter, the region directly above the hypocenter on the earth surface is called the epicenter and these are around the earth's fault lines.

The Primary (P-wave) is the fastest of them, travelling at 1.6 to 8 kilometers per second in propagating medium depending on its density and elasticity [20]. When they pass through gasses, liquids and solids, their effect is to move this medium back and forth, example on rocks; they expand and contract the rock particles.

The Secondary waves (S-waves) also called shear waves travels slower than the primary-waves and causes a displacement perpendicular to its travel path. They travel with so much energy and so are the destructive earthquake waveform and can only travel through solid earth crust. It stops whenever it meets a liquid medium like the earth water bodies (the seas). Figure 2.1 gives a pictorial representation of these waveform.

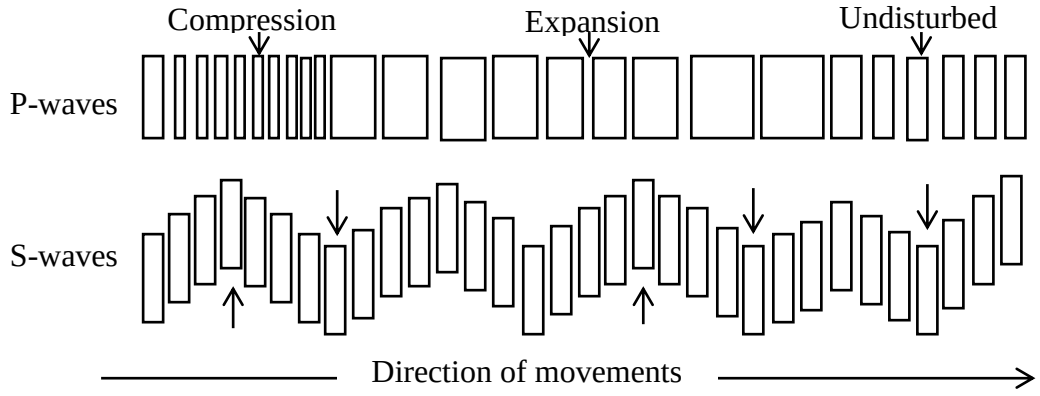

Figure 2.1: Schematic description of a travelling P & S-waves

This research work focus on the prediction of the arrival time of the S-wave after a P-wave has been detected (Ts-Tp) using artificial neural network. Figure 2 highlights how this time lag is obtained from a seismograph.

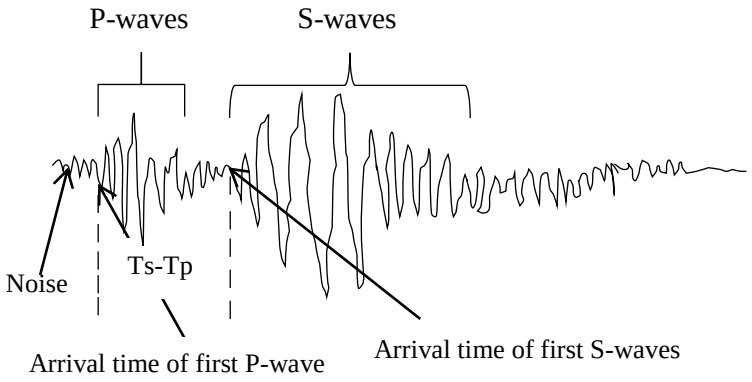

Figure 2.2: Depiction of P and S-wave time lag

2.3 Artificial Neural Network

A Neural Network is a massively parallel distributed processor made up of simple processing units that have a natural tendency for storing experimental knowledge and making it available for use. It is the type of Artificial Intelligence technique that mimics the behavior of the human brain [21]. The human brain is a highly complex structure, building up its own rules through experience that occurs over time. An artificial neural network resemblance to the brain is seen in these two capabilities; a neural network acquires knowledge through a learning process (training) and it has

interneuron connection weights (just like the synaptic cleft of biological neurons) which carry impulse (information) to other neurons and processing unit. These capabilities make a neural network a very reliable tool.

In the biological neuron the process of signal transmission begins by diffusion of chemicals across the synaptic cleft, which then travels along the dendrites to the cell body. In the cell body this information is stored until it exceeds a certain threshold, and then these impulse (inputs) is fired to other neurons which it is connected to along the axon. The simple perceptron of the neural network models this behavior in this way: first it receives the input values (x_o-x_n) with connection for each input having weights (w_o-w_n) ranging from 0-1. These inputs are summed and when it exceeds the threshold, the signal is then sent to the output [22]. The perceptron learns by adjusting its weight to approach the target output. The resemblance of the perceptron to the biological neuron is further highlighted in the Figures 2.3 and 2.4.

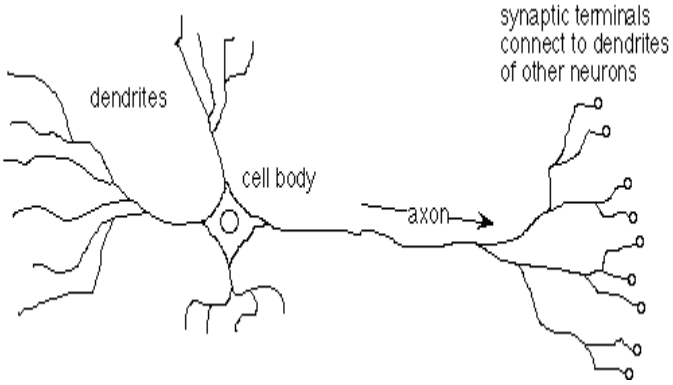

Figure 2.3: A simple biological neuron

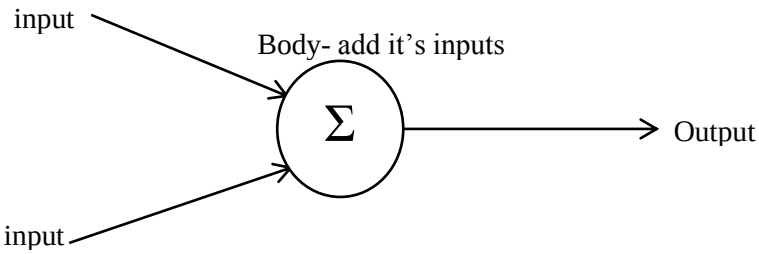

Figure 2.4: A simple perceptron

2.4 The Learning Process for Artificial Neural Network (ANN)

The ability of a neural network to learn is its primary significance, and then improve its performance in the process. This takes some time, entailing several iterations in the learning process.

Just like the human neurons, the artificial neural network learning follows this sequence; first it is stimulated, next it varies its free parameters because of the stimulation and then finally the artificial neural network responds.

There are two learning paradigms, which are;

(i) The Supervised learning and

(ii) Unsupervised learning

2.4.1 The Supervised Learning

The supervised learning can also be termed ‘learning with a teacher’. Illustration for this kind of learning uses a teacher. The teacher is believe to have full knowledge of the system, this knowledge is given in a set of input-output mapping, but the neural network does not know this. A training process for the teacher and the neural network is fixed; the teacher is expected to provide the desired response of an input set to the neural network for that training. This desired response is the optimum action expected of the neural network. Errors may still exist (the error for the system is the difference the desired response and the actual response), so the neural network tries to adjust its parameters base on the input vector and error signal iteratively, with the aim of making the network emulate the teacher. Thus, the knowledge of the system is transferred from the teacher to the neural network to a certain degree measured with statistical tools. When the neural network is trained to a satisfactory level, the teacher can now leave the neural network to completely deal with the system itself [21].

2.4.2 Unsupervised Learning

Under this learning method, the neural network is not taught by any teacher. The relationship between the input and the target values are not known and the training data set contains only input values, so there is need for right selection of examples for specific training. Usually the examples are selected using similarity principle [22].

Unsupervised learning looks into the way systems learn to represent input patterns in ways to reflect the structure of the entire collection of the inputs. For this method of learning, there are no explicit target outputs associated with the inputs [23].

Unsupervised learning cases are much more common in the human brain than supervised learning. For instance we have around 106 photoreceptors in each of the eye with their activities changing continuously with the visual world around to provide the information available to identify objects by their color, shape and distance without any prior supervised learning. Unsupervised learning works with observable patterns input patterns.

This thesis work uses a supervised learning strategy.

Chapter 3

METHODOLOGY

This design modeling and implementation is carried out using the Neural Network tools on MATLAB software. The MATLAB neural network toolbox provides a range of function for modeling non-linear complex systems. This toolbox supports supervised learning with feedforward networks, radial bias networks and dynamic networks. Also, it supports unsupervised learning with the self-organizing maps (SOM). This design implementation is better on MATLAB because of its easy matrix manipulation, implementation of algorithm, plotting of data, interfacing with programs in other languages and good user interface.

The goal of this work is to successfully design a model to predict P-wave and S-wave arrival time lag. The measured output which is the expected time lag from the arrival of the primary wave to that of the secondary wave will be tested and compared against real data from past earthquakes. Thus, this research work begins with data collection.

3.1 Data Collection

This is the first stage of this research. There are various techniques for collection of data used for research study purpose. This primary data can come from one or more of the following sources;

(i) Observations

(ii) Questionnaires and Surveys

(iii) Interviews

(iv) Focus groups

(v) Oral history and Case studies

(vi) Documentation and Records

In this work, documentation and records method was used. The collected data were gotten from the World Data Center for Seismology, Beijing: http://www.csndmc.ac.cn/wdc4seis@bj/earthquakes/csn_phases_p001.jsp. The design only worked with those data from January, 2012 to August, 2014. A total of 1478 readings were sampled, and they were all of the magnitude range of 6.0-7.0 on the magnitude scale. The data is then split into two sets; the first is the training set which is made up of data collected in 2012 with 1178 data sets from 58 cases across the globe, while the second is the test data collected in 2014 and 300 data sets from 28 cases were considered. This makes a total of 86 earthquakes cases that were analyzed in this study. This collected data is what is fed to the designed Neural Network MLP model.

3.2 The Multilayer Perceptron (MLP)

The MLP consist of three layers; the input layer, the hidden layer and the output layer as shown in Figure 3.1. The inputs propagate through the network layer by layer. Supervised learning is the learning method adopted for training in MLP and it uses the error back propagation algorithm which is the learning rule base on error correction.

There are basically two passes through the layers of the network in the error back propagation learning: the forward pass and the backward pass. The input nodes

receive the input vectors in the forward pass and propagate it layer by layer through the network. All the synaptic weights remain fixed in the forward pass. For the backward pass, the synaptic weights are then adjusted following an error correction rule; the rule adopts a propagation of the error signal backwards against the direction of the synaptic connections of the network, which then adjust these weights. This is the idea behind the error back propagation which is most times simply referred to as back propagation.

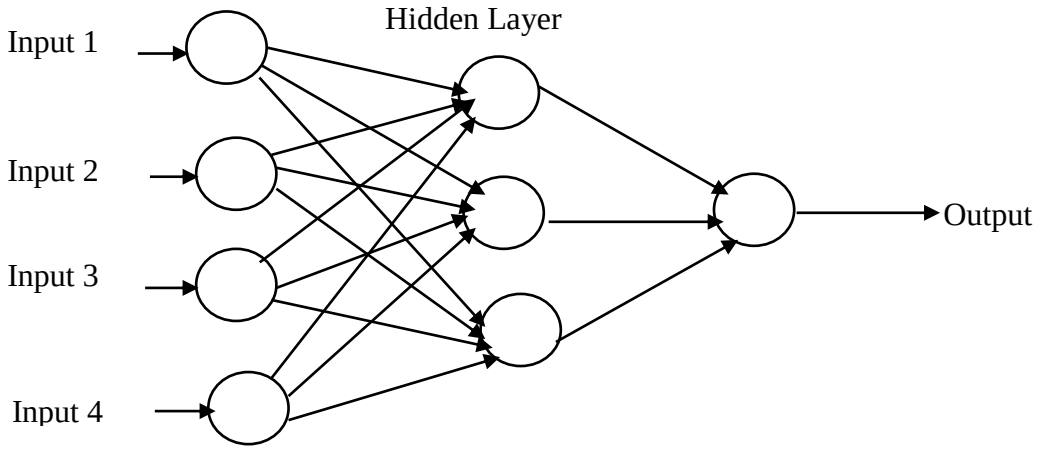

Figure 3.1: The MLP architecture

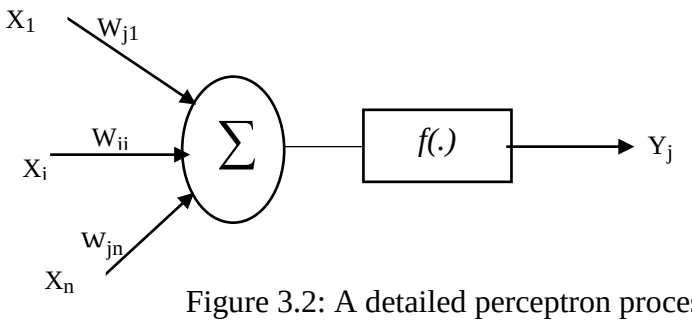

Figure 3.2: A detailed perceptron process

In Figure 3.2, the input neurons buffers the inputs x_i ($x_1, x_2, \ldots, x_i, \ldots x_n$) to the neurons in the hidden layer. Summation of inputs is done in each neuron j of the hidden layer, where also, these inputs are weighted with the interneuron connection weights w_{ji}

and the output y_j computed as a threshold function of the sum. The output neuron performs same computation.

$$y_j = f\left(\sum_{i=1}^{n} w_{ji}\, x_i\right) \tag{3.1}$$

The transfer function f can be a sigmoidal, hyperbolic or a simple threshold function. The selected transfer function gives extra information for the back propagation training algorithm. In the MLP structure, the threshold function is a continuous derivative. The goal is to minimize the error function, which is achieved by finding the squared error of the network.

In backpropagation, which is a gradient descent algorithm and adopted in the MLP training, the training weights are adapted as follows:

$$\Delta w_{ji} = \eta \delta_j x_i \tag{3.2}$$

The parameter η is the learning rate, it is user designated and it determines the level of modification to the link weights and node biases base on the change rate and direction.

A "momentum" term is added to help the network skip over the local minima and successfully reach the global minimum, while still maintaining the change rate and direction. This is adopted into the weight update equation as shown below:

$$\Delta w_{ji}(t+1) = \eta \delta_j x_i + \mu \Delta w_{ji}(t) \tag{3.3}$$

For the output neurons,

$$\delta = \left(\frac{\partial f}{\partial netj}\right)\left(y_j^{(t)} - y_j\right) \tag{3.4}$$

For the hidden neurons,

$$\delta = (\partial f / \partial netj)\left(\sum_q w_{jq}\, \partial_q\right) \tag{3.5}$$

And training continues until the error function reaches a certain minimum.

The parameters considered for this prediction work are;

1. The distance (D)
2. The azimuths (Az)
3. The magnitude (M)
4. The depth (Ep)
5. The measured time lag (Ts-Tp)

3.2.1 The Distance (D)

This is the representation of the distance from the earthquake's source and the seismological station (point of observation). This distance is given in degree which is the method for representing distances in spherical trigonometry, since the earth is a sphere, the shortest distance between two points on its surface is an arc and not a line. For this research work, the recorded distance tabulated in excel are given on the first column of the sheet.

3.2.2 The Azimuth (Az)

This is a clockwise measurement referenced from the earth's true north, also in units of degrees. This angle is measured clockwise starting with zero degrees at the true north. This is given on the second column of the excel sheet for all the cases

3.2.3 The Magnitude (M)

This magnitude is for the primary wave as recorded by the seismograph at the station. This first case of this experiment considers measurements for earthquake magnitude range 6-0 to 7.0 on the magnitude scale (strong earthquakes). For the second case, we are going to consider analysis for magnitude of 3.0 to 4.0 (Minor earthquakes) and those of 9.0 to 10.0 (great earthquakes).

3.2.4 The Depth (D)

This is the distance from the earthquake's hypocenter (wave origin) to the epicenter. This distance is given in kilometers and it is the last of the input values recorded on column 4 of the excel sheet.

3.2.5 The Time Lag (Ts-Tp)

This is time difference between the arrival of the first primary wave and the first secondary wave signals. Figure 2.2 gives a schematic depiction of how the time lag is computed. This time lag is recorded in seconds to express the difference in the arrival times. It is the only output value for the network.

3.3 Designing the Neural Network

The design for a neural network on MATLAB adopts certain systematic procedures. Depending on what is to be built and the nature and size of the collected data, certain parameters have to be adjusted. In general, these five basic steps in Figure 3.3 are followed;

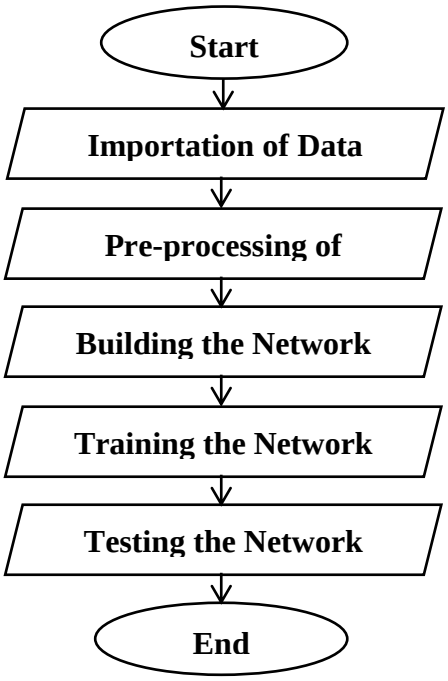

Figure 3.3: Flowchart for developing MLP using MATLAB

3.3.1 Importation of the data

The MATLAB function “xlsread” is used to import the data from the saved excel sheet. The data are first grouped in two sets; the training set and the testing set. The training set consist of 58 earthquake cases and 1178 data sets (stations) were considered, and while the testing set has 28 cases and from 300 stations worldwide.

3.3.2 Preprocessing of Data

In the preprocessing stage, normalization of the data set is applied. This is necessary considering the range of values of the parameter, since the parameters in consideration largely varies. For instance the azimuth measured in degrees has a minimum value of zero while the distance has a maximum value 80000 meters, therefore normalization is very necessary for this data set. When a variable of large values is mixed with those of small values, the network becomes confused with the values and this may lead it to reject the smaller values [24].

3.3.3 Building the Network

In MATLAB the built-in "newff" function is used to develop the MLP model. This tool allows the user to change the various parameters (the number of hidden layers, the number of neurons in each of these layers, learning rate, momentum constant, training function, the transfer function, and the performance functions). Initialization of the weights is automatic using these commands.

3.3.3.1 The number of hidden layers

For this design a single hidden layer is used. Provided there is sufficient number of hidden neurons in a single hidden layer, it can implement any multi-layer feed forward network.

3.3.3.2 The number of hidden neurons

This is a very important consideration for the network architecture and goes a long way in affecting the network's performance; too few hidden layer neurons will result in underfitting. This means that the number of neurons is not adequate to detect the signals in the data set. Also the use of too many hidden neurons has its own problems; first, the network will experience overfitting. Overfitting is when a network with high information processing capability is exposed to limited information in the training set that makes it insufficient to train all the hidden layer neurons. Secondly, it can unnecessarily slow the network.

In [25], a suggestion of the following rule-of thumb is given for selecting the number of hidden layer neurons;

1. Choose a number between the size of the output and the size of the input layer
2. Select the number to be 2/3 the size of the input layer, plus the size of the output layer.

3. The number of hidden neurons should be less than twice the size of the input layer.

It should be noted that this rules only provides a starting point for consideration. Following this, the number of hidden neurons was adjusted from two (2) up to seven (7) and then with 10 & 20 hidden neurons.

3.3.3.3 The Learning rate (η)

For this experiment we used values ranging from 0.1-0.9. It determines the level of modification to the link weights and node biases base on the change rate and direction.

3.3.3.4 The Momentum constant (μ)

The "momentum" term is added to help the network skip over the local minima and successfully reach the global minimum, while still maintaining the change rate and direction.

3.3.4 Training the Network

This is the stage where the network is taught how to generalize for the presented data set. For training, a set of data is presented to the network. These data sets consist of input-output pairs. The neural network then learns from the input and updates its weight, this is why it is termed a supervised learning, since the neural network is taught what the output should be from the input set introduced to it. The updated set is necessary since the main goal of the training process is to minimize the error. For feedforward networks, the performance is rated by the value of the mean square error, thus by adjusting the weights, what the neural network is trying to achieve is the possible minimum mean square error it can get to. This is user defined, for the neural network's goal in this work we used 0.01 and 0.001 in some cases. Slowly the neural network learns from the training set and improves on its generalization ability

so as to later yield result (network output) when it is fed with unseen data (testing input data) [21].

The nntool box (neural network toolbox) in MATLAB splits the data to three different set; the training set, the validation set and the testing set. From the training sets, the network is able to update the weight of the network during the training. The network also utilizes the validation set during the training, this set has just the input fed to the network, and the network is observed throughout the training. If the number of validation failure increases up to a particular value the training is stopped. When it stops, the network returns the minimum number of validation errors. Next is the test set which is used for testing the performance of the trained network. If this set reaches a minimum mean square error at a significantly farther iteration than the validation set, performance of the neural network will be unsatisfactory.

The architecture used has four (4) inputs neuron, one (1) hidden layer with hidden number of neurons varied from 3-7, 10 & 20. Each of this architecture was trained and tested with a learning rate (η) of 0.1 to 0.9. The network was observed while varying the number of neurons in the hidden layer, the momentum constant and also the learning rate for 9 different structures and the best performing structures selected. The training is stopped whenever any of the network's performance parameter is met.

3.3.5 Testing the Network

After training is completed, the network is tested with unseen data and the output is compared with the target (measured result). This is to check how well the network can generalize (predict output from the unseen inputs). Checking the performance is carried out using statistical measures on the obtained results: the root mean square

error (RMSE), the mean absolute error (MAE) and the mean bias error (MBE) are computed for the experimental result.

3.3.5.1 The Root Mean Square Error (RMSE)

The RMSE is obtained by squaring the difference between the measured output and the predicted value, and next is finding the average over the sample. Finally the square root of this is taken. It is the most commonly standard metric used to model error forecast in geoscience. Since the difference is squared, it is notice that the RMSE gives more weight to errors with larger absolute error values than those of smaller absolute error values. Thus, in analysis where large errors are not desired it is particularly very useful. It provides information on short term performance and is computed as;

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(t-O)^2} \tag{3.6}$$

where,

n= number of samples

t= target output (measured value)

O= network output (predicted value)

3.3.5.2 The Mean Absolute Error (MAE)

In words, this is the absolute value of the difference between the predicted value and the measured value, averaged over the sample. It measures accuracy for continuous variables, measuring the average value of the errors in a set of forecast. MAE weights the individual differences equally and is usually smaller in magnitude than the RMSE. It is computed as;

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|t - O| \tag{3.7}$$

If MAE=RMSE, it means all the errors in the sample are of the same magnitude.

3.3.5.3 The Mean Bias Error (MBE)

The MBE is the mean deviation of predicted values (produced from testing the network) to the measured value. It provides information on the long term performance of the model, the lower the value of the MBE the better is the long term model prediction.

$$MBE = \frac{1}{n}\sum_{i=1}^{n}(t - O) \tag{3.8}$$

For every simulation, the computed values are recorded in excel and used to evaluate the system performance. The performance of the trained network on tested data is the focus, it is most important measure of the training success.

Chapter 4

RESULTS AND DISCUSSION

The experimental set-up used MATLAB neural network toolbox on a personal computer (PC). The PC's is an Inspiron 15 3000 series, with a 4GB RAM, 64-bit operating system, and x64-based processor of Intel core i3 at 1.90GHz processing speed. The entire experiment took several iterations; different network variations are investigated to get the architecture with optimum performance.

The first architecture used a 4-2-1 structure (four inputs, two hidden neurons and one output). The test was carried out varying the learning rate (η) from 0.1 to 0.9 and saving the performance result. Also we made adjustment to the momentum constant (μ) and the results obtained at 0.01 & 0.001 are given in the table. The third parameter that was constantly varied is the number of hidden neurons (N). This study uses the RMSE, MAE & MBE values for the performance measure which are already explained in section 3.3.5 of this work. The computed data for the network architecture is tabulated in Table 4.1, also this setup process is maintained for all the architecture tested. The results are all tabulated and discussed in this section of the work

Table 4.1: Performance result using two hidden neurons

N=2							
μ=0.01	Test error statistics			μ=0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1241	0.0955	0.1241	0.1	0.2139	0.154	-0.0247
0.2	0.1391	0.1059	-0.007	0.2	0.2138	0.1549	-0.0276
0.3	0.1069	0.0944	-0.0234	0.3	0.2196	0.1585	-0.0223
0.4	0.1396	0.1148	-0.0822	0.4	0.2157	0.1561	-0.0239
0.5	0.1104	0.0977	-0.0222	0.5	0.2191	0.1578	-0.0211
0.6	0.1033	0.0902	-0.0578	0.6	0.2223	0.1464	0.0123
0.7	0.1085	0.087	-0.0239	0.7	0.22	0.1596	-0.0209
0.8	0.1027	0.0915	-0.031	0.8	0.2207	0.1484	0.0038
0.9	0.1118	0.0936	0.0253	0.9	0.2145	0.153	-0.0242
Average	0.1152	0.096733		Average	0.2177	0.1543	

For the experiment with the 4-2-1 and 4-3-1 architectures of Table 4.1 & Table 4.2, the best RMSE values are obtained at a learning rate of 0.8 & 0.7 respectively.

Table 4.2: Performance result using three hidden neurons

N=3							
μ=0.01	Test error statistics			μ=0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1126	0.0925	-0.0243	0.1	0.2009	0.1474	-0.0024
0.2	0.1148	0.0937	-0.0219	0.2	0.2117	0.1517	-0.0233
0.3	0.1085	0.0954	-0.0256	0.3	0.2056	0.1392	-0.0505
0.4	0.1108	0.089	-0.0253	0.4	0.2155	0.1546	-0.0223
0.5	0.1046	0.0927	-0.0252	0.5	0.2026	0.1491	-0.0037
0.6	0.108	0.0934	-0.004	0.6	0.2252	0.1556	-0.024
0.7	0.1067	0.0944	-0.0332	0.7	0.2164	0.1542	-0.0258
0.8	0.115	0.0946	-0.0235	0.8	0.2037	0.139	-0.0463
0.9	0.1091	0.0959	-0.0237	0.9	0.21	0.1483	-0.0343
Average	0.1100	0.0935		Average	0.2102	0.1488	

Table 4.3: Performance result using four hidden neurons

N=4							
μ=0.01	Test error statistics			μ=0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1072	0.0937	-0.0079	0.1	0.2132	0.1526	-0.0236
0.2	0.107	0.0931	-0.0488	0.2	0.2109	0.1497	-0.0093
0.3	0.1053	0.0931	-0.0272	0.3	0.2154	0.1555	-0.0226
0.4	0.1088	0.0955	-0.0222	0.4	0.2078	0.1468	-0.0268
0.5	0.1049	0.0886	-0.036	0.5	0.2144	0.1535	-0.0356
0.6	0.1092	0.0953	-0.0188	0.6	0.2118	0.1513	-0.0291
0.7	0.1153	0.0969	-0.0172	0.7	0.2109	0.1497	-0.0093
0.8	0.1078	0.0961	-0.0273	0.8	0.2078	0.1468	-0.0268
0.9	0.1037	0.0816	-0.0405	0.9	0.212	0.1519	-0.0324
Average	0.1066	0.0927		Average	0.2116	0.1509	

For the experiment in Table 4.3, the best result is seen to be at 0.1037 also with the momentum constant of 0.01. At the same momentum constant we also obtained the best RMSE value of 0.1003 in Table 4.4.

Table 4.4: Performance result using five hidden neurons

N=5							
μ=0.01	Test error statistics			μ=0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1003	0.0867	-0.0661	0.1	0.2173	0.1555	-0.0222
0.2	0.1138	0.0996	-0.0452	0.2	0.2193	0.1532	-0.0134
0.3	0.1167	0.0958	-0.0672	0.3	0.2142	0.153	-0.0259
0.4	0.1052	0.0924	-0.0201	0.4	0.2123	0.1515	-0.0197
0.5	0.1073	0.0879	-0.025	0.5	0.2162	0.1341	0.0455
0.6	0.1101	0.0979	-0.0264	0.6	0.2191	0.1582	-0.0232
0.7	0.1039	0.088	-0.0596	0.7	0.2103	0.1385	0.0059
0.8	0.1104	0.0879	-0.0206	0.8	0.2123	0.1537	-0.0424
0.9	0.1131	0.0924	-0.0249	0.9	0.2162	0.1556	-0.0258
Average	0.1089	0.0921		Average	0.2152	0.1504	

Table 4.5: Performance result with six hidden neurons

N=6							
μ=0.01	Test error statistics			μ=0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1065	0.0899	-0.0054	0.1	0.2193	0.1468	-0.0054
0.2	0.1072	0.0931	-0.0225	0.2	0.2123	0.1499	-0.0372
0.3	0.1053	0.0931	-0.0272	0.3	0.2137	0.1547	-0.0428
0.4	0.1105	0.0984	-0.0282	0.4	0.2122	0.1392	-0.0494
0.5	0.1109	0.097	-0.0201	0.5	0.2015	0.1296	0.0019
0.6	0.1083	0.0963	-0.0385	0.6	0.2195	0.1556	-0.0158
0.7	0.1065	0.0941	-0.0234	0.7	0.213	0.1531	-0.0595
0.8	0.1092	-0.0234	-0.0329	0.8	0.2233	0.1548	-0.0186
0.9	0.1065	0.0942	-0.0258	0.9	0.2208	0.1539	-0.0183
	0.1079	0.0814			0.2151	0.1486	

In Table 4.5 the learning rate of 0.3 and momentum constant of 0.01, the minimum RMSE value of 0.1053 is obtained. In Table 4.6, the best value was obtained at the learning rate of 0.1.

Table 4.6: Performance result with seven hidden neurons

N=7							
μ=0.01	Test error statistics			μ=0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1049	0.0915	-0.0553	0.1	0.2175	0.1387	-0.0318
0.2	0.1202	0.0995	-0.0236	0.2	0.2161	0.1562	-0.0244
0.3	0.1157	0.0962	-0.025	0.3	0.2162	0.156	-0.0457
0.4	0.1311	0.0991	-0.0373	0.4	0.2113	0.1497	-0.0395
0.5	0.1143	0.0937	-0.0251	0.5	0.2126	0.1475	-0.0431
0.6	0.1167	0.0959	-0.0253	0.6	0.2155	0.1574	-0.0541
0.7	0.1104	0.0979	-0.0247	0.7	0.2152	0.153	-0.0284
0.8	0.1052	0.0935	-0.0252	0.8	0.2222	0.1528	-0.0193
0.9	0.1075	0.0938	-0.0163	0.9	0.2257	0.1571	-0.0181
	0.114	0.0957			0.2169	0.152044	

Table 4.7: Performance result with ten hidden neurons

N=10							
μ=0.01	Test error statistics			μ=0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1188	0.0998	-0.0439	0.1	0.2142	0.1497	-0.0252
0.2	0.1052	0.0841	-0.0463	0.2	0.2268	0.162	-0.0329
0.3	0.1045	0.098	-0.0202	0.3	0.2262	0.1629	-0.0228
0.4	0.1092	0.0951	-0.0217	0.4	0.2172	0.1568	-0.0305
0.5	0.1047	0.0918	-0.0227	0.5	0.2267	0.1625	-0.025
0.6	0.1178	0.1009	-0.0676	0.6	0.2272	0.1602	-0.0138
0.7	0.1121	0.1087	-0.0252	0.7	0.2238	0.1565	-0.0404
0.8	0.2086	0.1591	-0.0928	0.8	0.2086	0.1591	-0.0928
0.9	0.2181	0.1565	-0.0334	0.9	0.2181	0.1565	-0.0334
Average	0.1321	0.1104		Average	0.2210	0.1585	

These further test with the 4-10-1 & 4-20-1 architectures of Table 4.7 and Table 4.8 respectively, are to show the resulting effect on our performance values when the number of hidden neurons are increased. These networks gave higher average values as compared to the others.

Table 4.8: Performance result with twenty hidden neurons

N=20							
μ =0.01	Test error statistics			μ =0.001	Test error statistics		
η	RMSE	MAE	MBE	η	RMSE	MAE	MBE
0.1	0.1132	0.0971	-0.0414	0.1	0.224	0.1576	-0.0047
0.2	0.1426	0.1167	-0.0265	0.2	0.2184	0.1525	-0.015
0.3	0.1108	0.0956	-0.0188	0.3	0.2225	0.1518	-0.012
0.4	0.2499	0.1085	0.0027	0.4	0.2228	0.1503	0.0117
0.5	0.1268	0.1093	-0.0322	0.5	0.2287	0.1617	-0.0118
0.6	0.1261	0.1077	-0.0501	0.6	0.2306	0.1596	-0.0258
0.7	0.1091	0.096	-0.0401	0.7	0.2151	0.1528	-0.0139
0.8	0.1217	0.0984	-0.0124	0.8	0.2195	0.1455	-0.0308
0.9	0.1096	0.0796	-0.0193	0.9	0.214	0.1484	-0.022
Average	0.1344	0.1010		Average	0.2217	0.1534	

The RMSE and MAE values are seen to be a lot better using a momentum constant of 0.01 (μ=0.01). Also from the tables, we can deduce that the best overall RMSE value was at five (5) hidden neurons with a learning rate of 0.1 and at the momentum constant of 0.01, while the test using the three (3) hidden neurons gave the best RMSE with the momentum constant of 0.001 value at a learning rate of 0.1.

The average values were computed just to show the effect of changing the learning rate at each number of hidden neurons for the two momentum constants considered (μ=0.01 & 0.001). For μ=0.01, the architecture with four (4) hidden neurons gave the least average over the nine (9) learning rate of 0.1-0.9 (which is considered the best value), while the architecture with three (3) hidden neurons had the least average RMSE value for μ=0.001. Figure 4.1 shows this comparison.

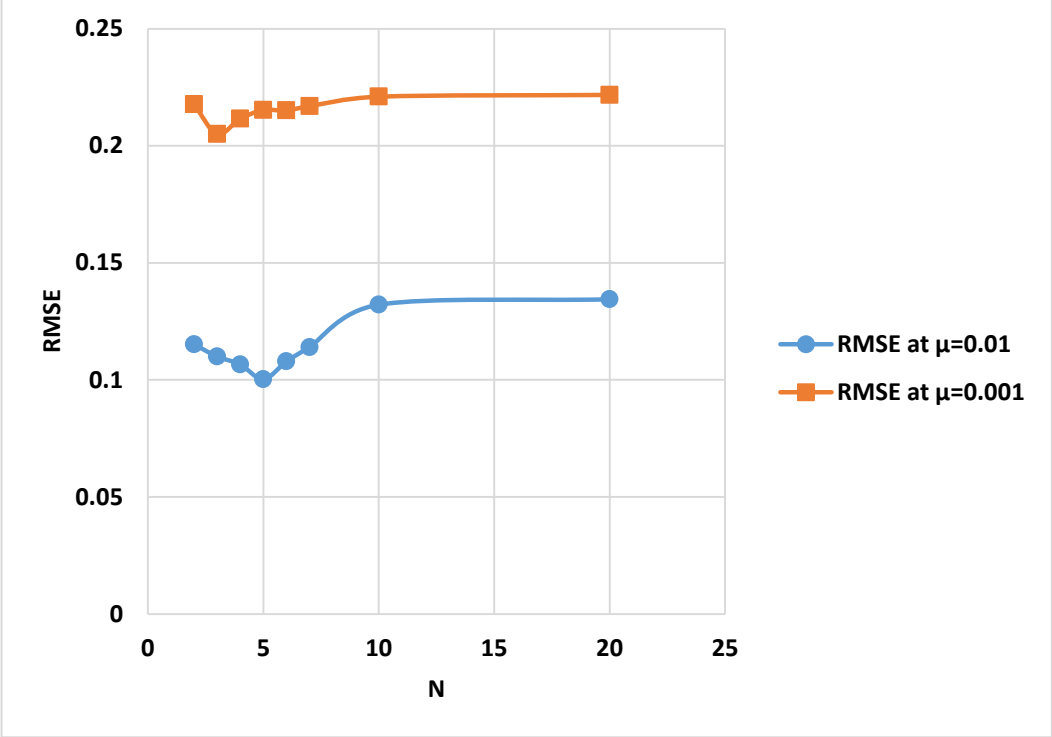

Figure 4.1: RMSE values for different number of hidden neurons

We also notice how the difference RMSE values and the MAE values of the various architecture maintained a range of 0.01-0.04 for the different architectures using

μ=0.01, and 0.05-0.07 for μ=0.001. The RMSE value is always larger than the MAE value, which is consistent with standard observation, and also the smaller the difference the lesser is the variance in the individual errors. Thus for this experiment, we can conclude that using a momentum constant of 0.01 gives a better result than using 0.001.

Another observation from the table is the fact that the RMSE and MAE values were considerably consistent from N=2 to N=7, this indicates that optimal performance of the network was within this range.

The performance plot of the training (which is called by the 'plotperf' command in MATLAB) also indicates a good training. The performance plots selected for N=5 for μ=0.01, and N=3 for μ=0.001 is shown in the figures 4.2 and 4.3.

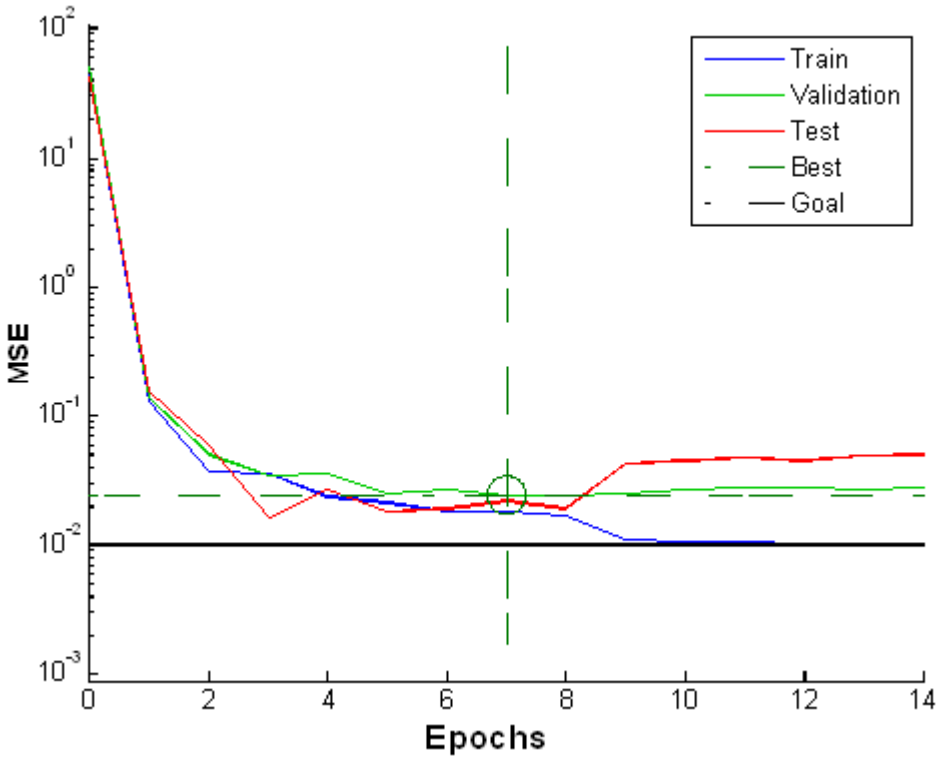

Figure 4.2: The performance plot for N=5 at μ=0.01

The plot in Figure 4.2 shows that the best validation was reached at epoch 7, even though the training still proceeded for seven more epochs and with a mean square error (MSE) of 0.024839. Also the test plot is noticed to be similar with that of the validation, this indicates that there is no major problem with the training.

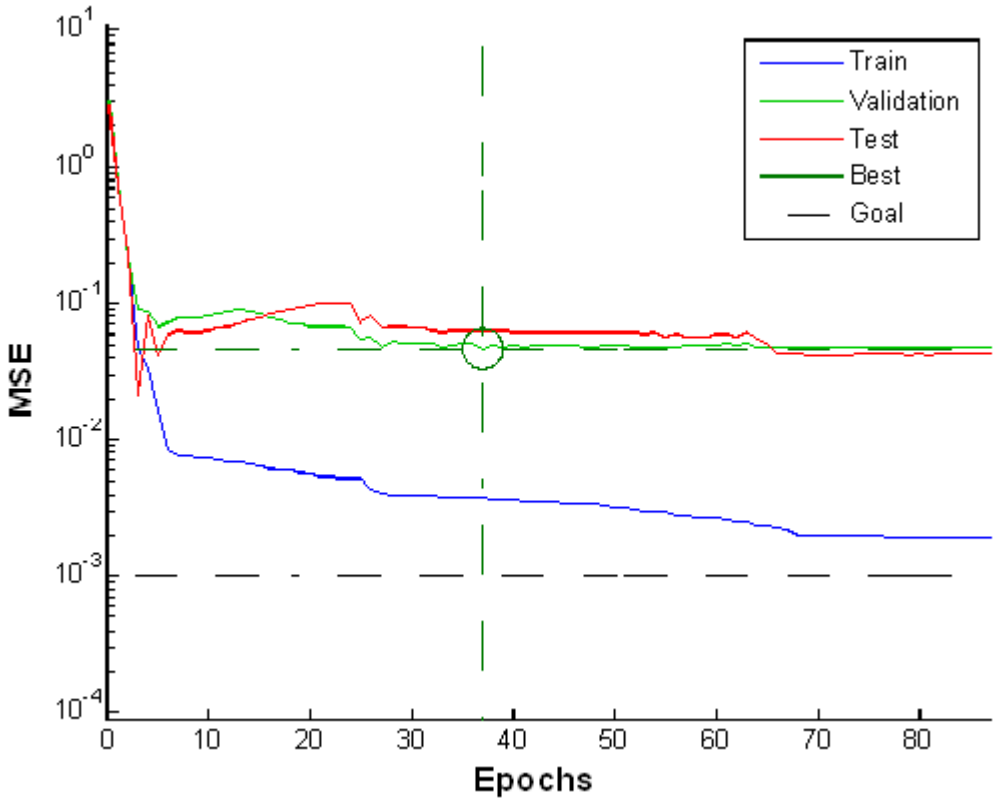

Figure 4.3: The performance plot for N=3 at μ=0.001

From Figure 4.3 we see how the curves are very similar and also how the best validation s at a mean square error (MSE) of 0.047033 at epoch 37 (the circled point in the graph).The training also continued for 50 more epochs before stopping.

Another performance tool which was observed is the regression plot, this gives the linear regression between the output and the target points for the experiment.

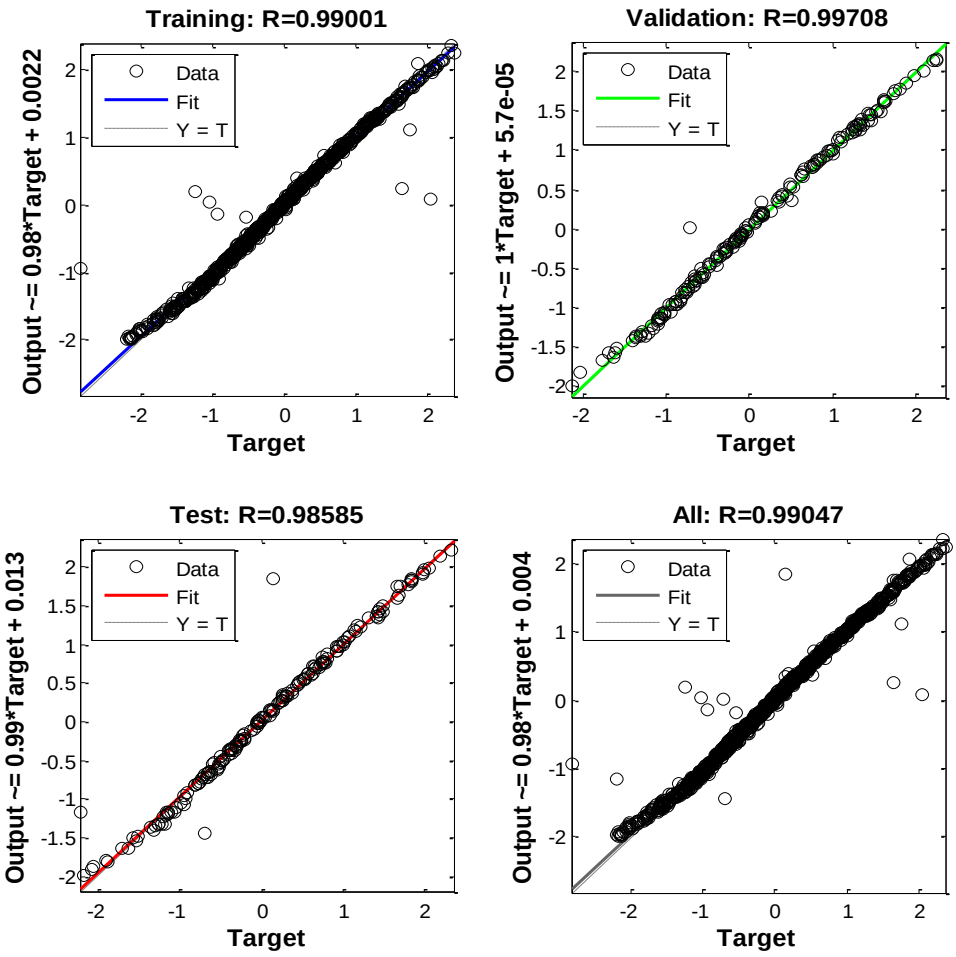

Figure 4.4: Schematic of the regression plot at N=3 for μ=0.01

The regression plot of Figure 4.4 shows the regression value (R) for the training, validation, test and all combined with values of 0.99001, 0.99708, 0.98585 & 0.99047 respectively.

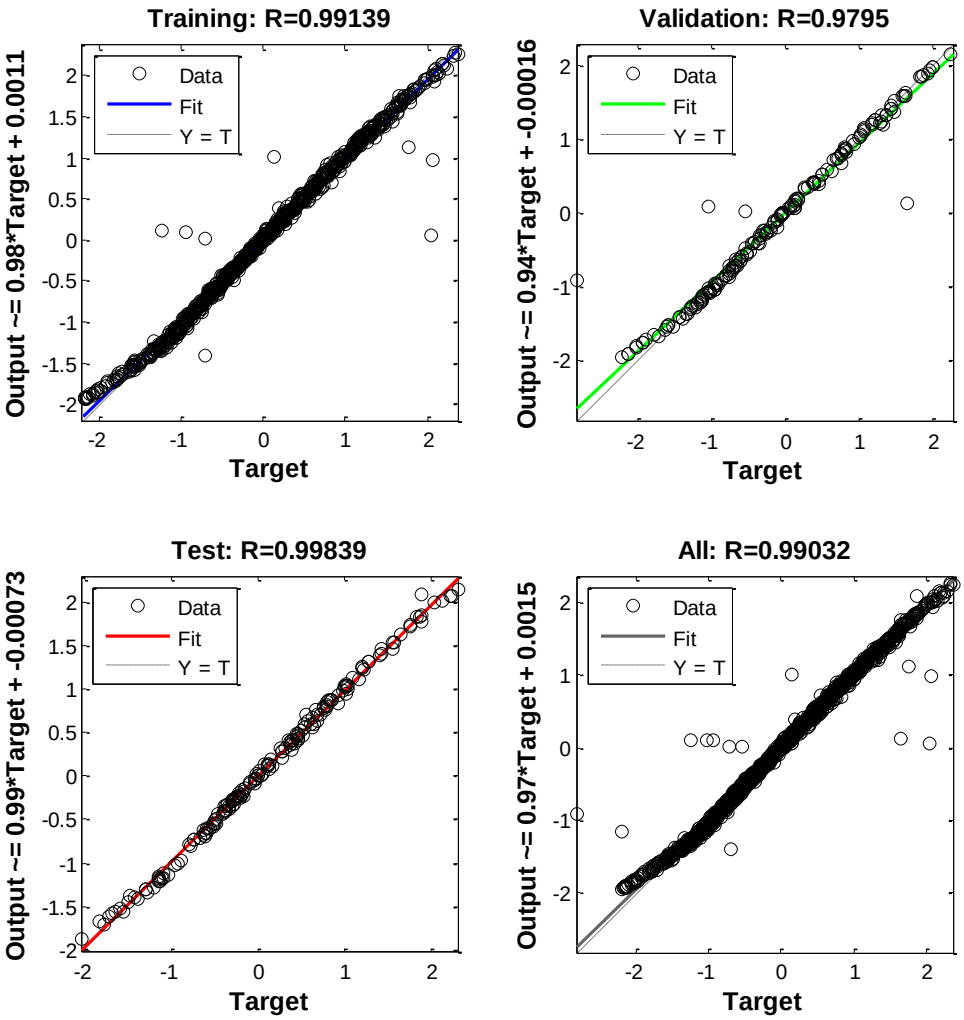

Figure 4.5: Schematic of the regression plot at N=3 for μ=0.001

In Figure 4.5, the R-values are all greater than 0.9, this is an indication of a very good fit for the training data and it shows how very close the output of the network and the target (measured) values are. The few scattered plot indicates that those points show poor fit. The solid line from the origin represents the best fit linear regression line. The dotted line represents a perfect result, values where output equals target.

The training stops whenever any of the performance goal is met.

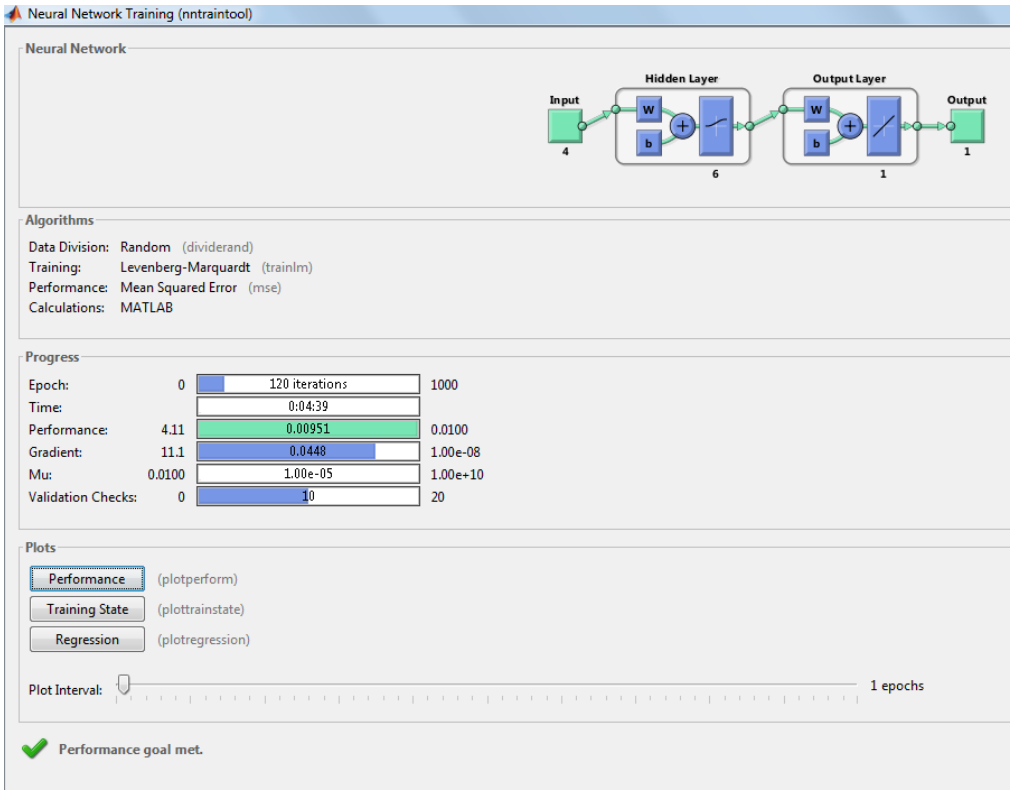

Figure 4.6: Overview of the MATLAB nntool training

Further testing was carried out but this time to check the generalization of the system for inputs outside the training set. The initial test input with patterns for earthquakes of magnitude 6.0-7.0 were now replaced with patterns of magnitude 3.0-4.0 and 9.0-10.0. This is to show the performance of the model for lower and higher magnitude earthquakes. For this analysis, only the 4-5-1 (the notation means, 4 inputs, 5 hidden neurons and 1 output) & 4-3-1 architecture which had the best performance for the initial patterns were considered and the results tabulated as shown.

Table 4.9: Test results with external values

Earthquake Magnitude(3.0-4.0)						
Architecture	Test error statistics at μ=0.01			Test error statistics at μ=0.001		
	RMSE	MAE	MBE	RMSE	MAE	MBE
4-5-1	0.2333	0.1665	-0.0428	0.2147	0.1554	-0.0428
4-3-1	0.2423	0.179	-0.0832	0.2243	0.1527	-0.1527
Earthquake Magnitude (9.0-10.0)						
Architecture	Test error statistics at μ=0.01			Test error statistics at μ=0.001		
	RMSE	MAE	MBE	RMSE	MAE	MBE
4-5-1	0.2227	0.1595	-0.0212	0.2207	0.1752	-0.0612
4-3-1	0.2958	0.1692	-0.0109	0.2225	0.1614	-0.242

The performance statistic measure gives satisfactory results. This indicates that the model can also do well for patterns outside the ones used in the training so long the parameters are unchanged as seen in Table 4.9.

It had been observed that in the previous patterns, the RMSE value increases when we test using the momentum constant of 0.001 compared to the test with value of 0.01, but for these two newly introduced sets (for the earthquakes of lower and those of higher magnitudes) we notice it is the reverse. Also direct comparison of the statistical measures gives a better performance for the previous pattern, this is the obvious expectation, since the training was carried out with patterns of the same magnitude.

Another observation was in the fitting of the error curve. Though the curve shape is considered satisfactory, but in comparison, the previous pattern still has a better fit.

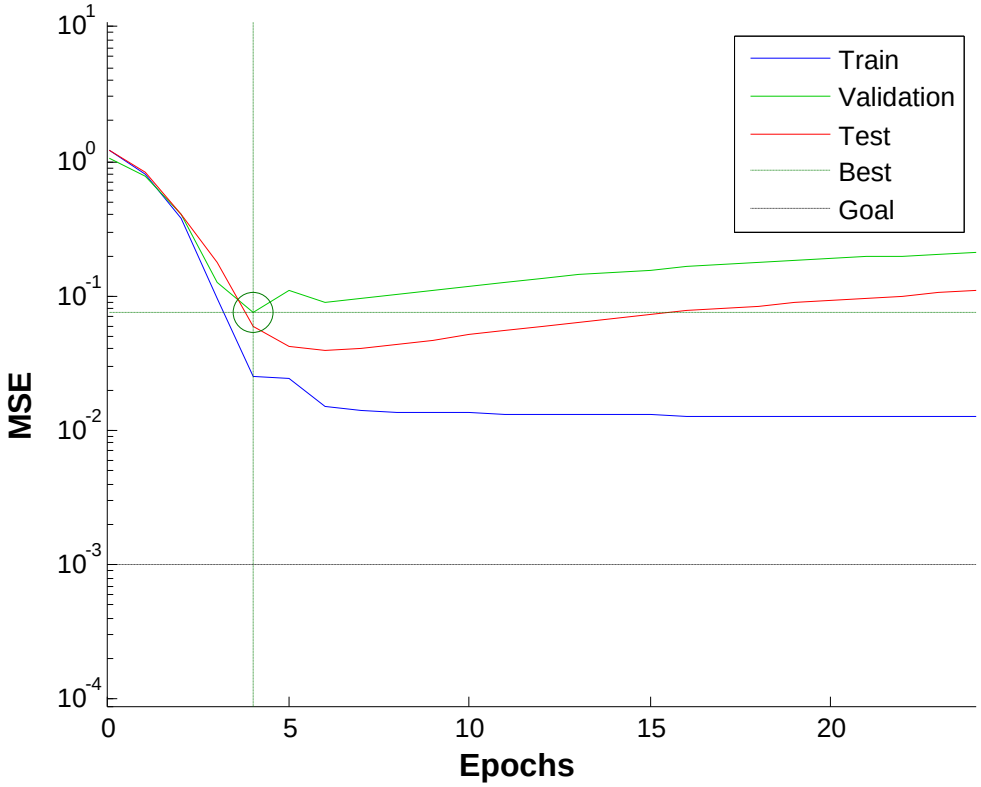

Figure 4.7: Performance plot with external input patterns of magnitude (3.0-4.0)

Figure 4.7 plot shows the best performance to be at epoch 4 with a mean squared error (MSE) of 0.075756. We see that the MSE value this time is much higher than those of the first cases with test samples sharing same magnitude with those of the training data.

Chapter 5

CONCLUSION

This research work presents a novel idea of the possibility of earthquake prediction using the time-lag between primary and secondary earthquake waves (P & S-waves). The neural network was trained to a level that it was able to recognize a good relationship between the input parameters (distance, azimuth, depth of the source wave and the magnitude of the received primary wave) & the output (time-lag between the P & S-waves).

The model proves to be dependable considering the performance measure. For further designs, the simulation with a momentum constant of 0.01 should be applied for training the set, since this as seen in the tables gave a better result. Also for practical systems, it will be advisable to conduct the training on all range of earthquake data available, so that optimum results can be obtained with all the data set, because we observed the model giving better results when tested with data range with which it had earlier been trained with compared to the result from sets outside this range.

It is also advisable to run the trainings several times for each set of parameters. This is a way of helping the system to achieve the global minimal at least in one of the running times. The average values after running several tests should also be recorded.

This model can be built into seismograph machines or used to design other sensor nodes. This makes the model an idea that is not just restricted to a region, as it can be applied in all regions experiencing earthquakes to reduce its effects. For instance, if a 10-30 minutes prior warning is received, it will go a long way to mitigate the effects: flights to such regions can be cancelled, the people around can be informed to avoid places with crowded buildings, place large objects on lower shelves, secure objects such as books, lamps, framed photos and other objects that may become flying hazards with adhesives and hooks to keep them in place.

5.1 Future Works

Most of the readings were taken from seismic stations that are very far from the earthquake's epicenter. Future study will consider the deployment of smaller sensing machines or nodes like wireless sensor nodes (WSNs) distributed around the earth's fault lines.

Also the training set will accommodate data for all magnitude range so as to be able to obtain the optimum result for any testing set.

REFERENCES

[1] Peter Molnar. "Earthquake Recurrence Interval and Plate Tectonics." *Bulletin of Seismological Society of America*. Vol.69. no.1. 1979. pp. 115-133.

[2] Guojin L., Rui T., Ruogu Z., Guoliang X., Wen-Zhan S., & Jonathan M.L. "Volcanic Earthquake Timing Using Wireless Sensor Networks." *IPSN'13*. Philadelphia. 2013. pp. 91-102.

[3] Turcotte D.L., Smalley R.F., Chatelain J.L and Prevot R. "Fractual Approach to the Clustering of Earthquakes: Application to the Seismicity of the New Hebrides." *Bulletin of the Seismological Society of America*. vol.77. no.4. 1987. pp. 1368-1381.

[4] Wyss M. & Zuniga R. "Inadvertent changes in magnitude reported in earthquake catalogues: Influence on b-value estimate." *Bulletin of the Seismological Society of America*. vol.85. 1995. Pp. 1858-1866.

[5] Friedemann T. Freund, Ipek G. K., Gary C., Julia L., Mathew W., Jeremy T., & Minoru M. "Air Ionization at Rock Surface and Pre-Earthquake Signals." *Journal of Atmospheric and Solar-Terrestial Physics.* Vol.71. 2009. pp. 1824-1834.

[6] Earthquake Prediction Lecture notes (2008). pp. 1-14.

[7] Stefan Wiemer. “Earthquake Statistics and Earthquake Prediction Research.” *Institute of Geophysics*. CH093. 2000. Zurich, Switzerland. pp.1-12.

[8] Stuart Crampin. “Developing Stress-monitoring Sites Using Cross-hole Seismology to Stress Forecast the Times and Magnitudes of Future Earthquakes.” *Technophysics 338*. 2001. pp.233-245

[9] Rafig A., Tomas M., Christian A., Herbert K. & Keh-Jian S. “Monitoring of Landslides and Infrastructure with Wireless Sensor Networks in an Earthquake Environment.” *5th International Conference on Earthquake Geotechnical Engineering*. 2011. pp. 10-13.

[10] Joseph L. Kirschvink. “Earthquake Prediction by Animals: Evolution and Sensory Perception.” *Bulletin of the Seismological Society of America*. Vol. 90. 2000. pp.312-323.

[11] Adi Schnytzer & Yisrael Schnytzer. “Animal Modeling of Earthquake and Prediction Markets.” *Department of Economics and Life Sciences Bar Ilan University, Israel.* 2011.

[12] R. A. Grant & T. Halliday. “Predicting the Unpredictable; Evidence of Pre-Seismic Anticipatory Behavior in the Common Toad.” *Journal of Zoology*, 2010. pp. 1-9.

[13] P. Varotsos, N. Sarlis, E. Skordas & M. Lazaridou. "Additional Evidence on some Relationship between Seismic Electric Signals (SES) and Earthquake Focal Mechanism." *Technophysics 412*. 2006. pp. 279-288.

[14] Varotsos P. & Alexopoulos. "Physical Properties of the Variation of Electric Field of the Earth Preceeding Earthquakes." *Techtonophysics 110*. 1984. pp.73-98.

[15] Maria M., Marios A. & Chris C. "Artificial Neural Networks for Earthquake Prediction Using Time Series Magnitude Data or Seismic Electric Signals." *Expert Systems with Applications 38*. 2011. pp. 15032-15039.

[16] Maitha H., Ali H., Hassan A. "Using MATLAB to Develop Artificial Neural Network Models for Predicting Global Solar Radiation in Al Ain city-UAE." *UAE University*. 2011.

[17] Neeti Bhargava, V.K. Katiyar, M.L. Sharma & P. Pradhan. "Earthquake Prediction Through Animal Behavior." *NCBM*. 2009. pp. 159-165.

[18] Retrieved from, http://www.geo.mtu.edu/UPSeis/intensity.html. January 2015.

[19] Masashi Hayakawa & Yasuhide Hobara. "Current Status of Seismo-Electromagenetics for Short-Term Earthquake Prediction." *Geomatics, Natural Hazards and Risk.* vol.1, no.2, 2010. pp.115-155.

[20] Retrieved from http://www.teara.govt.nz/en/diagram/4404/the-earthquake-magnitude-scale. February, 2015.

[21] Simon Haykin (1998). *Neural Networks: A comprehensive foundation* (2nd ed.). pp. 63-66. Prentice hall international.

[22] Fiona Nielsen, (2001). "Neural Networks-algorithms and applications." *Niels Brock Business College*. pp. 1-19.

[23] Peter Dayam. "Unsupervised Learning." *The MIT Encyclopedia of the Cognitive Science.* pp. 1-7.

[24] Tymvois F., Michaelides S. & Skoutelli C. "Estimation of Solar Surface Radiation with Artificial Neural Network in Modeling Solar Radiation at the Earth Surface." Springer. 2008. pp. 221-256.

[25] Jeff Heaton (2011). *Introduction to the Math of Neural Networks* (1st edition). pp.71-73. Heaton Research.

APPENDICES

Appendix A: Sample of Collected data

Latitude (◦)	Longitude (◦)	Distance (◦)	Azimuth (◦)	Magnitude	Depth (m)	Ts-Tp (sec)
32.09	132.11	9.4	267	6.2	20000	104
32.09	132.11	10.9	311	6.2	20000	120
32.09	132.11	11.3	273	6.2	20000	128
32.09	132.11	11.9	327	6.2	20000	135
32.09	132.11	12.7	352	6.2	20000	147
32.09	132.11	12.8	338	6.2	20000	139
32.09	132.11	13.1	292	6.2	20000	154
32.09	132.11	13.8	242	6.2	20000	158
32.09	132.11	15.1	306	6.2	20000	174
32.09	132.11	15.3	269	6.2	20000	182
32.09	132.11	18.3	247	6.2	20000	206
32.09	132.11	18.7	304	6.2	20000	203
32.09	132.11	19.5	282	6.2	20000	227
32.09	132.11	19.7	302	6.2	20000	211
32.09	132.11	22.9	262	6.2	20000	238
32.09	132.11	23.7	287	6.2	20000	254
32.09	132.11	23.9	242	6.2	20000	261
32.09	132.11	24.2	275	6.2	20000	271
32.09	132.11	26.6	262	6.2	20000	265
32.09	132.11	27.1	295	6.2	20000	311
32.09	132.11	35.1	277	6.2	20000	332
32.09	132.11	35.1	302	6.2	20000	347
32.09	132.11	45.9	295	6.2	20000	401
14.58	73.59	149	335	6.7	80000	620
14.58	73.59	152.3	334	6.7	80000	227
14.58	73.59	153.4	351	6.7	80000	415

Appendix B: The nntool program code

```
Training_Data=xlsread('training.xlsx');

Testing_Data=xlsread('testing.xlsx');

Training_Set=Training_Data(1:1178,1:4);

Training_Target_Set=Training_Data(1:1178,5);

Testing_Set=Testing_Data(1:300,1:4);

Testing_Target_Set=Testing_Data(1:300,5);

net= newff(pn,tn,4,{'logsig'});

net=configure(net,pn,tn);

net.trainparam.min_grad=0.00000001;

net.trainParam.epochs=1000;

net.trainParam.goal=0.001;

net.trainParam.mu=0.1;

net.trainParam.lr=0.1;

net.trainParam.max_fail=50;

net=train(net,pn,tn)

yn=(Testing_Set');

zn=(Testing_Target_Set');

y=sim(net,yn);

Yj=mapstd('reverse',y,zs);
```