

AUTOMATIC PHASE PICKING OF SEISMIC SIGNALS USING ANN

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

MASTER OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

(With Specialization in Measurement & Instrumentation)

By

MURUGAVEL RAJA P



**DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE-247 667 (INDIA)**

JUNE, 2006

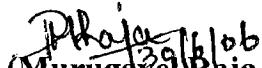
CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled “**Automatic Phase Picking of Seismic Signals Using ANN**” in the partial fulfillment of the requirements for the award of degree of M. Tech. in the Measurement & Instrumentation specialization, submitted in the Department of Electrical Engineering, IIT Roorkee, is an authentic record of my own work carried out under the guidance of **Miss Ambalika Sharma**, Assistant Professor, Electrical Engg. Deptt. and **Dr. M. L. Sharma**, Associate Professor, Earthquake Engg. Deptt, IIT Roorkee.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree or diploma of this institute or any other university/institute.

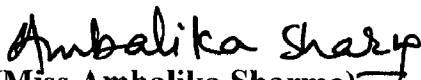
Dated: June, 2006


Place: Roorkee


(Murugavel Raja P)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.


(Miss Ambalika Sharma)
Assistant Professor
Electrical Engg. Deptt.
IIT Roorkee
247667 (India)


(Dr. M. L. Sharma)
Associate Professor
Earthquake Engg. Deptt.
IIT Roorkee
247667 (India)

ABSTRACT

An Artificial Neural Network (ANN) algorithm has been applied to the automatic phase picking of seismic signal. The aim of this work is to detect and accurately pick the onset time of seismic arrivals for a set of 3-component seismic data using ANN. A variety of features for signal detection and phase identification were analyzed in terms of sensitivity and efficiency. Comparing the performance of each feature in discriminating the *P* and *S* wave, eight features were selected as input attributes to ANN in which the first four attributes used for the ANN *P*-phase picker and next four attributes for ANN *S*-phase detector. They are as follows: (1) Ratio between short-term average and long-term average (STA\LTA), (2) Rectilinearity, (3) Ratio of maximum to minimum horizontal amplitude (hmxnm), (4) Long-axis incidence angle of polarization ellipsoid (Inc1), (5) Ratio between horizontal power to total power (Rh2t), (6) Planarity, (7) Ratio of horizontal-to-vertical power (Hvratp), (8) Short-axis incidence angle of polarization ellipsoid (Inc3).

These attributes were calculated in the frequency band of 1-10 Hz with a length of 0.5 sec moving window for seismic phase identification and another length of 4 sec moving window for *P*-wave picking. The detection and phase picking is achieved using Back propagation Neural Network (BPNN). The results of preliminary training and testing with a set of broadband seismic recordings shows that the ANN seismic phase picker can achieve a good performance in phase identification and onset-time estimation. In overall result, 72% correct rate of phase identification has been achieved by the both: the trained ANN *P*-phase detector and the trained ANN *S*-phase detector, and 53% of *P*-wave is precisely picked with onset time error less than 0.1 sec by trained ANN *P*-phase picker using STA \ LTA algorithm. The algorithms developed in the present study have been tested on the seismic data obtained from seismically active Garhwal Kumaon Himalayan region. The results provide accurate and robust automatic picks on a large experimental data.


CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in the dissertation entitled “Automatic Phase Picking of Seismic Signals Using ANN” in the partial fulfillment of the requirements for the award of degree of M. Tech. in the Measurement & Instrumentation specialization, submitted in the Department of Electrical Engineering, IIT Roorkee, is an authentic record of my own work carried out under the guidance of **Miss Ambalika Sharma**, Assistant Professor, Electrical Engg. Deptt. and **Dr. M. L. Sharma**, Associate Professor, Earthquake Engg. Deptt, IIT Roorkee.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree or diploma of this institute or any other university/institute.

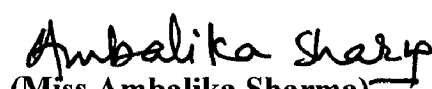
Dated: June, 2006


Place: Roorkee


(Murugavel Raja P)

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.


(Miss Ambalika Sharma)
Assistant Professor
Electrical Engg. Deptt.
IIT Roorkee
247667 (India)


(Dr. M. L. Sharma)
Associate Professor
Earthquake Engg. Deptt.
IIT Roorkee
247667 (India)

ACKNOWLEDGEMENTS

I take this opportunity to express my sincere gratitude to **Miss Ambalika Sharma**, Assistant Professor, Electrical Engineering Department IIT Roorkee and **Dr. M. L. Sharma**, Associate Professor, Earthquake Engineering Department IIT Roorkee for encouraging me to undertake this dissertation work as well as providing me all the necessary guidance and inspirational support throughout this work, without which this work would not have been in its present shape. Also I am grateful to Ms. Shipra Malik SRF, Earthquake Engineering Department, IIT Roorkee for providing the important seismic data for this dissertation work.

I am indebted to my friends specially Mr. Emjee Puthooran who took interest in discussing the problem and encouraging me.

Special sincere heartfelt gratitude to my mother and family members, whose sincere prayer, best wishes, concern, support, unflinching encouragement and thoughtfulness has been a constant source of strength and upliftment to me during the entire work.

Also, I cordially thank to Mr. S Ramakrishnan for being a constant source of encouragement during this work.

I thank all the members of Electrical Engineering Department for their help and encouragement at the hour of need.

Last, but not the least, I thank to Almighty God whose Divine Light provided me guidance, inspiration and strength to complete this work.

(Murugavel Raja P)

CONTENTS

	Page No
Candidate's declaration	i
Acknowledgements	ii
Abstract	iii
Contents	iv
List of Figures	vii
List of Tables	xi
Chapter-1 INTRODUCTION	1
1.1 Seismology	1
1.2 Seismic Waves	3
1.2.1 Body Waves	3
1.2.2 Surface Waves	4
1.3 Seismograph	6
1.4 Causes of earthquakes	9
1.5 Seismic Station	11
1.5.1 Analog Station	11
1.5.2 Digital station	12
1.6 Aim of the Dissertation	12
1.7 Introduction to Dissertation	13
Chapter-2 ARTIFIICAL NEURAL NETWORK	15
2.1 What is Artificial Neural Network?	15
2.2 History of Artificial Neural Network	15
2.3 Need for Artificial Neural Network	17
2.4 Introduction to ANN	17
2.5 Neuron modeling	19
2.6 Transfer Functions	20
2.6.1 Hard-Limit Transfer Function	21
2.6.3 Sigmoid transfer function	21
2.6.2 Linear transfer function	21
2.7 Summary of Artificial Neural Networks	22

2.8 Applications of Neural Network	23
2.8.1 Neural Network Applications in general	23
2.8.2 Neural network Applications in Earthquake Engineering	24
Chapter-3 BACKPROPAGATION NEURAL NETWORK	25
3.1 Backpropagation Neural Network	25
3.1.1 The Backpropagation algorithm	25
3.1.2 Step size	31
3.1.3 Local Minima	31
3.1.4 Learning Speed	31
3.1.5 Stopping Criteria	32
3.1.6 Network Size	33
3.1.7 Complexity of Learning	33
3.1.8 Generalization	
Chapter-4 POLARIZATION ANALYSIS OF THREE COMPONENT SEISMIC ARRAY DATA	36
4.1 Introduction	36
4.2 Attribute Selection for Seismic Arrivals	37
Chapter-5 AUTOMATIC PHASE PICKING OF SEISMIC SIGNALS USING ARTIFICIAL NEURAL NETWORK	49
5.1 Introduction	49
5.2 Identification of Seismic Phase Using ANN	50
5.2.1 P-Phase Identification Using ANN	50
5.2.2 Optimization and Training of the BPNNs	51
5.2.2.1 The Architecture of the BPNNs	51
5.2.3 Training Function (TRAINGDM)	53
5.2.3.1 Limitations and Cautions of TRAINGDM	54
5.2.4 Transfer or Activation Functions	54
5.2.5 Selecting Training Data Sets for P-Phase	55
5.2.6 Termination of Training	58
5.2.7 Phase Detection of P-Wave	58
5.2.8 S-Phase Identification Using ANN	63

5.2.9 Selecting Training Data Sets for S-Phase	63
5.2.10 Phase Detection of S-Phase	65
5.3 Picking of Onset Time for P-Wave	70
5.4 Software Details	73
Chapter-6 RESULTS	76
6.1 Automatic Phase Identification of P-Wave Using ANN	76
6.2 Automatic Phase Identification of S-Wave Using ANN	78
6.3 Automatic Phase Picking of P-Wave Using ANN	81
Chapter-7 CONCLUSIONS AND DISCUSSIONS	82
7.1 Conclusions	82
7.2 Discussions with Future Scope	83
REFERENCES	84

LIST OF FIGURES

Figure no.	Particulars	Page no
1.1	Earth as a filter	3
1.2	P-wave or primary wave	4
1.3	S-wave or secondary wave	4
1.4	Love wave	5
1.5	Rayleigh wave	5
1.6	A Simple Seismogram	6
1.7	The Basic Seismograph	7
1.8	The location of an epicenter	8
1.9	Seismic data file as displayed in SEISAN	9
1.10	The epicenter and focus of earthquake	9
1.11	Different types of faults due to which earthquakes occur	10
1.12	Flow chart representation of the dissertation	14
2.1	Structure of a Multilayer Artificial Neural Network	18
2.2	Single input neuron	19
2.3	Hard-Limit Transfer Function	21
2.4	Linear transfer function	21
2.5	Log-Sigmoid Transfer Function	22
3.1	The Backpropagation network	26
3.2	Search on the error surface along the gradient	30
4.1	Location of the station	42
4.2	Sample of a 3-Component waveform (normalized) for event 1 in table 4.1	43
4.3	Represents STA \ LTA attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in STA\LTA attribute	44
4.4	Represents Rectilinearity attribute time series for the seismograms in the above figure and the second figure an	44

	enlarged window represents the response direction of P and S wave in Rectilinearity attribute	
4.5	Represents Inc1 attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Inc1 attribute	45
4.6	Represents Hmxnm attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Hmxnm attribute	46
4.7	Represents Hvratp attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Hvratp attribute	46
4.8	Represents Inc3 attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Inc3 attribute.	47
4.9	Represents Rh2t attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Rh2t attribute.	48
4.10	Represents Planarity attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Planarity attribute.	48
5.1	Block diagram for the development of the ANN Seismic Phase detector. It consists of four preprocessors, four neural subnets, and one final decision neuron.	50
5.2	Structure of the ANN Phase Identification.	52
5.3	The training patterns of four attributes for P-arrivals and background noises. P-wave window was selected from 10 points before P arrival to 40 points after P-arrival. The background	57

	noises were extracted just before the signals arrive.	
5.4	Detection of P-phase by STA\ LTA algorithm, the first peak with maxima greater than 0.6 is identified as P-wave	59
5.5	Detection of P-phase by Rectilinearity, the first peak with maxima greater than 0.2 is identified as P-wave	60
5.6	Detection of P-phase by Hmxnm	61
5.7	Detection of P-phase by Inc1, the first peak towards the negative direction is identified as P-wave	62
5.8	The training patterns of four attributes for P-arrivals and background noises. P-wave window was selected from 10 points before P arrival to 40 points after P-arrival. The background noises were extracted just before the signals arrive.	64
5.9	Detection of S-phase byRh2t, the first peak with maxima greater than 0.1 is identified as S-wave	66
5.10	Detection of S-phase byPlanarity, the first peak the maxima peak with threshold grester than 0.6 is identified as S-wave	67
5.11	Detection of S-phase by Hvratp, the maxima peak with threshold greater than 0.4 is identified as S-wave	68
5.12	Detection of S-phase by Inc3, the first peak towards the negative direction is identified as S-wave	69
5.13	Input to the Neural Network for Phase Picking	71
5.14	Picking of P wave for a 5 min window seismogram. The lower panel shows an enlarged window around the P-arrival.	72
5.15	The time difference between picking by the ANN P-phase picker and by manual picking versus signal-to-noise-ratio.	73
5.16	Representation of Network Manager in Neural Network tool box	74
6.1	Percentage distribution of P-phase detection of ANN by STA \ LTA	76
6.2	Percentage distribution of P-phase detection of ANN by Rectilinearity	77
6.3	Percentage distribution of P-phase detection of ANN by Hmxnm	77

6.4	Percentage distribution of P-phase detection of ANN by Inc1	78
6.5	Percentage distribution of S-phase detection of ANN by Rh2t	79
6.6	Percentage distribution of S-phase detection of ANN by Planarity	79
6.7	Percentage distribution of S-phase detection of ANN by Hvratp	80
6.8	Percentage distribution of S-phase detection of ANN by Inc3	80
6.9	Percentage distribution of P-phase picking of ANN by STA \ LTA	81

LIST OF TABLES

Table no.	Particulars	Page no
1.1	List of some significant Earthquakes in India and its neighborhood.	1
4.1	Events used in this article, among these 10 events are used as training data for ANN P and S phase detector and P phase picker. All these events are used for testing.	41
4.2	Summary of responses of P and S waves of recording NTT and SRT in Table4.1	42

INTRODUCTION

1.1 SEISMOLOGY: Science of earthquakes

An earthquake is the rapid vibration of earth created by a sudden movement of large sections of rocks. Earthquakes are one of the most powerful natural forces that can disrupt our daily lives. Some of the recent devastating earthquakes are listed in Table 1.1. The science dealing with earthquakes is called **seismology**. Seismology is the study of earthquakes and seismic waves that move through and around the earth. A seismologist is a scientist who studies earthquakes and seismic waves.

Table: 1.1 List of some significant Earthquakes in India and its neighborhood.

Date	Epicenter		Location	Magnitude on Richter Scale
	Lat (°N)	Long (°E)		
1819 JUN 16	23.6	68.6	KUTCH,GUJARAT	8.0
1869 JAN 10	25	93	NEAR CACHAR, ASSAM	7.5
1885 MAY 30	34.1	74.6	SOPOR, J&K	7.0
1897 JUN 12	26	91	SHILLONG PLATEAU	8.7
1905 APR 04	32.3	76.3	KANGRA, H.P	8.0
1918 JUL 08	24.5	91.0	SRIMANGAL, ASSAM	7.6
1930 JUL 02	25.8	90.2	DHUBRI, ASSAM	7.1
1934JAN 15	26.6	86.8	BIHAR-NEPALBORDER	8.3
1941 JUN 26	12.4	92.5	ANDAMAN ISLANDS	8.1
1943 OCT 23	26.8	94.0	ASSAM	7.2
1950 AUG 15	28.5	96.7	ARUNACHAL PRADESH- CHINA BORDER	8.5

1956 JUL 21	23.3	70.0	ANJAR, GUJARAT	7.0
1967 DEC 10	17.37	73.75	KOYNA, MAHARASHTRA	6.5
1975 JAN 19	32.38	78.49	KINNAUR, HP	6.2
1988 AUG 06	25.13	95.15	MANIPUR-MYANMAR BORDER	6.6
1988 AUG 21	26.72	86.63	BIHAR-NEPAL BORDER	6.4
1991 OCT 20	30.75	78.86	UTTARKASHI, UP HILLS	6.6
1993 SEP 30	18.07	76.62	LATUR-OSMANABAD, MAHARASHTRA	6.3
1997 MAY 22	23.08	80.06	JABALPUR,MP	6.0
1999 MAR 29	30.41	79.42	CHAMOLI DIST, UP	6.8
2001 JAN 26	23.40	70.28	BHUJ, GUJARAT	6.9

The Earthquakes occur due to finite physical sources buried below the surface of the earth. These sources generate band-limited signals, which are recorded at the surface of the earth by seismic instruments. The medium through which these signals propagate, i.e., the Earth, also acts as a filter (*Figure 1.1*). The detection of these signals and noise at the recording station can be accomplished using signal processing techniques in time as well as frequency domain. These algorithms are known as earthquake signal/event detectors.

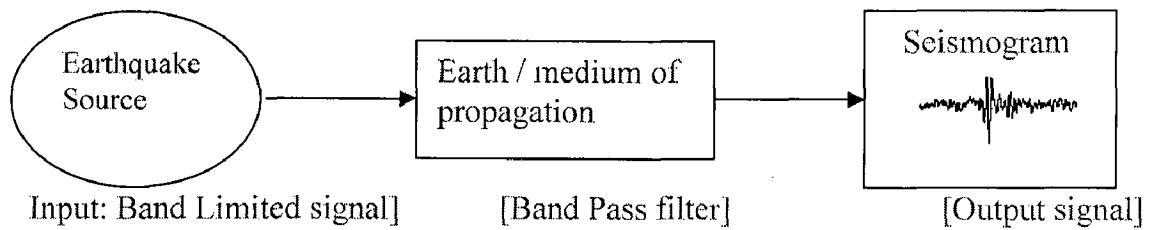


Figure 1.1: Earth as a filter.

The size of the earthquake is measured in terms of magnitude; generally the magnitude reported is Richter magnitude which is defined as the \log_{10} of the maximum amplitude recorded (in microns) at a distance of 100 km on standard Wood Anderson Seismograph. The effect of earthquake at any place is measured in terms of its intensity on a XII point MMI scale (Modified Mercalli Intensity). Thus the Richter scale measures the energy released in an earthquake by measuring the size of the seismic waves and the Mercalli scale measures the results of an earthquake, such as the shaking and damage that people actually feel and observe.

1.2 SEISMIC WAVES

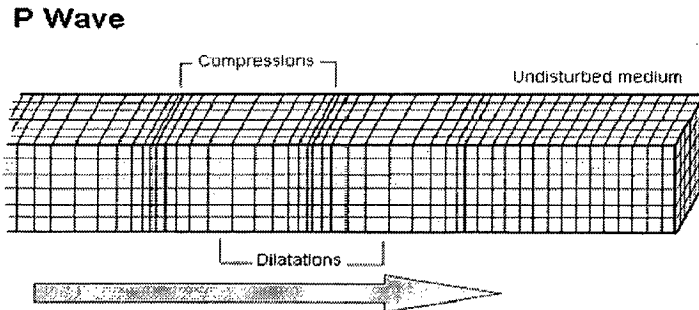
Seismic waves are the waves of intense energy caused by the sudden breaking of rock within the earth or an explosion. They represent the energy that travels through the earth and is recorded on seismographs.

The two main types of waves are **body waves** and **surface waves**. Body waves can travel through the earth's inner layers, but surface waves can only move along the surface of the planet like ripples on water. Earthquakes radiate seismic energy as both body and surface waves.

2.1 Body Waves

The first kind of body wave is the **P wave** or **primary wave** (*Figure 1.2*). This is the fastest kind of seismic wave. The P wave can move through solid rock and fluids, like water or the liquid layers of the earth. It pushes and pulls the rock, it moves through, just

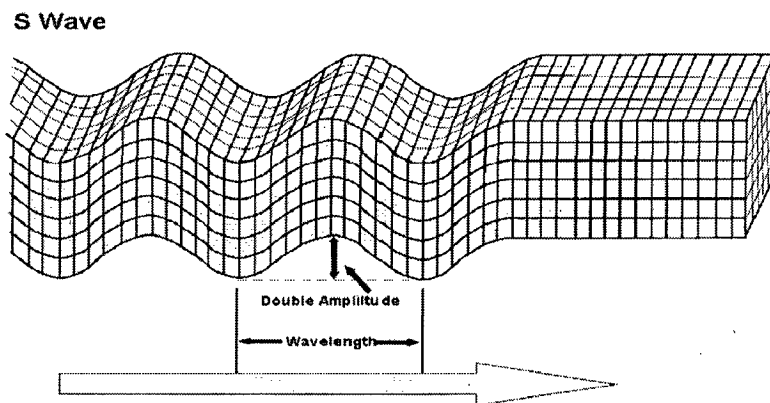
like sound waves push and pull air. *P* wave reaches the seismogram first and is recorded as the first seismic recording. Hence the detection of *P* waves for seismic warning systems is of utmost importance.



(The arrow shows the direction in which the wave is moving).

Figure 1.2: P-wave or primary wave

The second type of body wave is the **S wave** or **secondary wave** (*Figure 1.3*) that is the second wave felt in an earthquake. An *S* wave is slower than a *P* wave and can only move through solid rock. This wave moves rock up and down, or side-to-side.



(The arrow shows the direction in which the wave is moving).

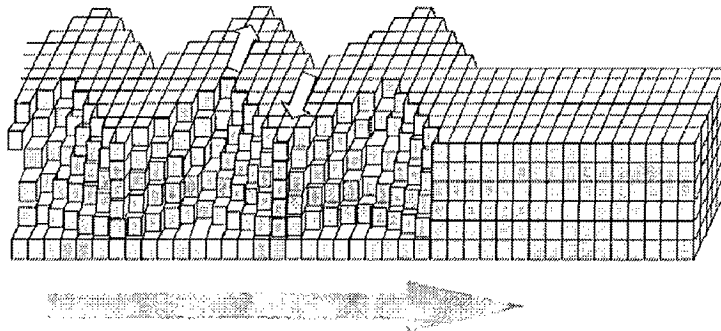
Figure 1.3: S-wave or secondary wave

1.2.2 Surface Waves

The first kind of surface wave is called a **Love wave** (*Figure 1.4*), named after A.E.H. Love, a British mathematician who worked out the mathematical model for this

kind of wave in 1911. It's the fastest surface wave and moves the ground from side-to-side (Shown in *Figure 1.4* with small arrows).

Love Wave



(The arrow shows the direction in which the wave is moving).

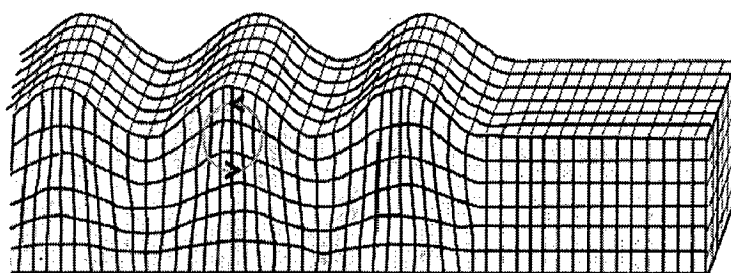
Figure 1.4:

Love

wave

The other kind of surface wave is the **Rayleigh wave** (*Figure 1.5*), named after John William Strutt, Lord Rayleigh, who mathematically predicted the existence of this kind of wave in 1885. A Rayleigh wave rolls along the ground (*Figure 1.5*) with rotating circle) just like a wave rolls across a lake or an ocean. Because it rolls, it moves the ground up and down and side-to-side in the same direction that the wave is moving. Most of the shaking felt from an earthquake is due to the Rayleigh wave, which can be much larger than the other waves.

Rayleigh Wave



(The arrow

direction in which the wave is moving).

shows the

Figure 1.5: Rayleigh wave

1.3 SEISMOGRAPH

Seismologists study earthquakes by observing the site of occurrence, assessing the damage caused by the earthquakes, and by using seismographs. A **seismograph** is an instrument that records the shaking of the earth's surface caused by seismic waves.

Most of the seismographs used today are electronic devices, but a basic seismograph (*Figure 1.7*) is made of a drum with paper on it, a bar or spring with a hinge at one or both ends, a weight, and a pen. The one end of the bar or spring is bolted to a pole or metal box that is bolted to the ground. The weight is put on the other end of the bar and the pen is stuck to the weight. The drum with paper on it presses against the pen and turns constantly. When there is an earthquake, everything in the seismograph moves except the weight with the pen on it. As the drum and paper shake next to the pen, the pen makes squiggly lines on the paper, creating a record of the earthquake. This record made by the seismograph is called a **seismogram**, which is shown in *figure 1.6*.

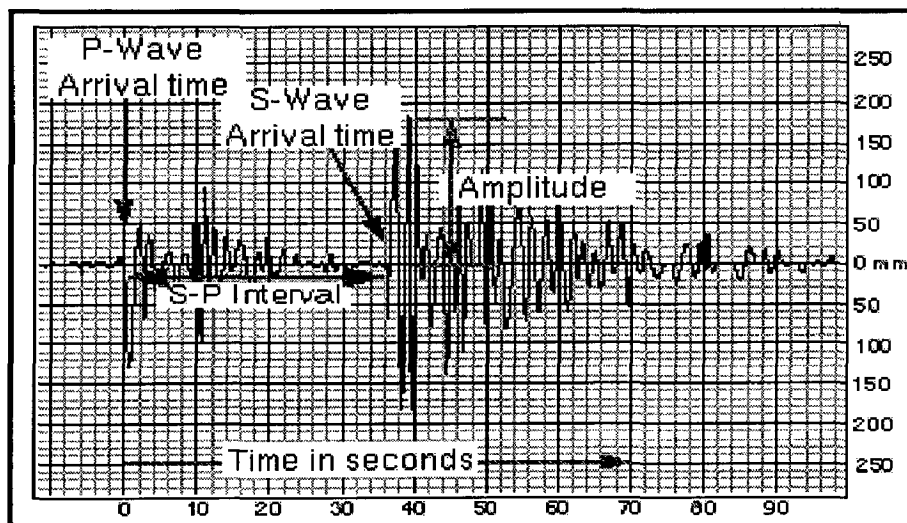


Figure 1.6: A Simple Seismogram

By studying the seismogram, the seismologist can tell the distance and strength of an earthquake. The seismogram gives no information to the seismologist regarding the exact location of the epicenter or whether the earthquake happened so many miles or kilometers away from that seismograph. (*Figure 1.7*). To find the exact epicenter, you need to know what at least two other seismographs in other parts of the country or world

recorded The location of an epicenter 1 is found by plotting its distance from three recording stations 2. Each station notes the different arrival times of *P* and *S* waves and uses a graph that allows the distance from the epicenter to be measured. The distance is then used as the radius of a circle around each station. The epicenter of the earthquake is located at the intersection of these three circles *figure 1.8*.

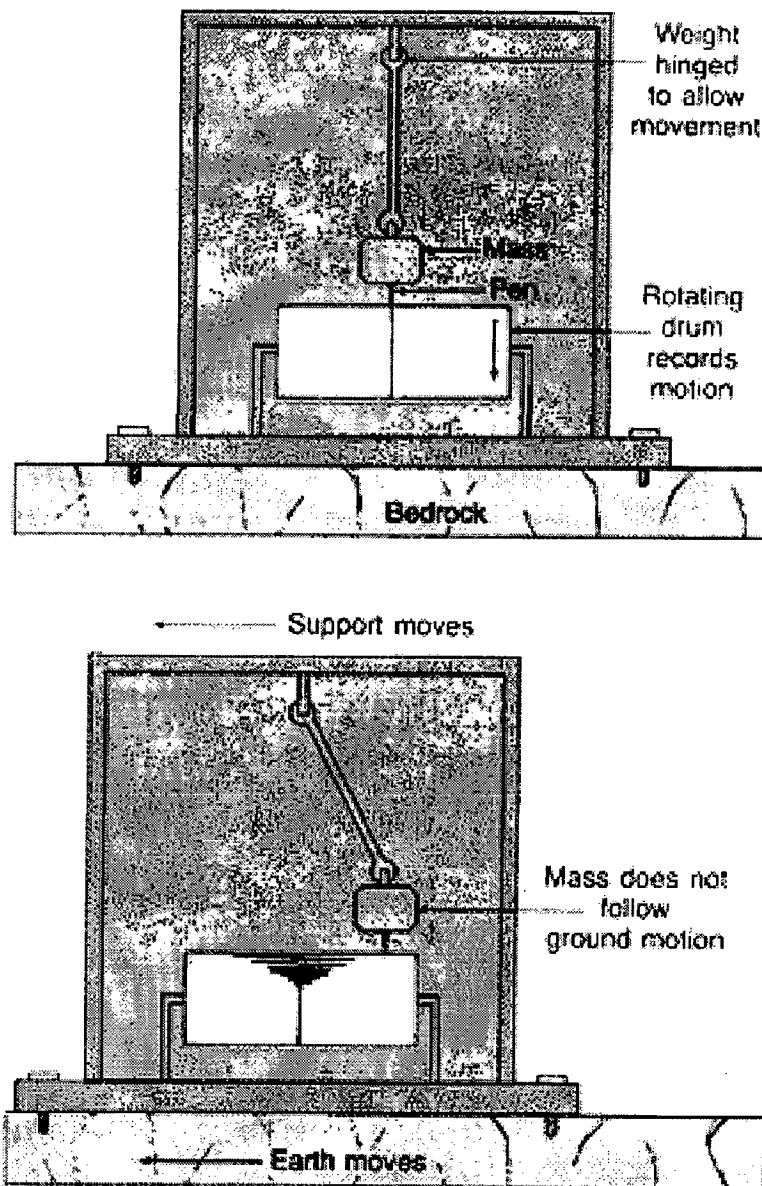


Figure 1.7: The Basic Seismograph

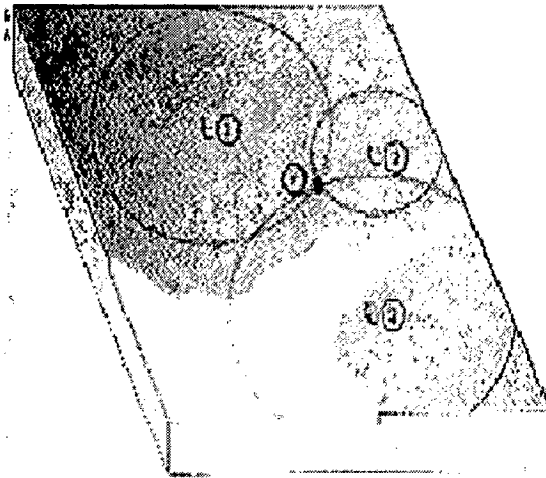


Figure 1.8: The location of an epicenter

At the recording station there may be Single component (1-C) recording i.e. Vertical component of ground motion or three components (3-C) recording i.e. East-West, North-South and Vertical component of ground motion.

The data used in this dissertation work has been obtained from Garhwal Kumaon Himalayan Region, where there is three component continuous (24 hour) recording. Data is recorded at 100 samples per second. So for a 1 hour recording 1080000 ($100 \times 3 \times 60 \times 60$) samples are recorded or for 24 hour 51840000 ($100 \times 3 \times 3600 \times 24$) bytes are to be stored. These data files are then segmented manually for event allocation. These events files range from 20 sec to 15 min. The size of these event files depends upon the magnitude of the earthquake and its epicentral distance. *Figure 1.9* shows an event recorded and segmented in SEISAN software.

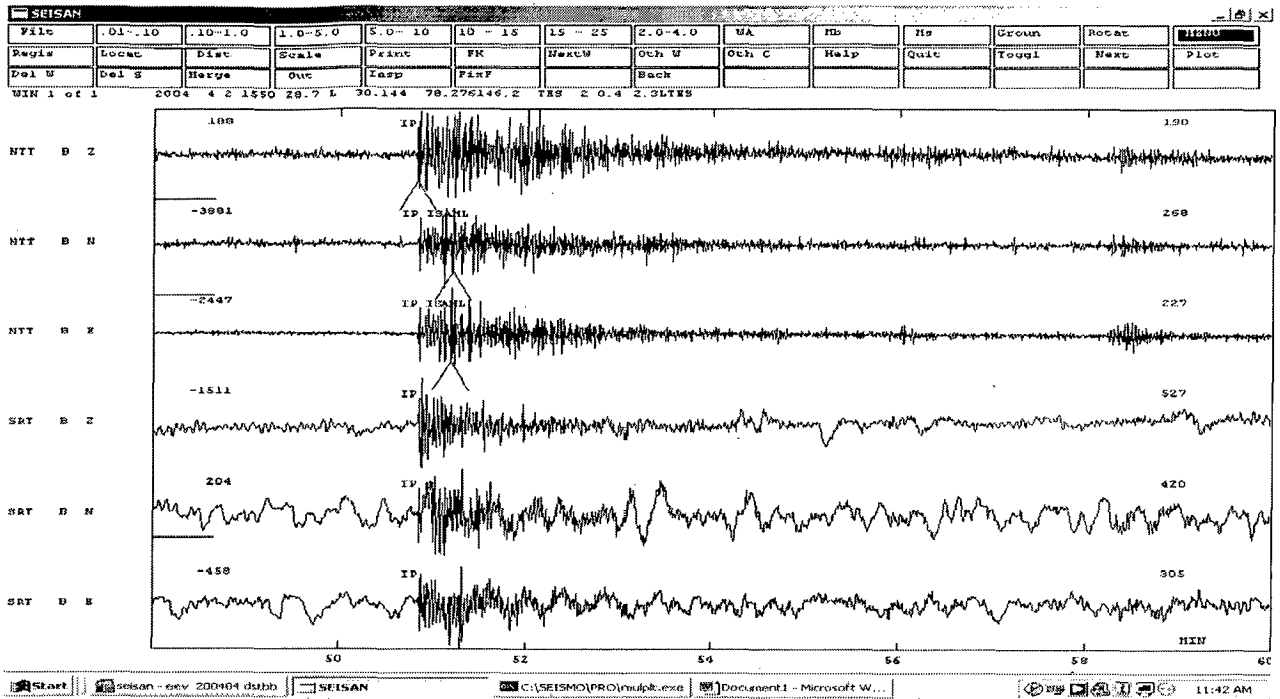


Figure 1.9: Seismic data file as displayed in SEISAN.

1.4 Causes of earthquakes

Earthquakes are usually caused when rock underground suddenly breaks along a fault. This sudden release of energy causes the seismic waves that shake the ground. When two blocks of rock or two plates rub against each other, they stick partially. In other words they don't slide smoothly and catch on each other.

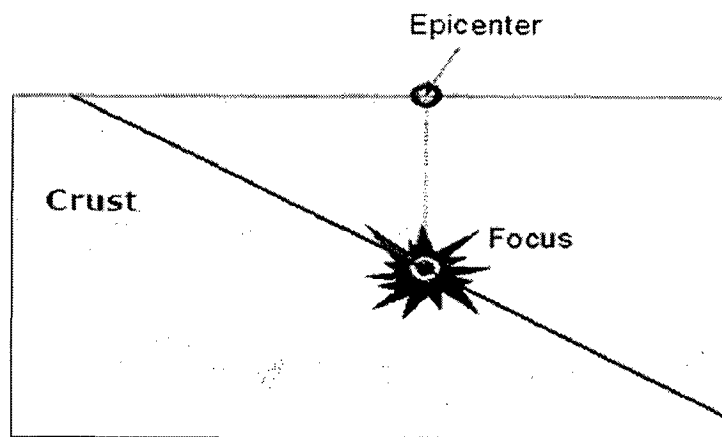


Figure 1.10: The epicenter and focus of earthquake

The rocks while pushing against each other do not move. After a while, the rocks break up because of the intense pressure built up. Eventually when the rocks break, the earthquake occurs. During the earthquake and afterwards, the plates or blocks of rock start moving, and they continue to move until they get stuck again. The spot underground where the rock breaks is called the **focus** (*Figure 1.10*) of the earthquake. The region right above the focus (on top of the ground) is called the **epicenter** (*Figure 1.10*) of the earthquake. The **epicenter** is the point on the earth's surface vertically above the hypocenter (or focus), the point in the crust, where a seismic rupture begins.

There are three main plate tectonic environments: **extensional, transform, and compressional** (*Figure 1.11*). These environments are also called **normal, reverse and strike-slip** faults respectively. Plate boundaries in different localities are subject to different inter-plate stresses, producing these three types of faults that cause earthquakes. Each type has its own special hazards. The crust moves along cracks called *faults*. A fault is a break in the earth's crust. The earth can move in different directions depending on the type of fault.

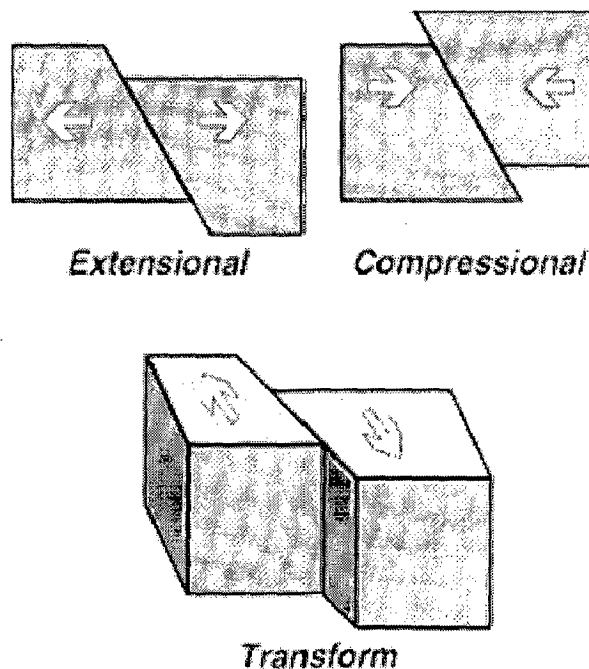


Figure 1.11: Different types of faults due to which earthquakes occur (In each fault are shown two blocks of earth surface rubbing or pushing over each other)

Tension, a pulling force that causes the plates to move apart, can create a **normal fault**. The rocks above a normal fault move downward as the plates below the fault move upward. When the earth's plates come together, they produce compression forces that push on rocks from either side. Sometimes the rocks bend. In other cases, they break and one rock slides up over the other. In a **reverse fault** the rock above the fault slides up over the rock below the fault. At a **strike-slip fault**, the rocks on either side of the fault slide past each other. This sliding force is called shearing. As the plates slide past each other, the forces bend and twist the land. Sometimes the land gets caught as it slides. When it releases or breaks, an earthquake occurs.

1.5 SEISMIC STATION

Seismic stations measure the ground motion at a specific location. The data are converted from the analog signal produced from the ground motion into digital information that computers are capable of understanding and manipulating. The difference between networks with analog stations and networks with digital stations is the point at which the analog signal is converted into digital information.

1.5.1 Analog Station

Analog stations are called "analog" because the analog signal is converted into digital information at the site of data processing. This means that the analog signal must be sent, in our case over phone lines, from each station to the central site. Each station's signal is then converted from analog to digital by hardware and processed by computers.

Signals from analog stations go off-scale quickly because the electronics and analog phone lines have limited dynamic range. However, each analog station is somewhat simpler, the time stamping of the data is done simultaneously, and the data conversion hardware is at central site, so the analog stations are somewhat easier to maintain.

1.5.2 Digital station

Digital stations, on the other hand, have high and low gain sensors and do their data conversion at the sensing site itself with 24 bit digitizers, thus allowing both small and large signals to stay on scale. The digital information is then sent via digital data link to central site where it is able to be used immediately by the computers processing and storing the data.

Using digital stations instead of analog stations provides several important benefits:

- The high and low gain sensors provide data on scale for both small and large earthquakes.
- The digital data can be error checked so that line noise won't cause the data to be corrupted.
- Although the data output by different data loggers is often of different formats, the network can incorporate them through simple software changes.

1.6 Aim of the Dissertation

Seismic warning systems (which release alert messages quickly after an earthquake) are dependent on automatic, quasi-real time, procedures for detecting, onset picking and identifying of signal phases in seismogram recordings. For systems which release alert messages immediately after an earthquake, the accuracy of these phase picks is essential for automatic and reliable hypocenter determination. Also instant detection and accurately picking the first *P*-wave arrival is immensely helpful in event localization, its identification and for analyzing source mechanism, especially in the era of large volumes of digital and real-time seismic data. Manual analysis of seismograms is time consuming and subjective and so there is a need to provide a more efficient alternative approach to detect the earthquake. Therefore an automatic method should be approached, ANN which is well known for its learning capability are used as detectors and pickers. Detectors are considered to be processes for detecting the presence of seismic phases, whereas pickers are considered to be processes for estimating accurately the onset time of

such phases. Hence Artificial Neural Network is used to detect the seismic phase and pick *P*-wave.

1.7 Introduction to Dissertation

The Broadband seismograms (3-component recordings) namely East, North and Vertical components are loaded to detect seismic phases and pick *P*-wave. Different attributes which can recognize *P* and *S* waves are selected. Depending upon the sensitivity and efficiency of these attributes, four attributes were selected for *P*-wave and another four attributes were selected for *S*-wave. These attributes were filtered in the range of 1-10 Hz using 2nd order Butterworth filter. These attributes were processed with a sampling frequency of 10 samples/s and calculated with a 0.5 second moving window. Each attribute is processed by 1 Neural Network at one time. The neural network were trained and tested. Then the seismic phase is detected and *P*-wave is picked. The flow chart in *figure1.12* represents the dissertation on Automatic Phase Picking of Seismic Signals Using Artificial Neural Network.

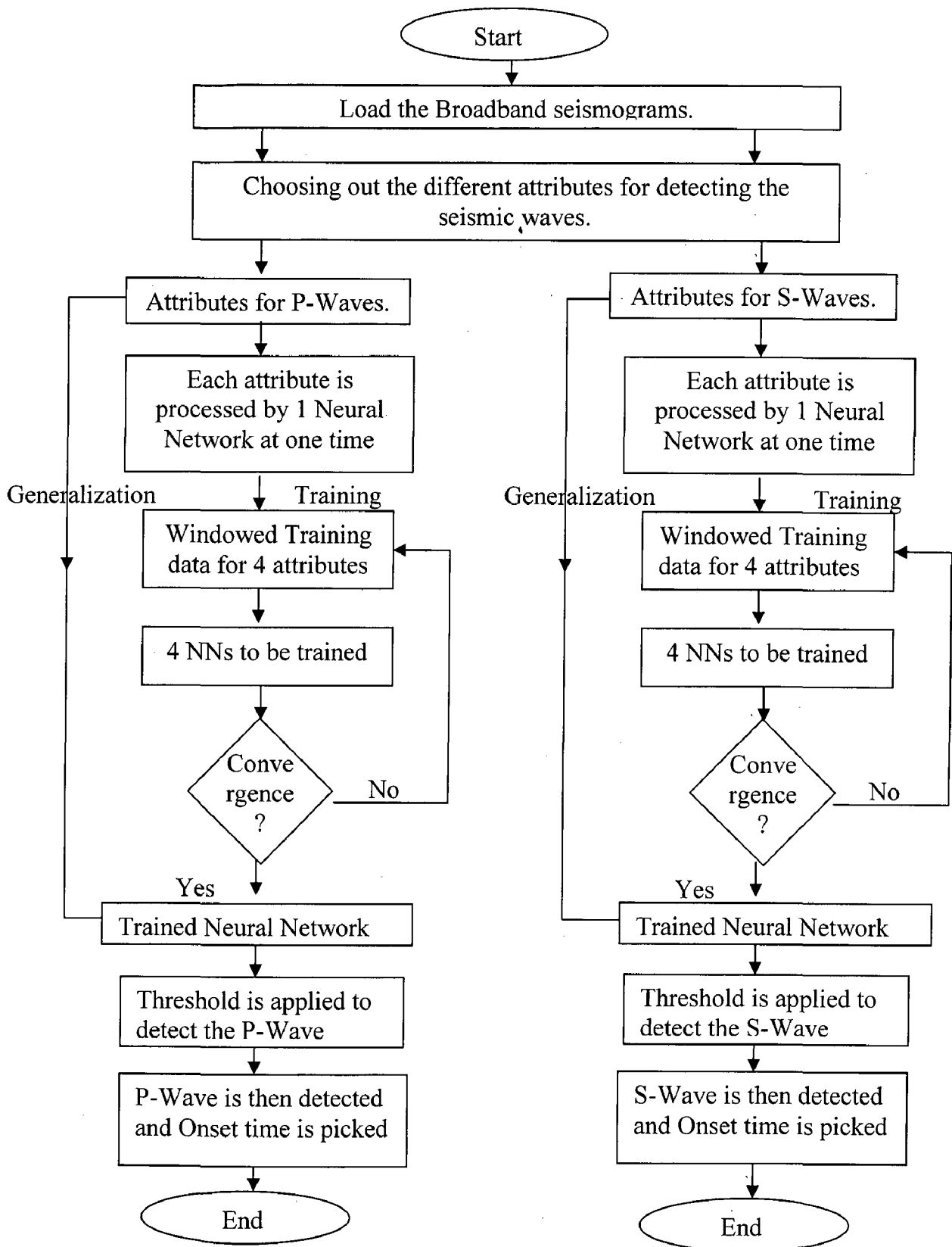


Figure 1.12: Flow chart representation of the dissertation

ARTIFIICAL NEURAL NETWORK

2.1 What is Artificial Neural Network?

Artificial Neural Network, one of the groups of techniques from the area of Artificial Intelligence is inspired by biological systems. An artificial neural network is an information processing system that emulates the structure and functions of the human brain. It consists of numerous simple processing elements connected together according to certain rules and is able to response dynamically to an outside stimulus and process information. As is the case for the human brain, the structure and processing sequence of the artificial network are parallel.

2.2 History of Artificial Neural Network

The progress of neurobiology has allowed researches to build mathematical models of neurons to simulate neural behaviour. This idea dates back to the early 1940s when of the first abstract models of a neuron was introduced by McCulloch and Pitts (1943). Hebb (1949) proposed a learning law that explained how a network of neurons learned. Other researchers pursued this notion through the next two decades, such as Minsky (1954) and Rosenblatt (1958). Rosenblatt is credited with the Perceptron learning algorithm. At about the same time, Widrow and Hoff developed an important variation of perceptron learning, known as the Widrow Holf rule.

Later, Minsky and Papert (1969) pointed out theoretical limitations of single layer neural network models in their landmark book *Perceptrons*. Due to this pessimistic projection, research on artificial neural networks lapsed into an eclipse for nearly two decades. Despite the negative atmosphere, some researchers still continued their research and produced meaningful results. For example, Anderson (1977) and Grossenberg (1980) did important work on psychological models. Khonen (1977) developed associative memory models.

In the early 1980s, the neural network approach was resurrected. Hopfield (1982) introduced the idea of energy minimization in physics into neural networks. His influential paper endowed this technology with renewed momentum. Feldman and Ballard (1982) made the term “connectionist” popular. Sometimes the term connectionism is also referred to as subsymbolic processes, which have become the study of cognitive and AI systems inspired by neural networks (Smolensky 1988). Unlike symbolic AI, connectionism emphasizes the capability of learning and discovering representations. Indiosly, connectionism has become a common ground between traditional AI and neural network research.

In the middle 1980s, the book *Parallel Distributed Processing* by Rumelhart and McClelland (1986) generated great impacts on computer, cognitive and biological sciences. Notably, the backpropagation learning algorithm developed by Rumelhart, Hinton and Williams (1986) offers a powerful solution to training a multilayer neural network and shattered the cruse imposed on perceptrons. A spectacular success of this approach is demonstrated by the NETtalk system developed by Sejnowski and Rosenberg (1987), a system that converts English text into a highly intelligible speech. It is interesting to note, however, that the idea of Backpropagation had been developed by Werbos (1974) and Parker (1982) independently.

Although the neural network approach rejects the notion of separating of knowledge from the inference mechanisms, it does not reject the importance knowledge in many tasks that require intelligence. It just uses a different way to store and manipulate knowledge.

The symbolic approach which has long dominated the field of AI was recently challenged by the neural network approach. There have been speculations about whether one approach should substitute for another or whether the two approaches should coexist and combine. More evidence favors the integration alternative in which the low-level pattern recognition capability offered by the neural network approach and the high level cognitive reasoning ability provided by the symbolic approach complement eachother (Kandel and Langholz 1992). The optimal architecture of future intelligent systems may well involve their integration in one way or another.

2.3 Need for Artificial Neural Network

The neural network processing typically involves dealing with large-scale problems in terms of dimensionality, amount of data handled and the volume of simulation or neural hardware processing. This large scale approach is both essential and typical for real life applications. The artificial neural network has a very strong learning ability and can adapt itself to the outside environment by learning. The artificial neural network can learn from incomplete and inaccurate data, even with considerable noise, and has a very robust error tolerance. The artificial neural network, if properly trained, can give an approximately optimal solution from limited and distorted information.

2.4 INTRODUCTION TO ANN

Various methods have been applied to the interpretation of seismic signals from a local seismic network. For example, JURKEVICS (1988) presented a technique for polarization analysis of three component seismic data. ROBERTS *et al* (1989) presented a technique for phase identification based on the auto- and cross correlations of the three component seismic data, etc..., The application of artificial intelligence methods to earthquake analysis is a relatively recent development.

Artificial neural networks (ANNs), provide a natural alternative to the application of earthquake analysis as they have proven useful in handling complicated pattern recognition problems in other applications such as character recognition, handwritten (processing checks) recognition, face recognition., etc..., The artificial neural network has many configurations and can provide methods to solve problems in real time seismology such as location of earthquake, earthquake detection, magnitude determination of earthquake, prediction of earthquake intensity distribution, etc...,

There are useful networks which contain only one layer, or even one element, but most applications require networks that contain at least the three normal types of layers - input, hidden, and output. The layer of input neurons receives the data either from input files or directly from electronic sensors in real-time applications. The output layer sends information directly to the outside world, to a secondary computational process, or to

other devices such as a mechanical control system. Between these two layers can be many hidden layers. These internal layers contain many of the neurons in various interconnected structures. The inputs and outputs of each of these hidden neurons simply go to other neurons. In most networks each neuron in a hidden layer receives the signals from all of the neurons in a layer before it, typically an input layer. After a neuron performs its function it passes its output to all of the neurons in the layer after it, providing a feedforward path to the output. *Figure 2.1* shown below is the structure of an artificial neural network.

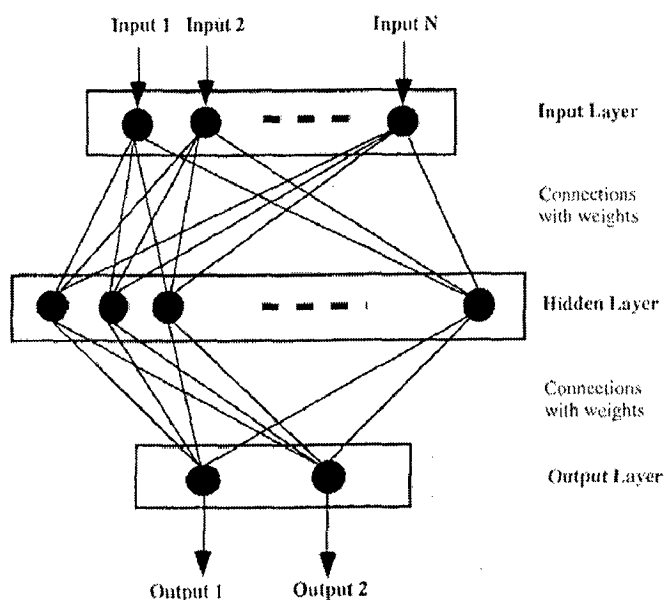


Figure 2.1: Structure of a Multilayer Artificial Neural Network

The above structure of the artificial neural network is a feed forward multilayer network consists of typically three layers; input, output and hidden layer. Neurons in one layer have connections with neurons in the next neighbouring layer. Every connection between neurons has a numerical value attached to it and is called its weight. Neurons in the input layer take input values from the variables and pass this input to each neuron of the successive hidden layer. The output of each neuron in the hidden layer is then multiplied by the weights of connected input neurons and then passed to the neurons in the next hidden layer or output layer.

There are two steps for neural network development. One is a “Learning” process where the ANN learns from a set of representative training samples of input and output patterns. Another is a “generalizations” process where the trained neural network is applied to unknown input signals. This is explained below.

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins. There are two approaches to training - supervised and unsupervised. Supervised training involves a mechanism of providing the network with the desired output either by improving the network's performance or by providing the desired outputs with the inputs. Unsupervised training is where the network has to make sense of the inputs without outside help. The vast bulk of networks utilize supervised training. Unsupervised training is used to perform some initial characterization on inputs.

2.5 Neuron modeling

A neuron with a single scalar input with no bias appears on the left below.

$$\text{Summer output } n = WP + b$$

$$\text{Neuron output } a = f(wp + b)$$

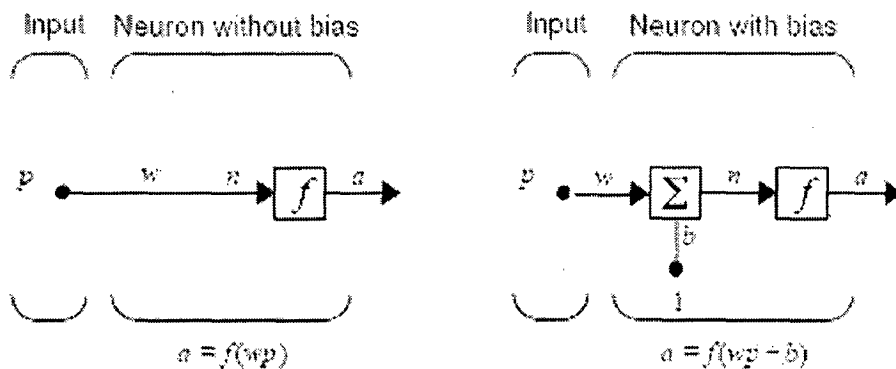


Figure 2.2: Single input neuron

The scalar input p is transmitted through a connection that multiplies its strength by the scalar weight w , to form the product wp , again a scalar. Here the weighted input wp is the only argument of the transfer function f , which produces the scalar output a . The neuron on the right has a scalar bias, b . The bias may be viewed as simply being added to

the product wp as shown by the summing junction or as shifting the function f to the left by an amount b . It has the effect of raising or lowering the net input of the activation function, depending on whether it is positive or negative, respectively. The bias is much like a weight, except that it has a constant input of 1.

The transfer function net input n , again a scalar, is the sum of the weighted input wp and the bias b . This sum is the argument of the transfer function f . Here f is a transfer function, typically a step function or a sigmoid function, which takes the argument n and produces the output a . It is to be noted that w and b are both *adjustable* scalar parameters of the neuron. The central idea of neural networks is that such parameters can be adjusted so that the network exhibits some desired or interesting behavior. Thus, the network can be trained to do a particular job by adjusting the weight or bias parameters, or perhaps the network itself will adjust these parameters to achieve some desired end.

2.6 TRANSFER FUNCTIONS

Referring to *figure 2.2*, the processing elements consist of two parts. The first part consists of an adder for summing up the input signals, weighted by the respective synapses of the neuron, and the second part consists of an activation function for limiting the amplitude of the output of the neuron. The activation function is also referred to as a squashing function, in that it squashes the permissible amplitude range of the output signal to some finite value. The normalized amplitude of the output of a neuron lies within the closed unit interval $[0, 1]$ or $[-1, 1]$.

The activation function, denoted by $\Phi(I)$, defines the output of the neuron in terms of the individual local field I . The threshold function passes information (usually a +1 signal) only when the output of the first part of the artificial neuron exceeds the threshold value T . A single neuron with a threshold activation function is known as a single-layer perceptron, whereas the same neuron with a signum activation function is known as an adaline.

$$\Phi(I), \begin{cases} 1 & \text{if } I \geq 1 \\ 0 & \text{if } I <= 0 \end{cases} \quad (\text{assume } \Phi(I) = a)$$

The different activation functions are as follows:

2.6.1 Hard-Limit Transfer Function

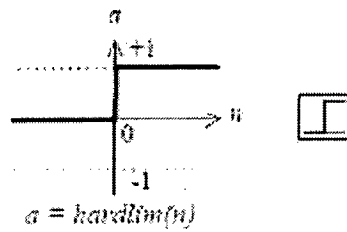


Figure 2.3: Hard-Limit Transfer Function

The hard-limit transfer function shown above limits the output of the neuron to either 0, if the net input argument n is less than 0; or 1, if n is greater than or equal to 0.

2.6.2 Linear transfer function

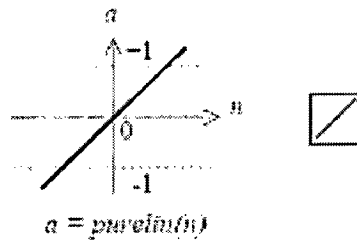


Figure 2.4: Linear transfer function

2.6.3 Sigmoid transfer function

The sigmoid function, whose graph is s-shaped, is the common form of activation function used in the construction of ANNs. It is defined as a strictly increasing function that exhibits a graceful balance between linear and non-linear behaviour. The sigmoid function passes negative information when the output is less than the threshold value T and positive information when the output is greater than the threshold value T . It is a continuous function that varies that gradually between two asymptotic values, typically 0 and 1, or -1 and +1. The sigmoid transfer function shown below takes the input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1.

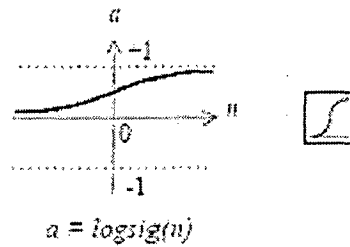


Figure 2.5: Log-Sigmoid Transfer Function

$$\Phi(I) = \frac{1}{1 + e^{-aI}}$$

Where α is a slope parameter of the sigmoid function which adjusts the abruptness of this function as it changes between the two asymptotic values. By varying the parameter α , one can obtain sigmoid function of different slopes.

This transfer function is commonly used in Backpropagation networks, in part because it is differentiable. The symbol in the square to the right of each transfer function graph shown above represents the associated transfer function. These icons will replace the general f in the boxes of network diagrams to show the particular transfer function being used.

2.7 Summary of Artificial Neural Networks

Feedforward:

- Perceptron
- Adaline and Madaline
- Backpropagation Network
- Radial Basis Function Network (RBFN)
- General Regression Network
- Modular Neural Network (MNN)
- Learning Vector Quantization(LVQ) Network
- Probabilistic Neural Network (PNN)
- Fuzzy Neural Network (FNN)

Recurrent:

- Hopfield Network
- Boltzmann Machine
- Kohonen's Self-Organizing Feature Map (SOFM)

- Recirculation Network
- Brain-State-in-a-Box (BSB)
- Adaptive Resonance Theory (ART) Network
- Bi-directional Associative Memory (BAM)

2.8 APPLICATIONS OF NEURAL NETWORK

2.8.1 Neural Network Applications in general

Neural network Applications, May provide a model for massive parallel computation, More successful approach of “parallelizing” traditional serial algorithms, Can compute any computable function, Can do everything a normal digital computer can do, and Can do even more under some impractical assumptions. Some of the applications where neural network used are: Recognizing and matching complicated, vague, or incomplete patterns, Data is unreliable, Problems with noisy data, Prediction, Classification, Data association, Data conceptualization, Filtering, Recognition and Planning. The above applications are explained as follows:

Prediction: learning from past experience

- Pick the best stocks in the market
- Predict weather
- Identify people with cancer risk

Classification

- Image processing
- Predict bankruptcy for credit card companies
- Risk assessment

Recognition

- Pattern recognition: SNOOPE (bomb detector in U.S. airports)
- Character recognition
- Handwriting: processing checks

Data association

- Not only identify the characters that were scanned but identify when the scanner is not working properly

Data Conceptualization

- Infer grouping relationships
e.g. extract from a database the names of those most likely to buy a particular product.

Data Filtering

- Take the noise out of a telephone signal, signal smoothing

Planning

- Unknown environments
- Sensor data is noisy
- Fairly new approach to planning

2.8.2 Neural network Applications in Earthquake Engineering

Many process related to earthquake engineering and engineering seismology use ANN as one of the major tools. Some of the uses of ANN are:

1. Discrimination of Earthquakes and Underwater explosions.
2. Picking of Seismic Arrivals
3. Seismic Signal classification.
4. Earthquake Prediction.
5. Magnitude Prediction.
6. Seismic Phase Identification.
7. Seismic data Processing.
8. Prediction of Seismicity cycles in highly seismic areas.
9. Prediction of Peak Ground Acceleration by utilizing the strong motion data.

In most of the applications mentioned above Multilayer Feedforward Backpropagation Neural Network has been widely used. The Back-Propagation Neural Network is explained in detail in the next chapter.

BACKPROPAGATION NEURAL NETWORKS

3.1 Backpropagation Neural Network

The Backpropagation network is probably the most well known and widely used among the current types of neural network systems available. In contrast to earlier work on perceptron, the Backpropagation network is a multilayer feedforward network with a different transfer function in the artificial neuron and a more powerful learning rule. The learning rule is known as Backpropagation, which is a kind of gradient descent technique with backward error (gradient) propagation, as depicted in *figure3.1*. The training instance set for the network must be presented many times in order for the interconnection weights between the neurons to settle into a state for correct classification of input patterns. While the network can recognize patterns similar to those they have learned, they do not have the ability to recognize new patterns. This is true for all supervised learning networks. In order to recognize new patterns, the network needs to be retrained with these patterns along with previously known patterns. If only new patterns are provided for retraining, then old patterns may be forgotten. In this way, learning is not incremental over time. This is a major limitation for supervised learning networks. Another limitation is that the Backpropagation network is prone to local minima, just like any other gradient descent algorithm.

The Backpropagation network in essence learns a mapping from a set of input patterns (e.g., extracted features) to a set of output patterns (e.g., class information). This network can be designed and trained to accomplish a wide variety of mappings. This ability comes from the nodes in the hidden layer or layers of the network which learn to respond to features found in the input patterns. The features recognized or extracted by the hidden units (nodes) correspond to the correlation of activity among different inputs units. As the network is trained with different examples, the network has the ability to

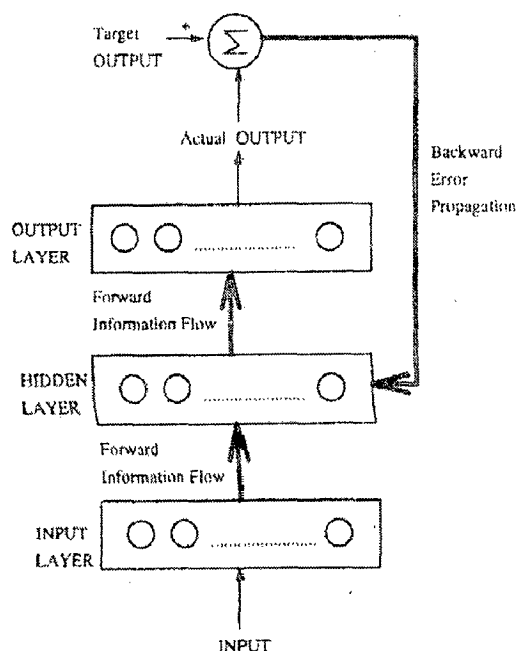


Figure 3.1: The Backpropagation network

generalize over similar features found in different patterns. The key issue is that the hidden units must be trained to extract a sufficient set of general features applicable to both seen and unseen instances. To achieve this goal, at first, the network must not be overtrained. Overtraining the network will make it memorize the individual input-output training pairs rather than settling in the mapping for all cases. To prevent this undesired effect, one way is to terminate training once a performance plateau has been reached. Another way is to prune the network, creating a bottleneck between the input and output layers. The bottleneck will force the network to learn in a more general manner. This issue is explored later.

The Backpropagation network is capable of approximating arbitrary mappings given a set of examples. Furthermore, it can learn to estimate posterior probabilities ($P(w_i/x)$) for classification. The sigmoid function guarantees that the outputs are bounded between 0 and 1. In the multiclass case, it is not difficult to train the network so that the outputs sum up to 1. With accurate estimation of posterior probabilities, the network can act as a Bayesian classifier.

The Backpropagation network consists of one input layer, one output layer, and one or more hidden layers. If the input pattern is described by n bits or n values, then there should be n input units to accommodate it. The number of output units is likewise determined by how many bits or values are involved in the output pattern. Theoretical guidance exists for determining the numbers of hidden layers and hidden units (see later discussion). They can be recruited or pruned as indicated by the network performance. Typically, the network is fully connected between and only between adjacent layers. The Backpropagation algorithm (Rumelhart, Hinton, and Williams 1986) is formulated below.

3.1.1 The Backpropagation algorithm

The Backpropagation algorithm is formulated as:

- Weight Initialization
 - Set all weights and node thresholds to small random numbers. Note that the threshold is the negative of the weight from the bias unit.
- Calculation of activation
 1. The activation level of an input is determined by the instance presented to the network.
 2. The activation level O_j of an hidden and output unit is determined by

$$O_j = F\left(\sum W_{ji} O_i - \theta_j\right)$$

where W_{ji} is the weight from an input O_i , θ_j is the node threshold and F is a sigmoid function :

$$F(a) = \frac{1}{1 + e^{-a}}$$

- Weight Training
 1. Start the output units and work backward to the hidden layers recursively.

Adjust weights by

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji}$$

where W_{ji} is the weight from unit i to j at time t and ΔW_{ji} is the weight adjustment.

2. The weight change is computed by

$$\Delta W_{ji} = \eta \delta_j O_i,$$

where η is a trial independent learning rate ($0 < \eta < 1$) and δ_j is the error gradient at unit j . Convergence is sometimes faster by adding a momentum term:

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta_j O_i + \alpha [W_{ji}(t) - W_{ji}(t-1)]$$

where $0 < \alpha < 1$

3. The error gradient is given by :

For the output units: $\delta_j = O_j(1 - O_j)(T_j - O_j)$

where T_j is the desired (target) output activation O_j is the actual output at unit j .

For the hidden units: $\delta_j = O_j(1 - O_j) \sum_K \delta_k W_{kj}$

where δ_j is the error gradient at unit k to which a connection points from hidden unit j .

4. Repeat iterations until convergence in terms of the selected error criterion. An iteration includes presenting an instance, calculating activations, and modifying weights

The name "Backpropagation" comes from the fact that the error (gradient) of hidden units are derived from propagating backward the errors associated with output units since the target values for the hidden units are not given. In the Backpropagation network, the activation function chosen is the sigmoid function, which compresses the output value in the range between 0 and 1. The sigmoid function is advantageous in that it can accommodate large signals without saturation while allowing the passing of small signals without excessive attenuation. Also, it is a smooth function so that gradients can be calculated, which are required for a gradient descent search.

A Derivation of Backpropagation (Fu Limin 2003) Here, we show how to derive the Backpropagation learning rule given by

$$\Delta W_{ji} = \eta \delta_j O_i$$

If unit j is an output unit, then its δ_j is calculated by

$$\delta_j = (T_j - O_j) F'(net_j)$$

where

$$(net_j) = \sum_i W_{ji} O_i$$

F is a sigmoid function, and

$$O_j = F(net_j) = F\left(\sum_i W_{ji} O_i\right)$$

If unit j is a hidden unit, then its δ_j is given by

$$\delta_j = F'_j(net_j) \sum_K \delta_K W_{Kj}$$

The Backpropagation procedure minimizes the error criterion

$$E = 1/2 \sum_j (T_j - O_j)^2$$

Gradient Descent yields

$$\Delta W_{ji} = -\eta \partial E / \partial W_{ji}$$

By using the chain rule, we obtain

$$\partial E / \partial W_{ji} = (\partial E / \partial O_j) (\partial O_j / \partial W_{ji})$$

In the case when unit j is an output unit,

$$(\partial E / \partial O_j) = -(T_j - O_j)$$

and

$$(\partial O_j / \partial W_{ji}) = F'_j(net_j) O_i$$

Thus,

$$\begin{aligned} \partial E / \partial W_{ji} &= (\partial E / \partial O_j) (\partial O_j / \partial W_{ji}) \\ &= -(T_j - O_j) F'_j(net_j) O_i \\ &= -\delta_j O_i \end{aligned}$$

So, we obtain

$$\Delta W_{ji} = \eta \delta_j O_i$$

When unit j is a hidden unit, T_j is not given. Applying chain rule gives

$$(\partial E / \partial O_j) = \sum_k \partial E / \partial O_k (\partial O_k / \partial O_j)$$

The output of unit k is given by

$$O_k = F(\sum_j W_{kj} O_j)$$

Thus the term $(\partial O_k / \partial O_j)$ can be transformed by

$$(\partial O_k / \partial O_j) = F'(net_k) W_{kj}$$

As a result,

$$\begin{aligned} (\partial E / \partial O_j) &= \sum_k \partial E / \partial O_k (\partial O_k / \partial O_j) \\ &= - \sum_k k (T_k - O_k) F'(net_k) W_{kj} \\ &= - \sum_k k \delta_k W_{kj} \end{aligned}$$

Thus,

$$\begin{aligned} \partial E / \partial W_{ji} &= (\partial E / \partial O_j) (\partial O_j / \partial W_{ji}) \\ &= - (\sum_k k \delta_k W_{kj}) F_j'(net_j) O_i \\ &= - \delta_j O_i \end{aligned}$$

Then we obtain

$$\Delta W_{ji} = \eta \delta_j O_i$$

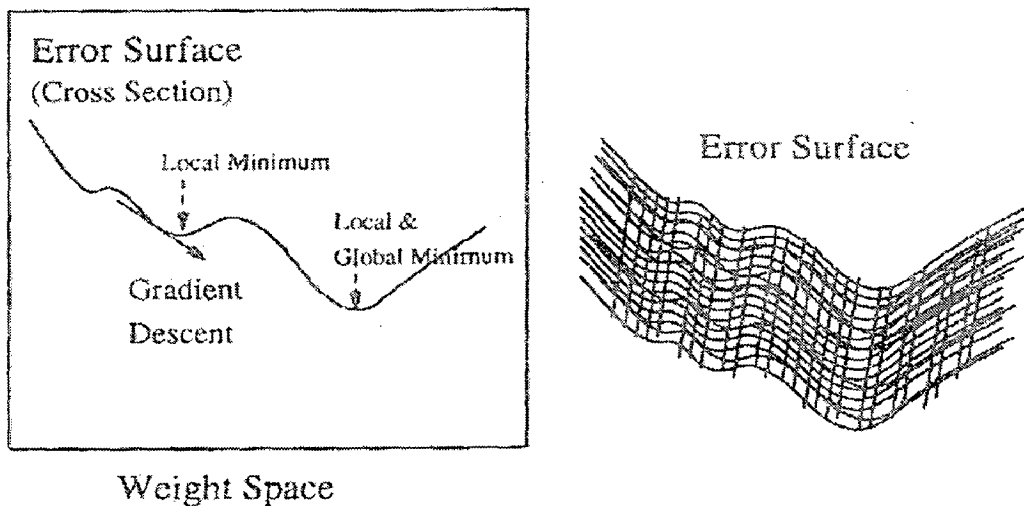


Figure 3.2: Search on the error surface along the gradient

3.1.2 Step size

A careful selection of the step size (the learning rate) is often necessary to ensure smooth convergence. It has been shown that a large step size may cause the network to become paralyzed. When network paralysis occurs, further training does little for convergence. On the other hand, if the size is too small, convergence can be very slow. Some have suggested to vary the step size adaptively during training. There appears to be more empirical guidance than theoretical understanding on this issue.

3.1.3 Local Minima

Figure 3.2 shows that back propagation searches on the error surface along the gradient (steepest descent) in order to minimize the error criterion.

$$E = 1/2 \sum_j (T_j - O_j)^2$$

It is likely to get stuck in a local minimum.

The problem of local minima has been tackled in a number of ways. Incorporating a random component into the weight adjustment is one way to escape the local minimum. As in simulated annealing, a parameter called the temperature is defined so that for a higher temperature, more randomness can be introduced into the training process. When the temperature goes down to zero, the process becomes completely deterministic. The genetic algorithm has also been applied to deal with this problem. It proceeds by starting with multiple initial weight settings and recombining trained weights during the process. The cost of either approach is the prolonged training time.

3.1.4 Learning Speed

Some techniques have been used to accelerate the convergence of a gradient descent technique like Backpropagation. Newton's method uses the information of the second-order derivatives. Quasi-Newton methods approximate second-order information with first-order information. Conjugate gradient methods compute a linear combination of the current gradient vector and the previous search direction (momentum) for the current search direction. A detailed discussion on this issue is made by Shanno (1990).

The network training time can also be improved by some tricks, such as juggling the weights or using slightly noisy data (Caudill 1991; Stubbs 1990). It has been found that if the activation level is restricted to the range from $-\frac{1}{2}$ to $\frac{1}{2}$, convergence time may be reduced by half compared with the 0-to-1 range (Stornetta and Huberman 1987). This improvement proceeds from the fact that a weight coming from neuron of zero activation will not be modified. This idea is implemented by changing the input range to $-\frac{1}{2}$ to $\frac{1}{2}$ and using the following activation function:

$$-\frac{1}{2} + \frac{1}{1 + e^a}$$

Where a is the argument of the function. An alternative function is the hyperbolic tangent function (*tanh*), which lies in the range from -1 to 1.

3.1.5 Stopping Criteria

The process of adjusting the weights based on the gradients is repeated until a minimum is reached. In practice, one has to choose a stopping condition. There are several stopping criteria that can be considered: (Fu Limin 2003).

- * Based on the error to be minimized: In pattern recognition, one might consider stopping the procedure once the training data are correctly classified. When this is not the case, a fixed threshold is used so that the procedure is stopped if the error is below it. However, this criterion does not guarantee generalization to new data.

- * Based on the gradient: The algorithm is terminated when the gradient is sufficiently small. Note that the gradient will be zero at a minimum by definition.

- * Based on cross-validation performance: This criterion can be used to monitor generalization performance during learning and to terminate the algorithm when there is no more improvement.

The first two criteria are sensitive to the choice of parameters and may lead to poor results if the parameters are improperly chosen. The cross-validation criterion does not have this drawback. It can avoid overfitting the data and can actually improve the generalization performance of the network. However, cross-validation is much more computationally intensive and often demands more data.

3.1.6 Network Size

The Backpropagation network can approximate an arbitrary mapping only when the network will produce the best result for a given problem. If the network is too small. It cannot learn to fit the training data well. On the other hand, if the network is able to learn many solutions that are consistent with the training data but most of them are likely to be poor approximations of the actual model.

The network should have such a size as to capture the structure of the data and eventually to model the underlying problem. With some specific knowledge about the problem structure, one can sometimes form a good estimate of the proper network size.

With little prior knowledge, the size of the network must be determined by trial and error. One approach is to grow the network starting with the smallest-possible network until the performance begins to level off or decline. The network size is increased by adding more nodes in fixed layers or by adding new layers. Alternatively, we can proceed the other way around. That is, starting with a large network, we apply a pruning technique to remove those weights and nodes which have little relevance to the solution. In this approach, one needs to know how to select a “large” network to begin with.

In most cases, a two-layer network (with a single hidden layer) suffices to solve the problem. Numerous bounds on the number of hidden nodes in a two-layer network have been derived. However, most of them assume that the activation function is a hard-limiting function. Formal analysis will be more difficult in the case of a sigmoid function.

3.1.7 Complexity of Learning

It has been shown that the problem of finding a set of weights for a fixed-size network which performs the desired mapping exactly for given training data is *NP-complete*. That is, we cannot find optimal weights in polynomial time. So, for a very

large problem, it is unlikely that we can determine the optimal solution in a reasonable amount of time.

Learning algorithms like Backpropagation are gradient descent techniques which seek only a local minimum. These algorithms usually do not take exponential time to run. Empirically, the learning time on a serial machine for Backpropagation is about $O(N_w^3)$ where N_w is the number of weights in the network (Hinton 1989). The slowness of finding a local minimum is perhaps due to the characteristics of the error surface being searched. In the case of a single-layer perceptron, the error surface is a quadratic bowl with a single minimum, and the search on it is relatively easy. By contrast, in the multilayer backpropagation network, the error surface is typically convoluted, which obviously prohibits an efficient search. Increasing the learning rate to speed up the learning process may cause instability when it reaches the steep part of the surface. Several alternative methods have been proposed, such as the momentum and the second-order methods mentioned earlier.

3.1.8 Generalization

Generalization is concerned with how well the network performs on the problem with respect to both seen and unseen data. It is usually tested on new data outside the training set. Generalization is dependent on the network architecture and size, the learning algorithm, the complexity of the underlying problem, and the quality and quantity of the training data. Research has been conducted to answer such questions as :

- How many training instances are required for good generalization?
- What size network gives the best generalization?
- What kind of architecture is best for modeling the underlying problem?
- What learning algorithm can achieve the best generalization?

It is often difficult to answer any of these questions without fixing some factors. But an important goal is to develop a general learning algorithm which improves generalization in most or all circumstances. This is addressed in the next section.

For simplicity, we focus our discussion on learning logic functions of d binary variables. In this problem, there are 2^d different patterns in the domain and there are 2^2 possible functions. The network for identifying the true function should have d input units and one output unit. In general, the larger the network, the larger the set of functions it can form and the more likely the true function is in this set. We can think of the use of training data as a means of rejecting incorrect functions. Hopefully, at the end of training, there is only one function left which is the true function sought. If this is not the case, we wish the function learned to be the function that best approximates the true function. In this sense, the more training instances, the more likely the network can identify the true function. However, this is true only when the network is large enough (but not too large) to implement the true function.

Hush and Horne (1993) describe the approaches to study generalization and the methods to improve generalization. In one approach, the average generalization of the network is defined to be the average of the generalization values of all functions that are consistent with the set of training instances. The generalization value of a function is the fraction of the domain (of 2^d instances) for which the function produces the correct output.

In the second approach, the generalization error of the network is defined as the difference between the generalization on the training data (which forms an estimate of the true generalization) and the generalization on the actual problem. Since the network tends to fit the training data, the generalization with respect to it will be overly optimistic. However, in many cases, the generalization error can be bounded (a worst-case analysis), and this bound can be made arbitrarily small by increasing the number of training instances.

POLARIZATION ANALYSIS OF THREE COMPONENT SEISMIC ARRAY DATA

4.1 Introduction

The accuracy of automatic event location is an important and practical problem in network seismology. Numerous reliable methods have been developed to automatically estimate the onset time and phase detection using *P* and *S* arrivals from a seismic network (e.g., Dai Henchang and Colin Macbeth 1995, reviewed *P*-phase pickers that had been implemented in practical applications of seismic networks at that time.

Many researchers have discussed the polarization properties of the seismic signals, Jurkevics (1988) presented a technique for polarization analyses of three component seismic data particularly of *S* waves that is based on a time domain algorithm originally. The availability of three component data improves the chance of detecting the seismic arrivals on the basis of physical differences in particle motion between *P* and *S* waves. Three-component recordings are important for broadband seismic monitoring because broadband seismic phases can exhibit large horizontal motions. Simple side-by-side displays of three component seismograms can assist an analyst performing interpretations. However, a more quantitative approach involves processing the signals and extracting or enhancing their polarization content. The objective of the work described in this paper has been to implement a practical algorithm for polarization analysis which can be applied to three component data from seismic arrays. Basically, the technique involves filtering the signals into a series of narrow frequency bands, applying short sliding time windows and then computing the polarization ellipse from the covariance matrix in each window in each band. Instead of polarization filtering the traces, a series of attributes describing the particle motion characteristics as a function of time are output. Application to three component arrays is accomplished by averaging covariance matrices for all array sensors before solving the eigenproblem. Various attributes characterizing the particle motions are extracted from the motion ellipse. All these attributes are functions of time.

4.2 Attribute Selection for Seismic Arrivals

Polarization is the main source of information derivable from the three-component seismic data, and it is widely used for event location, phase identification. The polarization is estimated in each window as follows.

Let $X = [x_{ij}]$ (Where $i = 1, \dots, N$; $j = 1, 2, 3$) be the data matrix in one window, where x_{ij} is the i th sample of component j and N is the number of the samples. The mean of each component over this window is taken here to be zero.

The covariance matrix S is evaluated as

$$S_{ij} = \frac{XX^T}{N} = \frac{1}{N} \sum_{j=1}^N x_{ij} x_{jk},$$

Where T denoted transpose. The S is a 3*3 real and symmetric covariance matrix. Explicitly, the terms of S are the auto variances and cross-variances of the three components motion:

$$S = \begin{bmatrix} S_{zz} & S_{zn} & S_{ze} \\ S_{nz} & S_{nn} & S_{ne} \\ S_{ez} & S_{en} & S_{ee} \end{bmatrix}$$

Where S_{zn} denotes the cross variances of the vertical and north components, etc. The index convention used here is $z=1$, $n=2$, $e=3$. The covariance matrix S is positive semi-definite, which means that its eigenvalues are real and non negative (some may be zero). S is the matrix of coefficients termed the polarization ellipsoid, obtained through the best fit to the data in a least squares sense. The principle axes of the ellipsoid are estimated by solving the algebraic eigen problem for S . Once the principale axes of the polarization are estimated, the particle motion in the data window is determined. Information describing the characteristics of ground motion is extracted using attributes computed from the principale axes. Those polarization parameters particularly useful for characterizing the broadband seismic phases are as follows except the first one (STA \ LTA) is a conventional algorithm and all the remaining seven are polarization parameters.

- (1) Ratio between short-term average and long-term average (STA\LTA),
- (2) Rectilinearity (Rect),

- (3) Ratio of maximum to minimum horizontal amplitude (hmxnm),
- (4) Long -axis incidence angle of polarization ellipsoid (Inc1),
- (5) Ratio between horizontal power to total power (Rh2t),
- (6) Planarity (Plan),
- (7) Ratio of horizontal-to-vertical power (hvratp)
- (8) Short -axis incidence angle of polarization ellipsoid (Inc3),

The first four attributes are used to identify for *P*-Wave Phase identification and the next four attributes are used for identification of *S*-Wave Phase identification. A variety features of attributes of *P*-Waves and *S*-Waves has been discussed previously by (e.g., Cichowicz., 1993; Tarvainen, 1991). To compare the performance of these attributes in discriminating the *P* and *S* phase, eight attributes are discussed in this study. They are defined are as follows:

- (1) Ratio of short term average (STA) to long term average (LTA) Allen (1978):

$$\beta = \frac{STA}{LTA}$$

$\beta(i)$ is calculated recursively in a moving window at time i , and its value will change if the amplitude of signal changes. It is not a polarization attribute but a commonly used detector.

- (2) Rectilinearity (Jurkevics, 1988):

$$Rect = 1 - \frac{\lambda_2 + \lambda_3}{2\lambda_1},$$

where λ_1 , λ_2 and λ_3 are the eigen values such that $\lambda_1 > \lambda_2 > \lambda_3$. Rect. is a *P*-Type attribute.

- (3) Ratio of maximum to minimum horizontal amplitude (Jin Wang, 2002):

$$hmxnm = \sqrt{\frac{\lambda_1}{\lambda_2}},$$

where λ_1 , and λ_2 are the maximum and minimum eigenvalues of the horizontal component.

(4) Long axis incidence angle (Jurkevics, 1988):

$$Incl = \frac{\cos^{-1}(|u_{11}|)}{\pi/2},$$

where Incl is an apparent incidence angle (measured clockwise from vertical and normalized by $\pi/2$) of the eigen vector associated with the largest eigen value and u_{11} is a direction of the cosine vector. For P waves Incl approaches zero since most of the P -Wave energy is in the vertical direction. In contrast, Incl approaches unity for S waves.

(5) Ratio between horizontal power to total power (Jurkevics, 1988):

$$Rh2t = \frac{\sum_{j=1}^N [x_n^2(j) + x_e^2(j)]}{\sum_{j=1}^N [x_n^2(j) + x_e^2(j) + x_z^2(j)]}$$

Where x_n , x_e and x_z are amplitudes of three component recordings and N is the total points in the window.

(6) Planarity (Jurkevics, 1988):

$$Plan = 1 - \frac{2\lambda_3}{\lambda_1 + \lambda_2},$$

Planarity is an S -Type attribute because the particle motion of S -wave should be confined to a plane.

(7) Ratio of horizontal-to-vertical power (Jin Wang, 2002):

$$hvratp = \frac{s_{nn} + s_{ee}}{2s_{zz}},$$

where s_{nn} , s_{zz} and s_{ee} are the diagonal elements of the covariance matrix.

(8). Short axis incidence angle (Jurkevics, 1988):

$$\text{Incl} = \frac{\cos^{-1}(|u_{31}|)}{\pi/2}$$

where Incl is a normalized angle measured clockwise from the vertical direction and u_{31} is the direction cosines. For S -wave, Incl approaches zero since most of the S -wave energy is the horizontal plane, For P -wave, Incl approaches 1 in contrast.

For the covariance matrix calculation, the choice of both window length and frequency bandwidth is subject to the usual trade-off between resolution and variance. To avoid smearing information between close arrivals and to capture the Physical varying properties of polarization, short window band are required. However, longer window band yield more stable and reliable estimates. In this experiment, a window length of 10samples/sec is chosen to study the signal polarization attributes. The Second-order recursive Butterworth filters are first applied to seismograms, and then the time-varying attributes are computed continuously in the moving window. After examining all computed attributes, we have found that the features in the 1 to 10Hz bands have better response to P and S -wave arrivals for Broadband seismograms.

A set of seismograms, three-component recordings recorded at Garhwal Kumaon Himalayan region is used for illustration. Recordings of 25 earthquakes at station NTT (New Tehri Town) and SRT (Srikot) are used to calculate the above mentioned signal characteristics. *Table 4.1* gives parameters of these earthquakes and *figure 4.1* shows the locations of stations used for calculation. *Table 4.2* summarizes the responses of calculated attributes to Broadband P and S waves recorded at station NTT and SRT. From *Table 4.2*, we can see that the STA/LTA and Rh2t have the best performance in response to the P - and S -wave arrivals.

Table 4.1: Events used in this article, among these 10 events are used as training data for ANN P and S phase detector and P phase picker. All these events are used for testing.

Event	Year/month/day	Hr/min/sec	Latitude (N°)	Longitude (E°)	Depth (Km)	Station
1	2004 / 04 / 01	10 48 20.07	29.198	76.951	15.0	NTT
2	2004 / 04 / 02	15 50 49.17	30.144	78.276	6.2	NTT
3	2004 / 04 / 02	22 34 20.89	34.690	71.016	15.0	SRT
4	2004 / 04 / 03	02 51 48.01	32.181	80.811	15.0	SRT
5	2004 / 04 / 04	14 16 28.23	30.483	78.547	1.2	NTT
6	2004 / 04 / 05	02 24 39.03	30.713	78.727	15.0	SRT
7	2004 / 04 / 05	17 03 34.16	30.486	78.479	15.0	NTT
8	2004 / 04 / 08	23 40 53.77	30.712	78.599	15.0	SRT
9	2004 / 04 / 10	17 14 57.44	36.015	71.129	15.0	NTT
10	2004 / 04 / 10	22 43 54.47	31.767	69.462	15.0	SRT
11	2004 / 04 / 12	15 09 32.27	34.646	73.723	15.0	NTT
12	2004 / 04 / 13	14 16 54.76	30.455	78.507	15.0	NTT
13	2004 / 05 / 01	07 09 32.92	35.495	78.853	14.0	NTT
14	2004 / 05 / 02	06 52 44.39	30.224	78.965	14.0	NTT
15	2004 / 05 / 03	14 22 52.93	37.163	73.844	1.1	NTT
16	2004 / 05 / 03	17 33 48.46	34.670	86.933	15.0	NTT
17	2004 / 05 / 04	05 08 49.23	35.844	98.949	15.0	NTT
18	2004 / 05 / 04	11 39 57.35	25.073	97.327	15.2	NTT
19	2004 / 05 / 04	12 03 16.11	30.453	78.509	15.0	NTT
20	2004 / 05 / 04	17 16 57.15	29.372	79.707	20.7	NTT
21	2004 / 05 / 04	21 55 13.49	28.624	77.336	15.0	SRT
22	2004 / 05 / 05	13 48 30.88	30.692	78.377	15.0	NTT
23	2004 / 06 / 01	01 39 55.89	30.377	78.434	15.0	NTT
24	2004 / 06 / 03	10 14 26.44	29.028	79.647	15.0	NTT
25	2004 / 06 / 04	06 32 50.78	35.487	86.130	15.0	NTT

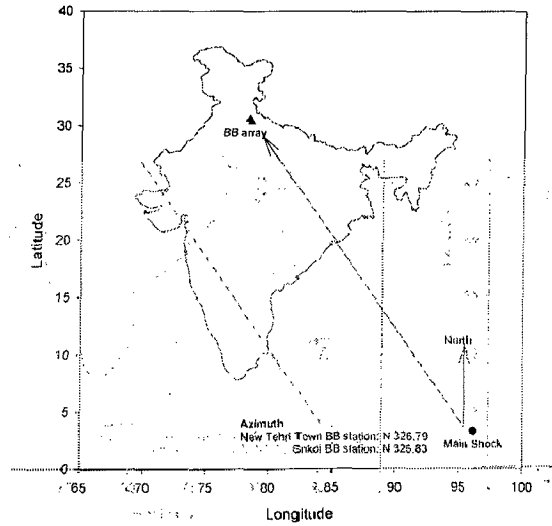


Figure 4.1: Location of the station

Among the eight calculated attributes, two types of response for local P - and S -wave signals are recognized. One type has the same response direction for both P and S waves, such as STA/LTA, Rect, Plan and hvratp. Another type has opposite response directions for local P and S waves, such as Rh2t, Inc3, hmxnm and Inc1. To distinguish P - and S -wave arrivals, we prefer to use those attributes that have different responses to P and S waves. This is explained from figures 4.3 to 4.10. Considering the performance and the physical meaning of those attributes, we decided to choose STA/LTA, Rect, hmxnm and Inc1 as four input attributes of the neural network for P -phase picker. And the other attributes such as Rh2t, Plan, hvratp and Inc3 are used as the input attributes for S -Phase picker.

Table 4.2: Summary of responses of P and S waves of recording NTT and SRT in Table 4.1

	Sta\Lta		Rect		Plan		Hvratp		Hmxnm		Rh2t		Inc1		Inc3	
	P	S	P	S	P	S	P	S	P	S	P	S	P	S	P	S
Response Direction	Up	Up	Up	Up	Up	Up	Up	Up	Down	Down	Down	Up	Down	Up	Up	Down
Correct Response Rate,%	100	100	92	84	100	92	96	100	92	96	88	78	64	68	68	68
Response Type	P and S waves have same response direction										P and S waves have opposite directions					

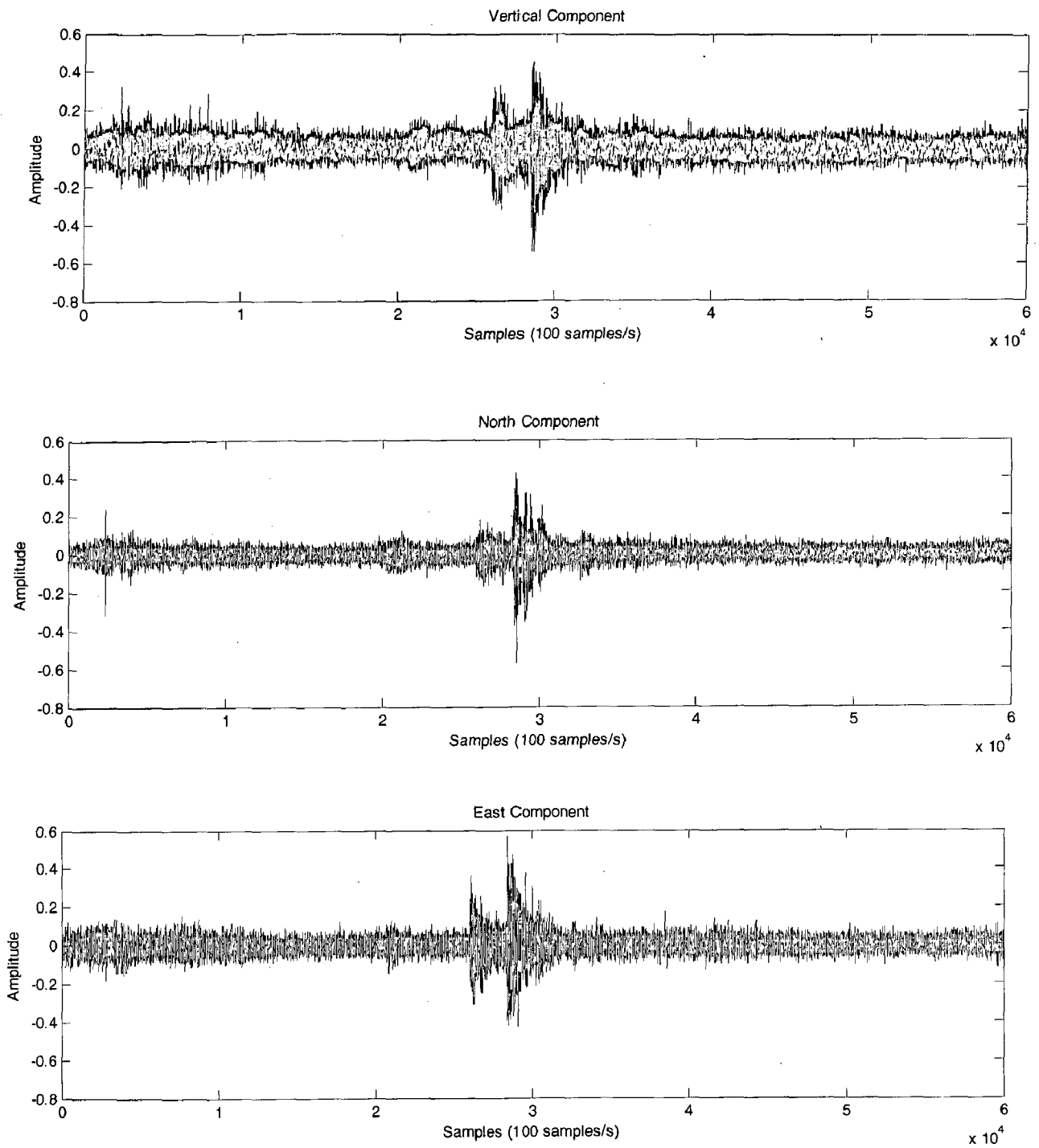


Figure 4.2: Sample of a 3-Component waveform (normalized) for event 1 in table 4.1

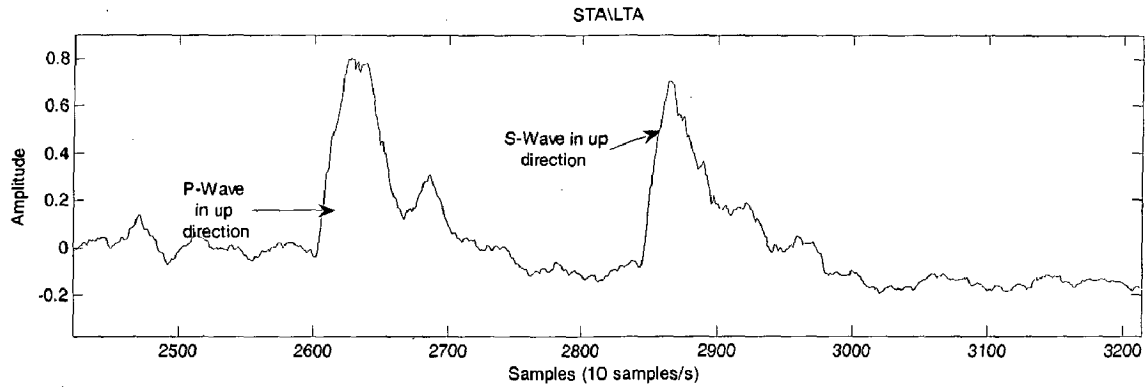
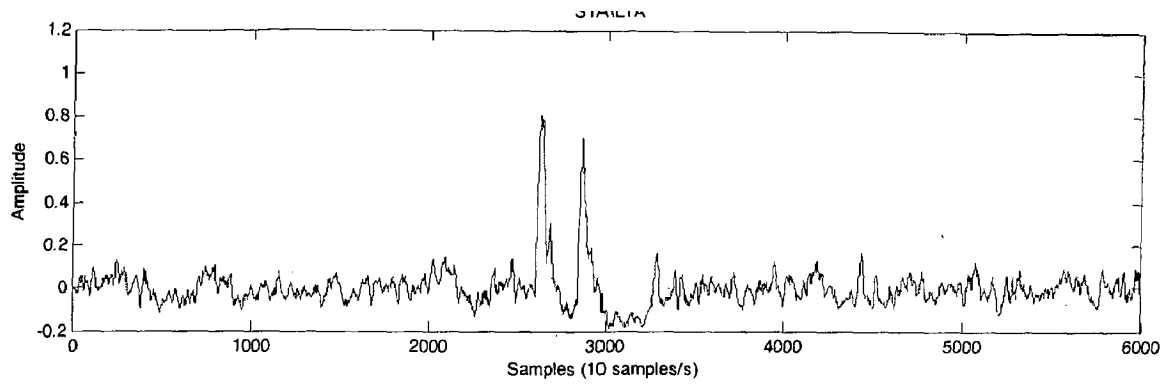


Figure 4.3: Represents STA \ LTA attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in STA\LTA attribute

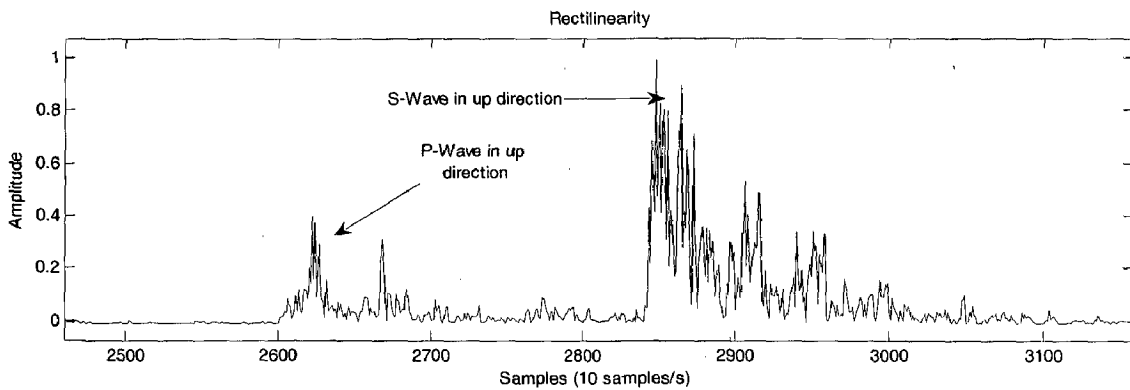
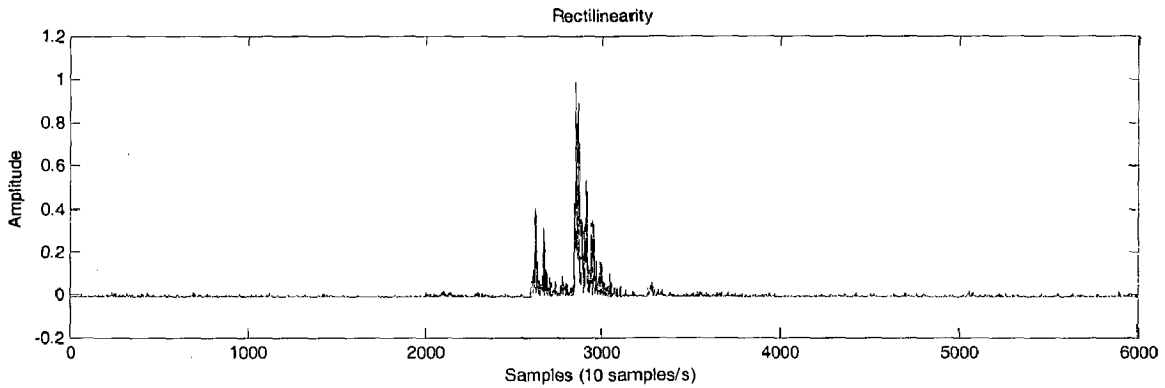


Figure 4.4: Represents Rectilinearity attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Rectilinearity attribute

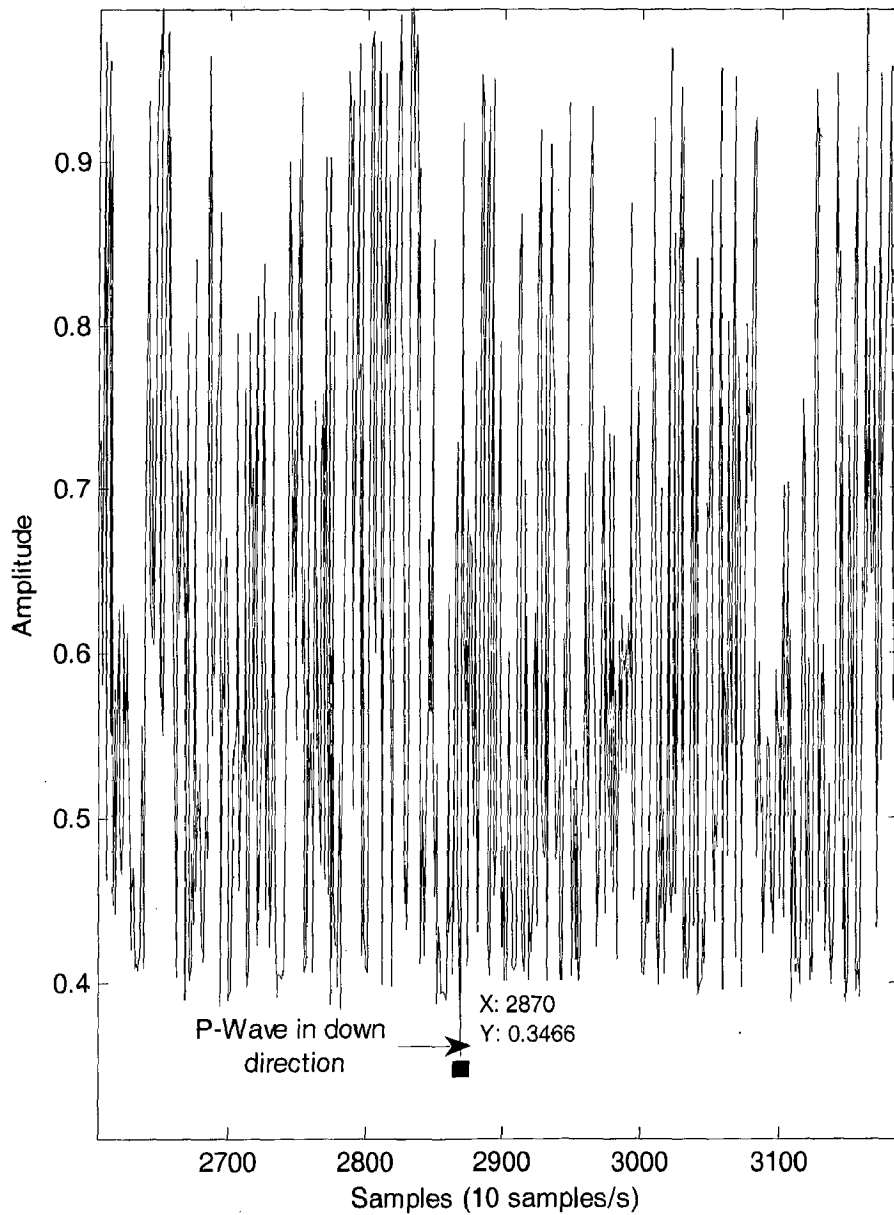
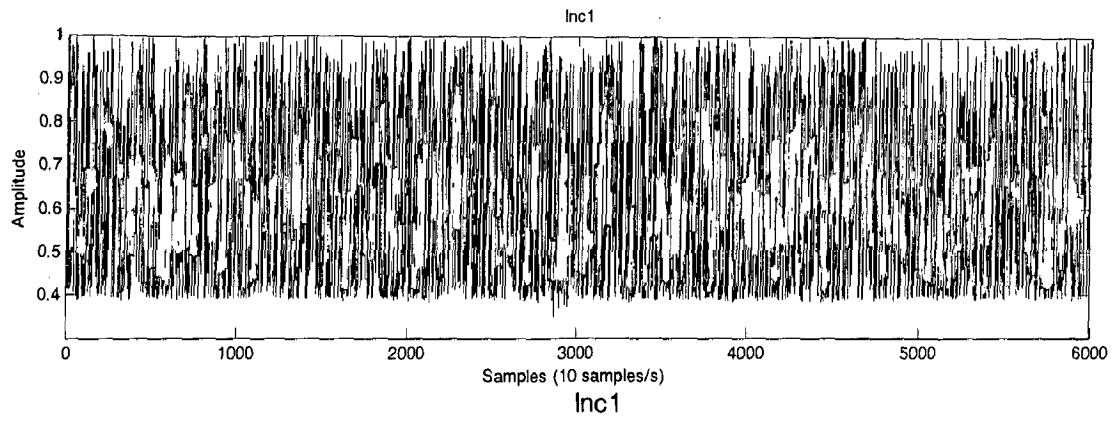


Figure4.5: Represents Inc1 attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Inc1 attribute

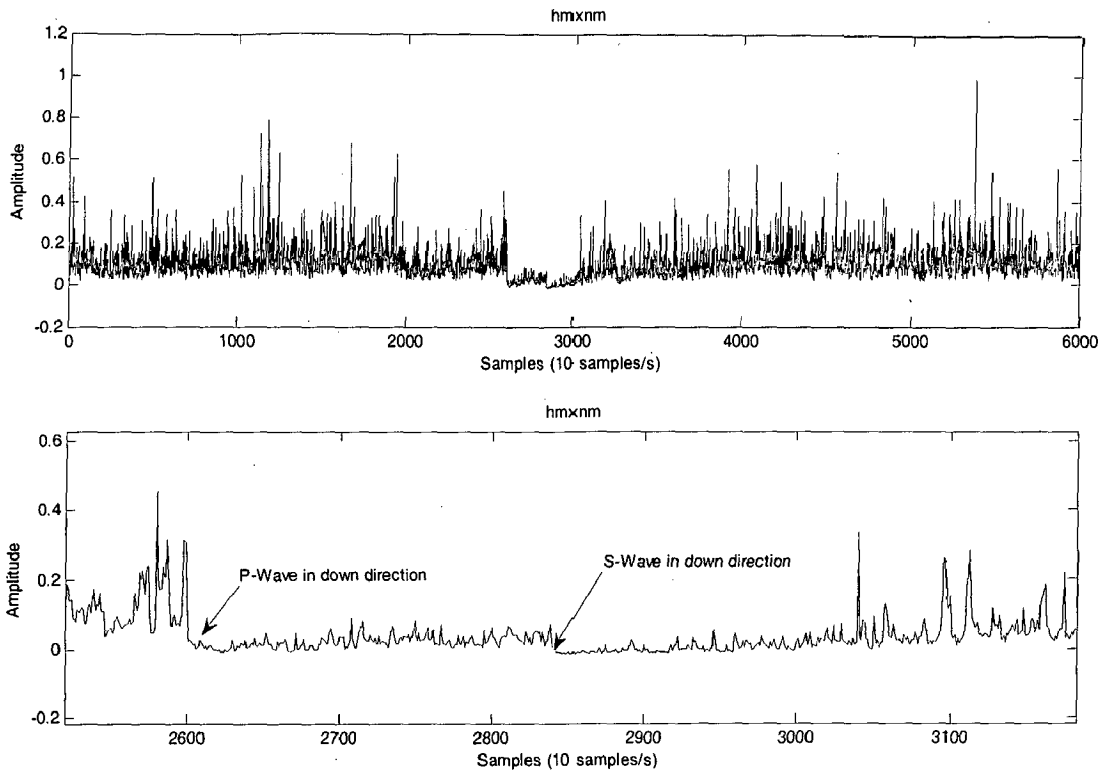


Figure 4.6: Represents Hmxnm attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Hmxnm attribute

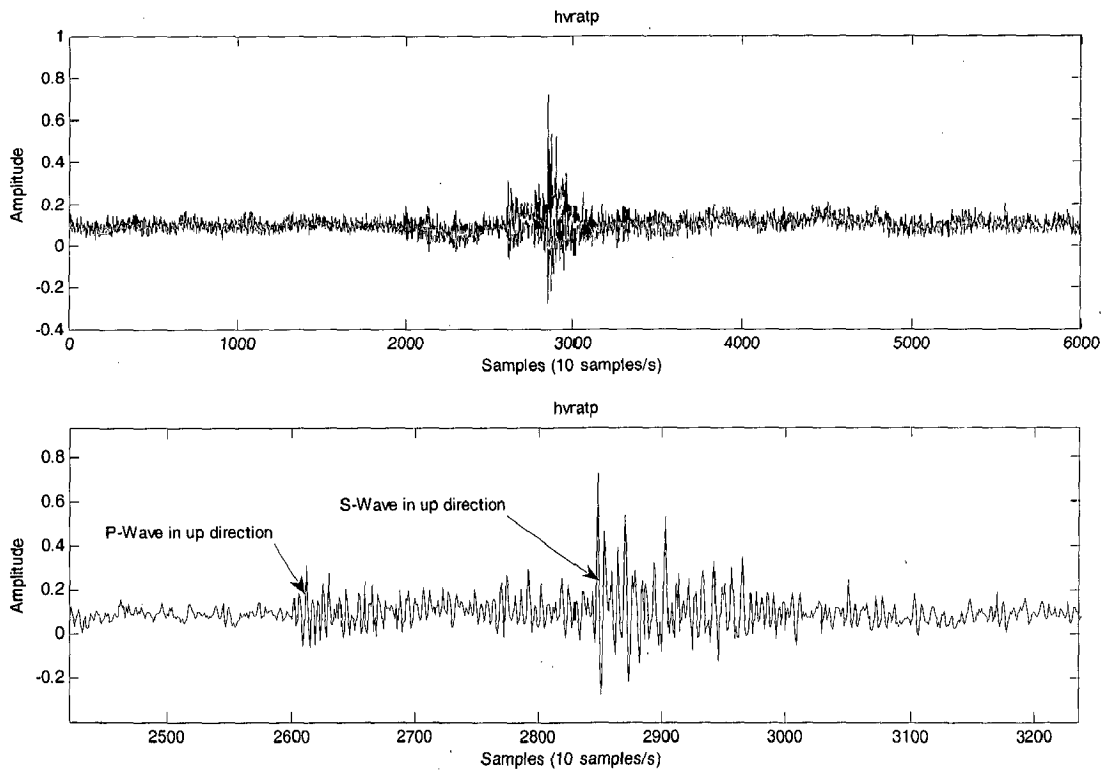


Figure 4.7: Represents Hvratp attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Hvratp attribute.

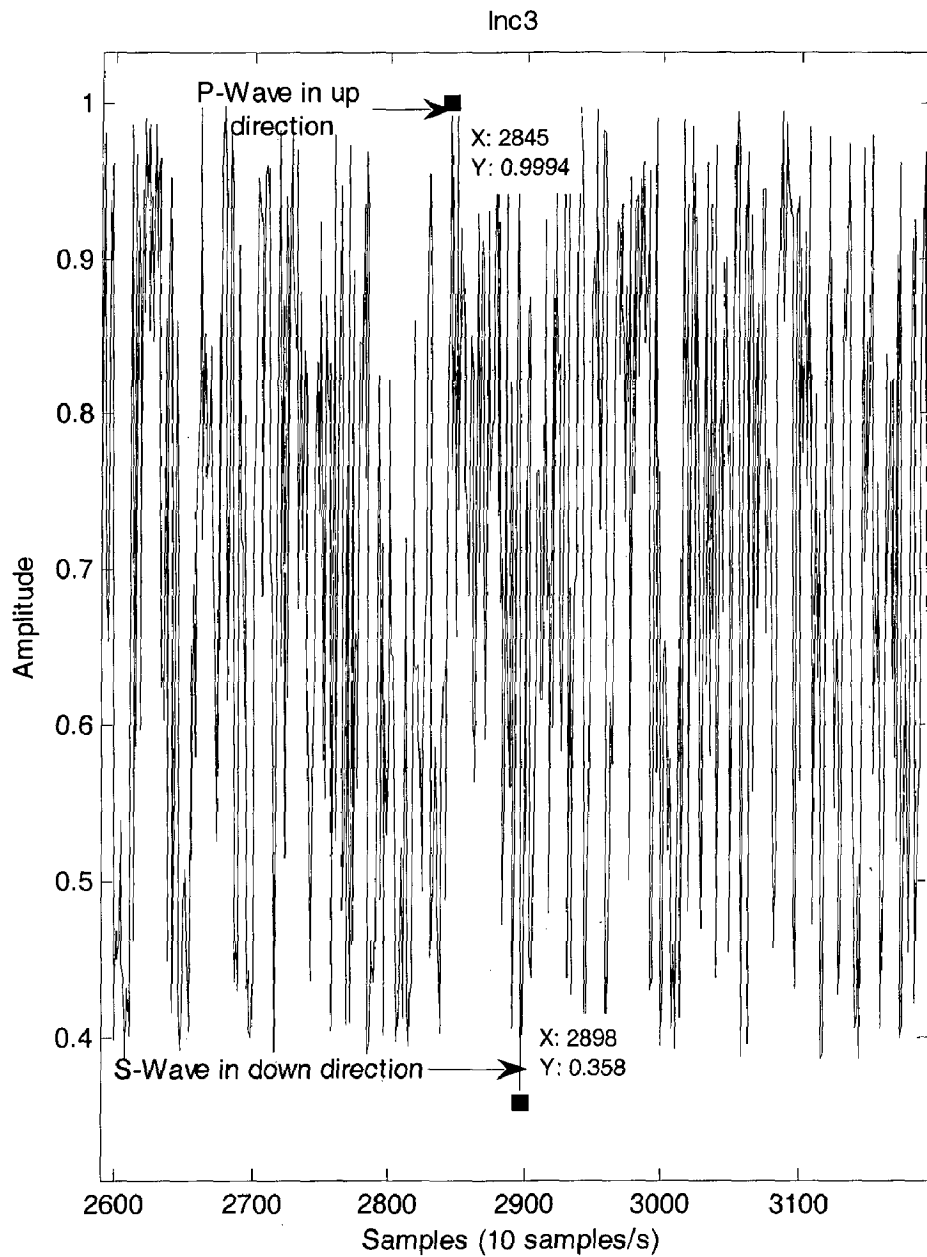
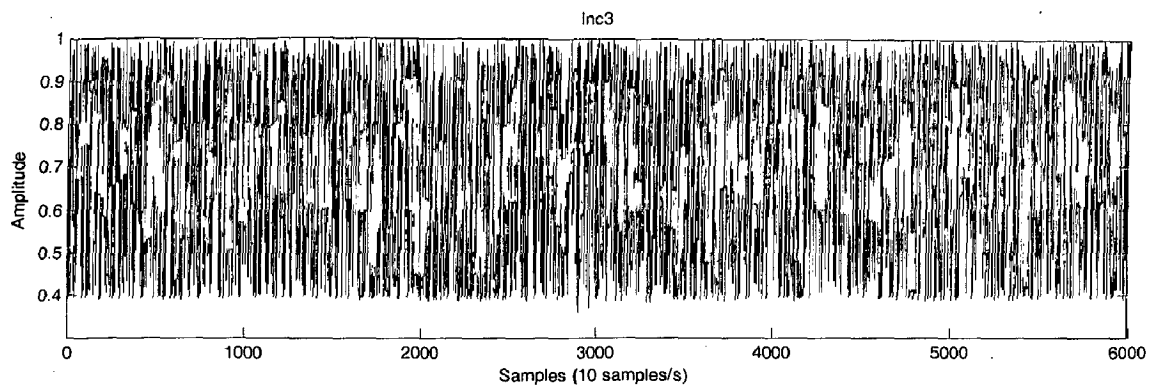


Figure 4.8: Represents Inc3 attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Inc3 attribute.

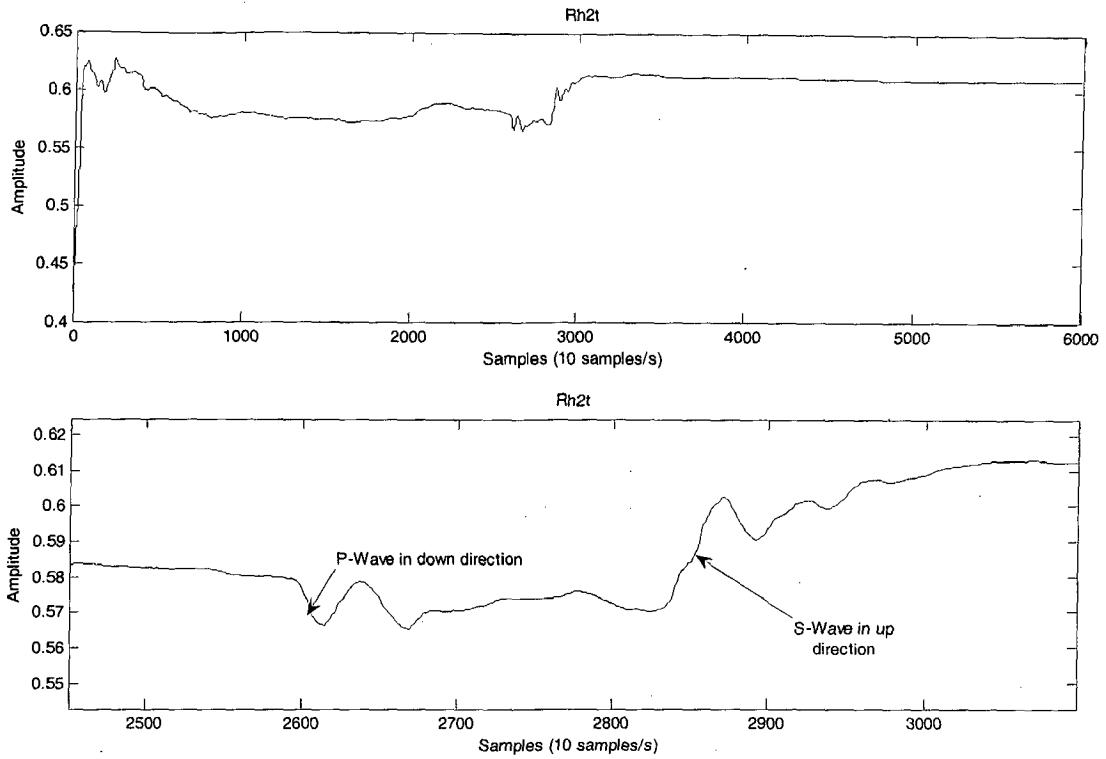


Figure 4.9: Represents Rh2t attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Rh2t attribute.

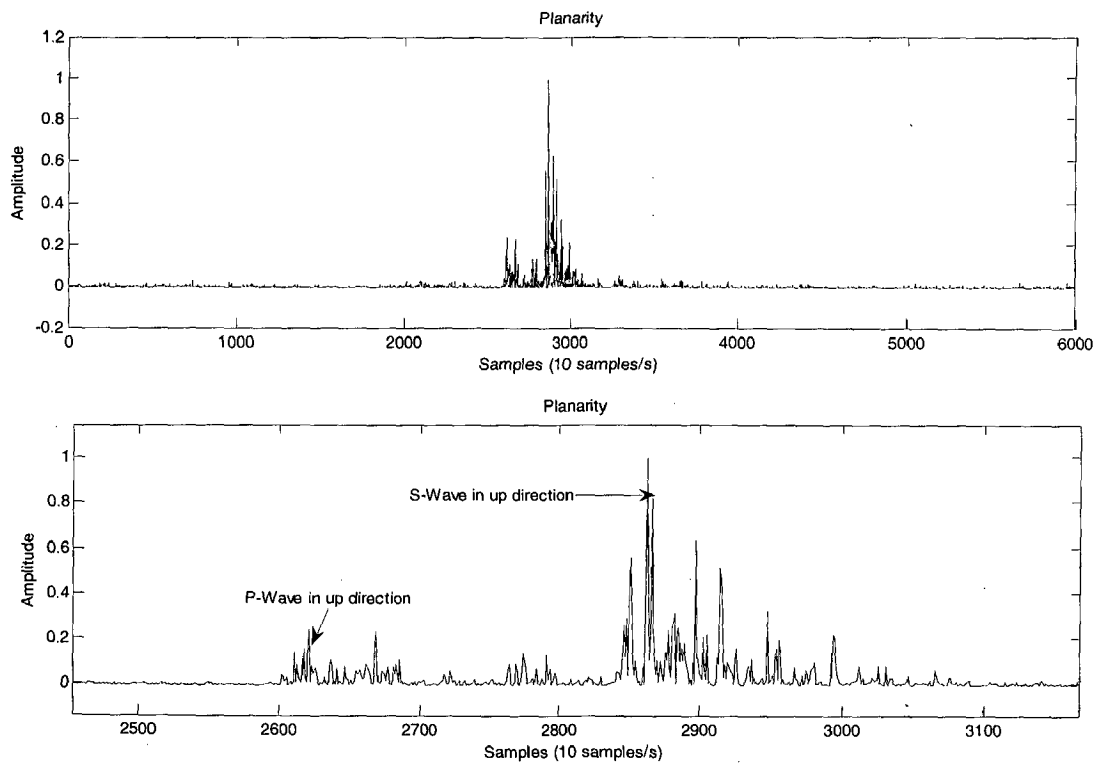


Figure 4.10: Represents Planarity attribute time series for the seismograms in the above figure and the second figure an enlarged window represents the response direction of P and S wave in Planarity attribute.

AUTOMATIC PHASE PICKING OF SEISMIC SIGNALS USING ARTIFICIAL NEURAL NETWORK

5.1 Introduction

As seismic networks have expanded dramatically in recent years more and more data must be processed, the hand picking procedure for seismograms has become not only time consuming but also inaccurate in the sense that reading error is easily influenced by the analyst. A reliable and accurate automatic phase picking system is therefore greatly in demand in modern seismology, not only in the studies of the earth structure but also in the other fields such as real time seismology, which could make use of an automatic phase picking system to automate epicenter determination, mechanism calculations and so on.

A phase picking task contains two steps; 1) Detecting a phase and 2) Reading its arrival time. A few applications of ANNs to Seismology have been carried out. For example, Del Edoardo Pezzo *et al* (2003), have applied Artificial Neural Networks to the explosion seismology for discriminating natural earthquakes versus explosions. and Robert Essenreiter *et al* (1998) have applied Artificial Neural Networks to reflection and McCormack Michael *et al* (1993) for refraction studies. Sharma Mukut *et al* (2005) have developed an approach to earthquake prediction in Himalayas using neural network techniques. Feng Xia-ting (1997) have applied neural network for modeling earthquake magnitude series. Pathak (2002), (1998) *et al* have applied neural networks for simulation of earthquake design in the himalayan region. Dai and Mcbeth (1995) have performed a study to test the ability of an ANN to detect and pick seismic arrivals. Jin Wang and Taliang Teng (1997), has designed two types of ANN-based based detectors for event triggering and *P* Picking and in (1995) as seismic detector. Wang Jin (2002) applied neural networks for seismic phase picking. We have designed a neural network system as an seismic phase picker that uses eight attributes in which four attributes are used as an

P-phase picker and another four attributes as an *S*-phase picker. The objective of this study is to design an ANN based automatic phase picker for seismic signal.

5.2 IDENTIFICATION OF SEISMIC PHASE USING ANN

We have designed a neural network system as an seismic phase picker that uses eight attributes in which four attributes are used as an *P*-phase picker and another four attributes as an *S*-phase picker. The below figures represent the block diagram for both *P*-phase picker and *S*-phase picker.

5.2.1 P-Phase identification using ANN

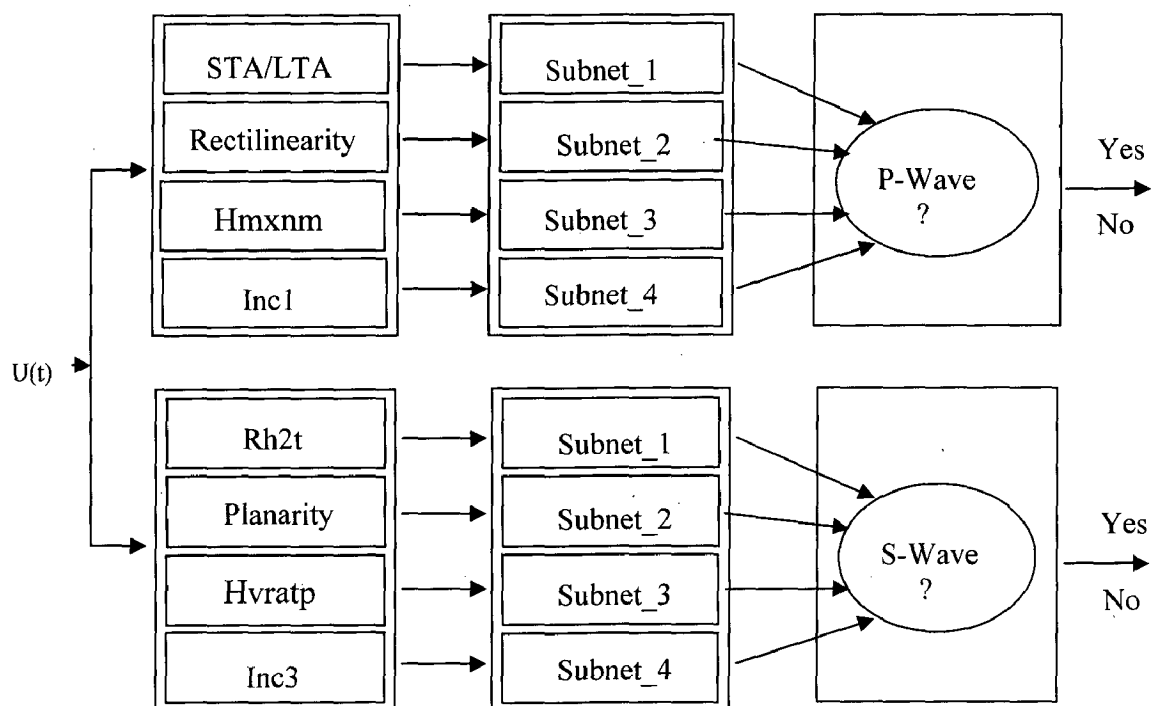
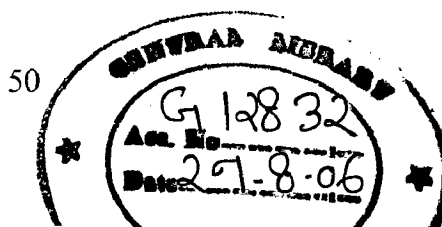


Figure 5.1: Block diagram for the development of the ANN Seismic Phase detector. It consists of four preprocessors, four neural subnets, and one final decision neuron.

$U(t) = [Z(t), N(t), E(t)]$, where Z,N,E represents three components of signals namely vertical, north and east direction.

For *P* Phase detection, the three component seismograms are input to the system, four attributes are computed by four preprocessors. The four attributes used here are 1) Ratio of short term average to long term average (STA/LTA) which is calculated from



one horizontal component 2) Rectilinearity which is calculated from the eigen values of the covariance matrix 3) Ratio of maximum to minimum horizontal amplitude (hmxmn) which is calculated from the eigen values of the horizontal component 4) The incident angle of long axis of eigen vector of the covariance matrix (Inc1). These attributes which are used as preprocessors, are given as input to the neural network, trained and the output that is, the phase detection and picking is then carried out. Each neural network process one attribute at a time and a final decision neuron to declare the phase arrival. The explanation of the optimization and training of Backpropagation Neural Network is given below.

5.2.2 Optimization and Training of the BPNNs

A BPNN must be trained before any picking work is carried out, but we first have to determine some important details concerning the network's architecture and parameters. Unfortunately, despite the wide use of BPNNs and their success in solving a variety of problems, there is still no substantial theoretical support for how the architecture and parameters should be determined. Usually, however, they are determined by constantly monitoring the performance of the network with test data.

5.2.2.1 The Architecture of the BPNNs

The Number of Layers: Values in a window containing n dots can be regarded as a vector in an n dimension space. BPNNs create the decision boundary for the n dimension space, making it possible to recognize patterns. Any given decision boundary can closely be approximated by a two layer network having a sigmoid activation function. Thus, we apply a two-layer network-one hidden layer and one output layer-in our system. Note that the input layer is not counted in the number of layers because there is no computation carried on this layer.

The Number of Input Nodes: One sample point corresponds to one input node in a BPNN. Because the sampling rate of the data is 100 samples per second, we reduce the sampling rate to 10 samples per second; the number of input nodes for the detector is 50.

The Number of Hidden Nodes: Many important issues, such as determining the appropriate number of hidden nodes in a network for a specific task are solved in practice by trial and error. The issues involved in this process are complex because they are highly problem dependent (Fu Limin 2003). With too few hidden nodes, the network may not be powerful enough for a given learning task. With a large number of hidden nodes, computation is too expensive, and its generalizability is more limited. In general, a network with a large number of nodes is capable of memorizing the training set but may not generalize well. Therefore, smaller networks are preferred over larger ones. Our experiments show that when the number of hidden nodes is 1/10 the number of input nodes, the network is powerful enough for accurate detection to occur. We also found that network performance is not sensitive to a small change in the number of hidden nodes and therefore we choose 4 neurons in the hidden layer.

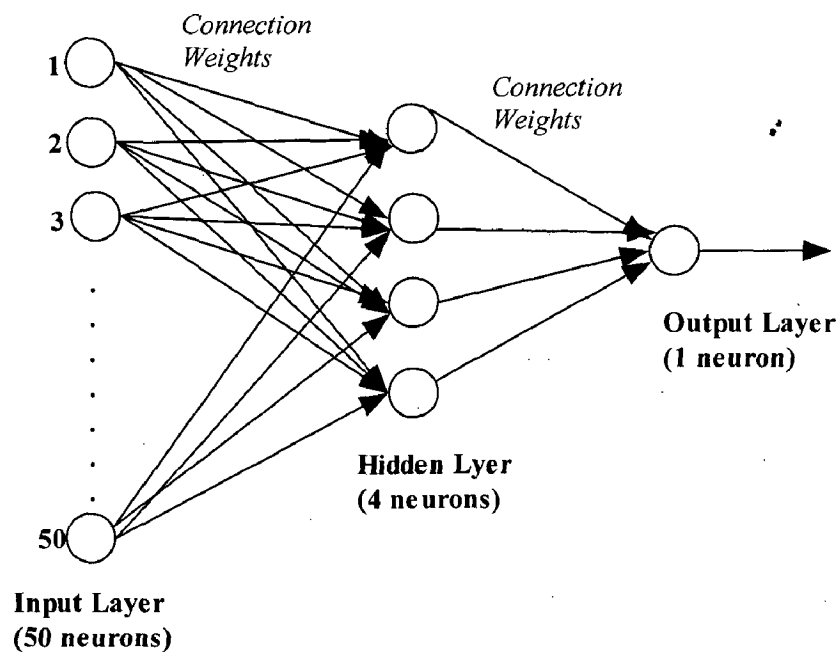


Figure 5.2: Structure of the ANN Phase Identification.

The Number of Output Nodes: When BPNNs are used for classification, there are usually N output nodes for an N class problem, with each node representing one class. For a two class problem, however, there is another approach that uses only one output node, with the ON and OFF states standing for each class, respectively. Usually, the two approaches make no essential difference, but the former approach is a general purpose one that can be applied to other multi-class problems. We therefore use a single output node in our system, which is trained to an output (ON) for signal patterns and an (OFF) for noise patterns.

5.2.3 Training function (TRAINGDM)

The performance function or error criteria (goal) for the algorithm have been set to Mean square error (MSE). The same neural architecture is trained for different values of goals and corresponding output values are compared with actual predicted values from calculation. The training algorithm used is TRAINGDM (Gradient Descent method). Gradient descent algorithm is used, as it improves the performance of ANN by reducing the total error by changing weights along its gradient. TRAINGDM is a network training function that updates weight and bias values according to gradient descent with momentum.

The Parameters of the Neural Network Training function is explained below

1. The range of randomly chosen initial weight values.
2. Slant of sigmoid activation function:
3. Values of network's ON and OFF: $(1 - \epsilon)$ and ϵ
4. Network's learning rate: η
5. Momentum coefficient: α

Generally, training begins with randomly chosen initial weight values. This phenomenon results from the behavior of the sigmoid function. A network's training efficiency can be improved by adjusting ϵ in the sigmoid function $F(x) = 1/(1 + e^{-\epsilon x})$, while our system works quite well when ϵ is simply set to 1 because we do not use

networks with large dimensionality. The sigmoid function also tells us that the output value will be 0 only when $x = -\infty$ and will be 1 only when $x = \infty$. Moreover, because the modification of weights is based on the derivative of the sigmoid function, the first derivative of the sigmoid function is very small for the large x . It is therefore preferable to use a larger value of ϵ than 0 as the desired output values. The learning rate η refers to the coefficient of the negative gradient that determines the change of weight in Backpropagation. A large value of η will lead to rapid learning, but the weight may then oscillate, whereas low values imply slow leaning. Values between 0.1 and 0.9 have been used in many applications; η is set at 0.65 in our application. Another coefficient to be determined is the so-called momentum coefficient α . The momentum is added to the weight update rule so that the training will not get stuck in some local minima and significantly reduce the number of iterations for convergence. In our system, we set to be 0.7.

5.2.3.1 Limitations and cautions of TRAININGDM

The gradient descent algorithm is generally very slow because it requires small learning rates for stable learning. The momentum variation is usually faster than simple gradient descent, since it allows higher learning rates while maintaining stability, but it is too slow for many practical applications. These two methods would be used only when incremental training is desired. You can use Levenberg-Marquardt training (Hagen et al, 2002) for small and medium size networks, if you have enough memory available. If memory is a problem, then there are a variety of other fast algorithms available.

5.2.4 Transfer or Activation functions

The selection of activation function plays an important role in designing neural network. In the present model (network) a bipolar sigmoid function $f[u(t)] = \tanh[g.u(t)]$ for the hidden layer and linear function $f[u(t)] = \log[g.u(t)]$ for the output layer were considered. The transfer functions used in the proposed neural network are TANSIG and LOGSIG. The training input vectors which are normalised values between 0 and 1. So, the

transfer function used in the input layer is TANSIG. The transfer function used in the output layer is LOGSIG.

5.2.5 Selecting Training Data Sets for P-phase

Selecting appropriate training data sets is one of the most critical steps in a successful training. Our training data set is a subset of actual data. For our phase-picking task, the training set should include both signal patterns and noise patterns. The signal pattern should represent the typical features of a signal with different frequency characters. We do not include the later phase signal patterns because they are much more complex and we do not have enough prepicked data for training and testing. Also, we exclude those patterns with a very low S/N ratio that can hardly be distinguished from noise patterns. A signal's onset point is fixed at a certain position in the moving window, following Dai's technique. The part before the onset is made longer than the part after it in order to achieve better distinction between the signal patterns and noise patterns. Further, the fixed onset point for training signal patterns provides a basis for determining the onset time from the detector's output. The noise patterns are extracted from the noise parts of the seismic waveforms and are the same length as the signal patterns. The below *figure5.5* provides details regarding the signal patterns and noise patterns for each attributes of *P*-wave. How many patterns are needed for good training? And among these patterns, how many should be signal patterns and how many noise patterns? A rule of thumb is that the patterns in the training set should cover the main categories of signal and noise patterns. And Therefore I selected 10 noise pattern categories and 10 signal pattern categories. The below figure shows the corresponding attributes of the training data for *P*-waves and background noises. The STALTA sequence normalized by the maximum value in each window indicates the signal-to-noise variation with time. The maximum signal-to-noise ratios of the training patterns for *P* wave ranges from 20-60 which result in a group of step like patterns while the background noises have a group of flattened lines. The signal patterns for Rectilinearity vary from 0.2 to 1.0 while the background noises are less 0.1. The signal patterns for ratio of maximum to minimum horizontal amplitude (hmxmn) which is calculated from the Eigen values of the horizontal component decay from 0.1 to 0, where as the background noise vary from 0.2

to 1. The Inc1, a polarization measure derives from the covariance matrix and normalized by 90° , has very stable low values for the *P*-waves. All patterns of Inc1 are below 0.35, corresponding to the long-axis incidence angle less than 22.5° , which indicates that most of the *P*-wave energy is distributed in the horizontal plane. Then the four neural networks were trained with the training sets.

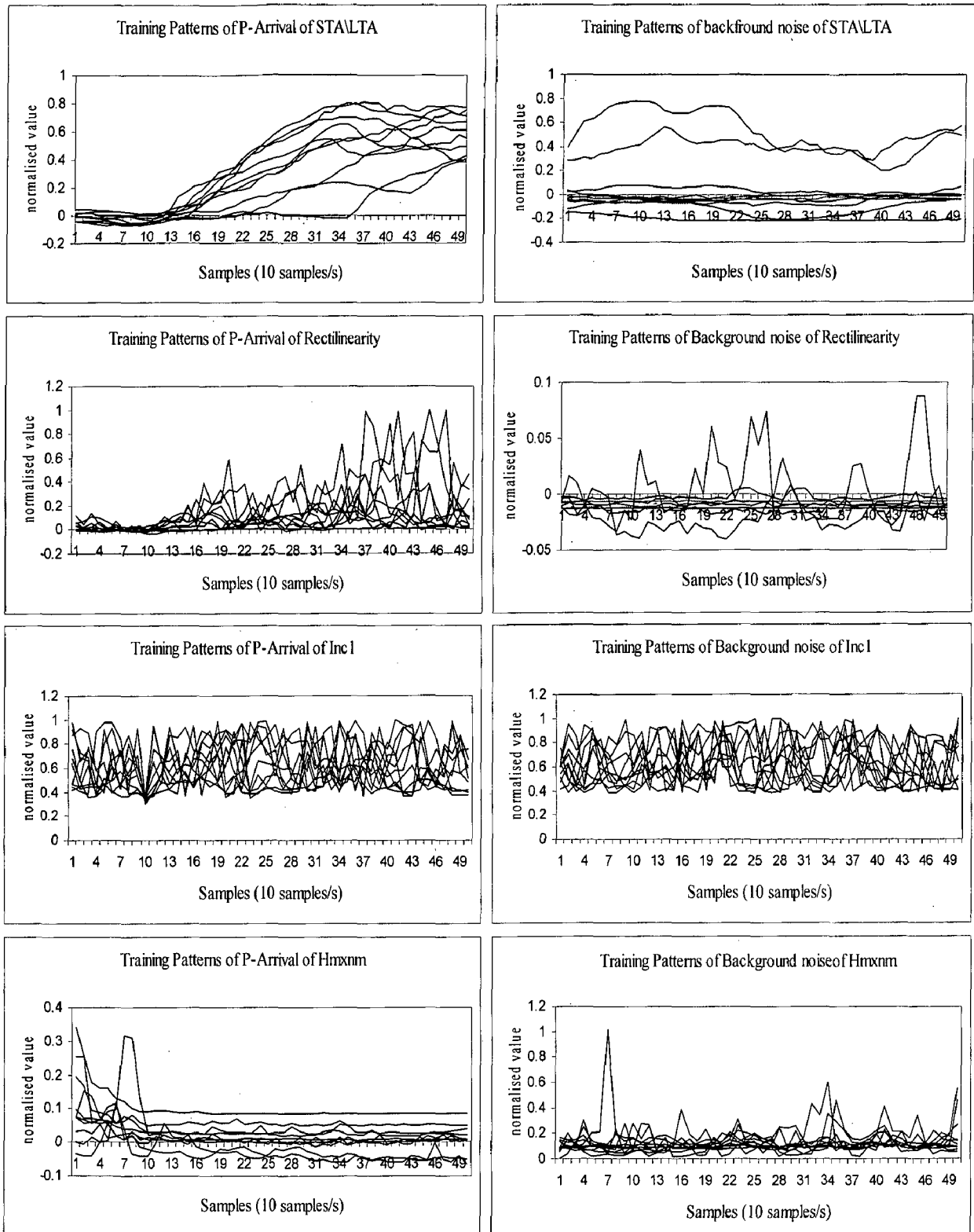


Figure 5.3: The training patterns of four attributes for P-arrivals and background noises. P-wave window was selected from 10 points before P arrival to 40 points after P-arrival. The background noises were extracted just before the signals arrive.

5.2.6 Termination of Training

In the neural computing area, there is a phenomenon called overtraining. This phenomenon refers to the idea that the network performs poorer with test data when it is over-fitted to the training data. Hence, training iteration should be halted at a certain point even though the error for the training set can be reduced further. In order to achieve proper termination of the training process, we form another independent data set that has the same number of samples as the training set and generally is called a validation set in the neural computing world. The error measured with the validation set often shows a decrease at first, followed by an increase as the network starts to over fit. Training can therefore be stopped at the point of smallest error with respect to the validation set.

5.2.7 Phase Detection of P-Wave

To detect the *P*-Phase we first calculate the observed preprocessing for each attribute, taking each windowed segment of this, and feed it to the trained neural network. We shift the window by one sample at a time and feed each segment in to the network, storing the output. The procedure is repeated until the end of the seismogram is reached. In general the output lies between the ideal for a signal or for a noise. The below figures represent the corresponding input and output of the neural network for each attribute. In each attribute an arrival corresponds to a sharp change so that a threshold may be sufficient to detect the arrival and a marker in each figure indicates the detection of *P*-phase.

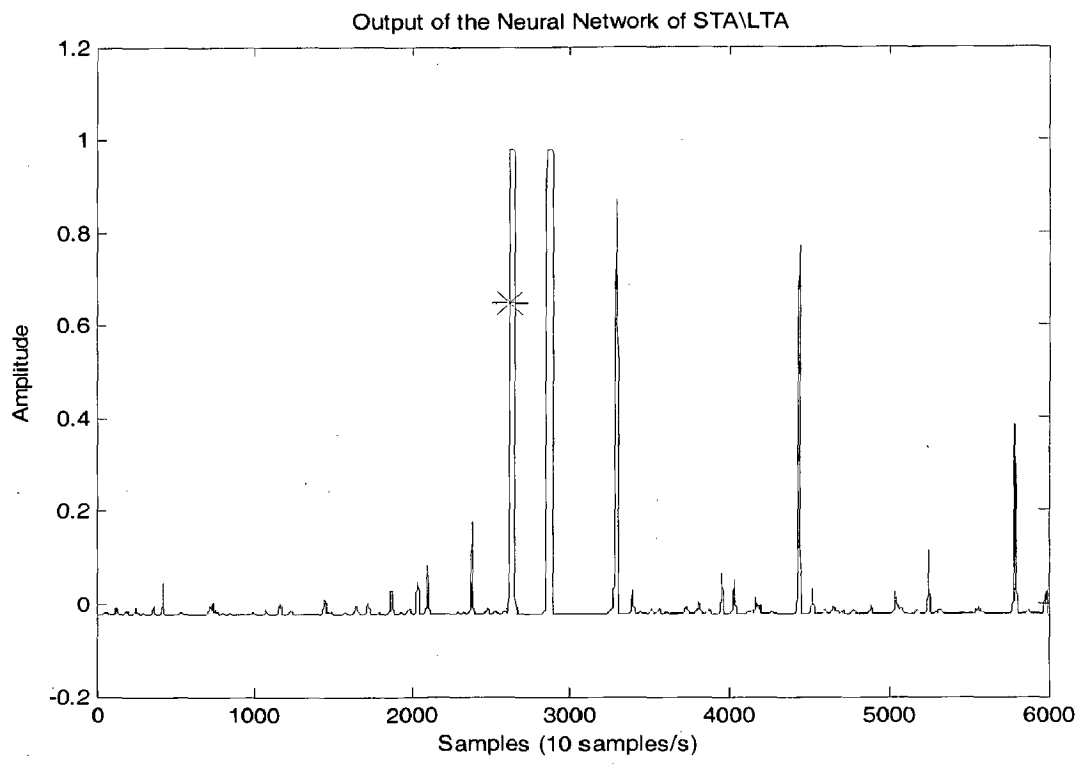
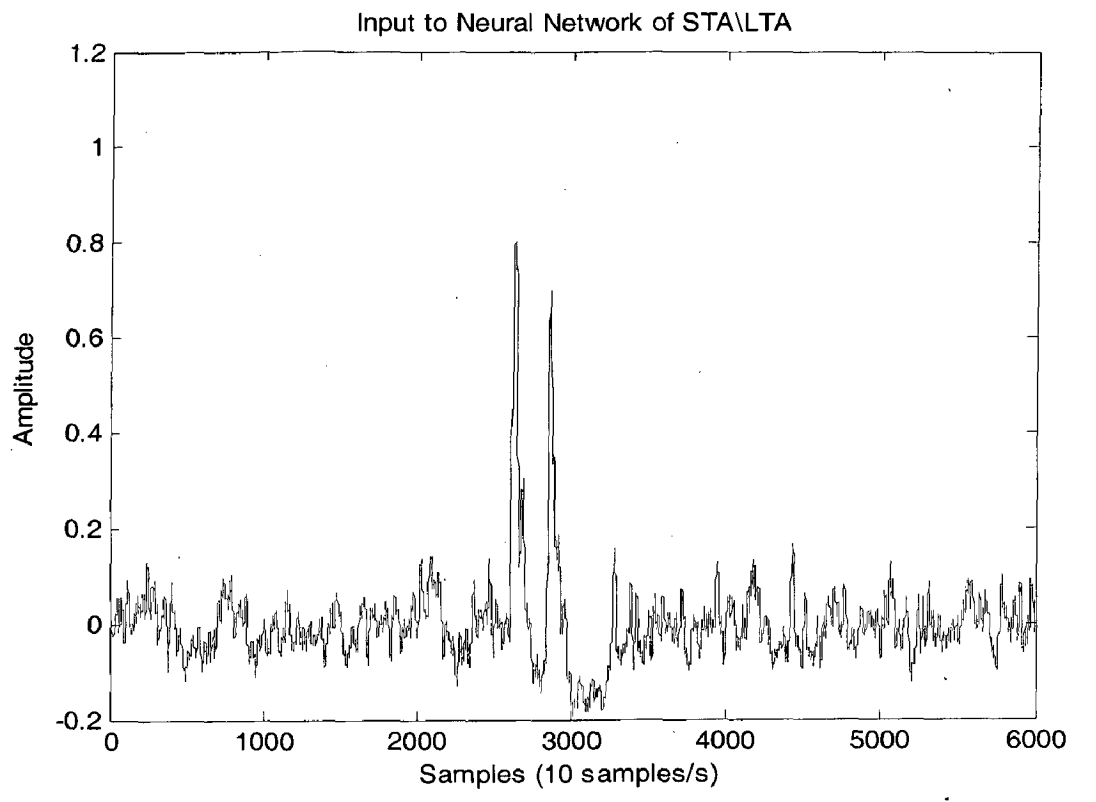


Figure 5.4: Detection of P-phase by STA\LTA algorithm, the first peak with maxima greater than 0.6 is identified as P-wave

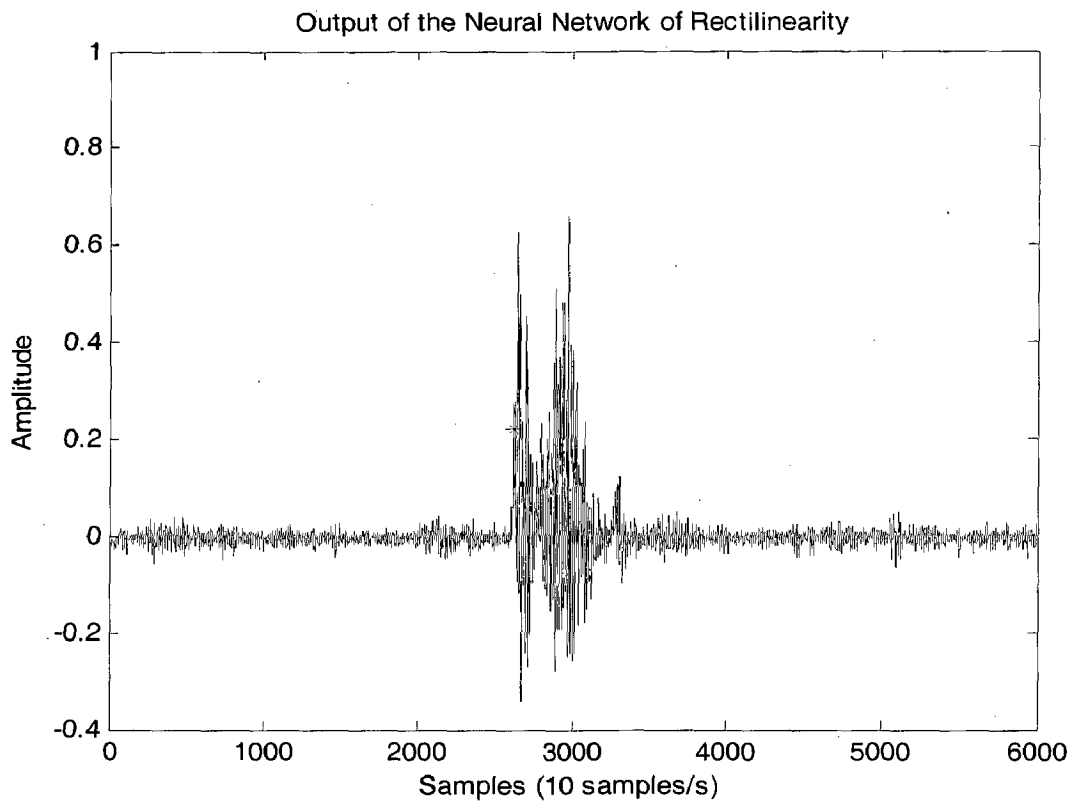
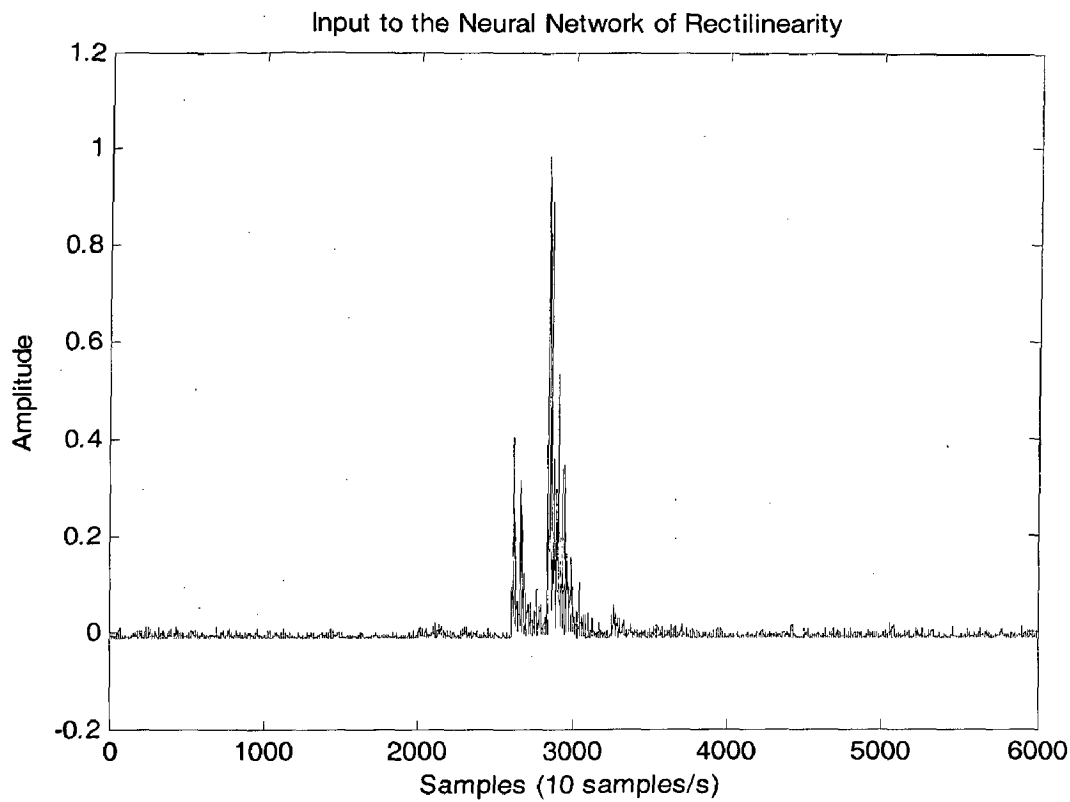


Figure 5.5: Detection of P-phase by Rectilinearity, the first peak with maxima greater than 0.2 is identified as P-wave

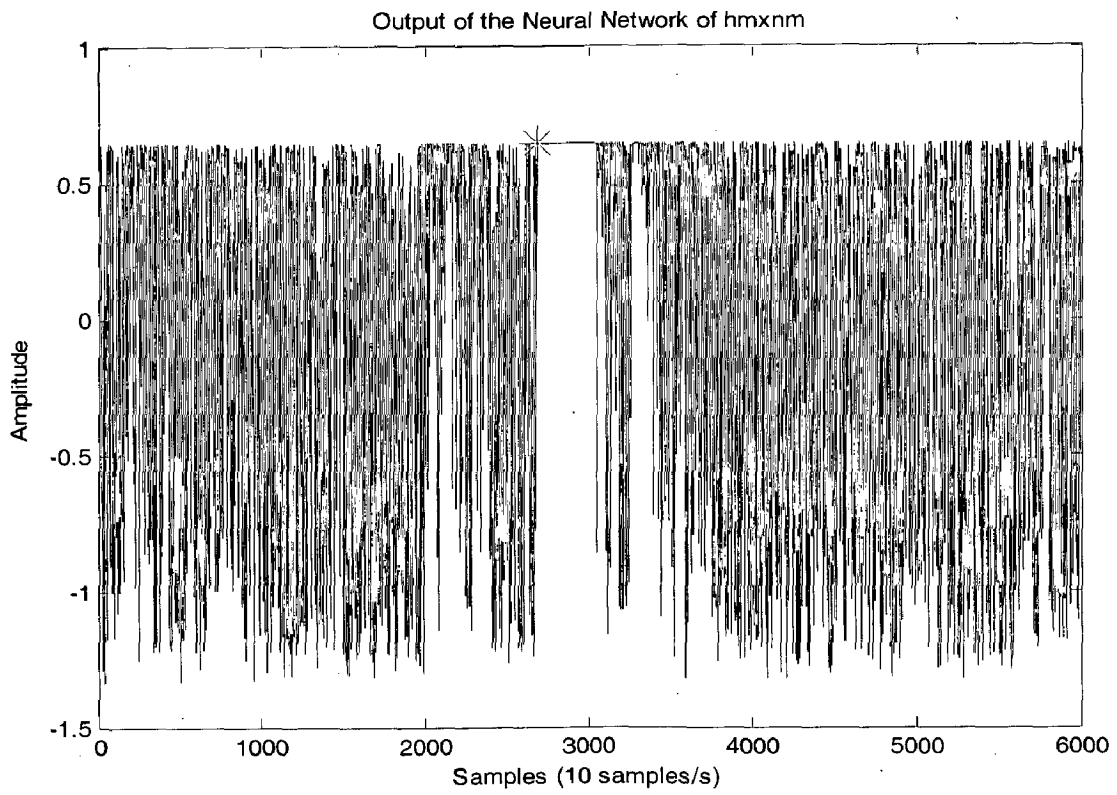
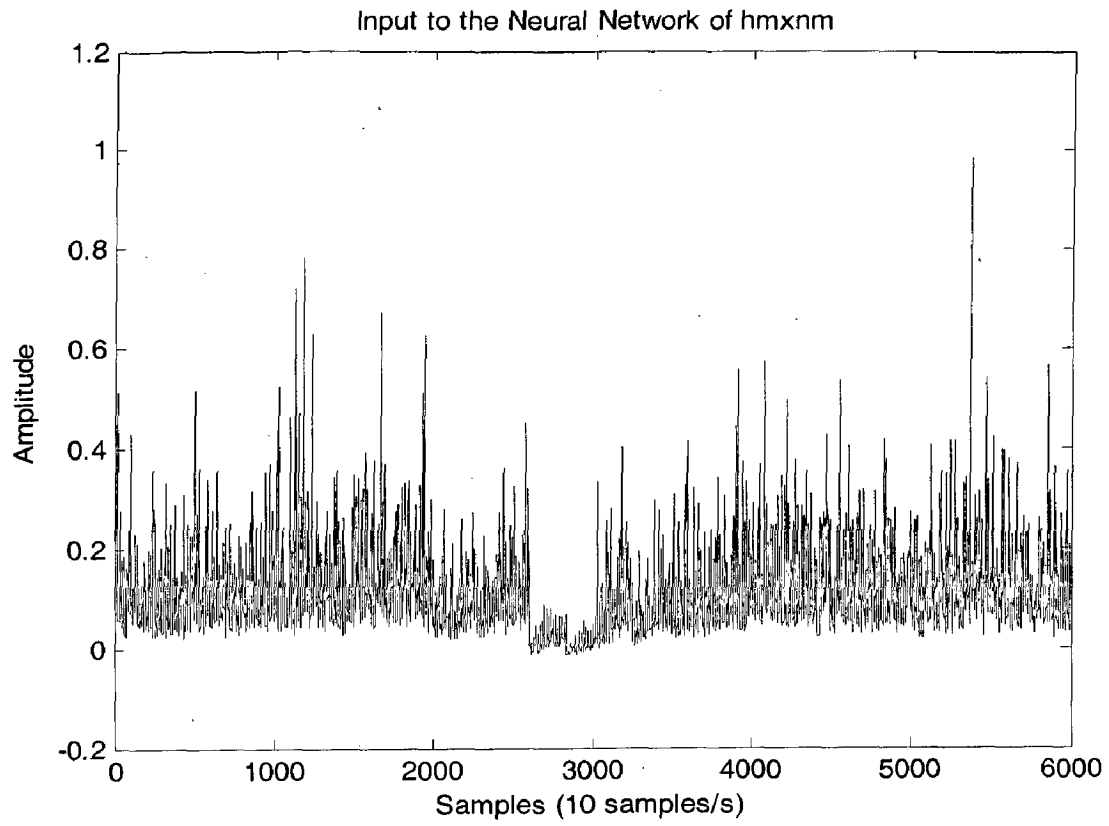


Figure 5.6: Detection of P-phase by Hmxnm

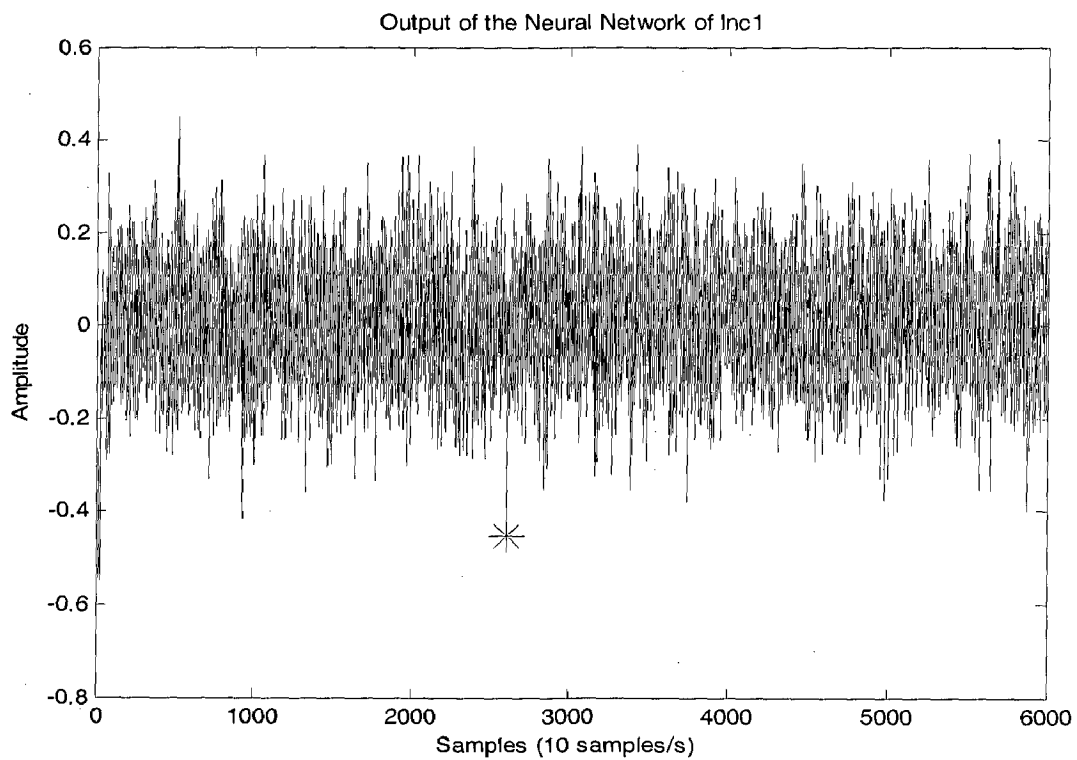
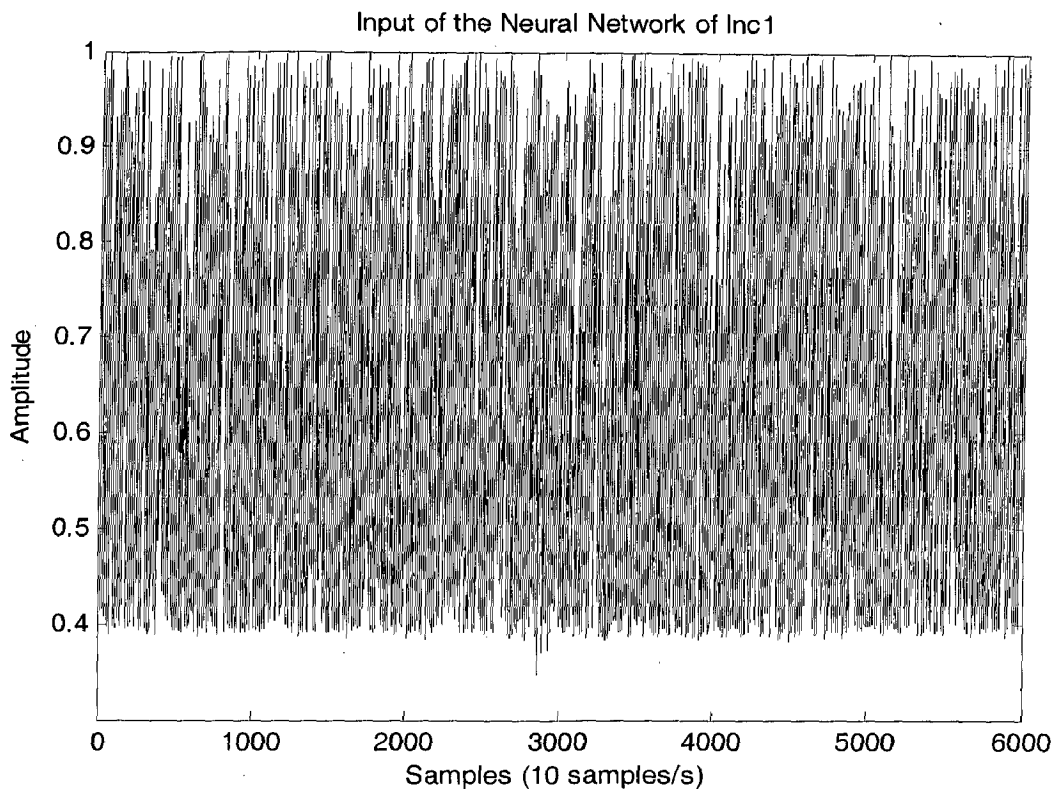


Figure 5.7: Detection of P-phase by Inc1, the maxima peak towards the negative direction is identified as P-wave

These attributes were tested on a data set of seismic events from Gharwal Kumaon Region. Around 25 events of varying length (20 sec-15 min) are checked using these algorithms. The results show that STA \ LTA algorithm has detected 68%, Rectilinearity has detected 84%, and Inc1 has detected 56% and 76% by Hmxnm. In overall 72% has detected correctly for *P*-phase

5.2.8 S-Phase identification using ANN

For *S*-Phase detection, the three component seismograms are input to the system, four attributes are computed by four preprocessors as shown in the *figure5.1*. The four attributes are as follows: (1) Ratio between horizontal power to total power (Rh2t), (2) Planarity, (3) Ratio of horizontal-to-vertical power (Hvratp), (4) Short -axis incidence angle of polarization ellipsoid (Inc3). The *S*-phase detection is identified similarly as the *P*-phase is identified; it has the similar structure of neural network 50 input neurons, 4 neurons in the input layer and 1 in the output layer. Similarly, the four neural subnets are input to the final decision neuron.

5.2.9 Selecting Training Data Sets for S-phase

Similar to *P*-phase, 10 noise patterns and 10 signal patterns are chosen. The below figure shows the corresponding attributes of the training data for *P*-waves and background noises. The Rh2t, a polarization measure from signal power, changes typically from 0.4 to 0.7 for outgoing *S*-waves, while background noises have almost the same values for the same window and have various values for different time windows. The Planarity which is a measure of planar characteristic of the polarization ellipsoid indicate a large peaks for the *S* arrival varying from 0.5-1.0, where the background noises are within the range of 0.01-0.1. Hvratp, which is ratio of horizontal to total power, indicates large peaks in signal patterns to *S* arrival. Inc3, a polarization measure derives from the covariance matrix and normalized by 90° , has very stable low values for the *S*-waves. All patterns of Inc1 are below 0.35, corresponding to the short-axis incidence angle less than 22.5° , which indicates that most of the *S*-wave energy is distributed in the horizontal plane. Then the four neural networks were trained with the training sets.

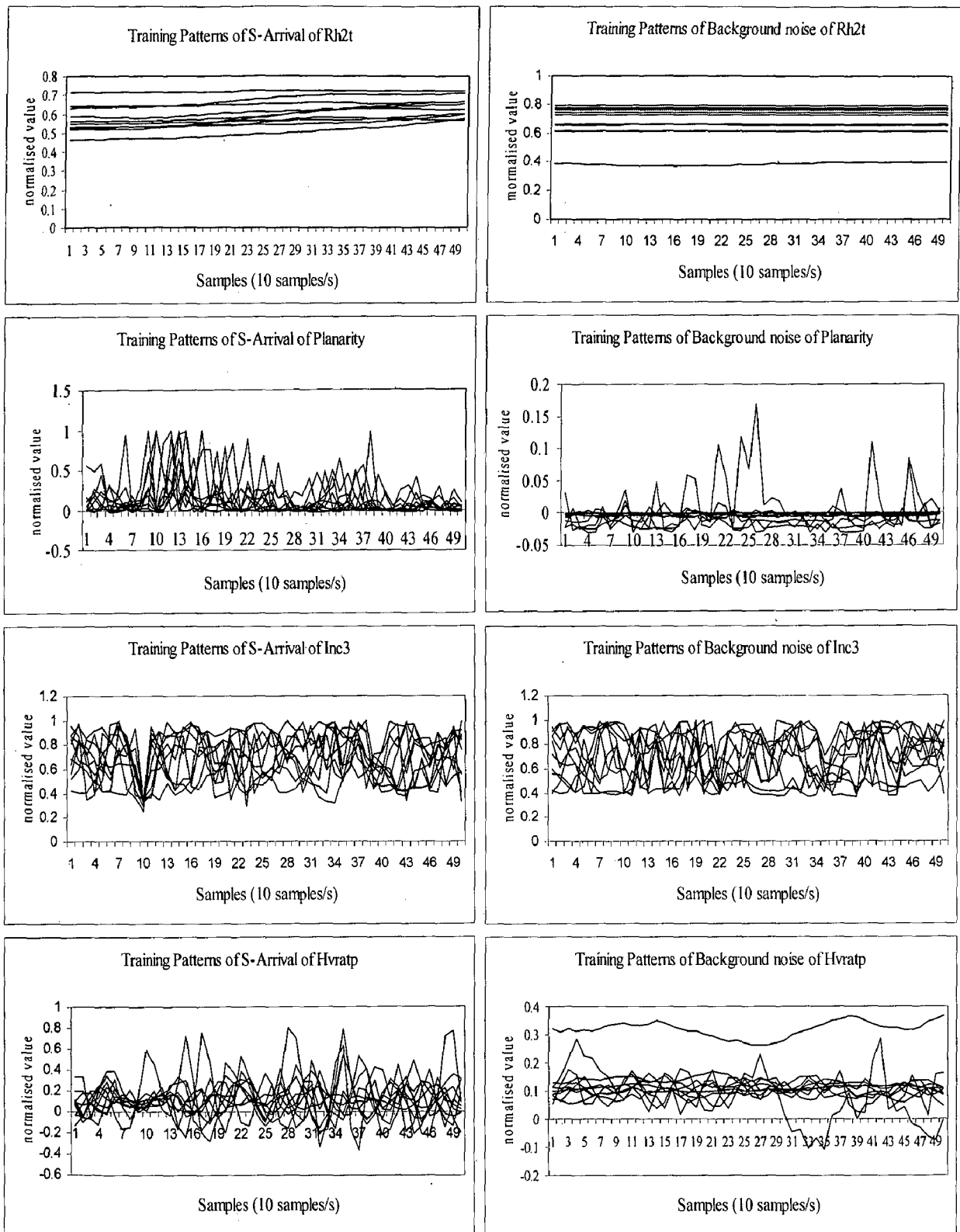


Figure 5.8: The training patterns of four attributes for S-arrivals and background noises wave window was selected from 10 points before S arrival to 40 points after S-arrival. The background noises were extracted just before the signals arrive.

5.2.10 Phase Detection of *S*-phase

Similarly to detect the *S*-Phase we first calculate the observed preprocessing for each attribute, taking each windowed segment of this, and feed it to the trained neural network. We shift the window by one sample at a time and feed each segment in to the network, storing the output. The procedure is repeated until the end of the seismogram is reached. In general the output lies between the ideal for a signal or for a noise. The below figures represent the corresponding input and output of the neural network for each attribute. In each attribute an arrival corresponds to a sharp change so that a threshold may be sufficient to detect the arrival and a marker in each figure indicates the detection of *S*-phase. The results show that Rh2t algorithm has detected 68%, Planarity has detected 92%, and Inc3 has detected 60% and 80% by Hvratp. In overall 72% has detected correctly for *S*-phase.

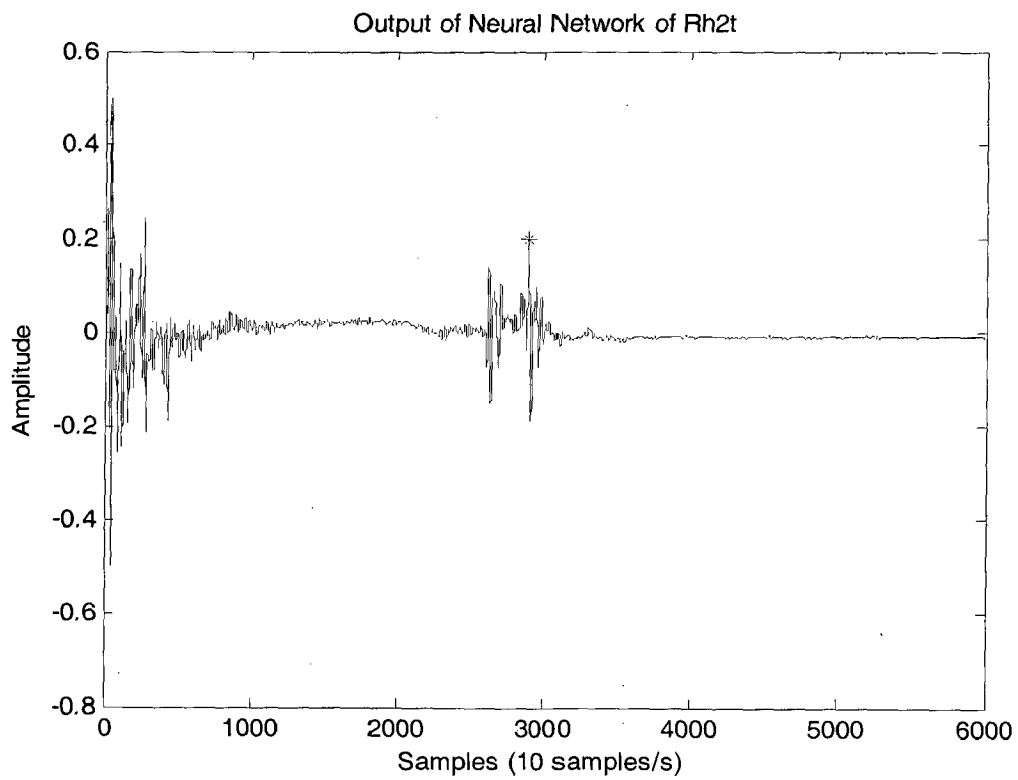
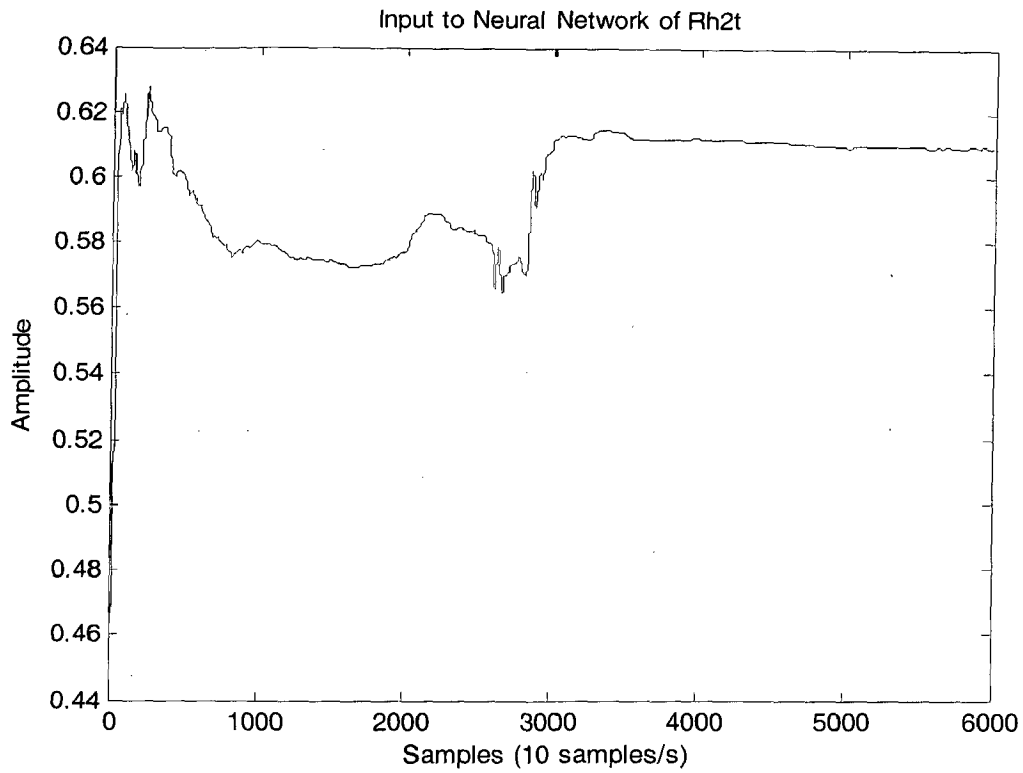


Figure 5.9: Detection of S-phase by Rh2t, the first peak with maxima greater than 0.1 is identified as S-wave

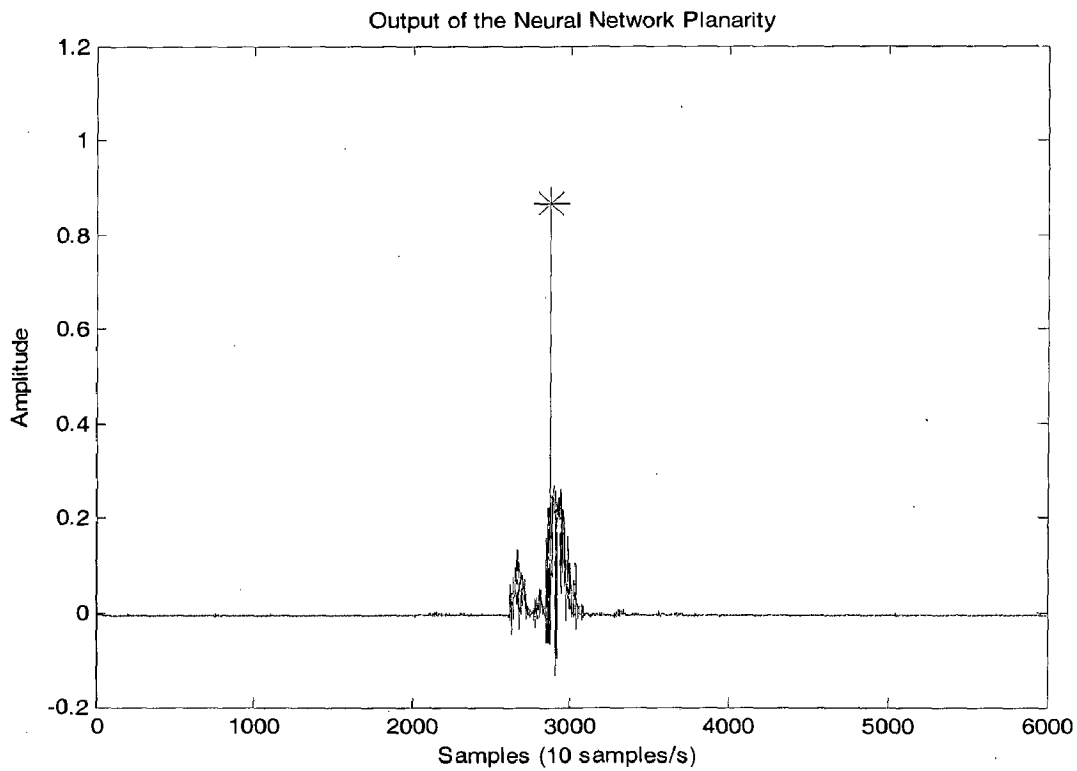
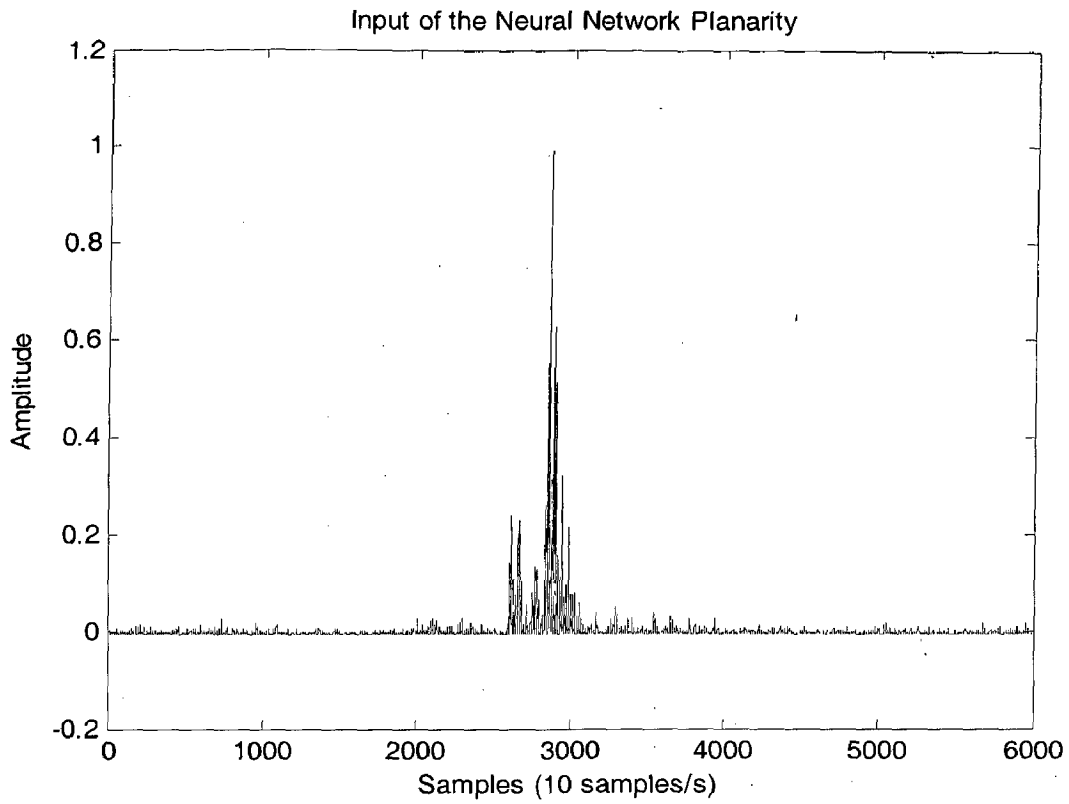


Figure 5.10: Detection of S-phase by Planarity, the maxima peak with threshold greater than 0.6 is identified as S-wave

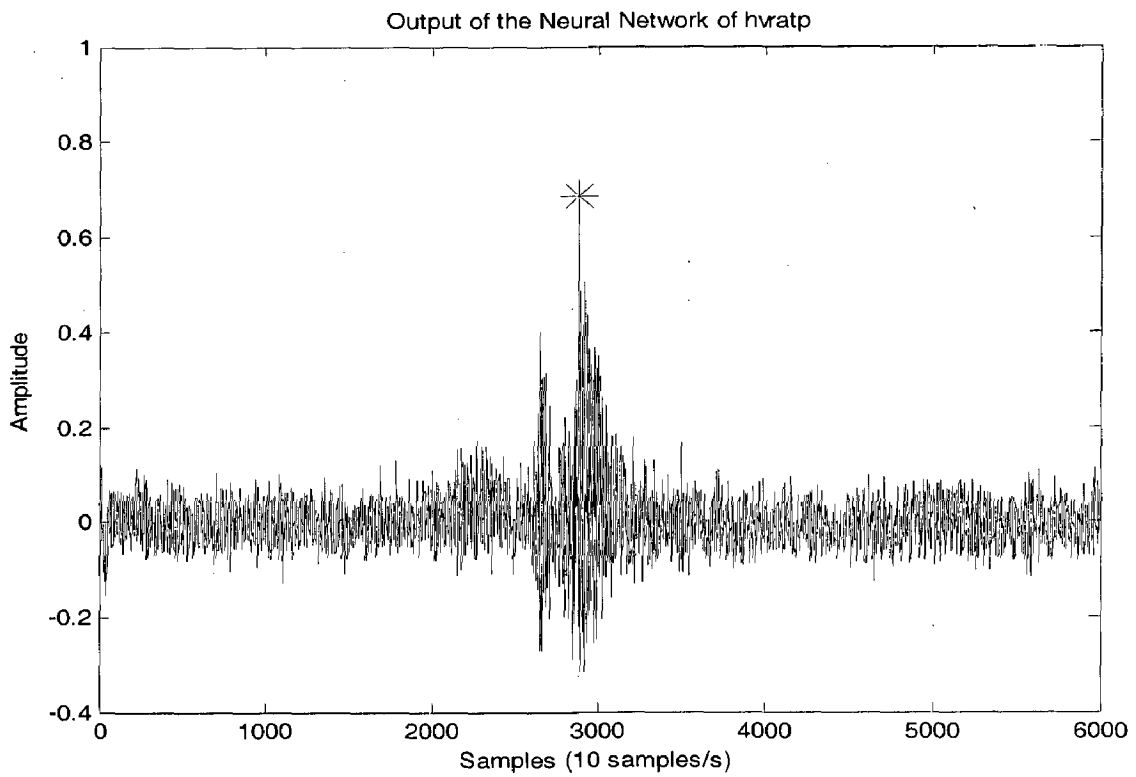
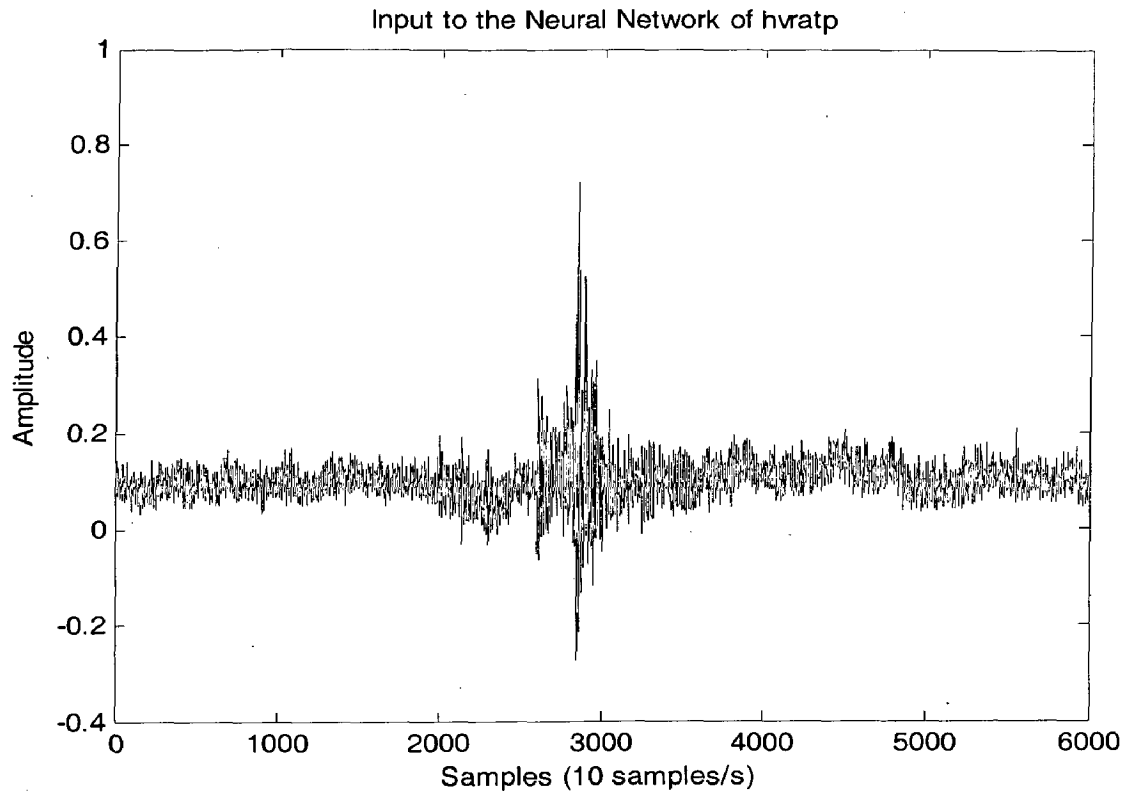


Figure 5.11: Detection of S-phase by Hvratp, the maxima peak with threshold greater than 0.4 is identified as S-wave

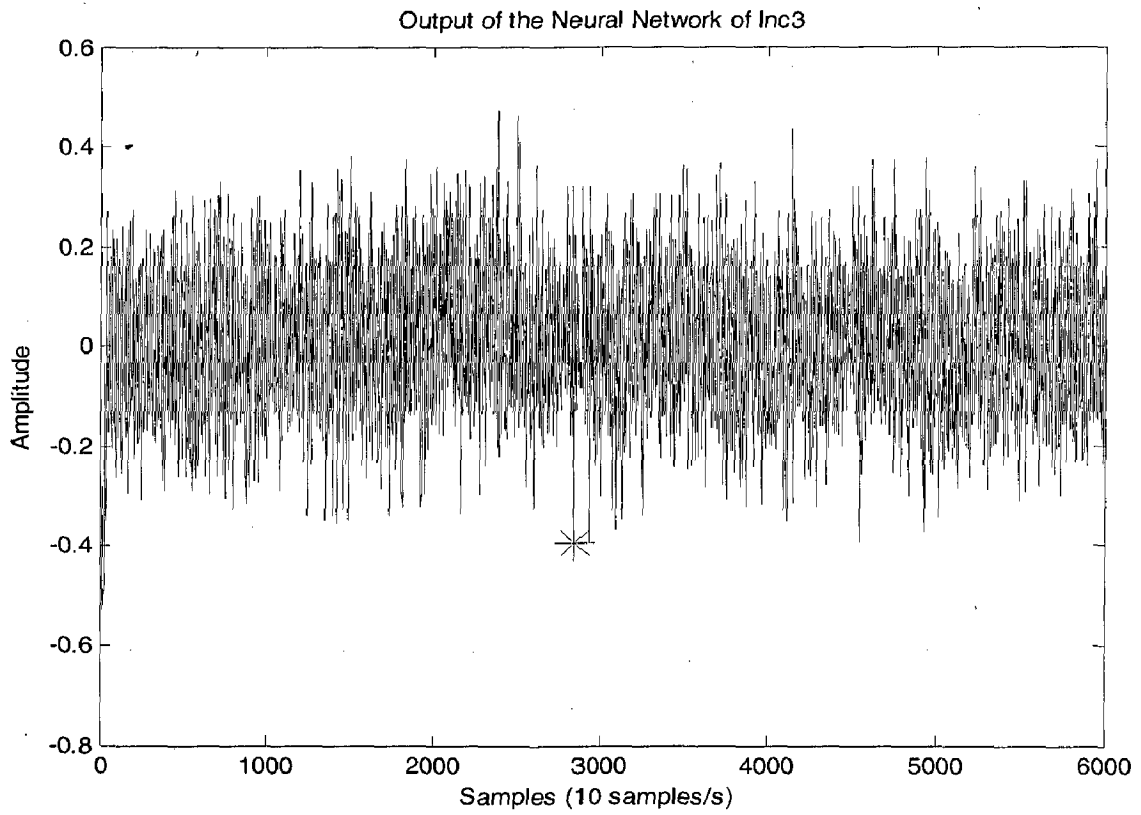
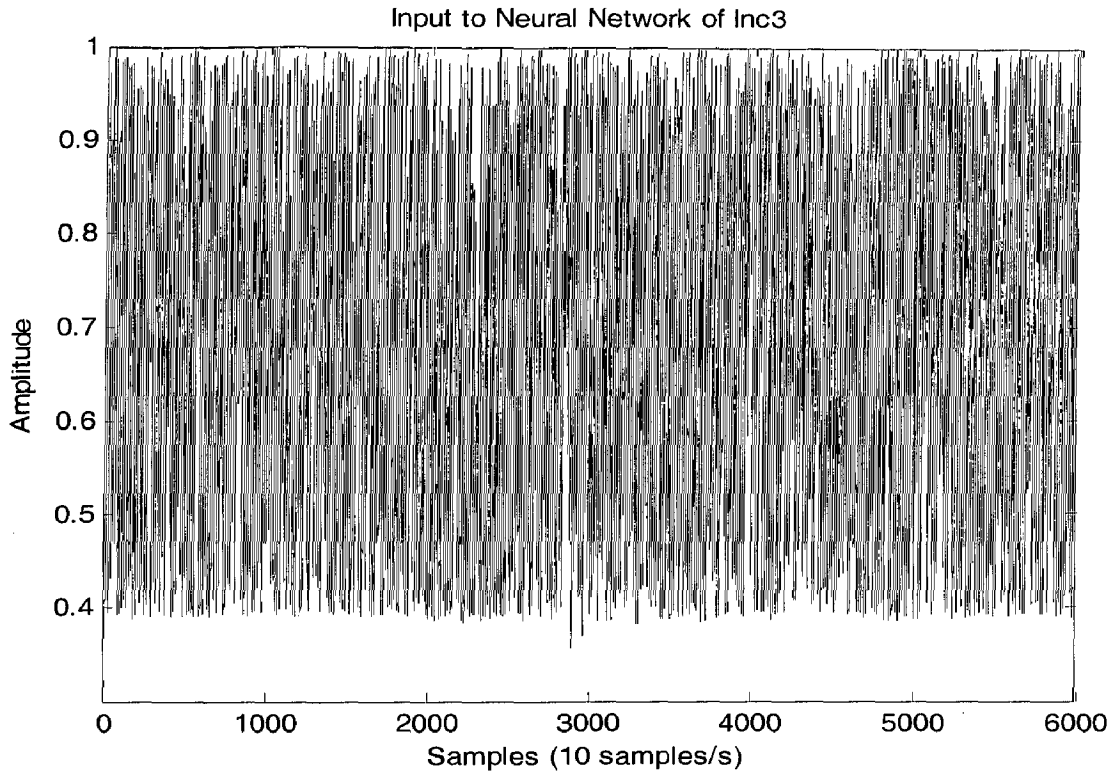


Figure 5.12: Detection of S-phase by Inc3, the maxima peak towards the negative direction is identified as S-wave

5.3 PICKING OF ONSET TIME FOR P-WAVE

In picking the onset time for *P*-wave, the raw data is used without resampling the original frequency. These data have sampling time of 100 samples per second. Since in the above method the sampling time was 10 samples per second, the error was so large in picking the onset time for *P*-wave. In order to get better accuracy and less error, the original data is used. From the results obtained from the detection of seismic phase it was found that the STA \ LTA algorithm produced better results than any other attribute it was found 53% of *P*-phase is detected with in sample difference of 0-10 samples. Therefore I choose this algorithm to find out the onset time for *P*-wave. The algorithm is preprocessed from one horizontal component and input to the artificial neural network. The neural network used here is Backpropagation neural network. The structure of artificial neural network consists of 400 input neurons, one hidden layer with 40 neurons and two neurons in the output layer. The two nodes in the output layer is to flag the result: the output is (1,0) for an arrival and (0,1) for pure noise. By means of trial and error method we finally choose 40 neurons in the hidden layer. Although this solution is considered optimal for the current application, further architecture optimization could undoubtedly be achieved by a more exhaustive search procedure on a more powerful computer. Similarly like *P*-phase detection a small number of recordings are used to train the network, and the remainders are used to test the performance of the trained network. The performance of a trained neural network depends on the training data sets. If we use incorrect or inconsistent data to train the neural network, we cannot expect it to give a correct answer for new data. For training, the input segments include either background signal or the *P* wave with some early background signal. The *P*-wave training segments are chosen to include waves with different characters. The function of this primary picker is to flag as many changes of input as possible and discard those which are *P* arrivals. The segments are arranged so that the predicted onset time of every signal lies at the hundredth sample, for which the network output flags (1, 0). This behavior is imprinted on every training example. Ten different *P*-Waves and Ten background signal training segments used in the study. After training the neural network is ready to pick the waves. To detect or pick an arrival we first calculate the observed STA \ LTA, taking each windowed segment of this, and feed it to the trained neural network. We shift the window

by one sample at a time and feed each segment into the network, storing the output. The procedure is repeated until the end of the seismogram is reached. In general, the output $(O_1(t), O_2(t))$ lies between the ideal for a signal or for a noise (for example, (0.8, 0.2) or (0.4, 0.6)). To provide a single indication of the onset, we use a function $F(t)$ which highlights the difference between the actual output and ideal noise:

$$F(t) = 0.5 [(O_1(t))^2 + (1 - O_2(t))^2]$$

Figure 5.14 shows an example of output of the neural network for one of the data segments in our chosen data set. The peaks in $F(t)$ correspond to abrupt changes in input of STA/LTA, with a small value implying a smooth change. These in turn are dependent on changes of the amplitude and frequency through the weightings in the network. In this curve, there is a large peaks corresponding P wave. Their positions of their maxima occur exactly at the manually chosen (onset times of the P wave). We find that for most cases an arrival corresponds to a sharp change in $F(t)$ so that a threshold may be sufficient to detect the arrival and the P -wave is picked. The below figure shows the picking the onset time for P -wave.

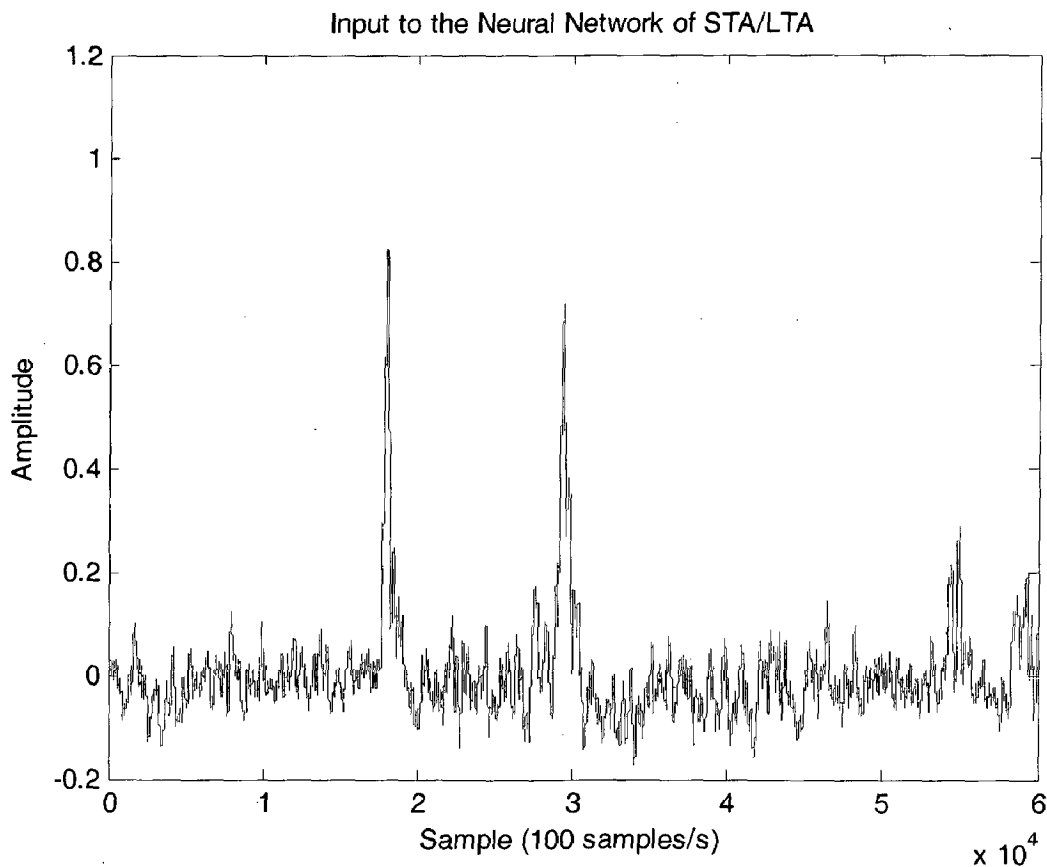


Figure 5.13: Input to the Neural Network for Phase Picking

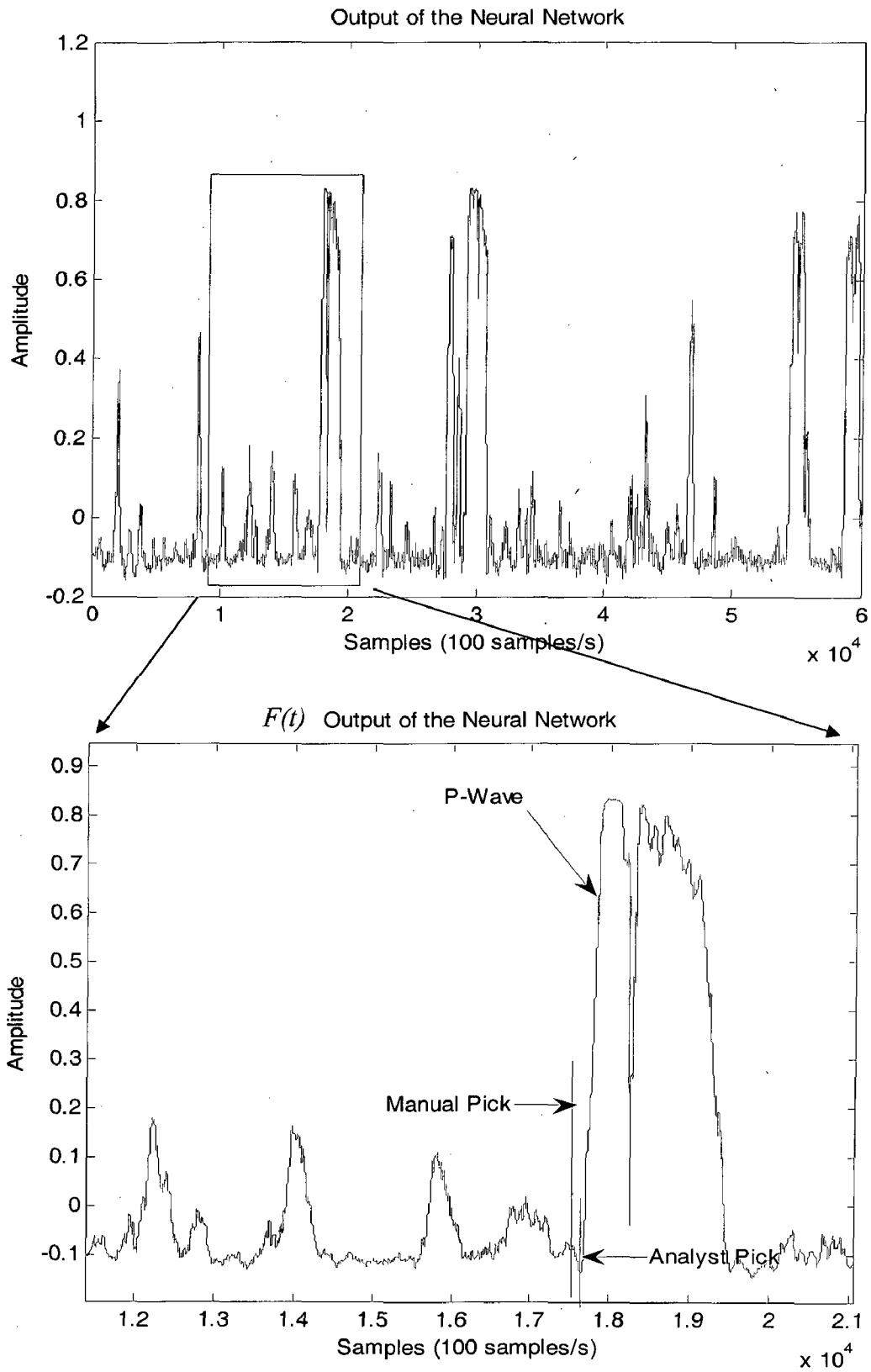


Figure 5.14: Picking of P wave for a 5 min window seismogram. The lower panel shows an enlarged window around the P-arrival.

The results showed that out of 25 events, 17 events are detected and picked in which 53% have an error < 0.1 second and 29% have time error with in 0.1-0.5 second and 6% time error with in 0.5-1 second and 12% have time error with in 1-1.5 second. The ANN also works well low SNR , the below figure shows the time difference between picking by ANN P-phase picker and by manual picking versus the signal-to-noise ratio(SNR).The results of ANN P-phase picker are very satisfactory.

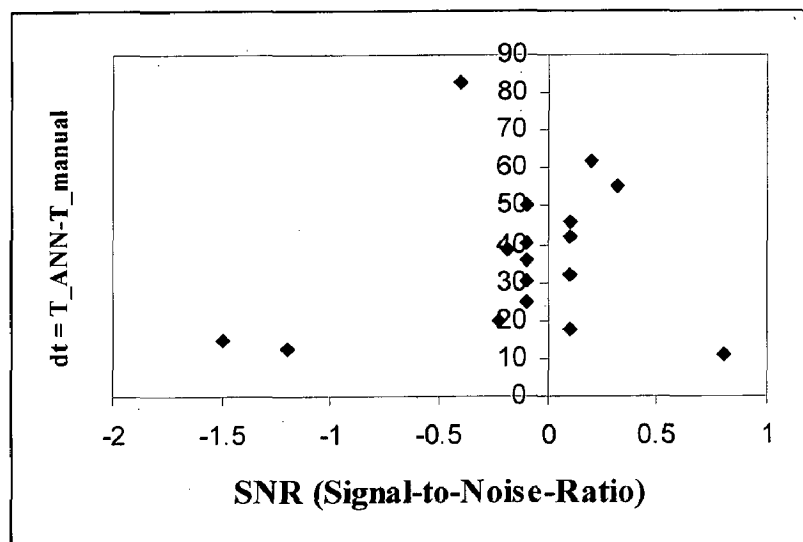


Figure 5.15: The time difference between picking by the ANN P-phase picker and by manual picking versus signal-to-noise-ratio.

5.4 Software details

The algorithms which are developed from the eight attributes have been implemented using Matlab 7. Neural Network tool box has been used for training and testing the neural network. The below figure shows network manager in the neural network tool box which is explained clearly.

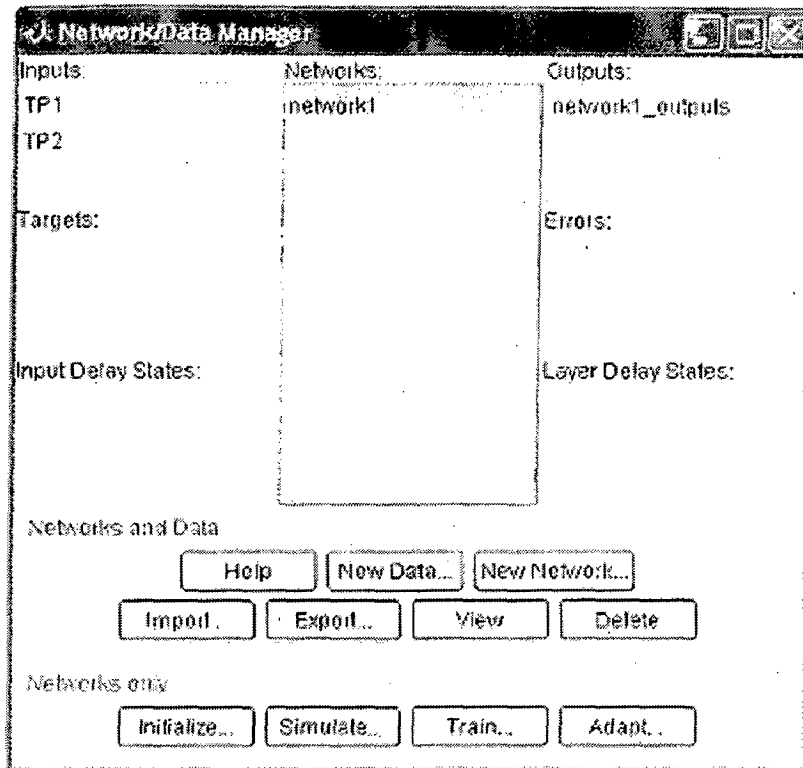


Figure 5.16: Representation of Network Manager in Neural Network tool box

To get started on a new problem does the following:

1. Create new Input and Target data with [NEW DATA] or import data from the workspace or a file with [IMPORT].
2. Create a new Network with [NEW NETWORK] or import a network from the workspace or a file with [IMPORT].
3. Select the network in the Network list and click [TRAIN...] to open the network's window to the training tab.
4. You can use the network's window to train the network, simulate it, or perform other tasks such as initialization or editing weights.

Here are descriptions of each button:

[HELP] – Opens Help window.

[NEW DATA...] - Allows you to type in data.

[NEW NETWORK...] - Allows you to create a new network.

[IMPORT] - Imports data and networks from the workspace or a file.

[EXPORT] - Exports data and networks to the workspace or a file.

[VIEW] - Opens the selected data or network for viewing and editing.

[DELETE] - Removes the selected data or network.

[INITIALIZE] - Opens the selected network for initialization.

[TRAIN] - Opens the selected network for training.

[ADAPT] - Opens the selected network for adaptive training.

Here are descriptions of each list:

Networks - List of networks.

Inputs - Data to present to a network.

Targets - Data defining the desired network outputs.

Outputs - Response of a network to its inputs.

Errors - Difference between targets and outputs.

Input Delays - Input delay states for networks with input delays.

Layer Delays - Layer delay states for networks with layer delays.

The results of seismic phase detection and P-wave picking are explained in the next chapter.

RESULTS

6.1 AUTOMATIC PHASE IDENTIFICATION OF P-WAVE USING ANN

The detection of *P*-phase which has been found out by the four attributes namely STA \ LTA, Rectilinearity, Hmxnm and Incl have been tested on 25 events obtained from Garhwal Kumaon Himalayan region. The parameters i.e. filter band, length of moving window, sampling frequency, and the parameters of Backpropagation neural network, like the structure of the neural network, learning rate and momentum coefficient are all optimized to same extent by trial and error method, applying over the same data. Pie chart below shows the percentage representation of *P*-phase detected by STA \ LTA using ANN. Out of 25 events used for the detection 68% (i.e., 17 events) has been detected *P*-phase correctly, The ANN *P*-phase detector of STA \ LTA shows that 82% have been detected within sample difference of 20 samples from the 17 events.

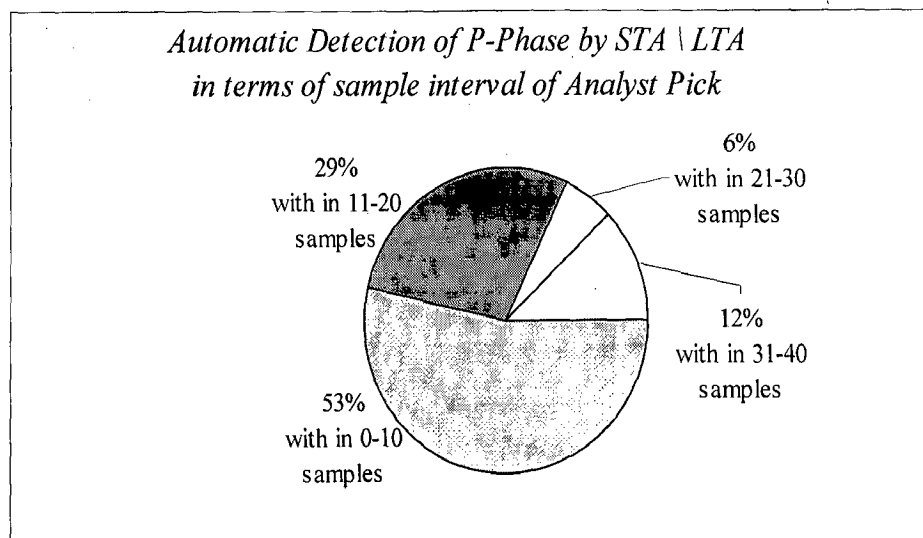


Figure 6.1: Percentage distribution of P-phase detection of ANN by STA \ LTA

The figure 6.2 represents the pie chart of detection by Rectilinearity, it was found that out of 25 events used for testing the ANN 84% (i.e., 21 events) has been correctly

detected the *P*-phase among which 62% of *P*- phase has been with in a sample interval of 0-20 samples.

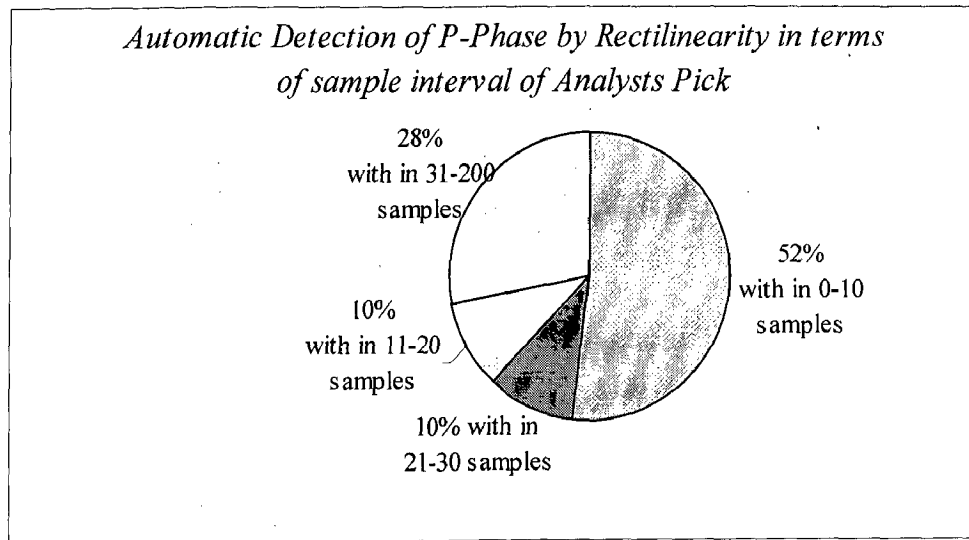


Figure6.2: Percentage distribution of P-phase detection of ANN by Rectilinearity

Out of 25 events which is used for testing the attribute Hmxnm for *P*-phase detection, 19 events has been detected correctly i.e.76% has been detected correctly, the results shows that 29% are with in a sample interval of 0-20 samples. And 37% have large sample interval of 51-150 samples.

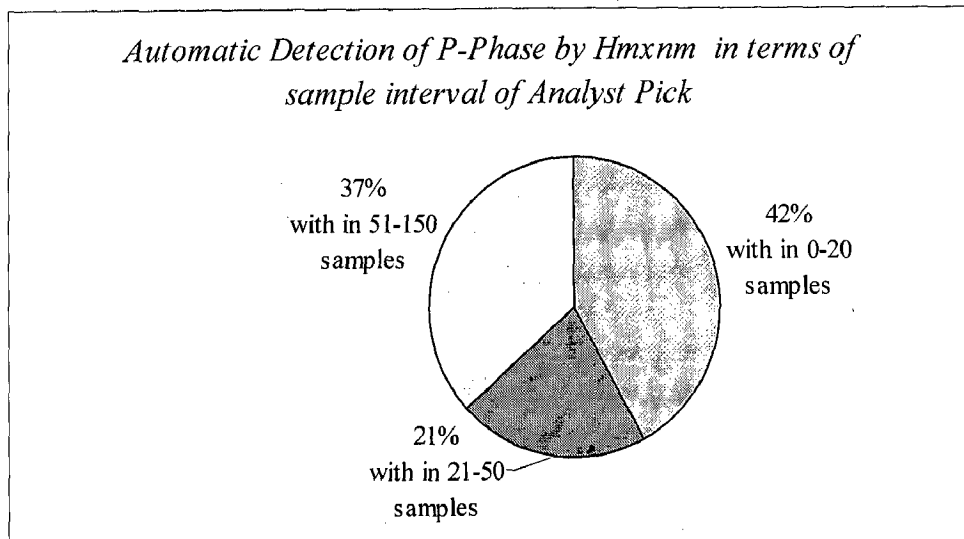


Figure6.3: Percentage distribution of P-phase detection of ANN by Hmxnm

The next pie chart in *figure 6.4* represents the phase detection by *Inc1*. Out of 25 events used for testing 14 events has been detected that is 56% has been detected. Among the 14 events 29% are with in a sample interval of 0-20 samples .This is shown below.

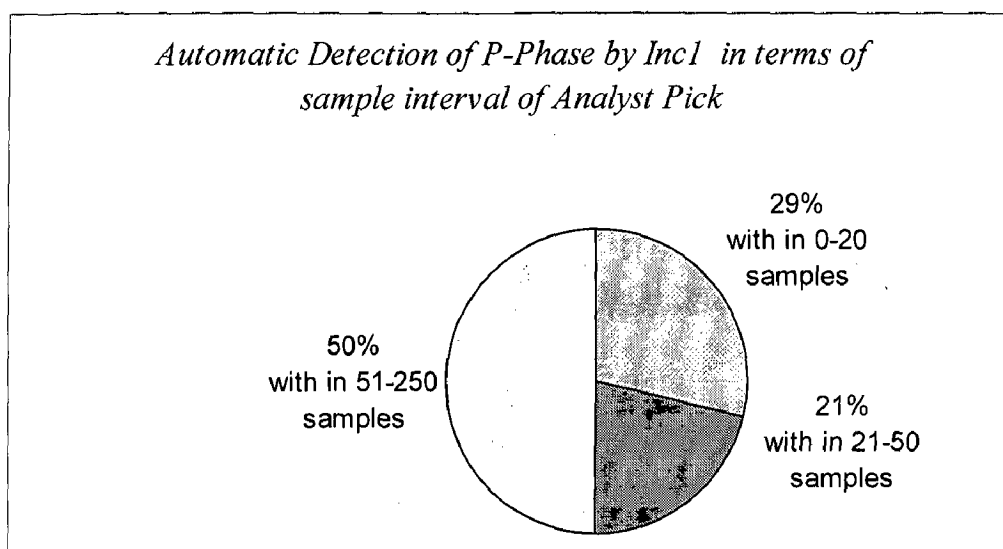


Figure6.4: Percentage distribution of P-phase detection of ANN by Inc1

For each and every event the above attributes were tested, therefore it was found that in each event if any atleast three attribute has detected *P*-phase correctly, then it has declared *P*-phase for that event. In overall out of 25 events 18 events has detected correctly that is 72% has declared *P*-phase correctly.

6.2 AUTOMATIC PHASE IDENTIFICATION OF S-WAVE USING ANN

The detection of *S*-phase which has been found out by the four attributes namely *Rh2t*, *Planarity*, *Hvratp* and *Inc3* have been tested on 25 events obtained from Garhwal Kumaon Himalayan region. The parameters i.e. filter band, length of moving window, sampling frequency, and the parameters of Backpropagation neural network, like the structure of the neural network, learning rate and momentum coefficient are all optimized to some extent by trial and error method, applying over the same data. Pie chart below shows the percentage representation of *S*-phase detected by *Rh2t* using ANN. Out of 25 events used for the detection 68% (i.e., 17 events) has been detected *S*-phase correctly,

The ANN S-phase detector of STA \ LTA shows that 23% have been detected within sample interval of 0-10 samples from the 17 events.

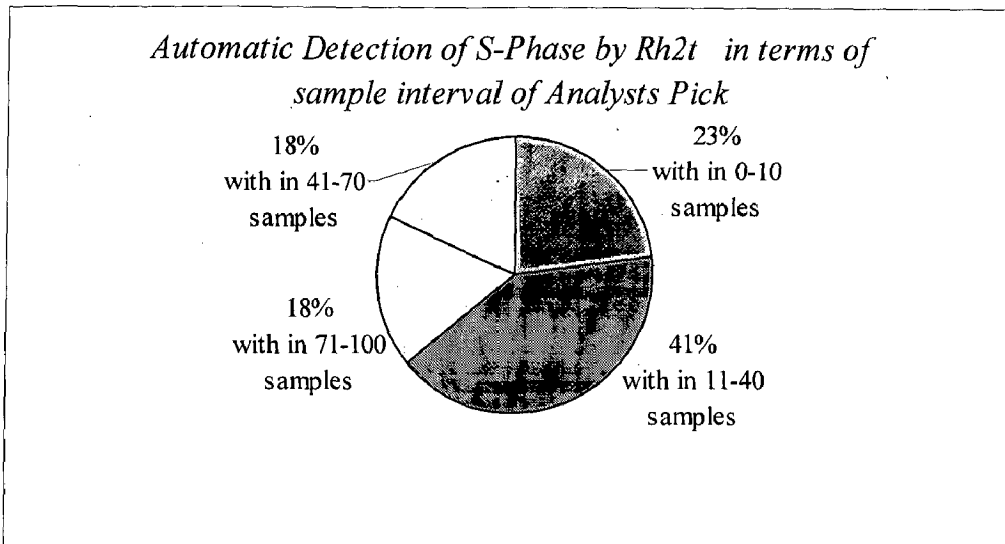


Figure6.5: Percentage distribution of S-phase detection of ANN by Rh2t

The figure 6.6 represents the pie chart of detection by Planarity, it was found that out of 25 events used for testing the ANN 92% (i.e., 23 events) has been correctly detected the S-phase among which 48% of S- phase has been with in a sample interval of 0-30 samples.

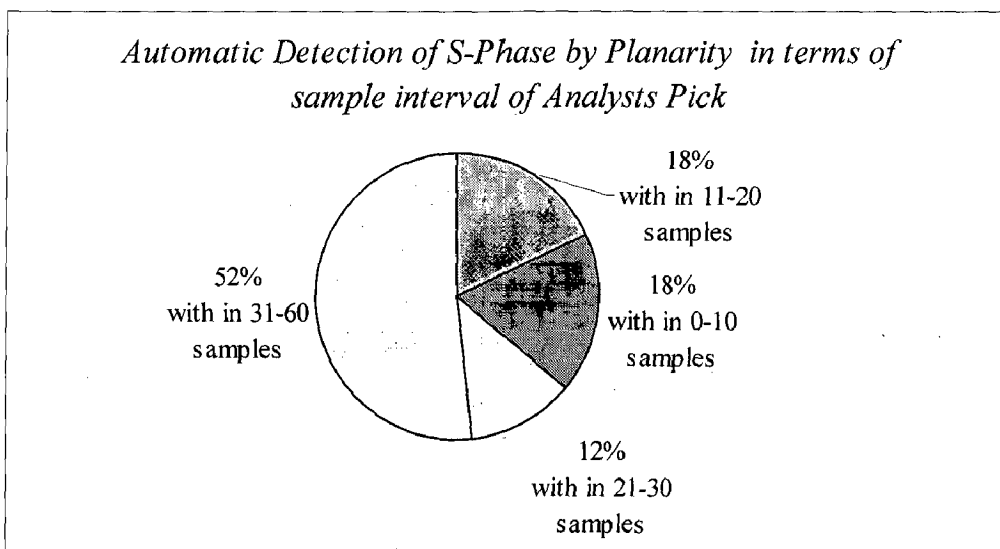


Figure6.6: Percentage distribution of S-phase detection of ANN by Planarity

Out of 25 events which is used for testing the attribute Hvratp for S-phase detection, 20 events has been detected correctly i.e.80% has been detected correctly, the results shows that 55% are with in a sample interval of 0-20 samples. And 10% have large sample interval of 51-120 samples.

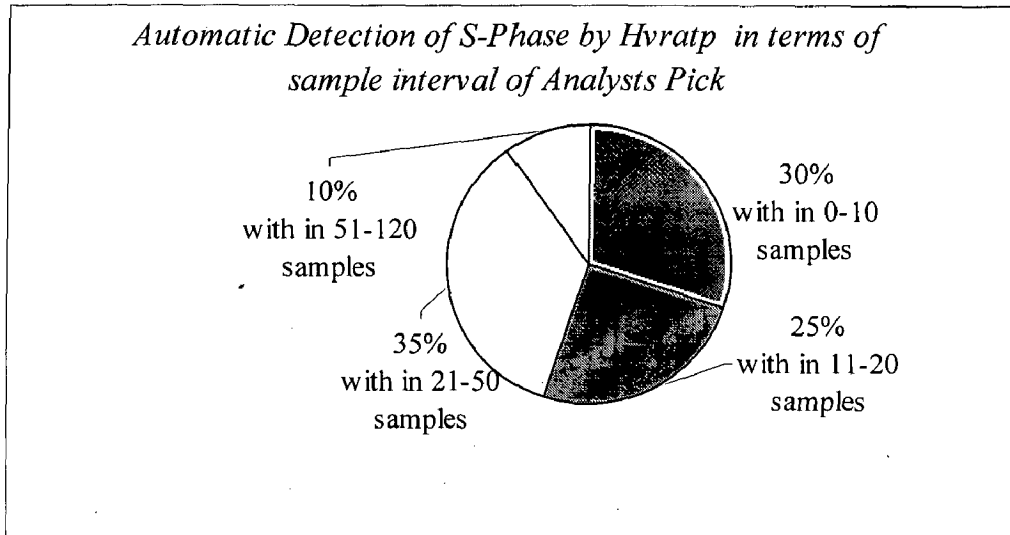


Figure6.7: Percentage distribution of S-phase detection of ANN by Hvratp

The next pie chart in *figure 6.8* represents the S-phase detection by Inc3. Out of 25 events used for testing 15 events has been detected that is 60% has been detected. Among the 15 events 34% are with in a sample interval of 0-20 samples .This is shown below.

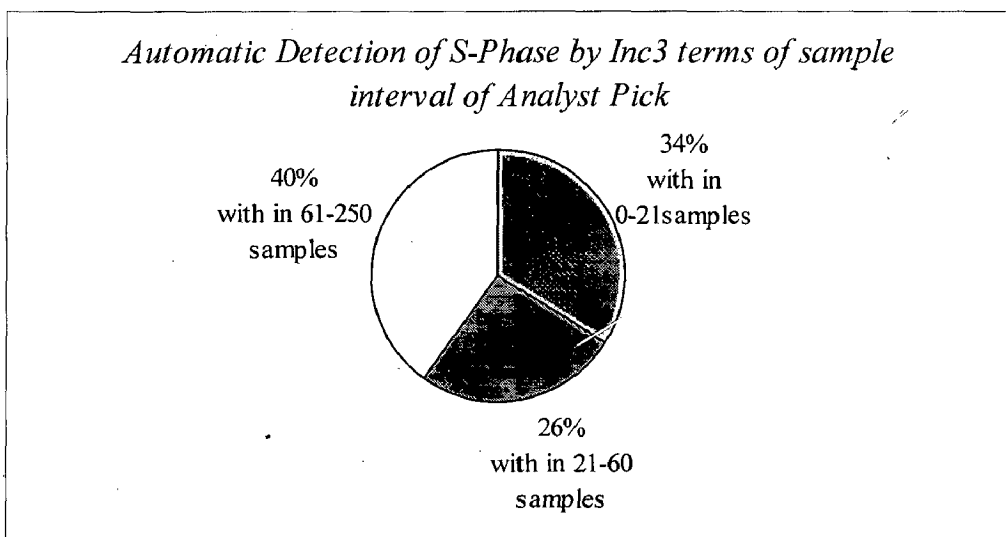


Figure6.8: Percentage distribution of S-phase detection of ANN by Inc3

6.3 AUTOMATIC PHASE PICKING OF P-WAVE USING ANN

The Phase picking of *P*-wave by ANN is done using the conventional algorithm STA \ LTA; they have been tested on 25 events obtained from Garhwal Kumaon Himalayan region. The parameters i.e. filter band, length of moving window, sampling frequency, and the parameters of Backpropagation neural network, like the structure of the neural network, learning rate and momentum coefficient are all optimized to some extent by trial and error method, applying over the same data. Here the structure of neural network is 400-40-2 which implies that 400 as input neurons, 40 neurons in hidden layer and 2 neurons in the output layer. In comparing all the eight attributes STA \ LTA was found to be precise with detecting *P*-phase with less sample interval, and so I choose this algorithm STA \ LTA for picking purpose. Here the resampling is not done, the sampling frequency is used as it is 100 samples / second. In order to get more accuracy in picking we used the original data itself. The pie chart explains clearly about the phase picking of *P*-wave using ANN. It has found that out of 25 events 17 events has been detected and picked correctly. Among the 17 events, 53% are picked correctly with less than 0.1 second, 29% are picked with in 0.1-0.5 second and 18% are picked with in 0.51-1.5 second.

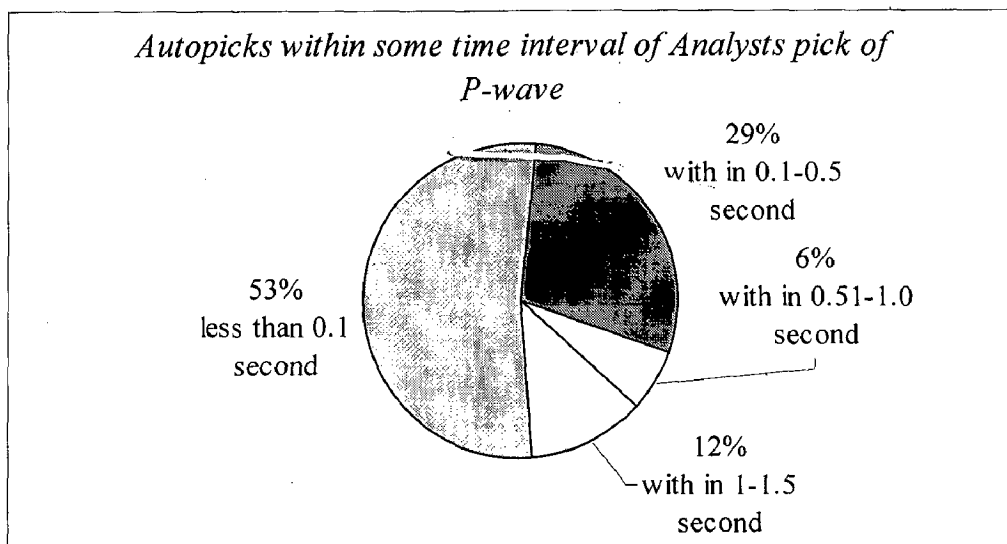


Figure6.9: Percentage distribution of *P*-phase picking of ANN by STA \ LTA

CONCLUSIONS AND DISCUSSION

7.1 Conclusions

The objective of the work described in this article is to implement the artificial neural networks for seismic phase detection and *P*-wave picking which is important for teleseismic, local and regional seismic monitoring. Faster computer, improved algorithms, and better data quality make it possible now to process real-time seismic data quickly, accurately and automatically. To take advantage of recent developments in the digital signal processing, we designed automatic phase identification and picking system that can process several properties of the signal simultaneously using artificial neural networks. The artificial neural network used in this study is multilayer feedforward Backpropagation neural network which is used in many applications. Feature selection from input signal is an important process in application of artificial neural networks. Properly selecting the features of signal will result in better performance of a neural network. In this study, several attributes related to *P*-phase arrival and *S*-phase arrival were analysed in terms of sensitivity and efficiency. Comparing the performance of each feature in discriminating the *P* and *S* wave, eight features were selected as input attributes in which the first four are used for the ANN *P*-Wave and next four attribute for *S*-Wave picker : (1) Ratio between short-term average and long-term average (STA/LTA). (2) Rectilinearity, (3) Ratio of maximum to minimum horizontal amplitude (hmxnm), (4) Long-axis incidence angle of polarization ellipsoid (Inc1) (5) Ratio between horizontal power to total power (Rh2t), (6) Planarity, (7) Ratio of horizontal-to-vertical power (Hvratp), (8) Short-axis incidence angle of polarization ellipsoid (Inc3). The preliminary training and testing were conducted with a set of seismic readings obtained from Garhwal Kumaon Himalayan region. The results of the training and testing show that the artificial neural networks *P*-phase picker and *S*-phase detector can detect seismic phase and pick *P*-wave with satisfactory time accuracy. Overall, an result of 72% correct rate of phase identification has been achieved by both, the trained ANN *P*-phase detector and the trained *S*-phase detector, and 53% of *P*-wave are precisely picked with less than 0.1 sec

onset time error by STA \ LTA algorithm using ANN *P*-phase picker. We believe that this method is a very promising approach in automatic phase identification and onset time estimation.

7.2 Discussions with Future Scope

This work forms part of an ongoing programme of research to develop a fully automatic subsystem for earthquake analysis, for which picking the seismic arrivals is the key procedure. Since in this analysis we develop an approach towards the task of arrival identification of *P* wave using polarization characteristics, the algorithms discussed in this dissertation work well for quasi-real time algorithms. We intend this approach so that more complex wave trains such as an *S* and *L* wave may be picked and analysed. Accuracy of these algorithms can be enhanced further if the noise encroachment at the recording station is reduced. For instance, there is a hut and there are three trees near the recording station of Garhwal Kumaon, which are the sources of man-made noise and air vibration (both are surface waves). Ultimately we hope to integrate other ANN units into a processing flow for record editing and event classification and mechanism determination. The neural network can also be applied to other seismic classification, discrimination and inversion problems and it will emerge as an important seismological tool in the future.

REFERENCES

- Andy Jurkevics, {October 1998}**, "*Polarization Analysis of Three-Component Array data*", Bulletin of the seismological society of America, 78, pp 1725-1743.
- Brijesh Pandya, Pankaj Porwal and Pradipta banerji, {2002}**, "*Application of artificial neural network in development of site-specific response spectra*", 12th SEE (Symposium on earthquake Engineering), IIT Roorkee. pp. 199-207.
- Carr Mathew, Richard Cooper, et al., {August 2001}**, "*The integration of surface seismic and borehole data using artificial neural network clustering methods*", ASEG 15th Geophysical conference and Exhibition, Brisbane.
- Dai Hengchang and Colin Macbeth, {1995}**, "*Automatic picking of seismic arrivals in local earthquake data using an artificial neural network*", Geophys.J.Int. 120, pp. 758-774.
- Del Edoardo Pezzo, Anna Esposito, et al., {2003}**, "*Discrimination of earthquakes and underwater explosions using neural networks*", Bulletin of the seismological society of America, 93, pp 215-223
- Estela Minaya, Percy Aliaga, et al.,** "*Neural network and wavelets model in seismic location for the central Andes of Bolivia*", Army space defense and missile command.25th Seismic Research Review – Nuclear Explosion Monitoring: Building the Knowledge Base.
- Feng Xia-ting, M. Seto and K. Katsuyama, {1997}**, "*Neural dynamic modeling on earthquake magnitudes magnitude series*", Journal of Geophysics 128, pp. 547-556.
- Fu Limin, {2003}**, "*Neural Networks in Computer Intelligence*", Tata McGraw-Hill Edition.
- Hagen, Demuth & Beale {2002}**, "*Neural Network Design*", Thomason Learning Haykin Simon "*Neural Networks*" Second Edition, Pearson Education.
- McCormack Michael D., David E. Zaucha, and Dennis W. Dushek, {January 1993}**, "*First- break refraction event picking and seismic data trace editing using neural networks*", Bulletin of the seismological society of America 58, pp. 67-78.
- Pathak, J D.K. Paul and P.N. Godbole, {2002}**, "*ANN based model for simulation of design earthquake in the Himalayan region*", 12th SEE (Symposium on earthquake Engineering), IIT Roorkee. pp. 187-198.
- Pathak, J D.K. Paul and P.N. Godbole, {December 1998}**, "*Application of ANN in simulation of earthquake: A critical review*", 11th symposium on earthquake Engineering, IIT Roorkee. pp. 271-280.
- Robert Essenreiter, Martin Karrenbach, & Sven Treitel {July 1998}**, "*Multiple Reflection Attenuation in Seismic Data Using Backpropagation*". IEEE Ttransactions on Signal Processing, Vol. 46, No. 7.

Sharma Mukat L., and Manoj K. Arora, {2005}, “*Prediction of seismicity cycles in the Himalayas using artificial neural network*”, ACTA Geophysica polonica, 53, pp. 299-309.

Stein S & Wysessern {2003}, “*An Introduction to Earthquakes and Earth’s Structures*” – Blackwell Publishing.

Wang Jin {2002}, “*Adaptive training of neural networks for automatic seismic phase identification*”, Pure applied geophysics. 159, pp 1021-1041

Wang Jin and Ta-liang Teng, {1997}, “*Identification and picking of S phase using an artificial neural network*”, Bulletin of the seismological society of America, 87, pp. 1140-1149.

Wang Jin and Ta-Liang Teng, {February 1995}, “*Artificial neural network-based seismic detector*”, 85, pp. 308-319.

www.deprem.cs.itu.edu.tr/neural, “*Neural network based electric field pattern recognition for earthquake prediction*”

www.quake.gatech.edu, “*Artificial neural network applications in earthquake prediction*”

Yair Shimshoni and Nathan Intrator, {May 1998} “*Automatic seismic classification by integrating ensembles of ANN*”, IEEE, Transaction in signal processing, Vol 46, No.7.

Zhao Y and Kiyoshi Takano {1999}, “*An artificial neural network approach for broadband seismic phase picking*”, Bulletin of the seismological society of America. 89, pp 670-680.

Zurada Jacek M, {2004}, “*Introduction to Artificial Neural Systems*”, Jaico Publishing House.