

**ANALYSIS OF LOCAL EARTHQUAKE DATA  
USING  
ARTIFICIAL NEURAL NETWORKS**

by  
Hengchang Dai

*B.Sc.(Hons) Geophysics, 1982  
The University of Sciences and Technology of China*

*M.Sc. Geophysics, 1984  
Institute of Geophysics, State Seismological Bureau, China*

Thesis submitted for the degree of Doctor of Philosophy  
Department of Geology and Geophysics  
University of Edinburgh

December 1995



TO

My father, *Jianxun Dai* (1926-1981) and my mother, *Xiulan Zhang*  
and  
my wife *Ying Zhou* and my son, *Zhoudi Dai*



## ABSTRACT

The aim of this thesis is to investigate the possibility of using artificial neural networks to develop automatic processing techniques for picking and identifying seismic arrivals. These two key procedures of an earthquake analysis system are extremely labour intensive, and any automation would allow further processing of larger datasets. Two approaches based on the back-propagation neural network (BPNN) are developed to pick and identify seismic arrivals for the dataset which includes 762 three component (3-C) recordings from stations DP and AY of a local earthquake network in Turkey.

A BPNN approach was developed to pick arrivals automatically from 3-C recordings and single component (1-C) recordings. In Chapter 4, this approach is applied to the vector modulus (amplitude) of 3-C recordings. A BPNN trained by *P*-arrivals with high signal-to-noise-ratio (SNR) and background noise from station DP can extend its ability to picking *S*-arrivals and picking arrivals from other stations and from seismograms with low SNRs. It successfully detects 94.3% of the *P*-arrivals and 86.4% of the *S*-arrivals, compared with manual picks. The onset times of 74.5% of the *P*-arrivals and 63.2% of the *S*-arrivals are successfully picked with an error of 10 ms (one sample increment).

In Chapter 5, this method is adapted to pick seismic arrivals from the absolute value (amplitude) of 1-C recordings. A BPNN trained by *P*-arrivals and background noise from the vertical component of station DP can extend its ability to the other two horizontal components and other stations. The picking rates are 93.1%, 89.4%, and 83.1% for *P*-arrivals, and 75.0%, 90.9%, and 87.2% for *S*-arrivals from the Vertical, E-W, and N-S components respectively. With an error of 10 ms, 66.2%, 59.2% and 63.3% of the *P*-arrivals, and 52.7%, 61.2% and 57.7% of the *S*-arrivals are picked from Vertical, E-W and N-S components respectively.

In Chapter 6, another BPNN approach is developed to identify *P*- and *S*-arrival types from local earthquake data, using the modified degree of polarization (DOP) of 3-C recordings. Only the arrival segments are input in this BPNN approach for processing. Compared with manual analysis, a BPNN trained with nine groups of training *P*-arrival, *S*-arrival, and noise burst segments of DOP from station DP can correctly identify 82.3% of the *P*-arrivals and 62.6% of the *S*-arrivals from station DP. Another BPNN trained with five groups of training datasets from station AY can correctly identify 76.6% of the *P*-arrivals and 60.5% of *S*-arrivals from station AY.

In order to understand how the BPNN works, a weight map is designed in this thesis to show the weight patterns of a trained BPNN. This new finding would be applied to any BPNN application, enabling illumination of the "block-box" approach of BPNN analysis. Applying this map to three trained BPNNs shows that it is a useful tool to investigate the interior and performance of BPNNs. For example, the weight map of a BPNN applying to pick arrivals from 3-C recordings shows that its weight pattern is divided into two portions which have different functions in picking.

Some factors which may affect the BPNN performance are examined. The results show that the training parameters strongly affect the training convergence but the BPNN performance is not affected too much. A relationship between the training parameters and the training convergence is obtained. The results also show that a three-layer BPNN is sufficient for the applications of picking and identifying seismic arrivals. Each approach has its own optimum number of input nodes. The performance of a trained BPNN is highly dependent on the training dataset.

This work shows that the BPNN has great potential to analyse earthquakes automatically. A significant feature of using BPNNs is their adaptiveness. The same programs are used to pick arrivals from 3-C and 1-C recordings, and to identify the arrival types with only minor changes of the BPNN structure and the input/output. Once the generic routines of BPNN are developed, it is easy to apply them to resolving a particular problem without requiring additional programs to construct special variables and parameters with complicated mathematics. The only necessary step is then to select suitable training examples for the new application. The

performance of a trained BPNN is objective and consistent and can be easily improved by adding or adjusting the training dataset. For the foreseeable future, this choice of dataset remains a subjective element in the procedure. Nevertheless, this thesis demonstrates that a relatively simple BPNN can correctly pick and identify the majority *P*- and *S*-arrivals for local earthquake data. This method could in principle easily be adapted to regional or teleseismic data, or a range of other applications in seismology.

## **ACKNOWLEDGEMENTS**

I would like to express my sincere gratitude to my supervisor, Dr Colin MacBeth. None of this research would have been possible without his enthusiasm, support and guidance. I am greatly indebted to Colin for his time, effort and patience he showed when explaining the artificial neural network to me and improving my English. I am lucky to have him as my supervisor. I am grateful to Dr Ian Main, my University supervisor, for his invaluable suggestions and helpful opinions on many aspects of my thesis.

I must thank Dr Chris Browitt, Assistant director of British Geological Survey (BGS) and the former manager of Global Seismological Research Group (GSRG), BGS, and Professor Sturt Crampin, the former head of Edinburgh Anisotropy Project (EAP), BGS, accepted me as a scientific visitor, later as a PhD student in BGS, and allowed me to use all the necessary facilities.

Special thanks must go to the number of staff at GSRG, BGS, who have helped me throughout this work, particularly, Dr David Booth (manager of GSRG) and Mr John Lovell for supplying the TDP3 earthquake data; Dr Enru Liu and Dr Xiangyang Li for many useful discussions; Dr Phil Wild for various computing aspects; Mr Terry Turbitt (once the acting manager of GSRG) and Mrs Angela Muir and Ms Rose Aitken (both GSRG secretaries) for their help.

I am particularly indebted to Dr Colin MacBeth, Dr Ian Main, Mr John Lovell, and Dr Enru Liu for reading the whole manuscript of this thesis and Dr David Booth, Mr Terry Turbitt, Dr Xiangyang Li, and Dr Jim Bolton for reading the part of manuscript of this thesis. Their comments and suggestions greatly improved the context and English of this thesis. Any mistake remains my responsibility.

The postgraduate students in BGS gave me constant advice, suggestion, criticism and support: Aphrodite Karnasspoulou, Dr Gordon Holmes, Brian Baptie, Dr

Stephen Horne, Colin Slater, Dr Yun Liu, Helen Rowlands, and Dr Xinwu Zeng. Particularly, Dr Yun Liu gave me great help in manual picking seismic arrivals.

I must thank Natural Environment Research Council (NERC) for awarding me a studentship in 1992. My studentship was founded by GSRG, BGS, which also supported me to attend several national and international meetings. All works in this thesis were carried out in BGS.

I also thank the Chinese State Education Commission, the Institute of Geophysics of State Seismological Bureau, China, and the British Council for sending me aboard in 1991 under the Sino-British Friendship Scholarship Scheme (SBFSS) and supporting me in my first year in Edinburgh

Last, but not least, I thank my wife, Ying for her continuous support my work. I apologize to Ying and my son Zhoudi for many absent evenings. I also thank my mother and parents-in-law for supporting my continuous education and encouraging me to study abroad.

# CONTENTS

Abstract . . . . .	i
Acknowledgements . . . . .	iv
Contents . . . . .	vi
 Chapter 1: Review of Automated analysis of seismic records . . . . .	 1
1.1 Introduction . . . . .	1
1.2 Fundamental concepts of seismic analysis . . . . .	2
1.2.1 Seismic recordings . . . . .	2
1.2.2 Concepts of seismic analysis . . . . .	2
1.2.3 Basic definitions . . . . .	3
1.3 Automatic seismic analysis system . . . . .	5
1.3.1 Procedures of seismic analysis system . . . . .	5
1.3.2 Automated seismic analysis system . . . . .	6
1.3.3 Key problem in building ASAS . . . . .	7
1.4 Outline of this thesis . . . . .	8
 Chapter 2: Introduction to artificial neural networks . . . . .	 10
2.1 Introduction . . . . .	10
2.2 Theory of artificial neural networks . . . . .	11
2.2.1 What is an artificial neural network? . . . . .	11
2.2.2 Biological neural networks . . . . .	12
2.2.3 Artificial neural networks . . . . .	12
2.2.4 How do ANNs work? . . . . .	13
2.2.5 Classifications of ANNs . . . . .	14
2.3 Back-propagation neural network . . . . .	15
2.3.1 Theory of back-propagation neural network . . . . .	15
2.3.1.1 Basic concepts of BPNN and its operation . . . . .	15

2.3.1.2 Learning algorithm - <i>Generalized Delta Rule</i> . . .	17
2.3.2 Training procedure of BPNN . . . . .	18
2.3.3 Computer programming . . . . .	20
2.3.4 Selecting BPNN structure and training parameters . . . . .	20
2.4 Discussions and Summery . . . . .	21
2.4.1 Advantages and disadvantages of BPNN . . . . .	22
2.4.2 Applications of BPNN . . . . .	23
2.4.3 Principles of utilizing BPNN . . . . .	24
2.4.4 Applications of BPNN in this thesis . . . . .	24
Chapter 3: Earthquake Data and Characteristics of Seismic Records . . . . .	25
3.1 Introduction . . . . .	25
3.2 Local earthquake data . . . . .	26
3.2.1 TDP3 seismic network . . . . .	26
3.2.2 Data acquired from TDP3 . . . . .	27
3.3 Characteristics of seismic recordings . . . . .	28
3.3.1 Characteristics from single-component recordings . . . . .	28
3.3.2 Characteristics from three-component recordings . . . . .	31
3.3.3 Covariance matrix analysis . . . . .	33
3.4 Summary . . . . .	35
Chapter 4: Arrival picking from three-component recordings using BPNN . . .	37
4.1 Introduction . . . . .	37
4.2 Brief history of automatic arrival picking . . . . .	38
4.3 Approach for arrival detection and picking . . . . .	40
4.3.1 Input characteristics of data . . . . .	41
4.3.2 BPNN structure . . . . .	42
4.3.3 Training procedure . . . . .	43
4.3.3.1 Selecting training dataset . . . . .	43
4.3.3.2 Selecting training parameters and initial weights .	44

4.3.4	Output function of trained BPNN . . . . .	48
4.3.5	Arrival detection . . . . .	49
4.3.6	Picking of arrival onset times . . . . .	49
4.3.7	Post-processing for discarding noise bursts and spikes . . .	50
4.3.7.1	Discarding small noise bursts . . . . .	50
4.3.7.2	Discarding spikes . . . . .	51
4.4	Performance of the BPNN . . . . .	52
4.4.1	Performance of the trained BPNN . . . . .	52
4.4.2	Improving performance by adding new training dataset . .	52
4.4.3	Sensitivity to segments length . . . . .	53
4.4.4	Sensitivity to signal-to-noise-ratio . . . . .	54
4.5	Application to complete dataset . . . . .	55
4.5.1	Data characteristics . . . . .	55
4.5.2	Performance . . . . .	56
4.6	Discussions and summary . . . . .	57
4.6.1	Weigh pattern analysis . . . . .	57
4.6.2	Effect of post-processing . . . . .	60
4.6.3	Comparison with other picking algorithms . . . . .	60
4.6.4	Summary . . . . .	61
4.6.5	Further work . . . . .	63
Chapter 5:	Arrival picking from single-component recordings using BPNN . . .	64
5.1	Introduction . . . . .	64
5.2	Approach of picking arrivals from 1-C recordings . . . . .	65
5.2.1	Characteristics of 1-C recordings . . . . .	65
5.2.2	BPNN structure . . . . .	66
5.2.3	Rules of picking arrivals . . . . .	66
5.2.4	Post-processing procedures . . . . .	67
5.3	Optimizing BPNN structure and training dataset . . . . .	67
5.3.1	Optimizing BPNN structure . . . . .	68



5.3.2 Optimizing training dataset . . . . .	68
5.3.3 Effect of learning parameters . . . . .	69
5.4 Testing on complete dataset . . . . .	69
5.4.1 Testing on vertical component recordings . . . . .	70
5.4.2 Testing on E-W component recordings . . . . .	71
5.4.3 Testing on N-S component recordings . . . . .	72
5.4.4 Overall performance from three 1-C recordings . . . . .	73
5.5 Discussions and summary . . . . .	75
5.5.1 Comparison with 3-C method . . . . .	75
5.5.2 Weight pattern analysis . . . . .	76
5.5.3 Summary . . . . .	77
Chapter 6: Arrival type identification using BPNN . . . . .	79
6.1 Introduction . . . . .	79
6.2 Degree of polarization . . . . .	80
6.3 Approach of identifying arrival types using BPNN . . . . .	81
6.3.1 Input characteristic: modified DOP, $MF(t)$ . . . . .	82
6.3.2 BPNN structure . . . . .	82
6.3.3 Training procedure . . . . .	83
6.3.4 Identifying arrival types . . . . .	84
6.4 Performances of the trained BPNN . . . . .	85
6.4.1 BPNN's performance for data from station DP . . . . .	85
6.4.2 BPNN's performance for data from station AY . . . . .	86
6.4.3 Effect of training dataset . . . . .	87
6.4.4 Effect of the input nodes . . . . .	88
6.5 Discussions and summary . . . . .	89
6.5.1 Weight pattern analysis . . . . .	89
6.5.2 Summary . . . . .	90

Chapter 7: Conclusions and future work . . . . . 92

    7.1 Introduction . . . . . 92

    7.2 Main results of this thesis . . . . . 92

        7.2.1 Arrival picking . . . . . 92

        7.2.2 Arrival identification . . . . . 94

        7.2.3 Weight pattern analysis . . . . . 95

        7.2.4 Factors affecting the BPNN performance . . . . . 95

        7.2.5 Summary . . . . . 97

    7.3 Future work . . . . . 98

        7.3.1 Dealing with regional and teleseismic earthquake data . . . 98

        7.3.2 Alternative approach of identifying arrival types . . . . . 98

        7.3.3 Applying ANNs to other earthquake analysis procedures . . 99

        7.3.4 Understanding more about ANNs . . . . . 99

    7.4 Conclusions . . . . . 100

Appendix A: Generalized Delta Rule . . . . . 101

References . . . . . 105

List of Publications . . . . . 115

**Papers attached to Back Cover**

Hengchang Dai and Colin Macbeth, 1995, Arrival type identification in local earthquake data using artificial neural network, *Proceedings of the 2nd Workshop on Application of Artificial Intelligence Techniques in Seismology and Engineering Seismology*", Walferdange, Luxembourg, 4-6 October 1995, *in press*.

Hengchang Dai and Colin MacBeth, 1995, Automatic picking of seismic arrivals in local earthquake data using an artificial neural network, *Geophysical Journal International*, **120**, 758-774.

Hengchang Dai and Colin MacBeth, 1994, Split shear-wave analysis using an artificial neural network, *First Break*, **12**, 605-613.

## CHAPTER 1:

### REVIEW OF AUTOMATIC SEISMIC ANALYSIS

#### 1.1 INTRODUCTION

One important aim of seismic analysis is to extract information on earthquake events from seismograms as quickly as possible after they occur. Such information includes the event position, origin time and magnitude which can be accessed for further scientific analysis or public inquiries. The traditional method is that a trained analyst visually checks the seismograms on which the earthquake signal is recorded and picks out events from a background of noisy signals according to his individual experience. This task is time-consuming and subjective. For decades, the need to improve our capabilities to detect, locate, and identify earthquake events has been a major motivation for the development of new technology and theory in seismology. In recent years, as many modern technology have been applied to seismology, large numbers of digital seismic data have been recorded. Manually analysing them is expensive and time-consuming. It becomes necessary to develop an alternative method that can automatically analyse a large volume of digital seismic recordings. Since Allen (1978) popularized his short-term and long-term average ratio (STA/LTA) to pick *P*-arrival automatically, many automatic or semiautomatic methods have been developed. However, they do tend to be data specific and are not generally applicable. The ultimate goal of global automation is still far from being achieved, and seismogram interpretation still forms a bottleneck in the routine work of many observatories.

Recently, the *artificial neural network*, one of a category of the artificial intelligence methods, has been introduced to a diversity of geophysical and geological problems. It provides a natural computational alternative to manual seismic analysis as it has proven useful at handling complicated pattern recognition problems in other applications. In this thesis, I shall investigate the application of artificial neural

networks to automatic analysis of digital seismic data.

## 1.2 FUNDAMENTAL CONCEPTS OF SEISMIC ANALYSIS

### 1.2.1 Seismic recordings

A seismic observation system consists of two basic components: the seismometer and the recorder. The seismometer outputs mechanical or electrical signals which represent graphically the ground motion (displacement, velocity or acceleration) at a point caused by the passing of seismic waves. According to the orientation of a seismometer, it outputs signals which coincide with the vertical or horizontal motions. A complete record of ground motion can be obtained by combining three orthogonal seismometers (one vertical and two horizontal components, usually along the Vertical, North-South and East-West directions). The recorder then records the output signals from seismometers. The earliest recording method was a moving stylus which scratched smoked paper around a revolving and translating drum to record the mechanical signals (helical recorder). Later, other analogue devices, such as pen and ink recorders and a heated stylus marking special recording paper, were developed to record electrical signals. Alternatively, many installations used a moving light spot on photographically sensitized paper or film to record either mechanical or electrical signals. Recently, many earthquake data are recorded on magnetic tape in analogue form and then can be displayed on suitable analogue playback facilities using which the seismograms are plotted either onto paper by pen and ink recorders or onto other analogue visual devices. The latest development in seismological recording is to record the digital signal on mass storage media: tape or disk, and to playback the recordings on computer screens.

### 1.2.2 Concepts of seismic analysis

The recorded seismic signals, which are called *seismograms*, are contaminated by various kinds of noise, such as artificial vibrations, traffic noise, background noise. Figure 1.1 shows a typical seismogram of a local earthquake. These observed datasets

Station: DP  
Date: 1984-05-06  
Start-time: 12h16m08s  
Scale: 564

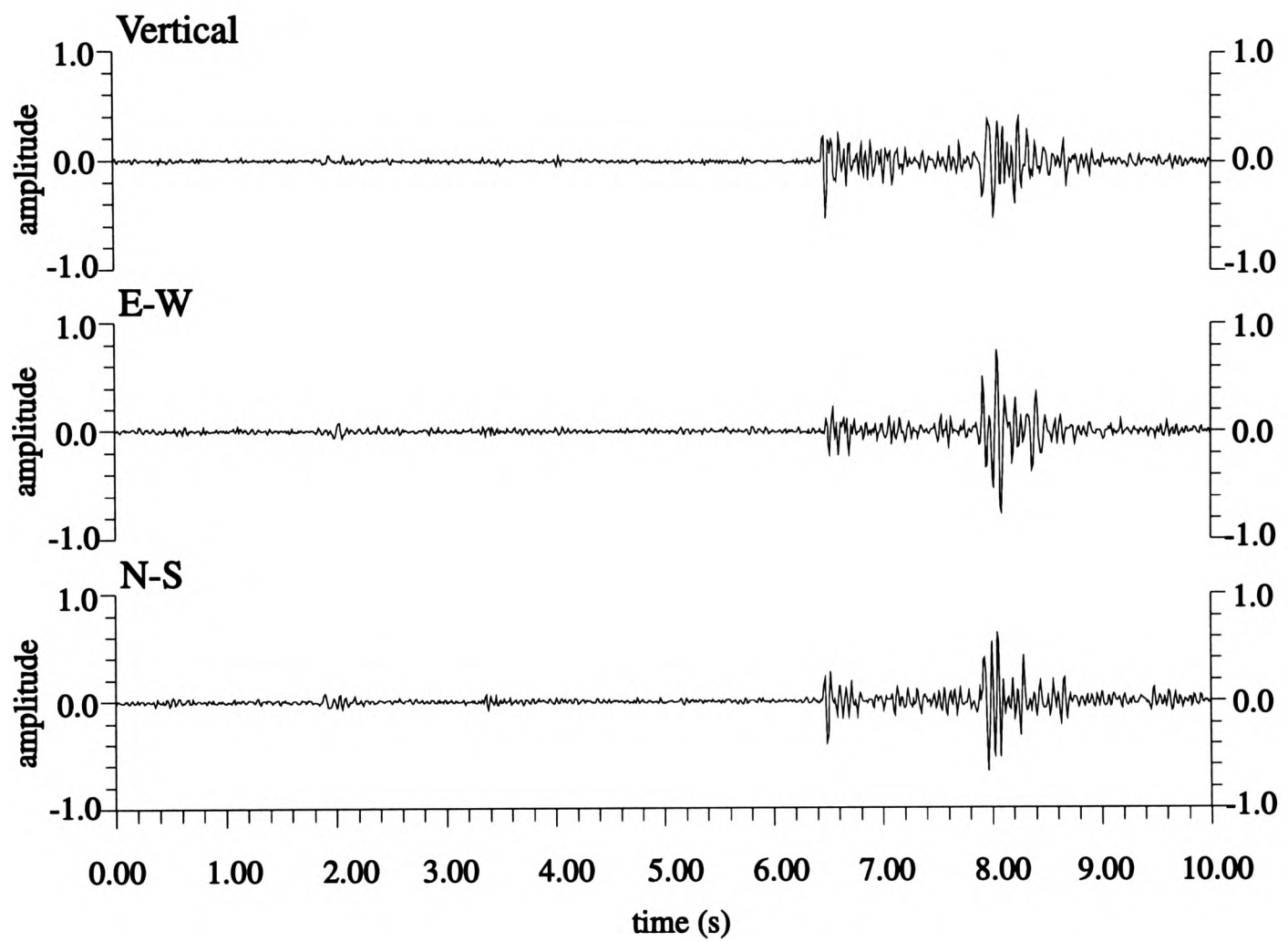


Figure 1.1. A typical seismogram of a local earthquake recorded on station DP, TDP3 local seismic network.

have been traditionally handled by empirical methods based on the expertise of the human operator to identify real seismic signals from the various source of noise.

In the early days of seismic analysis, seismic signals were recorded directly on paper. Once an event has occurred, analysts must visually check the seismograms which contains the event, first, to seek the possible seismic arrivals, for example, the primary ( $P$ ) and secondary ( $S$ ) wave arrivals, and then to pick their onset times. The results are used in subsequent analysis procedures, such as event location, event identification, source mechanism analysis, and spectral analysis. At this picking stage, the experience of the analyst is essential on the final interpretation.

As recording technology developed, seismic events were recorded directly on computers in digital format. The analysts can interactively view the seismograms on computer screens to recognize the seismic arrivals from the background signals and to pick their onset times. This is similar to picking arrivals on paper, but with higher precision, convenience, and a shorter time involved. However, this procedure still mainly depends on the analyst's experience. The picking results can be fed into the subsequent analysis procedures as the basic parameters from which other parameters are determined. An experienced analyst will usually be able to decide quickly the epicentral distance and size of a particular earthquake and its focal-depth.

### 1.2.3 Basic definitions

In order to present the current objective, it is necessary to distinguish between the following definitions:

- (a) *an arrival* - a seismic motion defined by a wavelet whose characteristics resemble the seismic waves, such as  $P$  or  $S$  in our case;
- (b) *arrival detection* - specification of an arrival time close to which an arrival may be bracketed within a predefined time window;
- (c) *arrival picking* - reliable and accurate estimation of the onset time of a definite seismic arrival;
- (d) *arrival identification* - classification of individual arrivals into categories

relating to their amplitude, their polarization and the nature of their propagation. For example, *P*-waves have near-linear polarization which direction is parallel its propagation direction, but *S*-waves have more complex polarization which direction is perpendicular to its propagation direction;

- (e) *a false alarm* - a spurious signal of non-seismic character or a disturbance with sufficient difference in statistical character from an event so that it cannot be readily utilized in defining the earth's structure, for examples, electrical spikes, and continuous traffic noise;
- (f) *an event* - a transient seismic signal generated by a phenomenon such as an earthquake, quarry blast, sonic boom, or underground explosion, which is recorded as a time sequence. An event possesses a fine structure given by a hierarchy of arrivals, which are important in defining the event type and the earth's structure;
- (g) *an event window* - a time sequence whose endpoints bracket a seismic event of interest. This window is usually obtained through use of a triggered seismic network, and may contain many possible false alarms in addition to the main event;
- (h) *event location* - estimation of source parameters, including source position (latitude, longitude, depth) and origin time by using the *prior* results of arrival analysis. This can be achieved by using three-component recordings for a single station and using vertical component recording for a seismic network;
- (i) *event identification* - classification of an event into categories according to its nature such as earthquake, volcanoes, underground nuclear explosions, traffic events, industrial explosions;
- (j) *signal to noise ratio (SNR)* - the ratio between maximum vector amplitude of signal and a quiescent period immediately before the arrival onset.

## 1.3 AUTOMATIC SEISMIC ANALYSIS SYSTEM

### 1.3.1 Procedures of seismic analysis system

Once an earthquake occurs, the aim of earthquake analysis is to give a report about where and when the earthquake occurred and how large it was. Figure 1.2 illustrates a flow chart of the routine procedures involved in analysing an earthquake. As the earthquake waves pass a seismometer, the seismometer outputs electrical signals which correspond to the ground motion, usually, in the form of ground velocity. This signal triggers a recorder to store them into memory or magnetic media in analogue or digital format. The recorded data are then processed either manually by analysts or automatically by using computers. In Figure 1.2, five necessary procedures of seismic analysis are listed. The arrival picking and identification in grey boxes are the key procedures which I will apply artificial neural networks to them in this thesis. Note these five procedures are necessary for both a single station and a seismic network including many stations, but some procedures themselves have inherent differences.

Among the five procedures, the most important one is the arrival picking which aims to estimate the onset time of a definite seismic arrival reliably and accurately. There is no shortage of techniques which profess to tackle this problem (Allen, 1982; Anderson, 1978; Bache *et al*, 1990; Baer and Kradolfer, 1987; Chiaruttini, Roberto and Saitta, 1989; Chiaruttini and Salemi, 1993; Houlston, Waugh and Laughlin, 1984; Joswig, 1990, 1995; Joswig and Schulte-Theis, 1993; Kracke, 1993; Klumpen and Joswig, 1993, Pisarenko, Kushnir and Savin, 1987; Takanami and Kitagawa, 1988, 1993). However they do tend to be data specific and are not generally applicable. Analysts still need to pick arrivals visually by interactive means. There is no difference between picking methods for a single station and for a seismic network.

Following the arrival picking, the second important procedure is the arrival identification which aims to classify individual arrivals into categories relating to their amplitude, frequency, polarization and propagation. This is more difficult than arrival picking due to the complexity of signals. For seismic networks, there are some



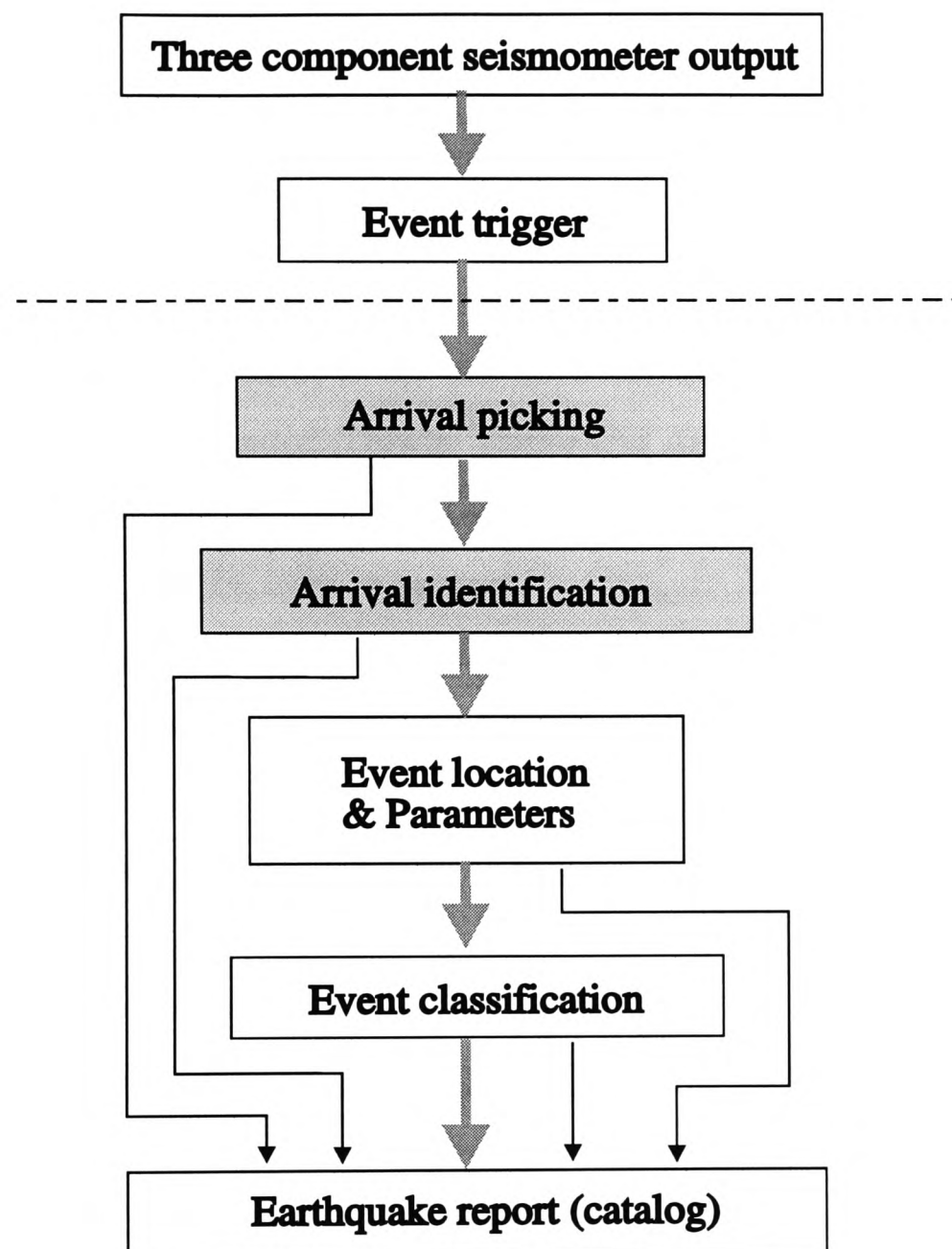


Fig 1.2. Standard procedures for seismic analysis. The whole flow includes two parts: above the dashed line, the functions are performed by hardware (seismometer); below the dashed line, the functions are performed by software or analysts. Among of them, the key procedures for automation are arrival picking and arrival identification (grey boxes) which are difficultly to automate by using conventional methods.

methods available for multi-station data (Mykkeltveit and Bungum, 1984; Bache *et al*, 1990; Kvaerna and Ringdal, 1992; Der, Baumgardt and Shumway, 1993). However, for a single station data, only a few methods can be used to pick special arrival types (Roberts, Christoffersson and Cassidy, 1989; Cichowicz, 1993; Tong, 1995; Tong and Kennett, 1995). It is an unsolved problem.

The third procedure, event location and description, is to estimate the source position of an event, its origin time, its magnitude, its focal mechanism, its spectrum, and so on. Some standard methods can be used to calculate them automatically by using computers if the arrival onset time and other recognition results are available.

The fourth procedure, event classification, is the most difficult one. Various methods have been suggested (Dowla, Taylor and Anderson, 1990; Leach, Dowla and Vergino, 1993; Riviere-Barbier and Grant, 1993; Hsu and Alexander, 1994), but no "agreed" methods can be used at present.

In the final procedure, results from the previous procedures are used in the production of an earthquake catalogue automatically using computers. A catalogue should include arrival onset time, source position, original time, magnitude, as well as focal mechanism, source spectrum if they are available.

### **1.3.2 Automated seismic analysis system**

An automated seismic analysis system (ASAS) should be capable of processing the seismic data automatically, quickly and reliably. Once an event triggers a seismic recording system, an ASAS should automatically recognize the event and calculate its parameters which include arrival onset times, arrival types, source location, magnitude, event types, *etc.* There are some systems and packages already available to this purpose (Roberts, Christoffersson and Cassidy, 1989; Ruud and Husebye, 1992; Nava, 1992; Musil, 1993; Oncescu and Rizescu, 1994; Chiarittini and Salemi, 1993; Joswig, 1993, 1995), but, most of them are interactive or semiautomatic and can only process seismic network data. The aim of this thesis is to improve on this by producing fully automated methods and then examine their performances.

### 1.3.3 Key problem in building ASAS

The key procedure in an ASAS is the automatic estimation of arrival onset time. At present, all seismic analysis systems need manual arrival-picking and arrival-identification procedures. Even if there is an automatic arrival picker, analysts are still required to interactively check the quality of the result. The procedures of arrival picking and identification involve extensive amounts of pattern recognition with which conventional methods are incapable of dealing. There is a pressing need to find an alternative tool to tackle this kind of complex pattern recognition problem. The application of artificial intelligence methods to earthquake analysis is a relatively new development which attempts to tackle these problems. Various methods have been applied to the interpretation of seismic signals from a local seismic network such as a knowledge-based system according to the blackboard method (Chiaruttini, Roberto and Saitta, 1989; Chiaruttini and Salemi, 1993), an intelligent monitoring system based on the knowledge-based system, database management systems and signal processing (Bache *et al.*, 1990), pattern recognition approaches for *P*-waves using a sonogram (Joswig, 1990), and identifying generic polarization patterns to estimate *P*- and *S*-wave onset times (Klumpen and Joswig, 1993).

The artificial neural network (ANN), one method of artificial intelligence, provides a natural alternative for automation of seismic analysis as it has proven useful at handling complicated pattern recognition problems due to its *learning or training* ability. For example, Dystart and Pulli (1990) use an ANN for the problem of automatic event classification. McCormack (1991) uses an ANN to combine synthetic spontaneous potential and resistivity logs to estimate lithology logs. Poulton, Sternberg and Glass (1992) use an ANN to estimate the offset, depth, and conductivity-area product of a conductive target given an electromagnetic image of the target. Wang and Mendel (1992) use a Hopfield network to implement an adaptive minimum prediction-error deconvolution. ANNs also have been used in the first-break picking of surface seismic data (Murat and Rudman 1992; McCormack, Zaucha and Dushek 1993) and earthquake detection (Wang and Teng, 1995). The wide range of

applications emphasizes the particular strength of ANNs over conventional methods incorporating a fixed algorithm to solve a particular problem. So the ANN will be used in this study.

## 1.4 OUTLINE OF THIS THESIS

In this thesis, I concentrate mainly on the development of the automated arrival picking and identification which form the basis of an automated seismic analysis system. The main objectives of these procedures are to pick *P*- and *S*-arrivals as reliably and accurately as possible, but also to reject noise bursts as reliably as possible. In this study, I will develop the approaches which can *automatically pick and identify seismic arrivals*. In principle, there will be essentially no difficulties in developing a complete ASAS based on these approaches if the time is available.

The analysis methods are different between single-component and three-component seismometers, between single station data and seismic network data, and between analogue data and digital data. In this thesis, I only focus on the seismic analysis of the *digital three-component* data from a *single* station. However, the developed methods can also be used to processing the seismic network data.

In Chapter 2, I first introduce the theory of ANNs and then develop my own *back-propagation neural network* (BPNN) system. There are some commercial ANN packages available, but these packages are difficult to combine with seismic procedures. Here, I intend to develop my own system which is simple and can easily be combined with seismic procedures.

In Chapter 3, I analyse the general characteristics of seismograms. In manual analysis, an analyst can directly look at the seismogram to pick the arrival and identify it, however a computer program cannot easily do this task. Some characteristics must first be extracted from the seismograms. In this chapter, I determine which characteristics can be obtained and which characteristics can be efficiently used in this study.

In Chapter 4, a BPNN approach is developed to *pick seismic arrivals using three-component data*. First, a BPNN is trained with a small number of real data, *P*-arrival segments which independently known its onset times and characteristics and background segments of vector modulus of three-component seismic recordings, and then this trained BPNN is used as a filter to pass through the entire seismic trace. Some simple criteria are applied to the output of BPNN to detect arrivals and pick their onset time. This approach can simultaneously pick both *P*- and *S*-arrivals. In this chapter, some post-processing are used to discard the noise and spikes.

In Chapter 5, the same BPNN approach is adapted to *pick seismic arrival using single component data* because at some stations only one-component data are available. In this approach, the absolute values of single component data are used as input both in the training and testing procedures. Due to the differences between single and three component data, this approach must be modified to suit the single-component data.

In Chapter 6, another BPNN approach is developed to *identify seismic arrivals* which are picked in Chapter 4. After the picking routines described in Chapter 4 and 5, the remaining problem is to identify the types of these picked arrivals. In this chapter, the identification is achieved by utilizing the polarization state of particle motion of *P*- and *S*-waves of three-component seismic recordings as a function of time, which directly input into a BPNN. It only deals with segments of the seismic arrivals which are previously picked. The BPNN output indicates the types of the picked arrivals.

Finally, Chapter 7 presents some discussions and conclusions of the whole thesis and speculation about possible future work. The work described in this thesis demonstrated that the ANN has an important role in the future automation of seismic analysis. In particular, the ANN can do the operations which conventional methods fail or find difficult to do and offers the ability to build up an automatic seismic analysis system which cannot be easily achieved by conventional methods.

## CHAPTER 2:

# INTRODUCTION TO ARTIFICIAL NEURAL NETWORKS

### 2.1 INTRODUCTION

Artificial neural networks (ANNs) are simple computer models that attempt to simulate the operation of neurons in the brain. This technology has been attracting attention in the artificial intelligence community for many years. The origins of ANNs can be traced back to the 1940s when psychologists began developing models of human learning. With the advent of the computer in the 1950s, researchers began to program ANN models to simulate the complex interconnections and interactions among neuronal cells in the brain. One of the most exciting developments in ANNs was the advent of the *Perceptron*, an idea that a network of elemental processors arrayed in a manner reminiscent of biological neural networks might be able to learn how to recognize and classify patterns in an autonomous manner. This model successfully exhibited various types of human learning behaviour. It was felt that with a large and fast enough computer, the entire human brain could be reproduced by a massive neural network. However, in 1969, Marvin Minsky, one of the founding fathers of artificial neural intelligence, proved mathematically that the *perceptron* - the simple ANN being studied at that time - was incapable of solving many simple problems. In the 1980s, after over a decade of being in the scientific wilderness, ANNs have once again become popular tools for many applications requiring algorithms with pattern recognition capability as those mathematical difficulties have been overcome by the introduction of more complex ANN architectures. These new ANN designs offer increased flexibility and robustness. They are particularly attractive as, unlike conventional methods that incorporate a fixed algorithm to solve a particular problem, ANNs utilize a learning scheme to develop an appropriate general solution, making them flexible and adaptive to different datasets. People began to realize the

potential value of ANNs as general purpose problem solvers, beyond their use as biological models.

Today, there are several dozen different ANNs paradigms available and widely used in many fields including a diversity of geological and geophysical problems. The details of theory and application of ANNs can be found in many research papers and comprehensive books which contribute ANN developments in theory and application (Fausett, 1994; Haykin, 1994). In this chapter, I shall only briefly introduce the basic concepts of ANNs and a special type of ANN - the *back-propagation neural network* - which is the most popular type of ANN in use today and used in this thesis.

## 2.2 THEORY OF ARTIFICIAL NEURAL NETWORKS

### 2.2.1 What are artificial neural networks?

Kohonen (1988) defined ANNs as follows: *"artificial neural networks are massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with the objects of the real world in the same way as biological neural systems do"*. This kind of ANN can be emulated by using a parallel computer or series computer which imitates the parallel process by software. In the terms of implementation of ANNs on a computer, *"an ANN is a network of many very simple processors (units), each possibly having a (small amount of) local memory. The units are connected by unidirectional communication channels (connections), which carry numeric (as opposite to symbolic) data. The units operate only on their local data and on the input that they receive via the connection"* (Prechelt, 1995). Most ANNs have some sort of *learning or training* rule by which the weights of connections are adjusted on the basis of presented patterns. As the concepts of an ANN originate from the human brain, the *biological neural network*, the knowledge of the human brain is essential for understanding ANNs. In this section, I first describe the basic operation of the biological neural network and then introduce the concepts of ANNs.

### 2.2.2 Biological neural networks

The biological neural network consists of neural cells (neurons), the fundamental elements of the brain, which are connected one to another in a complex spatial arrangement. A neuron is built up of three parts (Figure 2.1): the cell body which contains the *nucleus*, the hairlike *dendrites* around it and the *axon* which is the outgoing connection for a signal emitted by the neuron. The *axon* and *dendrites* are connected via a *synapse* which may modify the signal emitted by neurons.

In a brain, a neuron receives inputs from many other neurons via axons. If the energy level of the combined inputs exceeds a threshold level, then the neuron transmits to other neurons an electrical or chemical signal whose strength is modified by the synapses before entering other neurons. It is believed that certain forms of learning occur when the synapses are trained to assume certain strengths or weights by repeated exposure to the same stimulus. A single neuron is very simple and has very limited capability for solving problems, however, when several millions of those are connected to form a complex network, the brain can perform a range of extremely complex tasks including memory, pattern recognition and decision-making.

### 2.2.3 Artificial neural networks

An ANN consists of elemental processors called *nodes* (corresponding to the neurons) linked to others by interconnections (analogous to the dendrites and axons). Each interconnection has an associated scalar *weight* (corresponding to a synapse) whose value can be modified during the learning procedure of the ANN (Figure 2.1). Like a biological neuron, an artificial node consists of three parts: the activation function (corresponding to the nucleus), the output (corresponding to the axon) and the summed inputs (corresponding to the dendrites), associated with a scaling function called "weight" (corresponding to the synapse). The basic function of the node is to transmit the sum of weighted inputs to an output according to its preset activation function. It is used to assume that the weight is a real number which is applied as a simple multiplicative scalar to effectively amplify or attenuate the signal. Equation 2.1



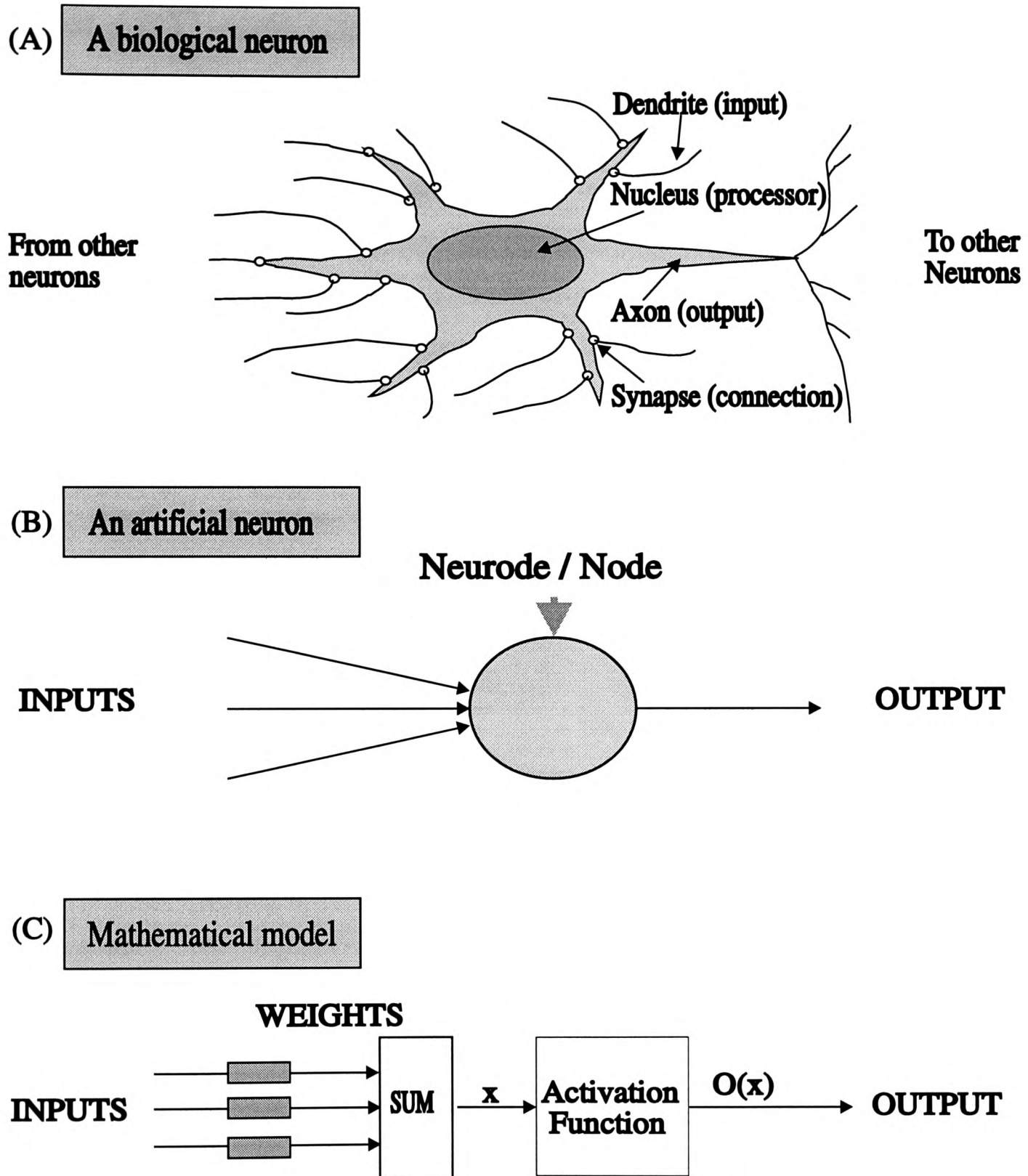


Fig 2.1. Schematic diagram showing various constituents of a biological neuron (A), an artificial node (B) and what they present (C). For the mathematical model of the artificial node, the factor  $x$  represents summed input to the node, and  $O(x)$  is the node output.

represents this procedure mathematically:

$$Output = f\left(\sum_{i=1}^N weight_i * Input_i\right) \quad [i=1,2,...,N]. \quad (2.1)$$

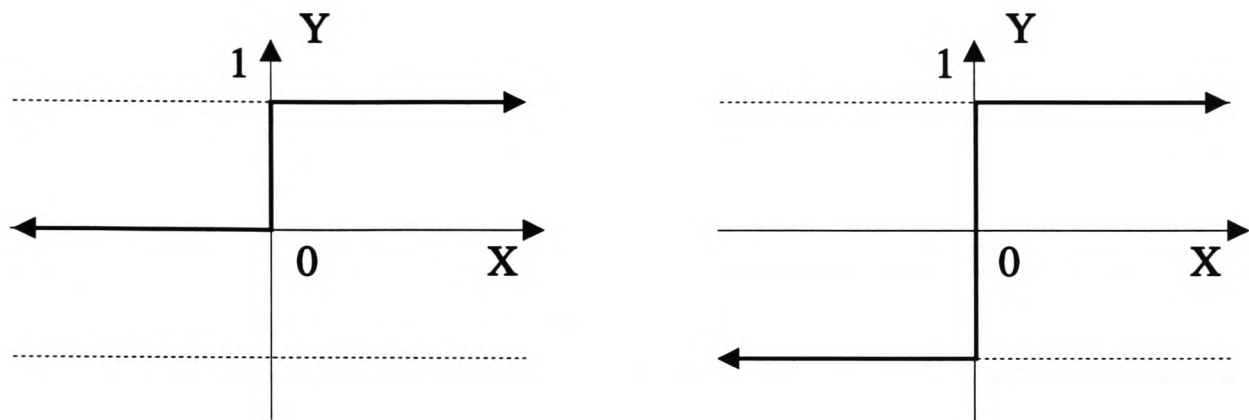
where,  $f$  is the *activation function*. There are many kinds of activation functions available, such as *Threshold logic*, *Hard-limit*, and *Sigmoidal* activation function (Figure 2.2). A node is typically characterized by its activation function, which operates on the input signal, summed individual weighted inputs from other nodes, and produces a signal output. Other nodes may each receive a different value via weighted connections to this node. The value may be either positive or negative, depending on the sign of the connection weight.

Such nodes connected together form an ANN in which all nodes operate simultaneously. Signals to an ANN from the outside world arrive via connections that originate in the outside world. Signals from the network to the outside world turnout via connections that leave the network.

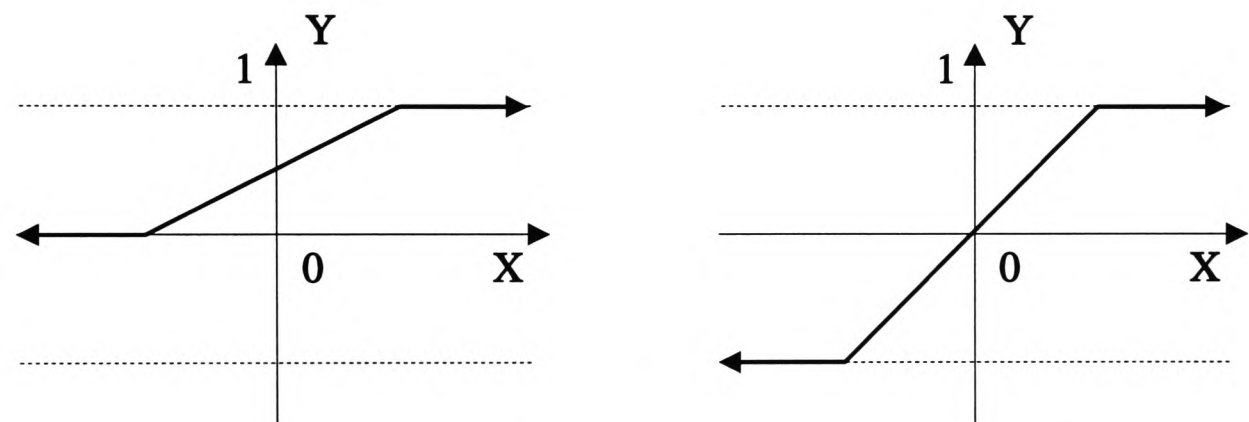
#### 2.2.4 How do ANNs work?

ANNs represent a fundamentally different approach to problem solving on conventional computers that are founded on underlying principles of logic and mathematics. The architecture used by most computers comprises a single central processing unit connected to an area of memory. This memory contains a stored program which is executed in a sequential manner by the central processor. Conventional computers concentrate on emulating human thought processes, rather than how they are actually achieved by the human brain. The disadvantage of this approach is that conventional computers cannot "learn" new knowledge. They have to be programmed precisely. ANNs, however, take an alternative approach in which they directly model the structure of the biological neural network and the way it processes information (but at a somewhat simpler level). The ability of an ANN to solve problems comes from emulating the natural "**learning (or training)**" procedure.

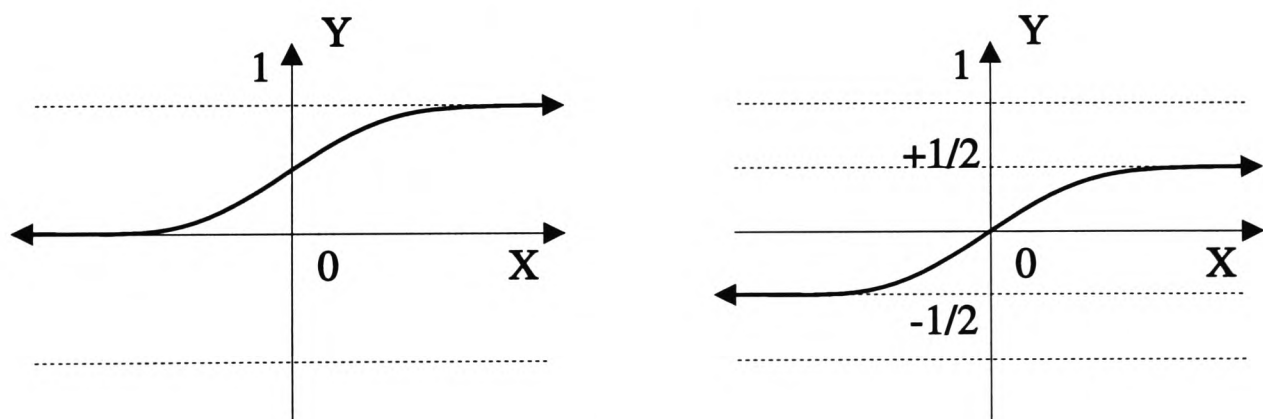
This is like teaching a child to study, the ANN is trained by repeated exposure



(a) Threshold logic activation function



(b) Hard-limit activation function



(c) Sigmoid activation function

Fig 2.2. The most commonly used activation functions. Here  $X$  is the summed input to a node,  $Y$  is the node output.

to patterns of training samples and saves these patterns in its memory somehow. After training, if new patterns are presented, this can be used to create recall. Figure 2.3 shows an example in which a supervised training scheme is applied to an ANN. In the training procedure, the input pattern of examples is fed into a new ANN to produce an output. A training rule then compares this output with the desired output and change the weights in the ANN so that the features of example patterns are stored in the ANN. After training, in the testing procedure, new data can be fed in the trained ANN to produce its output. In principle, the same ANN can be used to solve problems in many different fields according to the training samples. As a consequence, the learning procedure is important to the ANN as to a child.

### 2.2.5 Classifications of ANNs

Although the concept of an ANN appears initially to be quite straightforward, there is a bewildering array of different kinds of ANNs which now exist to solve different problems. Various ANNs have different topological structures of the neurons and their connections, and use different learning algorithms based on different philosophies. Figure 2.4 shows a classification of major types of ANN.

According to the node arrangement (usually in layers), ANNs can be divided into three major categories: *single-layer*, *bi-layers* and *multi-layers*.

According to the type of signal propagation direction in the different arrangement of nodes, ANNs can be *feed-forward* or *feed-back*. In a feed-forward network, all signals propagate only in a "forward" direction through the network layers. There is no self-connection, lateral connection, or back-connection. In a feedback network, signals may propagate "backward" as well as "forward" during processing, or propagate laterally between nodes in the same layer. Here, the input information defines the initial activity state of a system, and then the first output of the system is taken as the new input, which produces a new output.

According to the learning law, ANNs can be *supervised* or *unsupervised* in learning. In supervised learning, there is a "teacher" who teaches the network how well it performs or what the correct behaviour would have been. In unsupervised

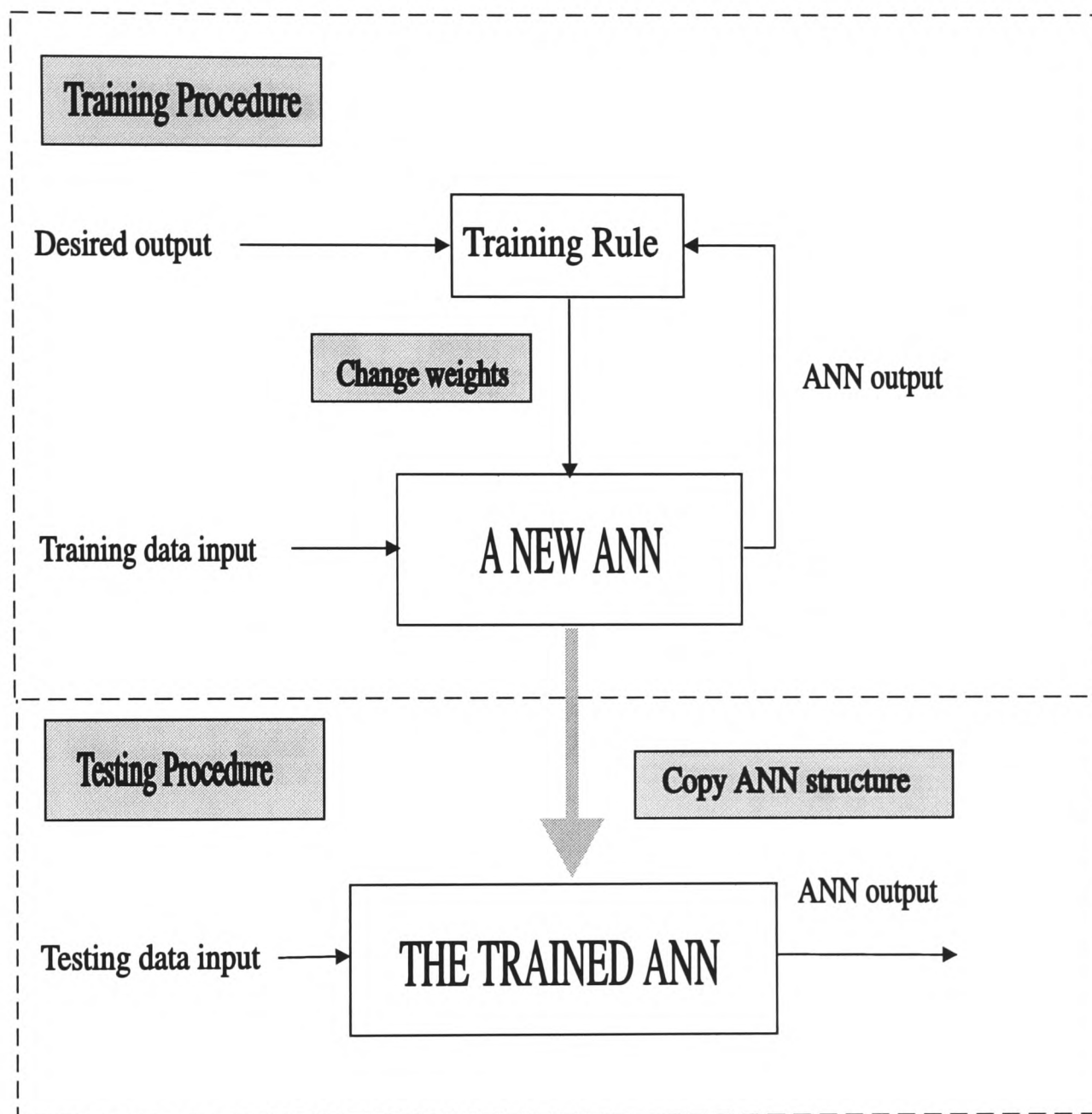


Fig 2.3. Schematic diagram showing the various component parts of an ANN training and generating algorithm. This is a form of supervised learning.

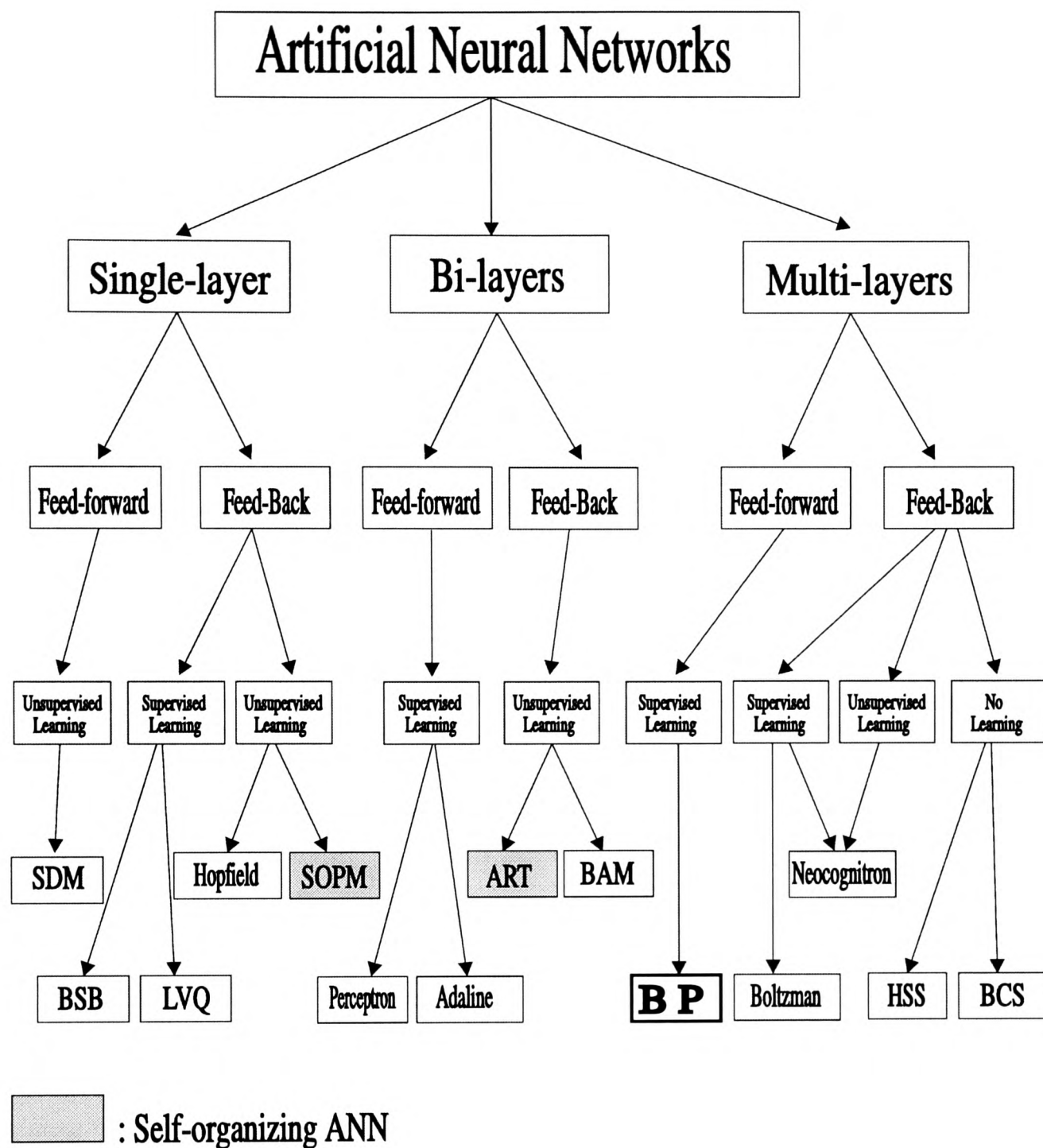


Fig 2.4 General classification of ANNs. The BP neural network is used in this thesis. Only small part of ANNs are shown. Their full names are listed as follows:

ART: Adaptive Resonance Theory  
 BAM: Bidirectional Associative Memory  
 BP: Back-Propagation  
 BSB: Brain-State-in-a-Box  
 HSS: Hierarchical scene structure  
 SDM: Sparse Distributed Memory Network.  
 SOPM: Self-Organizing Topology-Preserving Map

Adaline: ADaptive LINear Element  
 BCS: Boundary Contour System  
 Boltzman: Boltzman Machine  
 Hopfield: Hopfield network  
 LVQ: Learning Vector Quantization.

learning, the network is autonomous: it just looks at the data presented, finds out some properties of the dataset, and learns to reflect these properties in its output.

Some ANNs have the feature of *self-organizing*. In such a network, neighbouring nodes compete in their activation by means of mutual lateral interactions, and develop adaptively into specific detectors of different signal patterns.

There are many ANNs in use and many varieties and new ANNs are being developed. Figure 2.4 lists only a few of them. More ANNs can be found in the enclosed reference list (Maren, Harston and Pap, 1990; Fausett, 1994; Haykin, 1994). The most common one is the *back-propagation neural network* and is the one used in this thesis.

## 2.3 BACK-PROPAGATION NEURAL NETWORK

The back-propagation neural network (BPNN) (Rumelhart, Hinton and Williams, 1986, 1988) has the features of non-linear node activation function, multi-layer, feed-forward and back-propagation of error. The term *back-propagation* only refers to the training method by which the connection weights (as well as the node activation thresholds) of the BPNN are adjusted. The BPNN is probably the best-known and widely used option among the currently available ANN systems and is used here because it is suitable for the problems which involve a large amount of pattern recognition such as the arrival picking and identification. Figure 2.5 shows an example of the BPNN.

### 2.3.1 Theory of back-propagation neural network

#### 2.3.1.1 Basic concepts of BPNN and its operation

A BPNN is made up of several sets of nodes arranged in layers, consisting of an input layer, one or more intermediate hidden layers and an output layer. Each node, the basic processing unit, has a sigmoidal activation function (Figure 2.6). The sigmoidal activation function plays an important role in the BPNN because without such a continuous and nonlinear activation function, the BPNN is difficult to solve

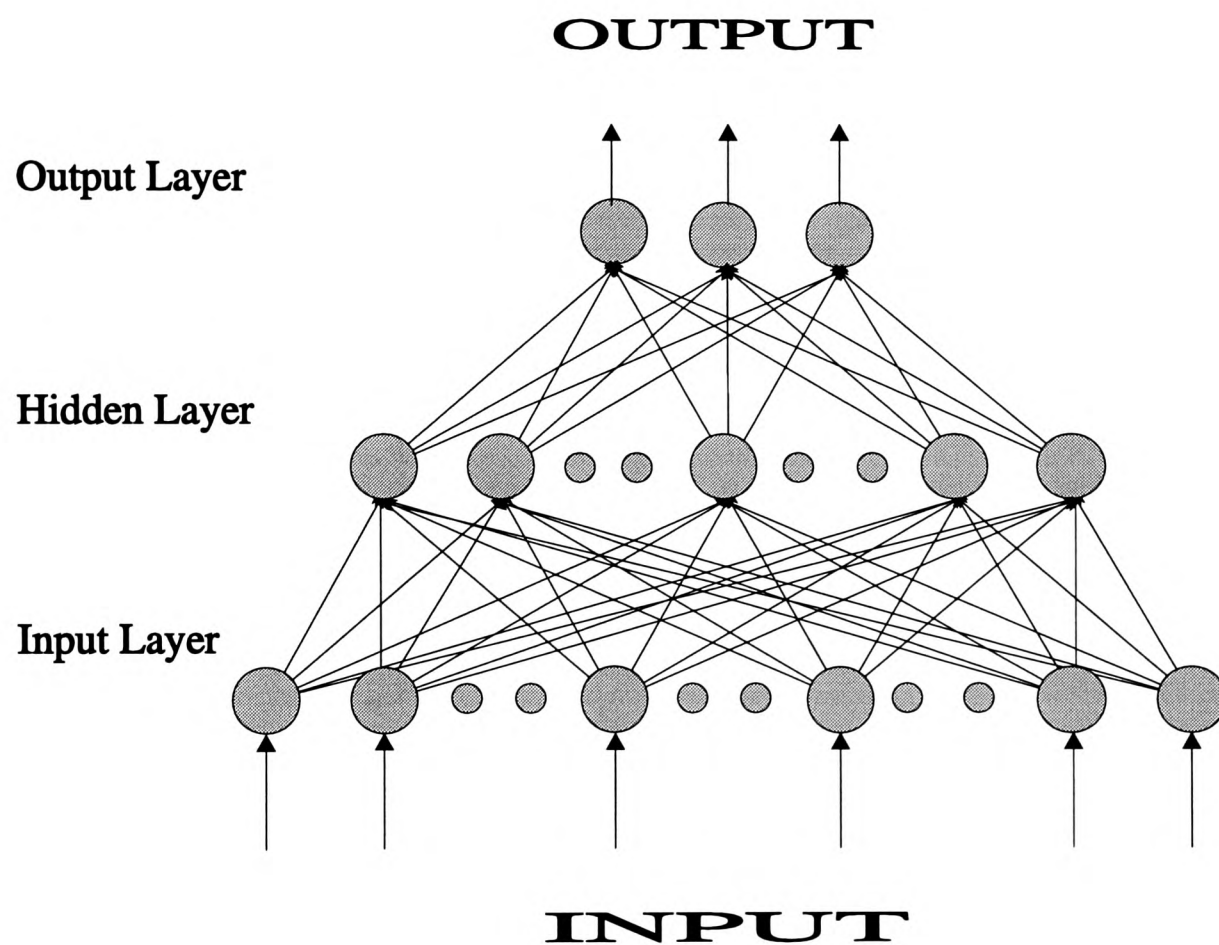
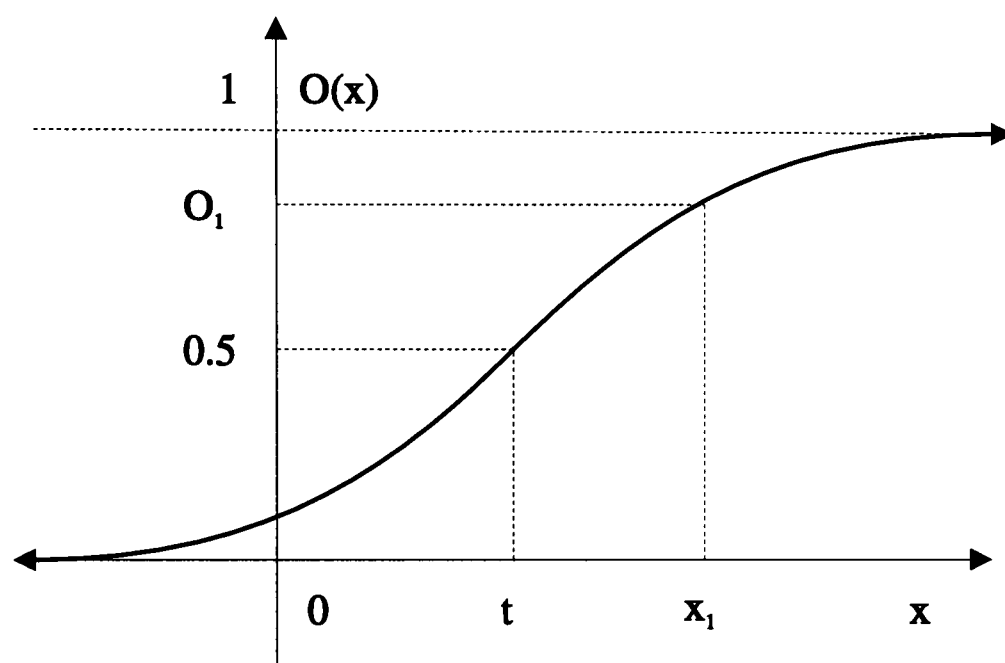


Fig 2.5. An example of BPNN. This BPNN has one input layer, one output layer and one hidden layer. This is the most commonly used structure for this kind of network.





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

Fig 2.6. The sigmoidal activation function of nodes for the BPNN. Here  $x$  presents summed input to the node,  $t$  the threshold and  $O(x)$  the output. Usually the output value given by the function closes to one of the asymptotic values,  $1$  for the higher-value asymptote, and  $0$  (or  $-1$ ) for the lower one. When the summed input  $x$  equal to threshold, the output value is  $0.5$ .

even simple problems (Rumelhart, Hinton, Williams, 1988; Pao, 1989; Maren, 1990a). The outputs of nodes in one layer are transmitted to nodes in another layer through weighted connections. Except the nodes in the input layer, the network input to each node is the sum of the weighted output of nodes in the previous layer. Each node is then active according to the summed input using a preset sigmoidal activation function and a threshold parameter for the function. In the input layer, the network inputs to each node are the components of the input pattern.

The BPNN architecture is schematically illustrated in Figure 2.7. Here,  $n_{ij}$  is defined as the  $j$ th node in the  $i$ th layer,  $w_{ijk}$  the weighted connection between  $n_{ij}$  and  $n_{i+1 k}$ ,  $q_{ij}$  the threshold of  $n_{ij}$ ,  $o_{ij}$  the output of  $n_{ij}$ , and  $net_{ij}$  the input of  $n_{ij}$ . For the training procedure, it is also necessary to define  $\delta_{ik}$  which reflects the change in error as a function of the change in the network input to the  $n_{i+1 k}$ . The components of an input pattern are defined as  $x_i$  [ $i = 0, 1, \dots, N-1$ ], and the desired output is defined as  $t_i$  [ $i = 0, 1, \dots, M-1$ ].

In the feed-forward operation, the information is presented to the input layer and then processed by one or more intermediate ("hidden") layers before appearing at the output layer. The network input to  $n_{i+1 k}$  in hidden layers and the output layer is:

$$net_{i+1 k} = \sum_{j=0}^{J-1} w_{ijk} o_{ij} + q_{i+1 k} \quad (2.2)$$

For convenience, the threshold  $q_{i+1 k}$  can be treated as a weight connected to a node ( $n_{iJ}$ ) which always has a unity output ( $o_{iJ}=1.0$ ) in the previous layer. That is:

$$q_{i+1 k} = w_{iJk} o_{iJ} = w_{iJk} \quad (2.3)$$

From this, Equation 2.2 becomes:

$$net_{i+1 k} = \sum_{j=0}^J w_{ijk} o_{ij} \quad (2.4)$$

The output of  $n_{i+1 k}$  is:

$$o_{i+1 k} = f(net_{i+1 k}) \quad (2.5)$$

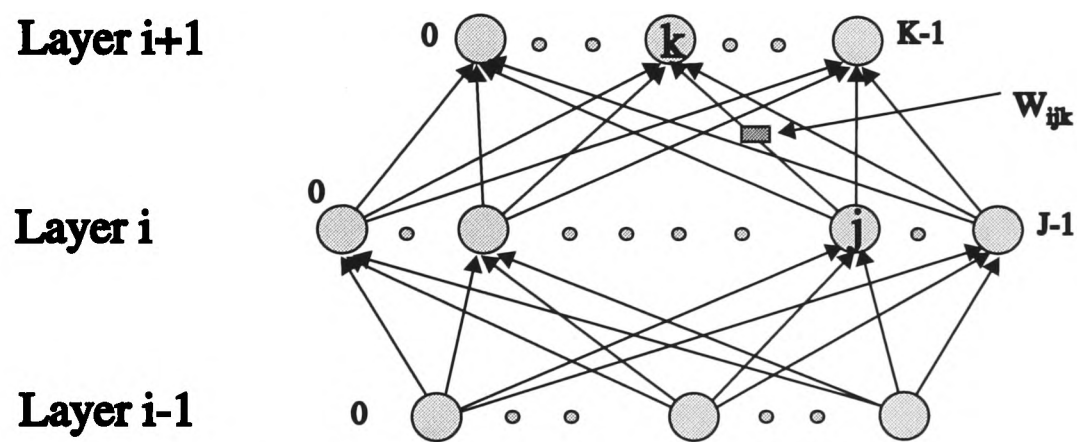


Fig 2.7. A schematic depiction of a part of a BPNN structure. The shaded circles are nodes  $n_{ij}$  and the straight lines are connections with weight  $w_{ijk}$

where  $f$  is the activation function of nodes. For a sigmoidal activation function,

$$o_{i+1k} = \frac{1}{1 + e^{-net_{i+1k}}} . \quad (2.6)$$

However, in the input layer, the output of the nodes is directly equal to a component of the input patterns,

$$o_{0j} = x_j , \quad [j=0,1,...,N-1] , \quad (2.7)$$

where layer 0 is the input layer.

### 2.3.1.2 Learning algorithm - *Generalized Delta Rule*

The BPNN is commonly trained by a supervised learning procedure which involves the presentation of a set of pairs of input and output patterns to it. Its ability to recognize new patterns depends on the training patterns used as references. The learning occurs when the weights and thresholds are trained to assume certain values by repeated exposure to the same patterns. The most popular way to adjust the weights and threshold values of the sigmoidal functions is to use the *Generalized Delta Rule* (Rumelhart, Hinton and Williams, 1986). It supports a wide range of applications, particularly for classification and prediction, and it is also particularly easy to develop neural networks by using it. Most ANN development tools support it or its variations (Tubb,1993).

In this learning algorithm, the BPNN first uses the input pattern to generate its own output pattern and then compares this with the desired output pattern. If there is no difference, no learning takes place. Otherwise, the weights and thresholds are changed to reduce the difference. The *Delta Rule* attempts to find the most suitable solution (in the form of numerical values of weights and thresholds) for global minimization of the mismatch between the desired output pattern and its actual value for all of the training examples. The degree of mismatch for each input-output pair is quantified for solving the unknown parameters (weights and thresholds) between the hidden and output layer and then the mismatch propagates backwards through the BPNN to adjust the parameters between the input layer and hidden layer.

Mathematically, in the Generalized Delta Rule, the change of weights can be written (see Appendix A) as:

$$\Delta w_{ijk} = \eta \delta_{ik} o_{ij} \quad , \quad (2.8)$$

where for the output layer:

$$\delta_{ik} = (t_k - o_{i+1k}) o_{i+1k} (1 - o_{i+1k}) \quad , \quad (2.9)$$

and for hidden layers:

$$\delta_{ik} = o_{i+1k} (1 - o_{i+1k}) \sum_{j=0}^J (\delta_{i+1j} w_{i+1jk}) \quad . \quad (2.10)$$

$\eta$  is a proportional constant called the *learning rate*. A large value corresponds to rapid learning but might result in unstable oscillations. A *momentum* term was introduced by Rumelhart, Hinton and Williams (1986) to overcome this problem. In this case, Equation (2.8) is written as:

$$\Delta w_{ijk}(n+1) = \eta \delta_{ik} o_{ij} + \alpha \Delta w_{ijk}(n) \quad , \quad (2.11)$$

where the quantity  $(n+1)$  indicates the  $(n+1)$ th step and  $\alpha$  is a proportional constant called the *momentum rate*. That means the change of weights at the  $(n+1)$ th step should be somewhat similar to the change of weights undertaken at the  $n$ th step.

### 2.3.2 Training procedure of BPNN

To train a BPNN, the first input pattern is presented to an initially randomized BPNN, and the weights and thresholds adjusted in all the connections. Other patterns are then presented in succession, and the weights and thresholds adjusted from the previously determined values. This process continues until all patterns in the training set are exhausted (*an iteration*). The final solution is generally independent of the order in which the example patterns are presented. However, a final check can be performed by looking at the *pattern error* ( $E_p$ ) and *system error* ( $E_s$ ) to determine whether the final BPNN solution satisfies all of the patterns presented to it within a certain error. Here  $E_p$  is defined as the square of the difference between desired output and BPNN output for each pattern:

$$E_p = \sum_{i=0}^{M-1} (o_i - t_i)^2, \quad (2.12)$$

where,  $o_i$  is the node output in the output layer.  $E_s$  is defined as the average of all pattern errors,

$$E_s = \frac{1}{P} \sum_{p=0}^{P-1} E_p. \quad (2.13)$$

The settings of weights and thresholds in the BPNN are now specifically tailored to "remember" each input and output pattern, and can consequently be used to recognize or generate new patterns from an unknown input. The BPNN is now "trained" and can be used in subsequent analyses. Figure 2.8 summarizes the various stages of training and implementation.

- Step 1:** Create a BPNN structure (node numbers in each layer) according to a problem. The input and output nodes can be determined by the problem, but the hidden numbers are ambiguously determined.
- Step 2:** Initialize all weights and thresholds to be small, random numbers. (e.g. between  $\pm 0.5$  or  $\pm 1.0$ )
- Step 3:** Select or create several examples consisting of input patterns and desired output patterns for training.
- Step 4:** Sequentially feed input patterns into the BPNN and propagate them through all connections and nodes to produce actual output patterns in feed-forward manner.
- Step 5:** For each input pattern, compute the error between the actual output and desired output and use this error to adjust weights and thresholds by using the *Delta Rule*.
- Step 6:** If the errors is greater than the allowed error, then go back to Step 4 or else finish training and go to Step 7.
- Step 7:** Save the weights and thresholds.

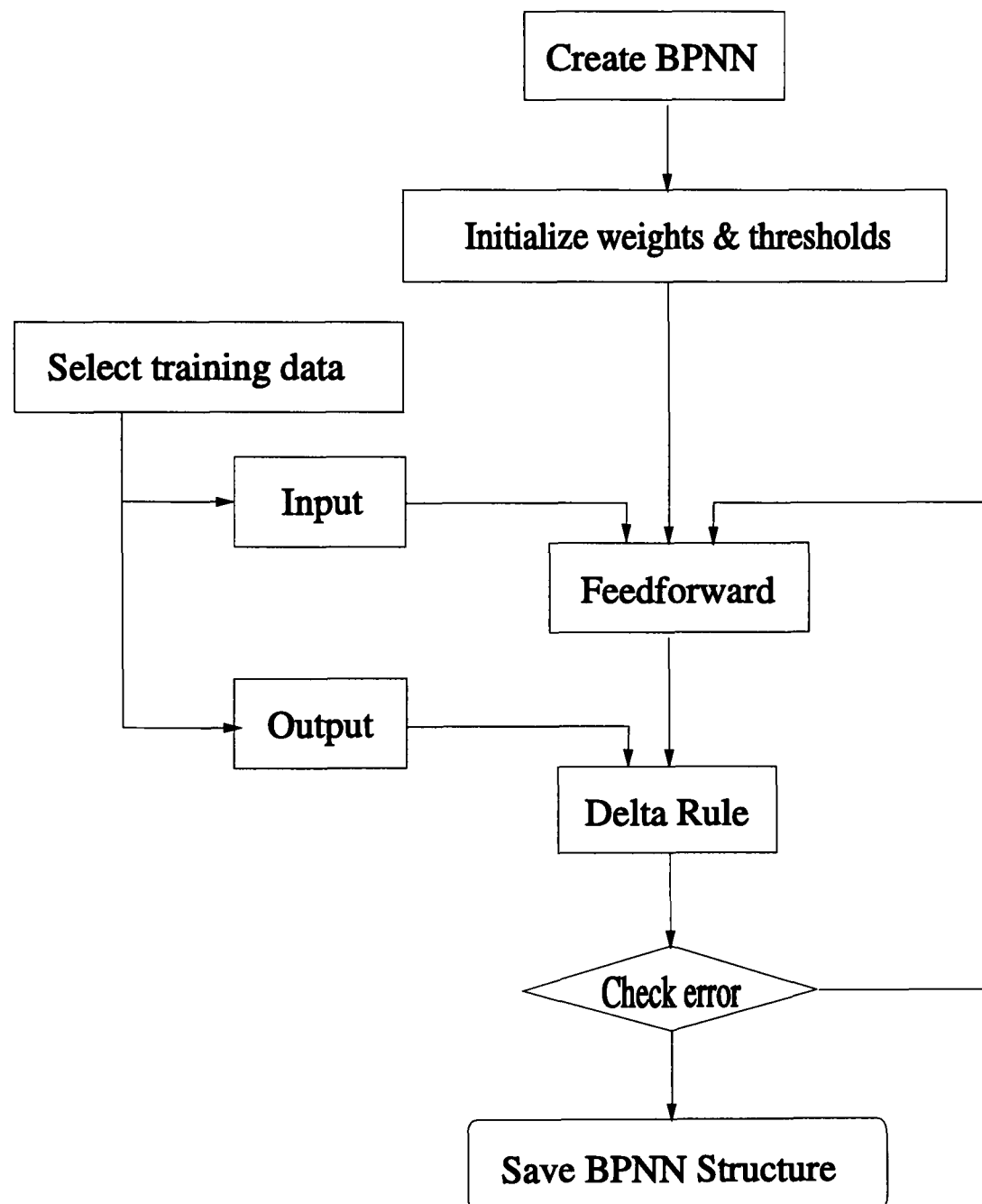


Fig 2.8. The various stages of training a BPNN using the *Generalized Delta Rule*.

### 2.3.3 Computer programming

Although there are some commercial ANN packages available, they cannot directly be applied to seismic analysis. It is considered better to write customized ANN software which can meet the needs of seismic analysis and can then be fed directly into conventional seismic processing software. The ANN will then be a component of a general seismic analysis system. The choice of a programming language is more critical than in other situations due to the computation demands of ANNs. Whatever language is used, it is advisable to seek a version that has been optimized for numeric data. The C language has become something of a de facto standard for neural network programming. My own BPNN was written using C programs on a VAX4000 computer under the VAX/VMS™ operation system.

### 2.3.4 Selecting BPNN structure and training parameters.

The need of applying the BPNN to a problem is, first, to select its structure which includes the numbers of input nodes, output nodes, hidden layers and hidden nodes, and then to initialize the weights and thresholds, and to determine the learning rate  $\eta$ , the momentum rate  $\alpha$ , and the thresholds of system error and pattern error which affect the convergence in the learning procedure. The input and output nodes are manually determined by the problem itself, input data and output required. The selection of hidden layers and hidden nodes is somewhat more indeterminate. Although a single hidden layer is sufficient to solve any function approximation problem, some problems may be easier to solve using a BPNN with two hidden layers (Fausett, 1994). The number of hidden nodes depends on various factors such as input nodes, output nodes, system error, pattern error and training samples. There is no fixed generic relationship between the number and these factors. We do know that in the BPNN learning, generalization is increased and memory is reduced by limiting the number of hidden nodes (Dowla, Taylor and Anderson, 1990). Too few hidden nodes will lead to a long learning process or no convergence. So an optimum number exists for any given problem (Gorman and Sejnowski, 1988).

The choice of initial values of weights and thresholds can be of tremendous



help in a successful network design. This choice will influence whether the BPNN reaches a global (or only a local) minimum of the error and, if so, how quickly it converges in training (Fausett, 1994). The BPNN must not be allowed to start off with a set of equal weights and thresholds. It has been shown that it is not possible to proceed from such a configuration to one of unequal weights (Rumelhart, Hinton, and Williams, 1986). A common procedure is to initialize the weights to random values. Note that the values for the initial weights must not be too large or too small. The range of the initial weights is normally between -0.5 and 0.5 (or between -1 and 1, or some other suitable interval) (Fausett, 1994).

The determination of the learning rate  $\eta$ , momentum rate  $\alpha$ , and the thresholds of system error and pattern error is also arbitrary. They may have different values depending on the problem and the researcher's favourite choice. Naturally, the higher the learning rate is, the larger the change in the weights. For practical purposes, it should be chosen as large as possible. However, if the learning rate is too high, it will lead to unstable oscillation. This oscillation can be filtered out by the *momentum term*. Rumelhart, Hinton and Williams (1988) suggested 0.9 for the *momentum rate*  $\alpha$ . In most of my simulations,  $\eta$  is 0.7 and  $\alpha$  is 0.9. The thresholds of pattern error and system error determine the termination of the training procedure. Here I select the thresholds 0.0001 for the pattern error and 0.00001 for the system error.

## 2.4 DISCUSSIONS AND SUMMARY

In principle, ANNs can compute the value of any real function, i.e. they can do everything a normal digital computer can do. However, ANNs are not panaceas. They can resolve a wide range of problems, but sometimes they cannot give expected results. The most common reason that ANNs do not work is that they are being used in a wrong place, trying to do a wrong sort of job. ANNs must be considered as one part of an intelligence system and their application must be clearly understood (Macleod, 1992). Before applying ANNs (especially the BPNNs) to problems, we must bear in mind what features they have and what they can do. There follows a general

discussion of these two features.

#### 2.4.1 Advantages and disadvantages of BPNN

The significant feature of ANNs is their learning ability. Although each ANN has its own special features, it seems that all ANNs offer the universal processing advantages which include:

***Adaptive learning:*** an ability to learn how to do tasks based on the data given for training or initial experience.

***Self-organization:*** an ANN can create its own organization or representation of the information it receives during learning and operation.

***Fault-tolerance via redundant information coding:*** partial destruction of an ANN leads to the corresponding degradation of performance; however, some capabilities of the ANN may be retained even with major damage of the ANN.

***Real-time operation:*** ANN's computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantages of this capability (Maren, 1990b).

However, the disadvantages are dependent on the types of ANNs. Here I only list the disadvantages of the BPNN, which include:

***Slow learning rate:*** for problems requiring a large and complex BPNN architecture or having a large number of training examples, the time needed to train the BPNN can become excessively long.

***Trapping in local minima:*** under some circumstances, the system error can remain large regardless of how many iterations are carried out. The training procedure then cannot converge however long the routine is run.

***Forgetfulness:*** an BPNN tends to forget old training examples as it is presented with new ones. A previously trained BPNN which must be updated with new information has to be retrained using both the old

and new examples.

***Imprecision:*** BPNNs do not provide precise numerical answers, but rather relate input patterns to the most probable output space.

***Black-box:*** BPNNs provide no insights into the physics of processing.  
(McCormack, 1991)

### 2.4.2 Applications of BPNN

A distinct characteristic of the BPNN is that it forms a mapping from a set of input patterns to a set of output nodes using features extracted from the input patterns. A BPNN can be designed and trained to accomplish a variety of mapping operations, some of which are very complex, and to develop the ability to generalize. This is because the nodes in the hidden layers of the BPNN learn to respond to general features found in the input patterns. Compared to conventional computing approaches, a BPNN is efficient to solve problems which involve a large amount of pattern recognition and is ideal when the relationship between two sets of data is unknown but one is expected to exist. It may offer a way of building a "black box" system without understanding how it works, but, in reality, a great deal of knowledge and insight is usually required to use the technology successfully.

BPNNs, as well as other ANNs, offer improved performance over conventional techniques in robust pattern detections, signal filtering, data segmentation, data compression, database mining and associative search, adaptive optimization, scheduling and routing, complex mapping and modelling complex phenomena, and adaptive interfaces for man/machine systems. BPNNs have been used to solve a diversity of geological and geophysical problems (Cai, et al., 1993; Cary and Upham, 1992; Dai and MacBeth, 1993, 1994a, 1994b, 1994c, 1995a, 1995b; Dowla, Taylor and Anderson, 1990; Dystart and Pulli, 1990; Higgins and Hsu, 1994; Langer, Nunnari and Occhipinti, 1993; Leach, Dowla and Vergino, 1993; Leggett, Sandham and Durrani, 1993; McCormack, 1991; McCormack, Zaucha and Dushek, 1993; Murat and Rudman, 1992; Palaz and Weger, 1990; Penn, Gordon and Wendlandt, 1993; Poulton, Sernberg and Glass, 1992; Raiche, 1991; Roth and Tarantola, 1994; Tao and Du,

1992; Wang and Mendel, 1992; Wang and Teng, 1995; Zhang, Song and Nie, 1991).

### **2.4.3 Principles of utilizing BPNN**

BPNNs have a great potential to resolve the problems which conventional methods find difficult to apply. However, they should not be viewed as direct competitors to conventional methods, but rather as complementary techniques. The most successful BPNN applications to data analysis have been those which operate in conjunction with other computing techniques, for example, using an BPNN to perform a first pass over a set of incoming data, then passing the results over to a conventional system for subsequent processing (Tubb 1993).

There are two groups of problems. For the first group, the nature of the problems and the steps which lead to their solutions are well-known and can be explicitly and logically described. Conventional computing approaches are effective for them. For the second group, it is not possible to describe an exact solution to the problem. Even though human beings can generally learn to perform such tasks well but computers are traditionally impossible to achieve. BPNNs are effective for problems involving learning from examples, rather than needing to be explicitly programmed. It is suggested that where there are conventional methods which are effective and work well, BPNNs should not be used, otherwise use them with care and appreciation of their limitation as well as their potential.

### **2.4.4 Applications of BPNN in this thesis**

In this thesis, the BPNN will be used for arrival picking and identification, which are the traditional pattern recognition problems in seismology already described in Chapter 1. In arrival picking, a BPNN operates as a fundamental filter in processing the entire seismic trace and produces a time series which better reflects the seismic arrival. Conventional methods are then used to process the time series to determine the arrivals, measure their onset time and discard noise. In arrival identification, a BPNN is used to classify the types of the picked arrivals.

## CHAPTER 3:

# EARTHQUAKE DATA AND CHARACTERISTICS OF SEISMIC RECORDINGS

### 3.1 INTRODUCTION

Generally speaking, a method for interpreting seismograms can be regarded as using a filter to operate on a characteristic function which is either extracted from the seismogram or consist of the seismogram itself. The filter will produce output features which indicate the signal properties such as the arrival onset of a wave, its polarization, its propagation direction, and its type. It may be any mathematical method such as a statistical method, a filtering process, a transformation, or an artificial intelligence method. The characteristic function is a time series which is obtained by performing a transformation on a seismogram. It might enhance desired signals from the seismogram or separate different signals according to their characteristics, so that the subsequent "filtering" procedure can be made easily or efficiently.

In this thesis, I have decided to utilize the BPNN as a filter for the purpose of arrival picking and identification. However, what characteristics of seismograms should be processed by the BPNN filter? It is important to select suitable characteristics of seismograms, otherwise they will give rise to a misleading interpretation. In this chapter, I will examine the general characteristics of seismograms and determine which particular ones can usefully be used in this study. Because a seismic recording is a complicated time series, it is difficult to generate synthetic data to emulate it, so I use real earthquake data from the TDP3 seismic network in this study. In this chapter, I shell briefly introduce the TDP3 seismic network and illustrate the data acquired from it, then I shall describes the characteristics of a seismic recording. Three characteristic functions: the *absolute value*

*function* of the single component recording, the *modulus* of the three component recordings and the *degree of polarization* of the three component recordings, used for the arrival picking and identification, are discussed.

## 3.2 LOCAL EARTHQUAKE DATA

### 3.2.1 TDP3 seismic network

The TDP3 seismic network was a part of the third Turkish Dilatancy Project (TDP3), a multidisciplinary geophysical project (Crampin, Evans and Üçer, 1985; Lovell, Crampin and Evans, 1987; Lovell, 1989). Lovell (1989) described the project in detail.

The TDP3 local earthquake network included nine stations equipped with three-component seismometers and two stations equipped with single-component seismometers, and was located near the North Anatolian Fault in Turkey. Figure 3.1 shows the geographic map of its stations. This network had an aperture of about 15km. Each station was equipped with either one or three *Willmore Mk III* seismometers, set to a free period of 1 second prior to installation, and with known sensitivities. This seismometer has an output proportional to ground velocity in the frequency range of around 5 to 20 Hz. The velocity response of the seismometer and recording system is flat between 2 and 26 Hz. The signals from the seismometer were fed into a *Racal FM amplifier/modulator*, fitted with a feedback circuit to produce a damping factor of 0.6. Data from stations were radiolinked to the base station at Hereke using *UHF FM* system. In the base station, signals were recorded by *Geostore* analogue tape recorders. Three *Geostores* were used on the base station. Each had a capacity of ten seismic and two flutter compensation channels, together with an internal clock and a channel for an external absolute time-standard. Signals from the three-component seismometer sets were recorded on adjacent tracks on the same *Geostore* head to minimise the timing error (0.002 second) resulting from head

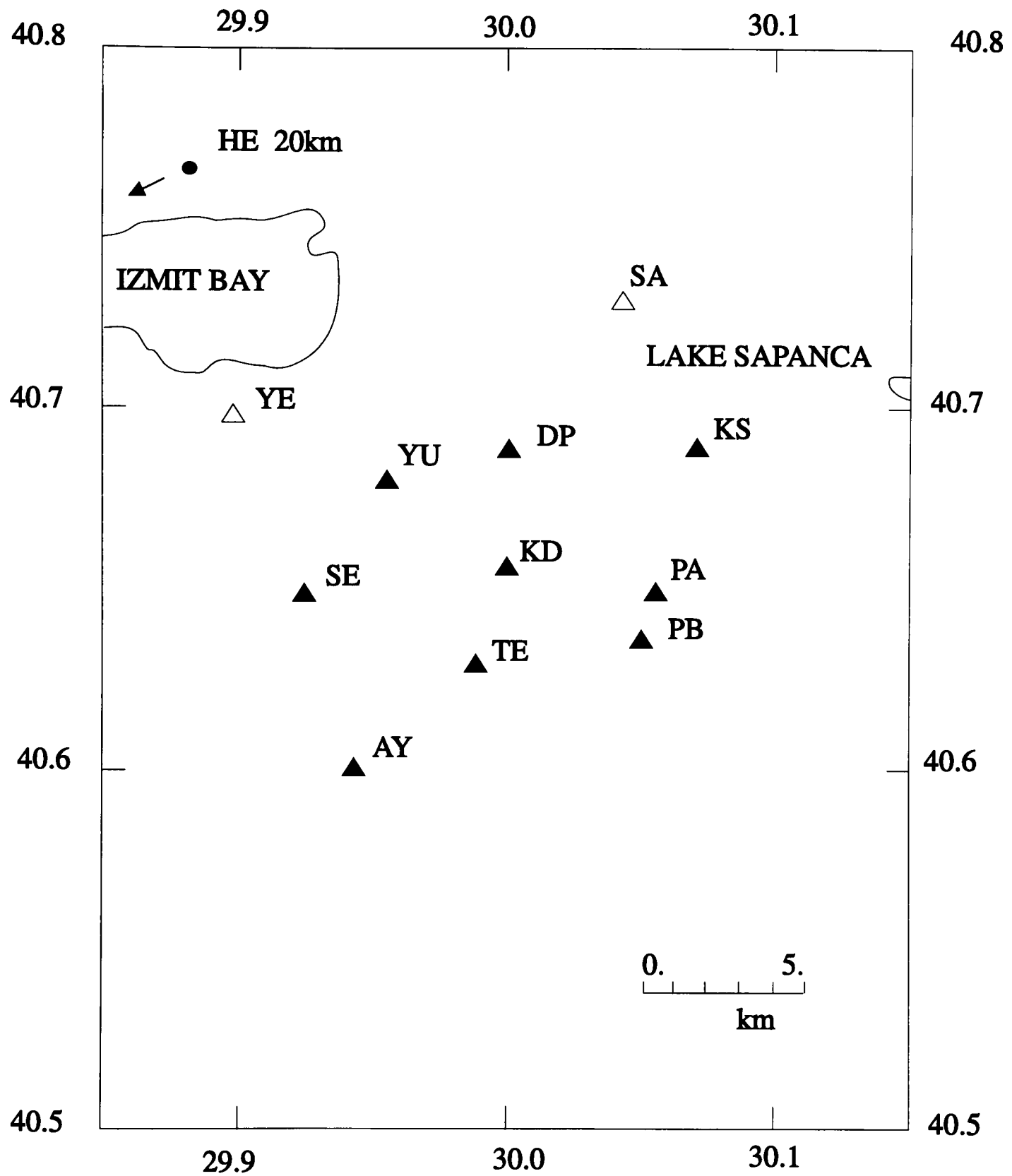


Figure 3.1. Map of the locations of stations used during TDP3. Filled and unfilled triangles represent three-component and single-component stations respectively. The base station is arrowed (re-drawn after Lovell, 1989).

misalignment. Systematic errors of up to 0.05 second can be introduced of a similar order as the reading and residual errors of the arrival time data.

### 3.2.2 Data acquired from TDP3

Between March and November 1984, over 150 data tapes (50 sets of three tapes, as three *Geostores* were used to record the incoming data signals) were accumulated during the TDP3 project. These tapes were transported back to British Geological Survey (BGS), Edinburgh, for bulk data processing. An audio replay unit was set up to identify the events and the data were digitized at 100 samples per second on a PDP-11 computer using software developed by Evans (1980) and an interface unit by Evans and Miller (1986) in BGS. Event locations were calculated using HYPO71 (Lee and Lahr 1975) adapted to run on the PDP-11. The digitized data were archived for subsequent analysis. All these data were analyzed by Lovell (1989).

Many hundreds of local earthquakes are recorded. Figure 3.2 shows their epicentres. In this thesis, only the recordings from stations DP and AY are used. Station DP is at the centre of the TDP3 seismic network and station AY is at the edge of the network. In total, 1754 three-component event recordings (877 from station DP, and 877 from station AY) were obtained. However, not all of these recordings can be used in this study. In some cases the seismometers did not function properly, and either one or two components were inactive or possessed high amplitude noise so that some three component sets were incomplete. There are also some recordings with excessive noise preceding the event or ringing throughout the record. By selecting the recordings manually, I find that 506 recordings from station DP and 486 recordings from station AY are unusable, leaving 371 and 391 recordings respectively for further processing. All events in these recordings were local with depths between 2km and 14km and epicentral distances less than 30k, and had magnitudes ( $M_L$ ) between -0.3 and 2.0. Most are closer to station DP than to station AY. They possess a wide distribution of SNRs lying between 1 and 200, with station DP being of higher fidelity than station AY (Figure 3.3). For these local events, the predominant *P* and *S* waves were manually identified from seismograms.



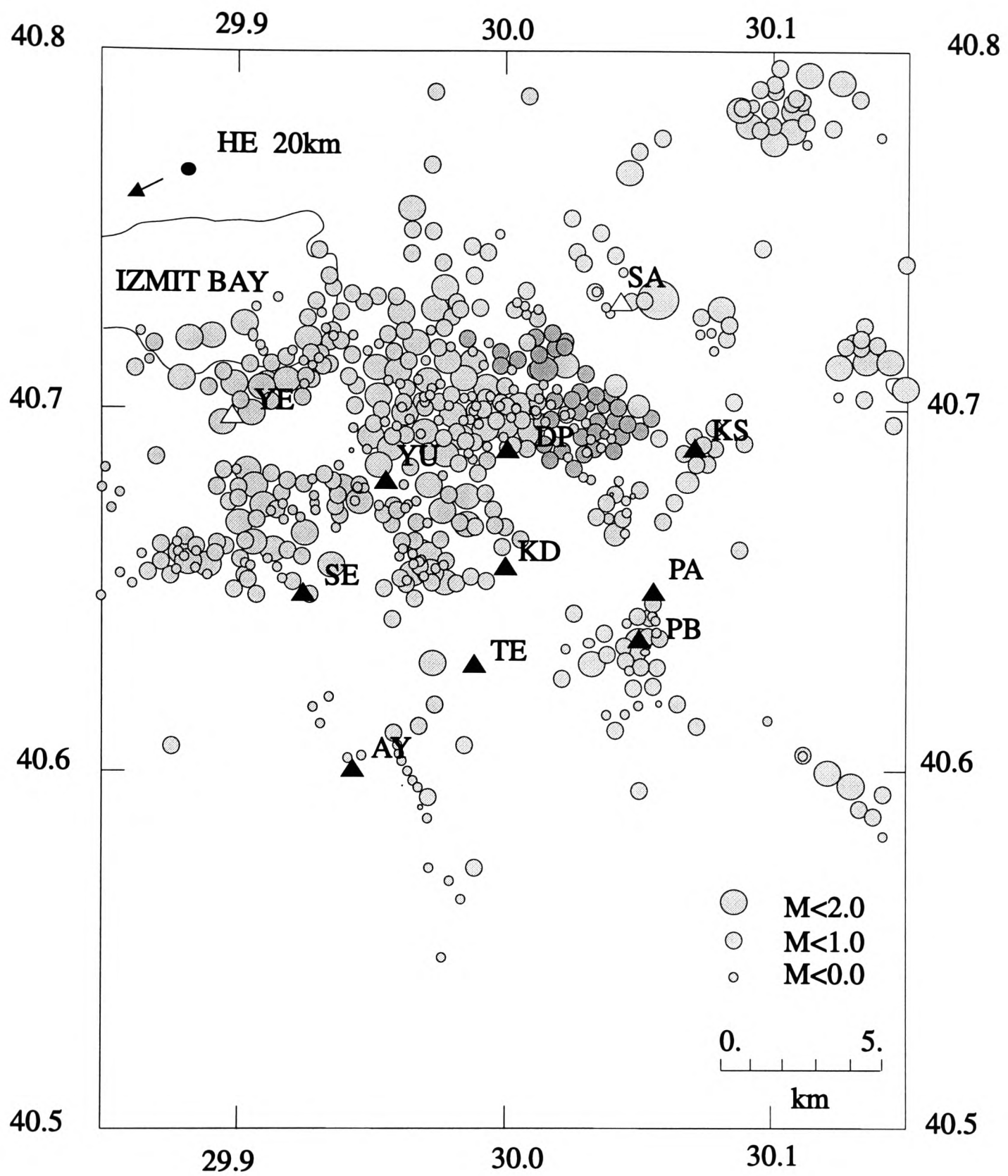
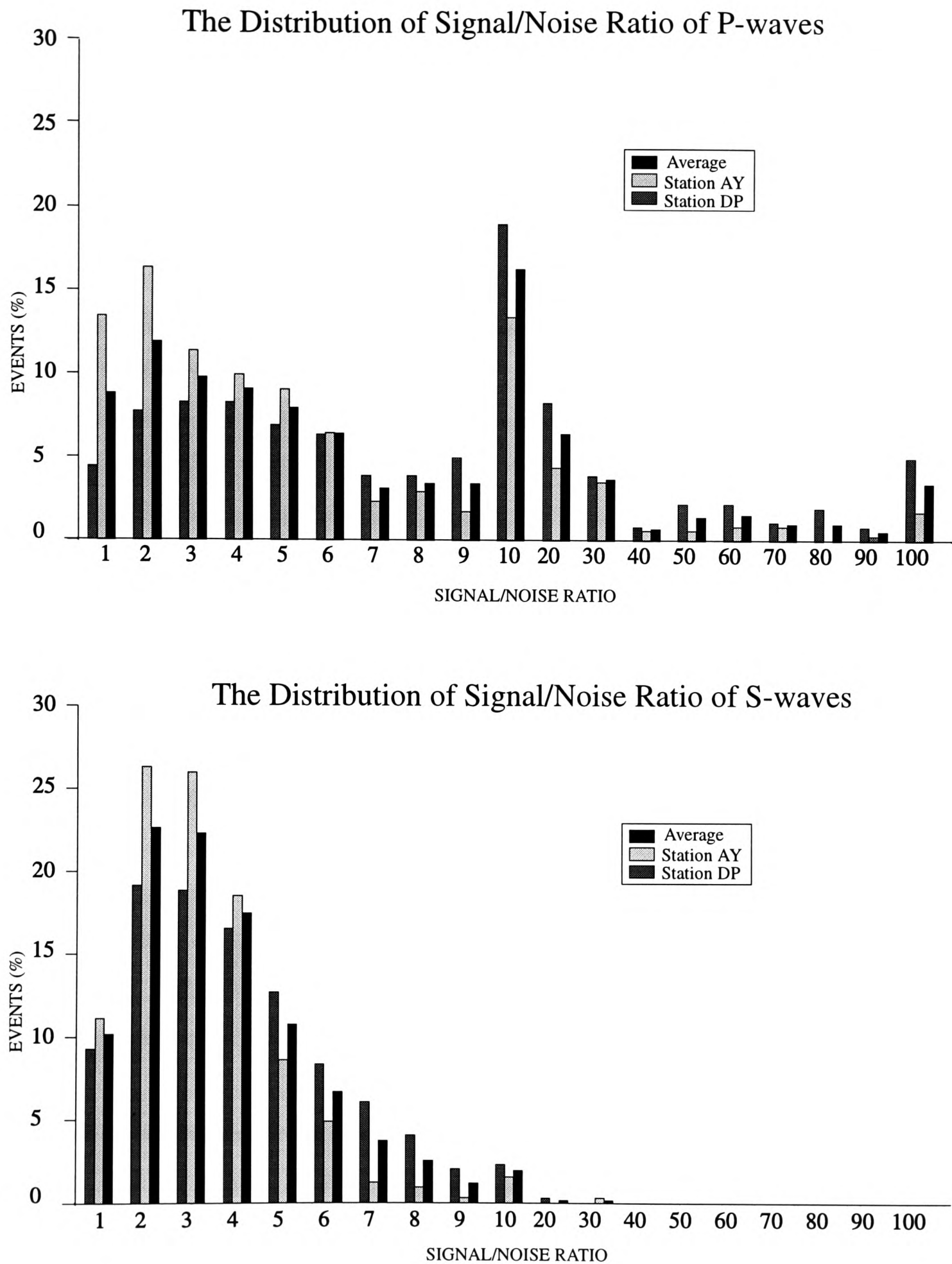


Figure 3.2. Map of the events located during the TDP3 (re-drawn after Lovell, 1989).



**Figure 3.3.** The distribution of signal/noise ratio (SNR) of P-waves and S-waves in the dataset. SNR is defined as the ratio between the maxima of modulus of 3-C recordings before and after the onset time. Notation for SNR is 1 for [1,2), 2 for [2,3), ..., 10 for [10,20), ..., and 100 for [100, 200).

### 3.3 CHARACTERISTICS OF SEISMIC RECORDINGS

The direct output of most seismometers represents the ground motions in Cartesian rectangular coordinates since they measure the linear motions along these orthogonal axes, usually along the Vertical (V), East-West (E-W), and North-South (N-S) directions. The original output represents the time-history of the seismic waves arriving at the seismometer. Although changes in amplitude, frequency, and direction of this motion indicate the arrival times of seismic waves, the type and polarization of wave motion, the direction of propagation, and other parameters (Aki and Richards, 1980; Lomax and Michelini, 1988), it is difficult to identify these characteristics directly from the original recordings because various characteristics are mixed together in the original recordings. However, such characteristics can be obtained by performing various transformations on the original recordings. These transformations can present the data in a form that can give greater weights to the most desired characteristics. Transformations of the ground motion which emphasize amplitude and direction information would greatly benefit seismogram analysis. Typically, this is done with a change of coordinate system or reference frame. Examples of such transformations include time series analysis, Fourier transform, axis rotation, or coordinate transform. Some of them can operate on single-component recordings and some on three-component recordings. In this section, I discuss the characteristics which are transformed from single component recordings or three component recordings for seismic arrival picking and identification.

#### 3.3.1 Characteristics from single component recordings.

There is a long history of using single component (1-C) recordings (usually vertical recordings) in seismic analysis because these have been available since the early day of seismic analysis and now are still in use. However, 1-C recordings only represent the ground motion in one direction, so polarization and propagation information cannot be obtained. However, various characteristics such as amplitude and frequency information can be obtained from the 1-C recordings by taking different

transformations. Here I only discuss the characteristics which are used in seismic arrival picking and identification. The characteristic may be as simple as the absolute value of an input series, or it may be quite a complicated function, depending on the type of signal expected and the required performance of seismic analysis. Five basic transformations are listed here: *absolute value function*, *square function*, *Allen's function*, *envelope function*, and *modified differential function*. Usually, the input signal is the vertical component recording,  $v(i)$ .

**(a) Absolute value function (AVF)**

This function is the simplest function to compute and is currently the most widely used for arrival picking. It is defined as:  $AVF(i)=|v(i)|$ . Although this function is not so responsive to changes in the input, it has been used by many successful picker algorithms, including those of Anderson (1978), Matsumura *et al.* (1981), and McEvilly and Majer (1982).

**(b) Square function (SF)**

This function is designed to enhance amplitude changes but not frequency changes. It is defined as:  $SF(i)=v(i)^2$ . Swindell and Snell (1977) reported its use in their investigation of picking algorithms, but it has not been widely used.

**(c) Allen's characteristics function (ACF)**

Allen (1978) designed this function to enhance changes in both amplitude and frequency. It is defined as:  $ACF(i)=v(i)^2+k[v(i)-v(i-1)]^2$ , where  $k$  is a weighting constant which varies with the sample rate and station noise characteristics. This function requires significantly more computing time than both the absolute value function and square function, so in some applications this might render it impractical. Allen (1978) developed his famous arrival picker based on this function.

**(d) Envelope function (EF)**

This function can be thought of as a positive outline of the seismogram and is defined as:  $EF(i) = \sqrt{v(i)^2 + \bar{v}(i)^2}$ , where  $\bar{v}(i)$  is the Hilbert transform of the  $v(i)$  (Kanasewich, 1981). Baer and Kradolfer (1987) and Earle and Shearer (1994) utilized this function or its square in their automatic picking algorithm.

**(e) Modified differential function (MDF)**

This function, introduced by Stewart (1977), enhanced the high frequencies of the input signal. There are two steps in defining it. The first step is to compute the difference of the input signal:  $DF(i) = v(i) - v(i-1)$ . The second step is to compare the sign of  $DF(i)$  and the prior  $DF(i-1)$ . If the  $DF(i)$  and  $DF(i-1)$  have the same sign and if the sign of  $DF(i)$  has persisted for less than some samples, the  $MDF$  is defined as:  $MDF(i) = MDF(i-1) + DF(i)$ , otherwise the  $MDF$  is defined as:  $MDF(i) = DF(i)$ . It has been used in arrival detection (Stewart, 1977, Veith, 1978; Houlston, Waugh and Laughlin, 1984).

Various picking algorithms were developed based on these functions. The most popular picking algorithm was the short-term-average/long-term-average ratio method (STA/LTA) which was developed by Allen (1978). This method is now widely used for seismic arrival picking (Allen, 1978; Mykkeltveit and Bungum, 1984; Baer and Kradolfer, 1987; Earle and Shearer, 1994). There are also some complex picking methods based on 1-C recordings, such as the sonogram (Joswig, 1990, 1993a, 1993b, 1995, Klumpen and Joswig, 1993) and the autoregressive model (Pisarenko, Kushnir and Savin, 1987; Takanami and Kitagawa, 1988, 1993). Both are computing-intensive because of their complex mathematics.

Of these functions, the absolute value function has the highest fidelity and processing speed and is also objective. Although the other four functions may enhance the changes of amplitude or frequency, they may also introduce some noise or alter

the patterns of the waves and hence they require more time for computing. In addition, ACF, EF, and MDF are not objective due to the parameters of them selected by analysts.

In the *real-time* operation of seismic arrival picking, the computing time should be reduced to minimum. At present this can only be done for large dataset by giving up the complex mathematics. In this thesis, the BPNN has been used to perform the task of feature extraction and pattern recognition of waves, so the input signal to BPNN has been keep as simple as possible by choosing the *absolute value function*.

### 3.3.2 Characteristics from three-component recordings

1-C recordings only provide partial information on the ground motion. Complete information of the ground motion can only be obtained from the three-component (3-C) recordings. Let us consider a ground motion vector  $M(t)$  which represents the position of a point at which the seismometers are located. In the Cartesian coordinate system,  $M(t)$  is projected on three orthogonal axes as  $V(t)$ ,  $E(t)$ ,  $N(t)$  (Figure 3.4). As the point moves, its projections on all three axes vary, splitting the information of the motion into three parts (three 1-C recordings) with each part consisting of incomplete energy and geometrical information. Using them to resolve ground motion requires the synthesis of three components of motion.

This problem can be overcome by using the spherical coordinate system in which  $M(t)$  is represented by  $m(t)$ ,  $\theta(t)$  and  $\phi(t)$  which are three axes of the spherical coordinate system. Here,  $m(t)$  is the *modulus*, the length of this vector,  $\theta(t)$  is the *inclination* of this vector above or below the horizontal plane and  $\phi(t)$  is the *azimuth* of the vector, clockwise from north of the projection of the radius vector on the horizontal (Figure 3.4).

The transformation between the Cartesian coordinate system and spherical coordinate system is reversible. The transformation from  $V(t)$ ,  $E(t)$ ,  $N(t)$  to  $m(t)$ ,  $\theta(t)$ ,  $\phi(t)$  can be written as:

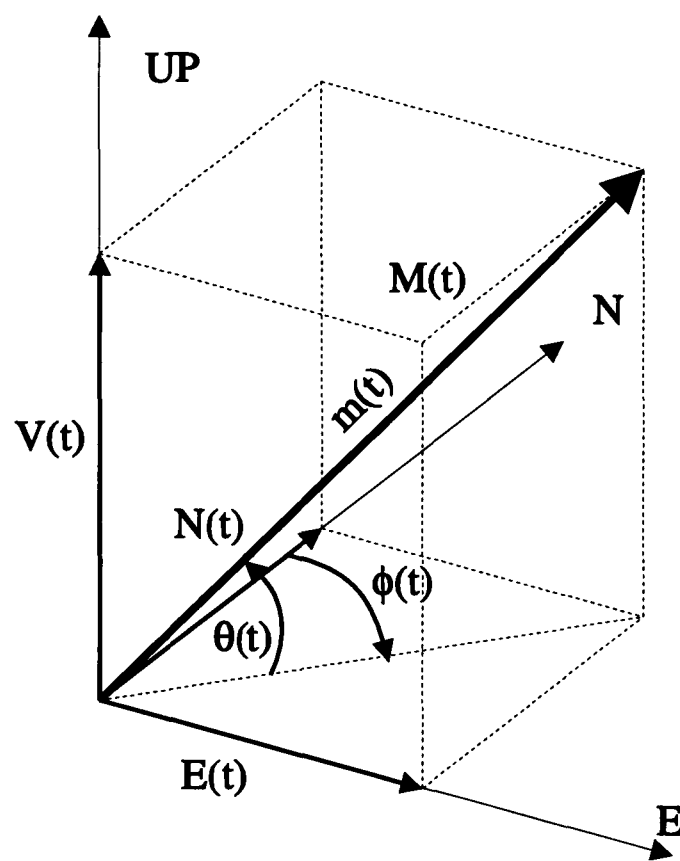


Figure 3.4. A vector  $M(t)$  represented in Cartesian rectangular system and spherical coordinate system.  $V(t)$ ,  $N(t)$  and  $E(t)$  are projection of  $M(t)$  on Cartesian rectangular system.  $m(t)$ ,  $\theta(t)$ , and  $\phi(t)$  are three axes of spherical coordinate system.

$$\left( \begin{array}{l} m(t) = \sqrt{V(t)^2 + E(t)^2 + N(t)^2} \\ \theta(t) = \arcsin\left(\frac{V(t)}{m(t)}\right) \quad [-90^\circ \leq \theta(t) \leq +90^\circ] \\ \phi(t) = \arctan\left(\frac{E(t)}{N(t)}\right) \quad [0^\circ \leq \phi(t) \leq 360^\circ] \end{array} \right. \quad (3.1)$$

and the transformation from  $m(t)$ ,  $\theta(t)$ ,  $\phi(t)$  to  $V(t)$ ,  $E(t)$ ,  $N(t)$  can be written as:

$$\left( \begin{array}{l} V(t) = m(t)\sin(\theta(t)) \\ N(t) = m(t)\cos(\theta(t))\cos(\phi(t)) \\ E(t) = m(t)\cos(\theta(t))\sin(\phi(t)) \end{array} \right. \quad (3.2)$$

The information in the spherical system shows the ground motion more clearly than in the Cartesian system (Lomax and Michelini, 1988; Kracke, 1993). However, it is necessary to remove any constant or long-period, non-zero baselines from the 3-C recordings before applying the spherical coordinate transformation. If this offset is not removed, shift in the values of the spherical coordinates will occur (Lomax and Michelini, 1988).

In the spherical coordinates system, the energy and geometrical information of the ground motion are separated into  $m(t)$  for energy information, and  $\theta(t)$  and  $\phi(t)$  for geometrical information. The  $m(t)$  has advantages over the 1-C recording for the purpose of arrival picking because it is independent of the source orientation and raypath. Figure 3.5 shows an example in which the modulus shows clearer onsets of  $P$ - and  $S$ -arrivals than their original 1-C recordings. Lomax and Michelini (1988) pointed out that the modulus may be particularly useful for automatic  $P$  and  $S$  selection. Kracke (1993) developed an automatic arrival picker using the modulus. Based on this, I will use the modulus in the automatic arrival picking in Chapter 4.

Theoretically,  $\theta(t)$  and  $\phi(t)$  in the spherical coordinate system represent the geometrical information which includes the source orientation or polarization direction. For a wave with a linear or elliptical polarization, its orientation can be read directly from the inclination and azimuth (Lomax and Michelini, 1988). However, in practice, it is difficult to use them in seismic interpretation because they are easily blurred by



Station: DP  
Date: 1984-10-20  
Time: 22h53m01s  
Scale: 356

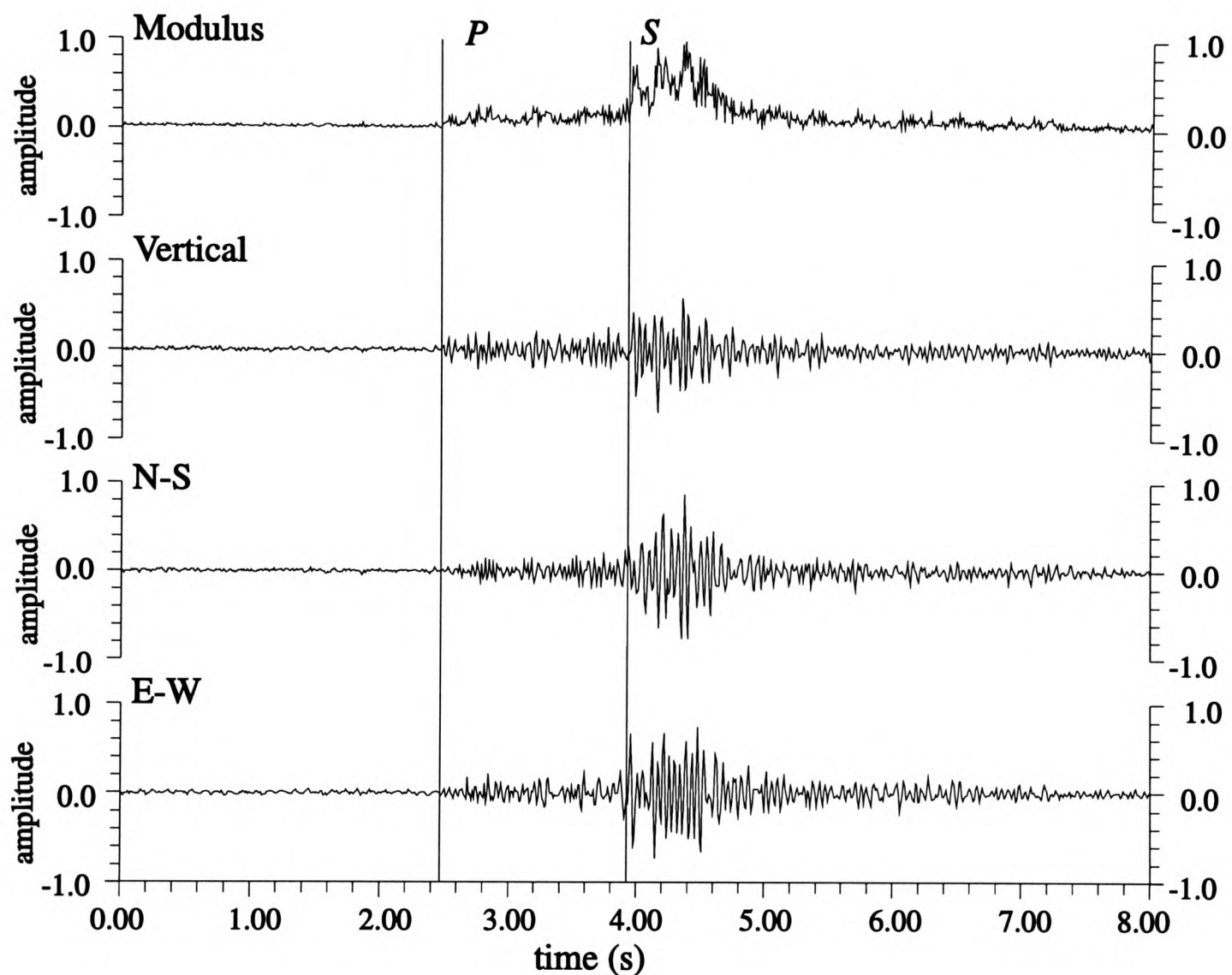


Figure 3.5. A local earthquake recorded on station DP. Its three 1-C recordings and the modulus are displayed. The vertical lines are drawn on the P- and S arrival onsets. The onsets of P- and S-arrivals in the modulus are clearer than in the three 1-C recordings.

background noise and are unstable. Instead, the covariance matrix analysis is widely used in seismic analysis.

### 3.3.3 Covariance matrix analysis.

The covariance matrix of 3-C recordings is a useful measurement of the polarizations of the seismic signals (Samson, 1977; Cichowicz, Green and Brink, 1988; Cichowicz, 1993). The covariance matrix is defined as:

$$C = \begin{Bmatrix} COV(X,X) & COV(X,Y) & COV(X,Z) \\ COV(Y,X) & COV(Y,Y) & COV(Y,Z) \\ COV(Z,X) & COV(Z,Y) & COV(Z,Z) \end{Bmatrix} ; \quad (3.3)$$

where the covariance is measured for  $N$  samples:

$$COV(X,Y) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (3.4)$$

where  $\bar{x}$  and  $\bar{y}$  are the average values of  $x$  and  $y$ , and  $N$  is the length of a window in which the covariances are calculated and can be determined from the predominant frequency of signals (Cichowicz, 1993). However, in the case of a seismic network, it is difficult to calculate  $N$  because the recorded events often have a large variation in frequency. Usually  $N$  has to be chosen after gaining some experience from real data.

The covariance matrix contains all the information needed to characterize the polarization state of a wave. It is a real symmetrical matrix which has three real eigenvalues and can be diagonalized to give the eigenvalues and eigenvectors of its principal axes. Some parameters can then be defined from the eigenvalues and eigenvectors to display the polarization state of the wave (Cichowicz, 1993). For example, the direction of polarization is measured by considering the eigenvector of the largest principal axis. This direction is parallel to the propagation direction for  $P$ -waves and is perpendicular to the propagation direction for  $S$ -waves in an isotropic medium. However, for the purpose of arrival picking and identification, it is difficult

to use the direction as a decision parameter because it is related to the source position which is unknown before the seismic analysis. Ideally, parameters which are independent of the source orientation should be defined to extract the polarization properties for arrival picking and identification.

A useful parameter which is independent of the source orientation is the *degree of polarization* defined from the eigenvalues (Samson, 1977, Cichowicz, 1993):

$$F(t) = \frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2}{2(\lambda_1 + \lambda_2 + \lambda_3)^2} \quad (3.5)$$

where the  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are eigenvalues of the covariance matrix of a moving window of width  $N$  samples. This equation can be written as:

$$F(t) = \frac{3 \text{tr} \mathbf{S}^2 - (\text{tr} \mathbf{S})^2}{2(\text{tr} \mathbf{S})^2} \quad (3.6)$$

where  $\text{tr} \mathbf{S}$ , defined as  $\lambda_1 + \lambda_2 + \lambda_3$ , is the trace of  $\mathbf{C}$ , and  $\text{tr} \mathbf{S}^2$  is defined as  $\lambda_1^2 + \lambda_2^2 + \lambda_3^2$ . This equation shows that the function can be calculated without having to diagonalize the covariance matrix. Mathematically, the trace of a matrix is independent of the rotation of coordinate system, and hence is also independent of the source orientation.

According to this definition, if only one eigenvalue is non-zero, then  $F=1$ , and the wave is linearly polarized; if all of the eigenvalues are equal, then  $F=0$ , and the wave can be considered as completely unpolarized or circularly polarized (Cichowicz, Green and Brink, 1988). Each wave has its own characteristic pattern with time, not just one particular value. The variation of this quantity along the seismogram may indicate the type of a wave. Thus  $F(t)$  enables us to study the evolution of the degree of polarization of a wave.

In order to calculate  $F(t)$ , the window length  $N$  must be determined according to the data features. Computing  $N$  is numerically difficult for real data but it might be visually determined by checking the plotting of  $F(t)$ . Figure 3.6 shows an example of the variation of  $F(t)$  with different window length  $N$ . The basic pattern of  $F(t)$  does not change as the window length varies from 5 to 15. However, the longer the window

Station: DP  
Date: 1984-05-06  
Time: 12h17m27s  
Scale: 846

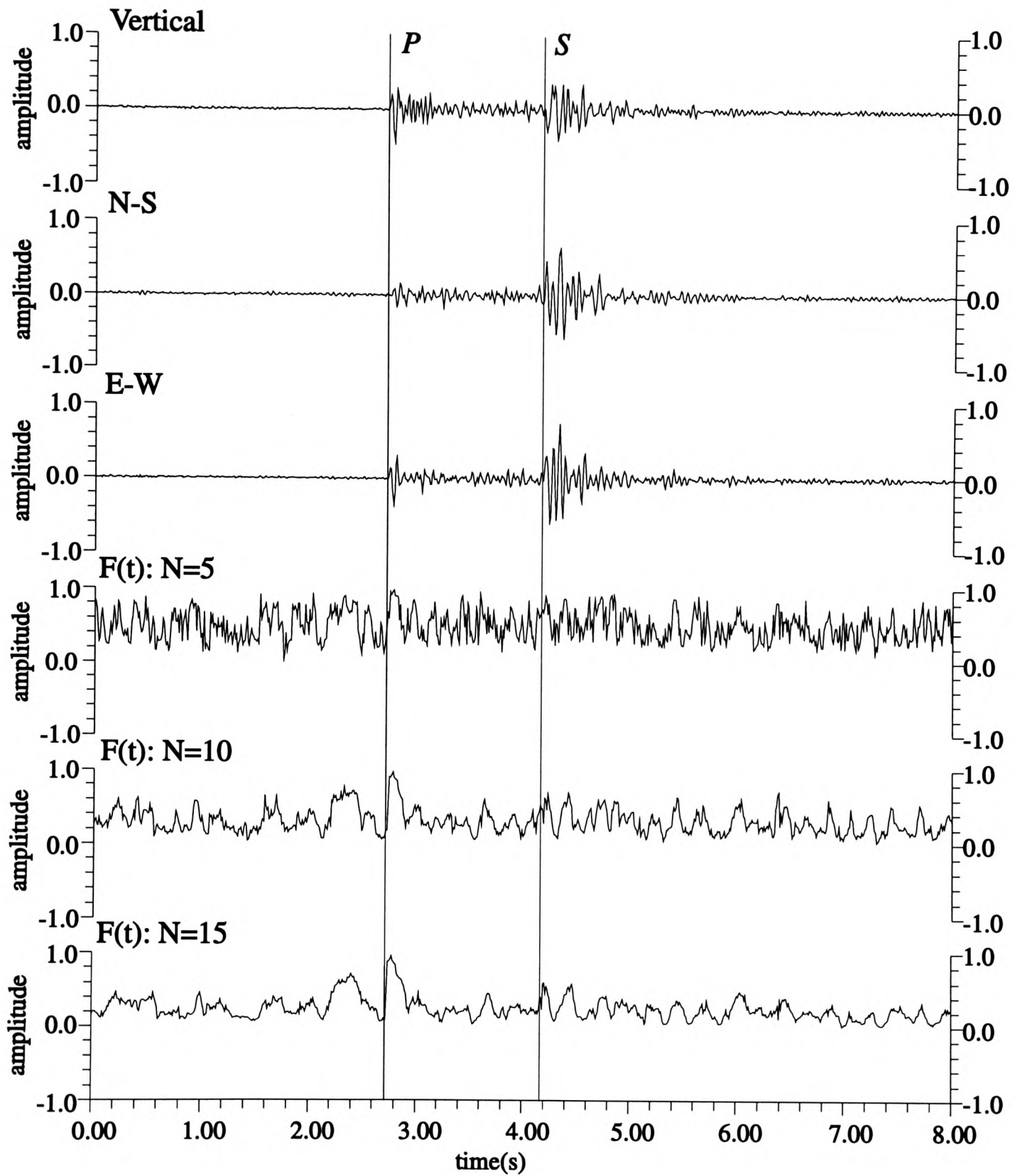


Figure 3.6. The 3-C recordings of a local earthquake and three traces of the degree of polarization  $F(t)$  of 3-C recordings with different window length  $N$ .

length, the smoother the  $F(t)$  is. An optimum number ( $N=10$ ) is obtained for the data used in this study. Note that all 3-C recordings must have the same frequency bandwidth, the same scale and the same noise level. If one of the 3-C recordings has a significantly different property from others, then  $F(t)$  is highly biased, and may give rise to a misleading interpretation.

I have investigated the  $F(t)$  patterns for data from the TDP3 network. Usually, most  $P$ -arrivals have high values of  $F(t)$ , most  $S$ -arrivals have medium values of  $F(t)$  and most noise bursts have low values of  $F(t)$  (Figure. 3.7). That means the  $P$ -arrivals have well-defined linear polarization patterns, while  $S$ -arrivals and noise bursts do not. The polarization patterns of seismic arrivals are complex. Most of the  $F(t)$  patterns of  $P$ -arrivals usually differ from those of  $S$ -arrivals and noise bursts. However, there are also some noise bursts whose  $F(t)$  patterns are similar to those of the seismic arrivals. Spikes manifest themselves as special patterns in which  $F(t)$  is very high, near unity, for a window length of *ten* samples (Figure 3.8). However, they can be easily discarded by using a conventional program. Comparing the data from stations DP and AY, the  $F(t)$  patterns are different even for the same arrivals from the same earthquake (Figure 3.9). It is difficult to find a simple method such as a threshold value to distinguish their type. Identifying them requires the intensive use of the pattern recognition technique.

### 3.4 SUMMARY

The data from the TDP3 local earthquake network in Turkey were examined and selected for the purpose of developing the methods of arrival picking and identification. Only a relatively "small" dataset from two stations, DP and AY, are presented as examples. Various characteristic functions have been inspected for the purpose of arrival picking and identification. Here, three characteristic functions, which are independent of the source orientation, were selected in this study:

Station: DP  
 Date: 1984-05-06  
 Time: 12h16m08s  
 Scale: 564

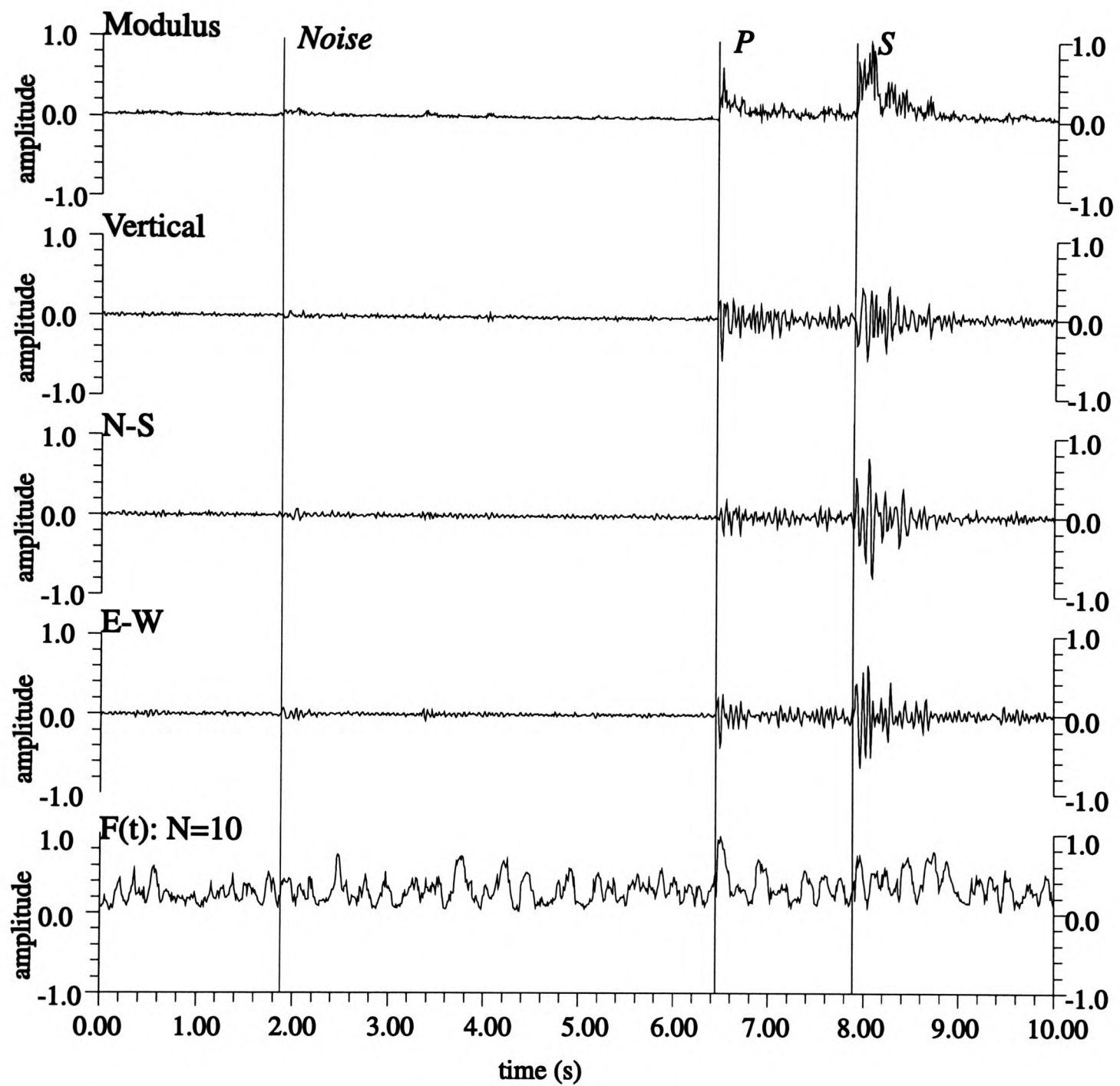


Figure 3.7. The 3-C recording of a local earthquake, its modulus and its degree of polarization  $F(t)$ . Three vertical lines show the arrival onsets which are identified as a noise burst with low value of  $F(t)$ , a P-arrival with high value of  $F(t)$  and a S-arrival with middle value of  $F(t)$ .

Station: DP  
Date: 1984-05-17  
Time: 12h13m33  
Scale: 1958

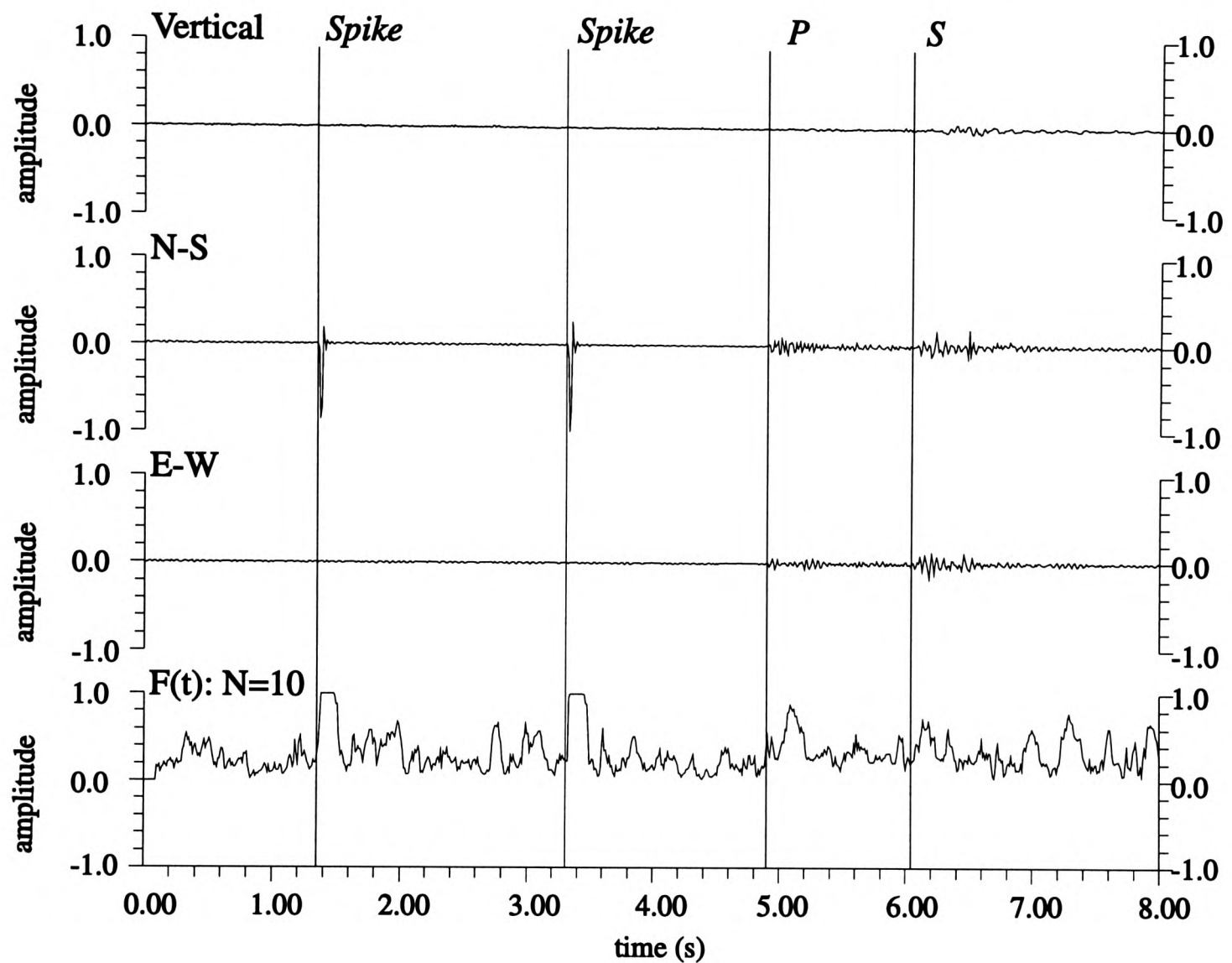
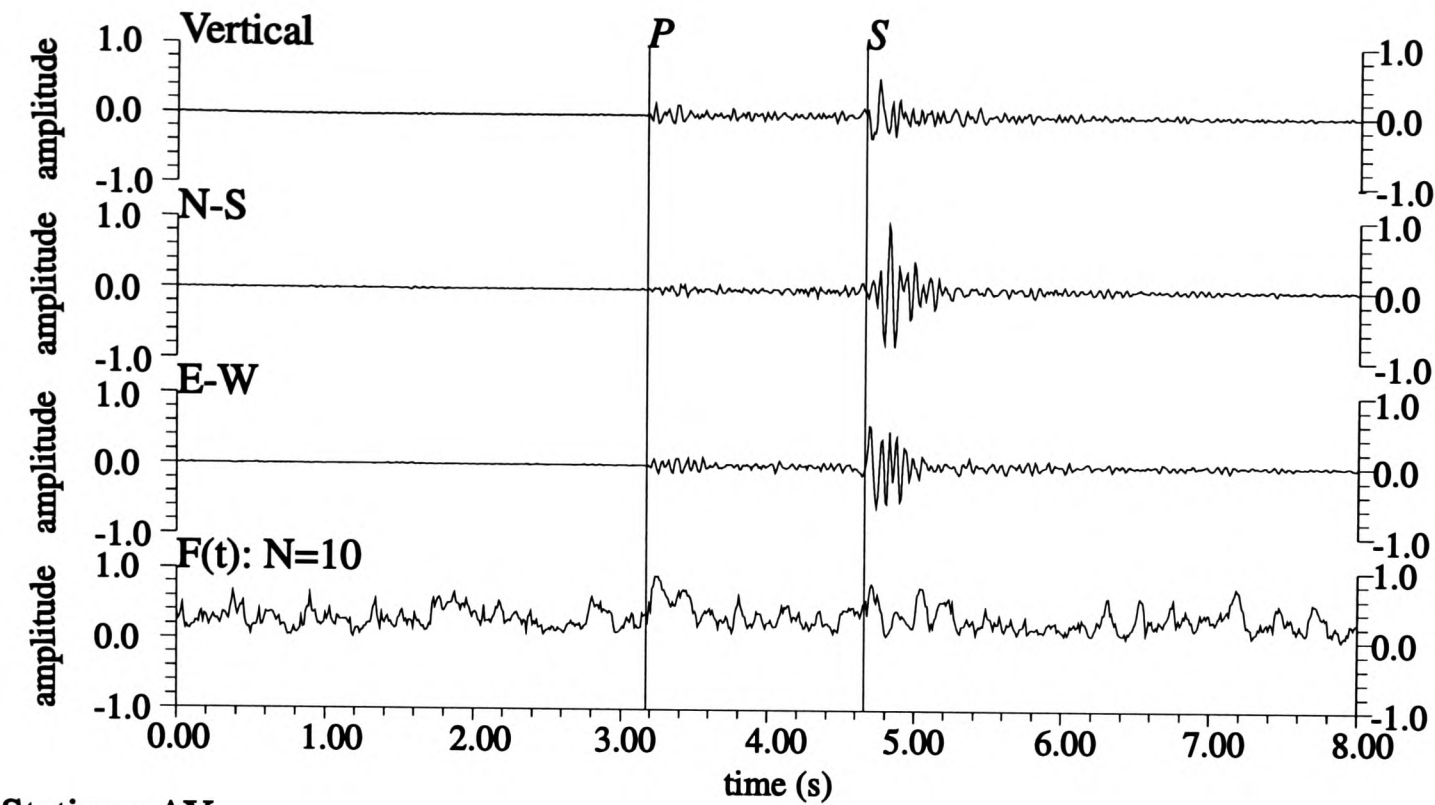


Figure 3.8. The 3-C recording of a local earthquake and its degree of polarization ( $F(t)$ ). Four vertical lines shows arrival onset of two spikes and a P-arrival and a S-arrival. Spikes have unique  $F(t)$  patterns whose values are near unity.

Station : DP  
Date: 1984-05-03  
Time: 16h25m26s  
Scale: 1386



Station : AY  
Date: 1984-05-03  
Time: 16h25m26s  
Scale: 379

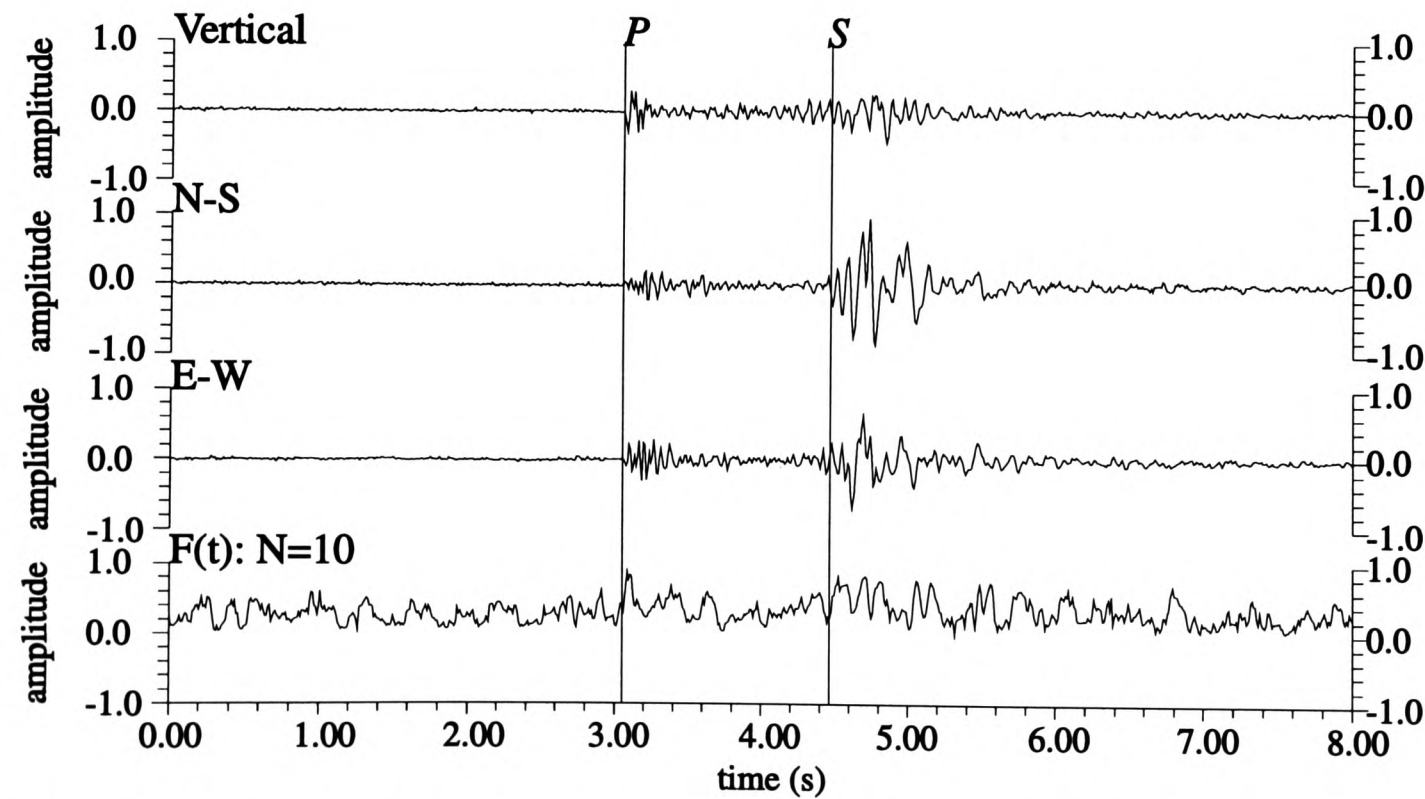


Figure 3.9. 3-C recordings and the degree of polarization  $F(t)$  of a local earthquake recorded on stations DP (above) and AY (below) respectively. The patterns of the degree of polarization are different for the same P- and S-arrivals from the two stations.



- (a) The *modulus* of 3-C recordings will be used to picking arrivals from 3-C recording in Chapter 4.
- (b) The *absolute value function* of 1-C recording will be used for picking arrivals from 1-C recordings in Chapter 5. I should point out that all functions of 1-C recordings are influenced by the source orientation. In this case, it is not possible to obtain a function which is independent of the station position with respect to source radiation pattern.
- (c) The *degree of polarization* of seismic waves from the covariance matrix of 3-C recordings will be used in the approach of arrival identification in Chapter 6.

## CHAPTER 4:

# ARRIVAL PICKING FROM THREE-COMPONENT RECORDINGS USING BPNN

### 4.1 INTRODUCTION

As I mentioned in Chapter 1, the task of estimating arrival times for  $P$  and  $S$  waves found in recordings of earthquake events forms an important foundation for schemes employing automatic processing for event location, event identification, source mechanism analysis, and spectral analysis. The traditional picking method is that a trained analyst visually checks the seismograms and picks out  $P$ - and  $S$ -arrivals according to his individual experience. This task is time-consuming and subjective. With the increase in the number of digital seismic networks being established worldwide, there is a pressing need to provide a more reliable and robust alternative, which is less time-consuming and more objective. A great deal of effort, stretching back several decades, has therefore been devoted to the automation of arrival picking, and many different varieties of algorithm exist. However they do tend to be data-specific and are not generally available. The goal of global automation is far from achieved and as a consequence such elementary seismogram interpretation still forms a bottleneck in the routine work of many observatories.

In this chapter, I attempt to use a BPNN to tackle this problem and to test its ability of automatic detecting and picking seismic arrivals. Specifically the BPNN approach is developed to deal with the data from the TDP3 local earthquake network. This is achieved by utilizing the vector modulus of 3-C recordings as the input of the BPNN. A BPNN trained by a small but representative training dataset can quickly process a large amount of data. The arrival onset times can be defined by a discriminant function, determined from the output of the trained BPNN. In conjunction with this BPNN, some post-processing procedures are used to improve the performance.

## 4.2 BRIEF HISTORY OF AUTOMATIC ARRIVAL PICKING

According to Blandford's review (1982), the first careful discussion of seismic detection is found in a report by Vanderkulk, Rosen and Lorenz (1965) and the next advances in detection theory were made by Lacoss (1972) and Wirth, Blandford and Shumway (1976). Following them, some methods for automatic arrival picking of seismic data were developed by Ambuter and Solomon (1974), Crampin and Fyfe (1974), Stevenson (1976) and Stewart (1977). Later, Allen (1978) developed his famous STA/LTA (short-term average/long-term average) ratio method to pick *P*-arrivals. This is the most popular method for arrival picking and many analysis systems are based on it (Mykkeltveit and Bungum, 1984; Bibbo, Etter and Breiding, 1991; Tarvainen, 1992; Earle and Shearer, 1994). Some of its variants have been implemented. For example, Baer and Kradolfer (1987) used an envelope function of each 1-C recording in this algorithm and then passed it through a nonlinear amplifier. The resulting signal is then subjected to a statistical analysis to yield *P*-arrival times and a measure of reliability for the picking. Kvaerna and Ringdal (1992) applied the STA/LTA to coherent and incoherent beams of an A-ring microarray data to detect arrivals. Ruud and Husebye (1992) developed a 3-C detector based on the STA/LTA method.

Other methods were also developed. For example, Houlston, Waugh and Laughlin (1984) implemented the Stewart method (1977) in their detection system. Pisarenko, Kushnir and Savin (1987) developed an optimal *P*-arrival picker using an asymptotic approximation of the likelihood function. Magotra, Ahmed and Chael (1987) developed an arrival detector based on the polarization analysis of the horizontal (N-S and E-W) recordings. Roberts, Christoffersson and Cassidy (1989) made an assessment of whether data are consistent with the arrival of a *P*-arrival or a linearly polarized *S*-arrival using the auto- and cross-correlations of 3-C recordings. Takanami and Kitagawa (1988, 1993) developed a method of picking *P*- and *S*-arrivals by fitting a locally stationary autoregressive model. Kracke (1993) developed a simple

method based on the displacement vector of a seismic trace in a spherical coordinate system for *P*-arrivals. Cichowicz (1993) developed an *S*-arrival picker based on a filter which combines polarization and energy ratios. Joswig and Schulte-Theis (1993) used a master-event-correlation method to detect *P*-arrivals in weak local earthquake records. All these are conventional methods which incorporate fixed algorithms to solve particular problems. They quantify some attributes of the seismic trace and use them as the basis of the decision. However, *they are not adaptive*. They work well under certain conditions, but quite often do not produce good results.

The application of artificial intelligence (AI) methods to arrival picking is a relatively recent development. Various methods have been applied to the interpretation of seismic signals such as knowledge-based systems based on the blackboard method used by Chiaruttini, Roberto and Saitta (1989), and later developed by Chiaruttini and Salemi (1993). Bache *et al.* (1990) developed an intelligent monitoring system based on this knowledge-based system, database management systems and signal processing. As an alternative strategy, Joswig (1990, 1993a, 1993b, 1995) developed a pattern recognition approach for *P*-arrivals using a sonogram. Klumpen and Joswig (1993) used this pattern recognition technique to estimate *P*- and *S*-arrival onset times by identifying generic polarization patterns from a time-frequency image of the principal value decomposition of the correlation matrix of 3-C recordings. The AI methods are quite different from the conventional methods. In these methods, some samples of the seismic arrivals are used as the references to compare with the real data. Rules based on the knowledge of the seismic arrival and noise control the analysis procedure which analyses the *mental images*, a two dimensional grey-scale picture of spectral energy, of the seismic arrivals extracted from seismograms by some numerical methods. Although these AI methods have advantages over conventional ones by putting the analyst's experience in the analysis procedure, they still need complex mathematics to extract features from seismograms.

BPNNs and other ANNs, another group of techniques from the area of AI, provide a natural alternative to this type of earthquake analysis. They had already been

proven useful in handling complicated pattern recognition problems in other applications (see Chapter 2). There are also a few approaches in the literature which used the BPNN method to pick the first-break of surface seismic data (Murat and Rudman 1992; McCormack, Zaucha and Dushek 1993) and detect the arrivals of earthquakes (Wang and Teng, 1995). As the BPNNs utilize a learning scheme to develop an appropriate solution, they are flexible and adaptive to different dataset.

### 4.3 APPROACH FOR ARRIVAL DETECTION AND PICKING

The tasks performed by the trained analyst in manually picking arrival onset times involve an intensive amount of pattern recognition. Experience provides a judicious balancing of wave characteristics such as amplitude, frequency and polarization from previous records to determine the most likely onset time. If questioned about a particular decision, however, the analyst may offer a few rules for guidance but can often give no obvious systematic reasoning because the decision is partly subjective and based upon his past experience. Consequently, different trained analysts may give different answers, and the same analyst may even choose a different interpretation after some time has elapsed. In this chapter, I attempt to use an BPNN approach to emulate this manual analysis procedure but overcome its weaknesses. The BPNN is designed to be similar in operation to an analyst, and is trained by presenting many different seismic arrivals. After training is accomplished, the BPNN can remember these arrivals and then should be able to recognize new arrivals from a variety of new seismograms. Its great advantages are higher objectivity and higher speed.

As mentioned in Chapter 3, a method to pick arrivals from a seismogram can be regarded as using a filter to act on a characteristic function of the seismogram. Here the BPNN is used as a filter to act on the modulus of 3-C recordings. This approach includes two phases: the *training (learning)* phase and the *testing (generating)* phase. In the training phase, some training dataset should, first, be selected by an analyst using his experience and then be used to train the BPNN. After training, the BPNN is applied to the entire seismogram by using a window which

slides along the entire modulus trace (Figure 4.1). The output of the BPNN then yields a time series which is interrogated for a decision regarding the seismic arrivals.

#### 4.3.1 Input characteristics of data

The characteristic function of seismograms for the detection and picking of arrivals in this approach is the vector modulus of the 3-C recordings already discussed in Chapter 3. This is useful as the signals in 1-C recordings are strongly dependent on the source position and ray direction, which may otherwise give rise to a misleading interpretation. The instantaneous vector modulus  $M(t)$ , calculated at each individual 3-C sample along the traces, separates the geometric dependency from the seismic vector motion whilst retaining its energy characteristics for picking.  $M(t)$  is then directly fed into a BPNN. It is believed that this attribute facilitates an easier identification, regardless of the polarization of the wave (Lomax and Michelini, 1988). I do not use the polarization properties of individual arrivals for *picking* as I believe they may not provide a satisfactory indicator due to such factors as: phase changes during propagation, fine-structure of the waveforms such as that due to shear-wave splitting, and directional dependency. The overriding concern is the uncertain applicability of such a parametric model of the wavefield in a heterogeneous crust (Der, Baumgardt and Shumway 1993). The polarization properties will be used for arrival *identification* in Chapter 6.

$M(t)$  is presented to the BPNN in segments which are selected from a sliding window passing across the entire 3-C seismogram. The modulus is strongly dependent on the magnitude and epicentral distance of an earthquake. For the data from TDP3, the maximum modulus of an event is more than 3600 counts in scale and the minimum modulus of an event is just above the background noise, about 20-30 counts. If such a big range of moduli are directly fed into the BPNN for training, a large training dataset is required to cover this range, with a consequent increase in the training time and a larger BPNN structure. Otherwise large changes in modulus may bias the estimates. This problem is overcome by using the *relative modulus* in which

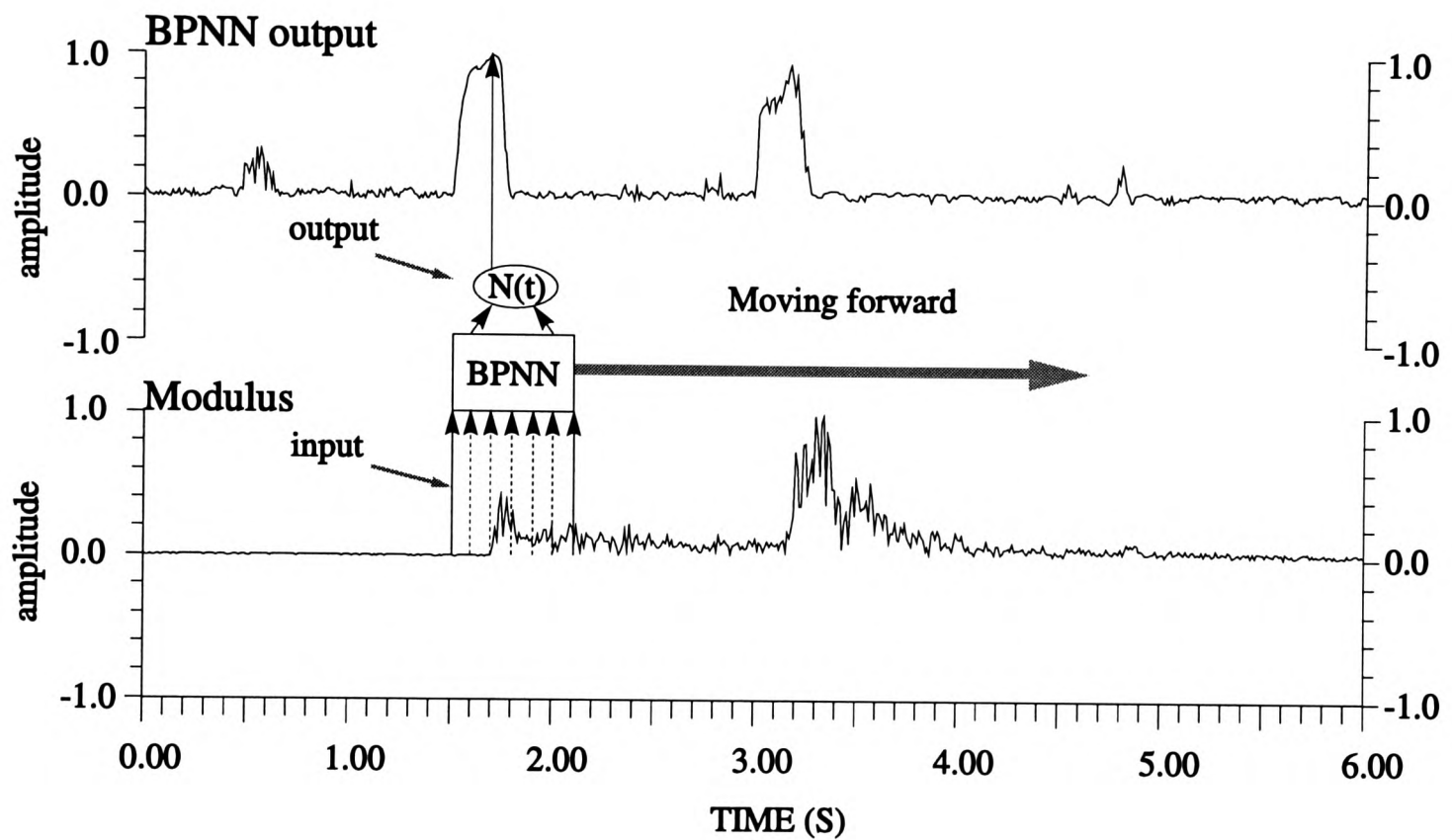


Figure 4.1. Schematic diagram showing the method of using a BPNN to pick seismic arrivals. Here, the input of the BPNN is the modulus of 3-C recordings (the lower diagram). A trained BPNN is treated as a filter in a sliding window across though the entire modulus trace. The output of the BPNN (the upper diagram) yields a time series which enhances the changes in the modulus to indicate the arrival onsets.

each segment is individually normalized so that it is not dependent upon the magnitude and distance of an earthquake. This reduces the number of training examples needed.

Use of the relative modulus means that the BPNN is forced to sense the relative amplitude and frequency content of the signals, and use them to detect the onset. For high SNR, the onset is characterized by a distinct change in the amplitude of the seismogram. However, if the SNR is low, the major discriminating factor is a frequency change due to the different spectra of the background signal and earthquake signal. The BPNN is similar to a sophisticated wavelet transform which enhances these changes informing its output.

The data used in this study are local earthquake events recorded at stations DP and AY in the TDP3 seismic network which was introduced in Chapter 3. For the first test of the performance of this approach, I will apply it to a small subset of these data. This dataset consists of 210 high quality events recorded at stations DP (120 events) and AY (90 events). In the second test, I will use this trained BPNN to process the complete dataset of 762 recordings at stations DP and AY in the TDP3 seismic network, with a mixture of good and bad data.

### 4.3.2 BPNN structure

The BPNN used in this chapter has three layers, including an input layer with 30 nodes, a hidden layer with 10 nodes and an output layer with two nodes (Figure 4.2). The 30 input nodes correspond to the input segment of modulus which is chosen to include several complete cycles of a wave in a sliding window with length fixed at 290ms (30 samples).

The two output nodes flag the result, (0,1) for an arrival and (1,0) for pure background noise. It seems that one output node can also flag the two states, 1 for arrivals and 0 for pure background noise. However, if the output states are more than two, one output node cannot properly flag these output states. For this kind of pattern recognition, the output node should equal to the desired output states (McCormack, 1991; Haykin, 1994). In order to be consistent with this, most authors prefer two



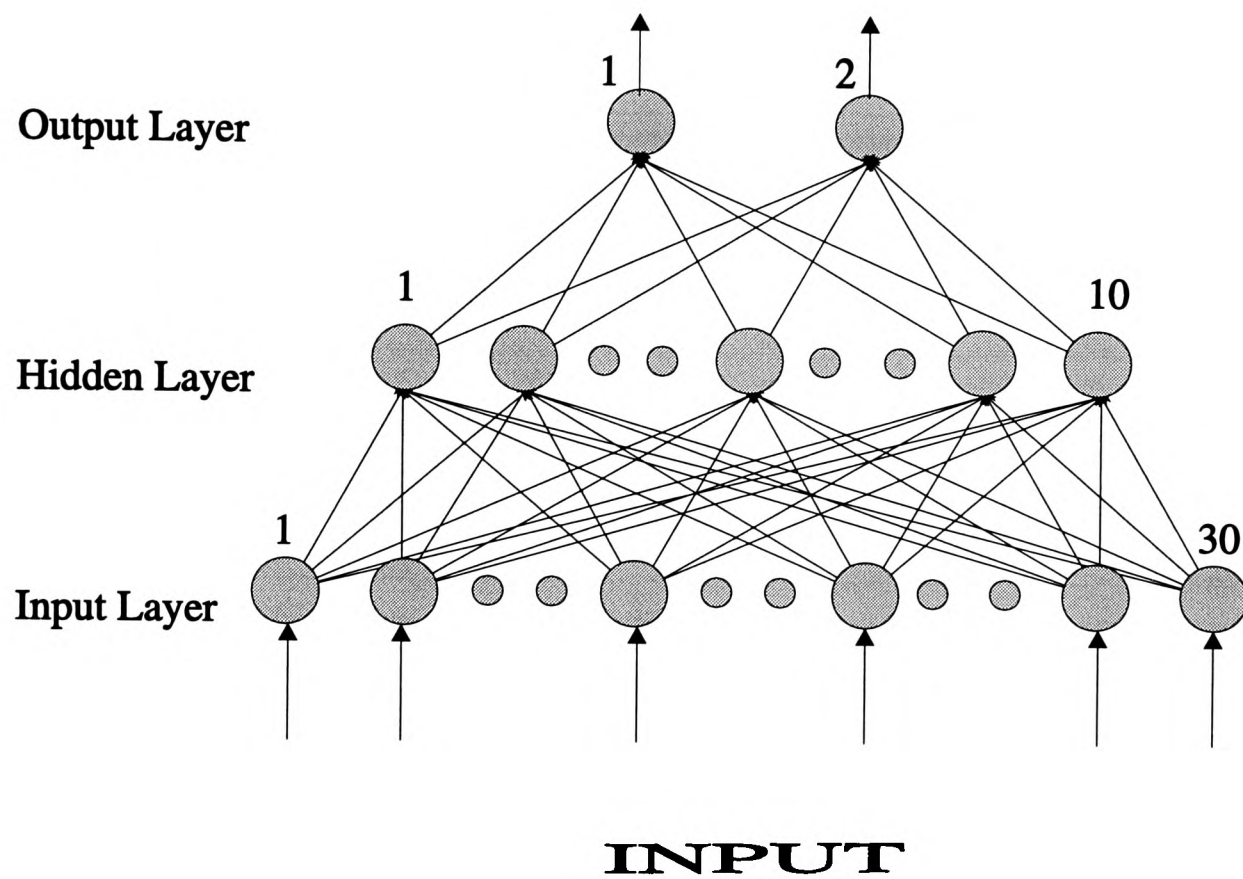


Figure 4.2. The structure of a BPNN for seismic arrival picking. It has three layers including an input layer with 30 nodes, a hidden layer with 10 nodes and an output layer with two nodes. The BPNN input is a segment of the modulus of the 3-C seismogram. The two output nodes indicate the input segment with (0,1) as a P-arrival and (1,0) as noise.

output nodes for the problem with two desired output states (Dowla *et al.*, 1990; Gorman, 1988; McCormack, 1993; Penn, 1993). I also use two output nodes in the BPNN used in this chapter.

The ten hidden nodes were finally chosen after a process of trial and error with different training runs. Although this solution is considered optimum for the current application, further architecture optimization could undoubtedly be achieved by a more exhaustive search procedure on a more powerful computer.

### 4.3.3 Training procedure

#### 4.3.3.1 Selecting training dataset

The function of this primary BPNN picker is to pick as many changes of  $M(t)$  as possible and discard those which are neither  $P$ - nor  $S$ -arrivals. The most important task in training a BPNN is the selection of suitable training dataset according to the purpose for which the BPNN is used because its performance depends on the training dataset. If incorrect or inconsistent data are used to train it, it cannot be expected to give a correct answer for new data. The same BPNN can be applied to a different problem with different training dataset. In this approach, this BPNN is employed to pick seismic arrivals from the background noise, so the training data should include the seismic arrival signal and background noise signal. Although the relative modulus  $M(t)$  is independent of the source position, the modulus patterns of seismic arrivals are still varied. Different patterns are required to train the BPNN. For training, the  $M(t)$  segments include either the pure background noise or the  $P$ -arrival with some early background noise. The  $P$ -arrival training segments are chosen to include wavelets with different characteristics.  $S$ -arrivals are not included in the training dataset because their  $M(t)$  appears to exhibit similar characteristic to  $P$ -arrivals. Background noise segments are extracted prior to the  $P$ -arrivals in the same seismograms. These segments are arranged so that the predicted onset time of every  $P$ -arrival lies at the eleventh sample. The BPNN then outputs (0,1) on its output nodes to flag a  $P$ -arrival. This behaviour

is imprinted on every training example. Figure 4.3 shows nine pairs of  $P$ -arrival and background noise segments for training in the study.

The number of training patterns required is determined by the nature of the problem and the anticipated performance of a trained BPNN. Although some rules give guidance on this (Baum and Haussler, 1989, Faussett, 1994), experience still plays an important role in selecting the suitable number. Too small a number may result in a poorly trained BPNN, and too large a number may result in the learning procedure becoming too long. It seems better to train a BPNN with a small training dataset first, and then to improve its performance by subsequently adding more training dataset. In order to balance the performance of a trained BPNN for  $P$ -arrivals and background noise, the same number of  $P$ -arrivals and background noise segments are required to train the BPNN. In the first experiment, I use seven pairs of  $P$ -arrivals and background noise segments (Figure 4.3) for training.

#### 4.3.3.2 Selecting training parameter and initial weights

In the training procedure, the second task is to select the training parameters of the *generalized delta rule*, which include: *learning rate*  $\eta$ , *momentum rate*  $\alpha$ , *system error threshold and pattern error threshold* and initialization of the weights and thresholds of the BPNN. In Section 2.3.4, I have mentioned that these parameters and the initial weights and thresholds are somewhat undetermined and their values depend on the nature of the problem. Here I attempt to determine them for the arrival picking problem by training the special BPNN whose structure is defined in Section 4.3.2 with the training dataset (seven pairs of  $P$ -arrival and background noise segments) defined in Section 4.3.3.1

##### (a) Determining training rate and momentum rate

In order to achieve this, two questions should be answered: (1) How do they affect the training procedure? and (2) How do they affect the BPNN structure? The training procedure can be measured using the *iteration number* for convergence. The

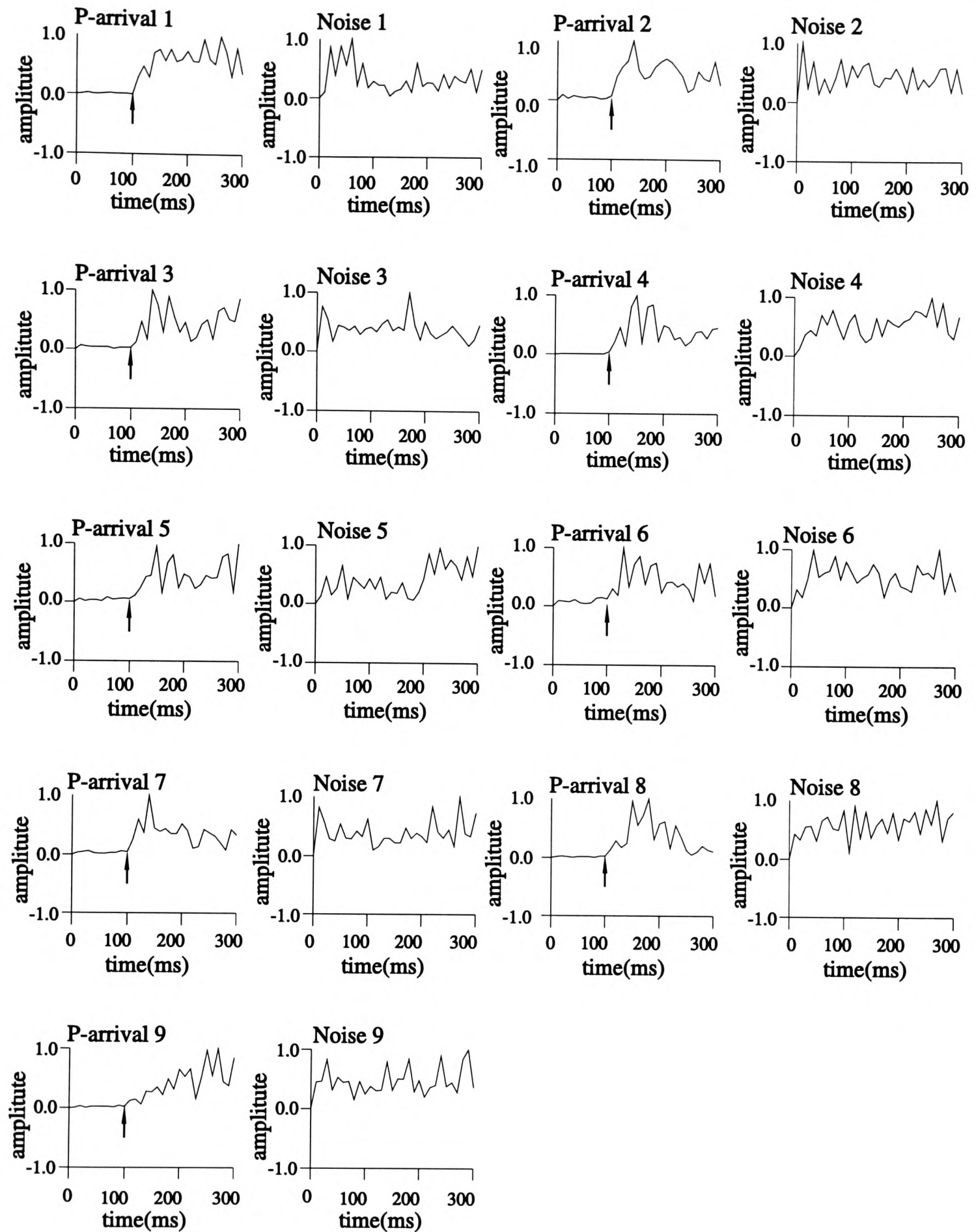


Figure 4.3. Nine P-arrival segments and nine background noise segments used for training a BPNN. Noise segments are extracted prior to the P-arrivals in the same seismograms. Arrows on P-arrival segments indicate arrival onset times used to train the BPNN, all are at the eleventh sample. These segments are individually normalized before being input into the BPNN.

trained BPNN structure can be measured by its *weight pattern* which displays the distribution of weight values (Dai and MacBeth, 1994c).

Table 4.1.A shows the iteration number required in training as a function of  $\eta$  and  $\alpha$ , but with the same system error threshold (0.00001) and pattern error threshold (0.0001) which are used to stop the training procedure. A relationship between the iteration number and  $\eta$  and  $\alpha$  can be obtained from this table:

$$\text{Iteration\_Number} = IN_0 * \frac{(1-\alpha)}{10*\eta} \quad (4.1)$$

where  $IN_0$  is the iteration number at  $\eta = 0.1$  and  $\alpha = 0.0$ . In order to verify this equation, I calculate the iteration number by using this equation with the same  $IN_0$ . Table 4.1.B is the calculated result which fits well with the data in Table 4.1.A. For example, the maximum relative error is 10% at  $\alpha = 0.9$  and  $\eta = 0.9$  with an absolute error of 8 iterations and the maximum absolute error is 31 iterations at  $\alpha = 0.0$  and  $\eta = 0.9$  and 1.0 with a relative error of 4%.

This equation shows that: (1) The iteration number is inversely proportional to  $\alpha$ , but  $\alpha$  must be less than 1.0, otherwise the learning procedure does not converge. (2) The iteration number is the reciprocal of the  $\eta$ , and naturally,  $\eta$  cannot be zero. Due to the nature of the reciprocal function, large  $\eta$  cannot reduce the iteration number too much. For example, when  $\eta$  is increased from 0.1 to 0.2, the iteration number is reduced by 50%, but when  $\eta$  is increased from 0.7 to 0.8, the iteration number is reduced by 12.5%. The relative decreasing rate is independent of  $\alpha$ . Because the learning procedure might be unstable and the weights of BPNN might trip in a local minimum by using a too large  $\eta$ , (Rumelhart, Hinton and Williams, 1988), I will be following other authors' suggestion that  $\eta = 0.7$  and  $\alpha = 0.9$  (McClelland and Rumelhart, 1988; Pao, 1988; Demuth and Beale, 1993).

Equation 4.1 might be used as a guide to check the convergence or performance of training a BPNN. In the case of using large  $\eta$  and  $\alpha$  to train a BPNN, the iteration number should not be too large. If the iteration number larger than the predicted value for convergence, it seems that either the BPNN is not well designed

**Table 4.1.A** The iteration number in training with different learning rate  $\eta$  and momentum rate  $\alpha$ . (NC means non-convergence)

Iteration number		$\alpha$										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\eta$	0.1	7248	6523	5799	5074	4350	3625	2900	2175	1451	730	NC
	0.2	3634	3274	2910	2547	2183	1819	1455	1091	728	368	NC
	0.3	2435	2191	1948	1705	1462	1218	974	729	486	247	NC
	0.4	1834	1651	1468	1285	1102	918	734	550	366	188	NC
	0.5	1474	1327	1180	1033	886	739	591	443	295	151	NC
	0.6	1234	1111	989	866	743	620	496	372	248	124	NC
	0.7	1063	957	852	746	641	535	428	321	215	102	NC
	0.8	935	842	750	657	564	471	378	284	190	85	NC
	0.9	836	753	670	588	505	422	338	254	171	73	NC
	1.0	756	681	606	532	457	382	307	231	156	67	NC
	1.1	691	622	554	486	418	350	281	212	143	65	NC

**Table 4.1.B** The iteration number calculated using equation 4.1

Iteration number		$\alpha$										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\eta$	0.1	7248	6523	5798	5074	4349	3624	2899	2174	1450	725	---
	0.2	3624	3262	2899	2537	2175	1812	1449	1087	725	363	---
	0.3	2416	2174	1932	1691	1450	1208	966	725	483	242	---
	0.4	1812	1631	1449	1269	1087	906	725	544	363	181	---
	0.5	1450	1304	1159	1015	870	725	580	435	290	145	---
	0.6	1208	1087	966	846	725	604	483	362	242	121	---
	0.7	1035	931	828	724	621	518	414	310	207	104	---
	0.8	906	815	725	634	543	453	362	272	181	91	---
	0.9	805	725	644	564	483	403	322	242	161	81	---
	1.0	725	652	580	507	435	362	290	217	145	73	---
	1.1	659	593	527	461	395	329	264	197	131	66	---

or the training dataset is not well selected.

The above analysis shows that these parameters greatly affect the convergence speed of the learning procedure. However, do they affect the trained BPNN structure? Here a weight map of the BPNN structure is produced to show the BPNN weight patterns. Figure 4.4 shows the weight maps of three BPNNs trained with different  $\eta$  and  $\alpha$  values. These maps show the magnitudes of each weight connecting to the different layers in the BPNN. In each map, solid circles are shaded on a grey scale corresponding to a magnitude range indicated by the key (I will explain the details of this weight map in Section 4.6.1). These maps show that the BPNNs trained with different  $\eta$  and  $\alpha$  have similar weight patterns. For example, the weight map between the input nodes and hidden nodes can be divided into two portions at the tenth input nodes. Weights connected to nodes between the first to tenth nodes show a "*high contrast*" pattern, in which most of the weights have large absolute values. Weights connected to nodes between the eleventh and thirtieth nodes show a "*low contrast*" pattern in which most of the weights have small absolute values. Although a few weights show some changes as  $\eta$  and  $\alpha$  change, the whole pattern is similar. Therefore, these trained BPNNs with similar weight patterns should have similar performances.

#### **(b) Determining system error threshold and pattern error threshold**

In order to determine the system error threshold and pattern error threshold, I train the same BPNN with the same training dataset and the same values of  $\alpha$  ( $=0.9$ ) and  $\eta$  ( $=0.7$ ), but with different system error and pattern error. The smaller the errors are, the larger the iteration number required. Figure 4.5 shows two examples of weight patterns of a trained BPNN. These figures show that changing the system error threshold and pattern error threshold does not affect the weight pattern too much. However, the lower error threshold (or longer training) might result in an *overtraining* phenomenon (Haykin, 1994; Prechelt, 1995) in which if a BPNN is overtrained, its generalization gets worse instead of better after a certain point during training. This is because such long training may make the BPNN "memorize" the training patterns,

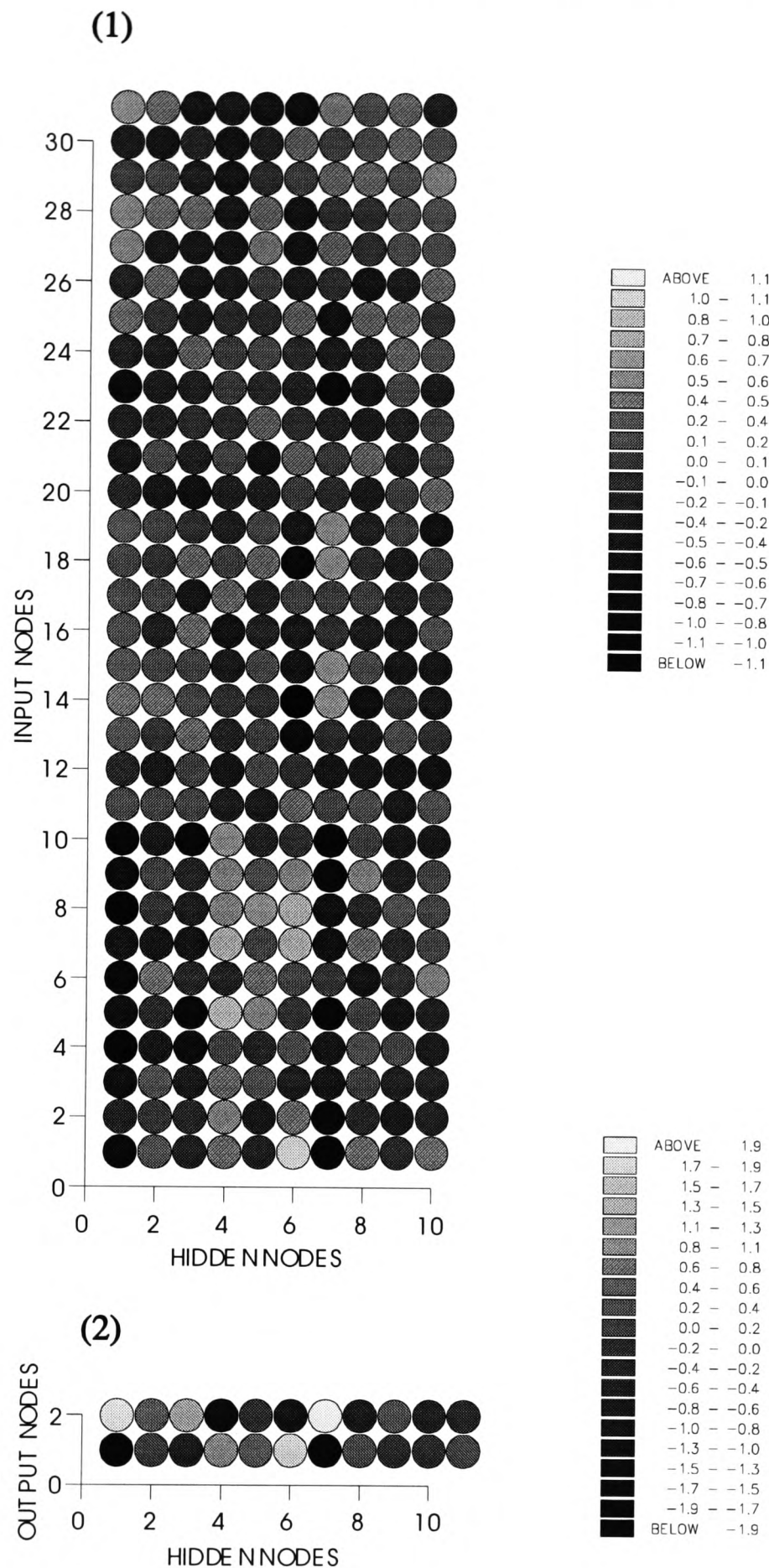


Figure 4.4 (a). Weight map for a trained BPNN (see Section 4.6.1 for a full explanation of the technique). (1) The mapping of weights between the input layer and the hidden layer. (2) The mapping between the hidden layer and the output layer. Shaded circles represent the weights and the contrast in each circle indicates the value of a weight. The top row of circles in (1) and the right row of circles in (2) represent the node thresholds. This BPNN is trained with seven pairs of P-arrival and noise segments. The training parameters are: learning rate = 0.1, momentum rate = 0.0, system error threshold = 0.00001 and pattern error threshold = 0.0001.



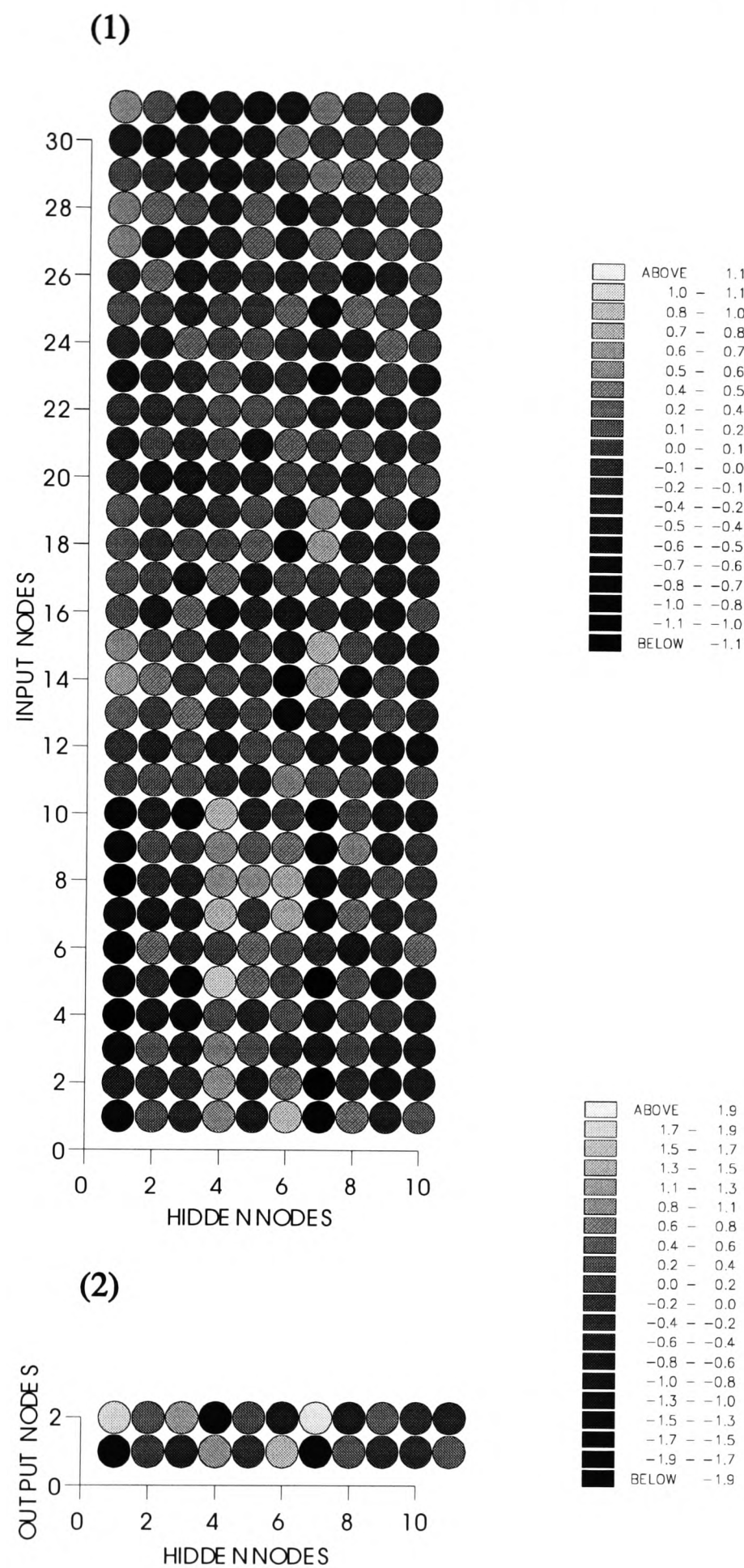


Figure 4.4 (b). Weight map for a trained BPNN. Notation as in Figure 4.4 (a) This BPNN is trained with seven pairs of P-arrival and noise segments. The training parameters are: learning rate = 0.7, momentum rate = 0.9, system error threshold = 0.00001 and pattern error threshold = 0.0001

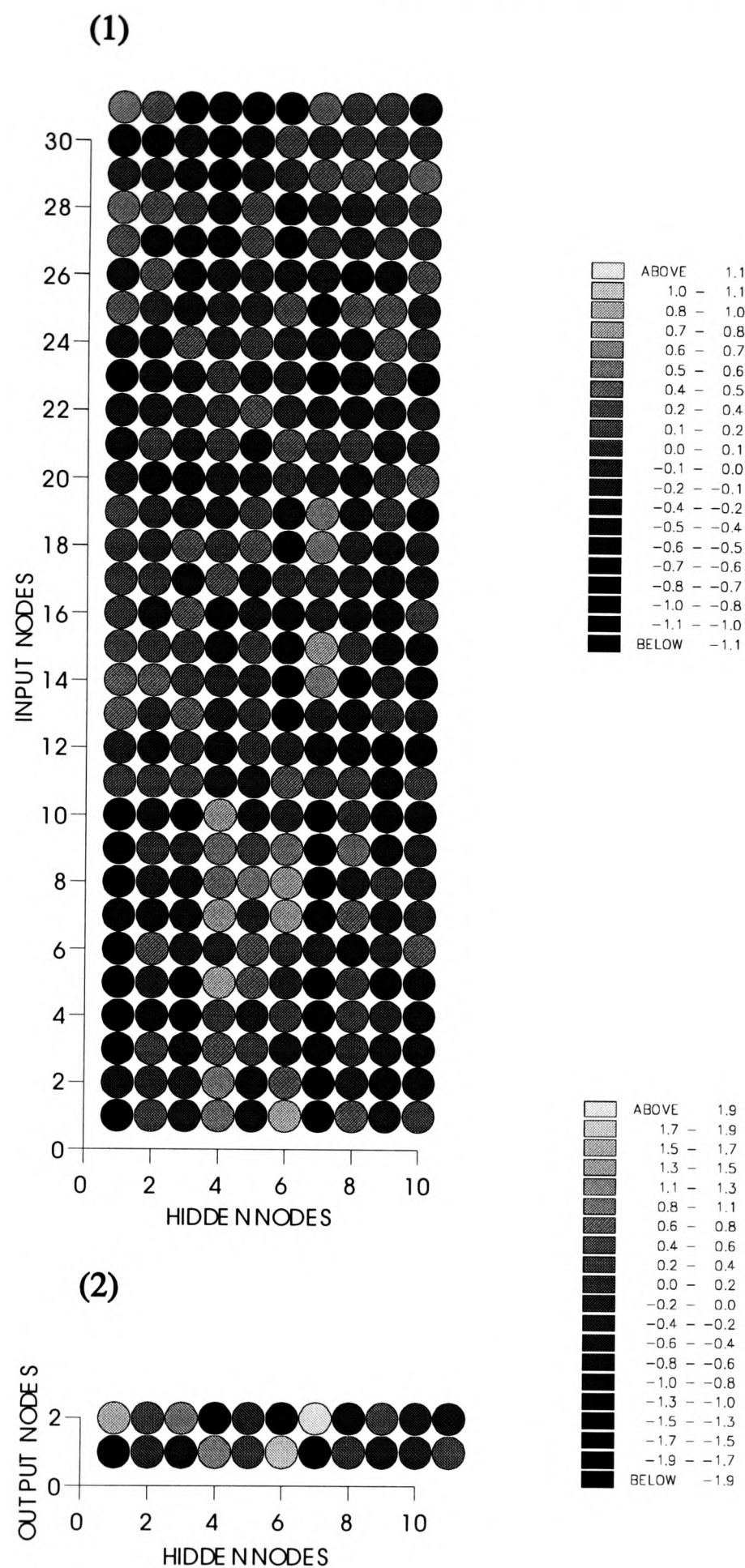


Figure 4.4 (c). Weight map for a trained BPNN. Notations as in Figure 4.4 (a). This BPNN is trained with seven pairs of P-arrival and noise segments. The training parameters are: learning rate = 0.9, momentum rate = 0.7, system error threshold = 0.00001 and pattern error threshold = 0.0001.

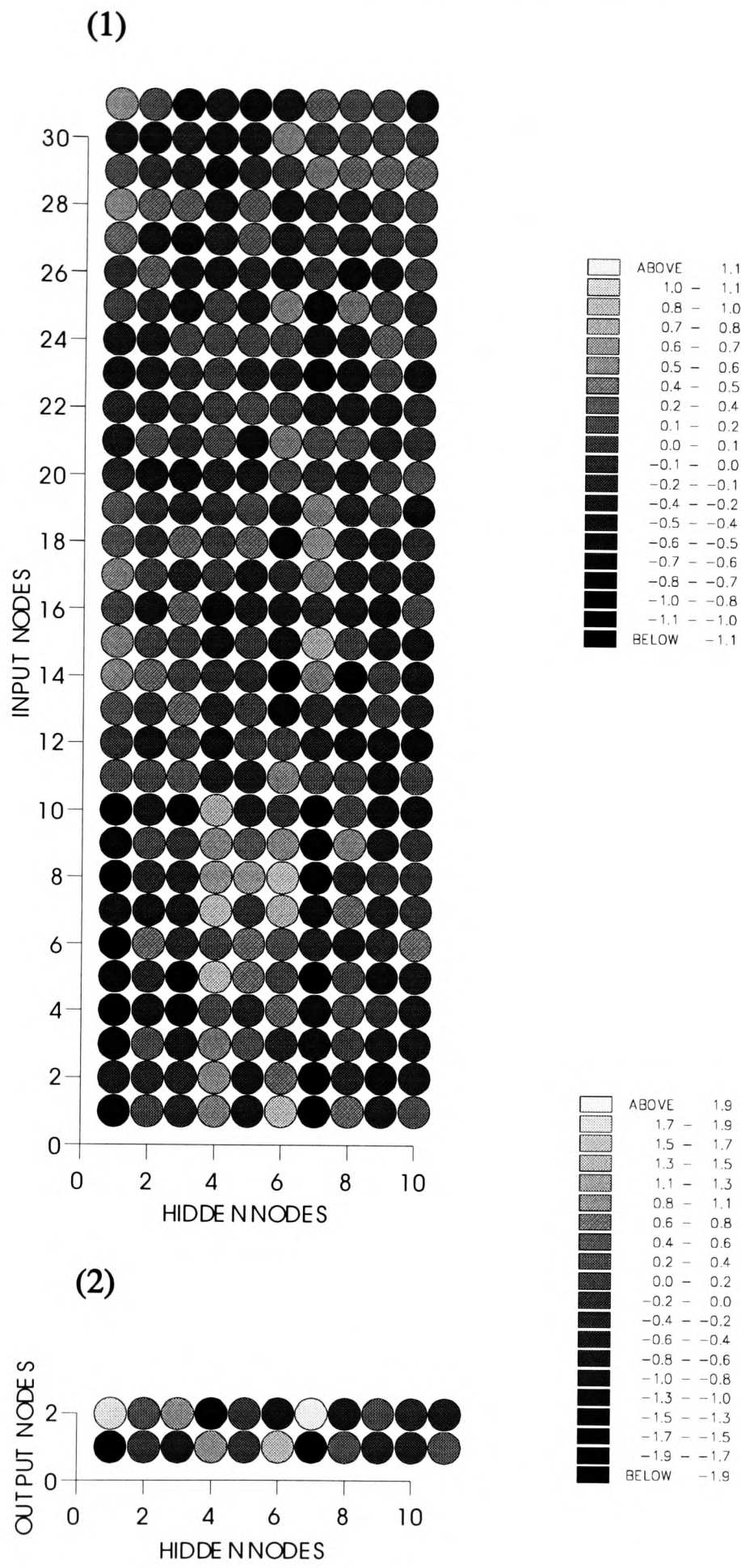


Figure 4.5 (a). Weight map for a trained BPNN. Notations as in Figure 4.4 (a). This BPNN is trained with seven pairs of P-arrival and noise segments. The training parameters are: learning rate = 0.7, momentum rate = 0.9, system error threshold = 0.0001 and pattern error threshold = 0.001.

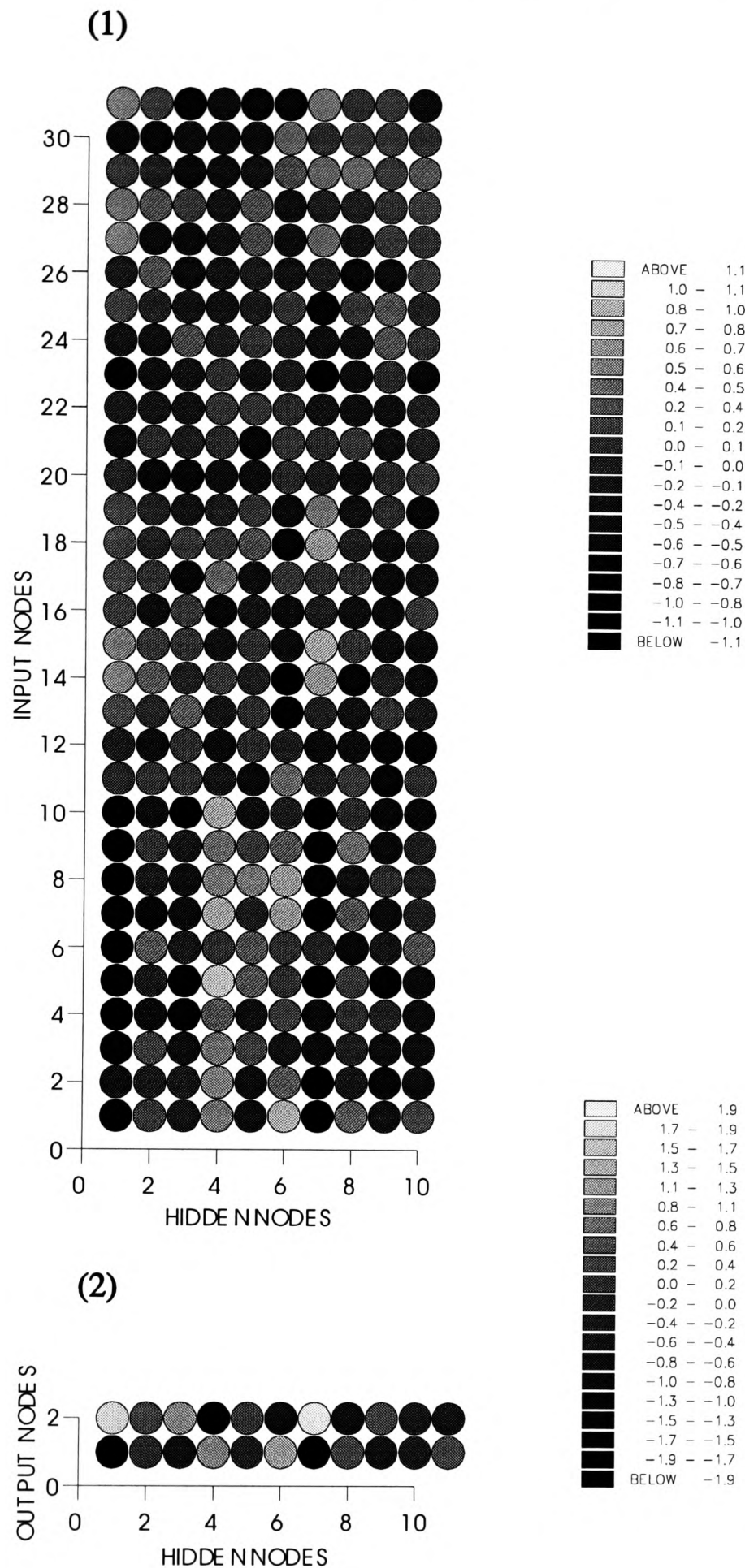


Figure 4.5 (b). Weight map for a trained BPNN. Notations as in Figure 4.4 (a). This BPNN is trained with seven pairs of P-arrival and noise segments. The training parameters are: learning rate = 0.7, momentum rate = 0.9, system error threshold = 0.000001 and pattern error threshold = 0.00001.

including all of their peculiarities.

### **(c) Determining the initial weights**

Normally, the weights (and thresholds ) of a BPNN are initialized with random values between an interval. However, this interval is also somewhat undetermined. In order to select a suitable value, I initialize the weights of a BPNN with different range of values. The BPNN then is trained with the same training dataset and training parameter. The result shows that the training procedure is longer when the weights of the BPNN are initialized with too small or too large a range of values. However, the weight patterns of the BPNNs are similar. Figure 4.6 shows two examples in which the BPNNs have different initial weights and threshold. Their patterns are similar with those in Figures 4.4 and 4.5.

### **(d) Values of training parameters and training results**

The above analysis shows that the structure of a trained BPNN depends mainly on the training dataset. The training parameters and the initial weights mainly affect the convergence of learning procedure. So I do not need to put too much effort into selecting their optimum values in each case. As a consequence, I use the following general values in the rest of this thesis:

learning rate = 0.7

momentum rate = 0.9

system error threshold = 0.00001

pattern error threshold = 0.0001

the range of initial weights is between -0.5 and +0.5.

With the above selected training dataset and the training parameters, the training procedure takes 102 iterations (less than a half minute CPU time on a VAX4000). The system error reached is 0.000032 with all pattern errors less than 0.0001. After this training procedure, the BPNN is ready to pick the seismic arrivals in the whole dataset.



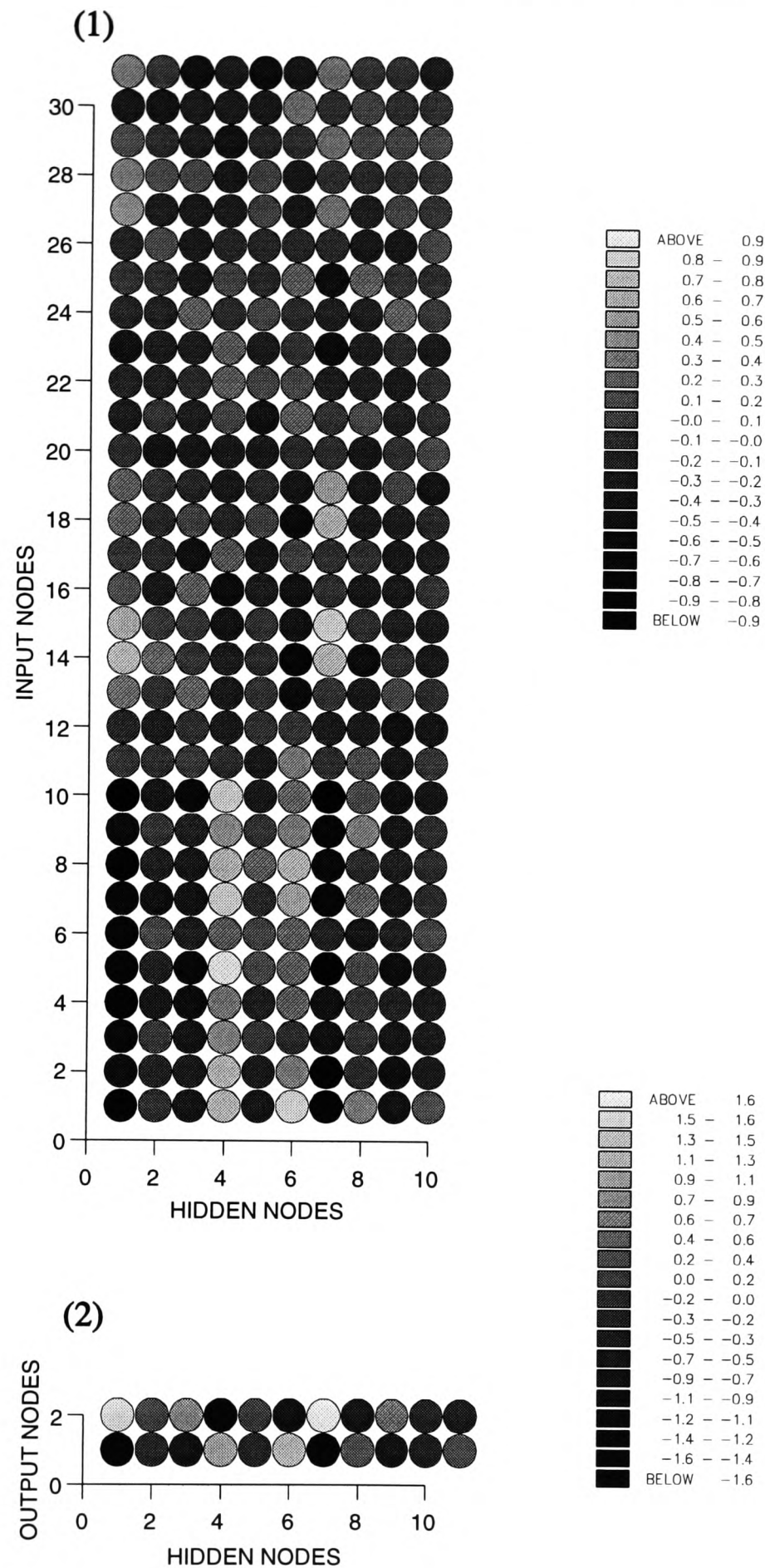


Figure 4.6 (a). Weight map for a trained BPNN. Notation as in Figure 4.4 (a) This BPNN is trained with seven pairs of P-arrival and noise segments. The training parameters are: learning rate = 0.7, momentum rate = 0.9, system error threshold = 0.00001 and pattern error threshold = 0.0001. The values of random initial weights are between -0.3 and +0.3. The iteration number is 101.

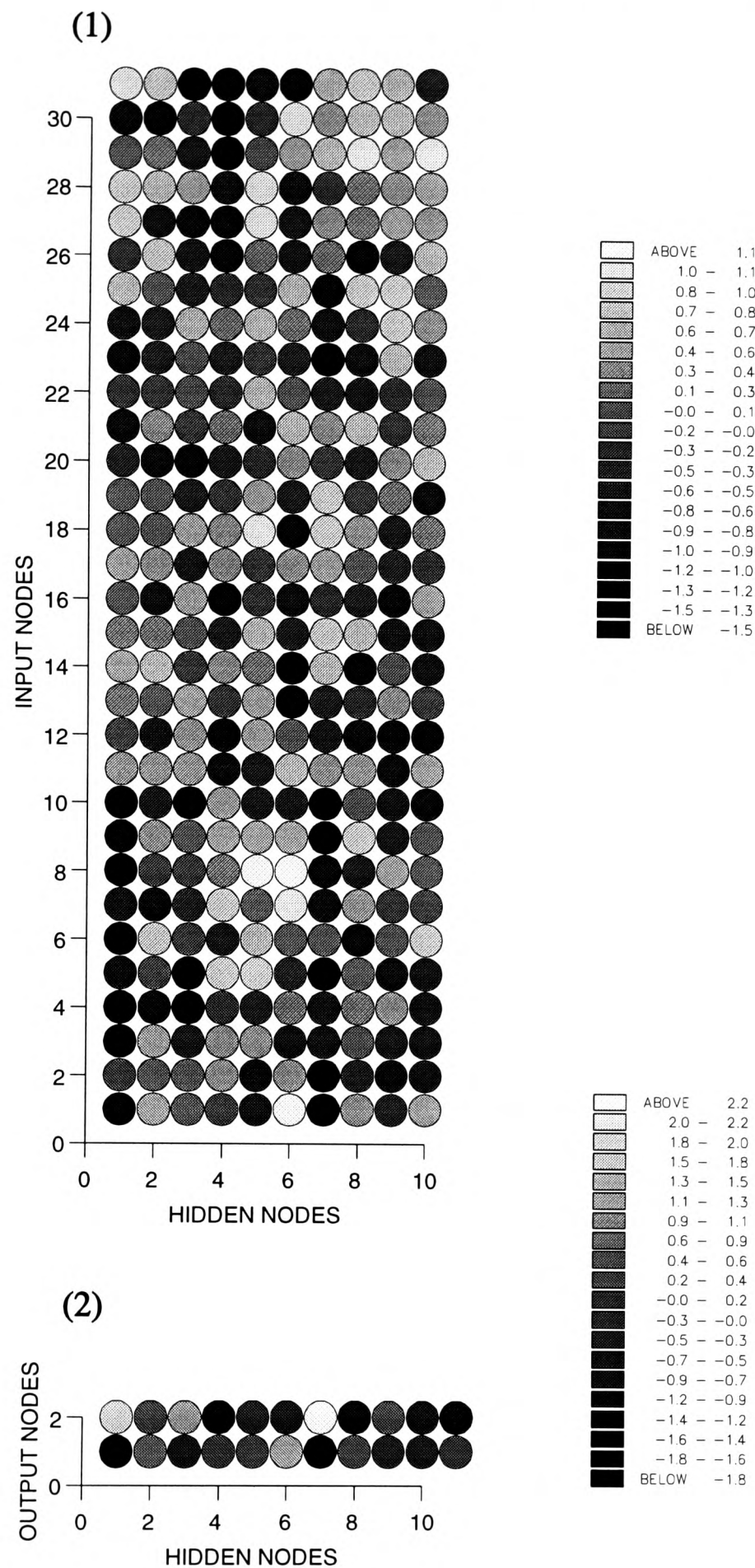


Figure 4.6 (b). Weight map for a trained BPNN. Notation as in Figure 4.4 (a) This BPNN is trained with seven pairs of P-arrival and noise segments. The training parameters are: learning rate = 0.7, momentum rate = 0.9, system error threshold = 0.00001 and pattern error threshold = 0.0001. The values of random initial weights are between -1.0 and +1.0. The iteration number is 152.

#### 4.3.4 Output function of trained BPNN

In order to detect and pick an arrival, I first calculate the vector  $M(t)$ , then take a windowed segment of  $M(t)$ , normalize the segment, and finally feed it into the trained BPNN. The BPNN will give its output which is stored for subsequent analysis. The window is then shifted by one sample at a time. This procedure is repeated until the end of the seismogram is reached. If a segment is the same as the training  $P$ -arrival segment or background noise segment, the output should be (0,1) or (1,0). In general, the output  $(o_1(t), o_2(t))$  lies between the ideal for a signal or for background noise (for example: (0.2, 0.8) or (0.6, 0.4)). To provide a single indication of the onset, I define a function  $N(t)$  which highlights the difference between the actual output and ideal noise:

$$N(t) = \frac{1}{2}[(1-o_1(t))^2 + o_2(t)^2] \quad (4.2)$$

$N(t) = 1.0$  if an input segment is the same as the training  $P$ -arrival segments or  $N(t) = 0.0$  if it is the same as the training background noise segment. Actually,  $N(t)$  is a measurement of the similarity between the input segment and the training segments. A time series of  $N(t)$  is obtained as the input window slides across the entire seismogram. Figure 4.7 shows an example of  $N(t)$  for one seismogram.

The significant feature of  $N(t)$  curve is that some peaks emerge from the smooth background. These peaks correspond to changes in  $M(t)$ , with values implying the level of change. These in turn are dependent on changes of the amplitude and frequency through the weighting in the BPNN. A high value peak indicates an abrupt change and a low value peak indicates a smooth change. For example the  $N(t)$  curve in Figure 4.7, has two large peaks corresponding to  $P$ - and  $S$ -arrivals. Note that this BPNN was only trained with  $P$ -arrival and background noise segments, so the prediction of  $S$ -arrivals is an added bonus. In this case, the positions of their maxima occur exactly at the manually chosen onset times of the  $P$ - and  $S$ -arrivals. There are also some small peaks in  $N(t)$  which show small changes in  $M(t)$ , regarded as noise



Station: DP  
 Date: 1984-07-04  
 Time: 00h16m19s  
 Scale: 546

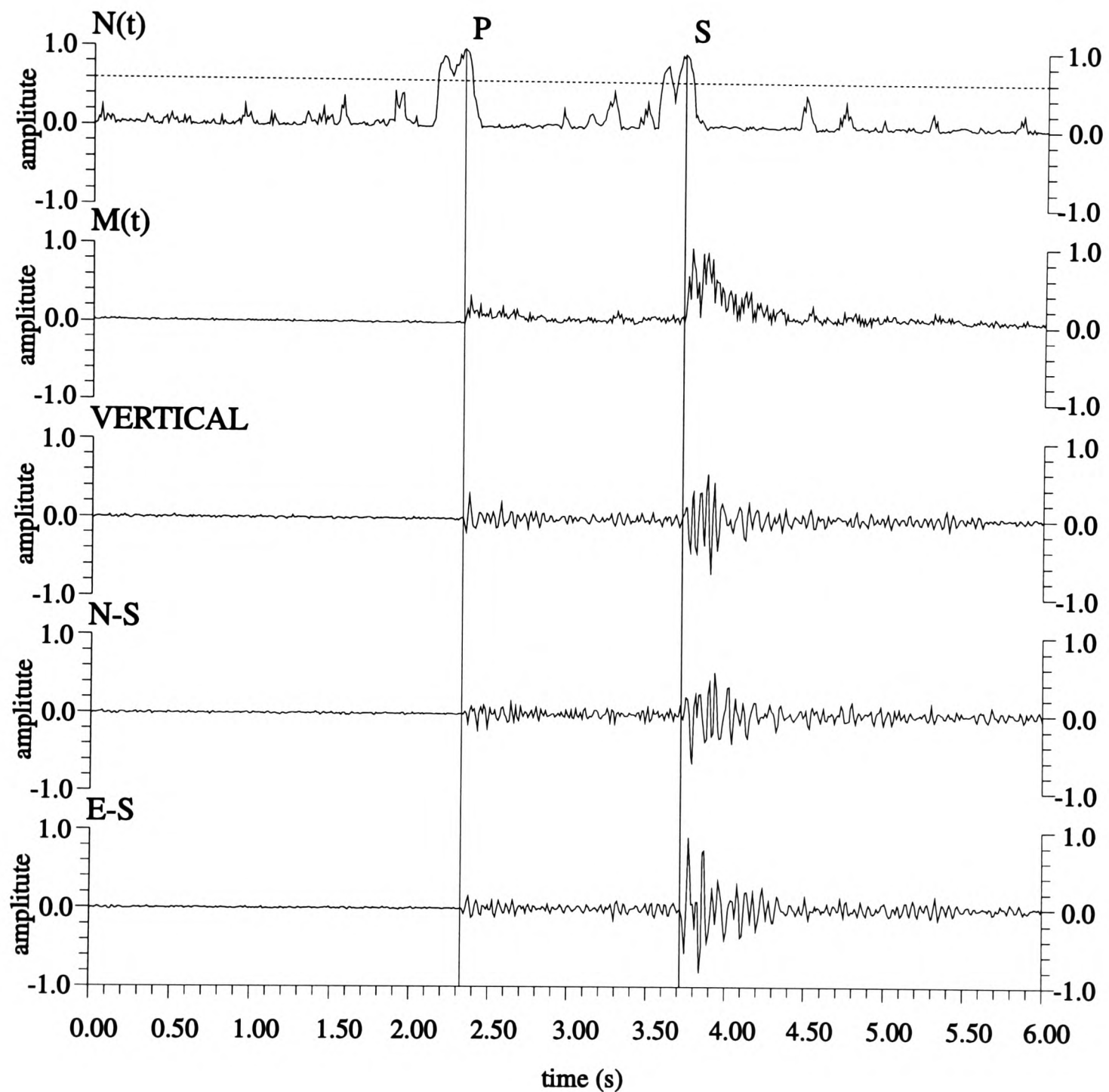


Figure 4.7. The 3-C seismogram (Vertical, N-S and E-W components) of a local earthquake, its vector modulus  $M(t)$  and  $N(t)$  function from the output of the trained BPNN. Two vertical lines are automatically drawn by the BPNN and exactly indicate P- and S-arrival onset times without error. The dashed line on  $N(t)$  shows the picking threshold (0.6) applied to  $N(t)$ .

bursts. From the transformation point of view, the trained BPNN transforms the rate of change of the  $M(t)$  into the value of  $N(t)$  from which the changes are enhanced and can then be more easily detected.

#### 4.3.5 Arrival detection

Usually, most arrivals correspond to high value peaks in  $N(t)$ , so it is quite easy to detect and pick the arrivals, and to measure their onset times. Here, a threshold rule is sufficient to detect the arrival. In this rule, if the value of  $N(t)$  exceeds a threshold, such as 0.6, it flags an arrival. For example, in Figure 4.7, only two peaks, corresponding to the  $P$ - and  $S$ -arrivals, exceed the threshold (0.6). It is interesting to note other peaks in this function, which indicate other wave arrivals, spikes or noise bursts, all of which are usually treated as *false alarms*. Figure 4.8 shows such an example including some spikes in the seismogram. Here, five peaks in  $N(t)$  exceed the threshold (0.6), indicating two spikes, a  $P$ -arrival, an  $S$ -arrival, and a noise burst. Other techniques are necessary to discriminate between these signals (see Section 4.3.7).

#### 4.3.6 Picking of arrival onset times

As the segments of  $M(t)$  are fed into the trained BPNN,  $N(t)$  reaches a maximum when the arrival time is at the eleventh point of the input window. Either side of this maximum,  $M(t)$  is shifted and the BPNN output  $N(t)$  decreases as shown on Figures 4.7 and 4.8. This implies that the onset time may be estimated by searching for a local maximum after  $N(t)$  exceeds the threshold. The positions of  $N(t)$  maxima exactly indicate the  $P$ - and  $S$ -arrival onset times in Figures 4.7 and 4.8. Note that a maximum might have a precursor which connects with the main peak, together with a base length equal to the input segment. According to the training procedure, small values of input samples before the tenth point increase the value of  $N(t)$ . As the arrival onset comes into the input window, the relative values of input samples before the onset decrease and the  $N(t)$  should increase. For a well-trained BPNN,  $N(t)$  should

Station: DP  
 Date: 1984-05-17  
 Time: 12h13m33s  
 Scale: 1958

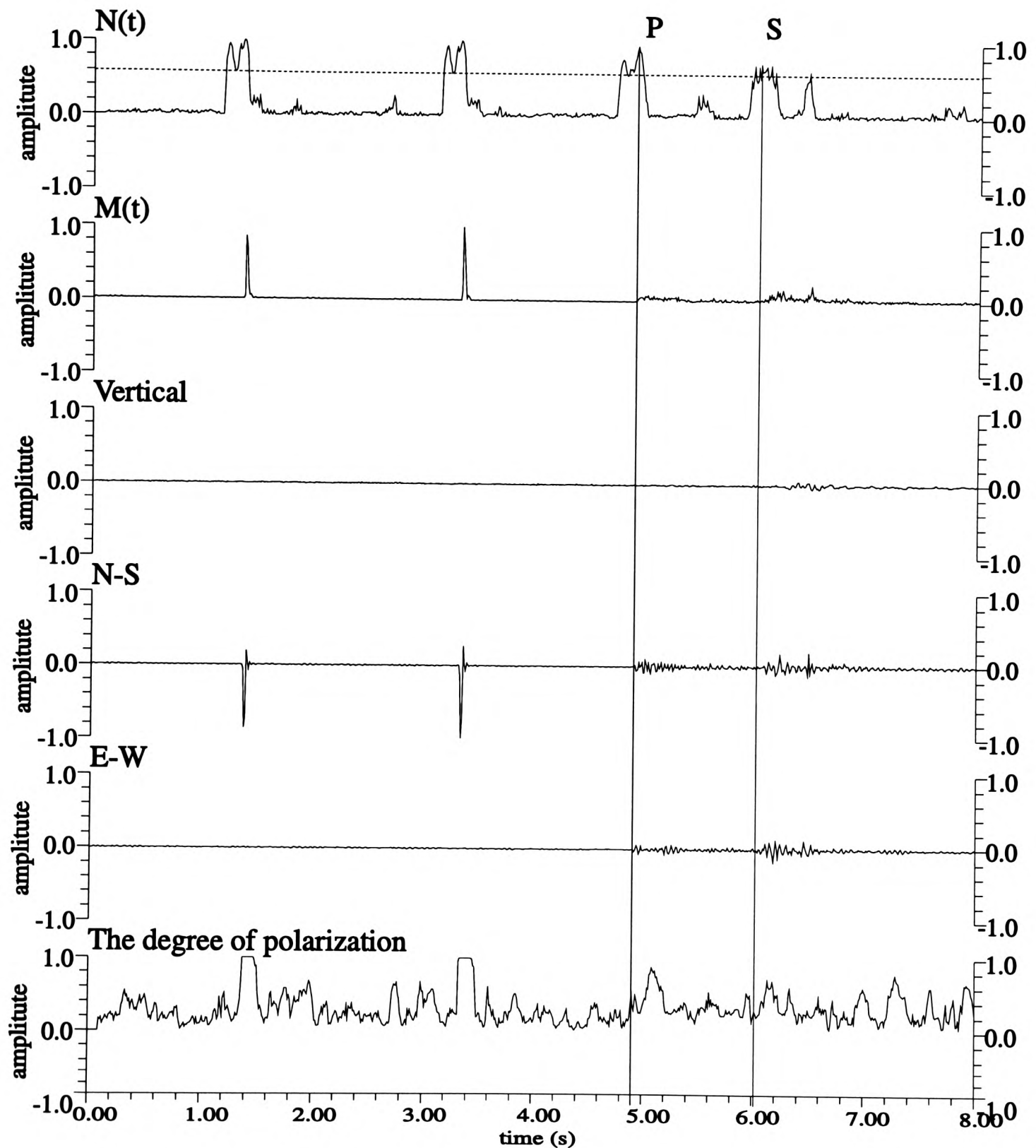


Figure 4.8. The 3-C seismogram (Vertical, N-S and E-W components) of a local earthquake, its vector modulus  $M(t)$ ,  $N(t)$  function from the output of the trained BPNN and its degree of polarization. Two electrical spikes interfere with the data on the N-S component. Two vertical lines show the P- and S-arrivals picked by the BPNN automatically. The dashed line shows the picking threshold (0.6) applied to  $N(t)$ .

gradually increase until the onset reaches the eleventh point and then  $N(t)$  gradually decrease until the onset moves out the input window. However, sometimes, for a less well-trained BPNN, as the onset moves forward in the window,  $N(t)$  may decrease at some points forming a precursor.

#### 4.3.7 Post-processing for discarding noise bursts and spikes

In the  $N(t)$  curve, some peaks correspond to noise bursts of low amplitude and low SNR (*small noise burst*), and some to spikes which are typically one or two sample points of anomalously large amplitude compared with the background signal. They may be discarded by using conventional algorithms described below.

##### 4.3.7.1 Discarding small noise burst

The features of a small noise burst are low amplitude and low SNR. Two criteria may therefore be used to discard this noise burst.

###### (a) mean-amplitude criterion

In the application of this criterion, the mean amplitude of a segment is calculated from the onset time to the end of the sliding window with 30 samples.

$$\text{mean-amplitude} = \frac{1}{N} \sum_{i=1}^N m_i \quad (4.3)$$

where  $m_i$  is the modulus value, and  $N$  is the window length. A threshold obtained from the background signals is then applied to the mean-amplitude. If the mean-amplitude is below the threshold, it is a small noise burst. For the data used, the mean-amplitude threshold is 16.

###### (b) mean-SNR criterion

The mean-SNR is different from the SNR defined in Chapter 1. It is

defined as the ratio between the mean-amplitudes after and before the onset .

$$\text{mean-SNR} = \frac{\text{mean-amplitude}_{\text{after-onset}}}{\text{mean-amplitude}_{\text{before-onset}}} \quad (4.4)$$

If the mean-SNR is below a threshold, it is a small noise burst. For the data used, the threshold is 1.7.

#### 4.3.7.2 Discarding spikes

A spike has the following features: (1) only one or two samples have much large values than other samples; (2) Its degree of polarization possesses a unique pattern (Section 3.3.3 and Figure 4.7) which is similar to a square wave with the amplitude of 1.0 and window length of ten. Two criteria can be used to discard the spike.

##### (a) spike-amplitude-ratio criterion

Three steps are used to define this criterion. First, in the sliding window in which an arrival is picked, the peaks of  $M(t)$  are chosen as  $p_i$  ( $i < \text{window-length}$ ). Second, the mean amplitude of peaks is calculated, excepting the largest two peaks. Thirdly, the ratio defined as:

$$\text{spike-amplitude-ratio} = \frac{\text{mean-peak-amplitude}}{\text{maximum-of-peaks}} \quad (4.5)$$

If this ratio is below a given threshold, it is a spike. For the data used, the threshold value was 0.1

##### (b) the degree of polarization criterion

This criterion is determined by counting the sample number of the degree of polarization whose value is greater than a threshold value. For the data used, the threshold value of the degree of polarization is 0.97 and the sample number is 8. If the counted number is greater than 8, it is a spike.



These criteria can be used to filter out most small noise bursts and spikes, however, other kinds of noise burst which are more similar to seismic arrivals are still left for further identification. I confine the discussion of such discrimination to a secondary stage of the analysis scheme, i.e. the arrival identification, developed in Chapter 6.

## 4.4 PERFORMANCE OF THE TRAINED BPNN

### 4.4.1 Performance of the trained BPNN

Using the above approach, with an  $N(t)$  threshold of 0.6, it is possible to detect 200  $P$ -arrivals (95.2%) and 184  $S$ -arrivals (87.6%) from 210 3-C recordings of the total dataset used in this experiment. If the  $N(t)$  threshold is decreased to 0.5, 205  $P$ -arrivals (97.6%) and 200  $S$ -arrivals (95.2%) are detected. Given that only seven pairs of  $P$ -arrival and background noise segments from station DP were used to train this BPNN, this result is extremely encouraging. The picked onset times are also compared with the results from an independent, conventional manual analysis. Figure 4.9 shows the statistical comparison. Here, this BPNN can estimate at least 75.2% of the  $P$ -arrivals and 50.0% of the  $S$ -arrivals having onset times  $\leq 10$ ms (one sample increment) using a threshold of 0.6, or 77.1% of the  $P$ -arrivals and 53.8% of the  $S$ -arrivals with a reduced  $N(t)$  threshold of 0.5. The onset time is relatively insensitive to the threshold, confirming that the automatic picking is determined predominantly by the local maximum of  $N(t)$  rather than the  $N(t)$  threshold.

### 4.4.2 Improving performance by adding a new training dataset

Let us now consider the possibility of improving the arrival picks missed by the trained BPNN. These missed arrivals do not have clear first motions, and the changes of their  $M(t)$  are not visually obvious, with a small corresponding maximum in  $N(t)$ . A more suitable strategy is to retrain the BPNN by including this type of data. This approach may be negated if too much training data needs to be used, as this

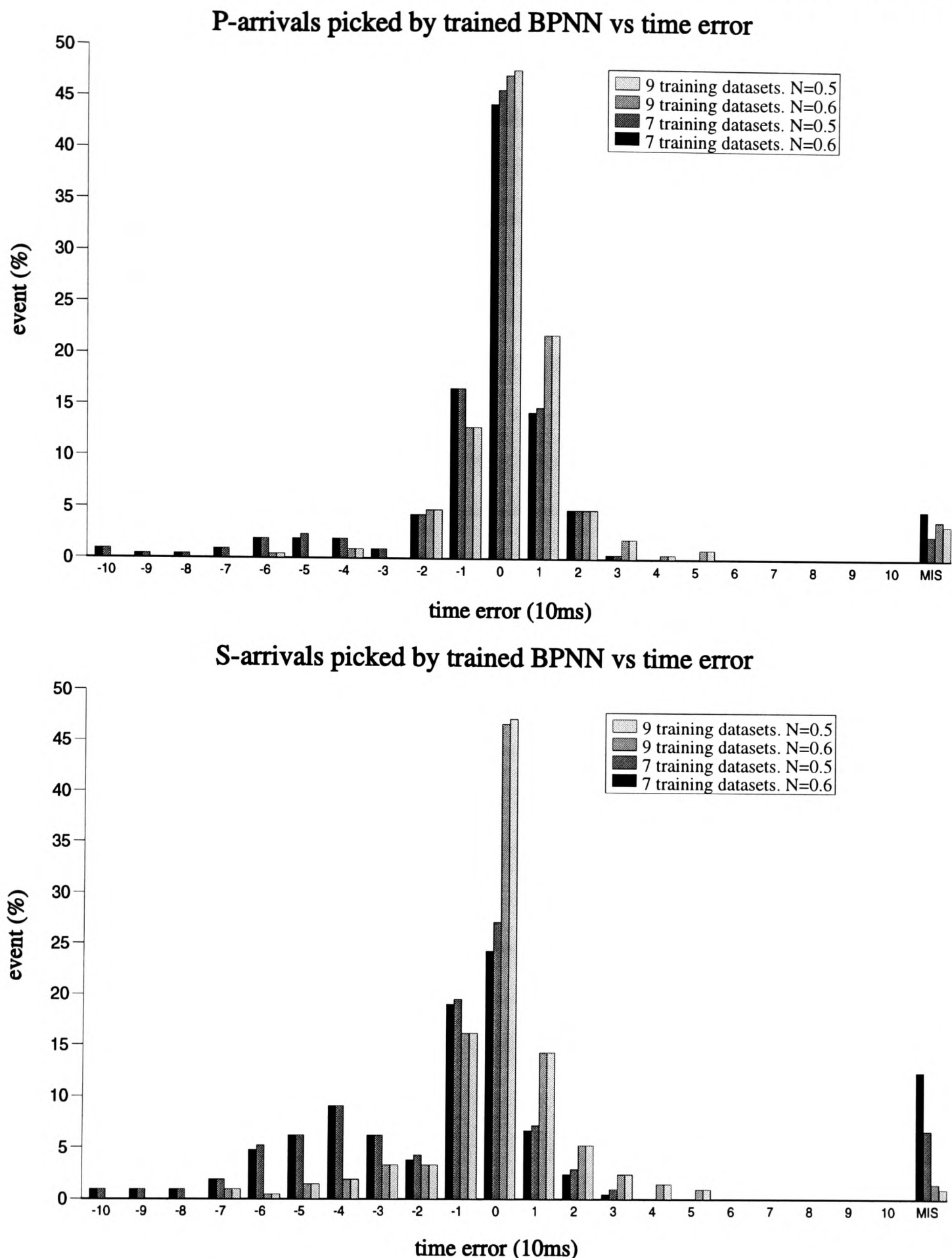


Figure 4.9. Statistical comparison of P- and S-arrivals picked on a dataset of 210 local earthquake records. A negative value of time error indicates a later pick by the trained BPNN approach than that given by visual analysis and the positive value indicates an earlier pick. MIS refers to all those unpicked arrivals, defined as picks with errors larger than 10 samples increments (100ms).  $N$  in the legend is the  $N(t)$  threshold. Results are shown both for the original (7) and retrained BPNN (9). Note the improved performance of the retrained BPNN.

increases the training time, and there is the possibility of having to accommodate more subtle variations using a larger BPNN structure. In order to tackle this, I include two extra *P*-arrival segments which have different shapes from the other training *P*-arrival segments and two corresponding background noise segments (groups 8 and 9 in Figure 4.3). This training procedure takes 112 iterations, with a system error of 0.000012 and all pattern errors less than 0.0001. Figure 4.10 shows an example of its performance for picking. Compared with Figure 4.7, this retrained BPNN has a smoother output. The two peaks in  $N(t)$  correspond of the *P*- and *S*-arrivals have the higher values, and other small peaks in  $N(t)$  have lower values than those in Figure 4.6. Its statistical result is also shown on Figure 4.9 compared with the previously trained BPNN. The retrained BPNN has an improved performance over the previous one. It detects 202 (96.1%) of the *P*-arrivals and 204 (97.1) of the *S*-arrivals using a threshold for  $N(t)$  of 0.6. The estimated onset times are also more accurate, with at least 172 (81.9%) of the *P*-arrivals and 162 (77.1%) of the *S*-arrivals (for  $N(t) > 0.6$ ) or 173 (82.3%) of the *P*-arrivals and 163 (77.6%) of the *S*-arrivals (for  $N(t) > 0.5$ ) having onset times  $\leq 10\text{ms}$  (one sample increment). Only one *P*-arrival and three *S*-arrivals have onset times with errors  $\geq 50\text{ms}$  (five sample increments), and eight *P*-arrivals and six *S*-arrivals are missed completely. Note that only two extra *P*-arrivals are detected but 20 more *S*-arrivals, with a commensurate increase in the picking ability. The ability to pick *S*-arrivals is improved as the additional *P*-arrival segments actually resemble many of the *S*-arrivals. Again, decreasing the  $N(t)$  threshold to 0.5 does not significantly improve the picking, with only one more *P*-arrival and one more *S*-arrival picked. This comparison substantiates the well-known adaptive behaviour of BPNNs, that improvement can always be achieved by judicious choice of the training dataset.

#### 4.4.3 Sensitivity to segment length

The time taken during analysis depends on the BPNN structure. The structure can be decreased by decreasing the nodes in the input layer and in the hidden layer. A reduction in the input nodes was tested by reducing from 30 to 20 input nodes,



Station: DP  
Date: 1984-07-04  
Time: 00h16m19s  
Scale: 546

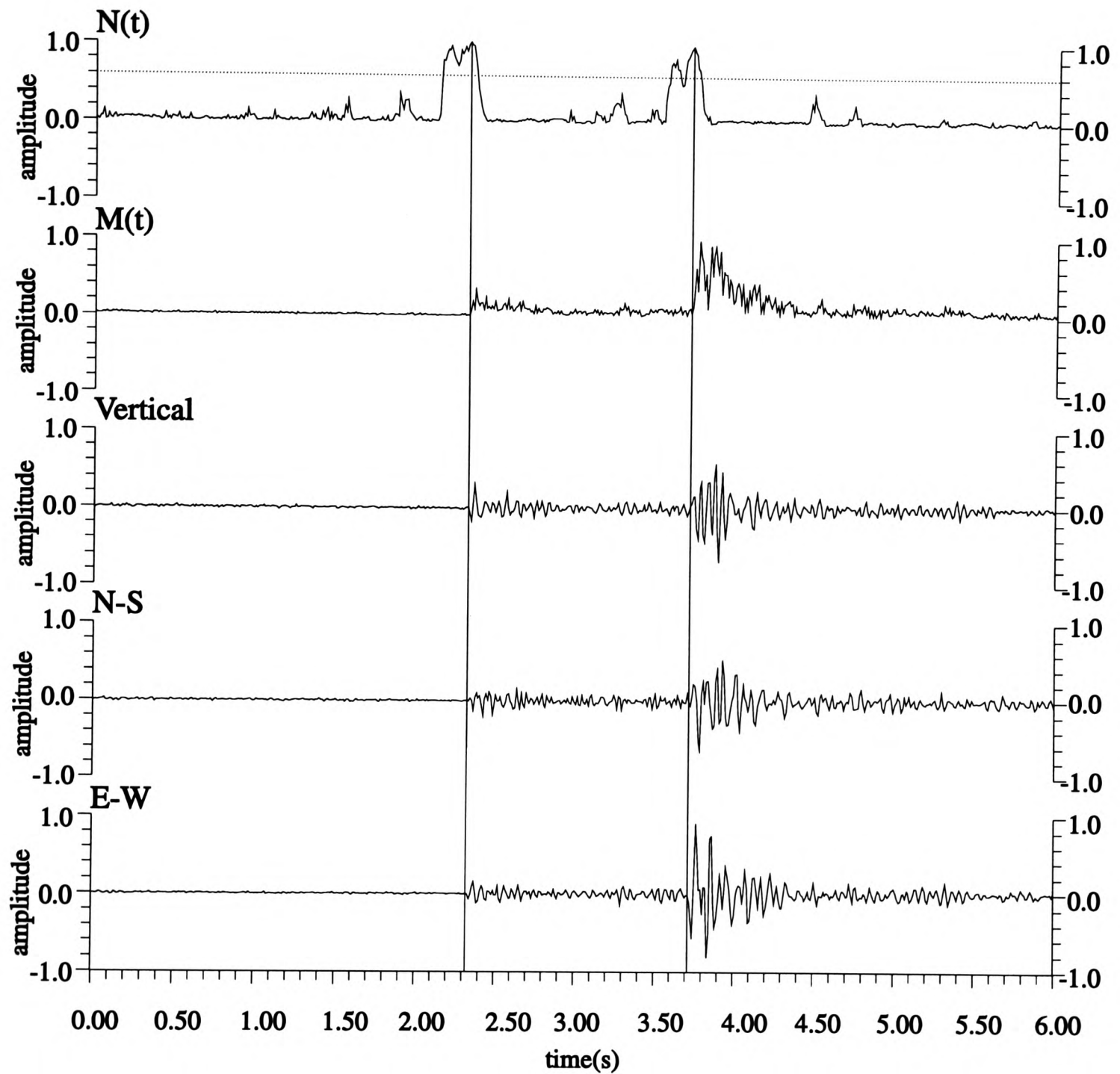


Figure 4.10. P- and S-arrivals picked by using a retrained BPNN with 9 pairs of P- and noise segments. The notations is as in Figure 4.6.

keeping the hidden nodes and output nodes the same. The predicted onset time of every *P*-arrival of training segments also lies at the eleventh sample. I still use the nine pairs of *P*-arrival and background noise segments to train this BPNN. In this case, the training procedure is slower, and it took 613 iterations to reach a satisfactory convergence point with a system error of 0.000016 and all pattern errors less than 0.0001. The results are now worse than before, although the number of unpicked *P*-arrivals remains the same (8 sets of records), with the number of unpicked *S*-arrivals increasing from 6 to 14 events. The onset estimation is worse, with only 57.6% of the *P*-arrivals and 55.2% of the *S*-arrivals having estimated times with error  $\leq 10$ ms. There are also a larger number of spurious picks. Figure 4.11 summarises the comparison for this case and the original one. The input nodes are also increased to 40 for comparison, with the BPNN now taking 113 iterations to converge with a system error of 0.00003 and all pattern errors less than 0.0001. Now, only 67.8% of the *P*-arrivals and 63.8% of the *S*-arrivals are picked with error  $\leq 10$ ms. There is a larger number of unpicked arrivals in this case. For the BPNN with 40 or 30 input nodes, its performance also decreases when the predicted onset time of the *P*-arrivals is at other positions.

I should point out that the input segment length depends on the characteristics of the signals. It is suggested that the segment should include several complete cycles of a wavelet. It appears that reducing or increasing the number of input nodes dramatically affects the performance and there appears to be an optimum number of input nodes for our particular configuration. This reflects the general observation that the BPNN architecture must be specifically tailored to individual applications. This represents the 'Achilles heel' of ANN applications, and further optimization is required to adapt to particular event types.

#### 4.4.4 Sensitivity to signal-to-noise ratio

One additional benefit of this approach is its ability to pick arrivals in low SNR condition, by using a BPNN trained with only high SNR *P*-arrival segments in

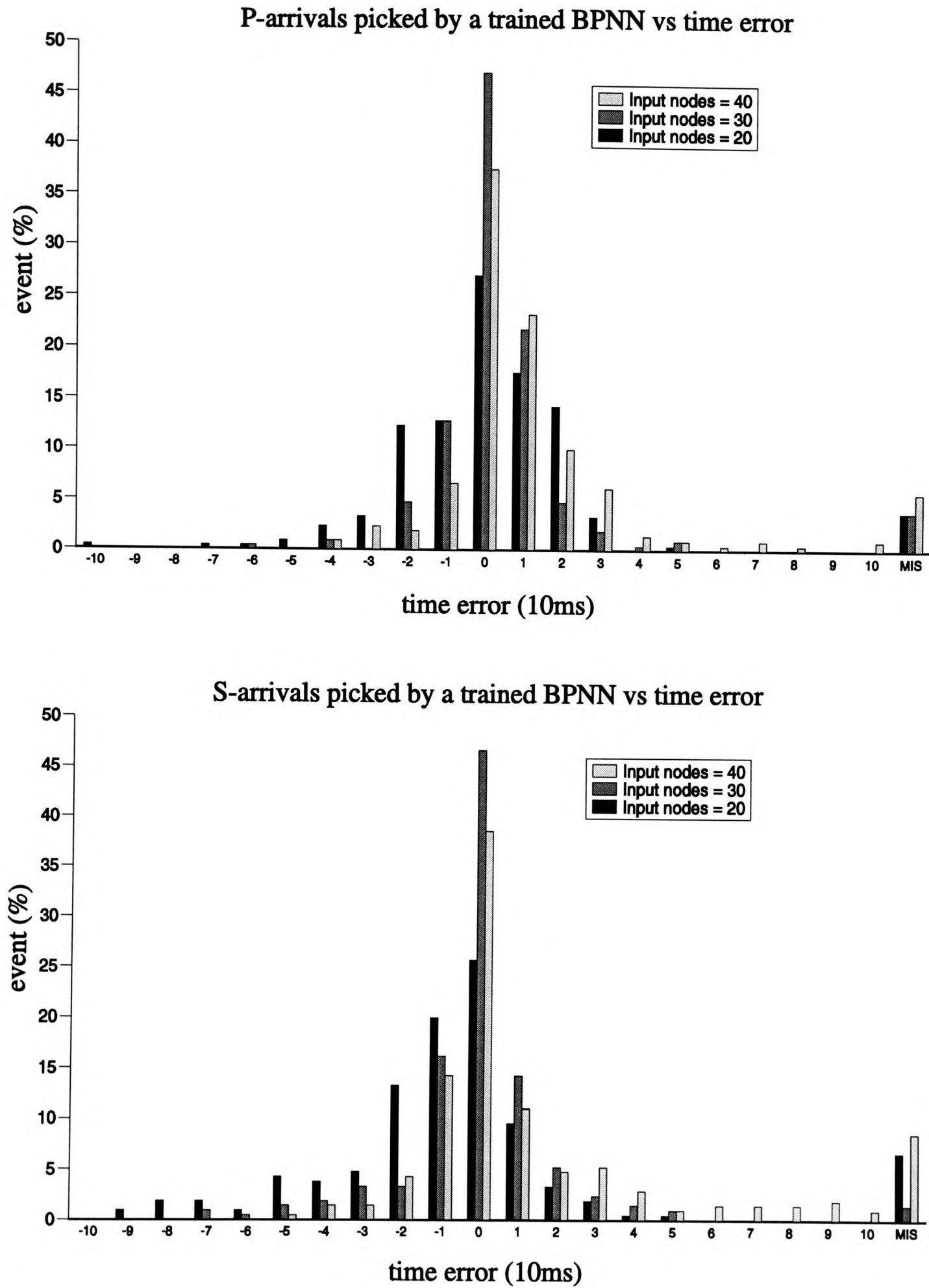


Figure 4.11. Statistical comparison of for P- and S-arrivals picked on a dataset of 210 local earthquake recordings by three BPNNs with different number of input nodes (see key). Notations as in Figure 4.8.

this study. All training  $P$ -arrival segments have the SNRs greater than 6.3. Figure 4.12 shows an example in which  $N(t)$  has two peaks, indicating two arrivals with high SNRs, 12.3 for the  $P$ -arrival and 7.3 for the  $S$ -arrival. The peaks are nearly rectangular in shape, with a width of roughly 30 sample points. Figure 4.13 shows an example in which two  $N(t)$  peaks indicate two arrivals with low SNRs, 4.3 for the  $P$ -arrival and 3.8 for the  $S$ -arrivals. Both peaks appear sharper. Although the 3-C particle motion does not display a significant difference between the seismic signal and the background noise for this case,  $M(t)$  does indicate the change in the nature of the seismic signal, and this change is translated to the narrow peak in  $N(t)$  which now indicates the onset. This reveals the possibility of interpreting the shape of these peaks to extend the ability of the trained BPNN beyond the boundaries of the training set. Figure 4.14 shows another low SNR example in which the  $P$ -arrival (SNR=2.0) and  $S$ -arrival (SNR=2.4) can be automatically picked by the BPNN although their onsets are not clear. Between the  $P$ - and  $S$ -arrivals, another peak in  $N(t)$  slightly exceeds the threshold 0.6, indicating a small change in  $M(t)$ . It is discarded by post-processing due to its low SNR (1.4) and low amplitude (12).

## 4.5 APPLICATION TO COMPLETE DATASET

### 4.5.1 Data characteristics

Here, I test the trained BPNN further by incorporating additional recordings from stations DP and AY, which now form the complete dataset of 1754 3-C recordings (877 from station DP, and 877 from station AY), described in Section 3.2.2, from which only the high quality data were selected in the previous section. By checking the quality of these recording, I can only select 762 (371 from station DP and 391 from station AY) recordings which are usable for further processing. However, not all of them include usable seismic signals, for example, some of them only record the background signals, some of them consist of non-seismic signals, some of them do not have the clear first breaks of seismic arrivals, and some of them

Station: DP  
Date: 1984-10-06  
Time: 20h01m42s  
Scale: 490

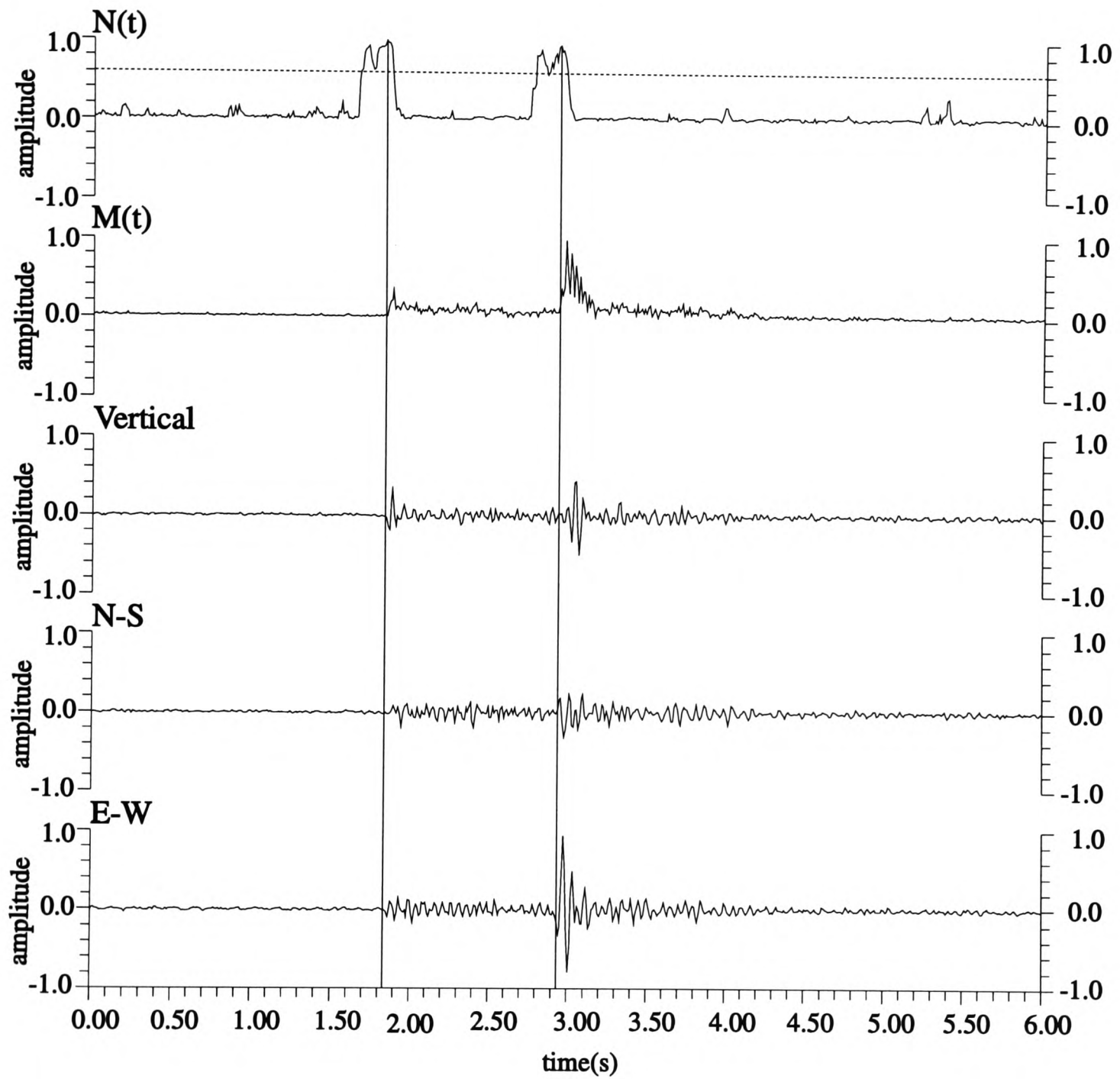


Figure 4.12. An example of picking P- and S-arrivals in a high SNR condition. Notations as in Figure 4.6.  $N(t)$  is computed from the output of a BPNN trained with nine pairs of P-arrival and background noise segments.

Station: DP  
Date: 1984-05-04  
Time: 03h17m15s  
Scale: 111

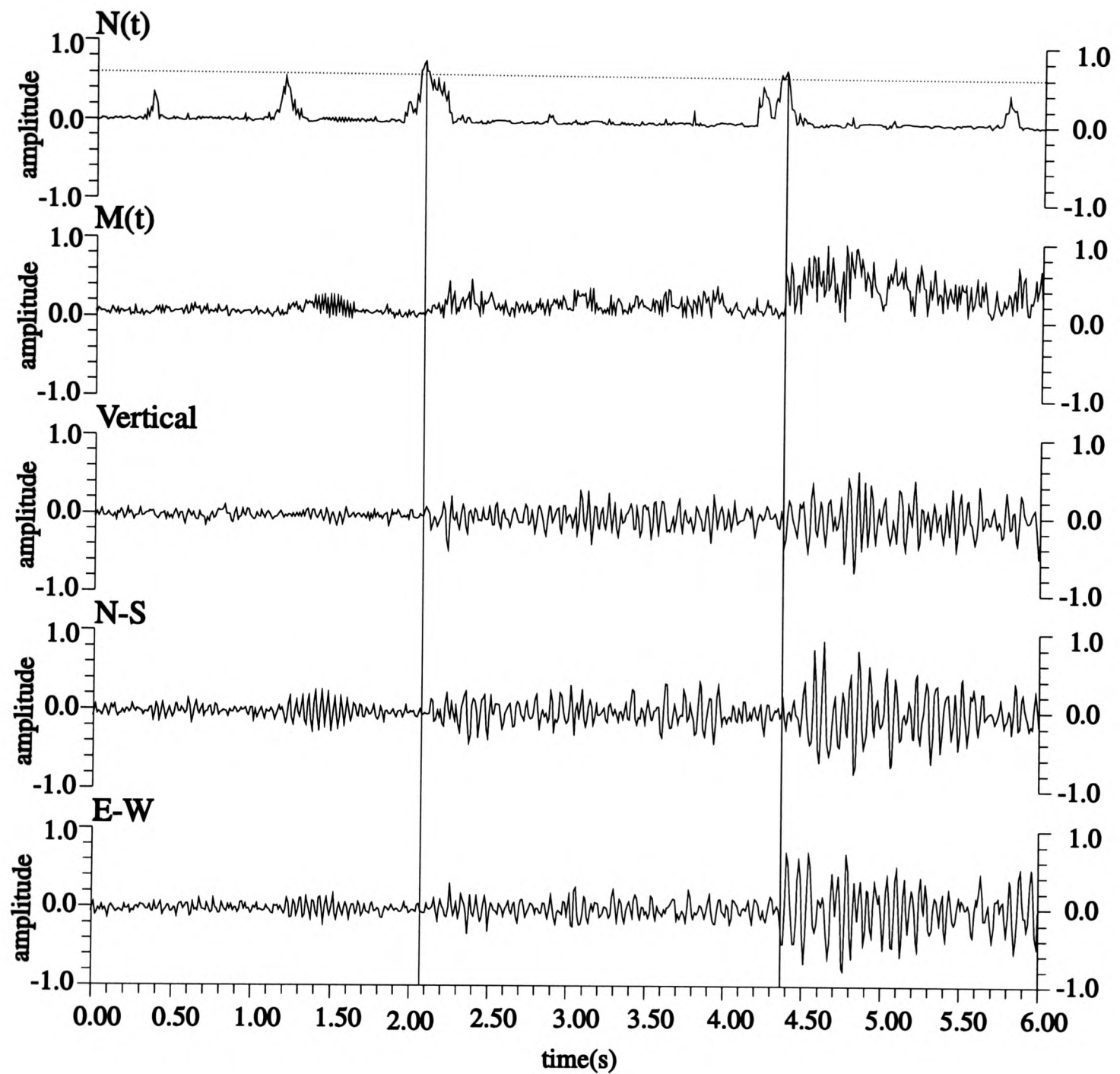


Figure 4.13. An example of picking P- and S-arrivals in a low SNR condition. Notations as in Figure 4.6.  $N(t)$  is computed from the output of a BPNN trained with nine pairs of P-arrival and background noise segments.

Station: AY  
Date: 1984-06-07  
Time: 01h52m46s  
Scale: 52

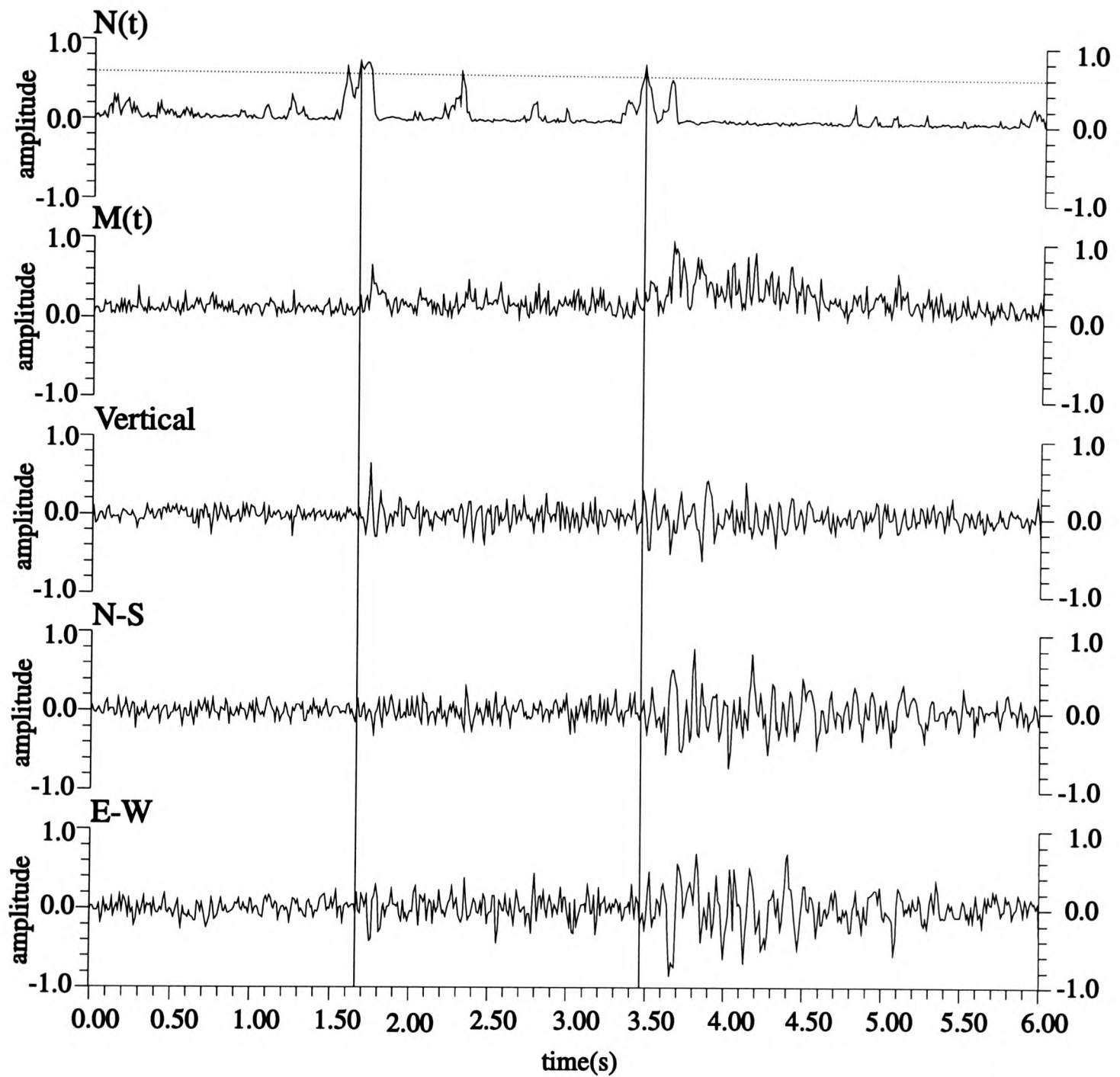


Figure 4.14. An example of picking P- and S-arrivals in a low SNR condition. Notations as in Figure 4.6.  $N(t)$  is computed from the output of a BPNN trained with nine pairs of P-arrival and background noise segments.

include two or more earthquake events in one recording. So these data are reduced further as manual picks for comparison with the BPNN results are only possible for 356 *P*-arrivals and 342 *S*-arrivals from 371 recordings at station DP and 300 *P*-arrivals and 285 *S*-arrivals from 391 recordings at station AY. In total, 656 *P*-arrivals and 627 *S*-arrivals are manually picked from 762 recordings. One should be aware that these statistics will appear more successful than if this procedure had been applied to all the data irrespective of quality.

#### 4.5.2 Performance

The BPNN used in this more extensive test is the same as the previous one, trained by nine pairs of *P*-arrival and background noise segments, with 30 input nodes. With an  $N(t)$  threshold of 0.6, the BPNN can detect 345 (96.9%) of the *P*-arrivals and 302 (88.3%) of the *S*-arrivals from station DP, and 274 (91.3%) of the *P*-arrivals and 240 (84.2%) of the *S*-arrivals from station AY. Figure 4.15 shows the picking result related to event positions. It seems the BPNN's picking ability does not relate to the event positions. The overall success rates are 619 (94.3%) for the *P*-arrivals and 542 (86.4%) for the *S*-arrivals. Most of the failures arise at the low SNRs of between 1 and 3. The method appears to pick all phases with an SNR > 3. Once picked, the SNR does not affect the accuracy of the onset estimation. Figure 4.16 shows a comparison of the trained BPNN results with the manual picks, in which for station DP, 285 (80.0%) of the *P*-arrivals and 233 (68.1%) of the *S*-arrivals have onset times within 10ms of the expected manual values, and for station AY, 204 (68.0%) of the *P*-arrivals and 163 (57.2%) of the *S*-arrivals have an onset times within the same tolerance. The overall rates are 74% (489) for *P*-arrivals and 63.2% (396) for *S*-arrivals. In addition, only 7.0% of the *P*-arrivals and 14.7% of the *S*-arrivals have onset times with errors  $\geq 50$ ms (five sample increments) or are missed completely.

There are in total 37 *P*-arrivals and 85 *S*-arrivals missed by the trained BPNN. They usually have indistinct first break motions so that their  $N(t)$  value is lower than 0.6. However, most of them have clear peaks in  $N(t)$  corresponding to their onsets



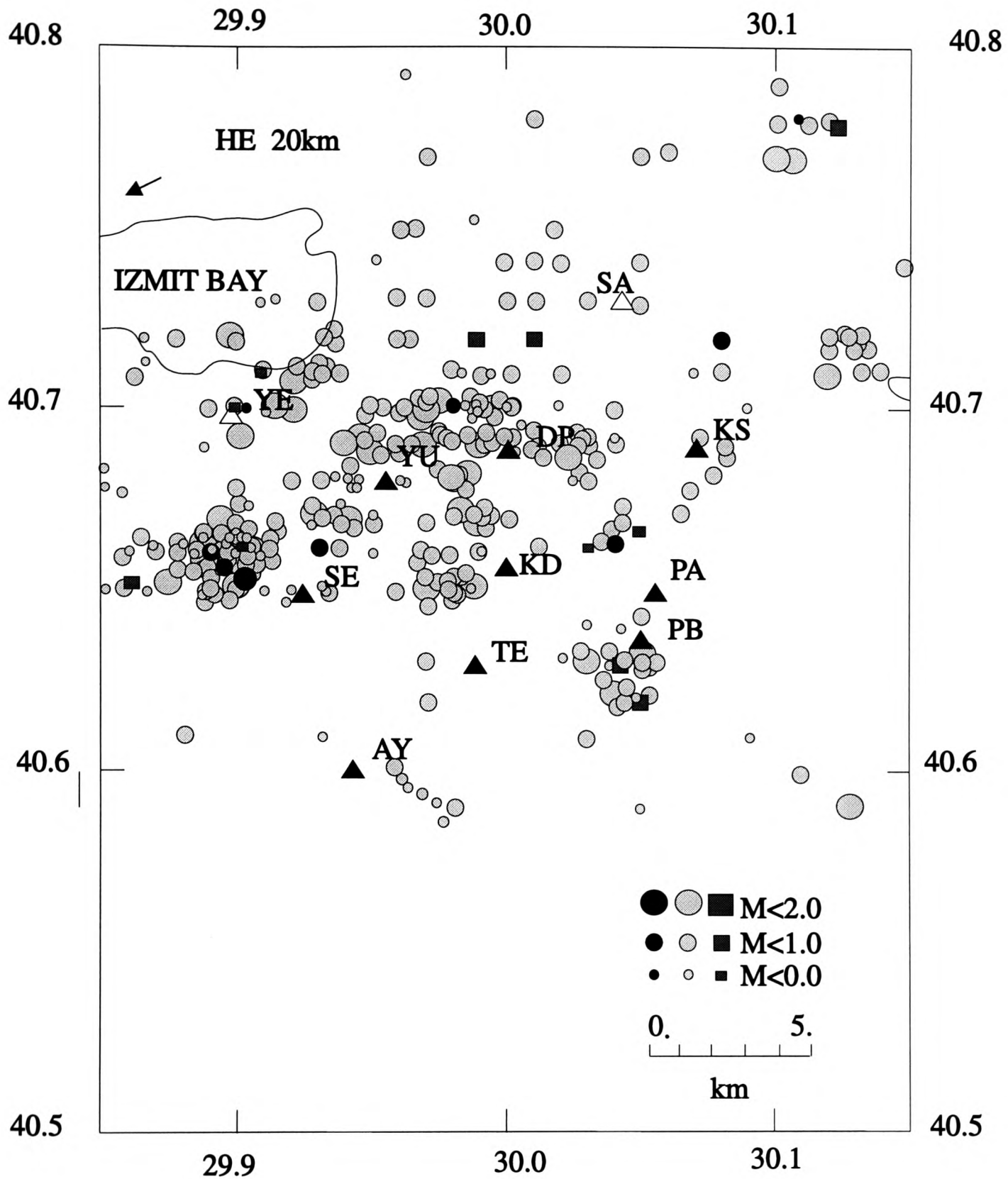


Figure 4.15(a). The location map of events which recorded in station DP. Each symbol represents an event whose P-arrival was manually picked. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals.

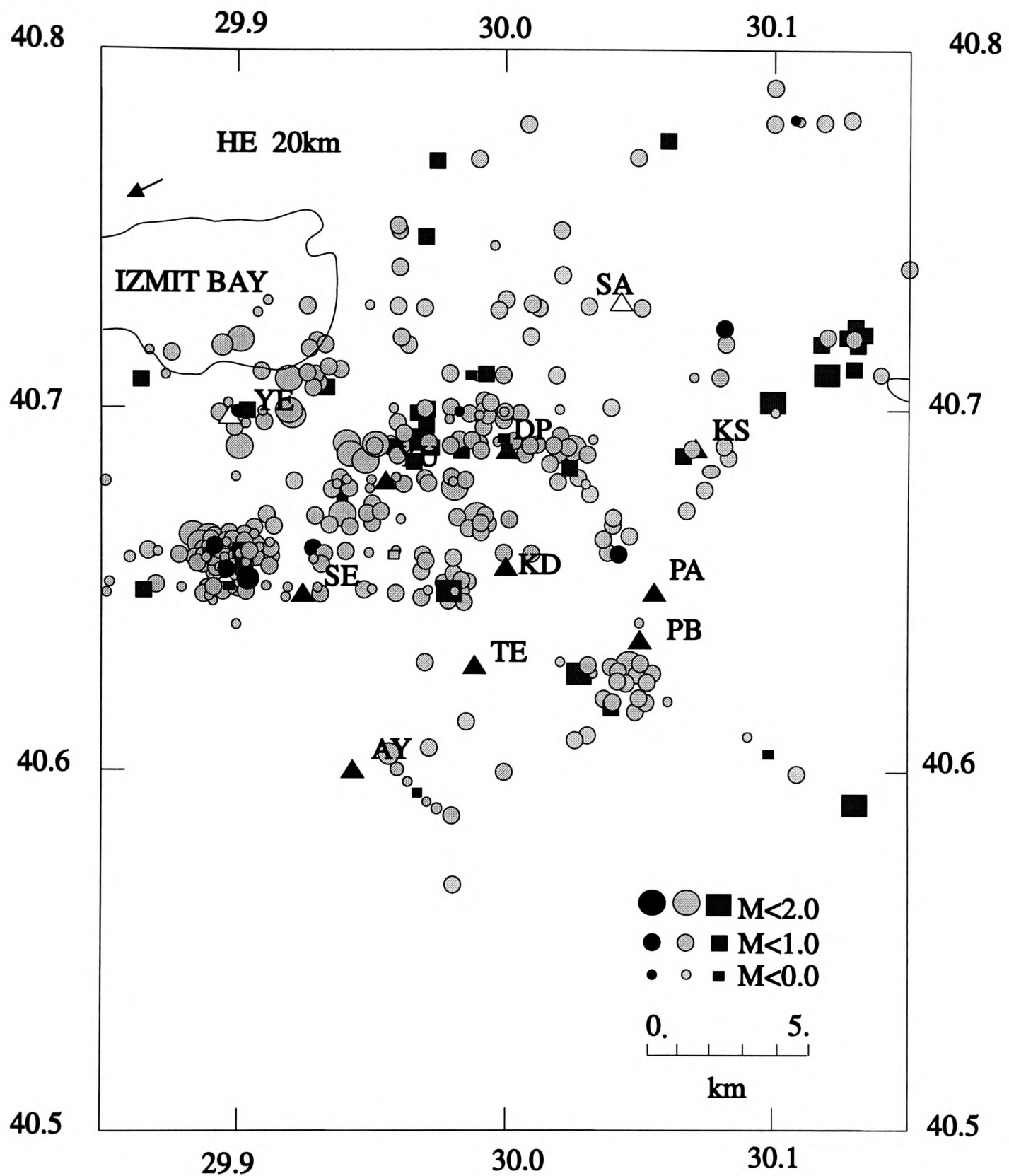


Figure 4.15(b). The location map of events which recorded in station DP. Each symbol represents an event whose S-arrival was manually picked. Grey circles represent S-arrivals which are picked by the BPNN, and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals (Note that this BPNN is trained by P-arrivals)

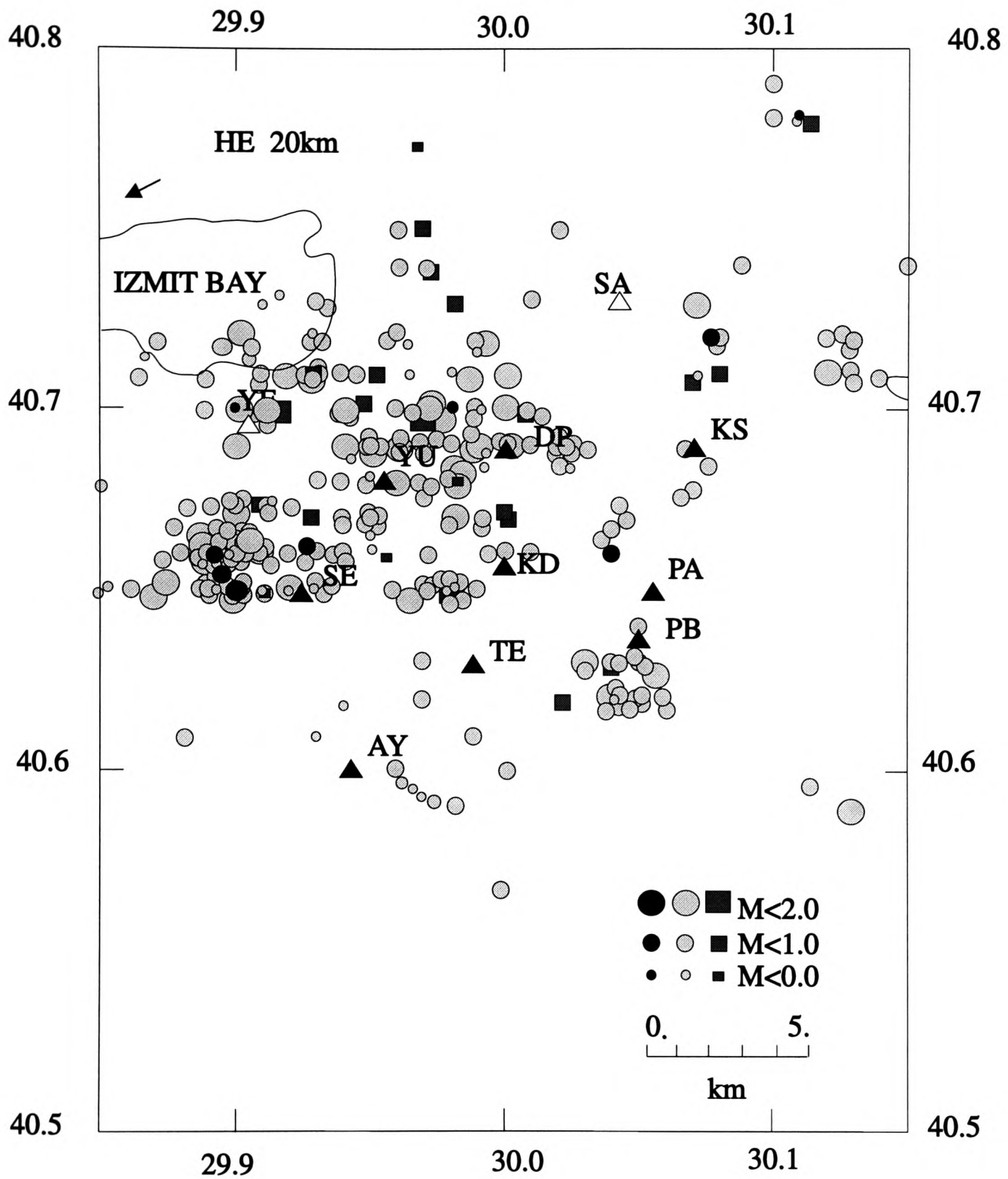


Figure 4.15(c). The location map of events which recorded in station AY. Each symbol represents an event whose P-arrival was manually picked. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals which are from station DP.

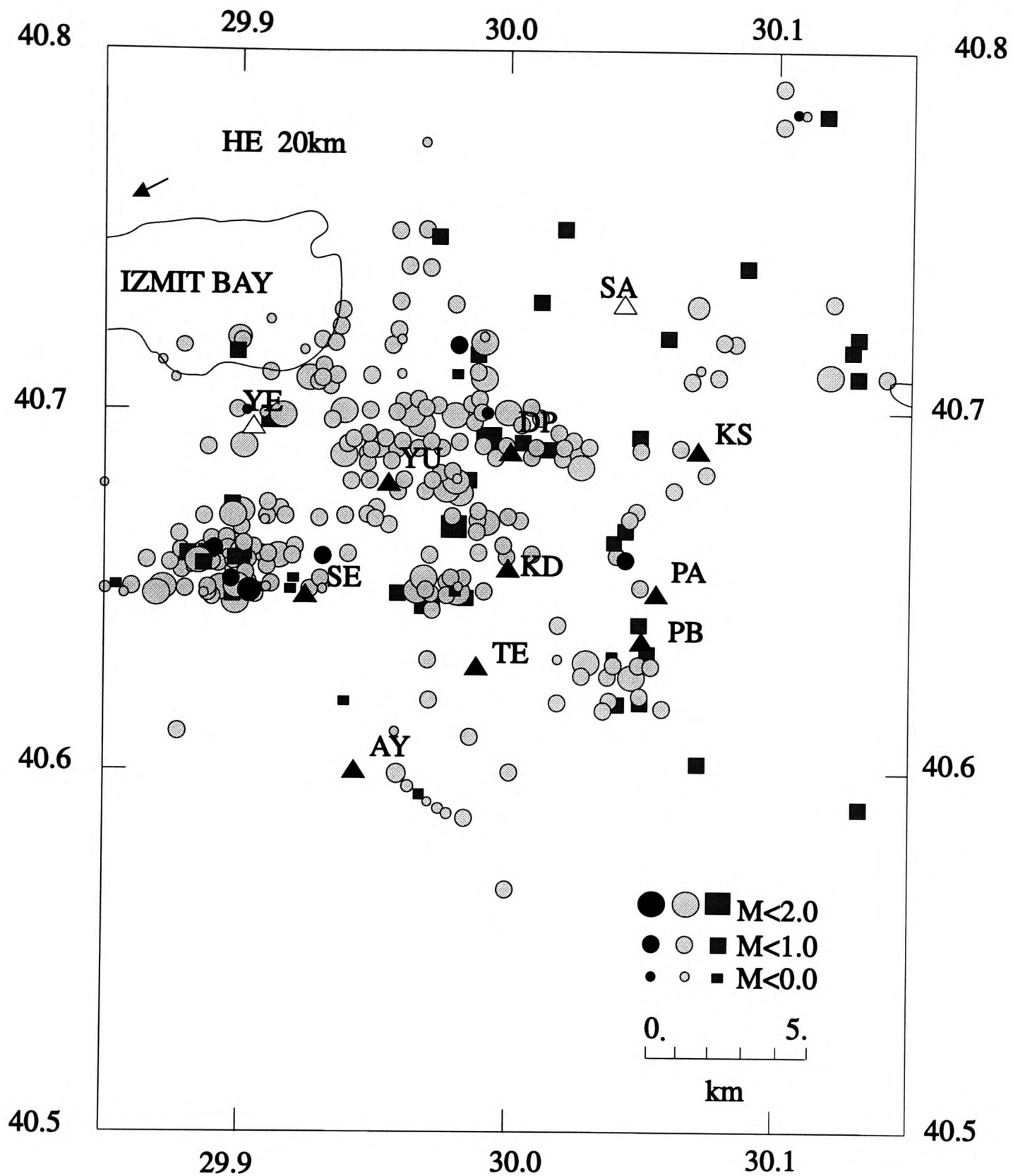


Figure 4.15(d). The location map of events which recorded in station AY. Each symbol represents an event whose S-arrival was manually picked. Grey circles represent S-arrivals which are picked by the BPNN and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals from station DP.

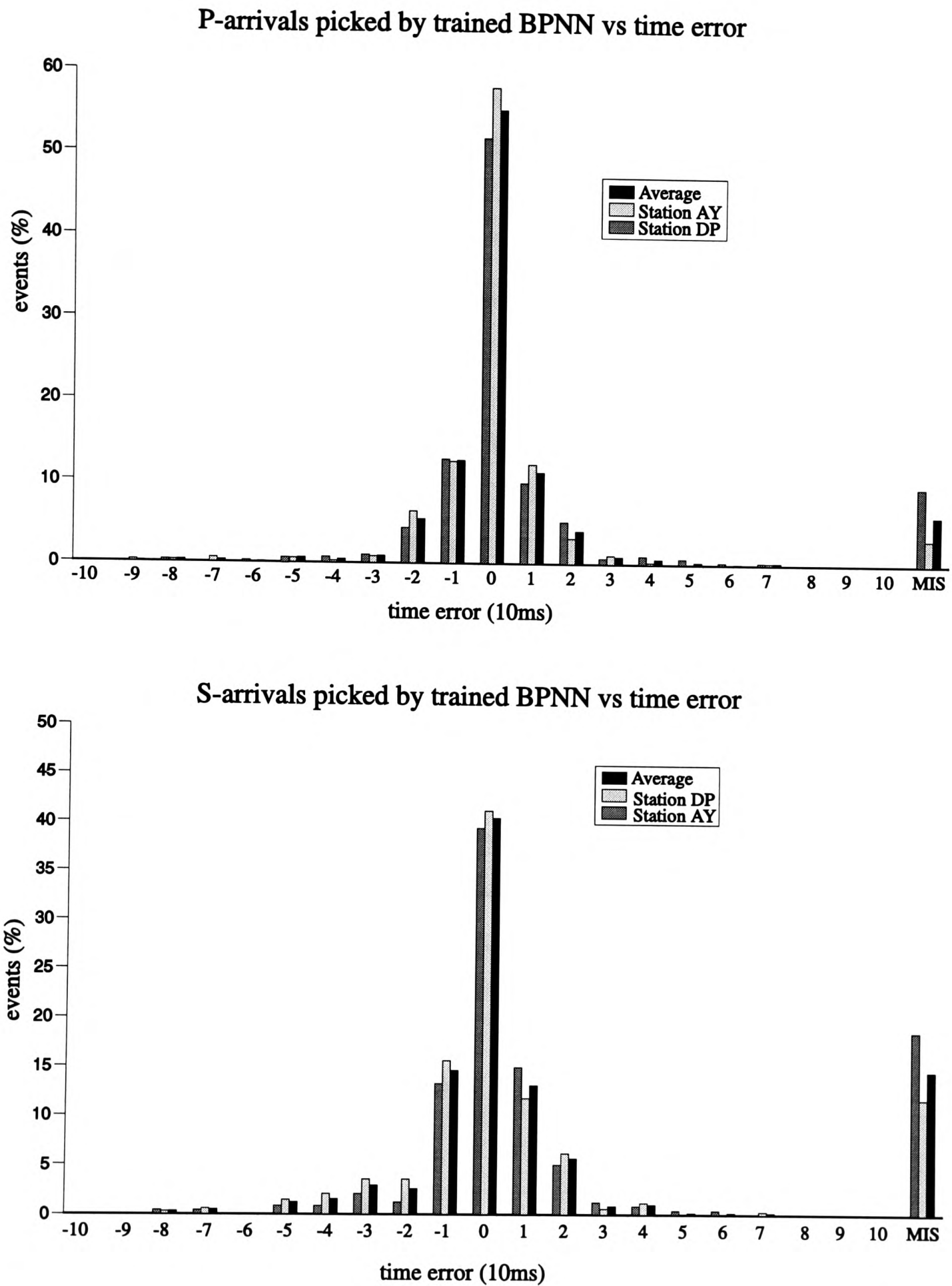


Figure 4.16. Statistical comparison of P- and S-arrivals picked by the BPNN on the complete dataset compared with manual picks, with notations as in Figure 4.8. The success rate of the trained BPNN relative to manual reference picks is quoted as a percentage.

from a smooth background. Reducing the picking threshold of  $N(t)$  might enable the picking of more arrivals. For example, as the threshold is reduced to 0.5, the BPNN can pick 10 more  $P$ -arrivals (7 from station DP and 3 from station AY) and 32 more  $S$ -arrivals (14 from station DP and 18 from station AY). The overall success rates at the 0.5 threshold level are 95.8% (629) for  $P$ -arrivals in which 98.8% (352) are from station DP and 92.3% (277) from station AY, and 91.7% (575) for  $S$ -arrivals in which 92.4% (316) are from station DP and 90.9% (259) from station AY. Among these there are 15 (2 from station DP and 12 from station AY)  $P$ -arrivals and one  $S$ -arrival from station AY with maximum of  $N(t)$  greater than 0.6, but these are discarded by post-processing due to their low amplitude or low mean-SNR. In the cases where manually picking of all  $S$ -arrivals is not possible, a valid comparison with the BPNN results cannot be made, and they are excluded from our statistics.

Although small noise burst and spikes can be discarded by using the post-processing (Section 4.3.7), some noise bursts which are similar to seismic arrivals are still picked by this approach. They might be other arrivals (neither  $P$  nor  $S$ ), traffic noise, artificial events such as exploration shots or sonic booms, so it is difficult for me to give an accurate false alarm rate. But the test result shows its ability to suppress the noise output. For example, with  $N(t)$  threshold 0.6, noise bursts (false alarm) are picked from only 98 (74 from station DP and 24 from station AY) recordings in the 762 recordings. Among the 98 recordings, most of them have only one noise burst detected by the BPNN from an entire recording.

## 4.6 DISCUSSIONS AND SUMMARY

### 4.6.1 Weight pattern analysis

In addition to demonstrating the ability of the trained BPNN to pick seismic arrivals, further insight is required to view the mechanism by which the trained BPNN performs its classification. This is achieved by investigating the weight map which shows the weight pattern of the trained BPNN. Figure 4.17 displays the weight maps

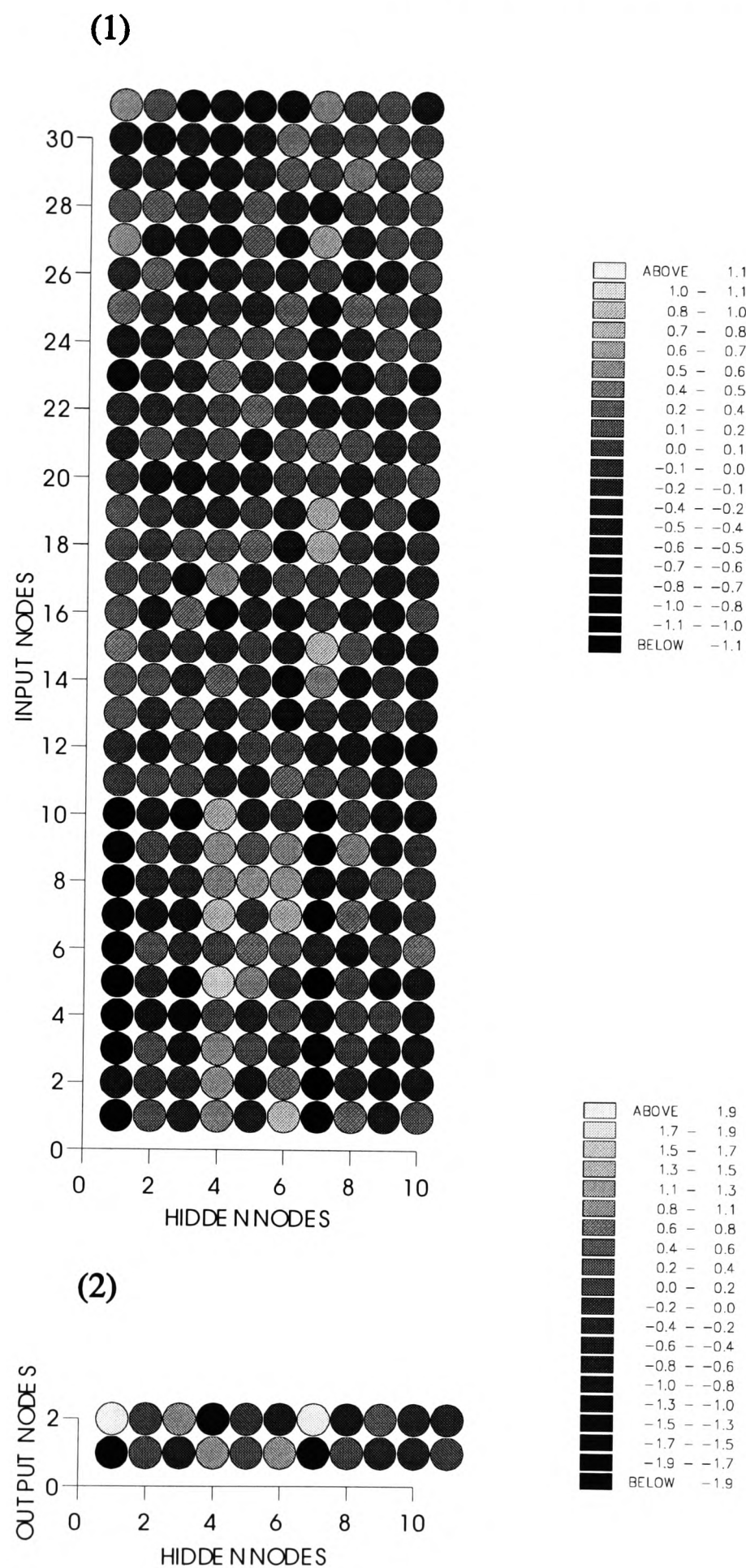


Figure 4.17. Weight map for a trained BPNN. Notations as in Figure 4.4 (a). This BPNN is trained with nine pairs of P-arrival and background noise segments. The training parameters are: learning rate = 0.7, momentum rate = 0.9, system error threshold = 0.00001 and pattern error threshold = 0.0001.



of the BPNN trained with nine pairs of *P*-arrival and background noise segments used in this study. These maps show the magnitudes of each weight connecting the different layers in the BPNN in Figure 4.2. In each weight map, solid circles are shaded on a grey scale corresponding to a magnitude range indicated by the key, with the top row of circles in map (1) and far right row of circles in map (2) representing the nodal thresholds.

Before training, the weights and thresholds are randomly initialized between -0.5 and +0.5, and after training their values range from -2.12 to +2.02. It appears that the pattern of weights connecting the input layer to the hidden layer is divided into two portions at the *tenth* sample, corresponding to the *P*-arrival onsets in the input signal in which before the onset the signal energy is low and after the onset the signal energy is high. The first portion between the *first* and *tenth* nodes shows a "*high contrast*" weight pattern in which the absolute values of most weights are large: for example, most weights connected to the *first* and *seventh* hidden nodes are large negative, and most weights connected to the *fourth* hidden node are large positive. The second portion between the *eleventh* and *thirtieth* nodes shows a "*low contrast*" weight pattern in which the absolute values of most weights are small. The weight pattern between the hidden nodes and the output nodes is related to the pattern between the input node and hidden node. For example, the weights connected to *first*, and *seventh* hidden nodes have large negative values when they connect to the *noise* output node and large positive values when they connect to the *P* output node, and the weights connected to the *fourth* hidden node have a large positive value when they connect to the *noise* node and large negative values when they connect to the *P* node. It seems the *first*, *fourth*, and *seventh* hidden nodes are active in the performance because the weights assigned to them have larger values than others.

According to the sigmoidal function and the input signal which contains all positive values normalized between 1 and 0, a negative weight decreases the node output and a positive weight increases the node output. In the case of a background noise segment being fed into the BPNN, because all values are high and the weights



are large in the first portion, the weights in the second portion will be neglected, but the weights in the first portion will play the principal role in the BPNN output. The *first* and *seventh* hidden nodes give low output and the *fourth* hidden node gives the high output. So the final output in the output nodes will be (high, low) values for the noise signal. However, in the case of a *P*-arrival segment being fed into the BPNN, because of the low values before the *P*-onset, the weights in the first portion can be neglected. The weights in the second portion play the principal role in the BPNN output which will be (low, high) values according to the training. As the sliding window moves along the seismogram, once the onset of an arrival goes into the input window, the values of input samples before the onset decrease, and the first portion will lose its effect on the output which attempts to reach the (low, high) state indicating a *P*-arrival. At the stage of the onset at the eleventh nodes, the output should reach the climax of the (low, high) state. However, as the onset moves into the position before the tenth nodes, the input values before the ten nodes increase, so that the first portion will strengthen its effect on the output which attempts to reach the (high, low) state indicating background noise. Although it is too complex to show the numerical result, this simple analysis shows a clear image of the details of BPNN performance under different conditions.

It is interesting to note that hidden nodes have different activities according to the values of weights connected to them. It seems that some hidden nodes (e.g. the first, fourth, and seventh nodes) can recognize the main features of an input segment and other hidden nodes may improve the BPNN performance by recognizing the subtle features of the input signal. In a comparison between Figure 4.17 and 4.4 (b), these two BPNNs have similar weight patterns, numeric examination shows that they have subtle changes. The changes are so small that the difference between two weight patterns is not obvious, Here, I list their values in Table 4.2 (A and B) for reference. One important feature is that the absolute value of most of weights in the BPNN trained with nine pairs of dataset are larger than those in the BPNN trained with seven pairs of dataset. Another feature is that the threshold of a node is inversely proportional to the weight values connected to it. The more the negative weights, and

**TABLE 4.2. A:** Weight values of a trained BPNN with nine pairs of training datasets.

weight value		hidden node										
		1	2	3	4	5	6	7	8	9	10	
input node	1	-1.00	0.26	-0.17	0.52	-0.23	0.86	-1.05	0.39	-0.28	0.30	
	2	-0.55	0.04	-0.12	0.67	-0.27	0.45	-1.16	-0.04	-0.53	-0.32	
	3	-1.14	0.25	-0.46	0.67	0.29	0.02	-0.73	0.19	-0.32	-0.31	
	4	-1.20	-0.48	-0.63	0.26	-0.10	0.26	-0.41	0.15	0.15	-0.34	
	5	-1.01	-0.01	-0.86	1.02	0.52	0.11	-1.22	0.24	-0.58	-0.32	
	6	-1.26	0.35	-0.11	0.18	0.41	0.29	-0.15	-0.43	0.02	0.43	
	7	-0.87	-0.46	-0.40	0.96	0.12	0.76	-0.85	0.41	-0.25	0.04	
	8	-1.17	-0.03	-0.10	0.56	0.63	0.70	-0.49	-0.22	0.23	0.05	
	9	-0.88	0.21	-0.08	0.63	0.32	0.53	-0.95	0.55	-0.32	0.07	
	10	-1.07	-0.31	-0.74	0.82	-0.16	0.01	-0.91	0.15	-0.40	-0.36	
	11	0.07	0.18	0.22	-0.12	-0.20	0.47	0.16	0.23	-0.44	0.23	
	12	-0.07	-0.30	0.16	-0.26	0.23	0.16	-0.21	-0.33	-0.51	-0.50	
	13	0.27	0.00	0.36	0.00	0.21	0.52	0.08	-0.19	0.22	-0.01	
	14	0.36	0.34	-0.01	0.37	0.03	0.59	0.50	-0.47	0.04	-0.29	
	15	0.39	0.11	0.07	-0.04	0.24	0.21	0.93	0.27	-0.25	-0.29	
	16	0.30	-0.30	0.39	0.53	-0.11	0.33	0.15	-0.22	-0.31	0.28	
	17	0.21	0.22	-0.36	0.52	-0.09	0.31	0.16	0.15	0.24	-0.09	
	18	0.17	0.01	0.30	0.50	0.38	-0.36	0.76	0.03	0.20	0.08	
	19	0.28	0.05	-0.05	-0.01	0.22	-0.22	0.76	-0.14	0.22	-0.39	
	20	0.09	-0.37	-0.38	-0.13	-0.13	0.22	0.14	-0.22	0.19	0.34	
	21	-0.17	0.26	0.04	0.25	-0.35	0.35	0.41	0.30	-0.09	0.07	
	22	-0.08	0.01	0.06	0.20	0.37	0.12	-0.13	-0.19	-0.18	0.09	
	23	-0.71	-0.15	-0.04	0.40	-0.01	0.03	-0.65	-0.25	0.23	-0.34	
	24	-0.34	-0.19	0.27	0.25	0.31	0.26	-0.47	-0.04	0.35	0.23	
	25	-0.45	0.08	-0.14	0.02	0.00	0.38	-0.38	0.39	0.33	0.04	
	26	-0.12	0.44	-0.27	-0.11	0.10	-0.12	0.18	-0.42	-0.24	0.35	
	27	0.62	-0.35	-0.28	-0.26	0.39	-0.30	0.74	-0.04	0.29	0.21	
	28	0.25	0.37	0.19	-0.02	0.47	-0.08	-0.29	0.16	0.13	0.26	
	29	-0.12	0.09	-0.31	-0.16	0.03	0.46	0.25	0.55	0.23	0.47	
	30	-0.32	-0.45	-0.14	-0.33	-0.12	0.56	0.22	0.32	0.26	0.20	
threshold		0.72	0.33	-0.54	-0.31	-0.46	-0.36	0.72	0.18	0.27	0.28	
output node	1	-2.22	0.38	-0.47	-1.12	0.26	1.19	-2.07	0.37	-0.35	-0.23	0.18
	2	2.02	0.12	1.03	-1.54	0.03	-0.46	2.22	-0.63	0.55	-0.27	-0.61
weight value		1	2	3	4	5	6	7	8	9	10	thre- shold
hidden node												

**TABLE 4.2. B:** Weight values of a trained BPNN with seven pairs of training datasets

weight value		hidden node										
		1	2	3	4	5	6	7	8	9	10	
input node	1	-0.96	0.21	-0.19	0.57	-0.32	0.91	-1.06	0.45	-0.33	0.31	
	2	-0.43	0.00	-0.08	0.68	-0.33	0.45	-1.00	-0.02	-0.55	-0.32	
	3	-0.83	0.27	-0.32	0.55	0.27	-0.10	-0.43	0.14	-0.30	-0.31	
	4	-1.10	-0.47	-0.61	0.33	-0.12	0.32	-0.43	0.22	0.08	-0.33	
	5	-0.83	-0.05	-0.80	1.02	0.44	0.14	-1.01	0.25	-0.59	-0.32	
	6	-1.06	0.38	-0.03	0.18	0.40	0.26	-0.03	-0.43	-0.02	0.43	
	7	-0.71	-0.48	-0.32	0.88	0.04	0.75	-0.67	0.40	-0.24	0.02	
	8	-1.13	-0.03	-0.13	0.67	0.62	0.82	-0.64	-0.12	0.14	0.06	
	9	-0.73	0.21	0.01	0.56	0.28	0.48	-0.78	0.52	-0.31	0.06	
	10	-0.96	-0.31	-0.72	0.85	-0.21	0.06	-0.92	0.18	-0.44	-0.37	
	11	0.12	0.16	0.22	-0.10	-0.24	0.50	0.17	0.25	-0.45	0.24	
	12	-0.04	-0.30	0.17	-0.31	0.19	0.14	-0.27	-0.32	-0.52	-0.51	
	13	0.34	0.02	0.41	-0.09	0.21	-0.62	0.05	-0.22	0.23	-0.01	
	14	0.65	0.39	0.12	0.20	0.03	-0.77	0.73	-0.59	0.14	-0.31	
	15	0.54	0.13	0.16	-0.18	0.22	-0.28	0.88	0.20	-0.20	-0.31	
	16	0.22	-0.27	0.39	0.46	-0.10	-0.31	0.08	-0.20	-0.33	0.28	
	17	0.29	0.16	-0.33	0.42	-0.19	0.28	0.20	0.13	-0.21	-0.11	
	18	0.29	0.03	0.36	0.27	0.38	-0.50	0.76	0.00	-0.18	0.08	
	19	0.30	0.10	-0.05	0.03	0.25	-0.23	0.67	-0.14	0.22	-0.40	
	20	-0.04	-0.40	-0.41	-0.06	-0.15	0.26	0.06	-0.21	0.21	0.35	
	21	-0.33	0.24	-0.01	0.29	-0.38	0.43	0.19	0.34	-0.11	0.06	
	22	-0.09	-0.04	0.02	0.31	0.34	0.21	-0.14	-0.18	-0.14	0.09	
	23	-0.55	-0.18	0.00	0.33	-0.09	0.03	-0.61	-0.26	0.24	-0.35	
	24	-0.23	-0.17	0.41	0.14	0.31	0.12	-0.20	-0.13	0.43	0.22	
	25	0.24	0.09	-0.17	0.13	0.01	0.46	-0.54	0.42	0.33	0.03	
	26	-0.09	0.39	-0.31	-0.10	0.03	-0.09	0.07	-0.38	-0.24	0.36	
	27	0.56	-0.31	-0.35	-0.05	0.44	-0.16	0.44	0.01	0.28	0.21	
	28	0.56	0.38	0.36	-0.21	0.43	-0.25	0.07	0.04	0.23	0.25	
	29	0.21	0.10	-0.19	-0.32	-0.01	0.30	0.25	0.46	0.32	0.47	
	30	-0.35	-0.46	-0.13	-0.29	-0.14	0.60	0.21	0.31	0.30	0.18	
threshold		0.62	0.34	-0.56	-0.17	-0.47	-0.29	0.56	0.23	0.25	-0.27	
output node	1	-1.88	0.12	-0.53	-1.13	-0.03	1.49	-1.86	0.49	-0.37	-0.26	0.27
	2	1.73	0.32	1.06	-1.49	0.28	-0.74	2.02	-0.68	0.54	-0.30	-0.72
weight value		1	2	3	4	5	6	7	8	9	10	threshold
hidden node												

the larger their absolute values, the larger the threshold is.

Although it is not possible to fully understand the logic underlying the BPNN solution by visually inspecting these weight patterns, and the weight combinations in this map are not as easily distinguished as the parameter estimates, the weight map may still lend some perspective as to which aspects of the seismic waveform are more relevant to the solution than others, and thus be of benefit to further processing schemes or studies.

#### 4.6.2 Effect of post-processing

The purpose of the post-processing described in Section 4.3.7 is to discard non-seismic arrivals, especially non-*P*- or non-*S*-arrivals, so that the false alarm rate can be decreased to a minimum. In the post-processing, only small noise bursts and spikes are considered, but other kinds of noise can be discarded by using similar procedures if their particular characteristics can be described. Using the present post-processing techniques, spikes can be firmly discarded. A failure of this process is that a small spike and an arrival may interfere so that discarding this signal means losing the arrival. Another failure could occur if two spikes are too close. However, it should not be a problem in this dataset because there are only two cases in the 762 recordings. To discard the small noise bursts is somewhat problematical because some small *P*- and *S*-arrivals are also discarded by this post-processing. The result shows that the mean-SNR criterion is more effective than mean-amp criterion. Particular values should be settled for them according to the recording quantity and the background noise level. Although the post-processing does not directly improve the ability to pick up the *P*- and *S*-arrivals, the  $N(t)$  threshold can be decreased so that more seismic arrivals can be picked with most false alarms being discarded by post-processing.

#### 4.6.3 Comparison with other picking algorithms

As discussed in Section 4.2, there are many picking algorithms already in use on many seismic networks worldwide. Table 4.3 gives a comparison of the performance of my technique for local earthquake data and the performance of a few

**Table 4.3:** A summary comparison of selected picking methods

Author	Input data	Method	Wave type	Picking result	Time error
Allen (1978)	single trace	STA/LTA	P	60-80%	$\leq 0.05$ sec
Baer & Kradolfer (1987)	single trace	modified STA/LTA	P	local: 65.9% Region: 79.5% Tele: 90%	$\leq 1$ sample $\leq 3$ sample
Joswig & Shulte-Theis (1993)	single trace	Master-event correlation	P	80% for weak events	$\leq 1$ sample
Klumpen & Joswig (1993)	3-component recordings	generic polarization	P & S	67% for P & S	$\leq 50$ ms
Kracke (1993)	modulus of 3-component recordings	LTA/threshold	P	96.5%	not mentioned
Dai & MacBeth (1995, this thesis)	modulus of 3-component recording	Artificial neural network	P & S	94% for P 86% for S 74% for P 63% for S	$\leq 100$ ms $\leq 100$ ms $\leq 10$ ms $\leq 10$ ms

selected techniques in common use. Because articles tend to describe principles and show a few examples, these cannot be directly or wholly compared with my result which is applied to a specific dataset of local events, so that this table may not be truly representative of the optimal forms of each technique. As false alarms were not fully treated in my algorithm, I do not suggest that, without further tests and development, this method is better. However, it does appear that the small estimated error for both  $P$ -arrival and  $S$ -arrival analysis is potentially encouraging for future work. I believe that an additional strength of the BPNN is that it can deal with raw data. This contrasts with many other techniques, which rely upon complex mathematical methods to generate control parameters from raw data. The BPNN presented here is relatively quick to train and has been shown to be adaptive to various types of arrival.

#### 4.6.4 Summary

A BPNN is used as a tool to pick  $P$ - and  $S$ -arrivals from local earthquake data. This is achieved by utilizing the vector modulus of their 3-C recordings as the BPNN input. A discriminant function,  $N(t)$ , determined from the output of the trained BPNN, is then employed to define the arrival onset. 762 recordings from two stations in a local earthquake network are analyzed by a BPNN trained with only nine pairs of  $P$ -arrivals and background noise segments. Compared with manual picks, the results are encouraging, and demonstrate that a BPNN trained with a small subset of the data can detect 94.3% of the  $P$ -arrivals and 86.4% of the  $S$ -arrivals. Using this to further pick the onset times, the success rates are 74.5% for the  $P$ -arrivals and 63.2% for the  $S$ -arrivals with an error of 10ms ( $N(t)$  threshold  $> 0.6$ ). The ability of this BPNN, trained with only  $P$ -arrivals and background noise from station DP, is extended to picking  $S$ -arrivals and to picking arrivals from station AY. Even though all the training dataset have high SNRs, the BPNN still works for seismograms with low SNRs. This illustrates the adaptiveness of the BPNN to various types of arrival. To fully demonstrate this feature, and avoid data-specific results, further tests are required on

the BPNN, to extend its application to other 3-C dataset and 1-C dataset. It may be also possible to process regional or teleseismic recordings using this approach. In these cases, however, the BPNN architecture may need to be adjusted to suit the behaviour of these data.

These results, combined with the advantage of not requiring programs to construct special variables and parameters with complicated mathematics, suggest that the BPNN is a natural choice for such applications. All that is necessary is to select suitable example arrivals in a training set. The method is adaptive, and training sets can be altered to enhance particular features of different dataset. Retraining the BPNN by adding new training dataset is easy and quick, and can improve its performance. Although the training time can be long, especially if the BPNN architecture becomes large, once trained, the BPNN is sufficiently quick to operate in most real-time applications and its performance is objective and consistent. However, its performance depends upon the training set and its ability to predict cannot lie too far outside its experience. The exact boundaries of this behaviour have not yet been completely explored.

However, a pre-processing stage is required to discard excessively noisy and unusable recordings. They are not included in the statistics which I quote. For the dataset used in this thesis, although small noise bursts and spikes can be discarded by using post-processing, 12.9% of the detections include false alarms. Inspection of the seismograms reveals that most of the false alarms are similar to *P*- or *S*-arrivals, and in fact it would not be possible to distinguish them visually if only one segment is available. Additional information is required for this task. These needs to be discriminated at a later stage.

The experiment on selecting training parameters shows that the performance of a trained BPNN is mainly determined by the training dataset. Changing the training parameters does affect the convergence of the learning procedure which might be important in the case of a large BPNN and a large training dataset is involved. However it does not affect the weight pattern of the trained BPNN and so its performance should be similar. This will allow us to devote our attention only to the

performance of the trained BPNN. It can make a complex problem simple. Although an optimum architecture exists for a particular application, finding such an optimum architecture is dependent on operator's experience and knowledge of the problem, and sometimes it is difficult. This limitation may be due to the performance of the BPNN being not yet fully understood. However, certain other ANN designs might be a solution (Fahlman and Lebiere 1990; Kusuma and Brown, 1992).

#### **4.6.5 Future work**

This approach shows that the BPNN has potential as a tool to pick seismic arrivals automatically from the vector modulus for 3-C recordings. However, 3-C recordings are not always available. There is a need to pick seismic arrivals from 1-C recordings. This current approach, limited to 3-C component recordings, should therefore be adapted for 1-C recordings. In Chapter 5, an BPNN approach is developed to tackle this. I should point out that this approach and that developed in Chapter 5 do not attempt to identify the types of picked arrival. However, in seismic analysis, once arrivals are picked, they need to be classified according to their properties. This is the second stage in an automatic analysis system. In Chapter 6, I will develop another BPNN approach to tackle this.



## CHAPTER 5:

### ARRIVAL PICKING FROM SINGLE-COMPONENT RECORDINGS USING BPNN

#### 5.1 INTRODUCTION

In Chapter 4, I developed an approach which used a BPNN to pick seismic arrivals from the vector modulus of 3-C recordings of local earthquake data. This BPNN, similar in operation to the procedure adapted by an analyst, is trained by presenting some different *P*-arrival and background noise. After training is accomplished, it can recognize new arrivals from a variety of new seismograms. A BPNN trained with nine pairs of *P*-arrival and background noise segments can pick 94% of *P*-arrivals and 86% of *S*-arrivals. Unfortunately, not all seismic stations have 3-C seismometers or can provide consistently high quality 3-C recordings due to the failure of one or more components. There is therefore a practical need to pick seismic arrivals from 1-C recordings in these situations.

There is no shortage of techniques in the literature which claims to pick seismic arrivals from the vertical component recordings. However, as I mentioned in Section 4.2, most of them are conventional methods which are not adaptive, working well only under certain conditions, but quite often not producing good results. As the BPNN has shown a good performance in picking seismic arrivals from 3-C recordings, I shall apply this BPNN approach to pick seismic arrivals from 1-C recordings, i.e. from Vertical component (V-C), East-West component (E-W) or North-South component (N-S). In this chapter, a modified version of the BPNN approach is developed to tackle this problem. This is achieved by utilizing the absolute values of 1-C recordings as the BPNN input. As the 1-C recordings have different overall characteristics from the 3-C recordings, the original approach might not give rise to an equally good performance in picking arrivals from 1-C recordings. In fact since the

information content is reduced, it is quite likely to be inferior. I must, therefore, choose a different BPNN structure, different training dataset and different post-processing procedures in this approach.

## 5.2 APPROACH OF PICKING ARRIVALS FROM 1-C RECORDINGS

The approach used in this chapter is similar but not equivalent to that used in Chapter 4. A BPNN is trained with a small number of training samples and is then used as a filter acting on the entire seismogram by sliding a window along the seismogram. The output of the BPNN yields a time series which is to be interrogated for a decision regarding the seismic arrivals. This approach is now adapted to the problem of picking arrivals from 1-C recording. As the performance of this method depends on the input characteristics, the training dataset, and the BPNN's structure, especially the input node number, it is necessary to modify them so that a better performance is obtained.

### 5.2.1 Characteristics of 1-C recordings

In contrast with 3-C recordings, 1-C recordings only represent behaviour of the particle motion of a seismic signal in orthogonal directions. Although 1-C recordings are strongly dependent on the source position and ray direction, the arrival of a seismic signal may still be observed on each separate 1-C recording by changes in the relative amplitude, frequency, and polarization characteristics. Despite the fact that complete polarization and propagation information cannot be obtained from 1-C recordings, various characteristics such as amplitude and frequency information might be obtained from them by taking different transformations. In Section 3.3.1, I have discussed five basic characteristic functions. Here I choose the *absolute value function* as the input signal of the BPNN because it has the highest fidelity and processing speed and is most objective amongst the five functions. I do not use the 1-C recording itself because the first motion of an arrival has two directions (up and down) and is source dependent. Figure 5.1 schematically shows the method with this modification. The amplitude of 1-C recordings is also strongly dependent on the magnitude and

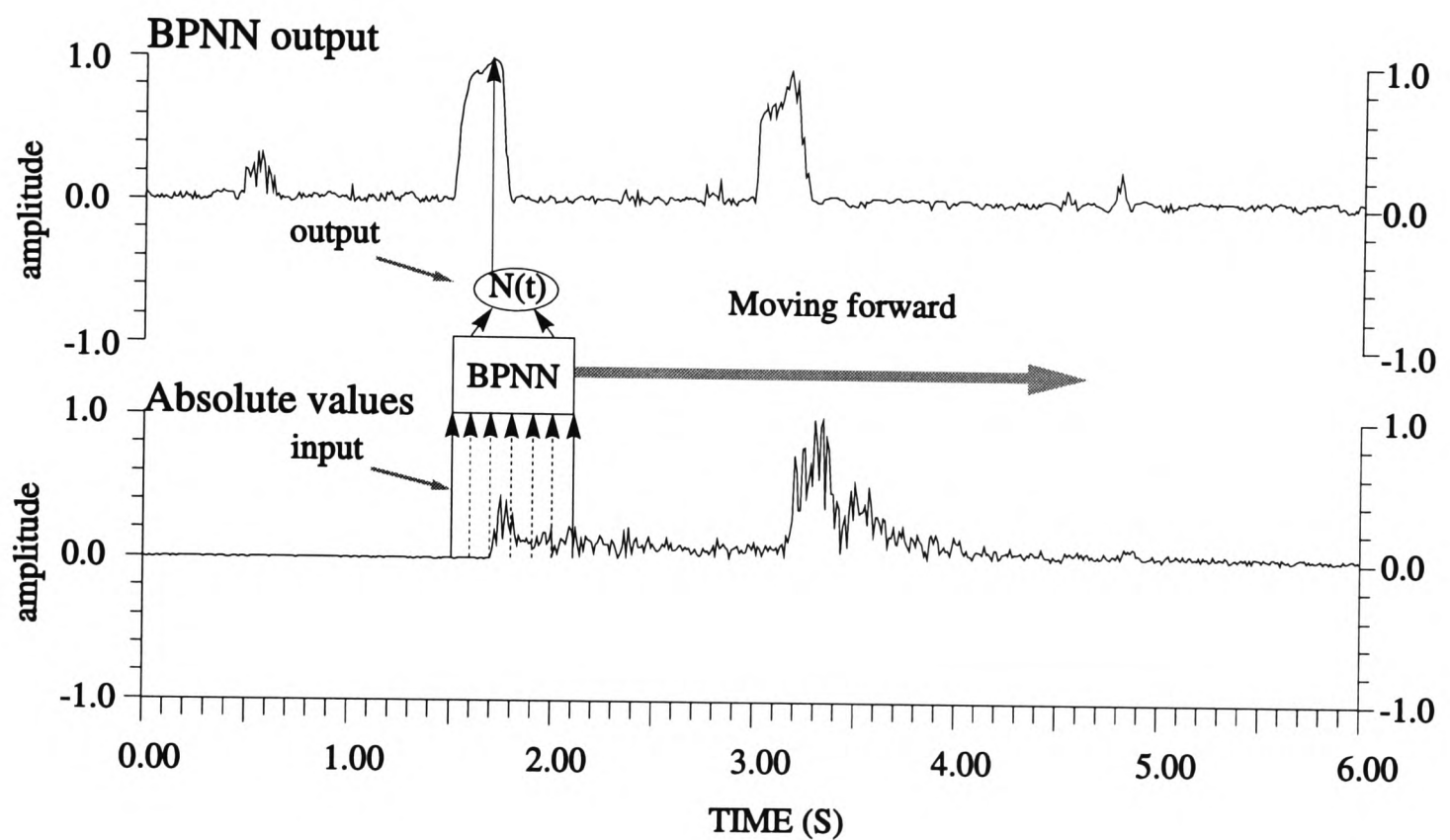


Figure 5.1. Schematic diagram showing the method of using a BPNN to pick seismic arrivals. Here, the input of the BPNN is the absolute value of the 1-C recordings (lower diagram). A trained BPNN is treated as a filter moving across the entire 1-C recording by a sliding window. The output of the BPNN (upper diagram) yields a time series which enhances the changes in the modulus to indicate the arrival onsets.

epicentral distance of an earthquake. In order to utilize a small training dataset to cover all the recordings with different amplitudes, each segment of absolute values is individually normalized before it is fed into the BPNN. Usually, the V-C recordings are used for picking arrivals in many other methods, so my first attempt is to use this approach to deal with the V-C recordings. The training dataset must be reselected from the V-C recordings of local seismic data because the onset times of *P*-arrivals in V-C recordings, sometimes, are different from those in the vector modulus.

### 5.2.2 BPNN structure

As the input signal is changed, the BPNN structure need to be changed to suit this problem too. As I discussed in Section 4.6.4, for each problem there exists an optimum BPNN, but finding it might be difficult. There is no fixed rule to select the structure and it must be determined by a process of trial and error. At the beginning, I still employed a BPNN with the same structure as that in Chapter 4 which has three layers with 30 nodes in the input layer, 10 nodes in the hidden layer and two nodes in the output layer. The input of this BPNN is a segment of normalized absolute values of the V-C recordings of local earthquake data with length 290ms (30 sample nodes). Two output nodes of the BPNN flag the input segment with (0,1) for *P*-arrivals and (1,0) for background noise. If this trained BPNN does not have a good performance, I shell optimize its structure to improve its performance in subsequent experiments.

### 5.2.3 Rules of picking arrivals

In this approach, during the learning procedure, several pairs of *P*-arrival and pre-event background noise segments are used together to train the BPNN by using the *Delta Rule*. After training, the BPNN acts on the entire 1-C recording by sliding a window along the absolute values trace. The resulting output is a time series  $N(t)$ :

$$N(t) = \frac{1}{2}[(1-o_1(t))^2 + o_2(t)^2] \quad (5.1)$$

which is to be interrogated for a decision regarding arrivals. This function exaggerates the difference between the desired output and the ideal background noise. A peak in  $N(t)$  corresponds to a characteristic change in  $M(t)$ . If the change is similar to a training  $P$ -arrival, the peak value should be close to 1. Usually, the peak value indicates the similarity between a change and a training  $P$ -arrival. The same rules are used to detect and pick arrivals as in Chapter 4:

- 1) detect an arrival using a simple threshold rule on  $N(t)$ .
- 2) pick the arrival onset time using the local maximum of  $N(t)$ .

#### 5.2.4 Post -processing procedures

Some post-processing procedures are also employed in this approach to discard noise bursts and spikes as in Chapter 4. In order to discard small noise bursts, the same two criteria are used but the threshold of the mean-amplitude criterion is reduced to 10.0 because the mean amplitude of the modulus of 3-C recordings is  $\sqrt{3}$  times the mean amplitude of 1-C recordings. In order to discard the spikes, only the spike-amplitude-ratio criterion is used. The degree of polarization criterion is not used because it cannot be obtained from 1-C recordings.

### 5.3 OPTIMIZING BPNN STRUCTURE AND TRAINING DATASET

In this approach, the training parameters are the same as those used in Chapter 4. Their values are:

learning rate = 0.7

momentum rate = 0.9

system error threshold = 0.00001

pattern error threshold = 0.0001

However, the BPNN structure and training dataset may need to be varied to suit this specific problem. In this section I optimize them by a process of trial and error.

### 5.3.1 Optimizing BPNN structure

At the beginning, the absolute values of V-C recordings of the same  $P$ -arrival and background noise segments are used to train a BPNN which is the same as that used in Chapter 4, with 30 input nodes, 10 hidden nodes and two output nodes. The training  $P$ -arrivals are positioned so that their arrival onset times are fixed at the eleventh sample as in Chapter 4. The training procedure takes 217 iterations (about half minute of interactive CPU time on a VAX 4000). After training, this trained BPNN is used to process some complete V-C recordings. Unfortunately, its performance is not good. In contrast with the output  $N(t)$  from the modulus of 3-C picking (Figure 4.7),  $N(t)$  from this BPNN has many peaks corresponding to false alarms (Figure 5.2 (a)) which, nevertheless, might be discarded by post-processing. In order to improve its performance, one method is to change the BPNN structure. The results from Chapter 4 show that the number of input nodes strongly affects the performance, and the number of hidden nodes does not. I need only to optimize its structure by changing the number of input nodes corresponding to the window length of the input segment. Several experiments have shown that as the number of input nodes is increased from 30 to 40, the BPNN trained by the same dataset with the  $P$ -onset time now at the twenty-first sample point has a much better performance than the previous one. Figure 5.2 (b) shows an example for the same recordings in which the  $N(t)$  has a clearly smooth background with only two large peaks corresponding to the  $P$ - and  $S$ -arrivals. Here, I am satisfied with the present output  $N(t)$  of the BPNN. So the BPNN used in this approach has a three-layer structure with 40 input nodes, 10 hidden nodes and 2 output nodes (Figure 5.3).

### 5.3.2 Optimizing the training dataset

Although the above BPNN shows a good output  $N(t)$ , its ability of detecting and picking arrivals is not good enough when it is used to process some other seismograms. As the performance of a trained BPNN depends on the training dataset too, I reselected suitable training dataset by a process of trial and error. Finally, ten

Station: DP  
 Date: 1984-07-04  
 Time: 00h16m19s  
 Scale: 377

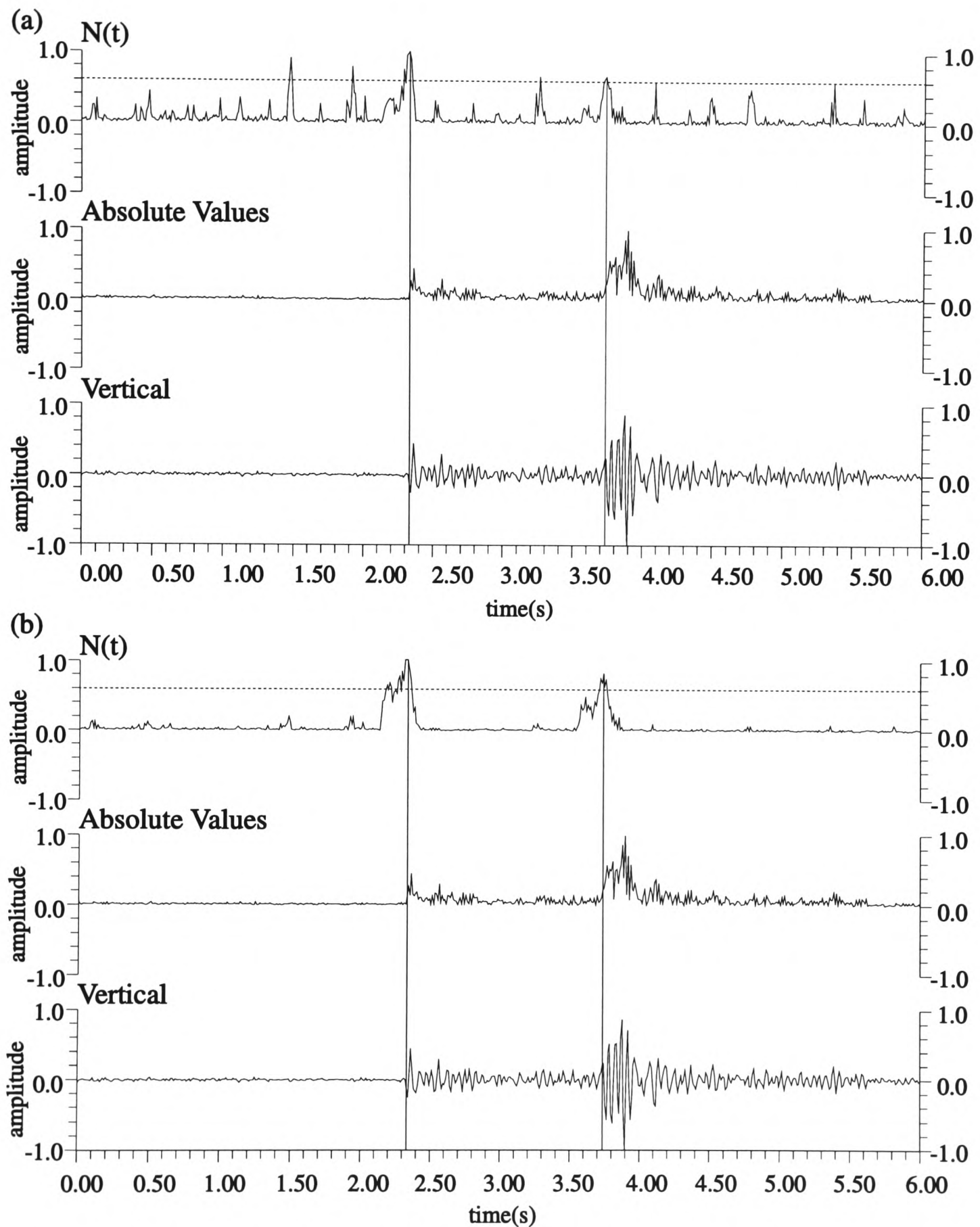


Figure 5.2. Two examples of the BPNN output  $N(t)$ , vertical component recording and its absolute values. Vertical lines are drawn automatically by this approach. Dashed lines are picking threshold (0.6) applied to  $N(t)$ . (a): BPNN has 30 input nodes. (b): BPNN has 40 input nodes. Both are trained with nine pairs of P-arrival and background noise segments.

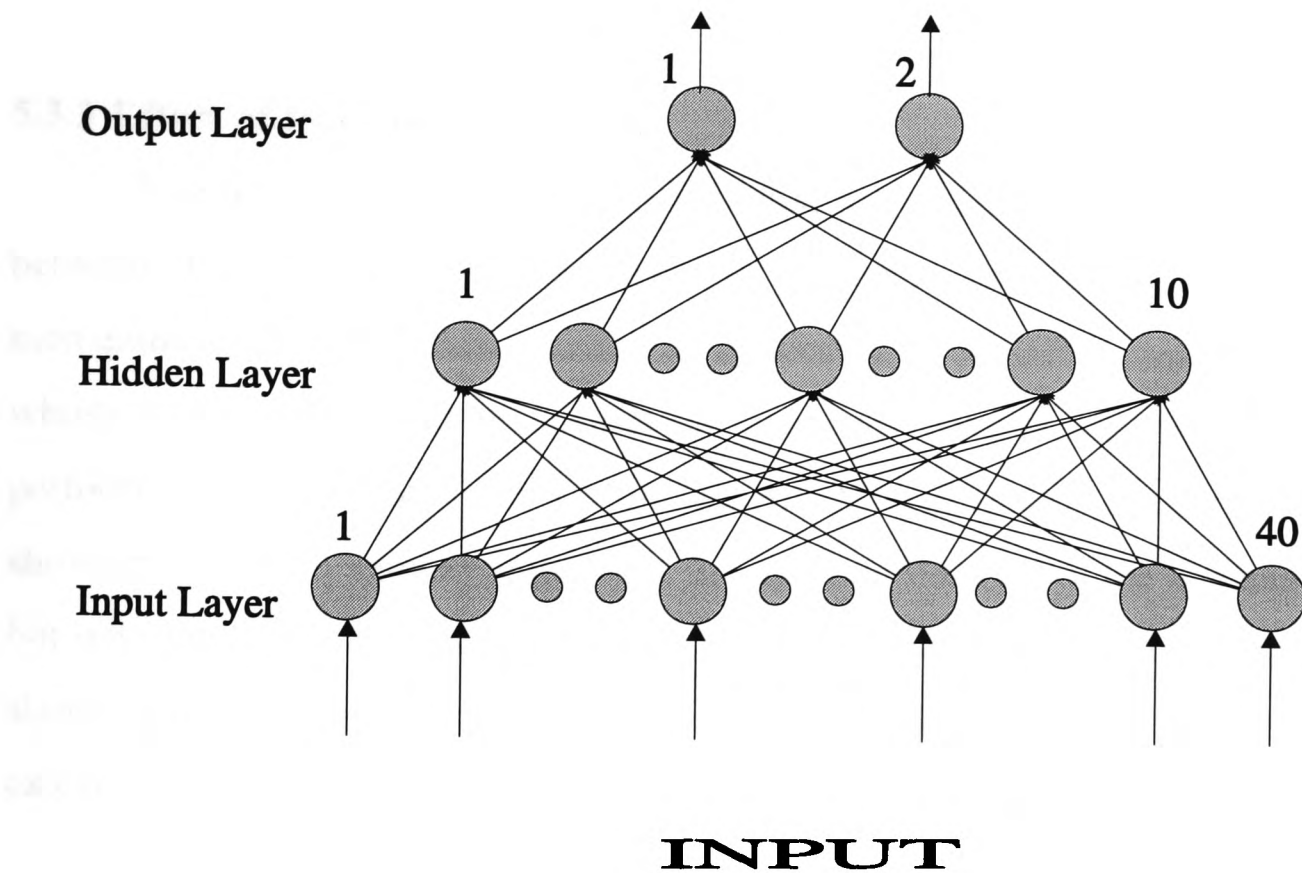


Figure 5.3. The structure of a BPNN for seismic arrival picking from 1-C recordings. It has three layers including an input layer with 40 nodes, a hidden layer with 10 nodes and an output layer with two nodes. The BPNN input is a segment of absolute values of 1-C recordings. The two output nodes indicate the input segment with (0,1) as a P-arrival and (1,0) as background noise.



pairs of *P*-arrival and background noise segments from station DP shown in Figure 5.4 are used to train the above BPNN. The training procedure took 173 iterations with a system error of 0.000017 and all pattern errors less than 0.0001. Figure 5.5 show the result of using this BPNN to process the same recording as in Figures 5.2. The shape of peaks of  $N(t)$  shows that this BPNN is better trained than that in Figure 5.2 (b).

### 5.3.3 Effect of learning parameters

In Section 4.3.3.2, I obtained Equation 4.1 which represents the relationship between the convergence of the learning procedure and the learning rate  $\eta$  and momentum rate  $\alpha$ . Can this equation be applicable to the BPNN used in this chapter whose structure and training dataset are changed? Several experiments have been performed to verify Equation 4.1 for the BPNN used in this Chapter. Table 5.1.A shows the iteration number required in training as a function of the different  $\eta$  and  $\alpha$ , but with the same thresholds of system error (0.00001) and pattern error (0.0001). It shows that Equation 4.1 can be applied to this BPNN. Table 5.2.B also shows the calculated result which is fit with the data in Table 5.1.A. It should be noted that the fit is better when  $\eta$  and  $\alpha$  are smaller. The error between the real iteration and the calculated one is mainly proportional to  $\eta$ . Compared with the result from Chapter 4 (Table 4.1), in both cases, the real iteration numbers are greater than the calculated ones.

## 5.4 TESTING ON COMPLETE DATASET

After optimizing the BPNN's structure and training dataset, the trained BPNN is applied to the same dataset including 762 recordings (371 from station DP and 391 from station AY) which was used in Chapter 4. Due to the raypath, some *P*-arrivals and some *S*-arrivals are absent from 1-C recordings, especially from small events, and some arrivals have different onset times on different 1-C recordings. This means that I must manually pick the *P*- and *S*-arrival onset times from three 1-C recordings again and use them as references to compare with the BPNN's picking results. The trained

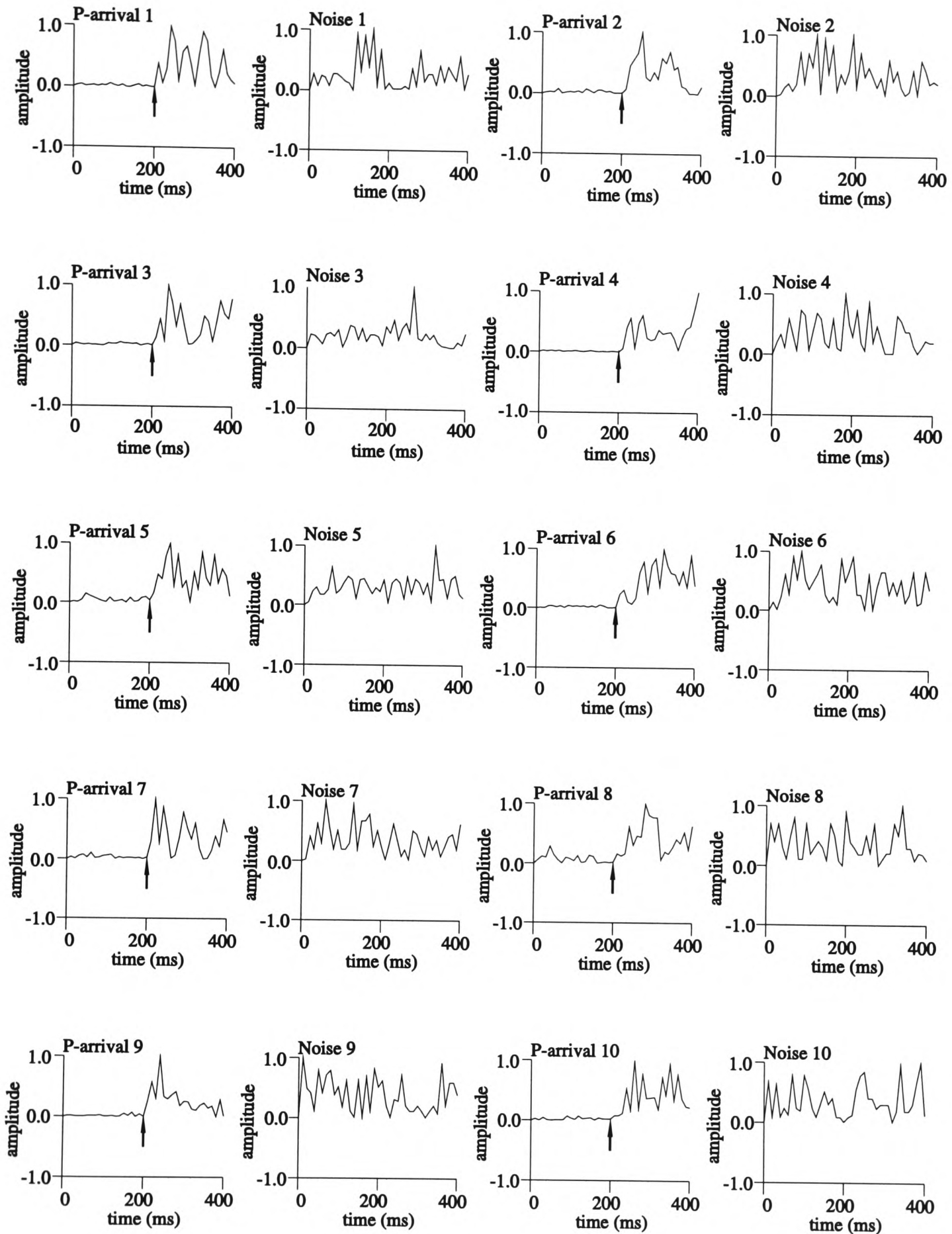


Figure 5.4. Ten pairs of P-arrival and background noise segments of absolute values of V-C recordings used for training a BPNN. Noise segments are extracted prior to the P-arrivals in the same seismograms. Arrows on P-arrival segments indicate arrival onset times used to train the BPNN, all are at the twenty first sample. These segments are individually normalized before being fed into the BPNN.

Station: DP  
Date: 1984-07-04  
Time: 00h16m19s  
Scale: 377

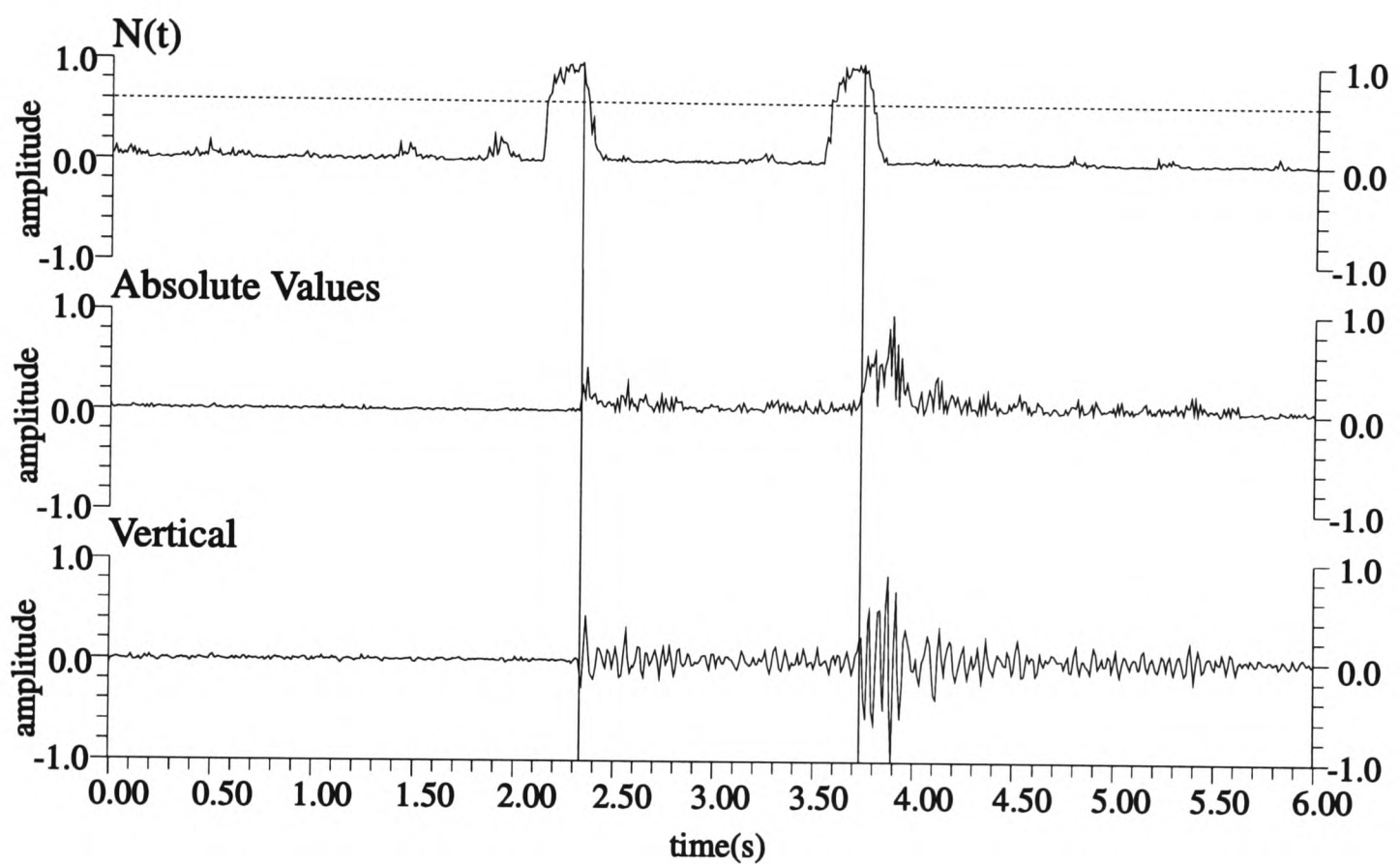


Figure 5.5. The BPNN output  $N(t)$ , V-C recording and its absolute values of a local earthquake. Vertical lines are drawn automatically by this approach. Dashed line is the picking threshold (0.6) applied to  $N(t)$ . The BPNN has 40 input nodes and is trained with ten pairs of P-arrival and background noise segments as in Figure 5.4.

**Table 5.1.A** The iteration number in training with different learning rate  $\eta$  and momentum rate  $\alpha$ . (NC means non-convergence)

Iteration number		$\alpha$										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\eta$	0.1	10134	9124	8116	7108	6098	5089	4086	3069	2058	1045	NC
	0.2	5093	4588	4084	3579	3074	2569	2063	1557	1050	541	NC
	0.3	3416	3079	2741	2404	2067	1730	1393	1054	714	371	NC
	0.4	2580	2326	2072	1819	1565	1312	1058	804	548	292	NC
	0.5	2081	1876	1672	1469	1265	1062	859	655	450	240	NC
	0.6	1751	1579	1407	1237	1067	897	726	556	385	203	NC
	0.7	1518	1368	1219	1072	925	779	632	486	338	173	NC
	0.8	1346	1212	1080	950	820	691	563	434	300	153	NC
	0.9	1214	1092	973	856	739	624	509	393	272	144	NC
	1.0	1111	998	889	782	675	570	466	359	253	136	NC
	1.1	1029	923	821	722	624	527	430	331	243	126	NC

**Table 5.1.B** The calculated iteration number by using Equation 4.1

Iteration number		$\alpha$										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\eta$	0.1	10134	9120	8107	7093	6080	5067	4053	3040	2026	1013	---
	0.2	5067	4560	4053	3547	3040	2533	2026	1520	1013	506	---
	0.3	3378	3040	2702	2364	2026	1689	1351	1013	675	338	---
	0.4	2534	2280	2026	1773	1520	1267	1013	760	506	253	---
	0.5	2027	1824	1621	1418	1216	1013	810	608	405	202	---
	0.6	1689	1520	1351	1182	1013	844	675	506	338	169	---
	0.7	1448	1302	1158	1013	868	723	579	434	289	145	---
	0.8	1267	1140	1013	886	760	633	506	380	253	126	---
	0.9	1126	1013	901	788	676	563	450	337	225	112	---
	1.0	1013	912	811	711	608	507	409	304	203	101	---
	1.1	921	829	737	644	552	460	360	276	184	92	---

BPNN will be used to deal with three 1-C recordings respectively.

#### 5.4.1 Testing on vertical component recordings

In total, 648 *P*-arrivals (350 from station DP and 298 from station AY) and 509 *S*-arrivals (334 from station DP and 175 from station AY) are visually picked from these V-C recordings. Using a  $N(t)$  threshold value of 0.6, the BPNN detected 603 (93.1%) of the *P*-arrivals, 328 (93.7%) and 275 (92.2%) from stations DP and AY, and 382 (75.0%) of the *S*-arrivals, 289 (86.5%) and 93 (53.1%) from stations DP and AY, respectively. Figure 5.6 shows the picking results related to event positions. A total of 45 (18 from station DP and 27 from station AY) *P*-arrivals and 124 (42 from station DP and 82 from station AY) are missed by this approach. Among these missed arrivals, 36 (11 from station DP and 25 from station AY) *P*-arrivals and 6 (one from station DP and 5 from station AY) *S*-arrivals are picked by the BPNN when their outputs are greater than 0.6, but discarded by post-processing due to their low amplitude, low SNR or low spike-amplitude-ratio. However, these arrivals have clear first motions and can be picked visually. 7 *P*-arrivals (4 from station DP and 3 from station AY) and 16 *S*-arrivals (3 from station DP and 13 from station AY) are picked but with error larger than 100ms (ten samples increments). Other missed arrivals with low BPNN outputs do not have clear first motions so that visual picking is also difficult. If the  $N(t)$  threshold is reduced to 0.5, this method can pick two more *P*-arrivals from station DP and 39 (17 from station DP and 22 from station AY) more *S*-arrivals. The picking rates for this are 93.4% for *P*-arrivals (94.3% from station DP and 90.2% for station AY), and 82.7% for *S*-arrivals (91.6% from station DP and 65.2% from station AY).

Figures 5.7 shows the comparison of the trained BPNN picking results with manual picks. With the  $N(t)$  threshold 0.6, 66.2% of the *P*-arrivals (67.1% for station DP and 65.1% for station AY) and 52.7% of the *S*-arrivals (58.9% for station DP and 40.6% for station AY) have onset times with error  $\leq 10$ ms (one sample increment). 3.4% of the *P*-arrivals (2.8% for station DP and 4.0% for station AY) and 4.5% of the

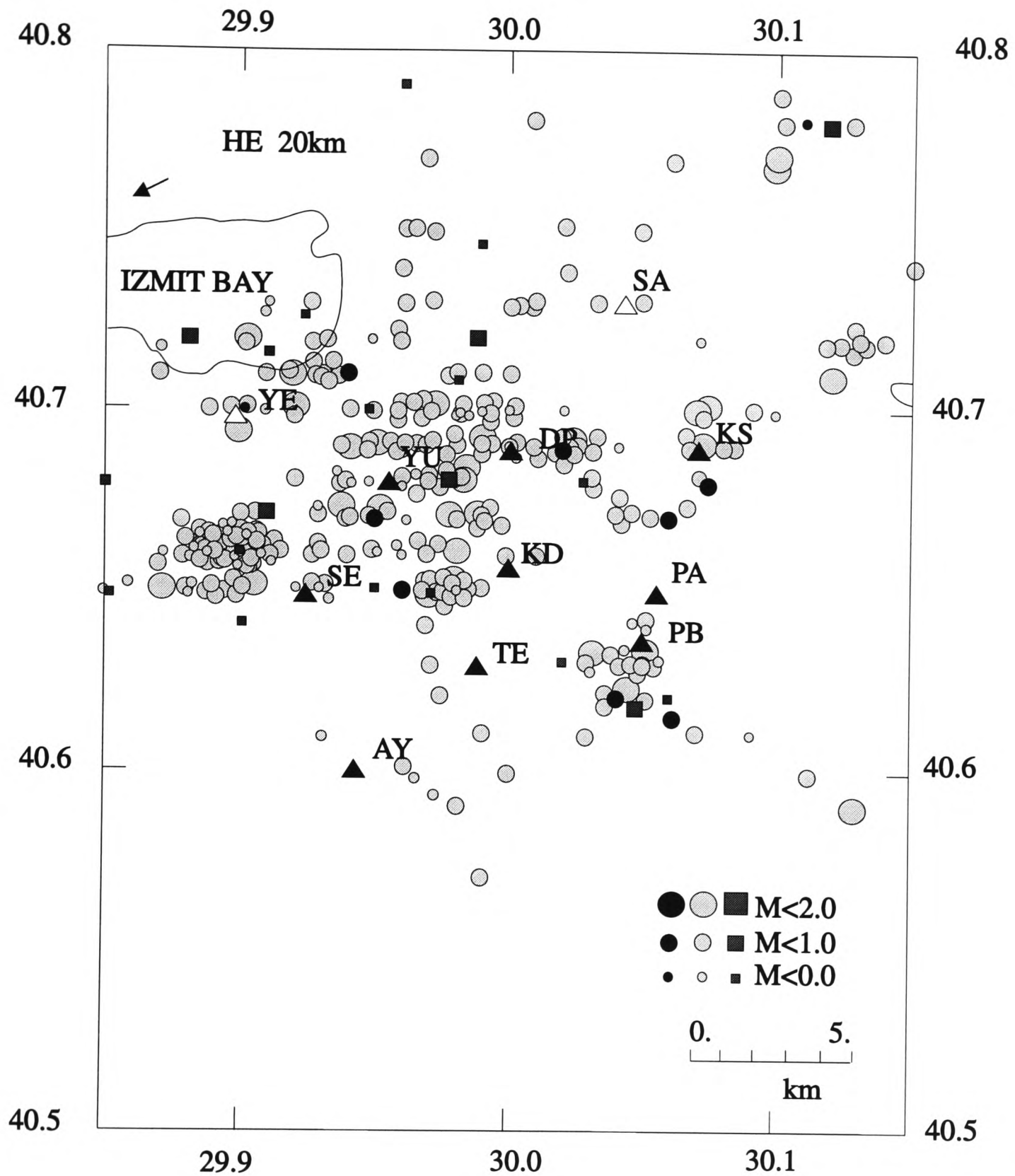


Figure 5.6(a). The location map of events which recorded in station DP. Each symbol represents an event whose P-arrival was manually picked in vertical component. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals.

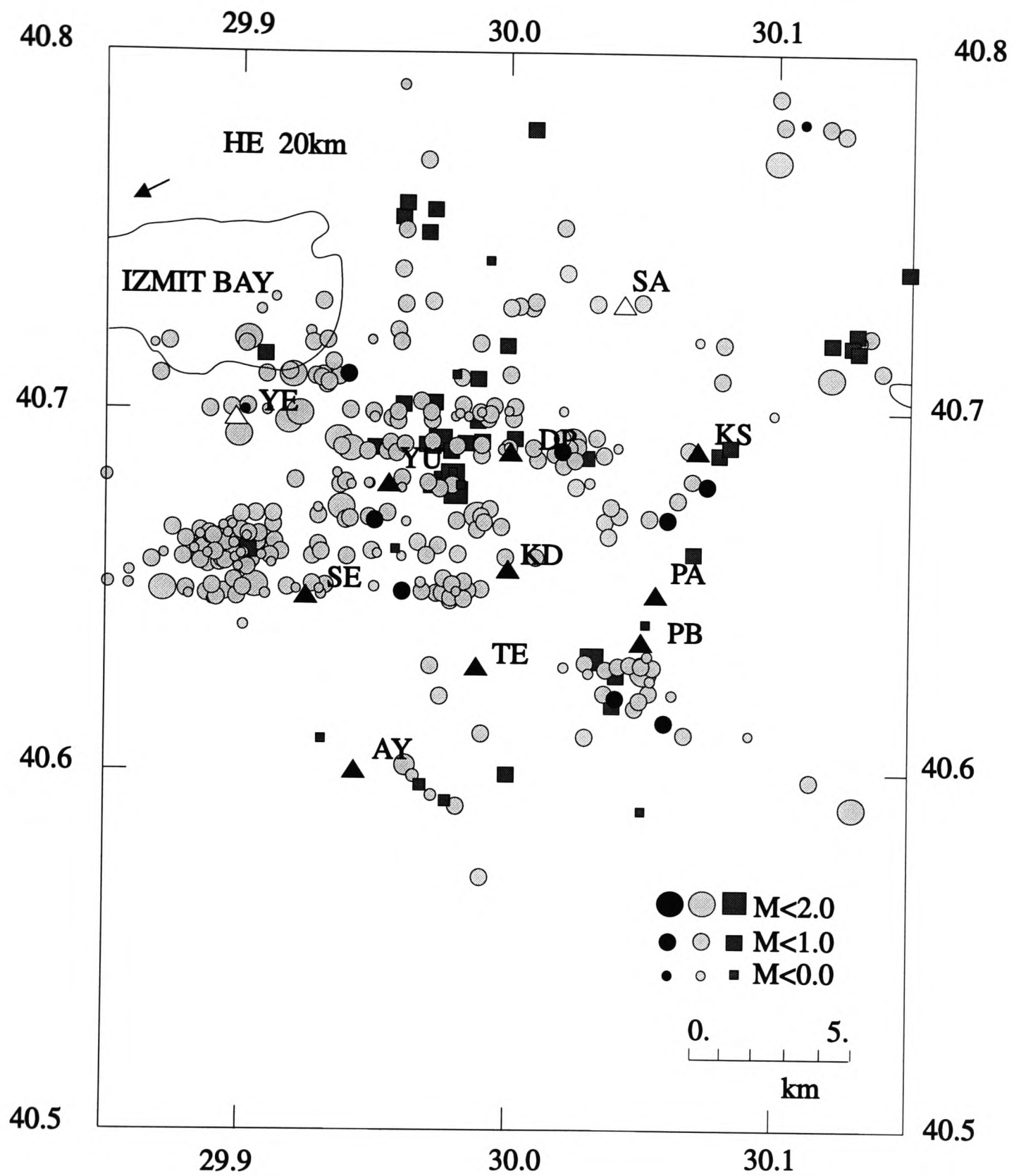


Figure 5.6(b). The location map of events which recorded in station DP. Each symbol represents an event whose S-arrival was manually picked in vertical component, Grey circles represent S-arrivals which are picked by the BPNN, and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals.

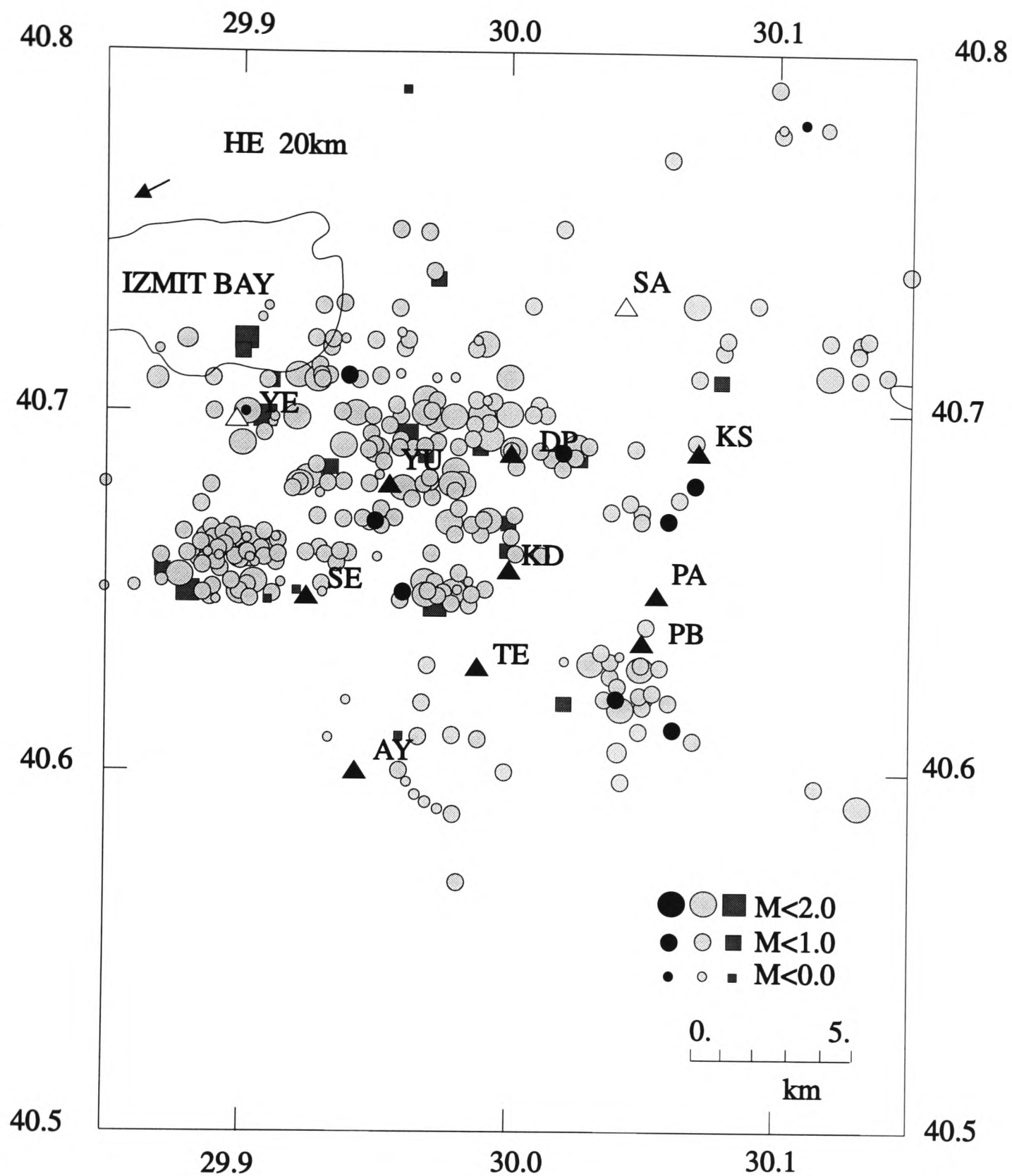


Figure 5.6(c). The location map of events which recorded in station AY. Each symbol represents an event whose P-arrival was manually picked in vertical component. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals from station DP.



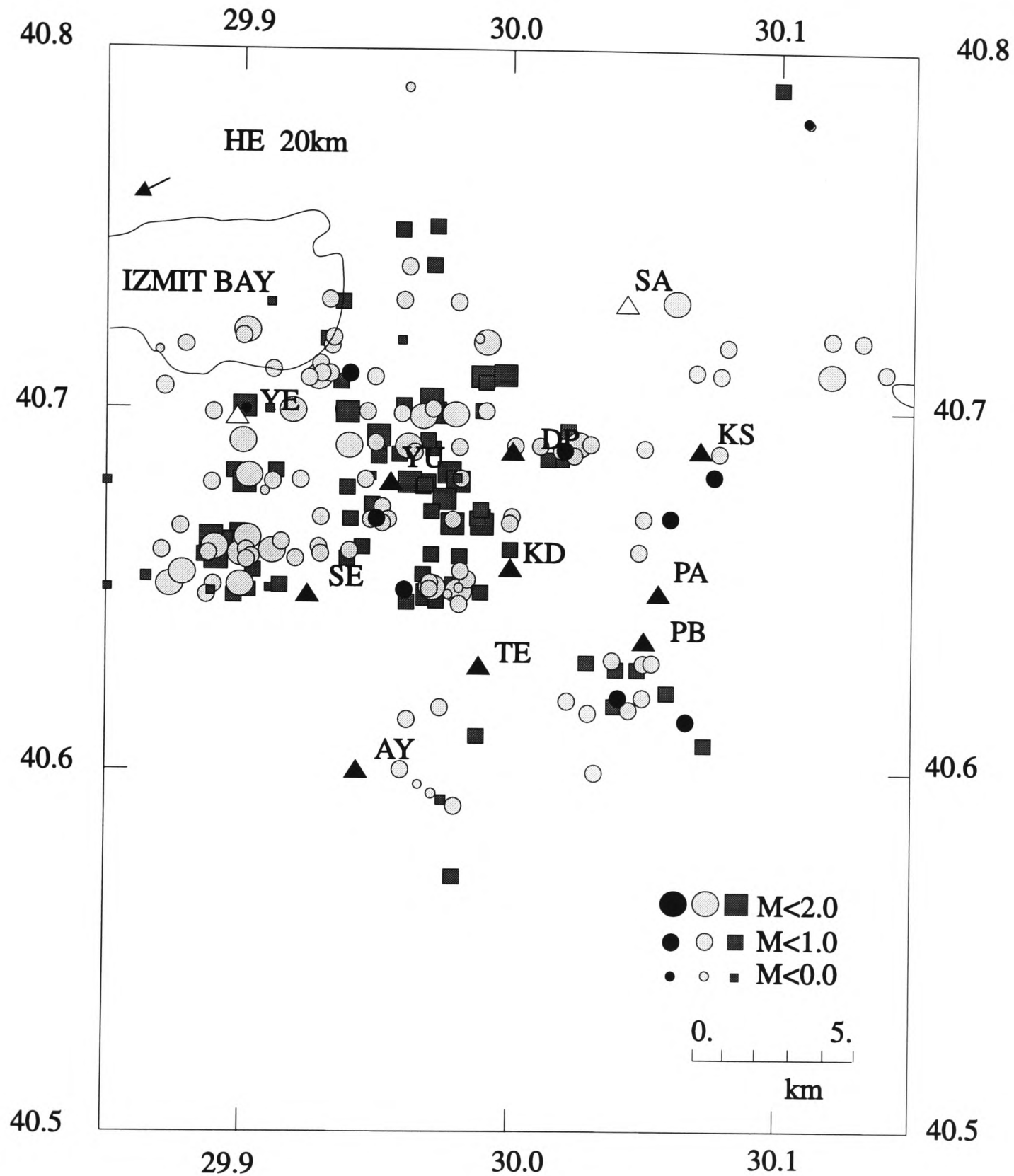


Figure 5.6(d). The location map of events which recorded in station AY. Each symbol represents an event whose S-arrival was manually picked in vertical component. Grey circles represent S-arrivals which are picked by the BPNN, and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals from station DP.

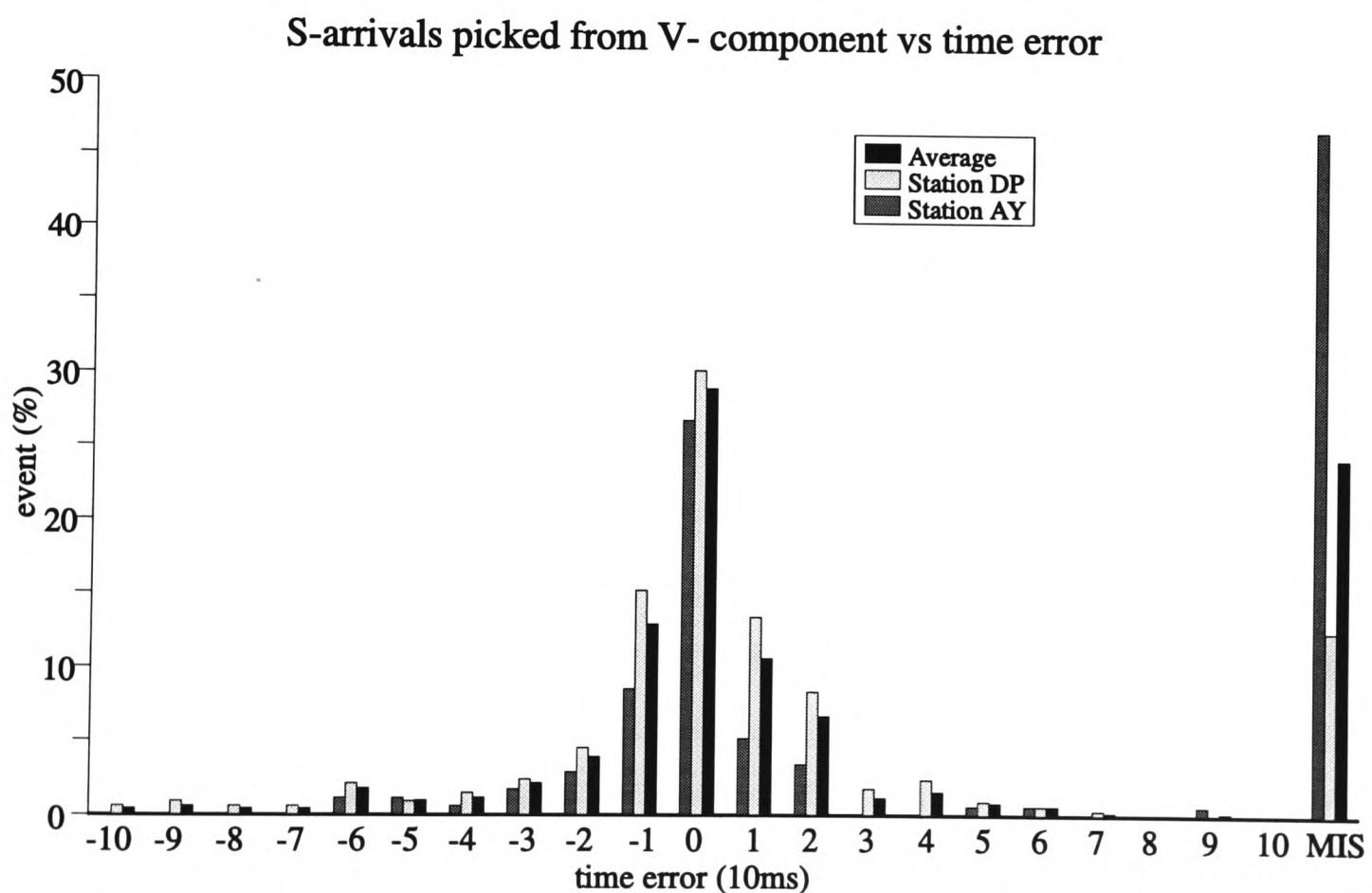
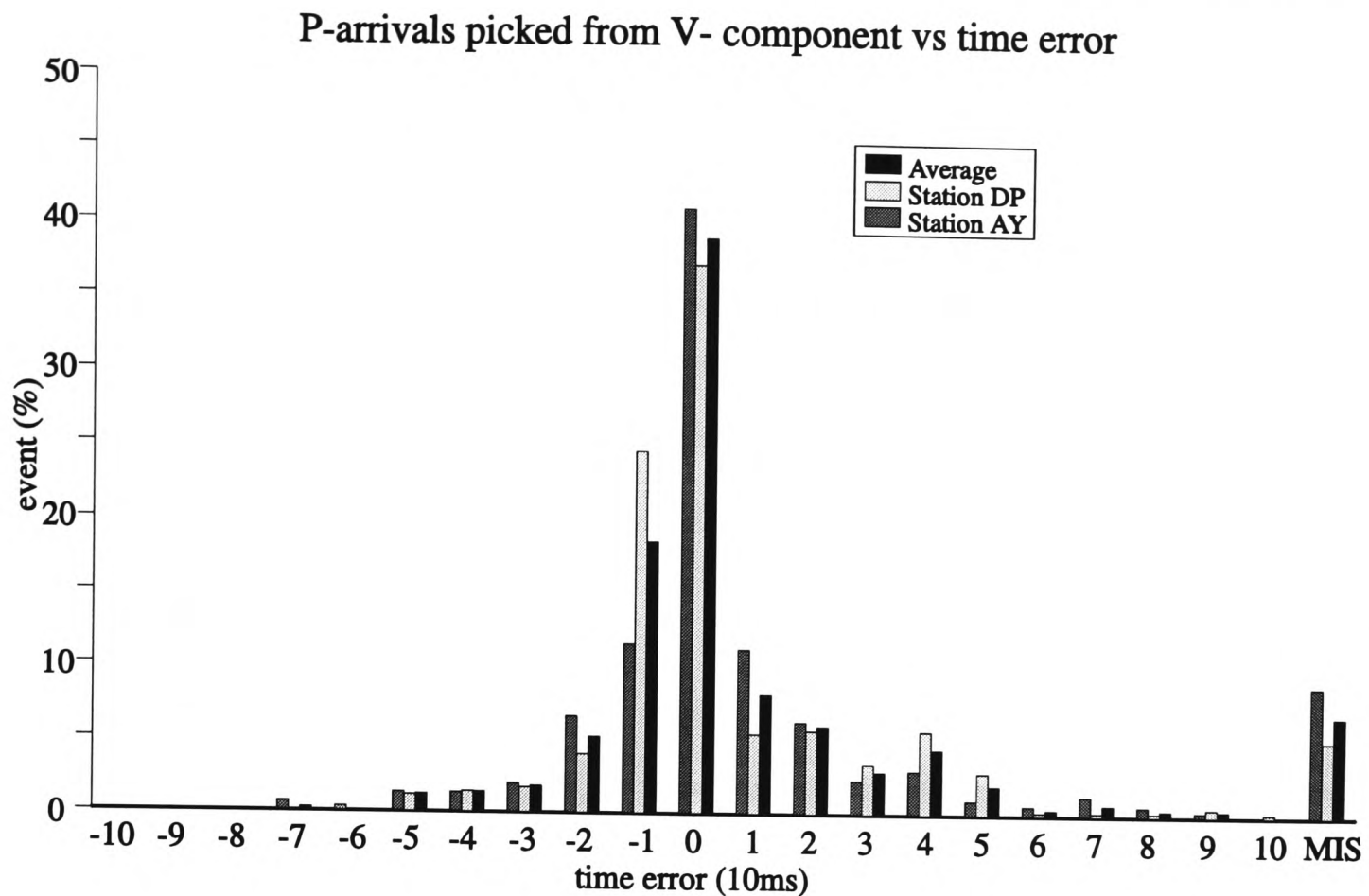


Figure 5.7. Statistical comparison of P- and S-arrivals picked by the BPNN on the complete dataset compared with manual picks with notations as in Figure 4.8. The success rate of the trained BPNN relative to manual reference picks is quoted as a percentage.

*S*-arrivals (5.7% for station DP and 2.2% for station AY) have an estimated error between 50ms (five sample increments) and 100ms (ten sample increments) whereas 6.9% of the *P*-arrivals (5.1% for station DP and 9.1% for station AY) and 24.4% of the *S*-arrivals (14.1% for station DP and 46.9% for station AY) have estimated greater than 100ms (ten sample increment) or are absent. The noise bursts are picked from 94 (12.3%) recordings in the total 762 recordings, in which 73 are from station DP and 21 are from station AY. Among the 94 recordings, most of them have only one noise burst in one recording.

#### 5.4.2 Testing on E-W component recordings

From the E-W component, 651 (356 from station DP and 295 from station AY) *P*-arrivals and 623 (342 from station DP and 281 from station AY) *S*-arrivals are manually picked. This BPNN can detect 582 (89.4%) of the *P*-arrivals, 327 (91.9%) and 255 (86.4%) from stations DP and AY, and 566 (90.9%) of the *S*-arrivals, 308 (90.1%) and 258 (91.8%) from stations DP and AY respectively, by using the  $N(t)$  threshold of 0.6. Figure 5.8 shows the picking results related to event positions. If the  $N(t)$  threshold is reduced to 0.5, it can detect 587 (89.2%) of the *P*-arrivals, 330 (92.7%) and 257 (85.1%) from stations DP and AY, and 580 (93.1%) of the *S*-arrivals, 315 (92.1%) and 265 (94.3%) from stations DP and AY respectively.

Figure 5.9 also shows the comparison of the picking results with manual picks. With the  $N(t)$  threshold 0.6, 59.2% of the *P*-arrivals (58.9% and 60.0% for stations DP and AY respectively) and 61.2% of the *S*-arrivals (54.6% and 69.0% for stations DP and AY respectively) have onset times with error  $\leq 10$ ms (one sample increment). 3.4% of the *P*-arrivals (3.9% and 2.7% for stations DP and AY respectively) and 3.7% of the *S*-arrivals (3.8% and 3.6% for stations DP and AY respectively) have an estimated error between 50ms (five sample increments) and 100ms (ten sample increments) whereas 11.6% of the *P*-arrivals (8.1% and 15.9% for stations DP and AY respectively) and 10.1% of the *S*-arrivals (9.9% and 10.3% for stations DP and AY respectively) have an estimated error greater than 100ms (ten sample increment) or are

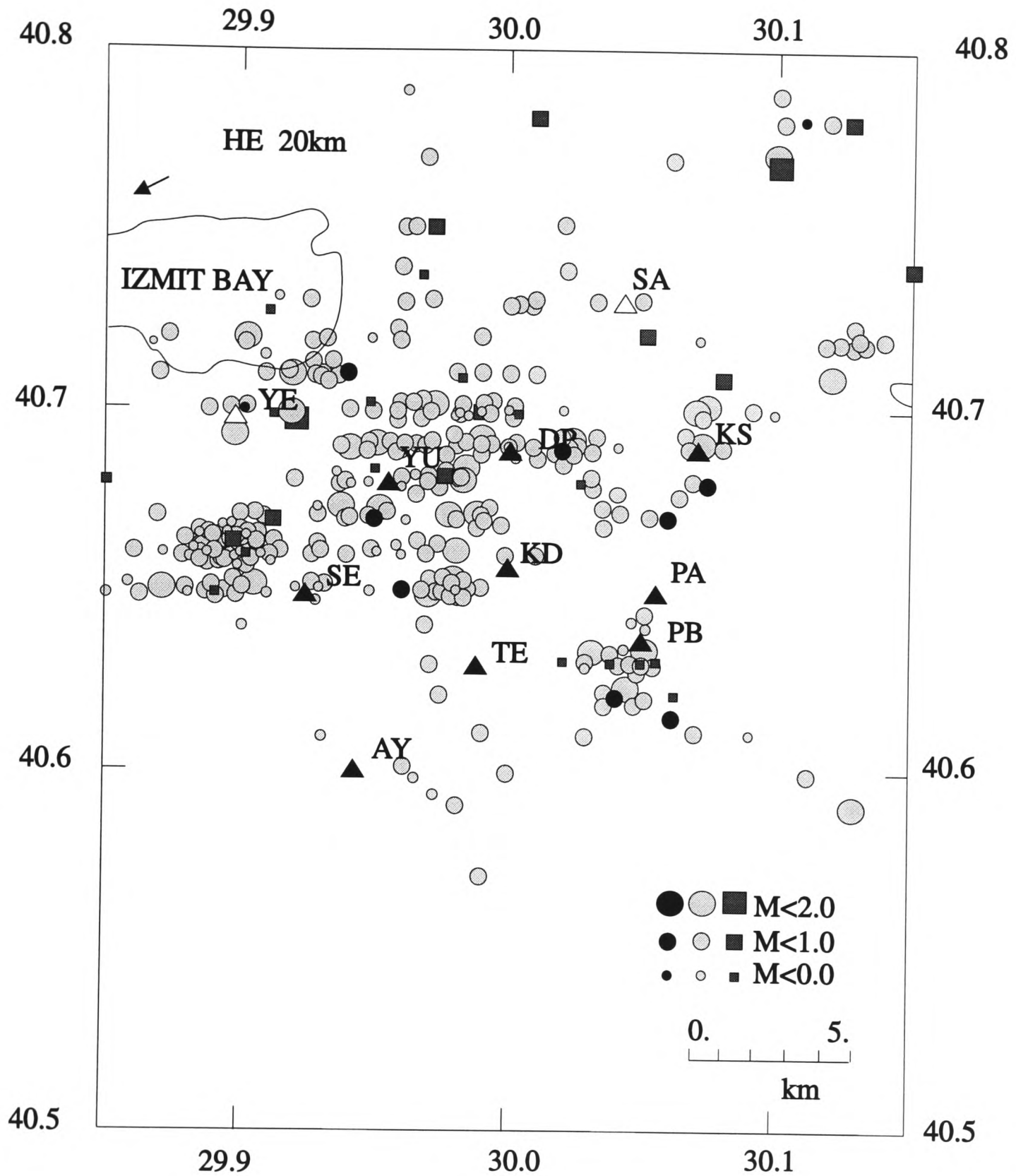


Figure 5.8(a). The location map of events which recorded in station DP. Each symbol represents an event whose P-arrival was manually picked in E-W component. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.

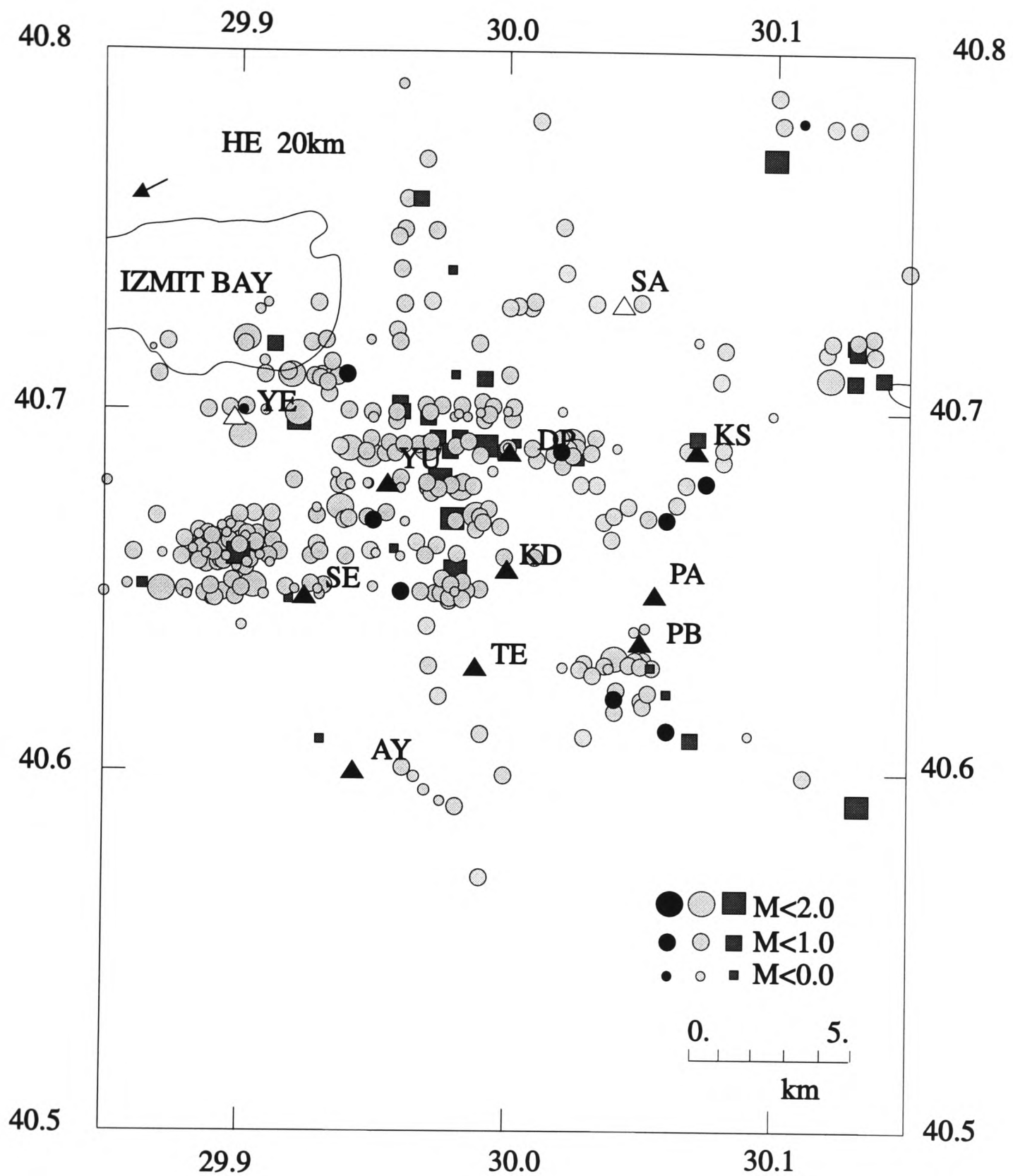


Figure 5.8(b). The location map of events which recorded in station DP. Each symbol represents an event whose S-arrival was manually picked in E-W component. Grey circles represent S-arrivals which are picked by the BPNN, and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.

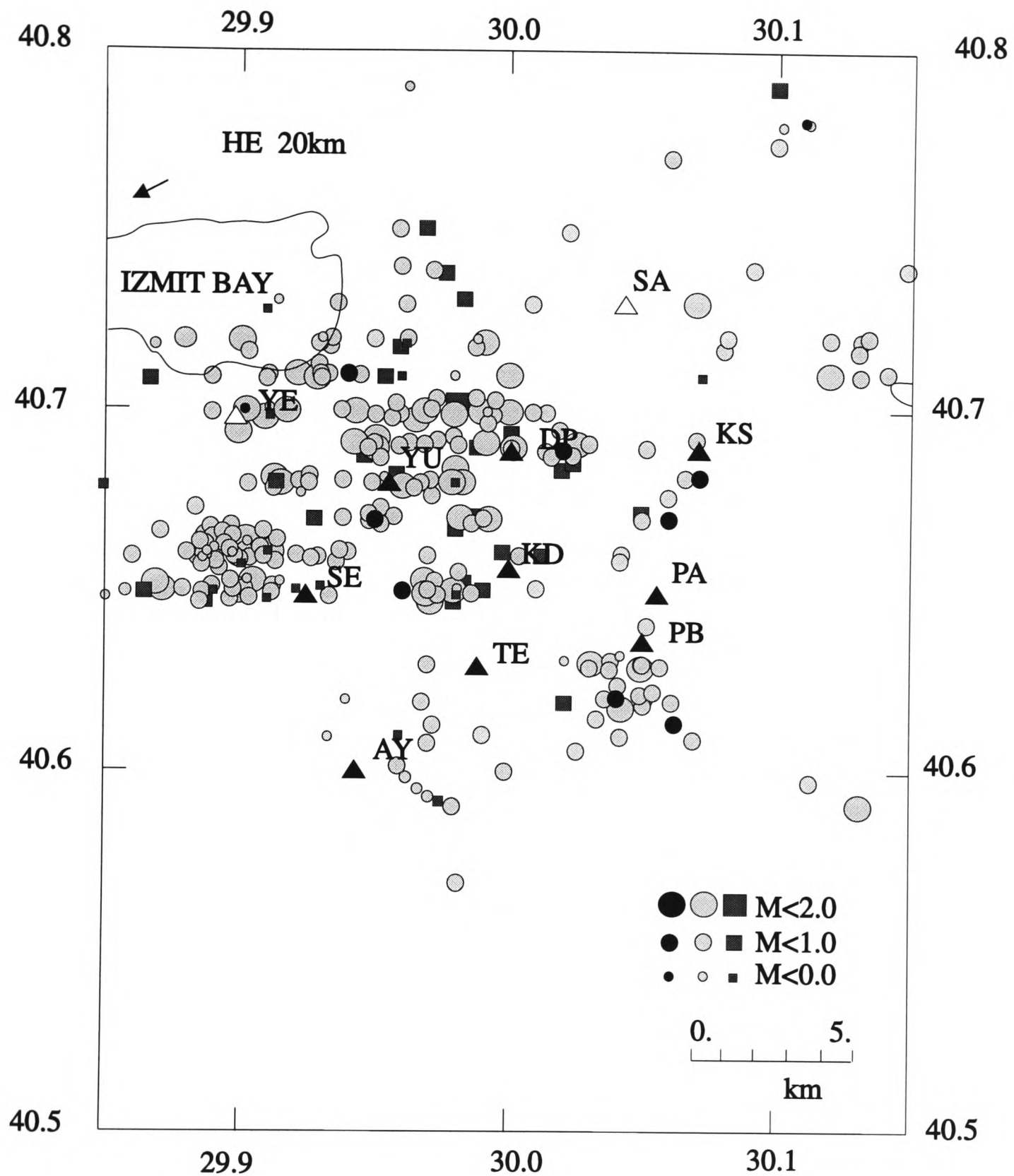


Figure 5.8(c). The location map of events which recorded in station AY. Each symbol represents an event whose P-arrival was manually picked in E-W component. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.

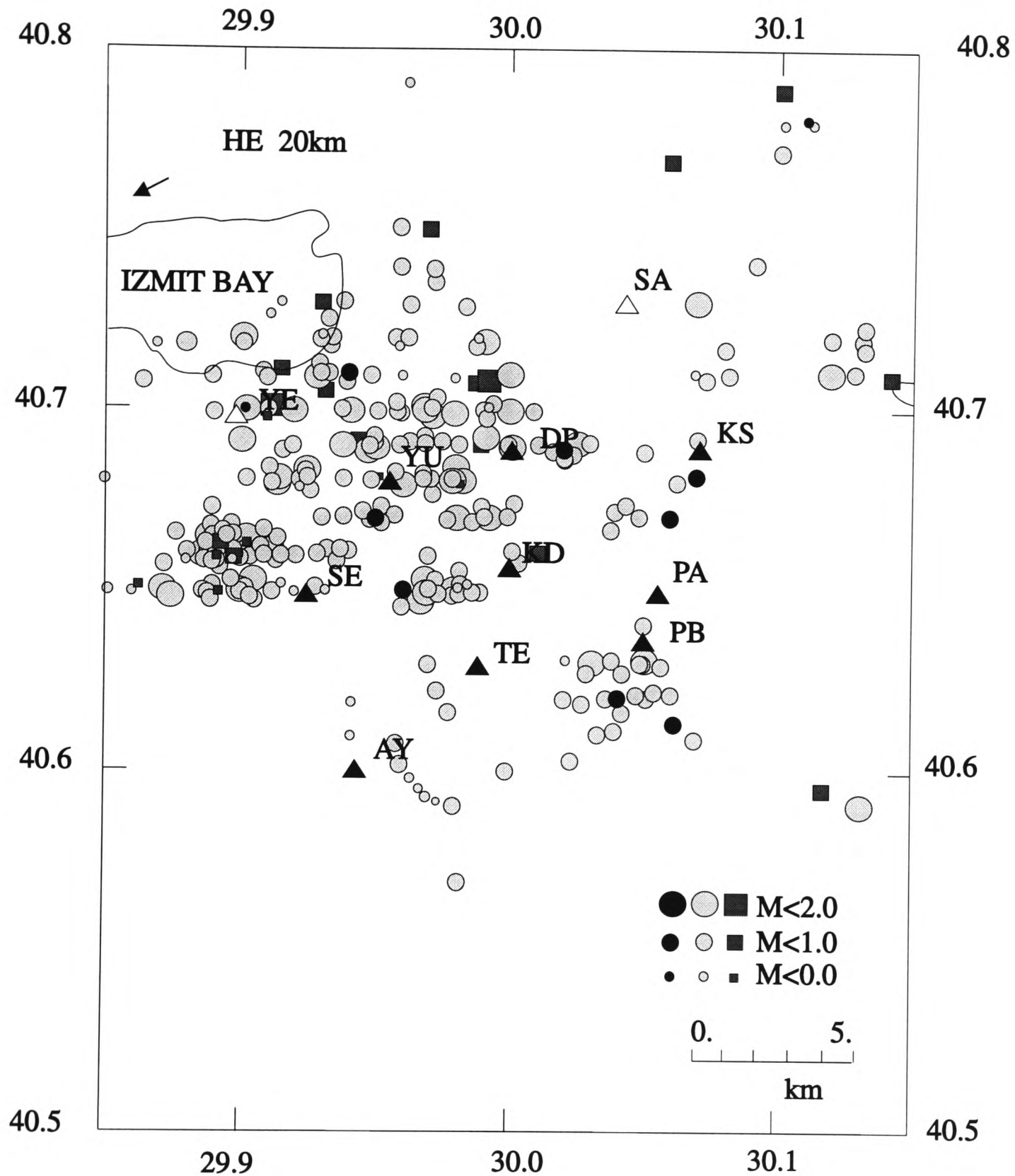


Figure 5.8(d). The location map of events which recorded in station AY. Each symbol represents an event whose S-arrival was manually picked in E-W component. Grey circles represent S-arrivals which are picked by the BPNN, and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.



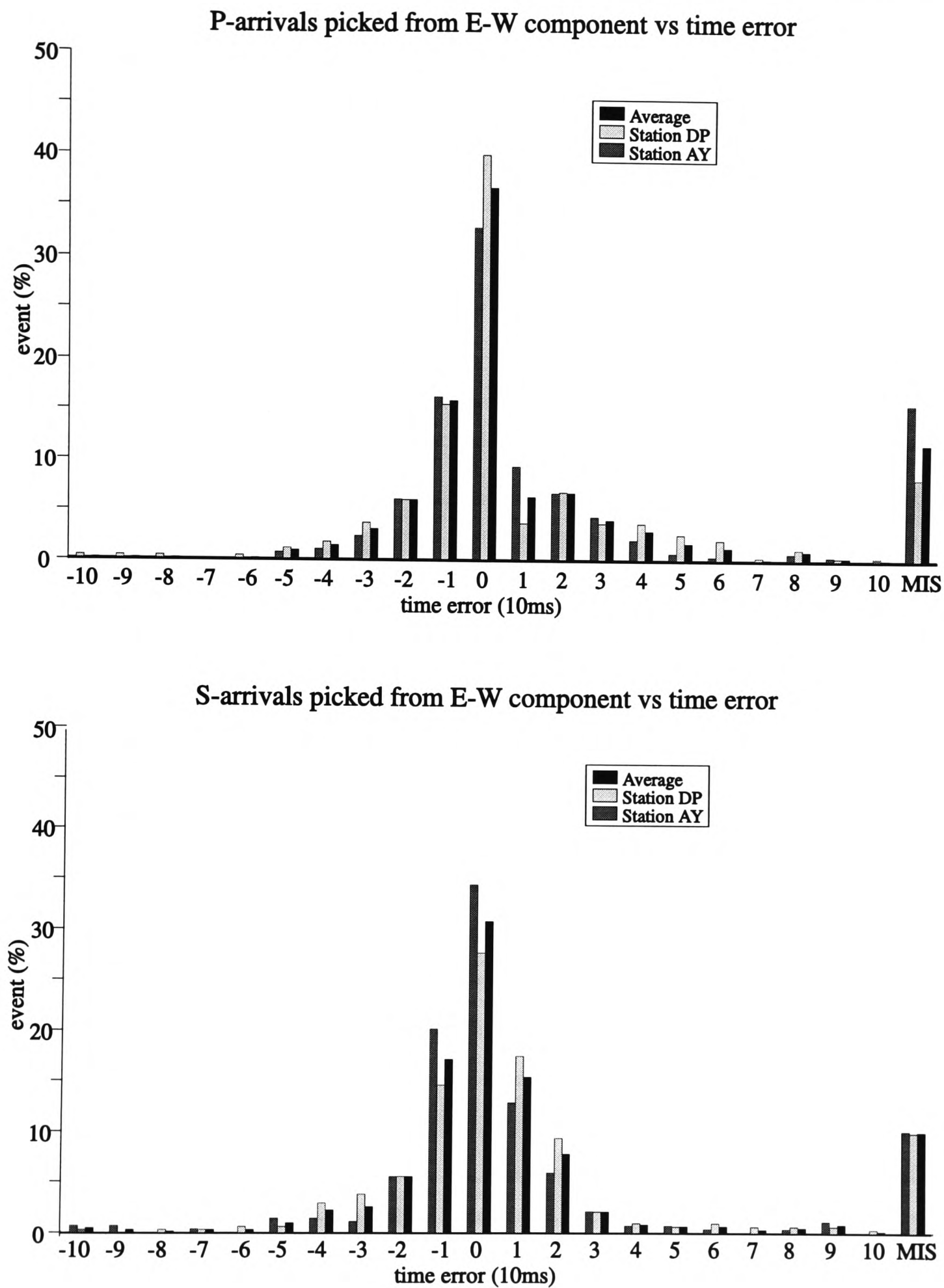


Figure 5.9. Statistical comparison of P- and S-arrivals picked by the BPNN on the complete dataset compared with manual picks with notations as in Figure 4.8. The success rate of the trained BPNN relative to manual reference picks is quoted as a percentage.



absent.

Among the missed 76 *P*-arrivals (26 from station DP and 47 from station AY) and 63 *S*-arrivals (34 from station DP and 29 from station AY), 36 *P*-arrivals (9 from station DP and 27 from station AY) and 5 *S*-arrivals (1 from station DP and 4 from station AY) are detected by the BPNN but are discarded by post-processing. 6 *P*-arrivals (3 from station DP and 3 from station AY) and 13 *S*-arrivals (7 from station DP and 6 from station AY) are picked but with error larger than 100ms (ten sample increments). The noise bursts are picked from 158 (20.7%) recordings in the total of 762 recordings, in which 104 and 54 are from stations DP and AY respectively. Among the 158 recordings, most of them have only one noise burst in one recording.

#### 5.4.3 Testing on N-S component recordings

From the N-S component, 621 (356 from station DP and 265 from station AY) *P*-arrivals and 623 (341 from station DP and 282 from station AY) *S*-arrivals are manually picked. This BPNN can detect 516 (83.1%) of the *P*-arrivals, 321 (90.2%) and 195 (76.4%) from stations DP and AY, and 543 (90.9%) of the *S*-arrivals, 302 (88.6%) and 241 (85.4%) from stations DP and AY respectively, by using the  $N(t)$  threshold of 0.6. Figure 5.10 shows the picking results related to event positions. If the  $N(t)$  threshold reduced is to 0.5, it can detect 528 (85.0%) of the *P*-arrivals, 327 (91.8%) and 201 (75.8%) from stations DP and AY, and 561 (90.0%) of the *S*-arrivals, 307 (90.0%) and 254 (90.1%) from stations DP and AY respectively.

Figure 5.11 also shows the comparison of its picking results with manual picks. With the  $N(t)$  threshold 0.6, 60.3% of the *P*-arrivals (63.8% and 55.8% for stations DP and AY respectively) and 57.7% of the *S*-arrivals (59.5% and 55.7% for stations DP and AY respectively) have onset times with error  $\leq 10$ ms (one sample increment). 3.2% of the *P*-arrivals (2.8% and 3.8% for stations DP and AY respectively) and 5.8% of the *S*-arrivals (3.2% and 8.9% for stations DP and AY respectively) have estimated with error between 50ms (five sample increments) and 100ms (ten sample increments) whereas 16.6% of the *P*-arrivals (9.6% and 26.0% for stations DP and AY

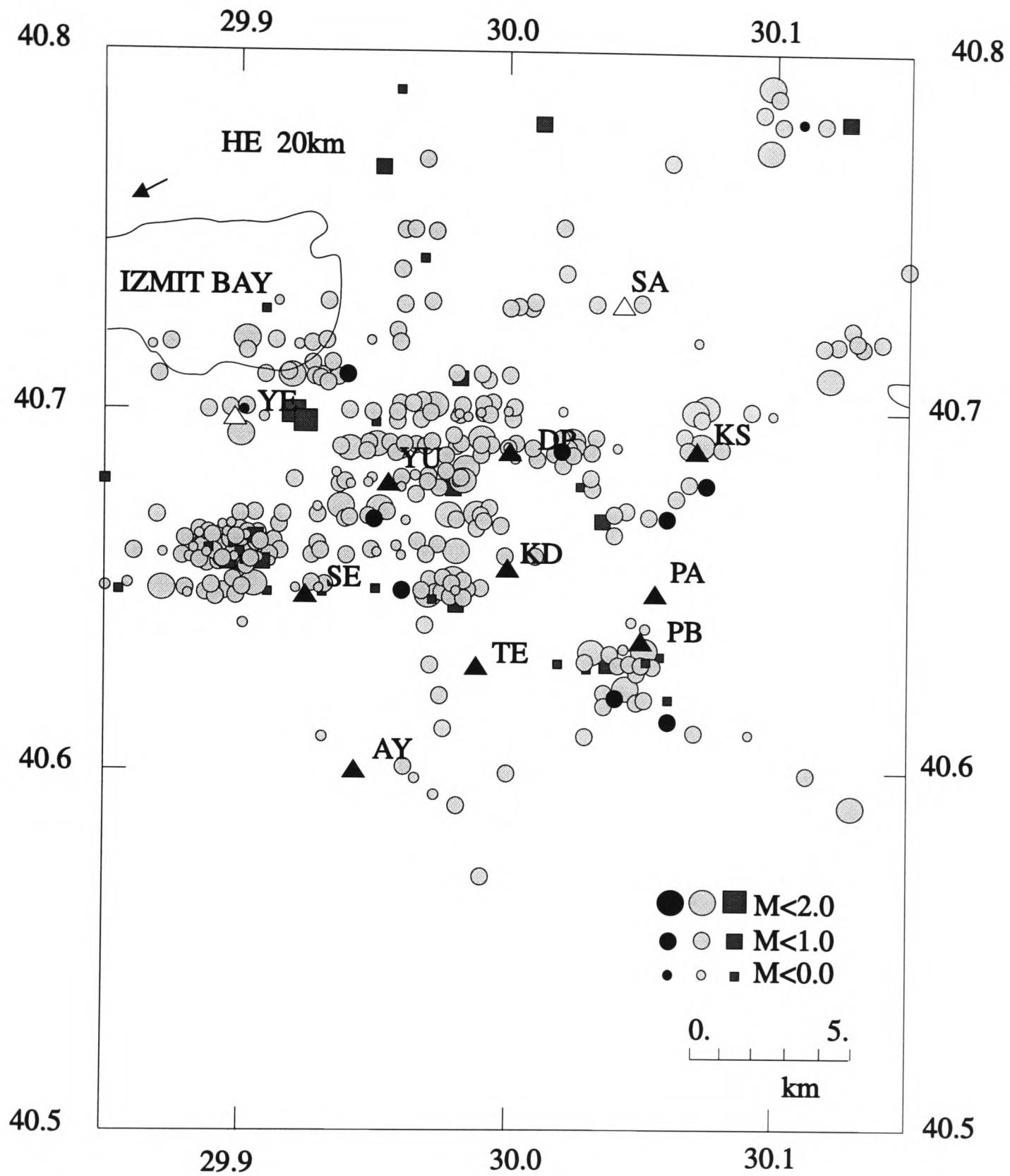


Figure 5.10(a). The location map of events which recorded in station DP. Each symbol represents an event whose P-arrival was manually picked in N-S component. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.

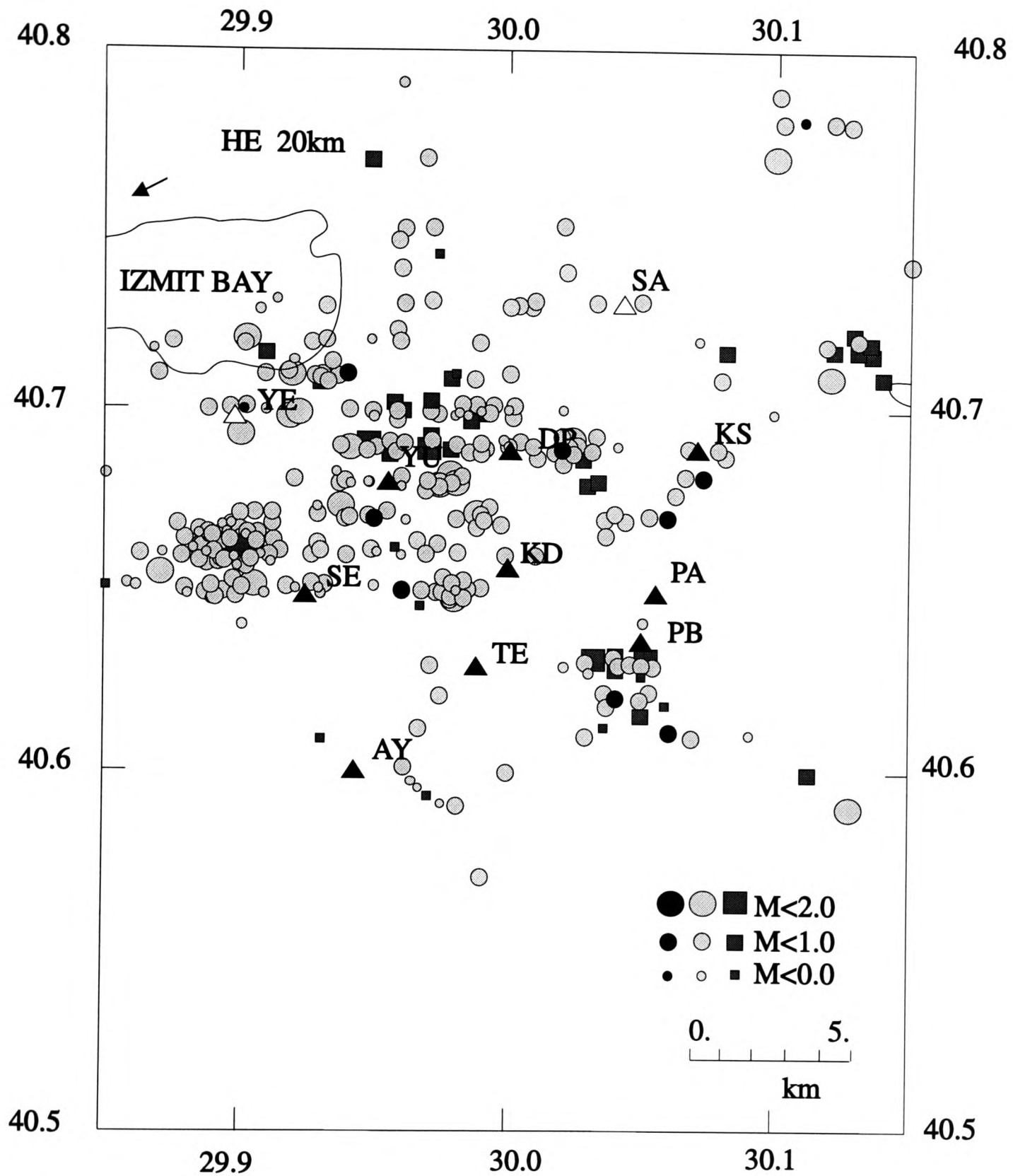


Figure 5.10(b). The location map of events which recorded in station DP. Each symbol represents an event whose P-arrival was manually picked in N-S component. Grey circles represent S-arrivals which are picked by the BPNN, and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.

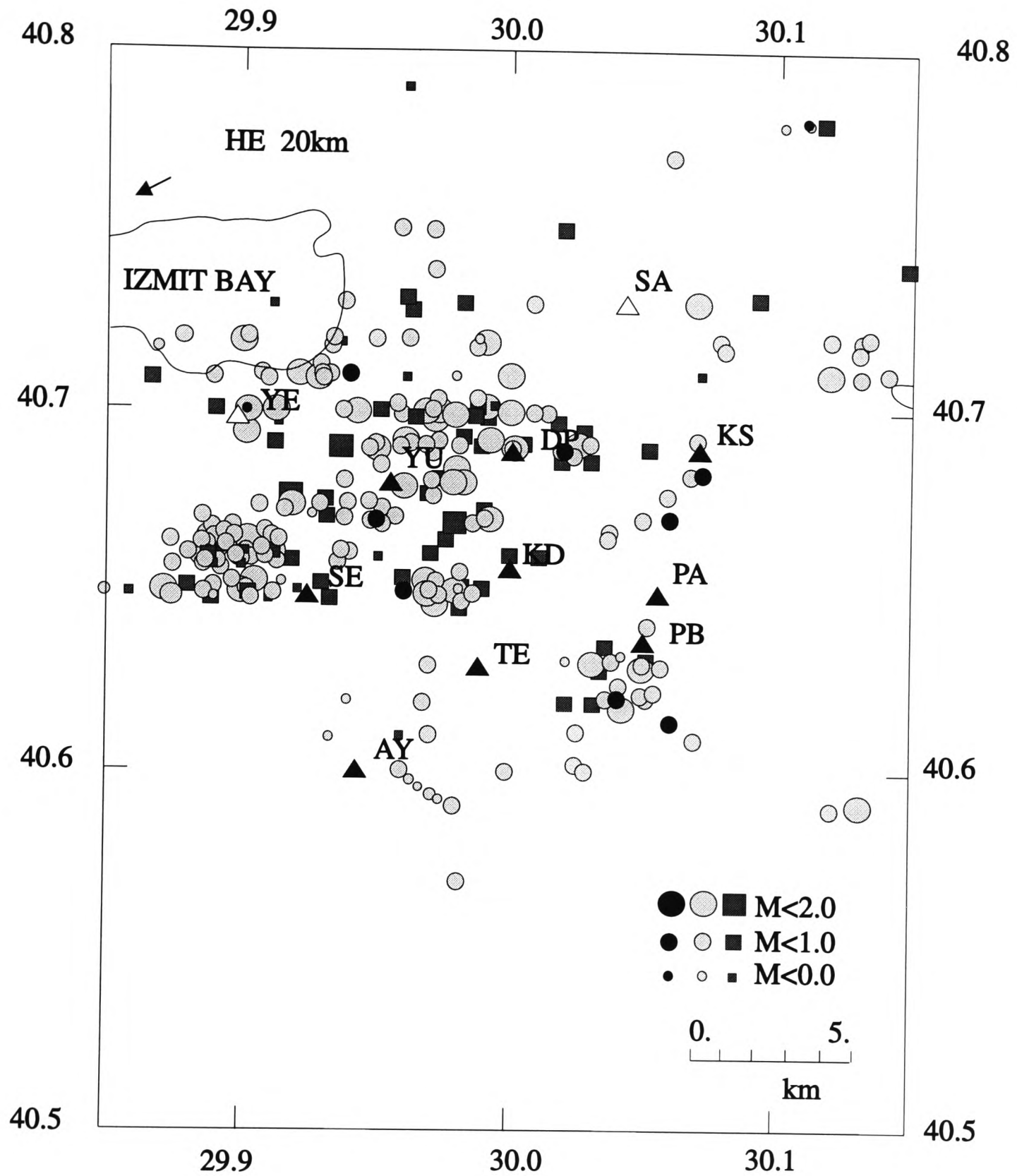


Figure 5.10(c). The location map of events which recorded in station AY. Each symbol represents an event whose P-arrival was manually picked in N-S component. Grey circles represent P-arrivals which are picked by the BPNN, and grey squares represent P-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.

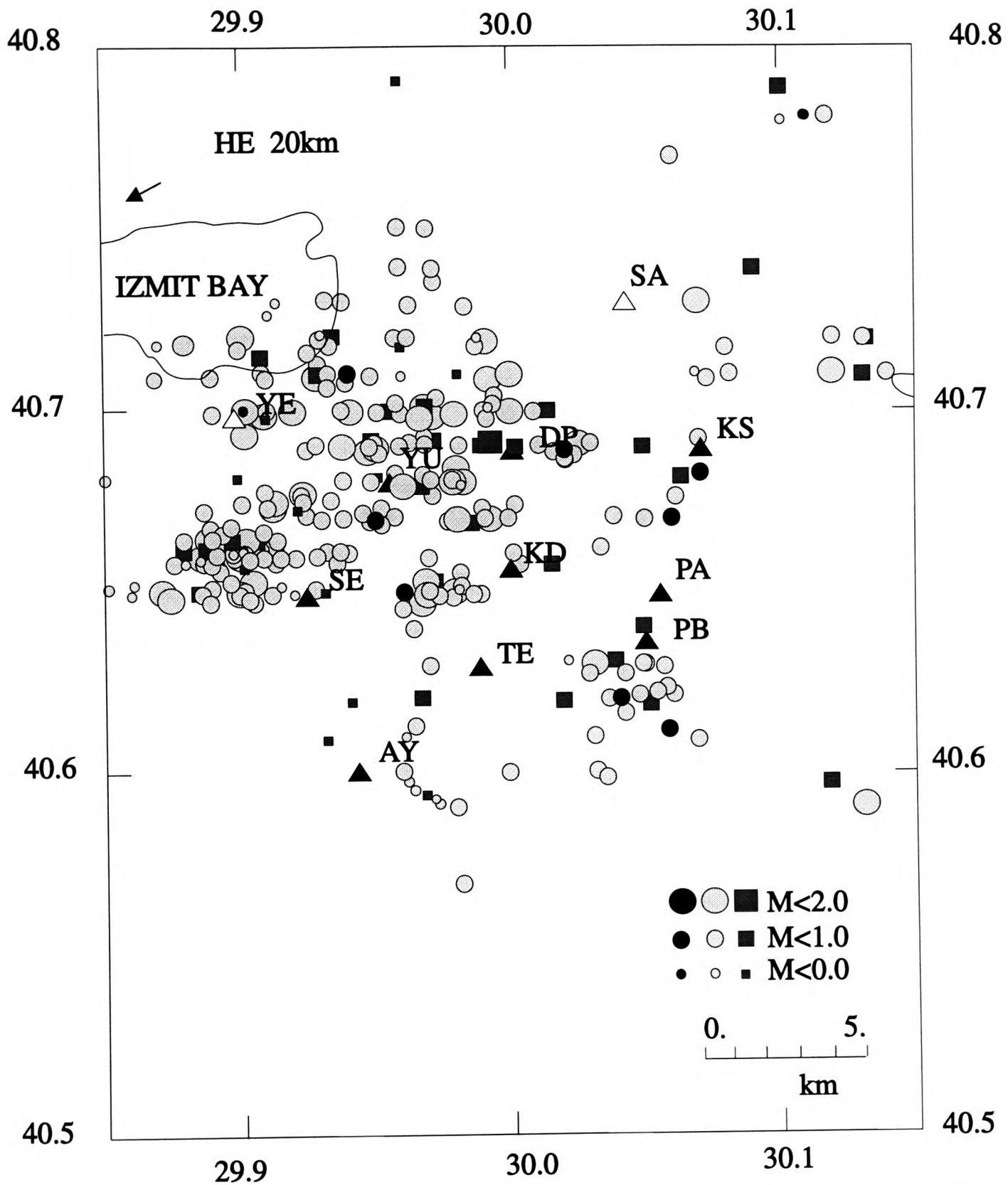


Figure 5.10(d). The location map of events which recorded in station AY. Each symbol represents an event whose P-arrival was manually picked in N-S component. Grey circles represent S-arrivals which are picked by the BPNN, and grey squares represent S-arrivals which are missed by the BPNN. Black circles represent the training P-arrivals of vertical component from station DP.

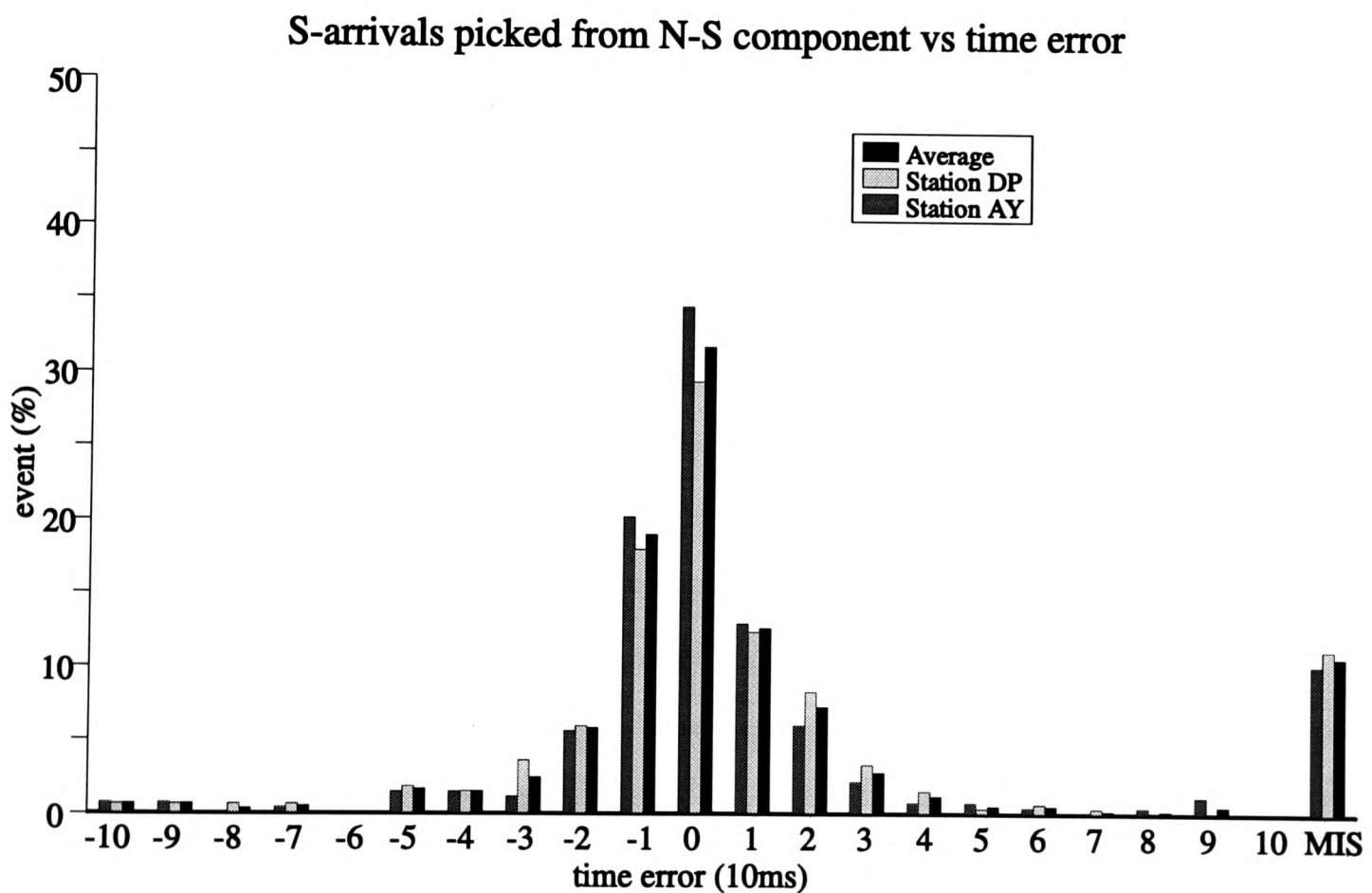
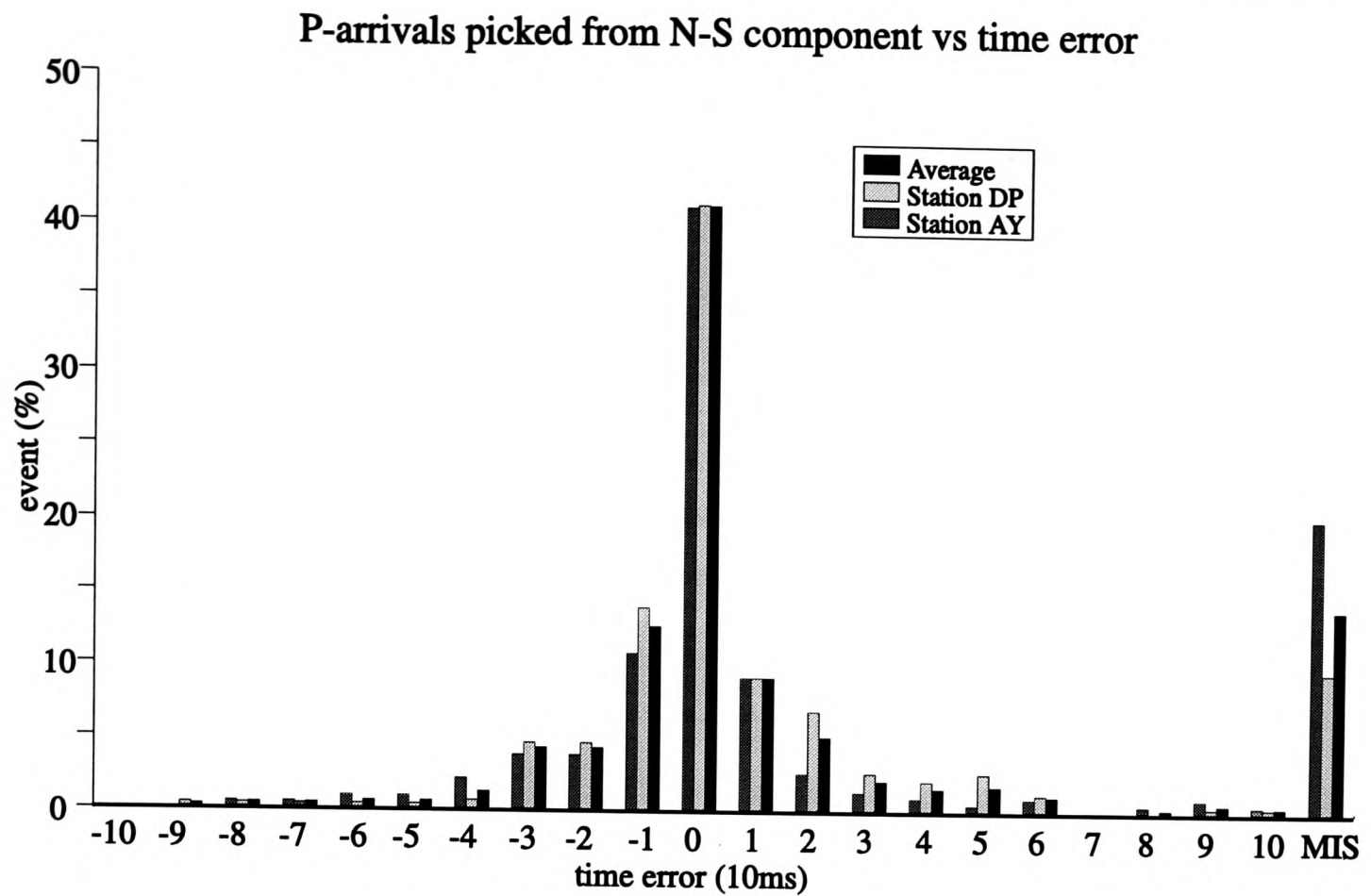


Figure 5.11. Statistical comparison of P- and S-arrivals picked by the BPNN on the complete dataset compared with manual picks with notations as in Figure 4.8. The success rate of the trained BPNN relative to manual reference picks is quoted as a percentage.

respectively) and 12.7% of the *S*-arrivals (11.1% and 14.5% for stations DP and AY respectively) have an estimated error greater than 100ms (ten sample increment) or are absent.

Among the missed 103 *P*-arrivals (34 from station DP and 69 from station AY) and 79 *S*-arrivals (38 from station DP and 41 from station AY), 49 *P*-arrivals (11 from station DP and 38 from station AY) but no *S*-arrivals are detected by the BPNN, however, they are discarded by the post-processing. One *P*-arrival from station DP and 19 *S*-arrivals (one from station DP and 18 from station AY) are picked but with error larger than 100ms (ten samples). The noise bursts are picked from 139 (18.2%) recordings in the total 762 recordings, in which 95 are from station DP and 44 are from station AY. Among the 139 recordings, most of them have only one noise burst in one recording.

#### 5.4.4 Overall performance from three 1-C recordings

In order to obtain a clear image about the BPNN performance for 1-C recordings, Table 5.2 summarizes the results from both the 1-C method and the 3-C method. This table shows that stations DP and AY have different performance on the 1-C recordings. As the data quality at station DP is higher than that at station AY, the source orientation has less effect on station DP than on station AY. For example, with the  $N(t)$  threshold 0.6, the picking rates from three 1-C recordings, for the data from station DP, are from 90.2% to 93.7% for the *P*-arrivals and from 86.1% to 90.1% for the *S*-arrivals, but for the data from station AY, they are from 76.2% to 92.2 for *P*-arrivals and from 53.1% to 91.8% for *S*-arrivals.

Detecting and picking results from station AY are poorer than those from station DP. This is largely a consequence of the data quality. The recordings on station DP usually have large amplitude, stronger energy and higher SNR than those on station AY. The arrivals on station DP are much clearer than on station AY, so the arrival times picked by the BPNN from station DP are more accurate than those from station AY. The source orientation has a larger effect on the picking of *S*-arrivals than

**TABLE 5.2 A.** Summary of picking performances for P-arrivals from different component recordings.

		DP		AY		Overall	
N(t) threshold		0.6	0.5	0.6	0.5	0.6	0.5
3-C	manual	356		300		656	
	BPNN	345	352	274	277	619	629
		96.9%	98.8%	91.3%	92.3%	94.3%	95.8%
V-C	manual	350		298		648	
	BPNN	328	330	275	275	603	605
		93.7%	94.3%	92.2%	90.2%	93.1%	93.4%
E-W	manual	356		295		651	
	BPNN	327	330	255	257	582	587
		91.9%	92.7%	86.4%	87.1%	89.4%	90.2%
N-S	manual	356		265		621	
	BPNN	321	327	195	201	516	528
		90.2%	91.8%	76.2%	75.8%	83.1%	85.0%

**TABLE 5.2 B.** Summary of picking performances for S-arrivals from different component recordings.

		DP		AY		Overall	
N(t) threshold		0.6	0.5	0.6	0.5	0.6	0.5
3-C	manual	342		285		627	
	BPNN	302	316	240	259	542	575
		88.3%	92.4%	84.2%	90.9%	86.4%	91.7%
V-C	manual	334		175		509	
	BPNN	289	306	93	115	382	421
		86.5%	91.6%	53.1%	65.2%	75.0%	82.7%
E-W	manual	342		281		623	
	BPNN	308	315	258	265	566	580
		90.1%	92.1%	91.8%	94.3%	90.9%	93.1%
N-S	manual	341		282		623	
	BPNN	302	307	241	254	543	561
		88.6%	90.0%	85.4%	90.1%	87.2%	90.0%



that on the picking of *P*-arrivals. For example, *S*-arrivals of the V-C recordings in station AY are drowned out by the *P*-wave coda for some small events, so it is difficult to detect and pick these *S*-arrivals, even by visual analysis. Figure 5.12 shows such an example in which the *S*-arrival in V-C recordings has a smooth first motion and is consequently not picked by the BPNN, but on the horizontal components, it is clearer. This *S*-arrival is picked from both N-S and E-W components (Figure 5.13). In contrast with the *S*-arrival, the *P*-arrival is missed from the two horizontal components. However, both *P*- and *S*-arrivals are picked from the modulus (Figure 5.14). It is difficult to say from which component the method has the best performance. For example, for *P*-arrivals, the best performance on station DP is from the V-C recordings, but on station AY, the best one is from the E-W component; for *S*-arrivals, on station DP, the best performance is still from the V-C recordings, but on station AY, the best one is from the N-S component.

It seems that the arrivals are more easily picked by the BPNN when an event is closed to the training event. However, there are some big events, closing to the training ones, whose arrivals are missed by the BPNN. In contrast with this, there are some small event, far from the training ones, whose arrivals are picked by the BPNN. There is not any obvious relationship between the picking ability of this trained BPNN and event positions.

Three onset times obtained from three 1-C recordings of the same arrival might be different due to the signal quality and SNR which are dependent on the source position and many other factors. For the event shown on Figures 5.12 and 5.13, the *P*-onset time picked from V-C recordings is 40ms (four samples) later than that from modulus, and both the *S*-onset times from the E-W and the N-S component are 10ms (one sample) later than the modulus results. In comparison with the manual picks, the onset times picked from the modulus are more accurate.

It should also be noted that this BPNN is trained only with *P*-arrivals and background noise from station DP. However, it can be applied to different stations for picking other kinds of arrivals, although the picking ability varies as the data quality

Station: AY  
Date: 1984-05-03  
Time: 04h52m38s  
Scale: 641

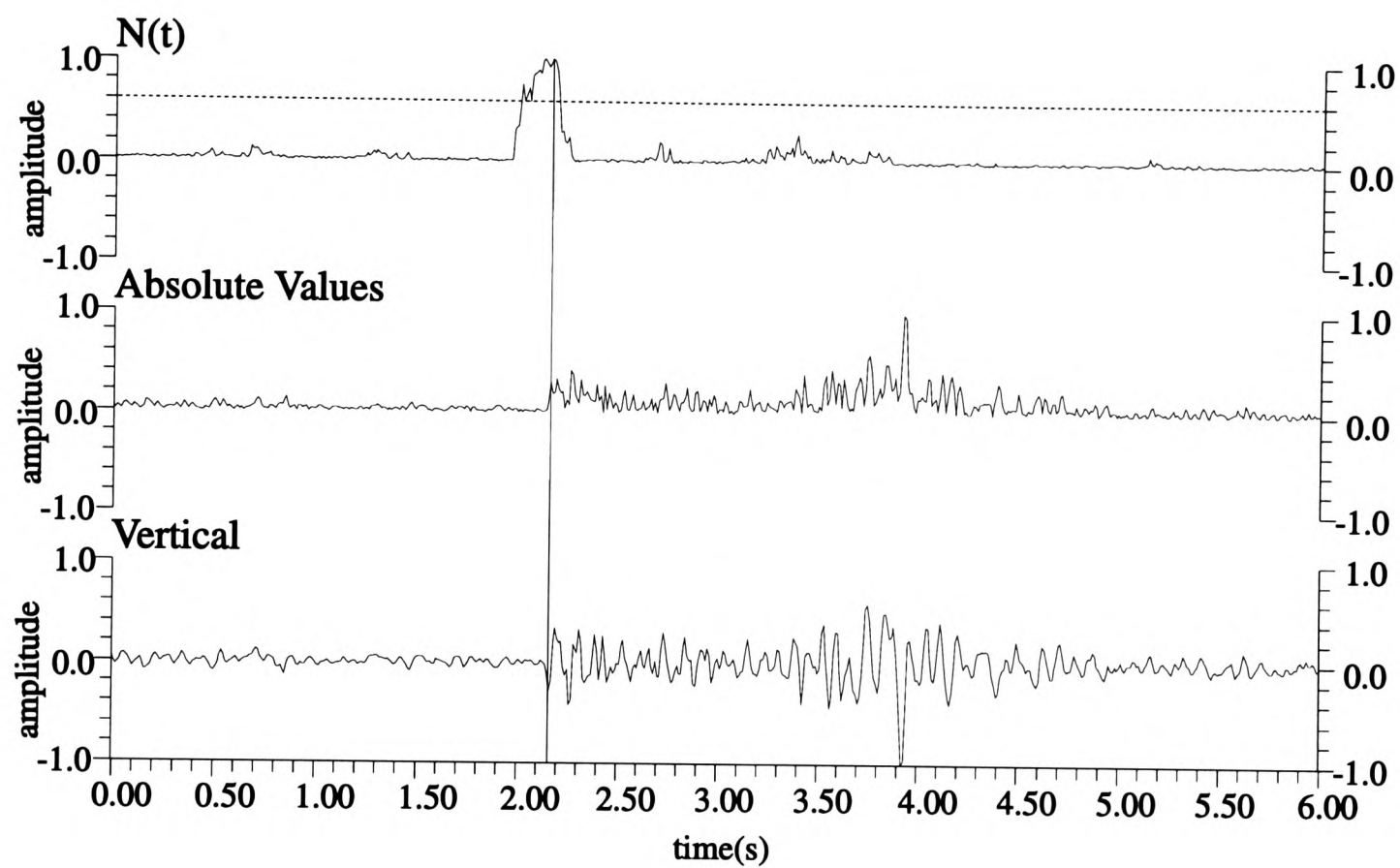


Figure 5.12. The BPNN output  $N(t)$ , V-C recording and its absolute values of a local earthquake. Vertical lines are drawn automatically by this approach. Dashed line is the picking threshold (0.6) applied to  $N(t)$ . The BPNN has 40 input nodes and is trained with ten pairs of P-arrival and background noise segments as in Figure 5.4.

Station: AY  
 Date: 1984-05-03  
 Time: 04h52m38s  
 Scale: 1086 in (a) and 1336 in (b)

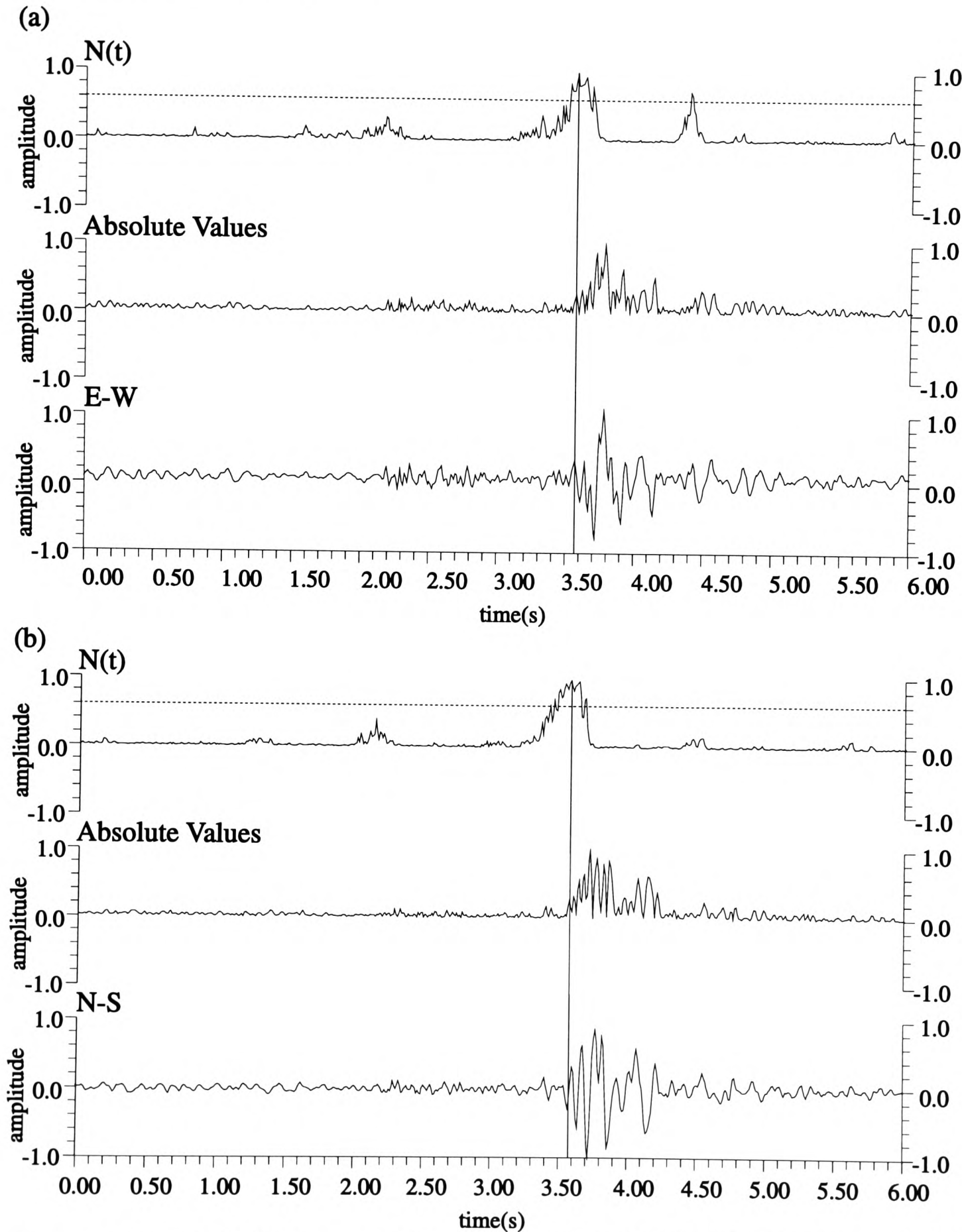


Figure 5.13. The BPNN output  $N(t)$ , E-W in (a) or N-S in (b) recording and their absolute values. The BPNN is the same as in Figure 5.9. Vertical lines are drawn automatically by this approach. Dashed lines are picking threshold (0.6) applied to  $N(t)$ .

Station: AY  
 Date: 1984-05-03  
 Time: 04h52m38s  
 Scale: 1650

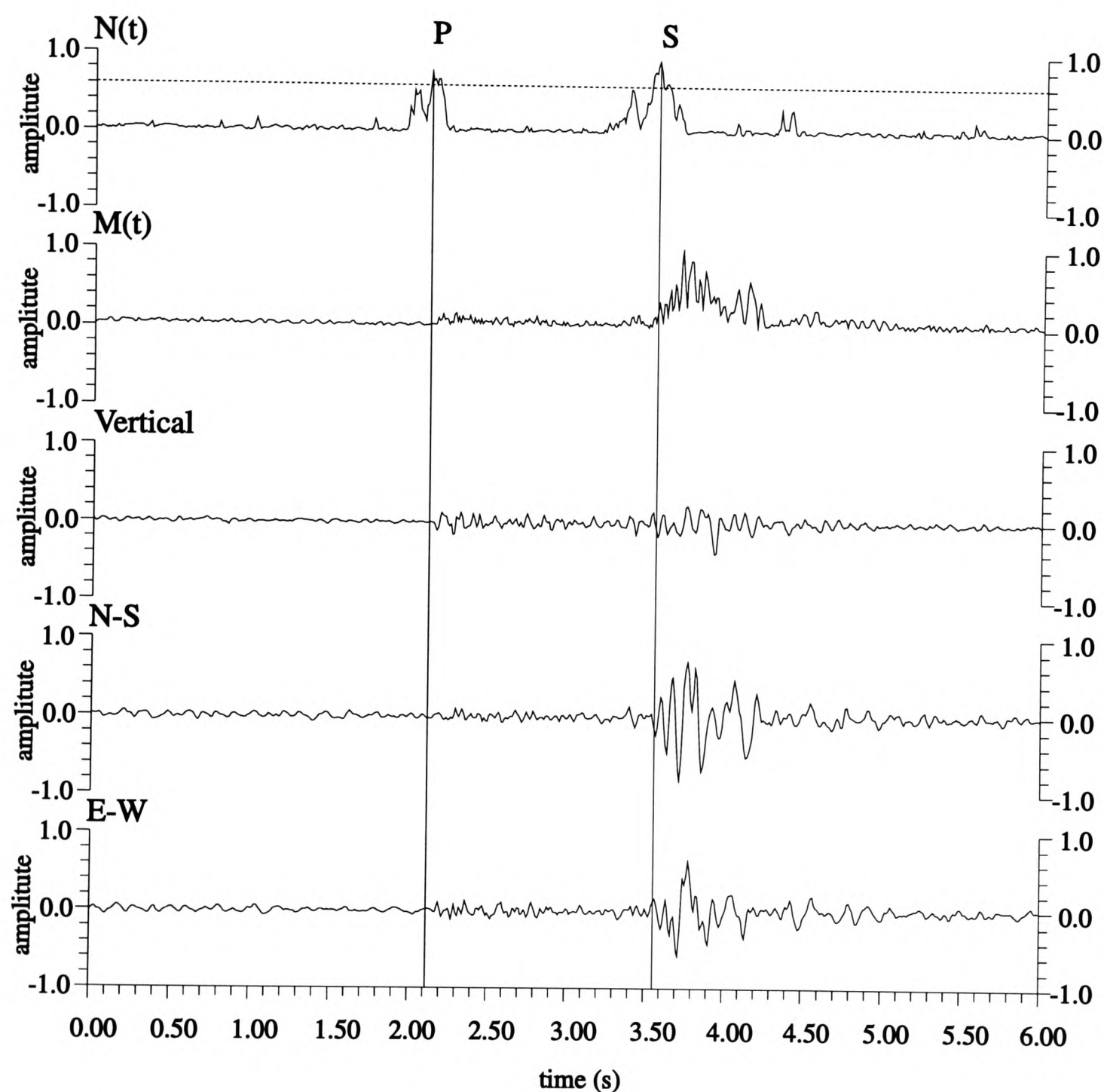


Figure 5.14. 3-C seismogram (Vertical, N-S and E-W components), its vector modulus  $M(t)$  and  $N(t)$  function from the output of a trained BPNN used in Chapter 4. Two vertical lines are automatically drawn by the BPNN and exactly indicate P- and S-arrival onset times without error. The dashed line on  $F(t)$  shows the picking threshold (0.6) applied to  $N(t)$ .

is changed. This shows that seismic arrivals have some general characteristics. This BPNN trained with *P*-arrivals has learned these general characteristics, and can use them to pick other kind of arrivals, even from other stations. However, if an arrival has specific characteristics which are quite different from this general character, the BPNN will not be able to pick it.

In comparing the noise bursts picked from three 1-C recordings, the result from V-C recordings is better than those from N-S and E-W component recordings which have similar results. Most of the noise bursts are similar to the seismic arrivals. Even in manual analysis, it is also difficult to discard them by only using the noise burst segments itself. An analyst must compare the noise burst with other picks, even with the data from other stations to discard them.

The above results suggest that a trained BPNN can be applied to all three components individually. It is then possible to combine and compare results from each component, in which some arrivals may be absent, to obtain both *P*- and *S*-arrival onset times, such as the case shown on Figures 5.12, 5.13 and 5.14. Even if one of the three components fails, the other two can give useful results. The arrival times picked by the BPNN on three 1-C recordings are, sometimes, different, which is thought to be due to the effect of raypath or variations in the inhomogeneous upper crust.

## 5.5 DISCUSSION AND SUMMARY

### 5.5.1 Comparison with 3-C method.

Table 5.2 also shows the picking results from 3-C method. Compared with the results from the 1-C method, the 3-C method has quite obviously a better performance, being more accurate. For example, in the 3-C method, 74.5% of the *P*-arrivals and 63.2% of the *S*-arrivals are picked with error  $\leq 10$ ms, but in the 1-C application the best result is only 66.2% for the *P*-arrivals from the V-C recordings and 61.2% for the *S*-arrivals from the E-W recordings picked with the same error.

Visually checking the seismograms shows that the arrivals onset times picked from the 3-C modulus are closer to the manual picks than those from 1-C recordings. The seismic arrivals on 1-C recordings are sometimes not clear or missing, especially for the *S*-arrivals, for example, only 175 *S*-arrivals are manually picked from the V-C recordings of station AY, half of those (305) picked from the modulus. But even of the 175 *S*-arrivals, some have no clear first motion, or very low SNR and amplitude. In this case, the BPNN loses its ability to detect them. It seems that if an analyst can manually pick more arrivals, so can the BPNN; but if the analyst has difficulty, the BPNN has also. This is due to the fact that the BPNN is trained by manual picking results and uses this experience to deal with new data, so that it cannot go beyond the manual picking results.

Comparing the noise bursts picked from 1-C recordings and 3-C recordings, the 3-C method is found to be superior to the 1-C method and can efficiently suppress the noise output. This is because that the seismic arrivals in three 1-C recordings are synchronous and the noise are not synchronous. The modulus of 3-C can sum up the changes of an arrival which are projected on the three 1-C recordings, but the noise in the three 1-C recordings may counteract each other. Using 3-C recordings also requires a small BPNN so that its speed in training and processing is quicker than that of 1-C recordings.

### 5.5.2 Weight pattern analysis

In order to better understand the mechanism by which this BPNN performs its classification, a weight map for this BPNN is shown on Figure 5.15. This map is similar to those shown in Chapter 4, but the BPNN with 40 input nodes was trained by ten pairs of *P*-arrival and background noise segments. Before training, the weights and thresholds were randomly initialized between -0.5 and +0.5.

This map shows that, after training, the values of weights and thresholds range from -2.41 to 2.26. Like the weight pattern in Figure 4.17, this pattern also shows that the weights connecting the input layer to the hidden layer are divided into two portions at the twentieth sample, corresponding to the *P*-arrival onset in the input

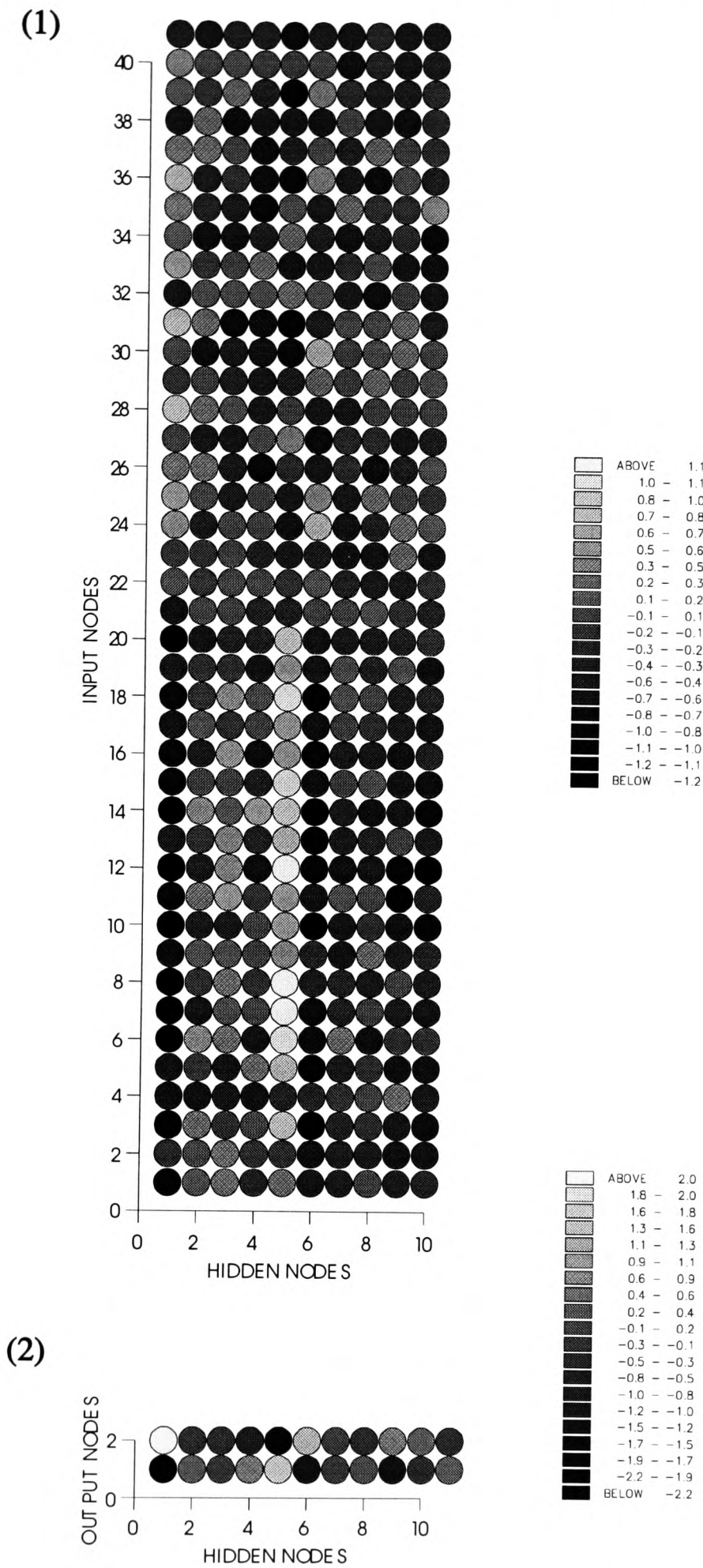


Figure 5.15. Weight map for a trained BPNN with 40 input nodes. Notation as in Figure 4.4.A. This BPNN is trained with ten pairs of absolute value segments of Vertical component of P-arrival and noise signals. The training parameters are: learning rate = 0.7, momentum rate = 0.9, system error threshold = 0.00001 and pattern error threshold = 0.00001.

signal where before the onset the signal energy is low and after the onset the signal energy is high. The first portion between the first and twentieth nodes also shows a "high contrast" weight pattern in which the absolute values of most weights are large, for example, most weights connected to the *first* and *sixth* hidden nodes are larger negative, and most weights connected to the *fifth* hidden node are larger positive, but the second portion between the twenty-first and fortieth nodes also shows a "low contrast" weight pattern in which the absolute values of most weights are small. The weight pattern between the hidden nodes and the output nodes is relative to the pattern between the input nodes and hidden nodes. For example, the weights connected to *first*, and *sixth* hidden nodes have large negative values when they connect to the *noise* output node and large positive values when they connect to the *P*-output node, and the weights connected to the *fifth* hidden node have large positive values when they connect to the *noise* output node and large negative values when they connect to the *P*-output node. It seems that the *first*, *fifth*, and *sixth* hidden nodes are active in the performance because the weights connected to them have larger values than others.

Comparison with the weight map in chapter 4 shows that the two BPNNs have similar weight patterns, although their structure and the training datasets are quite different. Their similar weight patterns means they operate in the same way and can do the same work. The significant feature in the two weight patterns is that each of them is divided into two portions at the onset of the training *P*-arrival segments. These two portions have different functions in doing the picking work. For example, the "high contrast" portion has a strong effect on picking *P*-arrival and the "low contrast" portion has a strong effect on picking background noise. Another feature is that some hidden nodes are more active than others.

### 5.5.3 Summary

A BPNN is used as a tool to pick *P*- and *S*-arrivals from three 1-C recordings. The input of this BPNN is the absolute value of the 1-C seismic recordings. This BPNN is trained by a small subset of the data (ten *P*-arrival and background noise



segments) from the V-C recording of station DP, and can successfully detect and pick seismic arrivals not only from V-C recordings but also from the other two horizontal components and other stations. The performance is different for each of the three 1-C recordings due to strong effects of raypath and source position. The effect on *P*-arrivals is smaller than on *S*-arrivals. For example, the picking rates for *P*-arrivals are 93.1% from the V-C recordings, 89.4% from the E-W recordings, and 83.1% from the N-S recordings; but the picking rates for *S*-arrivals are 75.0% from the V-C recordings, 90.9% from the E-W recordings, and 87.2% from the N-S recordings. The accuracy of the onset times picked from each individual 1-C recordings is similar. For example, with the error  $\leq 10\text{ms}$  (one sample), 66.2% , 59.2% and 63.3% of the *P*-arrivals and 52.7%, 61.2%, and 57.7% of the *S*-arrivals are picked from the V-C, E-W and N-S recordings respectively.

In comparison to the 3-C picking, the performance of 1-C picking is lower, with the associated disadvantage of a larger BPNN structure. It depends on the raypath effects, and the accuracy of arrival onset times is reduced. But it has an obvious advantage of flexibility over 3-C picking as it can be used on 1-C recordings when 3-C recordings are not available.

The work in this and the preceding chapters demonstrate the adaptive nature of the BPNN. Both the 1-C approach and the 3-C approach employ the same computer programs. Only the BPNN structure and the training dataset are changed. Further, without changing the programs, this BPNN approach might also be used to deal with regional and teleseismic earthquake data.

## CHAPTER 6:

### ARRIVAL TYPE IDENTIFICATION USING BPNN

#### 6.1 INTRODUCTION

As I mentioned in Chapter 1, the estimation of arrival onset times includes two steps: 1) arrival picking, which is a reliable and accurate estimation of the onset times of seismic arrivals; and 2) arrival identification, which classifies individual arrivals into categories relating to their polarization, amplitude and propagation characteristics. In Chapters 4 and 5, I developed a BPNN approach for arrival picking. In this chapter, I will develop another BPNN approach to tackle the problem of arrival identification.

For the purpose of automation of seismic analysis, identifying arrival types is more difficult than picking their onset times. Perhaps, the most efficient method of identifying arrival types is still manual analysis. In some cases, though, for a seismic network, arrival identification based on horizontal velocity from a  $f$ - $k$  filter can provide a major simplification of the interpretation task (Mykkeltveit and Bungum, 1984; Bache *et al.*, 1990; Kvaerna and Ringdal, 1992). Der, Baumgardt and Shumway (1993) have also investigated the feasibility of adaptive and automatic recognition of regional arrivals by a wavefield extrapolation scheme for data from a mini-array. However, for single station data, only a few methods can be used to pick some special types of arrivals. Roberts, Christoffersson and Cassidy (1989), based on the auto- and cross-correlations of the three orthogonal components within a short time window, detect the arrival of a  $P$ -wave or a linearly polarized  $S$ -wave. Cichowicz (1993) developed an  $S$ -phase picker which depends on a well-defined pulse of the first-arrival  $P$ -wave. Tong (1995) developed a phase separator based on phase features extracted from intelligent segmentation of seismograms. Tong and Kennett (1995) also developed an approach to identify later seismic phase by analysing the energy content of seismic traces. There is no general method to identify the  $P$ - and  $S$ -arrival

simultaneously and the automation of identifying arrivals is still an unresolved problem. In this chapter, a BPNN approach is designed to identify both *P*- and *S*-arrivals by using the polarization information of the seismic arrivals. Although not perfect, this method is shown to be more successful overall than those mentioned above.

## 6.2 DEGREE OF POLARIZATION

One of the features distinguishing *P*- and *S*-arrivals is their polarization directions, i.e. the polarization direction of a *P*-arrival is parallel to its propagation direction, and the polarization direction of an *S*-arrival is perpendicular to its propagation direction in an isotropic medium. It seems simple to identify *P*- and *S*-arrivals by comparing their polarization directions with their propagation directions because calculating the direction is easy with the 3-C recordings now available. However, it is not practicable because the polarization direction is related to the propagation direction which is not available before an event is analyzed.

Another feature distinguishing the *P*- and *S*-arrivals is their polarization state. In general, it is observed that the direct *P*-arrival is predominately linearly polarized while the arrivals following this direct *P*-arrival, such as *S*-arrivals, have considerably more complex polarization patterns involving phase shift (Basham and Ellis, 1969; Roberts and Christoffersson, 1990). The polarization state of a seismic arrival can be measured by using the degree of polarization (DOP) discussed in Section 3.3.3. According to this definition, the DOP is independent of the source orientation. For a linearly polarized wave, the DOP equals 1.0, and for a completely unpolarized or circularly polarized wave, it equals 0.0 (Cichowicz, Green and Brink, 1988). Cichowicz (1993) pointed out that both *P*- and *S*-wave arrivals exhibit a high degree of linear polarization, but the *P*-wave coda manifests a generally elliptical polarization with a significantly lower value of DOP. For real data, although the first *S*-arrival is usually associated with a far larger value of the DOP than that of the *P*-wave coda,

it does not reach a value of 1.0 due to the effect of the *P*-wave coda. The variation of this quantity along the seismogram forms a pattern which may indicate the type of a wave. Figure 6.1 shows a typical example in which the DOP has a high value pattern for a *P*-arrival, a middle value pattern for an *S*-arrival and a low value pattern for a noise burst.

This particular definition of DOP does not consider the signal amplitude. Different arrival types not only have different polarization characteristics, but also have different amplitude characteristics. To consider both polarization and amplitude information, I define a modified function of the DOP:

$$MF(t) = F(t) \times \bar{M}(t) \quad (6.1)$$

Where  $\bar{M}(t)$  is a smoothed relative function of modulus  $M(t)$  of the 3-C recording in a window which is also independent of the source position. The normalization factor is taken from the window between the onset point and the following ten points, in which the maximum is defined as unity. Note that  $MF(t)$  and  $F(t)$  may have slightly different patterns. As the  $MF(t)$  patterns are complex and identifying them requires an intensive amount of the pattern recognition, it is difficult to find a method which is both simple and reliable to distinguish their types. Here the identification is accomplished by using a BPNN to recognize the  $MF(t)$  patterns. The  $MF(t)$  is presented to the BPNN in a segment selected from a window in which the arrival onset-time is at the centre.

### 6.3 APPROACH OF IDENTIFYING ARRIVAL TYPES USING BPNN

Fig 6.2 shows a flow chart of the approach of identifying arrival types using a BPNN. In this approach, unlike the BPNN approach of picking arrivals developed in Chapters 4 and 5, which use a BPNN as a filter to deal with an entire seismic trace, only arrival segments are input into the BPNN. An arrival segment is selected by its onset time which is obtained using another method, such as the conventional STA/LTA method

Station: DP  
 Date: 1984-05-06  
 Start-time: 12h16m08s  
 Scale: 564

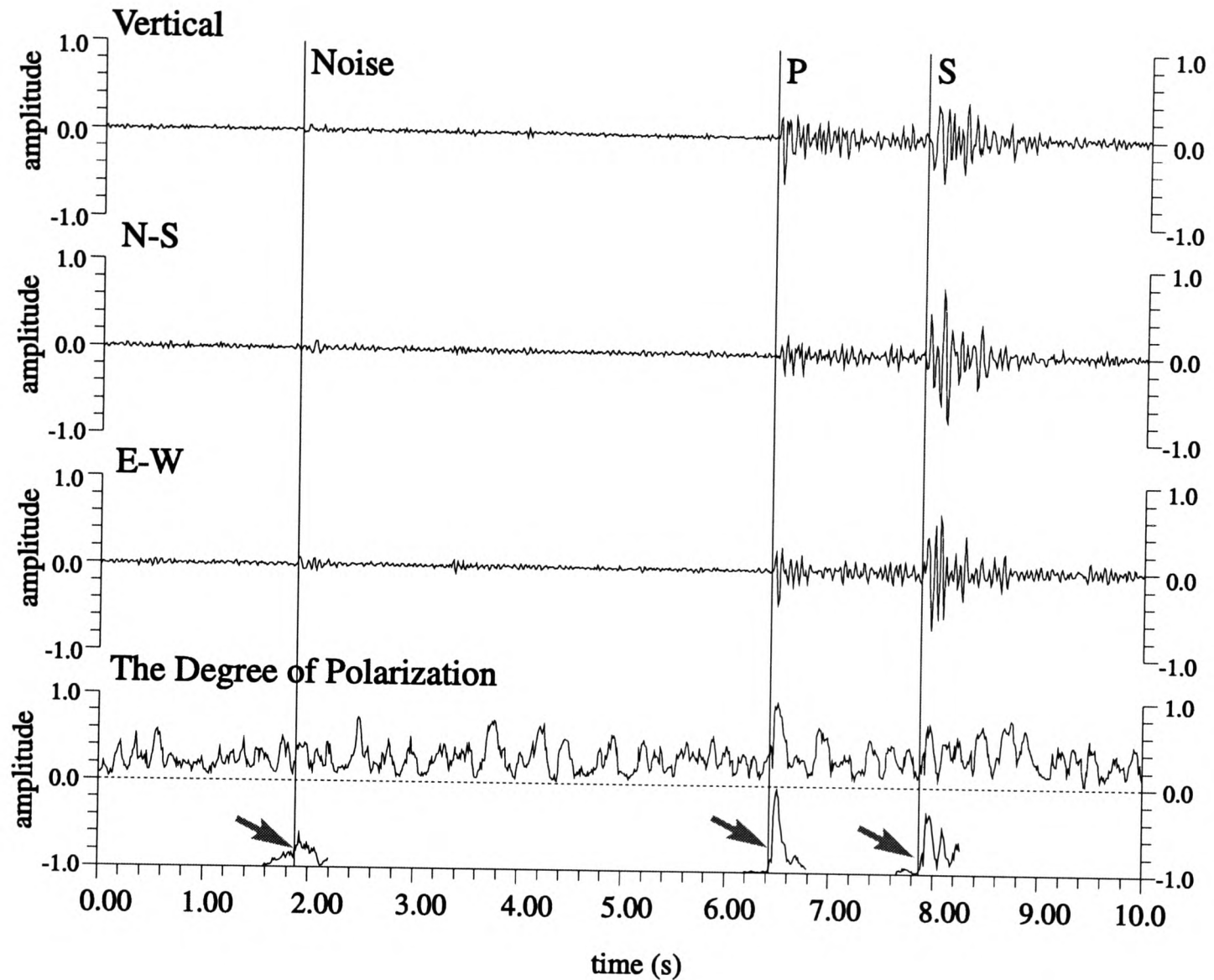


Figure 6.1 The degree of polarization (lower diagram) determined from 3-C seismograms (upper diagram). Three vertical lines indicate the arrival onset times of a noise burst, a P-arrival and an S-arrival. The degree of polarization has a low value for the noise burst, a high value for the P-arrival and a middle value for the S-arrival. The corresponding modified DOP are shown below them respectively (arrowed).

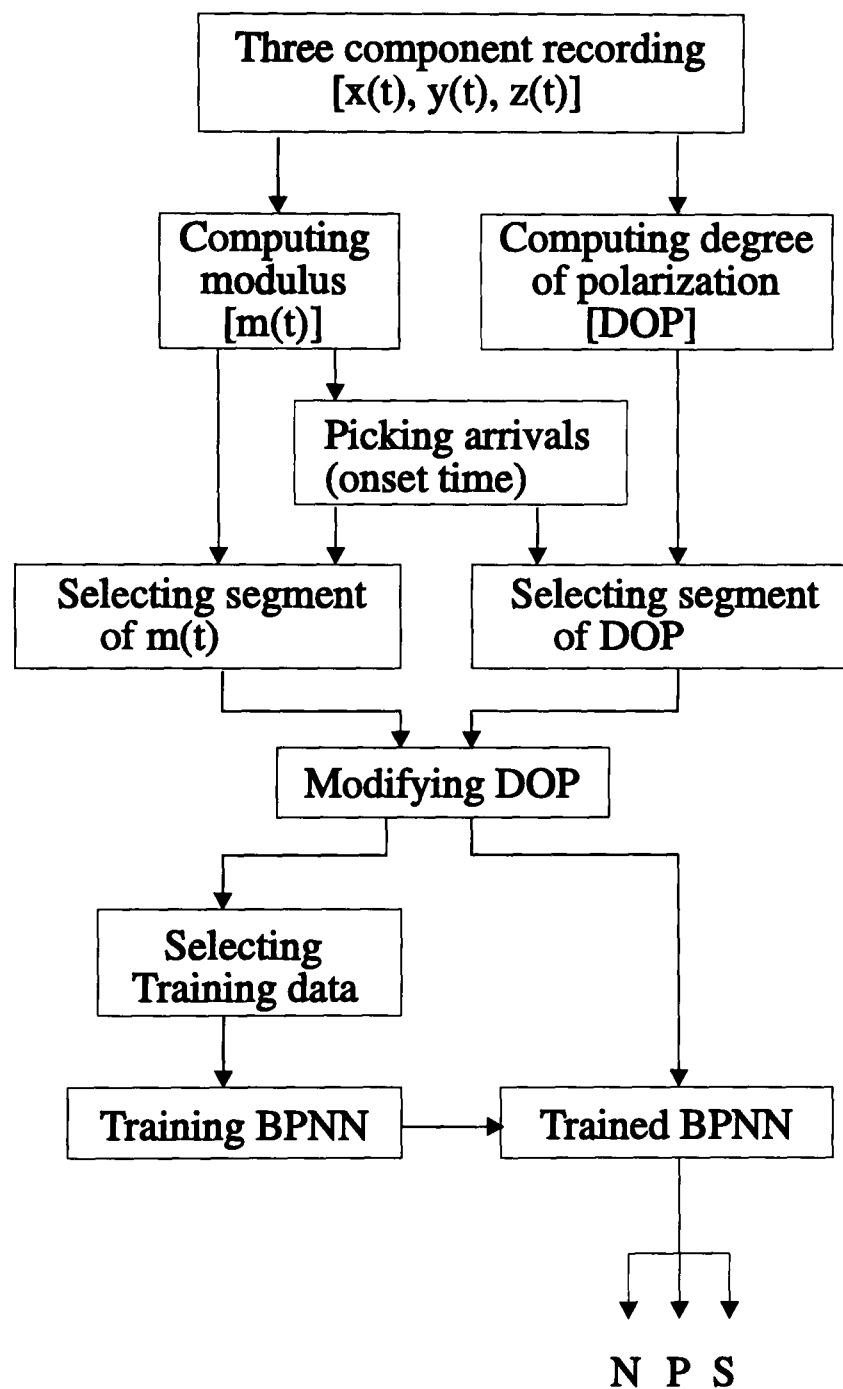


Figure 6.2 The flow chart of the approach of identifying arrival types using a BPNN.

or my own BPNN picker. Here, I use the BPNN picker developed in Chapter 4 to pick the arrival onset times.

The first step in this approach is to calculate the vector modulus ( $m(t)$ ) and the DOP. The second step is to use a BPNN arrival picker to pick possible arrivals and measure their onset times. The third step is to use the onset times to select the segments of DOP and the modulus. The fourth step is to modify the DOP by using the smoothed relative modulus. The modified DOP is then fed into a BPNN for training and testing. The fifth step is to select a suitable training dataset of the modified DOP to train the BPNN. The final step is to use this trained BPNN to process the new data and identify the arrival types.

### 6.3.1 Input characteristics: modified DOP, $MF(t)$

In this approach, the input to the BPNN is the  $MF(t)$  segment of an arrival which is calculated using Equation 6.1. This arrival segment is selected by its onset time which is previously picked and the onset time is positioned at the centre of the segment. Due to the picking error, the onset time may not be exactly at the centre of the segment. It is found that this may affect the output. In order to avoid this effect, an adjustment of the onset time is necessary to ensure that the performance of the trained BPNN is not affected by the onset time error. For each  $MF(t)$  segment, its first local maximum after the onset-time is set at the centre of the segment. The  $MF(t)$  segment has 60 samples or 590ms length which are chosen to include the complete  $MF(t)$  pattern of an arrival.

### 6.3.2 BPNN structure

The BPNN used in this approach also has three layers (Figure 6.3). Its input layer has 60 nodes, giving a  $MF(t)$  segment with a fixed 590ms (60 samples) length. There are three nodes in its output layer to flag the result: the output is (1,0,0) for a noise burst, (0,1,0) for a  $P$ -arrival, and (0,0,1) for an  $S$ -arrival in training. Ten hidden nodes are chosen after a process of trial and error with different training runs.

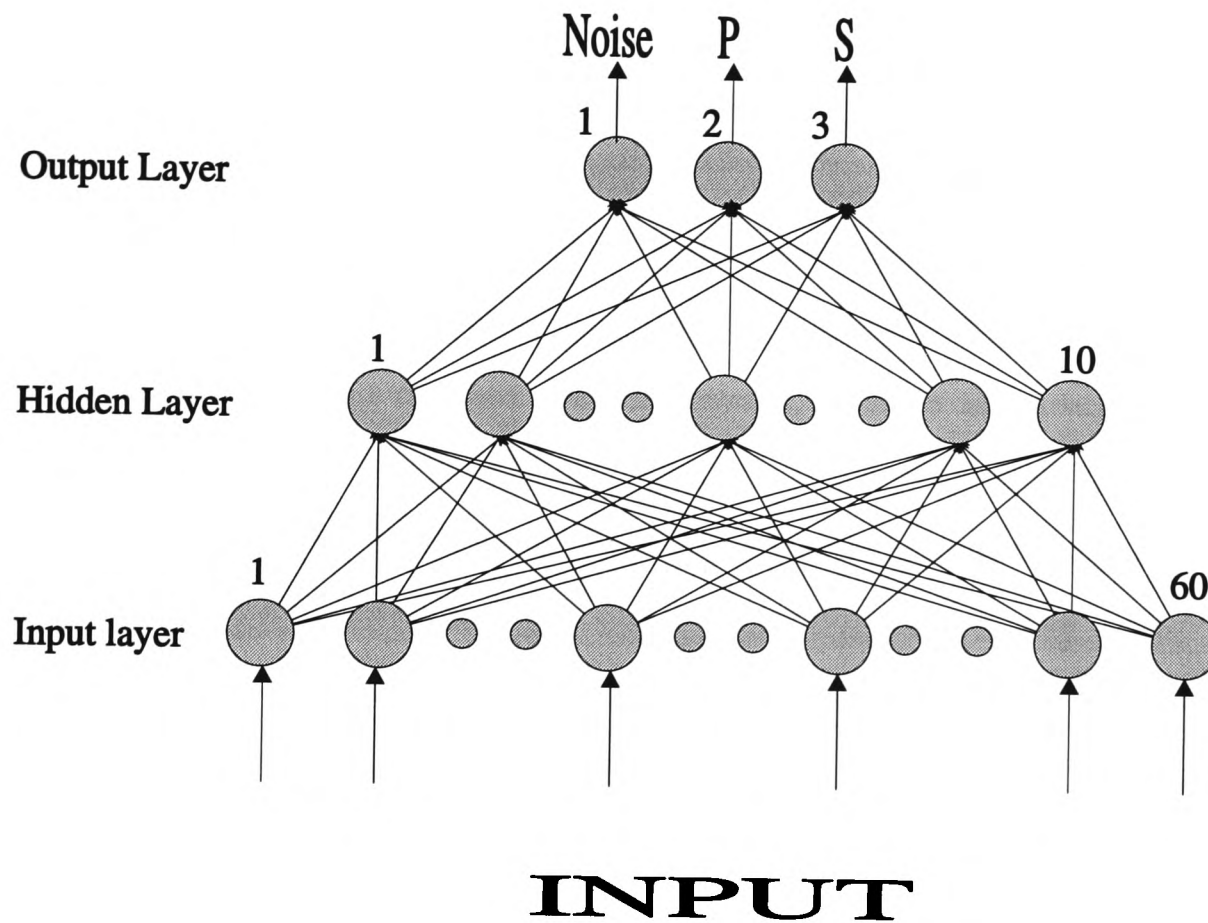


Figure 6.3. The structure of a BPNN for seismic arrival identification. It has three layers including an input layer with 60 nodes, a hidden layer with 10 nodes and an output layer with three nodes. The BPNN input is a MF(t) segment of the 3-C seismogram. The three output nodes indicate the input segment with (1,0,0) as a noise burst, (0,1,0) as a P-arrival and (0,0,1) as an S-arrival.



### 6.3.3 Training procedure

In this procedure, only a small number of recordings from station DP are used to train the BPNN. As the BPNN performance depends on the training datasets, selecting the training dataset is crucial. For example, if incorrect or inconsistent data are used to train the BPNN, the BPNN cannot be expected to give a correct answer to new data. *P*- and *S*-arrivals with similar  $MF(t)$  patterns should be avoided in the training datasets because the BPNN cannot distinguish them and the training procedure might not be convergent. At the beginning of training, only three  $MF(t)$  segments are selected for training, including a noise burst, a *P*-arrival and an *S*-arrival with the desired output (1,0,0; 0,1,0; 0,0,1) respectively. After training, this BPNN is used to handle other data. Using manual analysis results, another three segments, which are wrongly identified by this BPNN, are selected, combining with the former training segments, to train this BPNN again. This procedure is repeated until the performance of the trained BPNN cannot be improved by adjusting the training dataset. Figure 6.4 shows all training segments of  $MF(t)$  used in this particular study.

The training parameters used in this approach are still the same as those used in Chapters 4 and 5:

learning rate = 0.7

momentum rate = 0.9

system error threshold = 0.00001

pattern error threshold = 0.0001

With the above selected training dataset and training parameters, the final training procedure with nine groups of training datasets from station DP took 1352 iterations (less than two minutes CPU time on a VAX4000). The system error reached 0.000023 with all pattern errors less than 0.0001. After this training procedure, the BPNN is ready to identify the seismic arrivals in the whole dataset.

I have previously used Equation 4.1 to show the relationship between the training parameter (learning rate  $\eta$  and momentum rate  $\alpha$ ) and the training convergence (iteration number). However, in the approach described in this chapter, the BPNN's structure and the training datasets are greatly changed. Can this equation

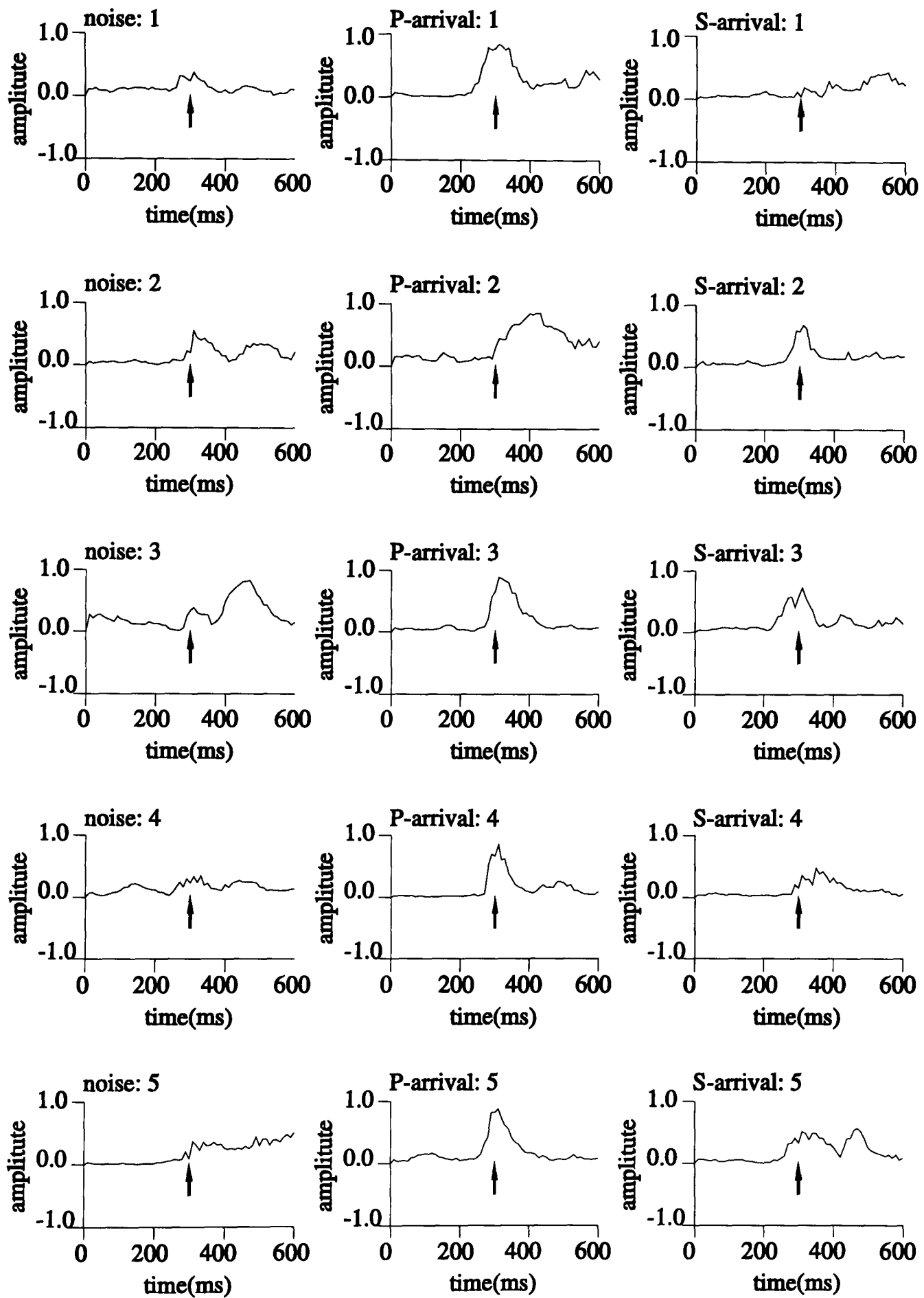


Figure 6.4. Nine groups of MF(t) segments of noise bursts, P-arrivals and S-arrivals for training a BPNN for arrival identification. Arrows on segments indicate the pre-picked onset times for these arrivals and all at the 31st sample.

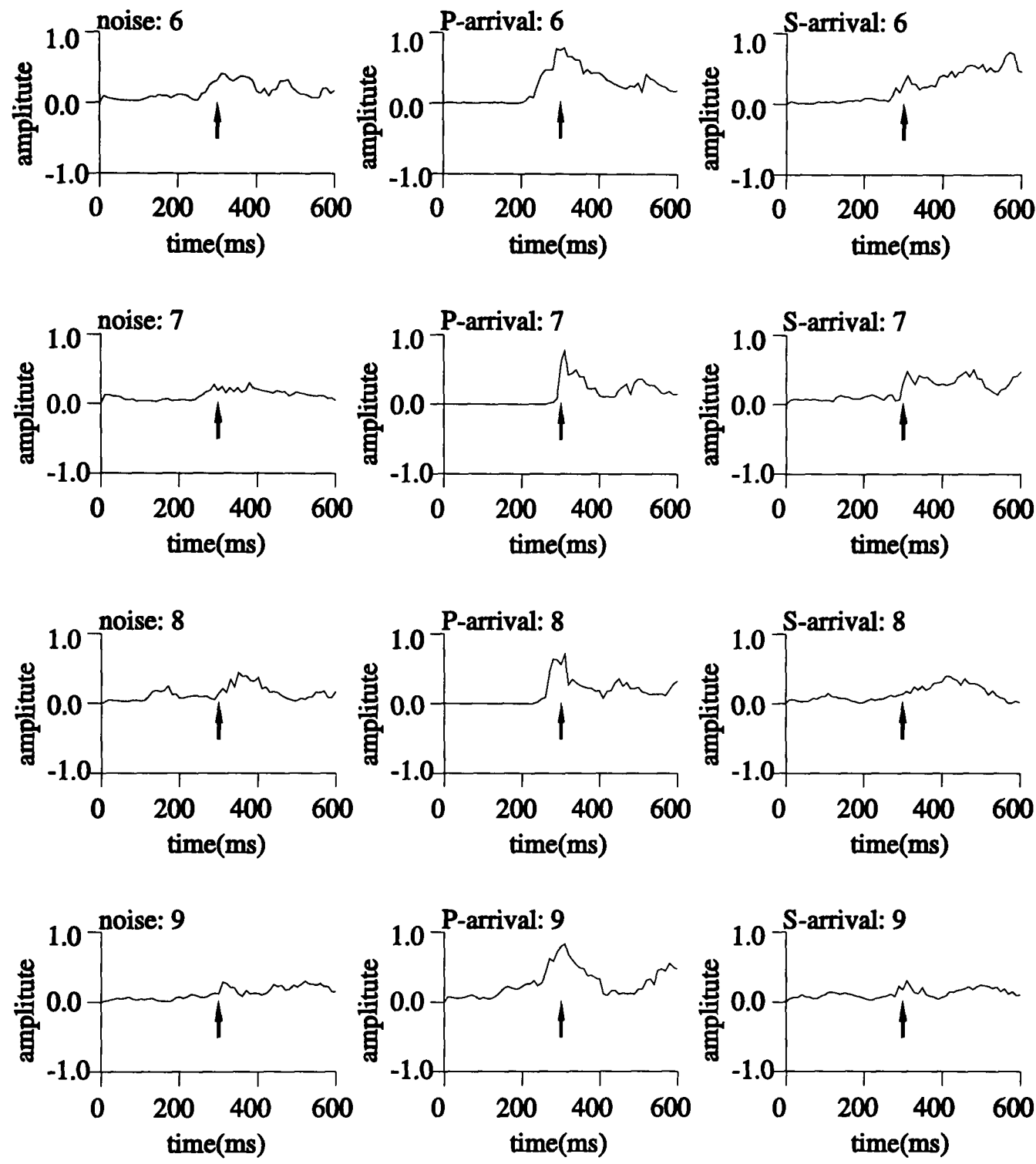


Figure 6.4. (continued).

still be applied to this BPNN? Several experiments were performed to verify this. Table 6.1 shows the training result and the calculated result. Unfortunately, this result is worse than those in Sections 4.3.3.2 and 5.3.3. It means the arrival type identification is less reliable than the arrival picking. For this BPNN, the difference between the real iteration and the calculated one depends on two factors: the learning parameters ( $\eta$  and  $\alpha$ ) and the system error. When  $\eta$  and  $\alpha$  are small, the differences are small. As  $\eta$  increases, the difference increases. This reflects the fact that a large deviation can take place in the weight space with a high  $\eta$  in the training procedure (Pao, 1988). During the learning procedure with a high  $\eta$ , the weight changes might jump over the desired weight state and necessitate re-searching. As a result, the learning procedure requires more iterations to reach this state. Although using the momentum term can reduce this jump, it cannot be totally overcome. A large training parameter may therefore not necessarily result a short learning procedure, in direct contrast to the result in Sections 4.3.3.2 and 5.3.3. The iteration number is also related to the system error. It seems that the deviation of iteration number is proportional to the deviation of the system errors from that at  $\eta = 0.1$  and  $\alpha = 0.0$ . In the training procedure, I have mainly used the pattern error to terminate the training procedure. The training procedure stops when all training patterns have an error less than 0.0001, but the system error, varying with different learning parameters, does not need to reach its threshold (0.00001). A small system error corresponds to a long learning procedure. It seems that a better fit between the real and calculated data is obtained if the system error is used to terminate the training procedure.

#### 6.3.4 Identifying arrival types

As each  $MF(t)$  segment is fed into the trained BPNN, the BPNN outputs three values:  $o_1$ ,  $o_2$ , and  $o_3$ . For the training segments, the output should be desired ones: (1,0,0) for the noise burst, (0,1,0) for the  $P$ -arrival, and (0,0,1) for the  $S$ -arrival. For non-training segments, the output ( $o_1$ ,  $o_2$ , and  $o_3$ ) is a measurement of similarity between a new segment and training segments. If a non-training segment is similar to a training segment, the BPNN output will close to a desired output of the training

**Table 6.1.** The iteration number in training with different learning rate  $\eta$  and momentum rate  $\alpha$ . In each cell, the top number is the training iteration, the middle one is the system error ( $\times 10^{-5}$ ), and the bottom one is the calculated iteration by using Equation 4.1. NC means non-convergence.

Iteration number		$\alpha$										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\eta$	0.1	20809 2.23 20809	18737 2.24 18728	16667 2.24 16647	14586 2.25 14566	12510 2.27 12485	10431 2.29 10405	8356 2.33 8324	6293 2.41 6243	4324 2.41 4162	2405 2.10 2081	NC
	0.2	10615 2.35 10405	9628 2.32 9364	8617 2.30 8327	7600 2.27 7283	6588 2.24 6242	5587 2.19 5202	4577 2.13 4162	3499 2.12 3121	2351 2.70 2081	1356 2.27 1040	NC
	0.3	7288 2.25 6936	6587 2.24 6243	5883 2.23 5549	5176 2.22 4855	4464 2.22 4162	3744 2.24 3468	3033 2.31 2775	2507 2.45 2081	1912 2.18 1387	1051 3.08 694	NC
	0.4	5528 2.66 5202	5199 1.90 4682	4535 2.51 4162	3890 2.35 3641	3396 2.30 3121	2989 2.26 2601	1906 2.35 2081	2672 1.19 1561	1439 2.19 1040	2369 1.81 520	NC
	0.5	5466 1.91 4162	4418 2.79 3746	4029 2.86 3329	3933 1.80 2913	3535 1.53 2497	2608 2.19 2081	2319 1.68 1665	3044 1.00 1249	1788 1.36 832	1270 2.14 416	NC
	0.6	3851 3.31 3468	3523 3.00 3122	3291 3.04 2775	3039 3.41 2428	3430 1.84 2081	3553 1.14 1734	3127 1.00 1387	2572 1.00 1040	2188 1.16 694	2051 1.61 347	NC
	0.7	3567 2.64 2973	3309 2.35 2675	2982 2.41 2378	2650 2.80 2081	2829 1.71 1784	2936 1.07 1486	2678 1.00 1189	2087 1.78 892	1696 2.34 585	1352 2.28 297	NC
	0.8	2827 3.62 2601	2754 2.76 2341	2592 2.30 2081	2327 2.43 1820	2468 1.67 1561	2818 1.00 1300	2659 1.34 1041	1589 1.70 780	2507 1.28 520	1833 1.45 260	NC
	0.9	2747 2.98 2312	3067 1.97 2081	2402 2.14 1850	2674 1.67 1618	2694 1.13 1387	2627 1.31 1156	2795 1.19 925	1422 3.03 694	2898 1.60 462	1204 1.60 231	NC
	1.0	2980 1.40 2081	2245 2.60 1873	2794 1.27 1665	3036 1.00 1457	3130 1.00 1249	2037 2.42 1041	1734 1.76 832	1883 2.01 624	1342 2.69 416	547 3.22 208	NC
	1.1	2397 1.72 1892	1923 2.39 1703	1776 2.77 1513	2126 1.63 1324	2220 1.84 1135	1604 2.84 946	1774 1.50 757	1437 1.31 568	2079 2.15 378	3762 1.23 189	NC

segment. In order to identify segment types, I only need to seek the maximum of the three outputs ( $o_1$ ,  $o_2$ ,  $o_3$ ). If  $o_1$  is the maximum, this segment belongs to noise bursts; if  $o_2$  is the maximum, it belongs to  $P$ -arrivals; and if  $o_3$  is the maximum, it belongs to  $S$ -arrivals. Normally, the trained BPNN should have a high output and two low outputs in its three output nodes. However, sometimes it has two or three high outputs or three low outputs. This means these segments are quite different from training segments, but their types can still be identified by the maximum of the BPNN's output.

## 6.4 PERFORMANCE OF TRAINED BPNN

To test the trained BPNN performance, all the previously picked arrival segments are fed into this trained BPNN. As I mentioned in Section 3.3.3, the  $MF(t)$  patterns are different for the data from stations DP and AY. In order to analyse the performance of this approach clearly, I will analyse the data from stations DP and AY separately.

### 6.4.1 BPNN's performance for data from station DP

For the data from station DP, a BPNN picker picked 345  $P$ -arrivals, 302  $S$ -arrivals and 174 noise bursts from 371 recordings. A BPNN for arrival identification is finally trained with nine groups of  $MF(t)$  segments of noise bursts,  $P$ -arrivals and  $S$ -arrivals from station DP shown on Figure 6.4. Figure 6.5 shows an example in which this trained BPNN can correctly identify the arrival types. For the noise burst and  $P$ -arrival, the trained BPNN outputs one high value and two low values in its output nodes, (1.1, 0.0, -0.1) for the noise burst, (0.0, 1.0, -0.1) for the  $P$ -arrival, closing to the training output, but for the  $S$ -arrival, its output has one high, one middle and one low values (-0.1, 0.6, 1.1), which is rather different from the training output. Table 6.2.(a) shows the performance of this trained BPNN for identification. Figure 6.6 shows the identifying results related to event positions. This BPNN has a better performance in identifying the  $P$ -arrivals (82.3%) than for identifying  $S$ -arrivals

Station: DP  
 Date: 1984-05-31  
 Start-time: 09h23m36s  
 Scale: 373

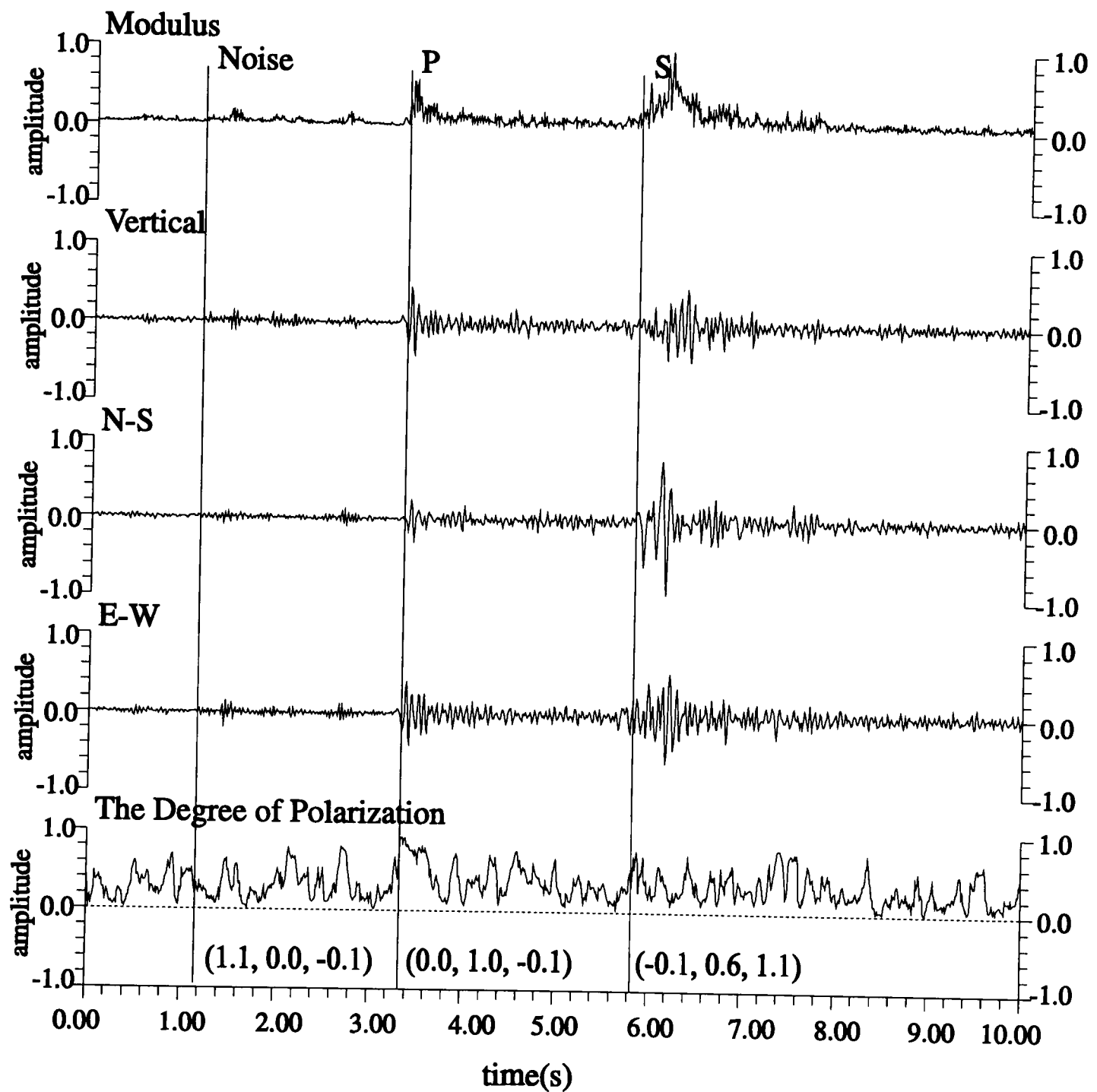


Figure 6.5. 3-C seismograms, the vector modulus, and the degree of polarization of a local earthquake. Three vertical lines indicate the arrival onsets of a noise burst, a P-arrivals and a S-arrival. In this case, the BPNN correctly identifies them with its output (1.1, 0.0, -0.1), (0.0, 1.0, -0.1) and (-0.1, 0.6, 1.1) respectively.

**Table 6.2** The performance of the trained BPNN for arrivals identification. This BPNN is trained with 9 groups of training segments from station DP and has 60 input nodes.

(a) identifying results for the data from station DP

Pre-picked arrivals	P-arrivals (345)	S-arrivals (302)	Noise (174)
BPNN identifying P	82.3% (284)	22.0% ( 67)	9.2% (16)
BPNN identifying S	10.4% ( 36)	62.6% (189)	43.0% (75)
BPNN identifying N	7.2% ( 25)	15.2% ( 46)	47.7% (83)

(b) Identifying results for the data from station AY.

Pre-picked arrivals	P-arrivals (274)	S-arrivals (240)	Noise (28)
BPNN identifying P	42.0% (115)	48.8% (117)	42.9% (12)
BPNN identifying S	43.8% (120)	38.8% ( 93)	25.0% ( 7)
BPNN identifying N	14.2% ( 39)	12.5% ( 30)	32.1% ( 9)



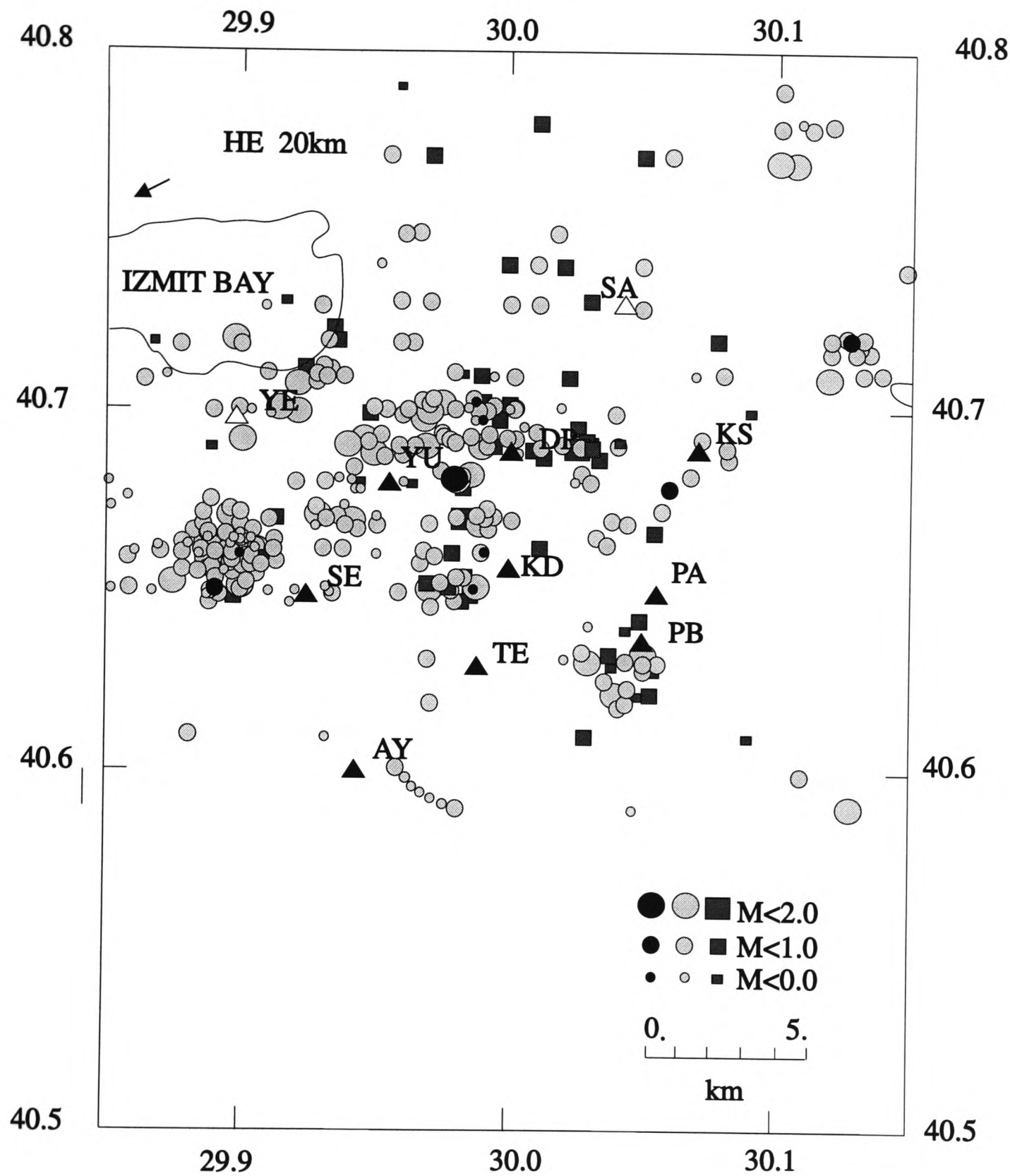


Figure 6.6(a). The location map of events which recorded in station DP. Each symbol represents an event whose P-arrival was picked for identification. Grey circles represent P-arrivals which are correctly identified by the BPNN, and grey squares represent P-arrivals which are wrongly identified by the BPNN. Black circles represent the training P-arrivals.

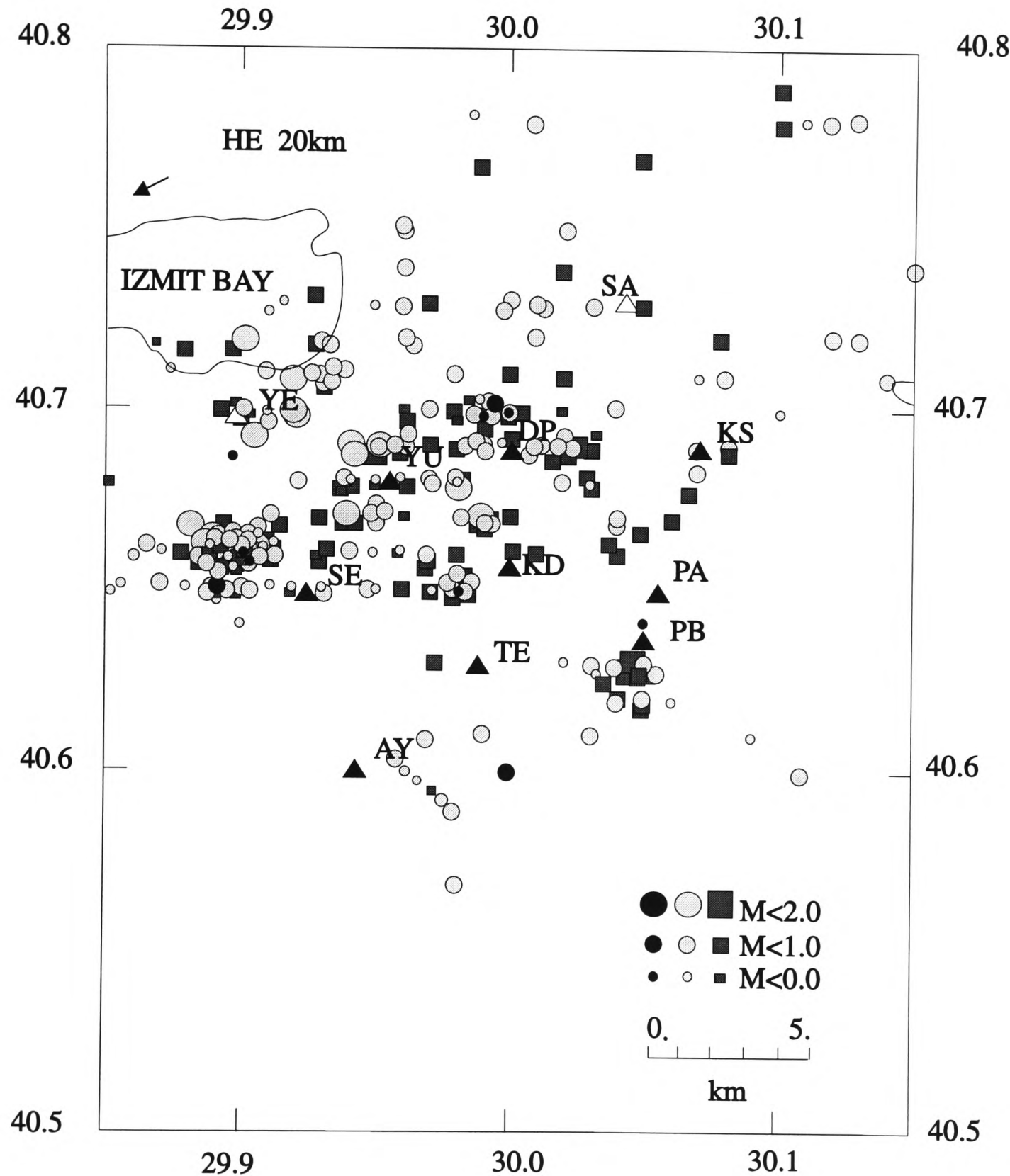


Figure 6.6(b). The location map of events which recorded in station DP. Each symbol represents an event whose S-arrival was picked for identification. Grey circles represent S-arrivals which are correctly identified by the BPNN, and grey squares represent S-arrivals which are wrongly identified by the BPNN. Black circles represent the training S-arrivals.

(62.6%) and noise bursts(47.7%). Visual checking of the seismograms shows that the  $MF(t)$  patterns of arrivals indicate their types. If a  $P$ -arrival has a  $MF(t)$  pattern which is similar to the pattern of a training  $S$ -arrival or a noise burst, the trained BPNN wrongly classifies it as an  $S$ -arrival or a noise burst. If an  $S$ -arrival has the  $MF(t)$  pattern which is similar to the pattern of a training  $P$ -arrivals or a noise burst, the BPNN wrongly classifies it as a  $P$ -arrival or a noise burst. In these cases, even manual identification cannot make a correct decision by only using the  $MF(t)$  segment itself. Figure 6.7 shows such an example in which the  $P$ -arrival has a  $MF(t)$  pattern with low values, similar to the pattern of a training noise burst, and the  $S$ -arrival has a  $MF(t)$  pattern with high values, similar to the pattern of a training  $P$ -arrival. The trained BPNN classified the  $P$ -arrivals as a noise burst with the output (1.1, 0.0, -0.1) and classified the  $S$ -arrival as a  $P$ -arrival with the output (-0.1, 0.7, 0.6). The fact that the different types of arrivals have similar  $MF(t)$  patterns makes the trained BPNN difficult to deal with them.

Note that 17 recordings have been contaminated with excessive noise. In these recordings, noise bursts are very similar to the coherent seismic arrivals and their  $MF(t)$  patterns are similar to the patterns of  $P$ - or  $S$ -arrivals, therefore the BPNN cannot be expected to identify them correctly. Figure 6.8 shows such an example in which three of the noise bursts are incorrectly classified as  $S$ -arrivals. If such recordings are omitted, there are only 118 noise bursts picked. The BPNN then classifies 11(9.3%) of them as  $P$ -arrivals, 41(34.5%) as  $S$ -arrivals and 66 (56.0%) as noise bursts.

#### 6.4.2 BPNN's performance for data from station AY

For the data from station AY, a BPNN picker picked 274  $P$ -arrivals, 240  $S$ -arrivals and 28 noise bursts from 391 recordings. The above BPNN is applied to these arrivals to identify their types. Table 6.2 (b) shows its identification performance. Unfortunately it only correctly identifies 42% of the  $P$ -arrivals, 38.8% of the  $S$ -arrivals and 32.1% of the noise burst. As I mentioned in Section 3.3.3, the  $MF(t)$  patterns of

Station: DP  
 Date: 1984-06-20  
 Start-time: 04h37m31s  
 Scale: 634

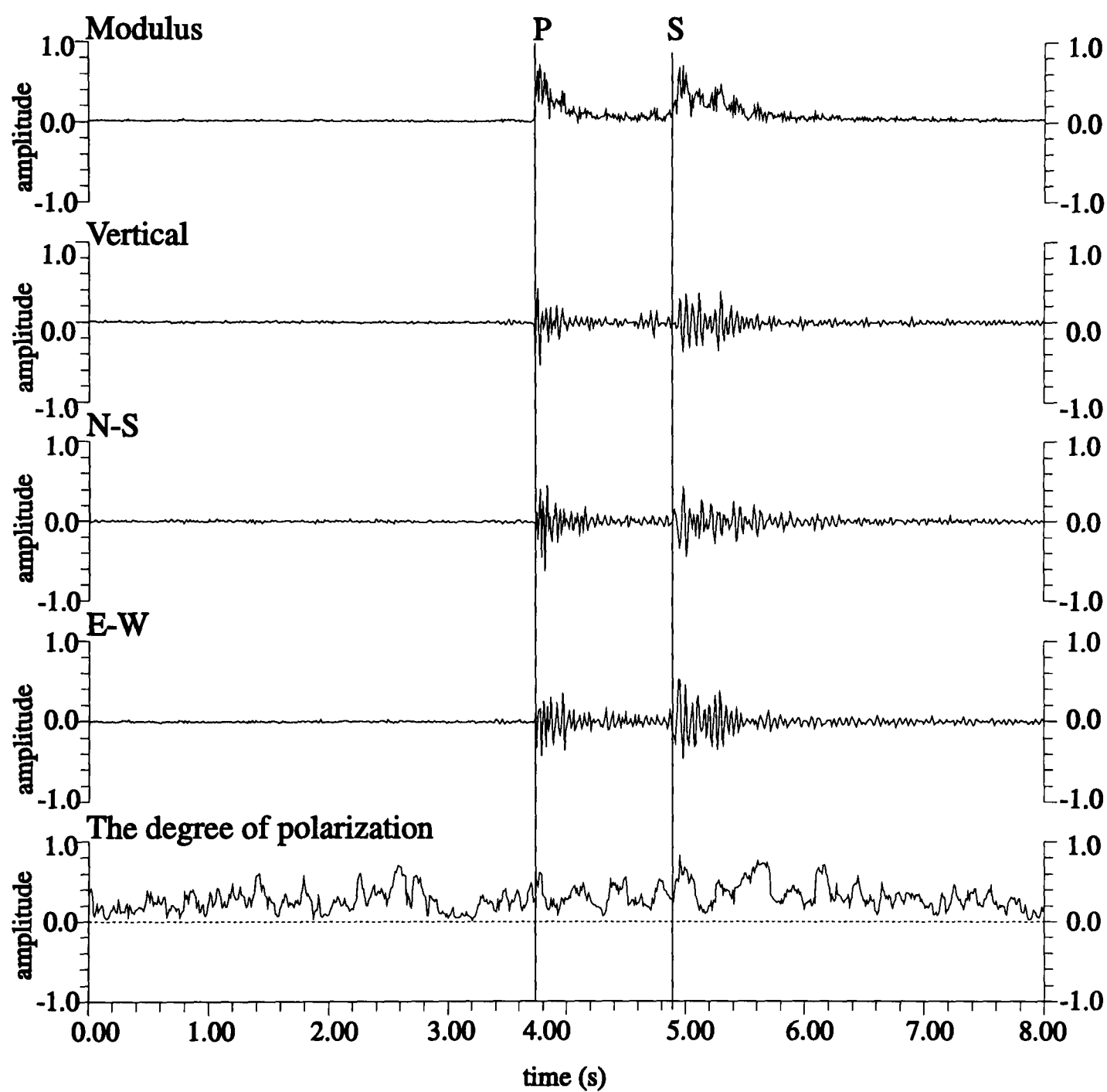


Figure 6.7. 3-C seismograms, the vector modulus, and the degree of polarization of a local earthquake. Two vertical lines indicate the arrival onsets of a P-arrival and an S-arrival. The BPNN output are (1.1, 0.0 -0.1) for the P-arrival and (-0.1,0.7, 0.6) for the S-arrivals.

Station: DP  
 Date: 1984-05-29  
 Start time: 12h57m54s  
 Scale: 267

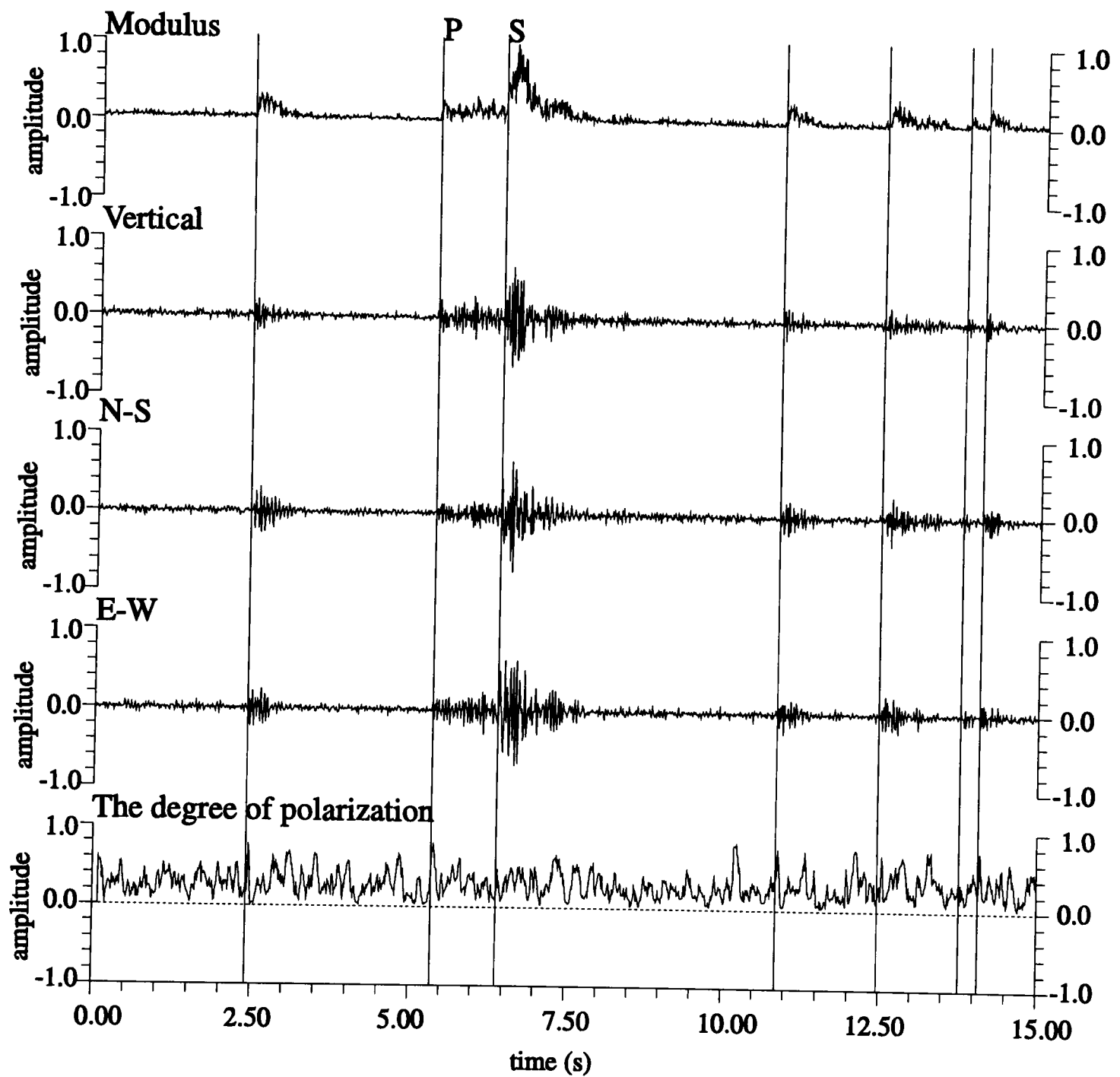


Figure 6.8. 3-C seismograms, the vector modulus and the degree of polarization of a local earthquake. Vertical lines indicate a P-arrival, an S-arrival and some noise bursts. The MF(t) patterns of these noise bursts are similar to the patterns of seismic arrivals.

seismic arrivals are different between the data from stations DP and AY, even for the same arrivals of the same event (Figure 3.9). In the data from station AY, many *P*-arrivals have middle or low values of  $MF(t)$  and many *S*-arrivals have high values of  $MF(t)$ , unlike the training dataset. Such differences make the BPNN fail to identify these arrivals correctly because the output of the trained BPNN is the measurement of the similarity between the testing segment and training segments. For example, some recordings have *P*-arrivals with low or middle values of  $MF(t)$  pattern and *S*-arrivals with high values of  $MF(t)$  pattern, and these patterns are opposite to the training ones, so that the trained BPNN cannot correctly classify them. To deal with the data from station AY, it is necessary to train this BPNN by using the data from station AY, so that it can remember the  $MF(t)$  patterns from station AY and use this to classify the new data from station AY correctly.

Another BPNN with the same structure is trained by using the data from station AY. The best performance is obtained when the BPNN is trained by using five group datasets shown on Figure 6.9. The training procedure took 527 iterations. The system error reached 0.000033 with all pattern errors less than 0.0001. Table 6.3 (a) shows its identification performance. Figure 6.10 shows the identifying results related to event positions. This BPNN has a much better performance for the data from station AY than the one trained with data from station DP, although 76.6% for the *P*-arrivals and 60.4% for the *S*-arrivals is not as good as the previous one for station DP. However, the two BPNNs have a similar performance in dealing with data from their own stations. This BPNN is also used to test the data from station DP (Table 6.3 (b)). Although it can identify 84.9% of the *P*-arrivals, the 36.1% success rate for the *S*-arrivals is too low. Note that the identification performance for the noise bursts seems statistically meaningless because only 28 noise bursts are tested.

### 6.4.3 Effect of training dataset

As the performance of a trained BPNN depends on the training dataset, an investigation of the performance of the trained BPNN is taken by varying the training

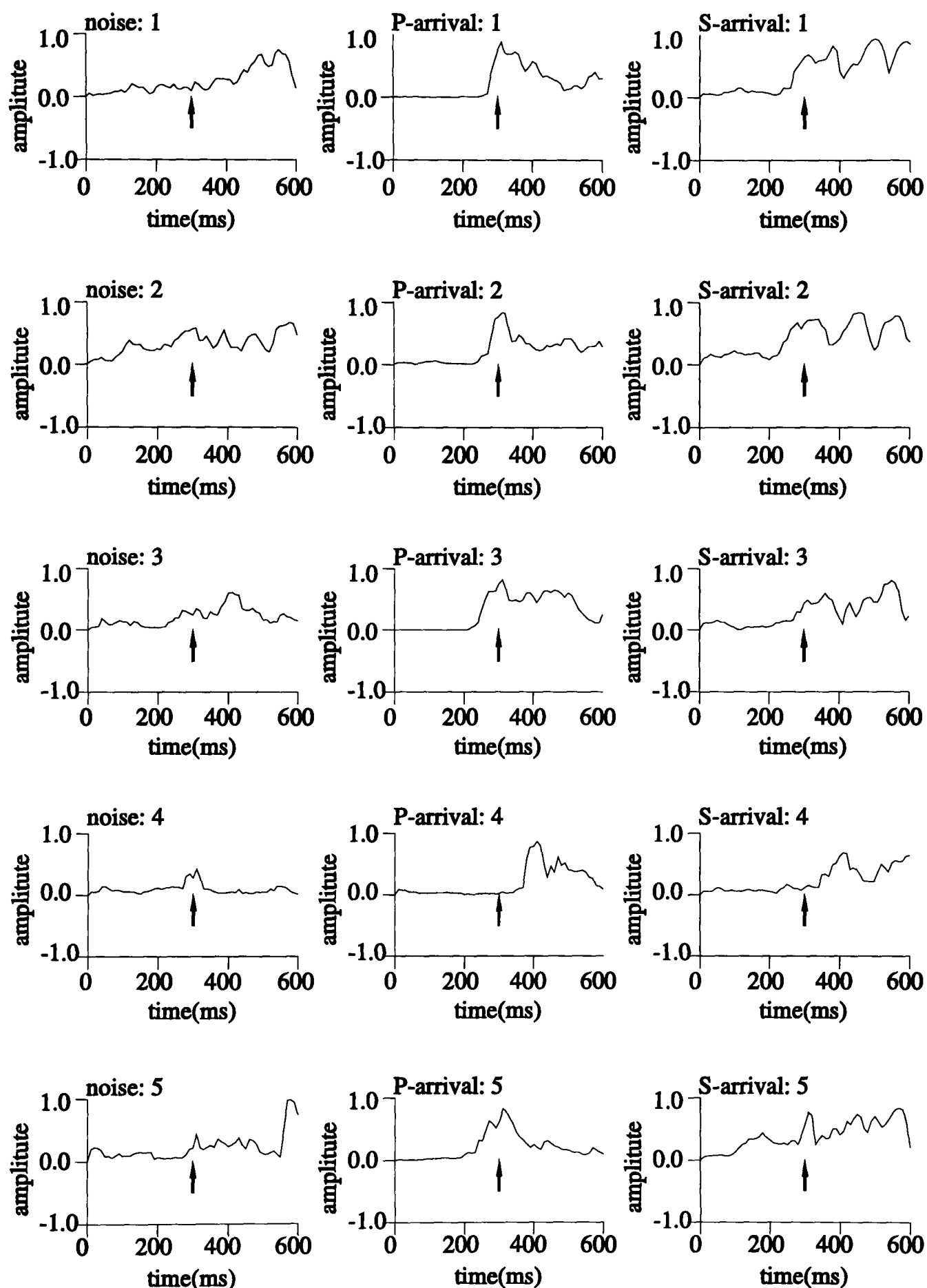


Figure 6.9. Five groups of MF(t) segments of noise bursts, P-arrivals and S-arrivals from station AY for training a BPNN for arrival identification. Arrows on segments indicate the pre-picked onset times for these arrivals and all at the 31st sample.

**Table 6.3.** The performance of the trained BPNN for arrival identification. This BPNN is trained with 5 groups of training segments from station AY and has 60 input nodes.

(a) Identifying results for the data from station AY

Pre-picked arrivals	P-arrivals (274)	S-arrivals (240)	Noise (28)
BPNN identifying P	76.6% (210)	22.1% (53)	21.4% (6)
BPNN identifying S	12.8% (35)	60.4% (145)	53.6% (15)
BPNN identifying N	11.3% (31)	17.5% (42)	32.1% (9)

(b) Identifying results for the data from station DP

Pre-picked arrivals	P-arrivals (345)	S-arrivals (302)	Noise (174)
BPNN identifying P	84.9% (293)	45.4% (137)	30.0% (52)
BPNN identifying S	8.1% (28)	36.1% (109)	17.8% (31)
BPNN identifying N	7.0% (24)	18.5% (56)	52.3% (91)



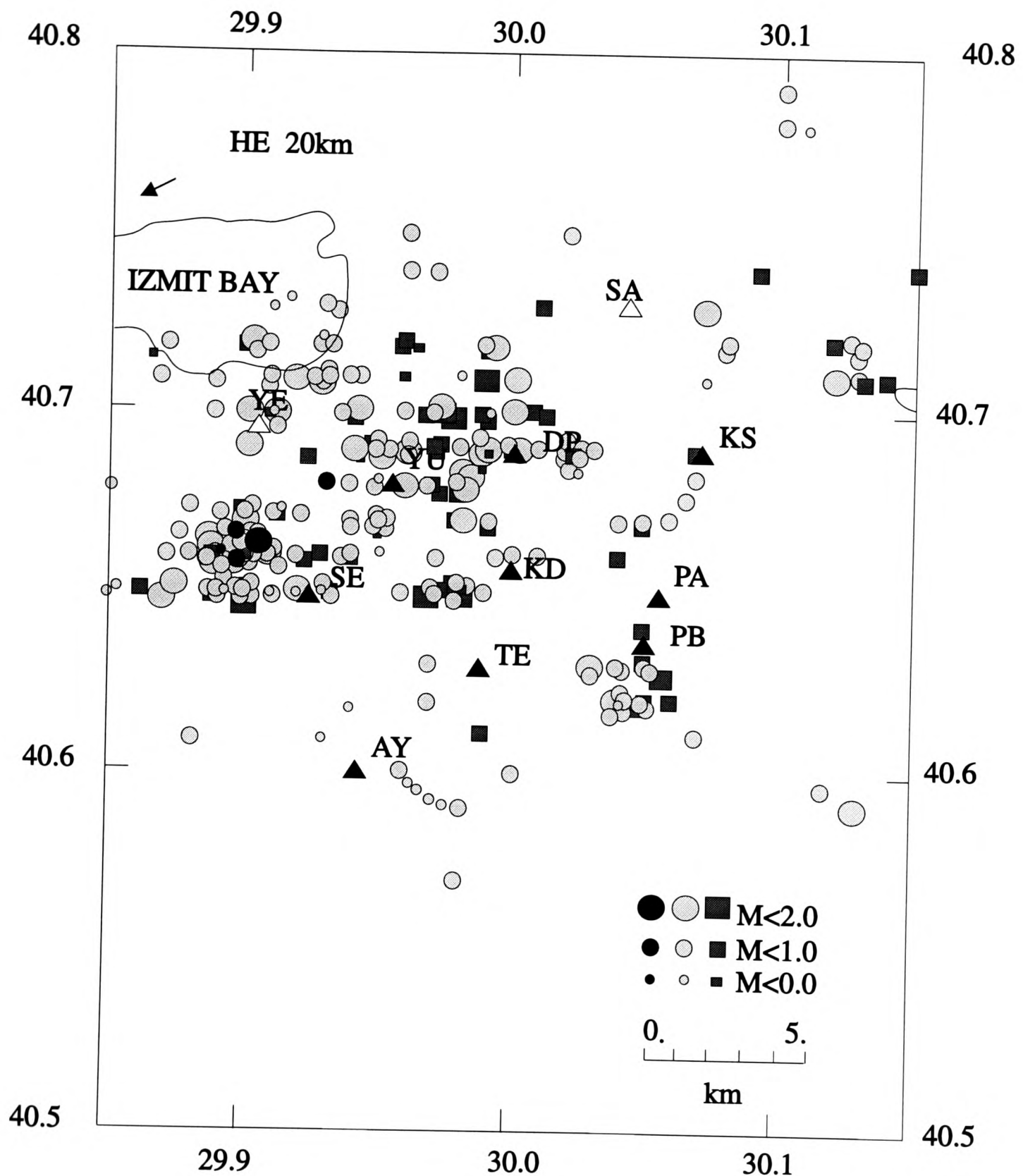


Figure 6.10(a). The location map of events which recorded in station AY. Each symbol represents an event whose P-arrival was picked for identification. Grey circles represent P-arrivals which are correctly identified by the BPNN, and grey squares represent P-arrivals which are wrongly identified by the BPNN. Black circles represent the training P-arrivals.

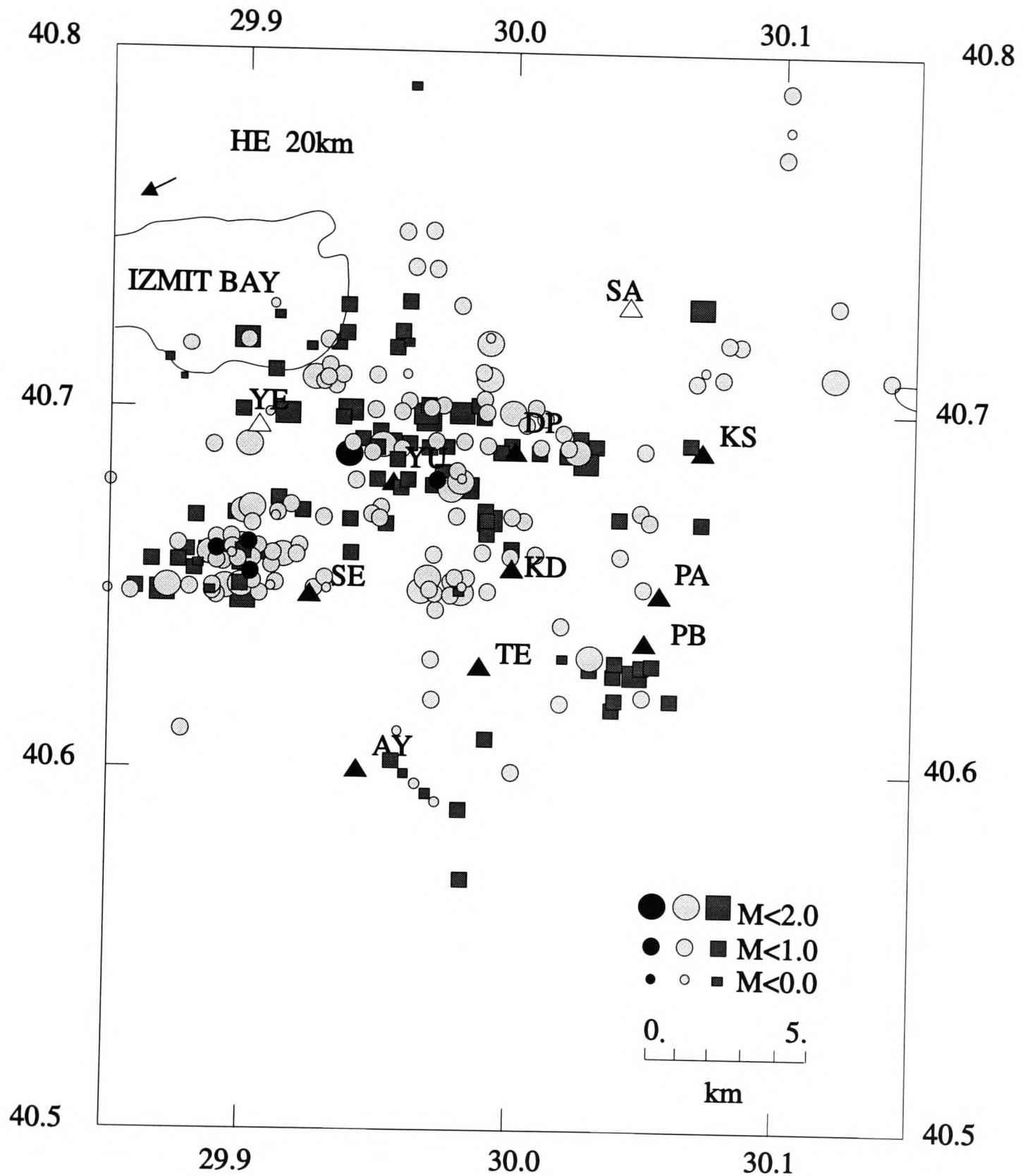


Figure 6.10(b). The location map of events which recorded in station AY. Each symbol represents an event whose S-arrival was picked for identification. Grey circles represent S-arrivals which are correctly identified by the BPNN, and grey squares represent S-arrivals which are wrongly identified by the BPNN. Black circles represent the training S-arrivals.

dataset from Station DP. Table 6.4 (a) shows a comparison of three BPNNs trained with different datasets. As the number of segments in the datasets increases, the performance for identifying *P*-arrivals improves, but the performance for identifying *S*-arrivals and noise bursts becomes worse. This is due to the complexity of the  $MF(t)$  patterns.  $MF(t)$  patterns of *P*-arrivals are more typically alike, but  $MF(t)$  patterns of *S*-arrivals are often quite different. In addition, some *P*-arrivals, *S*-arrivals and noise bursts have similar  $MF(t)$  patterns. If such a *P*-arrival pattern is used to train the BPNN, the trained BPNN will classify all of them as *P*-arrivals irrespective of what they actually are. In theory, the BPNN should classify these arrivals according to the linearity of their polarization as defined in the training datasets. The *P*-arrival, *S*-arrivals and noise bursts have high, middle and low linearity of polarization respectively. It seems that in practice the  $MF(t)$  alone is not enough to distinguish these arrivals. Other properties of the seismic arrivals such as the direction of polarization and frequency might be needed. But this is outside the scope and time frame of the present work.

#### 6.4.4 Effect of input nodes

I also investigated the BPNN's sensitivity to the input segment length as this decides the BPNN structure. Various input nodes between 50 and 70 nodes were tested, retaining the same hidden nodes and output nodes. The training procedure is the same: beginning with one group of training segments and increasing to nine groups. The training segments are different for these three BPNNs due to their different performance at every training stage but are all from station DP. Table 6.4 (b) shows the testing results of three BPNNs. On balance, the BPNN with 60 input nodes has the optimum performance. This suggests that, the segments should include appropriate information of an arrival, otherwise too much or too little information will degrade the BPNN performance. This also reflects the general observation that BPNN architecture must be specifically tailored to individual applications. Further optimization is required to adapt to particular event types.

**Table 6.4 (a):** The comparison of the performance of three BPNNs with 60 input nodes trained with different training datasets from station DP. Only correct identification percentages are shown in this table. The best performance is from the BPNN trained with nine training groups.

	P-arrivals	S-arrivals	Noise
8 training groups	81%	58%	51%
9 training groups	83%	63%	47%
10 training groups	86%	59%	38%

**Table 6.4 (b):** The comparison of the performance of three BPNNs with different input nodes trained with nine groups of segments from station DP. Only correct identification percentages are shown in this table. The best performance is from the BPNN with 60 input nodes.

	P-arrivals	S-arrivals	Noise
50 input nodes	88%	45%	47%
60 input nodes	83%	63%	47%
70 input nodes	87%	44%	48%

## 6.5. DISCUSSIONS AND SUMMARY

### 6.5.1 Weight pattern analysis

In order to demonstrate the ability of the trained BPNN to identify the arrival types, a weight map for the trained BPNN is shown on Figure 6.11. This map is similar to the previous figures (Figures 4.4, 4.5, 4.6, 4.17 and 5.15) shown in Chapters 4 and 5, but this BPNN with 60 input nodes was trained by nine groups of noise burst, *P*-arrival and *S*-arrival segments from station DP. Before training, the weights and thresholds were initialized randomly between -0.5 and +0.5.

This map shows that, after training, these values range from -7.26 to +5.74, much greater than those in Figures 4.17 and 5.15. In contrast with the weight pattern in those maps, this map shows that the pattern of weights connecting the input to the hidden layer concentrates on the portion of the signal between samples 25 and 45 (240ms to 440ms), corresponding to the main energy in the training input  $MF(t)$ . The weights in this portion have larger values than the weights outside this portion. Large weights have a strong effect in the summation of the products of weights and signal inputs because all values in the input segment are positive and normalized between 1.0 and 0.0. According to the nature of the sigmoidal activation function in which its denominator has a negative exponential with the exponent being the summation, these large values produce in turn a large negative or positive exponent, thus yielding a high output or low output in the hidden nodes. The weight pattern between the hidden nodes and the output nodes is related to the pattern between the input nodes and hidden nodes. For example, the weights connected to the *noise* node and the *third*, *fifth* and *eighth* hidden nodes, and the weights connected to the *S*-node and *first*, *ninth* and *tenth* hidden nodes have large negative values. In contrast, no weight connected to the *P*-node and hidden nodes has a larger positive value. It seems the hidden nodes have different activities to indicate the different type segments.

In Figure 6.4, the main difference among three types of  $MF(t)$  is localized on the portion roughly between sample 25 and 45. The shape of an input segment in this

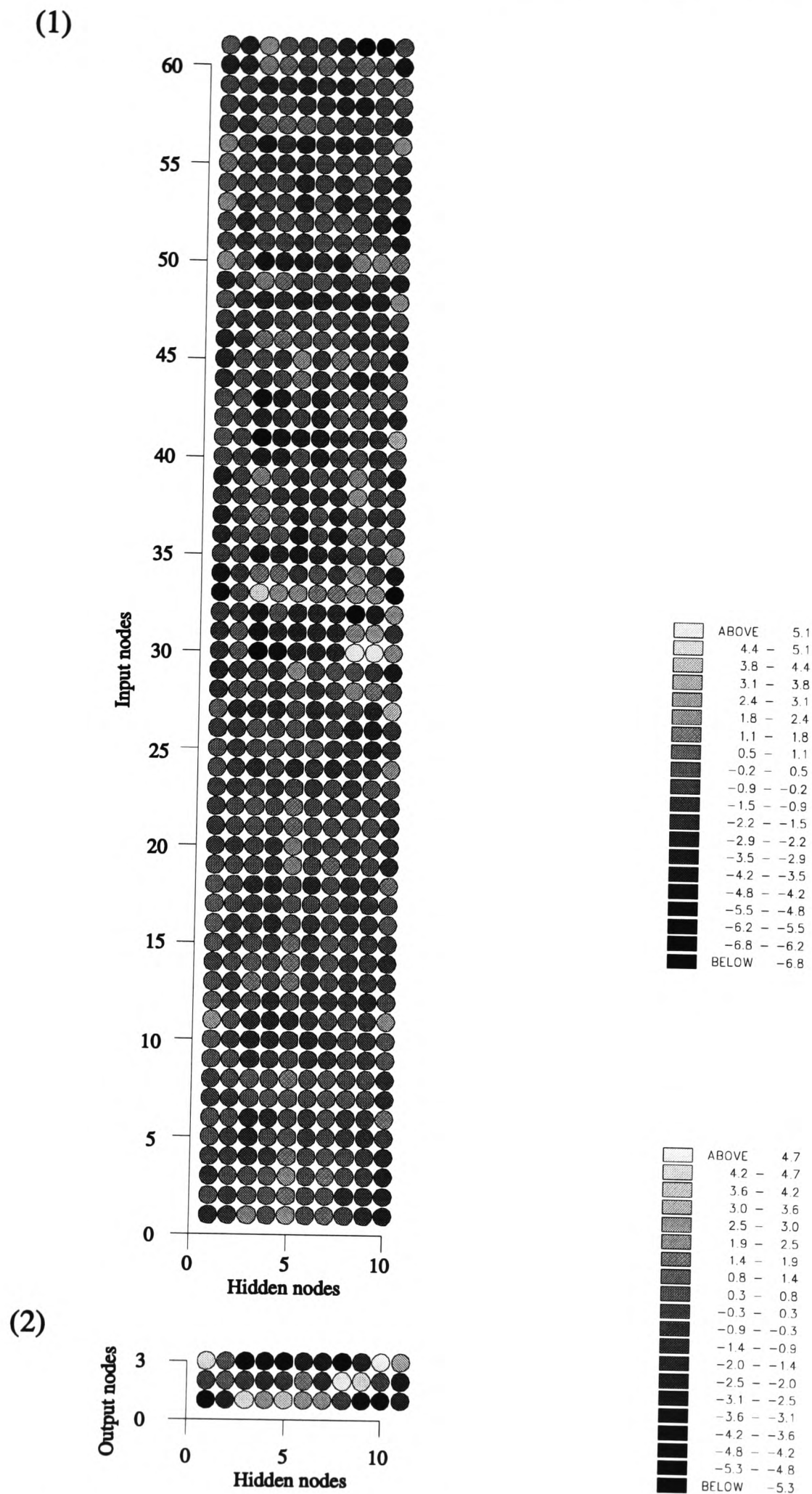


Figure 6.11. Weight map of the trained BPNN for arrival identification. Notations as in Figure 4.4(a). This BPNN is trained with nine groups of MF(t) segments of noise bursts, P-arrival and S-arrival.

portion mainly indicates its type. As an input segment is fed into the BPNN, its values in the portion will activate the hidden nodes according to the input values and the weights connected to them, and then, in turn, light the output nodes. Although it is not possible to fully understand the logic underlying the network solution by visually inspecting the weight pattern, the weight map may still lead some perspective as to which aspects of the modified DOP are more relevant to the solution than others, and thus be of benefit to further processing schemes.

### 6.5.2 Summary

In this chapter, a BPNN approach has been developed to identify *P*- and *S*-arrival types from local earthquake data using the polarization state, *the modified degree of polarization*,  $MF(t)$ , of 3-C recordings. The results show that a BPNN trained with a small subset of the data from station DP can correctly identify 82.3% of the *P*-arrivals and 62.6% of the *S*-arrivals from station DP, and another BPNN trained with data from station AY can correctly identify 76.6% of the *P*-arrivals and 60.5% of *S*-arrivals from station AY. The overall performances are 79.8% for the *P*-arrivals and 61.6% for the *S*-arrivals. This performance, combined with the advantage of not requiring programs to construct special variables and parameters with complicated mathematics, suggests that the BPNN is a natural choice for such applications. This method is adaptive, and training datasets can be altered to enhance particular features of different datasets. Adding new training datasets and retraining a BPNN is easy and quick, and can improve its performance. Although the training time in this approach is longer than that in Chapter 4 and 5, once trained the BPNN is sufficiently quick to operate in most real-time applications.

The performance of the trained BPNN, however, has inherent limitations due to the complexity of  $MF(t)$  patterns. The first limitation is that the training dataset and test data must be from the same station due to the inter-station complexity of  $MF(t)$  patterns. For example, the BPNN trained with data from station DP can only identify 48% of the *P*-arrivals and 44% of the *S*-arrivals from station AY, and another BPNN

trained with data from station AY only identify 84.9% of the *P*-arrivals and 36.1% of the *S*-arrivals from station DP. It means that the polarization information is dependent on stations. This is unlike the approach developed in Chapters 4 and 5 which can use one BPNN trained with *P*-arrivals from station DP to pick both *P*- and *S*-arrivals from both stations DP and AY. The second limitation is that the BPNN's performance cannot be improved by simply adding more training datasets due to the complexity of  $MF(t)$  patterns. It shows that selecting the input information is critical. This suggests that other information such as the direction of polarization and frequency information may be needed. The third limitation is in finding an optimum architecture for a particular application because no theory can be used to help find this optimum architecture. The BPNN's performance depends upon the training set, and its ability to predict cannot lie too far outside its experience. Although the weight map can give us a rough impression of how this BPNN works, the exact boundaries of this behaviour have not yet been completely explored.

In fact, these limitations are also due to the disadvantage of the supervised learning scheme being used to train the BPNN. Without training, a BPNN cannot learn new strategies for a particular situation that is not covered by the set of examples used to train the network (Haykin, 1994). However, this might be overcome by the use of an unsupervised learning scheme or other kind of neural network such as ART2 (Carpenter and Grossberg, 1987).



## CHAPTER 7

### CONCLUSIONS AND FUTURE WORK

#### 7.1 INTRODUCTION

The aim of this thesis was to investigate the possibility of using artificial neural networks (ANNs) to develop automatic processing techniques to pick and identify seismic arrivals from earthquake data. To reach this target, I wrote my own back-propagation neural network (BPNN) system, and, based on it, developed two BPNN approaches to pick and identify seismic arrivals for data including 762 three component recordings from stations DP and AY of the local earthquake network of the TDP3 project. The BPNN's performances and the effect factors are examined in the previous chapters. In this chapter, I shall summarize the main results and then give some suggestions for further research.

#### 7.2 MAIN RESULTS OF THIS THESIS

##### 7.2.1 Arrival picking

A BPNN approach was developed to pick *P*- and *S*-arrivals automatically from three-component (3-C) recordings in Chapter 4 and single-component (1-C) recordings in Chapter 5. This approach employs a BPNN trained by a small but representative training dataset which acts as a filter on the entire seismograms using a sliding window along the entire input characteristic trace which is the vector modulus of the 3-C recordings or the absolute value of the 1-C recordings. The input of the BPNN is a segment of the input characteristic which is individually normalized. The output of this trained BPNN yields a time series which is to be interrogated for a decision regarding the seismic arrivals. The significant feature of this output time series is that some peaks emerge from a smooth background. These peaks correspond to changes

in the input characteristics, with values implying the levels of changes. From the transformation point of view, the trained BPNN transforms the rates of changes of the input characteristics into the value of this time series from which the changes are enhanced and can then be more easily detected. If a peak value exceeds a given threshold, an arrival is detected and its maximum indicates the onset time. In order to improve its performance, some simple post-processing procedures are introduced to discard some false alarms.

In Chapter 4, this approach is used to deal with 3-C recordings. The input characteristic is a segment of the *relative vector modulus* of these 3-C recordings. A BPNN trained by only nine pairs of *P*-arrivals and background noise segments from station DP can detect and pick not only *P*-arrivals but also *S*-arrivals. This ability is extended to picking arrivals from station AY as well. This approach can detect 94.3% of the *P*-arrivals and 86.4% of the *S*-arrivals with the detection threshold at 0.6, compared with manual picks as references. Using this to further pick the onset times, the success rate is 74.5% for the *P*-arrivals and 63.2% for the *S*-arrivals with an error of 10ms (one sample increment). Even though all the training datasets have high signal-to-noise-ratios (SNRs), this BPNN still works for seismograms with low SNRs.

In Chapter 5, this method is adapted to pick arrivals from 1-C seismograms. The input characteristic is a segment of the *relative absolute value* of 1-C recordings. Here the BPNN trained by ten pairs of *P*-arrivals and background noise segments from vertical components (V-C) of station DP can successfully detect and pick seismic arrivals not only from the vertical components but also from the other two horizontal components (E-W and N-S) and other stations. Its performance is different for each of three 1-C recordings due to strong effects of raypath and source position. The effect on *P*-arrivals is smaller than on *S*-arrivals. The picking success rates for *P*-arrivals are 93.1%, 89.4%, and 83.1% from the V-C, E-W, and N-S recordings respectively, and for *S*-arrivals are 75.0%, 90.9%, and 87.2% from the V-C, E-W, and N-S recordings respectively. The accuracy of the onset times picked from each individual 1-C recording is similar. With the error  $\leq 10\text{ms}$  (one sample increment), 66.2%, 59.2% and

63.3% of the *P*-arrivals and 52.7%, 61.2%, and 57.7% of the *S*-arrivals are picked from the V-C, E-W and N-S recordings respectively. In comparison to the 3-C picking, the performance of 1-C picking is thus understandably lower with the associated disadvantage of a large BPNN structure. However, it has an obvious advantage of flexibility over 3-C picking and can be used on 1-C recordings when 3-C recordings are not available.

### 7.2.2 Arrival identification

In Chapter 6, another BPNN approach is developed to identify *P*- and *S*-arrival types from local earthquake data, using the polarization state, *the modified degree of polarization* (DOP), of 3-C recordings. As a procedure applied after arrival picking, this approach has a different logic behind it. As seismic arrivals including some noise bursts have been picked by the BPNN approach developed in Chapter 4, only the arrival segments are input in this BPNN approach for processing. This greatly reduces the processing time compared to processing the whole seismic recording. This approach is also used to deal with the same data used in Chapter 4.

Due to the inter-station complexity of DOP, the training dataset and test data must be from the same station. Compared with manual analysis, a BPNN trained with nine groups of training *P*-arrival, *S*-arrival, and noise burst segments of DOP from station DP can correctly identify 82.3% of the *P*-arrivals and 62.6% of the *S*-arrivals from station DP, and another BPNN trained with five groups of training dataset from station AY can correctly identify 76.6% of the *P*-arrivals and 60.5% of *S*-arrivals from station AY. The overall performances are 79.8% for the *P*-arrivals and 61.6% for the *S*-arrivals. The performance cannot be improved by simply adding more training datasets due to the complexity of DOP patterns. This suggests that using the degree of polarization only is not enough to resolve this problem, and other information such as the direction of polarization and frequency information may be necessary.

### 7.2.3 Weight pattern analysis

In order to understand how a BPNN works, a *weight map* is designed in this thesis to show the weight patterns of a trained BPNN. Applying the weight map to three trained BPNNs shows that it is a useful tool to investigate the interior and performance of BPNNs. For the two BPNNs which are used to pick arrivals, although their structures and training datasets are different, they have similar weight patterns both of which are divided into two portions corresponding to the onset of training *P*-arrivals and having different functions in the picking work. For the BPNN which is used to identify arrival types, its weight pattern has a special portion which is linked to the main energy of segments in the training dataset. These weight patterns show that hidden nodes have different individual activities according to the weight connected to them. The hidden nodes with high activity can recognize the main features of an input segment and other hidden nodes with low activity recognize the subtle features of the input signal, improving the BPNN performance. It seems that if such low active nodes are reduced or damaged, the BPNN may still recognize the input segment, However, the BPNN's performance may be decreased.

Although it is not possible to fully understand the logic underlying the BPNN solutions by visually inspecting the weight patterns, it may still provide some perspective as to which aspects of the input characteristics are more relevant to the solution, and thus be of benefit to refine the methods or apply it to other processing techniques.

### 7.2.4 Factors affecting the BPNN performance

#### (a) Training parameters.

Results from this thesis show that the training parameters strongly affect the training convergence. A relationship between the training parameters (learning rate  $\eta$  and momentum rate  $\alpha$ ) and the training convergence (iteration number) is obtained (Equation 4.1). Testing on three BPNNs shows that this equation is better fitted when  $\eta$  and  $\alpha$  are small than when  $\eta$  and  $\alpha$  are larger. The fitting is varied with different

BPNNs. It seems that the fitting is better for a small BPNN which is applied to some simple problems. The deviation of this relationship is for two causes: the oscillation in finding a desired weight state during training, and the different system error at the termination of the training phase. Although different values of the training parameters may be used in the training procedure, the weight pattern of a trained BPNN and its performance is not affected too much. It may be important if a large BPNN structure and a large training dataset are involved. This will allow us to devote our attention only to the performance of a trained BPNN, and can make a complex problem simple.

### **(b) BPNN structure**

The structure of a BPNN is decided by the number of its layers and the numbers of nodes in each layer. The results in this thesis show that a three-layer BPNN is sufficient for the applications of picking and identifying seismic arrivals. The number of input nodes of a BPNN is important in these approaches. Each approach has his own optimum number of input nodes. This number corresponds to the segment of input characteristic which should include suitable information for its purpose. Too much or too little information will make its performance worse. The number of hidden nodes has a small effect to its performance although too few or too many hidden nodes make the BPNN performance worse. In this thesis, ten hidden nodes in all three BPNNs are selected by a process of trial and error.

### **(c) Training dataset**

The most important factor in these approaches is the training dataset. The performance of a trained BPNN is highly dependent on the training dataset. A similar BPNN can be used for different purposes with the different training datasets. As our human experiences and prejudices are embodied in the selection of a training dataset, the BPNN actually acts according to the experiences and may fail to handle the data which are different from the training dataset. Selecting a suitable and correct training dataset is crucial.

### 7.2.5. Summary

Compared with other conventional methods of picking and identifying seismic arrivals, there are some distinct advantages of using BPNNs. One of the significant advantages is its adaptiveness. In this thesis, I applied the same programs to the problems of picking arrivals from 3-C recordings and 1-C recordings, and identifying the arrival types with only changes of the BPNN structure and the input/output. Once the generic routines of BPNN are developed, it is easy to apply them to resolve a particular problem without requiring additional programs to construct special variables and parameters with complicated mathematics. The knowledge of resolving a problem is minimized so that it is only necessary to select suitable training examples for the new application. The input data may be the raw data which is the most objective, but may be handled with difficulty by conventional methods. The BPNN's performance can be easily improved by adding or adjusting the training dataset. In addition, a trained BPNN has an objective and consistent performance.

However, there are also some limitations of using BPNNs. The performance of a trained BPNN depends strongly upon the training set and its ability to predict cannot lie too far outside its experience. Although the weight map can give us a rough idea of how a BPNN works, the exact boundaries of its behaviour have not yet been completely explored and not yet fully understood. Another limitation is in finding an optimum BPNN architecture for a particular application. At present, no theory can be used to help find this optimum architecture. Although such an optimum architecture exists for a particular application, finding it is dependent on the operator's experience and knowledge of the problem, and sometimes it is difficult to achieve.

Although potentially useful, it is important to emphasize that the ANN is not a panacea. As a general rule, it should not be used when an explicit algorithm already exists or can be simply developed for solving a particular problem. The ideal application of ANNs is that the relationship between two sets of data is believed to exist but its nature is unknown. In order to obtain a better performance, ANNs must be considered as one part of an intelligent system, in combination with other methods. For example, the approach used in Chapters 4 and 5 employed some simple methods

to discard small noise bursts and spikes which would otherwise have been incorrectly picked by the BPNN. These simple methods improved the performance of the approach.

### **7.3 FUTURE WORK**

So far, the result in this thesis shows that the BPNN is a natural choice for applications in many problems in seismology. In particular, it has great potential as an automatic tool to analyze earthquake data, especially for the problems of picking and identifying seismic arrivals. This is important because of the exponential increase in the quantity of digital seismic data becoming available. This automatic processing is faster, more consistent and less labour-consuming than manual processing. However, before it can be put into practice in building a fully automatic seismic analysis system (ASAS), there is still some work ahead.

#### **7.3.1 Dealing with regional and teleseismic earthquake data**

In this thesis, I focus the BPNN approach on dealing with the local earthquake data. However, the adaptive nature of the BPNN led us to expect similar results when applying the same method to processing regional or teleseismic observations. For the regional and teleseismic events, their recordings may have features, such as dominating frequency, bandwidth, sample rate, and arrivals types, different from the local events. Their arrivals may be more complex and composed of several distinct signal processes (Der, Baumgardt and Shumway, 1993). As a consequence, the BPNN structure, input characteristics and training dataset may need to be altered somewhat to suit them. However, this could be achieved in principle by relatively simple modification to the methods presented here.

#### **7.3.2 Alternative approach of identifying arrival type**

In Chapter 6, the approach was used to identify seismic arrivals from 3-C recordings. However, if 3-C recordings are unavailable, as is often the case, a method

is necessary to identify the seismic arrivals from 1-C recordings. The limitations of this approach highlighted in Chapter 6 shows that other information which might be necessary to tackle this problem. For example, in manual identification, the analyst does not only look at the specific segment of an arrival, but also looks at the whole seismograms to make his decision. Usually, the information from the whole seismograms has a bigger weight in making such a decision. It might be possible to follow this method to improve the performance of the BPNN approach described in Chapter 6 using other information such as amplitude, spectrum, and arrival-onset-time series.

### 7.3.3 Applying ANNs to other earthquake analysis procedures

The whole procedure of earthquake analysis (see Chapter 1) includes many steps of which the picking and identifying of arrivals are the key steps. Consequently, in future schemes, it is ultimately hoped to integrate ANN units into other procedures which involve intensive pattern recognition, including filtering and editing poor recordings (before arrival picking) and classifying the event type (after arrivals identification). In future, it may be possible to apply the method to other applications such as focal mechanism determination, shear-wave splitting analysis, or seismic velocity inversion.

### 7.3.4 Understanding more about ANNs

Although the BPNNs used in this thesis have been successfully applied to the problems of picking and identification of arrivals, and their performances and effect factors have been examined, it is still difficult to fully understand or inspect their interior which hides in a *black-box*. In addition, there are no well defined rules to find an optimum architecture and develop a specialized structure by building prior information into its design. The BPNN used in this thesis has some limitations on its performance due to its supervised learning scheme. In order to overcome this difficulty, other kind of ANNs and learning schemes may be necessary in future.



## **7.4 CONCLUSIONS**

The work in this thesis shows encouraging results obtained by applying ANNs to earthquake analysis. ANNs have great potential in the development of a fully automatic system for earthquake analysis. This can be achieved by embedding ANN units into an existing earthquake analysis system to replace their equivalents which are difficult to be automated by conventional methods. Such an automatic system will be more reliable, robust, objective and less time-consuming than other conventional systems and will be of benefit to the development of seismology.

## APPENDIX A: GENERALIZED DELTA RULE

The *Generalized Delta Rule* is used to train the back-propagation neural network (BPNN). It was first introduced by Rumelhart, Hinton and Willams (1986), later Pao (1988) gave a good mathematical summary. This appendix is based on Pao's description, but I have changed some notations for convenience in computer programming.

To train a BPNN by using the *Delta Rule*, the sample patterns  $\mathbf{X}=\{\mathbf{x}_j\}$  are presented as input and the BPNN is asked to adjust the set of weights in all the connecting links and also thresholds in the nodes so that the desired outputs  $\mathbf{T}=\{\mathbf{t}_k\}$  are obtained at the output nodes. In general, the output  $\{\mathbf{o}_k\}$  is not the same as the desired values  $\{\mathbf{t}_k\}$ . For each pattern, the square of the error is:

$$E = \frac{1}{2} \sum_{k=0}^{K-1} (t_k - o_k)^2, \quad (\text{A.1})$$

where the factor of  $\frac{1}{2}$  is inserted for mathematical convenience.

In the learning procedure, the weights and thresholds are varied in a manner calculated to reduce the error  $E$  by as much as possible. The error at each node propagates backward from the output layer to the input layer with changes of weights and thresholds. The changes of the weights and thresholds are achieved by taking

incremental change  $\Delta w_{ijk}$  proportional to  $-\frac{\partial E}{\partial w_{ijk}}$ , that is:

$$\Delta w_{ijk} = -\eta \frac{\partial E}{\partial w_{ijk}}. \quad (\text{A.2})$$

Here,  $\eta$  is a proportional constant. The change of threshold  $q_{i+1\ k}$  is expressed by this as  $\Delta w_{ijk}$  (see Section 2.3.1.1). However,  $E$  is expressed in terms of the output  $o_{i+1k}$  each of which is the non-linear output of the node  $n_{i+1k}$  in the  $i+1$ th layer:

$$o_{i+1k} = f(\text{net}_{i+1k}) \quad , \quad (\text{A.3})$$

where  $\text{net}_{i+1k}$ , the weighted linear sum of the outputs from the previous layer, is the input to the  $k$ th node in the  $i+1$ th layer:

$$\text{net}_{i+1k} = \sum_{j=0}^J w_{ijk} o_{ij} \quad . \quad (\text{A.4})$$

The partial derivative  $-\frac{\partial E}{\partial w_{ijk}}$  can be evaluated using the chain rule:

$$\frac{\partial E}{\partial w_{ijk}} = \frac{\partial E}{\partial \text{net}_{i+1k}} \frac{\partial \text{net}_{i+1k}}{\partial w_{ijk}} \quad . \quad (\text{A.5})$$

Replacing  $\text{net}_{i+1\ k}$  with  $\sum_{j=0}^J w_{ijk} o_{ij}$ , we obtain:

$$\frac{\partial \text{net}_{i+1k}}{\partial w_{ijk}} = \frac{\partial}{\partial w_{ijk}} \left( \sum_{j=0}^J w_{ijk} o_{ij} \right) = o_{ij} \quad . \quad (\text{A.6})$$

$\delta_{ik}$  can be defined as:

$$\delta_{ik} = -\frac{\partial E}{\partial \text{net}_{i+1k}} \quad . \quad (\text{A.7})$$

So  $\Delta w_{ijk}$  is written as:

$$\Delta w_{ijk} = \eta \delta_{ik} o_{ij} \quad . \quad (\text{A.8})$$

To compute  $\delta_{ik}$ , the chain rule is used to express the partial derivative in terms of two factors. One factor expresses the rate of change of error with respect to the output  $o_k$ , and the other one expresses the rate of change of the output of the node  $k$  with respect to the input to the same node. That is:

$$\delta_{ik} = -\frac{\partial E}{\partial net_{i+1k}} = \frac{\partial E}{\partial o_{i+1k}} \frac{\partial o_{i+1k}}{\partial net_{i+1k}} . \quad (A.9)$$

If the  $i+1$ th layer is the output layer, the two factors are obtained as follows:

$$\frac{\partial E}{\partial o_{i+1k}} = -(t_k - o_{i+1k}) , \quad (A.10)$$

and

$$\frac{\partial o_{i+1k}}{\partial net_{i+1k}} = f'(net_{i+1k}) . \quad (A.11)$$

Because

$$o_{i+1k} = f(net_{i+1k}) = \frac{1}{1 + \exp^{-net_{i+1k}}} , \quad (A.12)$$

so that:

$$\frac{\partial o_{i+1k}}{\partial net_{i+1k}} = f'(net_{i+1k}) = o_{i+1k}(1 - o_{i+1k}) . \quad (A.13)$$

From this,  $\delta_{ik}$  is obtained:

$$\delta_{ik} = (t_k - o_{i+1k})f'(net_{i+1k}) = (t_k - o_{i+1k})o_{i+1k}(1 - o_{i+1k}) . \quad (A.15)$$

If the  $i+1$ th layer is a hidden layer, we also have:

$$\frac{\partial o_{i+1k}}{\partial net_{i+1k}} = f'(net_{i+1k}) = o_{i+1k}(1 - o_{i+1k}) . \quad (A.16)$$

But the factor  $\frac{\partial E}{\partial o_{i+1k}}$  can not be evaluated directly. Instead, it can be written in terms

of quantities that are known and other quantities that can be evaluated. Specifically, it is written as:

$$\frac{\partial E}{\partial o_{i+1k}} = -\sum_{j=0}^J \frac{\partial E}{\partial net_{i+2j}} \frac{\partial net_{i+2j}}{\partial o_{i+1k}}$$

$$\begin{aligned}
 &= \sum_{j=0}^J \left( -\frac{\partial E}{\partial \text{net}_{i+2j}} \right) \frac{\partial}{\partial o_{i+1k}} \left( \sum_{m=0}^M w_{i+1jm} o_{i+1m} \right) \\
 &= \sum_{j=0}^J (\delta_{i+1j} w_{i+1jk}) \quad .
 \end{aligned} \tag{A.17}$$

From these emulation for the hidden layer,  $\delta_{ik}$  is obtained:

$$\delta_{ik} = o_{i+1k} (1 - o_{i+1k}) \sum_{j=0}^J (\delta_{i+1j} w_{i+1jk}) \quad . \tag{A.18}$$

So that the *Delta Rule* can be written as :

$$\Delta w_{ijk} = \eta \delta_{ik} o_{ij} = \eta o_{i+1k} (1 - o_{i+1k}) (t_k - o_{i+1k}) o_{ij} \tag{A.19}$$

for the output layer, and

$$\Delta w_{ijk} = \eta \delta_{ik} o_{ij} = \eta o_{i+1k} (1 - o_{i+1k}) \sum_{j=0}^J (\delta_{i+1j} w_{i+1jk}) o_{ij} \tag{A.20}$$

for the hidden layer.

Equations (A.19) and (A.20) show that the change of  $w_{ijk}$  is proportional to the output  $o_{ij}$  of node  $n_{ij}$  and  $\delta_{ik}$  of node  $n_{i+1k}$ . For an output node, Equation (A.15) shows that  $\delta_{ik}$  is calculated from its output and its error, the difference between its network output and its desired output. For a hidden node, Equation (A.18) shows that  $\delta_{ik}$  is also calculated from its output and its error, but the error is more complex and is defined as the sum of weighted errors of output nodes. This implies the errors of output nodes propagate backward from the output layer to the hidden layer for changing weights and thresholds.

## REFERENCES

- Aki, K., and Richards, P. G., 1980, *Quantitative Seismology*, W. H. Freeman and Co., New York.
- Allen, R. V., 1978, Automatic earthquake recognition and timing from single traces, *Bulletin of the Seismological Society of America*, **68**, 1521-1532.
- Allen, R., V., 1982, Automatic phase pickers: their present use and future prospects, *Bulletin of the Seismological Society of America*, **72**, S225-S242.
- Ambuter, B. P., and Solomon, S. C., 1974, An event-recording system for monitoring small earthquakes, *Bulletin of the Seismological Society of America*, **64**, 1181-1188.
- Anderson, K. R., 1978, *Automatic processing of local earthquake data*, PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA,
- Bache, T., Bratt, S. R., Wang, J., Fung, R. M., Kobryn, C., and Given, J. W., 1990, The intelligent monitoring system, *Bulletin of the Seismological Society of America*, **80**, 1833-1851
- Baer, M., and Kradolfer, U., 1987, An automatic phase picker for local and teleseismic events, *Bulletin of the Seismological Society of America*, **77**, 1437-1445.
- Basham, P. W., and Ellis, R. M., 1969, The composition of P-code using magnetic tape seismograms, *Bulletin of the Seismological Society of America*, **59**, 473-486.
- Baum, E. B., and Haussler, D., 1989, What size net give valid generalization, *Neural Computation*, **1**, 151-160.
- Bibbo, J., Etter, D., and Breiding, D., 1991, A software tool for processing seismic data, *Computers and Geosciences*, **17**, 301-305.
- Blandford, R. R., 1982, Seismic event discrimination, *Bulletin of the Seismological Society of America*, **72**, S69-S87.

- Cai, Y. D., Gong, J. W., Gan, J. R. and Yao, L. S., 1993, A neural network approach for classification and recognition of loess porosity, *Scientia Geologica Sinica*, **2**, 473-478.
- Carpenter, G., and Grossberg, S., 1987, ART2: self-organization of stable category recognition codes for analog input patterns, *Applied Optics*, **26**, 4919-4930.
- Cary, P. and Upham, W., 1992, An evaluation of neural networks, *Geophysics: the Leading Edge of Exploration*, 45-47, September 1992.
- Chiaruttini, C., Roberto, V., and Saitta, F., 1989, Artificial intelligence techniques in seismic signal interpretation, *Geophysical Journal International*, **98**, 223-232.
- Chiaruttini, C., and Salemi, G., 1993, Artificial intelligence techniques in the analysis of digital seismograms, *Computers and Geosciences*, **19**, 149-156.
- Cichowicz, A., 1993, An automatic S-phase picker, *Bulletin of the Seismological Society of America*, **83**, 180-189.
- Cichowicz, A., Green, R. W., and Brink, A. van Z., 1988, Coda polarization properties of high-frequency microseismic events, *Bulletin of the Seismological Society of America*, **78**, 1297-1318.
- Crampin, S., and Fyfe, C. J., 1974, Automatic analysis of tape-recordings from seismic networks, *Geophysical Journal*, **39**, 155-168.
- Crampin, S., Evans, R., and Üçer, S. B., 1985, Analysis of records of local earthquakes: the Turkish Dilatancy Projects (TDP1 and TDP2), *Geophysical Journal of the Royal Astronomical Society*, **83**, 1-16.
- Dai, H. C., and MacBeth, C., 1993, Analysis of split shear-waves from near-offset VSP data using a neural network, *Extended Abstract in 55th meeting of European Association of Exploration Geophysicists, Stavanger, Norway, 7-11, June 1993*.
- Dai, H. C., and MacBeth, C., 1994a, A neural network picker for VSP data, *Extended Abstract in 56th meeting of European Association of Exploration Geophysicists, Vienna, Austria, 6-10, June 1994*.

- Dai, H. C., and MacBeth, C., 1994b, Automatic picking seismic arrivals of single component recordings from local earthquake data using artificial neural networks. *British Geological Survey internal report: WL/94/27*.
- Dai, H. C., and MacBeth, C., 1994c, Split shear-wave analysis using an artificial neural network, *First Break*, **12**, No. 12, 605-613.
- Dai, H. C., and MacBeth, C., 1995a, Automatic picking of seismic arrivals in local earthquake data using an artificial neural network, *Geophysical Journal International*, **120**, 758-774
- Dai, H. C., and MacBeth, C., 1995b, Arrival type identification in local earthquake data using artificial neural network, *Proceedings of the 2nd Workshop on Application of Artificial Intelligence Techniques in Seismology and Engineering Seismology*", Walferdange, Luxembourg, 4-6 October 1995, *in press*.
- Demuth, H. and Beale, M., 1993, *Neural Network Toolbox for Use with MATLAB: User's Guide*, The Math Works, Inc, Natick, Massachusetts.
- Der, Z. A., Baumgardt, D. R., and Shumway, R. H., 1993, The nature of particle motion in regional seismograms and its utilization for phase identification, *Geophysical Journal International*, **115**, 1012-1024.
- Dowla, F., Taylor, S. R., and Anderson, R. W., 1990, Seismic discrimination with artificial neural networks: preliminary results with regional spectral data, *Bulletin of the Seismological Society of America*, **80**, 1346-1373.
- Dystart, P. S., and Pulli, J., 1990, Regional seismic event classification at the NORESS array: seismological measurements and the use of trained neural networks, *Bulletin of the Seismological Society of America*, **80**, 1910-1933
- Earle, P. S., and Shearer, P. M., 1994, Characterization of global seismograms using an automatic-picking algorithm, *Bulletin of the Seismological Society of America*, **84**, 366-376.
- Evans, J. R., 1980, *ADC: a program for digitising seismic records into a format suitable for automatic processing*, British Geological Survey, Global Seismology Report No. 136.



- Evans, J. R. and Miller, A., 1986, *An interface unit for digitisation of geostore tapes*, British Geological Survey, Global Seismology Report No. 278
- Fausett, L., 1994, *Fundamentals of Neural Networks*, Prentice Hall, Englewood Cliffs, New Jersey.
- Fahlman, S. E. and Lebiere, C., 1990. The cascade-correlation learning architecture, in *Advances in Neural Information Processing System 2*, ed. Touretzky, D. S., Morgan Kauffmann.
- Gorman, R. P., and Sejnowski, T. J., 1988, Analysis of Hidden Units in a layered network trained to classify sonar targets, *Neural networks*, **1**, 75-89.
- Haykin, S., 1994, *Neural Networks, a comprehensive Foundation*, Macmillan College Publishing Company, New York.
- Higgins, W. and Hsu, C. M., 1994, Edge detection using two-dimensional local structure information, *Pattern Recognition*, **27**, 277-294.
- Houliston, D. J., Waugh, G., and Laughlin, J., 1984, Automatic real-time event detection for seismic network, *Computers and Geosciences*, **10**, 431-436.
- Hsu R. C. and Alexander S. S., 1994, Seismic signal recognition using layered perceptron and data compression neural networks, JSANN93 Dec 22-22 1994.
- Joswig, M., 1990, Pattern recognition for earthquake detection, *Bulletin of the Seismological Society of America*, **80**, 170-186.
- Joswig, M., 1993a, Artificial intelligence techniques applied to seismic signal analysis, *Proceedings of the Workshop: Dynamical Systems and Artificial Intelligence Applied to Data Banks in Geophysics, 15-17 November, 1993, Walferdange (Grand Duchy of Luxembourg, Cahiers du Centre Europeen de Geodynamique et de Seismologie*, **9**, 5-15
- Joswig, M., 1993b, Automated seismogram analysis for the tripartite BUG array: an introduction, *Computers and Geosciences*, **19**, 203-306.
- Joswig, M., 1995, Automated classification of local earthquake data in the BUG small array, *Geophysical Journal International*, **120**, 262-286.

- Joswig, M., and Schulte-Theis, H., 1993, Master-event correlations of weak local earthquakes by dynamic waveform matching, *Geophysical Journal International*, **113**, 562-574.
- Kanasewich, E. R., 1981, *Time Sequence Analysis in Geophysics*, University of Alberta Press, Edmonton, Alberta, Canada
- Klumpen, E., and Joswig, M., 1993, Automated reevaluation of local earthquake data by application of generic polarization patterns for P- and S-onsets, *Computers and Geosciences*, **19**, 223-231.
- Kohonen, T., 1988, Introduction to neural computing, *Neural networks*, **1**, 3-16.
- Kracke, D., 1993, A three-component event detector based on waveform analysis, *Computers and Geosciences*, **19**, 117-122.
- Kusuma, T., and Brown, M., 1992, Cascade-correlation learning architecture for first-break picking and automated trace editing, in *Processing of SEG 62nd Annual International Meeting*, pp10-13, Society of Exploration Geophysics.
- Kvaerna, T., and Ringdal, F., 1992, Integrated array and three-component processing using a seismic microarray, *Bulletin of the Seismological Society of America*, **82**, 870-882.
- Lacoss, R. T., 1972, *Variation of false alarm rate at NORSAR, Seismic Discrimination Semi-annual Technical Summary*, Lincoln Laboratory, MIT, Cambridge, Massachusetts, June 1972.
- Langer, H., Nunnari, G., and Occhipinti, L., 1993, Estimation of seismic waveform governing parameters with neural network, *Proceedings of the Workshop: Dynamical Systems and Artificial Intelligence Applied to Data Banks in Geophysics, 15-17 November, 1993, Walferdange (Grand Duchy of Luxembourg, Cahiers du Centre Europeen de Geodynamique et de Seismologie*, **9**, 5-15)
- Leach, R., Dowla, F. U., and Vergino, E., 1993, Yield estimation using bandpass-filtered seismograms: preliminary results using neural networks with  $m_b$  ( $P_n$ ),

- Short-time, long-time, and code energy measurements, *Bulletin of the Seismological Society of America*, **83**, 488-508.
- Leggett, M., Sandham, W. A., and Durrani, T. S., 1993, Seismic event classification using a self organising Kohonen network, *Proc. IEE/IEEE Int. Workshop on natural Algorithms in Signal Processing*, 14-16, November 1993, Chelmsford, U.K.
- Lee, W. and Lahr, L., 1975, *HYPO71 (revised): a computer program for determining hypocentre, magnitude and first motion pattern of local earthquakes*, US Geological Survey, Open file Report, 75-311.
- Lomax, A. J., and Michelini, A., 1988, The use of spherical coordinates in the interpretation of seismograms, *Geophysical Journal International*, **93**, 405-412.
- Lovell, J. H., 1989, *Source parameters of a microearthquake swarm in Turkey*, thesis for the degree of Master of Philosophy, University of Edinburgh.
- Lovell, J. H., Crampin, S., Evans, R., and Ucer, S. B., 1987, Microearthquake in the TDP swarm, Turkey: clustering in space and time, *Geophysical Journal of the Royal Astronomical Society*, **91**, 313-330.
- MacLeod, A. 1992, Solving hard problems with neural nets, *New Electronics*, 17-19, September, 1992
- Magotra, N., Ahmed, N., and Chael, E., 1987, Seismic event detection and source location using single-station (three-component) data, *Bulletin of the Seismological Society of America*, **77**, 958-971.
- Maren, A. J., 1990a, *Introduction to neural networks, Chapter One in Handbook of Neural computing application (edited by Maren, A. J., Harston, C. T., and Pap, R. M.)*, Academic Press, Inc, San Diego
- Maren, A. J., 1990b, *Neural network structures: form follows function, Chapter Four in Handbook of Neural computing application (edited by Maren, A. J., Harston, C. T., and Pap, R. M.)*, Academic Press, Inc, San Diego
- Maren, A., J., Harston, C. T., and Pap, R. M., 1990, *Handbook of Neural computing application*, Academic Press, Inc, San Diego.

- Matsumura, S., Hamada, K., Katsuyama, Y., Ishida, M., and Ohkubo, T., 1981, Data processing of the Kanto-Tokai Observational Network for Microearthquakes and Ground Tilt, *Proceedings of the Second Joint Meeting of the US-Japan Conference on Natural Resources (UJNR) Panel Open Earthquake Prediction Technology, US Geological Survey Open-File Report 82-180*.
- McClelland, J., and Rumelhart, D., 1988, *Explorations in parallel distributed processing, a handbook of model, program, and Exercises*, The MIT Press, Cambridge, Massachusetts, London, England
- McCormack, M. D., 1991, Neural computing in Geophysics, *Geophysics: The Leading edge of Exploration*, **10**, 11-15, January 1991.
- McCormack, M. D., Zaucha, D. E. and Dushek, D. W., 1993, First-break refraction event picking and seismic data trace editing using neural networks, *Geophysics*, **58**, 67-78.
- McEvilly, T. V., and Majer, E. L., 1982, Automated seismic processor for micro earthquake network, *Bulletin of the Seismological Society of America*, **72**, 303-325.
- Minsky, M., and Paterts, S., 1969, *Perceptrons*, MIT Press, Cambridge, MA.
- Murat, M. and Rudman, A., 1992, Automated first arrival picking: A neural network approach, *Geophysical Prospecting*, **40**, 587-604.
- Musil, M., 1993, Interactive and semiautomatic data-analysis procedures, *Computer and Geosciences*, **19**, 141-147.
- Mykkeltveit, S., and Bungum, H., 1984, Processing of regional seismic events using data from small-aperture arrays, *Bulletin of the Seismological Society of America*, **74**, 2313-2333.
- Nava, F. A., 1992, Interactive local earthquake location on a PC, *Computer and Geosciences*, **18**, 627-664.
- Oncescu, M. C., and Rizescu, M., 1994, Conversion program package for seismological digital data format on PC, *Computers and Geosciences*, **20**, 193-196.

- Palaz, I. and Weger, R., 1990, Waveform recognition using neural networks, *Geophysics: the Leading Edge of Exploration*, 28-31, March 1990.
- Pao, Y. H., 1989, *Adaptive pattern Recognition and neural networks*, Addison-Wesley Publishing Company, Inc. New York
- Penn, B. S., Gordon, A. J. and Wendlandt, R. F., 1993, Using neural networks to locate edges and linear features in satellite images, *Computers and Geosciences*, **19**, 1545-1565.
- Pisarenko, V. F., Kushnir A. F., and Savin I. V., 1987, Statistical adaptive algorithms for estimation of onset moments of seismic phases, *Physics of the Earth and Planetary Interiors*, **47**, 4-10
- Poulton, M. M., Sternberg, B. K., and Glass, C. E., 1992, Location of subsurface target in Geophysical data using neural networks, *Geophysics*, **57**, 1534-1544.
- Prechelt, L., 1995, *Neural-net-faq*, Usenet newsgroup comp.ai.neural-nets.
- Raiche, A., 1991, A pattern recognition approach to geophysical inversion using neural nets, *Geophysical Journal International*, **105**, 629-648.
- Riviere-Barbier, F. and Grant L. T., 1993, Identification and Location of Closely Spaced Mining Events, *Bulletin of the Seismological Society of America*, **83**, 1527-1546.
- Roberts, R. G., and Christoffersson, A., 1990, Decomposition of complex single-station three-component seismograms, *Geophysical Journal International*, **103**, 55-74.
- Roberts, R. G., Christoffersson, A., and Cassidy, F., 1989, Real-time event detection, phase identification and source location estimation using single station three-component seismic data, *Geophysical Journal International*, **97**, 471-480.
- Roth, G., and Tarantola, A., 1994, Neural networks and inversion of seismic data, *Journal of Geophysical Research*, **89**, No B4, 6753-6768.
- Rumelhart, D., Hinton, G. E., and Williams, R. J., 1986, Learning representations by back-propagating errors, *Nature*, **323**, 533-536.

- Rumelhart, D., Hinton, G. E., and Williams, R. J., 1988, *Learning Internal Representations by error propagation, Chapter 8 in Parallel Distributed Processing: Explorations in the microstructure of Cognition Volume 1: Foundations, (edited by Rumelhart, D. E., McClelland, J. L. and the PDP Research Group, 1988)*, The MIT Press, Massachusetts.
- Ruud, B. O., and Husebye, E. S., 1992, A new three-component detector and automatic single-station bulletin production, *Bulletin of the Seismological Society of America*, **82**, 221-237.
- Samson, J. C., 1977, Matrix and Stokes vector representations of detectors for polarized waveforms: theory, with some applications to teleseismic waves, *Geophysical Journal of Royal Astronomical Society*, **51**, 583-603.
- Stevenson, P. R., 1976, Microearthquakes at Flathead Lake, Montana: A study using automatic earthquake processing, *Bulletin of the Seismological Society of America*, **66**, 61-80.
- Stewart, S. W., 1977, Real-time detection and location of local seismic events in central California, *Bulletin of the Seismological Society of America*, **67**, 433-452.
- Swindell, W. H. and Snell, N.S., 1977, *Station processor automatic signal detection system, Phase 1: Final report: Station processor software development*, Texas Instrument Report No. Alex (01)-FR-77-01, Texas Instruments Incorporated, Dallas, Texas.
- Takanami T. and Kitagawa, G, 1988, A new efficient procedure for the estimation of onset times of seismic waves, *Journal of Physics of Earth*, **36**, 267-290.
- Takanami T. and Kitagawa, G, 1993, Multivariate time-series model to estimate the arrival times of S-wave, *Computers and Geosciences*, **19**, 295-301.
- Tao, X. X. and Du, W., 1992, Seismicity trends along the southeastern coastal seismic zone assessed by artificial intelligence, *Earthquake Research in China*, **6**, No.1, 91-98.

- Tarvainen, M., 1992, Automatic seismogram analysis: statistical phase picking and location methods using one station three-component data, *Bulletin of the Seismological Society of America*, **82**, 860-869.
- Tong, C., 1995, Characterization of seismic phase - an automatic analyser for seismograms, *Geophysical Journal International*, **123**, 937-947.
- Tong, C., and Kennett, B. L. N., 1995, Towards the identification of later seismic phases, *Geophysical Journal International*, **123**, 548-958.
- Tubb, N. R., 1993, An introduction to neural computing, *Embedded System Engineering*, **1**, No 5, 41-44.
- Vanderkulk, W., Rosen, F. and Lorenz, S., 1965, *Large aperture seismic array signal processing study*, IBM Final Report, ARPA contract SD-296, 15 July 1965.
- Veith, K. F., 1978, *Seismic signal detection algorithm: Technical Note 1/78*, Teledyne Geotech, Texas, pp 1-9
- Wirth, M H., Blandford, R. R. and Shumway, R. H., 1976, Automatic seismic array and network detection, *Bulletin of the Seismological Society of America*, **60**, 1375-1380.
- Wang, L. X. & Mendel, J. M., 1992, Adaptive minimum prediction error deconvolution and source wavelet estimation using neural network, *Geophysics*, **57**, 670-679.
- Wang, J., and Teng, T. L., 1995, Artificial neural network-based seismic detector, *Bulletin of the Seismological Society of America*, **85**, 308-319.
- Zhang, Q., Song, J. R. and Nie, X. Y., 1991, Application of neural networks models to rock mechanics and rock engineering, *Int. J. Rock Mech. Min. Sci. & Geomech. abstr.*, **28**, 535-540.

## LIST OF PUBLICATIONS

- Hengchang Dai and Colin Macbeth, 1995, Arrival type identification in local earthquake data using artificial neural network, *Proceedings of the 2nd Workshop on Application of Artificial Intelligence Techniques in Seismology and Engineering Seismology*", Walferdange, Luxembourg, 4-6 October 1995, *in press*.
- Hengchang Dai and Colin MacBeth, 1995, Automatic picking of seismic arrivals in local earthquake data using an artificial neural network, *Geophysical Journal International*, **120**, 758-774
- Hengchang Dai and Colin MacBeth, 1994, Split shear-wave analysis using an artificial neural network, *First Break*, **12**, No. 12, 605-613.
- Hengchang Dai and Colin MacBeth, 1994, Automatic picking seismic arrivals of single component recordings from local earthquake data using artificial neural networks. *British Geological Survey internal report: WL/94/27*.
- Hengchang Dai and Colin MacBeth, 1994, A neural network picker for VSP data, *Extended Abstract in 56th meeting of European Association of Exploration Geophysicists, Vienna, Austria, 6-10, June 1994*.
- Hengchang Dai and Colin MacBeth, 1993, Analysis of split shear-waves from near-offset VSP data using a neural network, *Extended Abstract in 55th meeting of European Association of Exploration Geophysicists, Stavanger, Norway, 7-11, June 1993*.



19-12-1995 11:45

\*\*\* ORB \*\*\* KSB \*\*\*

32 2 3749822 P.01

CENTRE EUROPEEN DE GEODYNAMIQUE ET DE SEISMOLOGIE  
EUROPEAN CENTER FOR GEODYNAMICS AND SEISMOLOGY  
(E.C.G.S.)

Président: Ing. J. FLICK  
Musée d'Histoire Naturelle

Secrétaire: Prof. B. DUCARME  
Observatoire Royal de Belgique

Tel : 352/441652  
Fax : 352/458940

Tel : 32/2/3730248  
Fax : 32/2/3749822

FAX COVER SHEET

Date : Brussels, December 19, 1995

Number of pages incl. this one : 1

Fax nr. : 00 44 131 667 1877

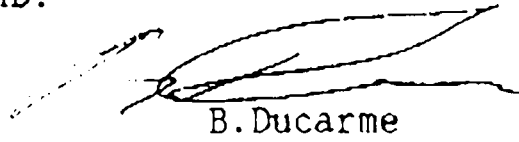
Attention : Mr. Hengchang Dai, British Geological Survey

From : B. Ducarme, Secretary ECGS

Concern : Paper to be published in " Cahiers " of ECGS

Message : Dear Mr. Dai,

You can of course include a copy of your paper "Arrival type identification in local earthquake data using an artificial neural network" in your Ph D thesis. You should indicate proposed for publication in the Proceedings of the 2nd Workshop on Application of Artificial Intelligence Techniques in Seismology and Engineering Seismology, Cahiers du Centre Européen de Géodynamique et de Séismologie. In fact there is still a reviewing procedure on the way. I wish you much success for your PHD.  
Sincerely yours

  
B. Ducarme

# ARRIVAL TYPE IDENTIFICATION IN LOCAL EARTHQUAKE DATA USING AN ARTIFICIAL NEURAL NETWORK

Hengchang Dai<sup>1,2</sup> and Colin MacBeth<sup>1</sup>

<sup>1</sup> *British Geological Survey, Murchison House, West Mains Road, Edinburgh EH9 3LA Scotland, UK.*

<sup>2</sup> *Department of Geology and Geophysics, University of Edinburgh, Edinburgh EH9 3JW, Scotland, UK.*

## ABSTRACT

A preliminary study is performed to test the ability of an artificial neural network (ANN) to identify seismic arrival types from local earthquake data after they are picked. This is achieved using the degree of polarization (DOP) for a segment of three-component time series, as the ANN input. The ANN was designed to classify arrivals into three groups: *P*-arrivals, *S*-arrivals and noise, corresponding to the maximum output of the output nodes. The ANN was trained with nine groups of segments of *P*- and *S*-arrivals and noise. 327 pre-triggered recordings from a station in a local earthquake network are firstly processed by an ANN picker for all possible *P*- and *S*-arrivals and measured their onset times. Segments of the DOP are selected according to these onset and then fed into the trained ANN. Compared with manual analysis, the trained ANN can correctly identify 84% *P*-arrivals and 63% *S*-arrivals. Its performance has inherent limitations due to the complexity of DOP patterns which cannot be improved by simply adding new training datasets. The example is shown that the ANN trained with data from one station fails to deal with seismic data from another station as the DOP patterns are station-dependent. This limitation shows that selecting the input information is critical. The ANN has potential as a tool to identify the arrivals type automatically but needs to be associated with other information.

**Keywords:** artificial neural network, seismic arrival identification, degree of polarization.

## 1. INTRODUCTION

The most important procedure of analysing earthquake events is the estimation of the arrival times of the primary (*P*) and secondary (*S*) waves, as these measurements form the basis of subsequent analysis schemes employing processing for event location, event identification, source mechanism analysis and spectral analysis. These tasks are often performed by the trained analyst who manually picks arrival times according to his individual experience, involving an intensive amount of pattern recognition. With the increase in the number of digital seismic networks being established worldwide, there is a pressing need to provide an automatic alternative, which is more reliable, robust, objective and less time-consuming.

The estimation of arrival times includes two steps: 1) arrival picking which is a reliable and accurate estimation of the onset time of a definite seismic arrival; 2) arrival identification which classifies individual arrivals into categories relating to their polarization, their amplitude and the nature of their propagation. A great deal of effort, stretching back several decades, has been devoted to the automation of arrival picking (Allen, 1982; Bache *et al*, 1990; Bear and Kradolfer, 1987; Chiaruttini, Roberto and Saitta, 1989; Chiaruttini and Salemi, 1993, Dai and MacBeth, 1995; Houlston, Waugh and Langhlin, 1984; Jowsig, 1990, 1995; Jowsig and Schulte-Theis, 1993; Kracke, 1993; Klumpen and Jowsig, 1993, Pisarenko, Kushnir and Savin, 1987; Takanami and Kitagawa, 1988, 1993). Identifying arrival types is

more difficult than picking their onset time and still is an unresolved problem. For a seismic network, arrival identification based on horizontal velocity from an  $f$ - $k$  filter can provide a major simplification of the interpretation task (Mykkeltveit and Bungum, 1984; Bache *et al.*, 1990; Kvaerna and Ringdal, 1992). Der, Baumgardt and Shumway (1993) have investigated the feasibility of adaptive, automatic recognition of regional arrivals by a wavefield extrapolation scheme for data from a mini-array. However for single station data, there are few methods which can be used to pick special type arrivals. Roberts, Christoffersson and Cassidy (1989), based on the auto- and cross-correlations of the three orthogonal components within a short time window, detect the arrival of a  $P$ -wave or a linearly polarized  $S$ -wave. Cichowicz (1993) developed a  $S$ -phase picker which depends on a well defined pulse of the first-arrival  $P$ -wave. In this paper, we will introduce an artificial neural network (ANN) approach to the identification problem.

## 2. THE LOCAL EARTHQUAKE DATA

In this work, real earthquake data are used to design and test an ANN approach. The data are local earthquake events recorded at station DP which is located near the centre of the TDP3 seismic network and station AY which is on the edge of the TDP3 seismic network (Lovell, 1989) between April 1984 and December 1984. Several hundred local earthquakes are recorded on three-component seismometers at a 10ms sampling interval. These recordings are not continuous and are triggered by a digital system (Evans *et al.* 1987). All are local, with depths from 2km to 14km and epicentral distances less than 30km from the stations, and most are closer to station DP than to station AY. For these local events, we identified predominant  $P_g$  and  $S_g$  waves in the seismogram records. Most events have magnitudes ( $M_L$ ) between -0.3 and 1.0, and possess a wide distribution of signal-to-noise ratio (SNR) which are shown in Fig. 1 for the complete dataset. All SNRs lie between 1 and 200, with station DP being of higher fidelity than station AY. Not all of these recording can be used due to following reasons. Some events were not earthquakes and some small earthquakes, recorded on Station DP or AY, were not confirmed by network data. These recordings are discarded by comparing with network data. In some cases, the seismometers did not function properly and either one or two components were inactive or possessed high amplitude noise so that some of the three component sets were incomplete; and some recordings have excessive noise preceding the events or ringing throughout the record which produce many false alarms. Here we manually selected the recordings in which the earthquake event has been confirmed by the seismic network data. We must be aware that our statistics will appear more successful than if this procedure had been applied to all the data irrespective of quality. In total, 327 recordings in station DP and 282 recordings in station AY were selected respectively for further processing. We can visually pick 333  $P$ -arrivals and 317  $S$ -arrivals at DP and 283  $P$ -arrivals and 261  $S$ -arrivals at AY. All these recordings were processed by an ANN arrival picker (Dai and MacBeth, 1995) to measure the onset times of all possible  $P$ - and  $S$ -arrivals. Compared with the manual analysis, the ANN picks 326 (97%)  $P$ -waves and 286 (92%)  $S$ -waves at station DP and 242 (87%)  $P$ -waves and 235 (90%)  $S$ -waves at station AY.

## 3. THE DEGREE OF POLARIZATION

The identification of different arrival types is accomplished using a combination of the degree of polarization (DOP) of arrival and the vector modulus of its three-component motion. In

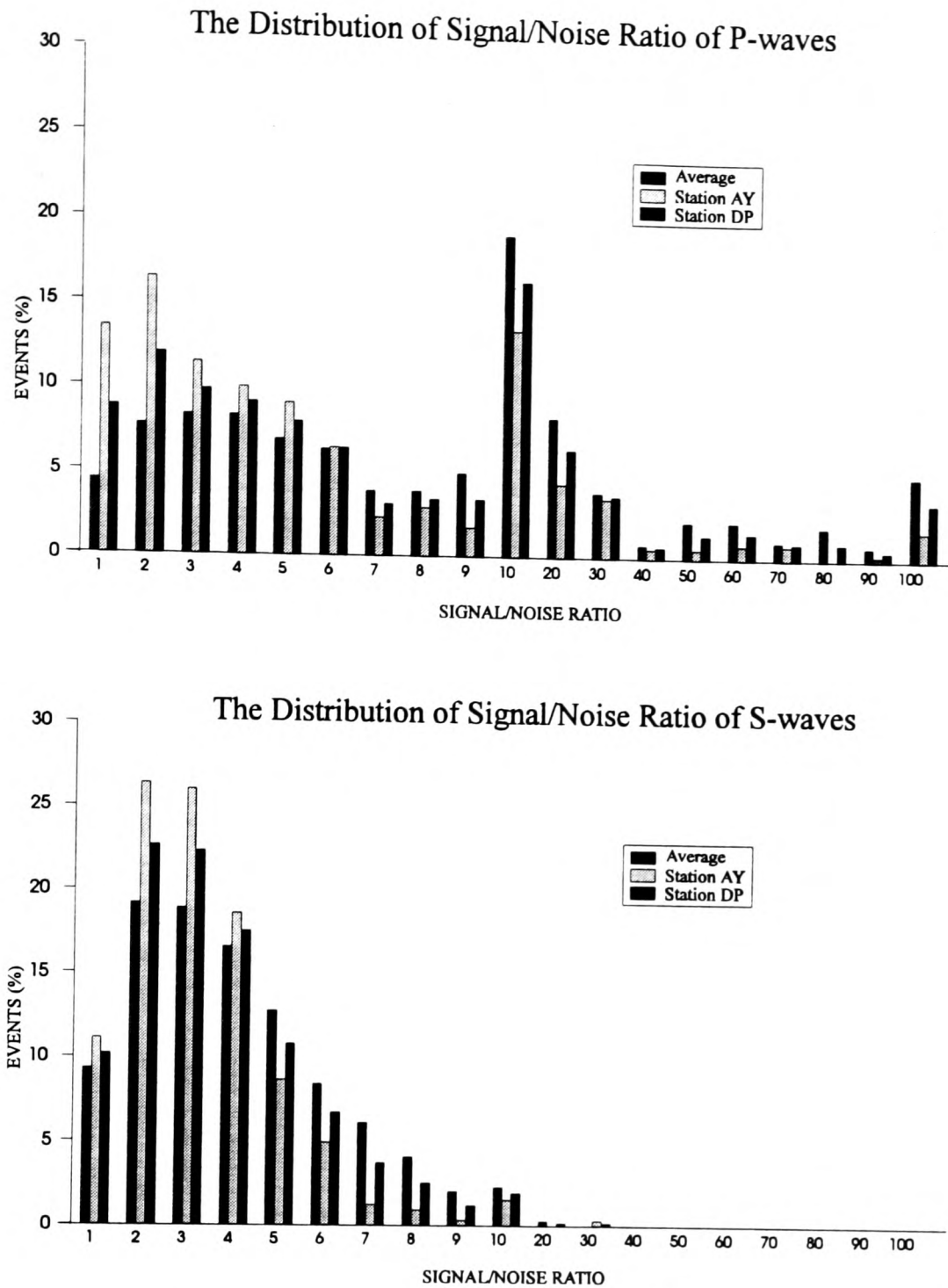


Figure 1 The distribution of signal/noise ratio (SNR) of P-waves and S-waves in the dataset. SNR is defined as the ration between the maximum of modulus of the three component recordings before and after the onset time. Notation for SNR represents 1 for [1,2), 2 for [2,3), ..., 10 for [10,20), ..., and 100 for [100, 200).

each single component, the seismic signals are strongly dependent on the source position and ray direction, which may otherwise give rise to a misleading interpretation. We must separate this dependency from the seismic recordings. In this paper, we input the DOP which is independent of the source position. The DOP is calculated from the covariance matrix of 3-C recordings which is a useful measure of the polarization of seismic signal (Samson, 1977; Cichowicz, Green and Brink, 1988; Cichowicz, 1993). The covariance matrix is defined as:

$$C = \begin{Bmatrix} COV(X,X) & COV(X,Y) & COV(X,Z) \\ COV(Y,X) & COV(Y,Y) & COV(Y,Z) \\ COV(Z,X) & COV(Z,Y) & COV(Z,Z) \end{Bmatrix} ;$$

where the covariance is measured for N samples:

$$COV(X,Y) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})$$

where  $\bar{x}$  and  $\bar{y}$  are the average values of  $X$  and  $Y$ .  $N$  is determined from the signal predominant frequency (Cichowicz, 1993), which is ten samples in this case.

The diagonalization of the covariance matrix gives the principal axis of this matrix. The direction of polarization is measured by considering the eigenvector of the largest principal axis. This direction is parallel to the propagation direction for a  $P$ -wave and is perpendicular to the propagation direction for a  $S$ -wave in an isotropy medium. It is difficult to use this direction, related to the source position, as a decision parameter for arrival identification. Some parameters which are independent of the source location should be defined to extract the polarization properties. Samson (1977) defines the degree of polarization as:

$$F(t) = \frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2}{2(\lambda_1 + \lambda_2 + \lambda_3)^2} = \frac{3trS^2 - (trS)^2}{2(trS)^2}$$

where the  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are eigenvalues of the covariance matrix of a moving window of width  $N$  samples;  $trS$ , defined as  $\lambda_1 + \lambda_2 + \lambda_3$ , is the trace of  $C$  and  $trS^2$ , is defined as  $\lambda_1^2 + \lambda_2^2 + \lambda_3^2$ . This equation shows that the function can be calculated without having to diagonalize the covariance matrix. As these are independent of the coordinate system, they also are independent of the source location and depend only on the polarization state. According to this definition, if only one eigenvalue is non-zero, then  $F=1$ , and the signal is linearly polarized; if all of the eigenvalues are equal, then  $F=0$ , and the signal can be considered as completely unpolarized or circularly polarized. Thus  $F(t)$  enables us to study the evolution of the degree of polarization (DOP).

For the data used, most  $P$ -arrivals have high values of  $F(t)$  and most  $S$ -arrivals have medium or low values of  $F(t)$ . Fig. 2 shows an example. The patterns of polarization are too complex to find

Station: DP  
Date: 1984-05-06  
Start-time: 12h15m58s  
Scale: 563

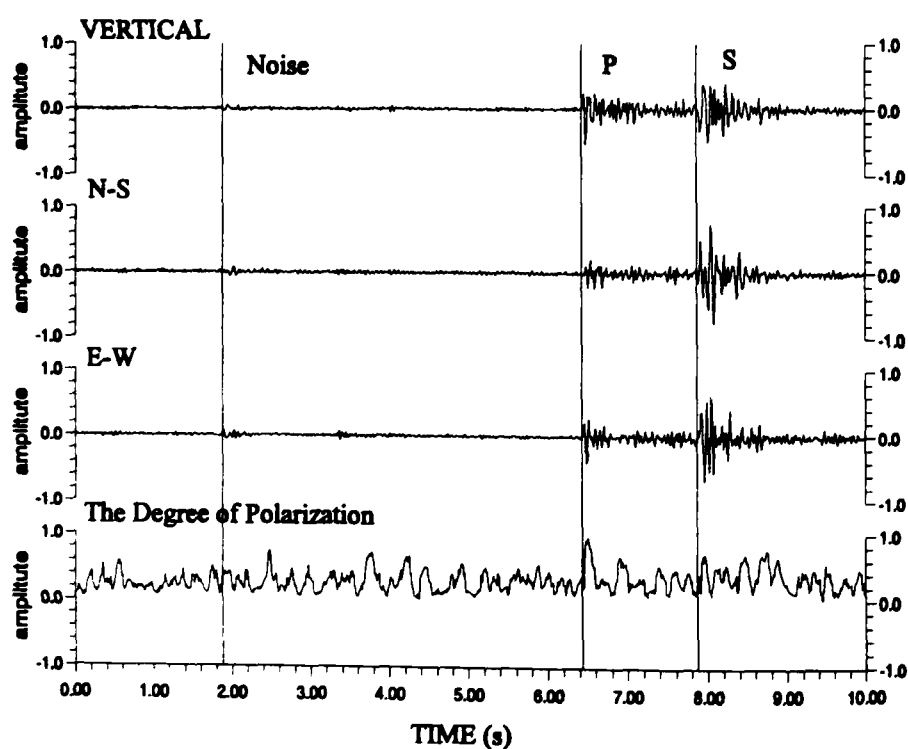


Figure 2. The degree of polarization and three-component seismograms. Three vertical lines indicate the arrivals onset times of noise, a  $P$ -arrival and a  $S$ -arrival. The degree of polarization has a low value for the noise, a high value for the  $P$ -arrival and a middle value for the  $S$ -arrival.

threshold values to distinguish them. Most of the  $F(t)$  patterns of  $P$ -arrivals usually differ from those of  $S$ -arrivals. There are also some noise bursts whose  $F(t)$  patterns are similar to those from the seismic arrivals. Spikes manifest as special patterns in which  $F(t)$  is very high, near unity, with about ten point length and can be easily discarded by a conventional program. Comparing the data from Stations DP and AY, the  $F(t)$  patterns are different even for the same earthquake and arrival.

To calculate the degree of polarization, all three components must have the same frequency bandwidth, the same scale and the same noise level. If one of the three-components has a different property, the DOP is highly biased. In this case, even manual identification using only the DOP cannot be used. This particular definition of DOP does not take into account the signal intensity. Different arrival types not only have different polarization characteristics, but also have different amplitude characteristics. To take into account both polarization and amplitude information, we define a modified function of the DOP:

$$MF(t) = F(t) \times \bar{M}(t)$$

where  $\bar{M}(t)$  is the smoothed relative function of modulus  $M(t)$ , defined as  $[x(t)^2 + y(t)^2 + z(t)^2]^{1/2}$ , of 3-C recording in a window which is also independent of the source position. The normalization factor is taken from the window between the onset point and following ten points, in which the maximum is defined as unity. Note that  $MF(t)$  and  $F(t)$  may have slightly different patterns.  $MF(t)$  is now presented to the neural network in segments selected from a window in which the centre is the onset-time of arrival.

## 4. IDENTIFYING ARRIVAL TYPES USING AN ANN

### 4.1 ANN structure

The ANN used in this study is a nonlinear, multilayer, feed-forward and back-propagation of error (Rumelhart, Hinton, and Williams, 1986). This is the most popular type of ANN in use today as it is well understood. It also incorporates a back-propagation learning algorithm, or *Delta Rule* which is usually used to train this type of ANN -- a good mathematical summary is given by Pao (1988).

This ANN has three layers, the input having 60 nodes, giving a  $MF(t)$  segment with a fixed 590ms (60 samples) length which is chosen to include several complete cycles of a wave. There are three nodes in the output layer to flag the result: the output is (1,0,0) for noise; (0,1,0) for a  $P$ -arrival; and (0,0,1) for a  $S$ -arrival in training. The number of hidden nodes depends on various factors such as input nodes, output nodes, system error, pattern error, and training samples. There is no fixed generic relationship between the number and these factors for this type of ANN. However, we do know that in ANN learning, generalization is increased and memory is reduced by limiting the number of hidden nodes (Dowla, Taylor and Anderson, 1990). Too few hidden nodes will lead to a long learning or no convergence. In this case we chose ten hidden nodes after a process of trial and error with different training runs.

### 4.2 Training procedure

As each segment of  $MF(t)$  is fed into the trained ANN, the output will be three values:  $o_1$ ,  $o_2$ , and  $o_3$ . If the segment is the same as the training segment, the output will be perfectly (1,0,0) for noise; (0,1,0) for the  $P$ -wave; and (0,0,1) for the  $S$ -wave. For non-training segments, the output ( $o_1$ ,  $o_2$ , and  $o_3$ ) is the measurement of similarity between the new



segment and training segment. To identify segment types, we simply seek the maximum of the three outputs ( $o_1$ ,  $o_2$ ,  $o_3$ ). If  $o_1$  is the maximum, this segment belongs to the noise category; if  $o_2$  is the maximum, it belongs to the *P*-wave; and if  $o_3$  is the maximum, it belongs to the *S*-wave. This method is also applied to some segments which are far different from training segments and have low outputs.

In this procedure, only a small number of recordings from station DP are used to train the ANN and the remainder to test its performance. The ANN performance depends on the training datasets, if we use incorrect or inconsistent data to train the ANN, we cannot expect it to give a correct answer for new data. *P*- and *S*-arrivals with similar MF(t) patterns should be avoided. At the beginning of training, we select only three MF(t) segments for noise, *P*- and *S*-arrivals. Using manual analysis results, we select another three segments, to combine with the former training segments, to train again. This procedure is repeated until the performance of the trained ANN cannot be improved by increasing the training dataset or we are satisfied by its performance. Fig. 3 shows all training segments of MF(t) used in this particular study.

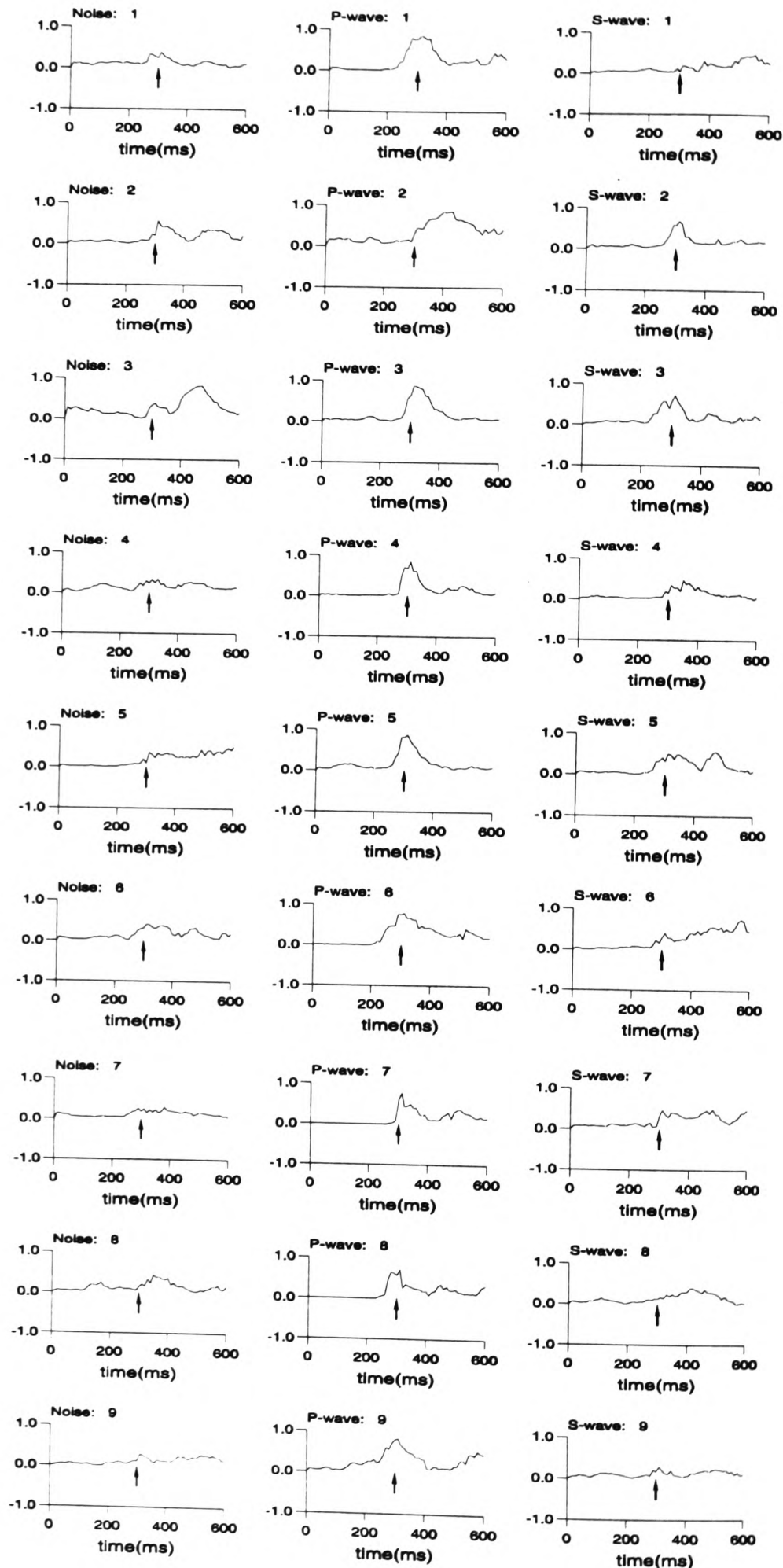


Figure 3. Nine group of MF(t) segment of noise, *P*-waves and *S*-waves. The arrows indicate the adjusted onset times

### 4.3 Testing procedure

Here we focus on the data from station DP. Unlike our earlier ANN arrival picking, the ANN is not used as a filter to deal with an entire seismic trace, but arrival segments are input which are picked previously. The results are time-sensitive and a small shift in segment can greatly effect the output. Because the onset-times of picked arrivals have errors, we must adjust the onset time to ensure the performance of the trained ANN is not affected by the onset time error. For each MF(t) segment, we set the first local maximum after the onset-time on the centre of the segment. To test the trained ANN performance, we input all the pre-picked arrival segments into this neural network.

**Table 1:** The performance of the ANN trained with 9 groups of training segments. The ANN has 60 input nodes. The percentage is the BPNN identifying result.

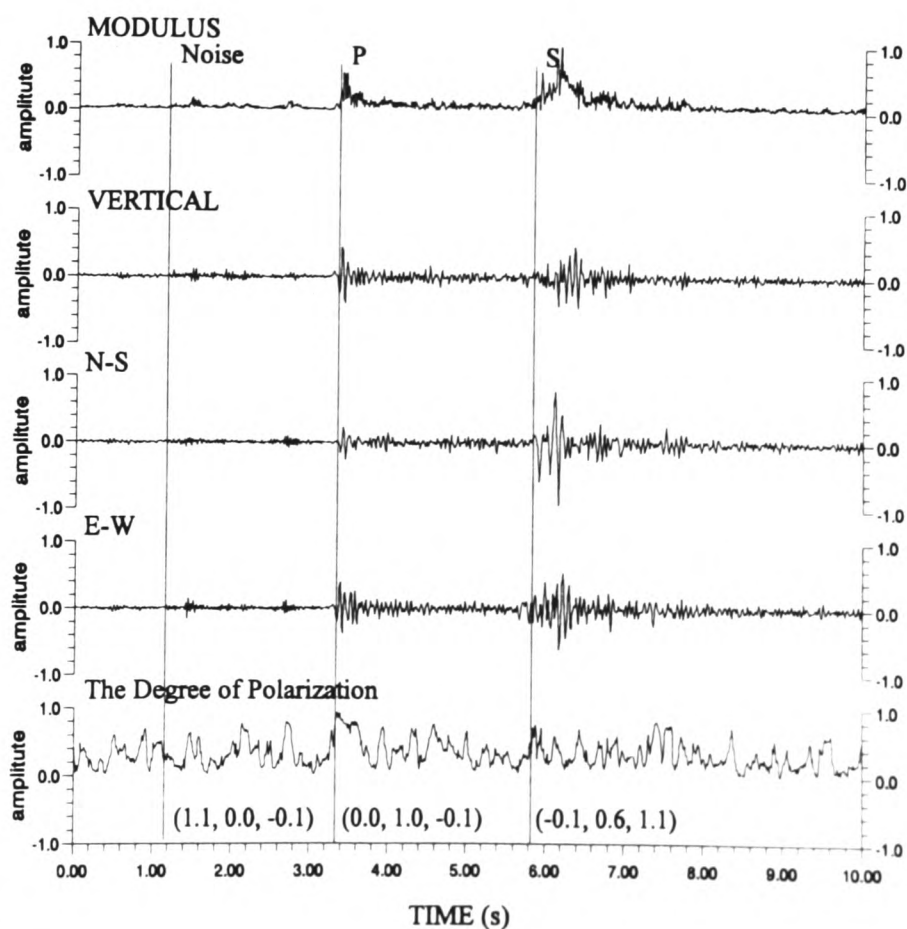
Manual Picks	P-arrivals (326)	S-arrivals (286)	Noise (146)
P-arrivals	84% (274)	20% ( 58)	10% (16)
S-arrivals	11% ( 35)	63% (180)	42% (62)
Noise	5% ( 17)	17% ( 49)	47% (69)

### 4.4 The performance of the trained ANN

For a three-layer ANN with 60, 10, 3 nodes, the final training was taken by using 27 training segments (nine noise, nine *P*-arrivals and nine *S*-arrivals). Training takes 1422 iterations, about 1.5 minutes on a VAX 4000. Table 1 shows the performance of the final trained ANN, with Fig. 4 displaying an example of correct identification. However, if we use this to deal with Station AY, it only identifies 43% of *P*-arrivals and 41% of *S*-arrivals since many *P*-arrivals have low values of  $F(t)$  and most of *S*-arrivals have high values of  $F(t)$  which is contrary to the training. Another ANN is needed to specifically process data from station AY.

Table 2 shows a comparison of three trained ANNs with different datasets. As the datasets increase, the performance for identifying *P*-arrivals improves, but the performance for identifying *S*-arrivals and noise becomes worse.

Station: DP  
Date: 1984-05-31  
Start-time: 09h23m36s  
Scale: 373



**Figure 4.** Three component seismograms, the vector modulus, and the degree of polarization of a local earthquake. Three vertical lines indicate the arrival onsets of noise, a *P*-arrivals and a *S*-arrival. The ANN correctly identifies them with its output (1.1, 0.0, -0.1), (0.0, 1.0, -0.1) and -0.1, 0.6, 1.1) respectively.



This is due to the complexity of the MF(t) patterns. MF(t) patterns of *P*-waves are more typically alike, and MF(t) patterns of *S*-waves are quite different. In addition, some *P*-arrivals, *S*-arrivals and noise have similar MF(t) patterns. If we use such a *P*-arrival pattern to train the ANN, the trained ANN will generate all them as *P*-arrivals regardless of what they are. It seems that using the MF(t) alone is not enough to solve this problem. Other properties, for example, the direction of polarization, or some other methods, such as an expert system, may be needed.

We also investigated the sensitivity to the input segment length as this decides the ANN structure. Various input nodes were tested, between 50 and 70 nodes, retaining the same hidden nodes and output nodes. The training procedure is the same: beginning with one group of training segments and increasing to nine groups. The training segments are different for these three ANNs due to their different performance at every training stage (Table 3). On balance, the ANN with 60 input nodes has the best performance. From this a rough guideline is suggested: the segments should include several complete cycles of a wave. This reflects the general observation that network architecture must be specifically tailored to individual applications. Further optimization is required to adapt to particular event types.

## 5. CONCLUSIONS

An ANN is used as a tool to identify *P*- and *S*-arrivals from local earthquake data, using the polarization state of three-component records. Our results demonstrate that an ANN trained using a small subset of the data can identify most *P*-arrivals (84%) and *S*-arrivals (63%) simultaneously. This high performance, combined with the advantage of not requiring programs to construct special variables and parameters with complicated mathematics, suggest that the ANN is a natural choice for such applications. The method is adaptive, and training sets can be altered to enhance particular features of different datasets. Adding new training

**Table 2:** The comparison of the performances of three ANNs trained with different training dataset Only correct identification oercentage are shown. The best performance is with 9 training groups

	P-arrivals	S-arrivals	Noise
8 training groups	82%	58%	51%
9 training groups	84%	63%	47%
10 training groups	87%	59%	38%

**Table 3:** The comparison of the performance of the ANN with different input nodes trained with nine groups of segments. Only correct identification percentages are shown in this table. The best performance is from the ANN with 60 input nodes.

	P-arrivals	S-arrivals	Noise
50 input nodes	89%	45%	47%
60 input nodes	84%	63%	47%
70 input nodes	88%	44%	48%

datasets and retraining an ANN is easy and quick, and can improve its performance. However it also appears to have a limitation due to the inter-station complexity of the DOP.

Although the training time can be long, especially as the ANN architecture becomes large, once trained the ANN is sufficiently quick to operate in most real-time applications. However, the ANN cannot be viewed as all encompassing, as the performance still depends upon the training set and its ability to predict cannot lie too far outside its experience. The exact boundaries of this behaviour have not yet been completely explored. Another limitation is in finding an optimal architecture for a particular application.

This work forms part of an ongoing programme of research to develop a fully automatic system for earthquake analysis. It is ultimately hoped to integrate other ANN units into a processing flow for record editing and event classification.

## ACKNOWLEDGEMENTS

This research was sponsored by the Global Seismology Research Group (GSRG) at the British Geological Survey (BGS), Edinburgh, and is published with the approval of the Director of BGS. We thank Dr. Chris Browitt, Programmes Director of BGS, Mr. Terry Turbitt, Programmes Manager of GSRG, Dr. David Booth and Mr. John Lovell for supplying the earthquake data. Thanks are also extended to the staff and students of GSRG for their support and encouragement with this work.

## REFERENCES

- Allen, R. V., 1978, Automatic earthquake recognition and timing from single traces, *Bulletin of the Seismological Society of America*, **68**, 1521-1532.
- Bache, T. C., Bratt, S. R., Wang, J, Fung, R. M., Kobryn, C., and Given, J. W., 1990, The intelligent monitoring system, *Bulletin of the Seismological Society of America*, **80**, 1833-1851.
- Bear, M. and Kradolfer, U., 1987, An automatic phase picker for local and teleseismic event, *Bulletin of the Seismological Society of America*, **77**, 1437-1445.
- Chiaruttini, C., Roberto, V., and Saitta, F., 1989, Artificial intelligence techniques in seismic signal interpretation, *Geophysical Journal International*, **98**, 223-232.
- Chiaruttini, C. and Salemi, G., 1993, Artificial intelligence techniques in the analysis of digital seismograms, *Computer and Geosciences*, **19**, 149-156.
- Cichowicz, A., Green, R., and Brink, A., 1988, Code polarization properties of high-frequency microseismic events, *Bulletin of the Seismological Society of America*, **78**, 1297-1318.
- Cichowicz, A., 1993, An automatic S-phase picker, *Bulletin of the Seismological Society of America*, **83**, 180-189.
- Dai, H., and MacBeth, C., 1995, Automatic picking of seismic arrivals in local earthquake data using an artificial neural network, *Geophysical Journal International*, **120**, 758-774.
- Der, Z., Baumgardt, D., and Shumway, R., 1993, The nature of particle motion in regional seismograms and its utilization for phase identification, *Geophysical Journal International*, **115**, 1012-1024.
- Dowla, F. U., Taylor, S. R., and Anderson, R. W., 1990, Seismic discrimination with artificial neural networks: preliminary results with regional spectral data, *Bulletin of the Seismological Society of America*, **80**, 1910-1933.

- Evans, S. A., Beamish, D., Crampin, S., and Ucer, S. B., 1987. The Turkish Dilatancy Project (TDP3): multidisciplinary studies of a potential earthquake source region, *Geophys. J. R. astr. Soc.*, **91**, 265-286.
- Houliston, D. J., Waugh, G., and Laughlin, J., 1984, Automatic real-time event detection for seismic networks, *Computer and Geosciences*, **10**, 431-436.
- Joswig, M., 1990, Pattern recognition for earthquake detection, *Bulletin of the Seismological Society of America*, **80**, 170-186.
- Joswig, M., 1995, Automated classification of local earthquake data in the BUG small array, *Geophysical Journal International*, **120**, 262-286.
- Joswig, M., and Schulte-Theis, H., 1993, Master-event correlations of weak local earthquake by dynamic waveform match, *Geophysical Journal International*, **113**, 562-574.
- Klumpen, E., and Joswig, M., 1993, Automated reevaluation of local earthquake data by application of generic polarization patterns for P- and S-onset, *Computer and Geosciences*, **19**, 223-231.
- Kracke, D., 1993, A three-component event detector based on waveform analysis, *Computers and Geosciences*, **19**, 117-122.
- Kvaerna, T. and Ringdal, F., 1992, Integrated array and three-component processing using a seismic microarray, *Bulletin of the Seismological Society of America*, **82**, 870-882.
- Lovell, J., 1989, Source parameters of a microearthquake swarm in Turkey, *Master of Philosophy thesis, University of Edinburgh. Scotland, UK.*
- Mykkeltveit, S. and Bungum, H., 1984, Processing of regional seismic events using data from small aperture arrays, *Bulletin of the Seismological Society of America*, **74**, 2313-2333.
- Pao, Y. H., 1988, *Adaptive pattern recognition and neural networks*, Addison-Wesley Publishing Company, Inc, New York.
- Pisarenko, V. F., Kushnir, A. F., and Savin, I. V., 1987, Statistical adaptive algorithms for estimation of onset moments of seismic phases, *Physics of Earth and Planetary Interiors*, **47**, 4-10.
- Roberts, R. G., Christoffersson, A., Cassidy, F., 1989, Real-time event detection, phase identification and source location estimation using single station three-component seismic data, *Geophysical Journal International*, **97**, 471-480.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, Learning internal representation by backpropagating errors, *Nature*, **323**, 533-536.
- Samson, J., 1977, Matrix and Stokes vector representations of detectors for polarized waveforms: theory, with some applications to teleseismic waves, *Geophysics J. R. astr. Soc.*, **51**, 583-603.
- Takanami, T., and Kitagawa, G., 1988, A new efficient procedure for the estimation of onset times of seismic waves, *Journal of Physics of Earth*, **36**, 267-290.
- Takanami, T., and Kitagawa, G., 1993, Multivariate time-series model to estimate the arrival times of S-waves, *Computers and Geosciences*, **19**, 295-301.

# Automatic picking of seismic arrivals in local earthquake data using an artificial neural network

Hengchang Dai<sup>1,2</sup> and Colin MacBeth<sup>1</sup>

<sup>1</sup> *British Geological Survey, Murchison House, West Mains Road, Edinburgh EH9 3LA, UK*

<sup>2</sup> *Department of Geology and Geophysics, University of Edinburgh, Kings Buildings, West Mains Road, Edinburgh EH8 3JW, UK*

Accepted 1994 September 23. Received 1994 September 23; in original form 1993 November 4

## SUMMARY

A preliminary study is performed to test the ability of an artificial neural network (ANN) to detect and pick seismic arrivals from local earthquake data. This is achieved using three-component recordings by utilizing the vector modulus of these seismic records as the network input. A discriminant function,  $F(t)$ , determined from the output of the trained ANN, is then employed to define the arrival onset. 877 pre-triggered recordings from two stations in a local earthquake network are analysed by an ANN trained with only nine  $P$  waves and nine noise segments. The data have a range of magnitudes ( $M_L$ ) from  $-0.3$  to  $1.0$ , and signal-to-noise ratios from 1 to 200. Comparing the results with manual picks, the ANN can accurately detect 93.9 per cent of the  $P$  waves and also 90.3 per cent of the  $S$  waves with a  $F(t)$  threshold set at 0.6 (maximum is 1.0). These statistics do not include false alarms due to other non-seismic signals or unusable records due to excessive noise. In 17.2 per cent of the cases the ANN detected false alarms prior to the event. Determining the onset times by using the local maximum of  $F(t)$ , we find that 75.4 per cent of the  $P$ -wave estimates and 66.7 per cent of the  $S$ -wave estimates are within one sample increment (10 ms) of the reference data picked manually. Only 7.7 per cent of the  $P$ -wave estimates and 11.8 per cent of the  $S$ -wave estimates are inaccurate by more than five sample increments (50 ms). The majority of these records have distinct local  $P$  and  $S$  waves. The ANN also works for seismograms with low signal-to-noise ratios, where visual examination is difficult. The examples show the adaptive nature of the ANN, and that its ability to pick may be improved by adding or adjusting the training data. The ANN has potential as a tool to pick arrivals automatically. This algorithm has been adopted as a component in the early stages of our development of an automated subsystem to analyse local earthquake data. Further potential applications for the neural network include editing of poor traces (before present algorithm) and rejection of false alarms (after this present algorithm).

**Key words:** arrival time, artificial neural network, pattern recognition, picking.

## 1 INTRODUCTION

The primordial task of estimating arrival times for the primary ( $P$ ) and secondary ( $S$ ) waves found in recordings of an earthquake event still forms an important foundation for schemes employing automatic processing for event location, event identification, source mechanism analysis and spectral analysis. There is no shortage of techniques which profess to tackle this problem; however, they do tend to be data specific and are not generally available. The goal of global automation is far from achieved and such elementary

seismogram interpretation still forms a bottleneck in the routine work of many observatories.

A great deal of effort, stretching back several decades, has been devoted to the automation of arrival picking, and many different varieties of algorithm exist. Only a few notable procedures are mentioned in this review for reasons of brevity. The method popularized by Allen (1978) uses a short-term and long-term average ratio to pick  $P$  waves. Numerous variants of this general scheme have been implemented. For example, Bear & Kradolfer (1987) used an envelope function for each signal trace in this algorithm



and then passed it through a non-linear amplifier. The resulting signal is then subjected to a statistical analysis to yield *P*-wave arrival times and a measure of reliability for the picking. Pisarenko, Kushnir & Savin (1987) developed an optimal *P*-wave picker using an asymptotic approximation of the likelihood function. Roberts, Christoffersson & Cassidy (1989) made an assessment of whether data are consistent with the arrival of a *P* wave or linearly polarized *S* wave using the auto- and cross-correlations of three-component data. Takanami & Kitagawa (1988, 1993) developed a method for *P* and *S* waves by fitting a locally stationary autoregressive model. Kracke (1993) developed a simple method based on the displacement vector of a seismic trace in a spherical coordinate system for *P* waves. Cichowicz (1993) developed an *S*-wave picker based on a filter which combines polarization and energy ratios. Joswig & Schulte-Theis (1993) used a master-event-correlation method to detect *P*-wave arrivals in weak local earthquake records. All of these are traditional programs: they quantify some attribute of the seismic trace and use this as the basis of the decision. These algorithms are not adaptive, working well under certain conditions, but quite often not producing good results. Analysts are still required to interactively check the quality of the result.

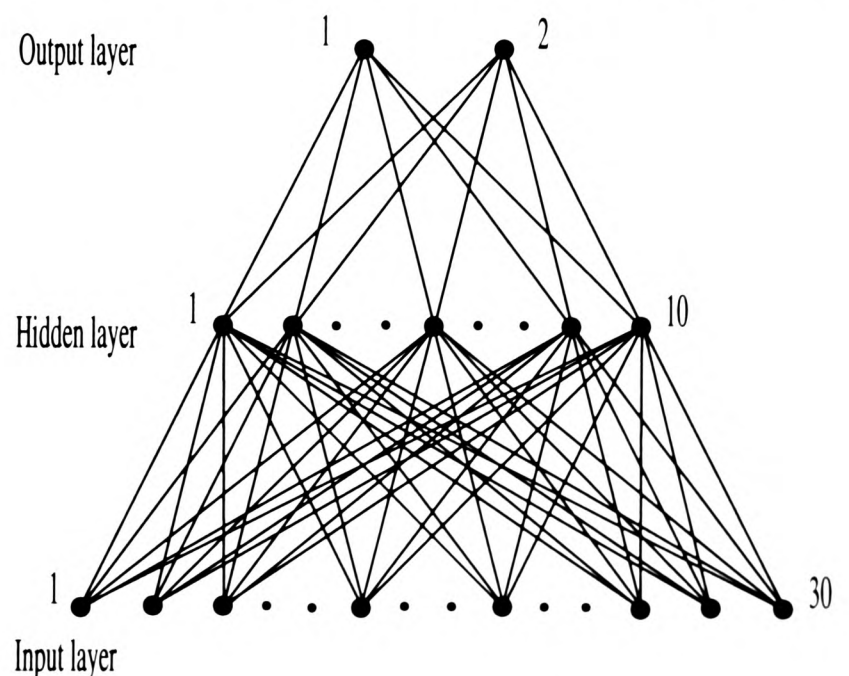
The tasks performed by the trained analyst in manually picking arrival times involve an intensive amount of pattern recognition. Experience provides a judicial balancing of wave characteristics such as amplitude, frequency and polarization from previous records at the same station to determine the most likely onset time. If questioned about a particular decision, however, the analyst may offer a few rules for guidance but can often give no obvious systematic reasoning because the decision has been partly subjective. This reasoning is based upon past experience. Consequently, different trained analysts give different answers, and the same analyst may choose a different interpretation after some time has elapsed. With the increase in the number of digital seismic networks being established world-wide, there is a pressing need to provide a more reliable and robust alternative, which is less time-consuming and more objective. The application of artificial intelligence methods to earthquake analysis is a relatively recent development which attempts to tackle these objectives. Various methods have been applied to the interpretation of seismic signals from a local seismic network such as knowledge-based systems according to the blackboard method used by Chiaruttini, Roberto & Saitta (1989), and later developed by Chiaruttini & Salemi (1993). Bache *et al.* (1990) developed an intelligent monitoring system based on this approach, data-base management systems and signal processing. As an alternative strategy Joswig (1990) developed a pattern recognition approach for *P* waves using a sonogram. Klumpen & Joswig (1993) used a pattern recognition technique to identify generic polarization patterns to estimate *P*-wave and *S*-wave onset times.

Artificial neural networks (ANNs), another group of techniques from the area of artificial intelligence, provide a natural alternative to this type of earthquake analysis as they have proven useful at handling complicated pattern recognition problems in other applications. ANNs have been used to solve a diversity of geological and geophysical problems. For example, Dystart & Pulli (1990) use them for

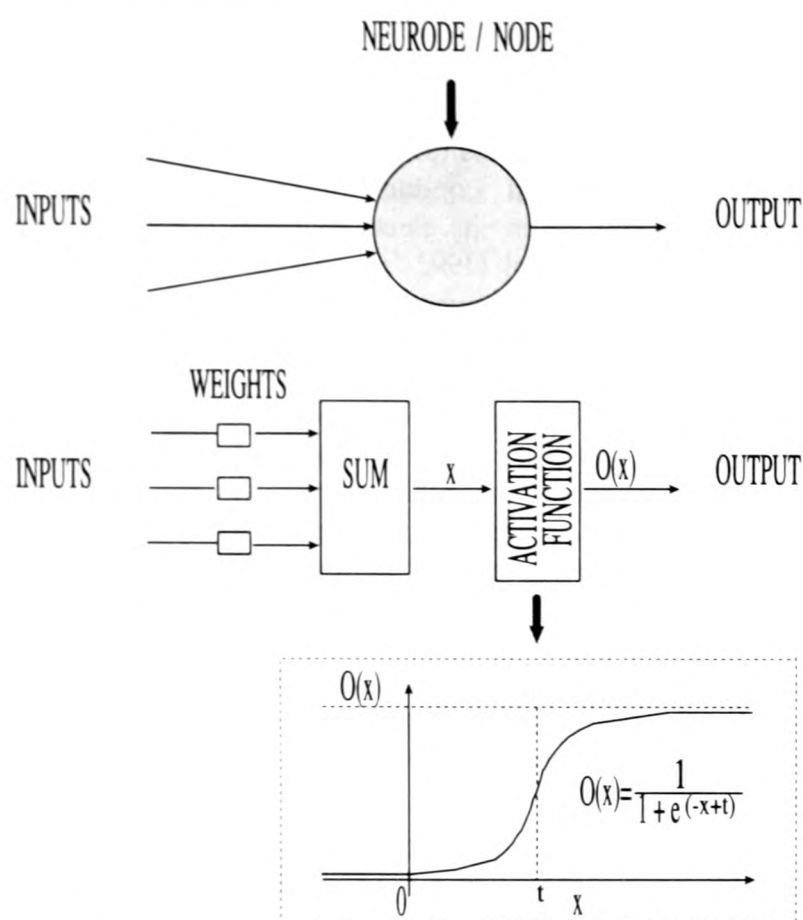
the problem of automatic event classification. McCormack (1991) uses an ANN to combine synthetic spontaneous potential (SP) and resistivity logs to estimate lithology logs. Poulton, Sternberg & Glass (1992) use an ANN to estimate the offset, depth and conductivity-area product of a conductive target given an electromagnetic image of the target. Wang & Mendel (1992) use a Hopfield network to implement an adaptive minimum prediction-error deconvolution. The ANN also has been used in the first-break picking of surface seismic data (Murat & Rudman 1992; McCormack, Zauha & Dushek 1993). The wide range of applications emphasizes the particular strength of this approach over traditional methods incorporating a fixed algorithm to solve a particular problem, as the ANNs utilize a learning scheme to develop an appropriate solution so that the network is flexible and adaptive to different data sets. In our application, the ANN may be likened to an analyst and is trained by presenting it with many different earthquake records. After training is accomplished, the ANN should be able to recognize new arrivals from a variety of new seismograms.

## 2 THE THEORY OF ARTIFICIAL NEURAL NETWORKS (ANNs)

The ANN is designed to simulate the neural connections of the human brain. It is made up of sets of nodes arranged in layers, consisting of an input layer, one or more intermediate hidden layers and an output layer. Fig. 1 schematically illustrates the architecture of such a neural network. Each node, the basic processing unit of the neural network, is usually drawn as a solid circle. Fig. 2 shows the components of this node, and what it represents. The outputs of the nodes in one layer are transmitted to nodes in another layer through links called 'weights'. These weights are real numbers, which are applied as simple multiplicative scalars, and effectively amplify or attenuate the signals. With



**Figure 1.** The neural network structure for picking arrivals on local earthquake records. Solid circles represent nodes, and straight lines represent weights. This is a three-layered neural network. The segment of vector modulus is input according to a time-series order. The output nodes give the results which are (1, 0) or (0, 1) for *P* wave or noise during training and are defined by the  $F(t)$  during testing.



**Figure 2.** Schematic showing various constituents of a node in the neural network of Fig. 1.  $X$  represents summed input to the node,  $t$  the threshold, and  $O(t)$  is the output. The node input is the sum of the weighted output of nodes in the previous layer and the node is then activated in accordance with the summed input using a pre-set function (usually a sigmoid function) and its threshold. All weights and thresholds are determined during the learning procedure.

the exception of the nodes in the input layer, the net input to each node is the sum of the weighted outputs of nodes in the previous layer. Each node is then activated in accordance with the summed input using a pre-set activation function (usually a sigmoidal function), and a threshold parameter for the function. In the input layer, the net inputs to each node are the components of the input pattern.

The ANN used in this present study is non-linear, multilayer, feed-forward and back-propagation of error (Rumelhart, Hinton & Williams 1986). This means that the activation function is non-linear, there are many layers in the network, and the signals feed through the network only in a forward direction. This is the most popular type of ANN in use today as it is well understood. It also incorporates a back-propagation learning algorithm, or *Delta Rule* which is usually used to train this type of neural network—a good mathematical summary is given by Pao (1988). This method attempts to find the most suitable solution (numerical values of weights and thresholds) for a global minimum in the mismatch between the desired output pattern and its actual value for all of the training examples. The degree of mismatch for each input–output pair is quantified by solving for unknown parameters between the hidden layer and output layer and then by propagating the mismatch backwards through the network to adjust the parameters between the input layer and hidden layer. In this learning procedure, the first pattern is presented as input to a randomly initialized network, and these weights

and thresholds are then adjusted in all the links. Other patterns are then presented in succession, and the weights and thresholds adjusted from the previously determined values. This process continues until all patterns in the training set are exhausted (*an iteration*). The final solution is generally accepted to be independent of the order in which the example patterns are presented. A final check can be performed by looking at the *pattern error*, which is defined as the square of the mismatch between desired and actual output for each pattern, and the *system error*, which is defined as the average of all of these pattern errors, to determine whether the final network solution satisfies all the patterns presented to it within a certain threshold error. The set of weights and thresholds in the network are now specifically tailored to ‘remember’ each input and output pattern, and can consequently be used to recognize or generate new patterns given an unknown input. The network is now trained, and can be used in subsequent analyses.

### 3 DETECTION AND PICKING OF SEISMIC ARRIVALS USING AN ANN

#### 3.1 Current objective

In order to present our current objective, it is necessary to distinguish between the following:

(1) *Arrival detection*—specification of an arrival time close to which an arrival may be bracketed within a pre-defined time window.

(2) *Arrival picking*—reliable and accurate estimation of the onset time of a definite seismic arrival. An arrival in our case is seismic motion defined by a wavelet for which the character resembles the training set wavelets.

(3) *Arrival identification*—this classifies individual arrivals into categories relating to their polarization (not necessarily linear) and to the nature of their propagation, for example *Pg*, *Sg* and *Lg* waves.

(4) *An event*—a transient seismic signal generated by a phenomenon such as an earthquake, quarry blast, sonic boom or underwater explosion, which is recorded as a time sequence. The event possesses a fine structure given by a definite hierarchy of arrivals, which are of importance in defining the event type and the Earth’s structure.

(5) *A false alarm*—a spurious signal of non-seismic character or a disturbance sufficiently different in statistical character from an event that it cannot be readily utilized in defining the Earth’s structure. Examples include electrical spikes, and continuous traffic noise.

(6) *An event window*—a time sequence whose endpoints bracket a seismic event of interest. This window is usually obtained through use of a triggered seismic network, and may contain many possible false alarms in addition to the main event. Determination of this feature is a robust and inherently stable operation. Arrival picking, which requires more resolution with a concomitant increase in sensitivity, is usually preceded by this stage.

(7) *Signal-to-noise ratio (SNR)*—the ratio between maximum vector amplitude of signal and quiescent period immediately before the arrival onset. For our purposes the noise level is evaluated within a window of 290 ms,



coinciding with the length of the input segment to the neural network.

In this paper we use the ANN for arrival detection and picking for  $P$  and  $S$  waves in local events, with the intention of constructing a hierarchical scheme of analysis in which (1) and (2) act as the basis for stage (3), and subsequently is used to define (4), (5) and (6). Hence, for the purposes of this work we do not consider the concept of an event window, and all examples are presented without consideration of false alarms. The distinction between a false alarm and an event is considered as a secondary objective in later work.

### 3.2 Input characteristics of data

The detection of the different arrivals is accomplished using the vector modulus of the three-component motion. This is useful as the recorded signal is strongly dependent on the source position and ray direction, which may otherwise give rise to a misleading interpretation. The instantaneous vector modulus  $M(t)$ , calculated at each individual three-component sample along the traces, separates the geometric dependency from our recorded vector motion whilst retaining the character of the seismogram for picking.  $M(t)$  is then used as direct input for the ANN. It is believed that this attribute facilitates an easier identification, regardless of the polarization of the wave (Lomax & Michelini 1988). We do not use the polarization properties of individual arrivals as we believe it may not provide a satisfactory indicator due to such factors as phase changes during propagation, fine structure of the waveforms such as that due to shear-wave splitting, and directional dependency. The overriding concern is the uncertain applicability of such a parametric model of the wavefield in a heterogeneous crust (Der, Baumgardt & Shumway 1993).

$M(t)$  is presented to the network in segments which are selected from a sliding window which passes across the entire three-component seismogram. Each segment is individually normalized so that it is not dependent upon the magnitude and the distance of an earthquake, as otherwise it may bias the estimates with large changes. This reduces the number of training examples, which would otherwise have to cover a range of magnitudes and distances required, with a consequent increase in the training time and a larger network structure. This means that the network is forced to sense the relative amplitude and frequency content of the signals, and uses this information to detect the onset. For a high signal-to-noise ratio (SNR), the onset is characterized by a distinct change in the amplitude of the seismic activity. However, if the SNR is low, the major discriminating factor is a frequency change due to the different spectra of the background signal and earthquake signal. The network is similar to a sophisticated wavelet transform.

The data we use are local earthquake events recorded at stations DP and AY on the TDP-H1 seismic network (Crampin, Evans & Ucer 1985; Lovell 1989) between 1984 April and 1984 December. Several hundred local earthquakes are recorded on three-component seismometers at a 10 ms sampling interval. These recordings are not continuous and are triggered by a digital system (Evans *et al.* 1987). All are local, with depths from 2 km to 14 km and

epicentral distances less than 30 km; most are closer to station DP than to station AY. For these local events, we identified predominant  $Pg$  and  $Sg$  waves in the seismogram records. Most events have magnitudes ( $M_L$ ) between  $-0.3$  and  $1.0$ , and possess a wide distribution of SNRs which are shown in Figs 3 and 4 for the complete data set. All SNRs lie between 1 and 200, with station DP being of higher fidelity than station AY.

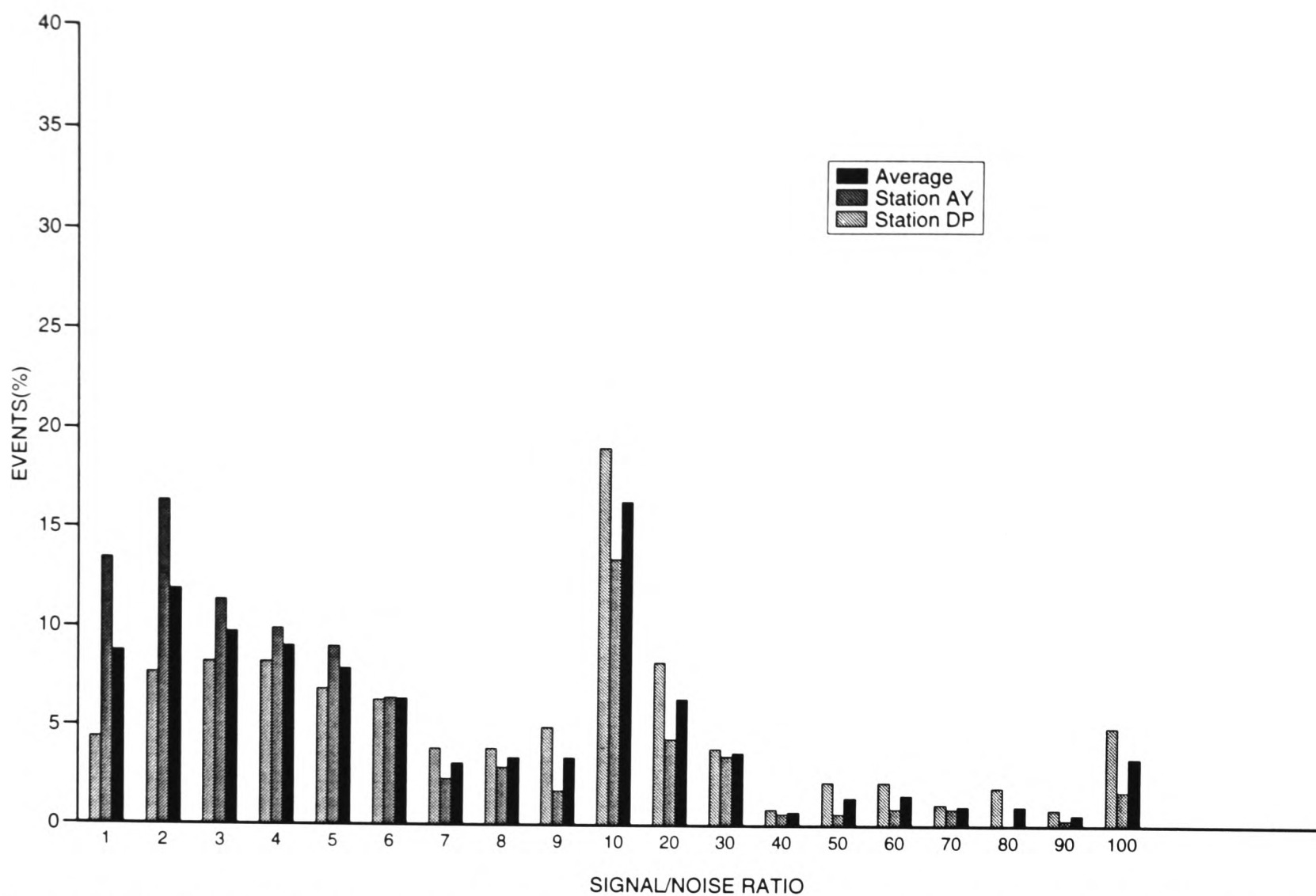
For the first test of the performance of the ANN we will apply it to a small subset of these data. The data set consists of 210 high-quality events recorded at stations DP (120 events) and AY (90 events). In the second test, we will use this trained neural network to process the complete data set of 877 recordings at stations DP and AY on the TDP-H1 network, with a mixture of good and bad data.

### 3.3 Neural network structure

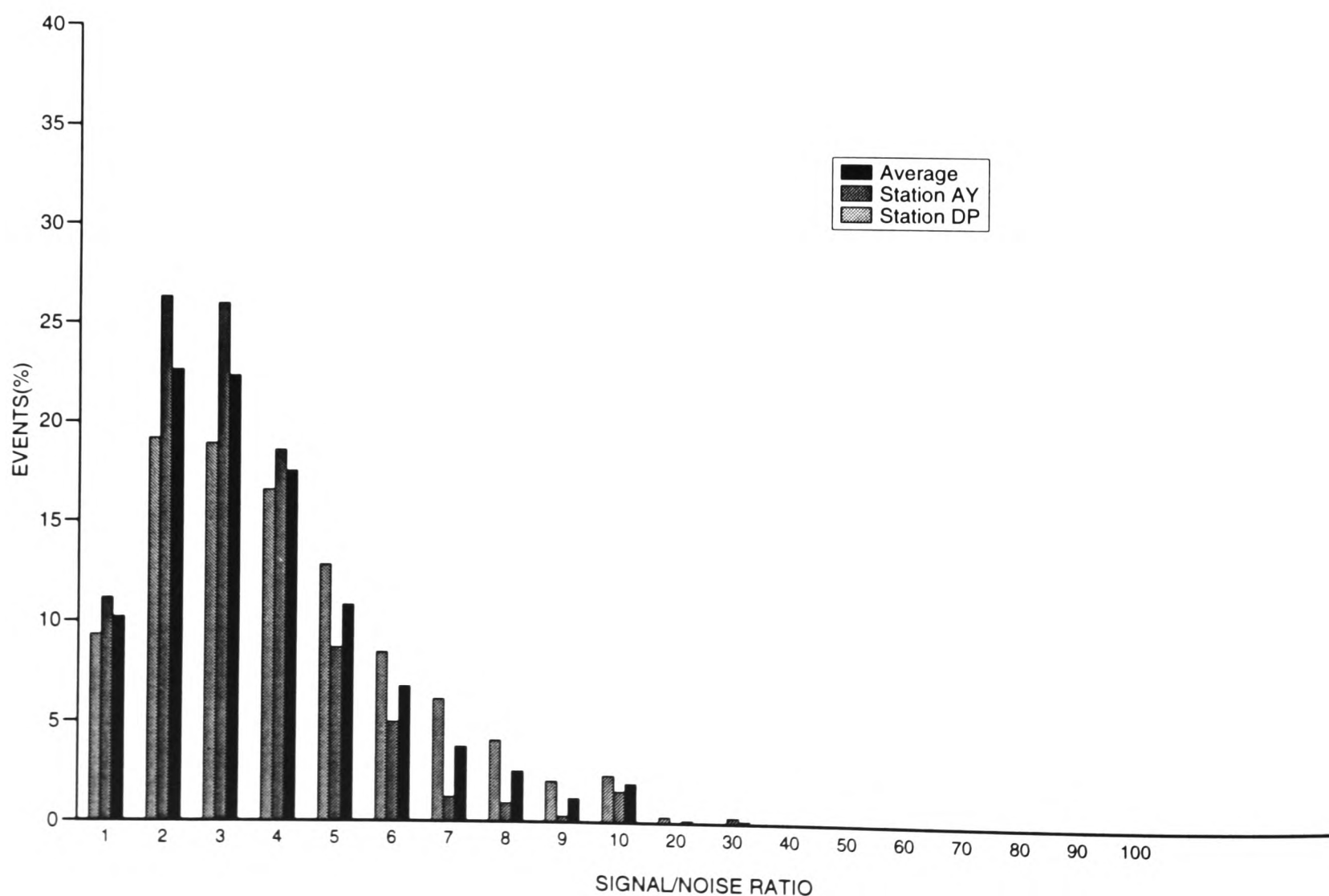
A sliding window length is fixed at 290 ms (30 samples), and is chosen to include several complete cycles of the waves, giving 30 input nodes. There are two nodes in the output layer to flag the result: the output is (1, 0) for an arrival; and (0, 1) for pure noise. The number of hidden nodes depends on various factors such as input nodes, output nodes, system error, pattern error and training samples. There is no fixed generic relationship between the number and these factors for this type of network. However, we do know that in network learning, generalization is increased and memory is reduced by limiting the number of hidden nodes (Dowla, Taylor & Anderson 1990). Too few hidden nodes will lead to a long learning process or no convergence. In this case we finally chose 10 hidden nodes after a process of trial and error with different training runs. 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.

### 3.4 Training procedure

A small number of recordings are used to train the network, and the remainder 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  $M(t)$  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. We do not include  $S$  waves in the training data set because  $M(t)$  for these displays appears to exhibit a similar character. The function of this primary picker is to flag as many changes of  $M(t)$  as possible and discard those which are neither  $P$  nor  $S$  arrivals. The segments are arranged so that the predicted onset time of every signal lies at the tenth sample, for which the network output flags (1, 0). This behaviour is imprinted on every training example. Fig. 5 shows the nine  $P$ -wave and nine background signal training segments used in the study. In the first experiment, we use seven training segments of  $P$

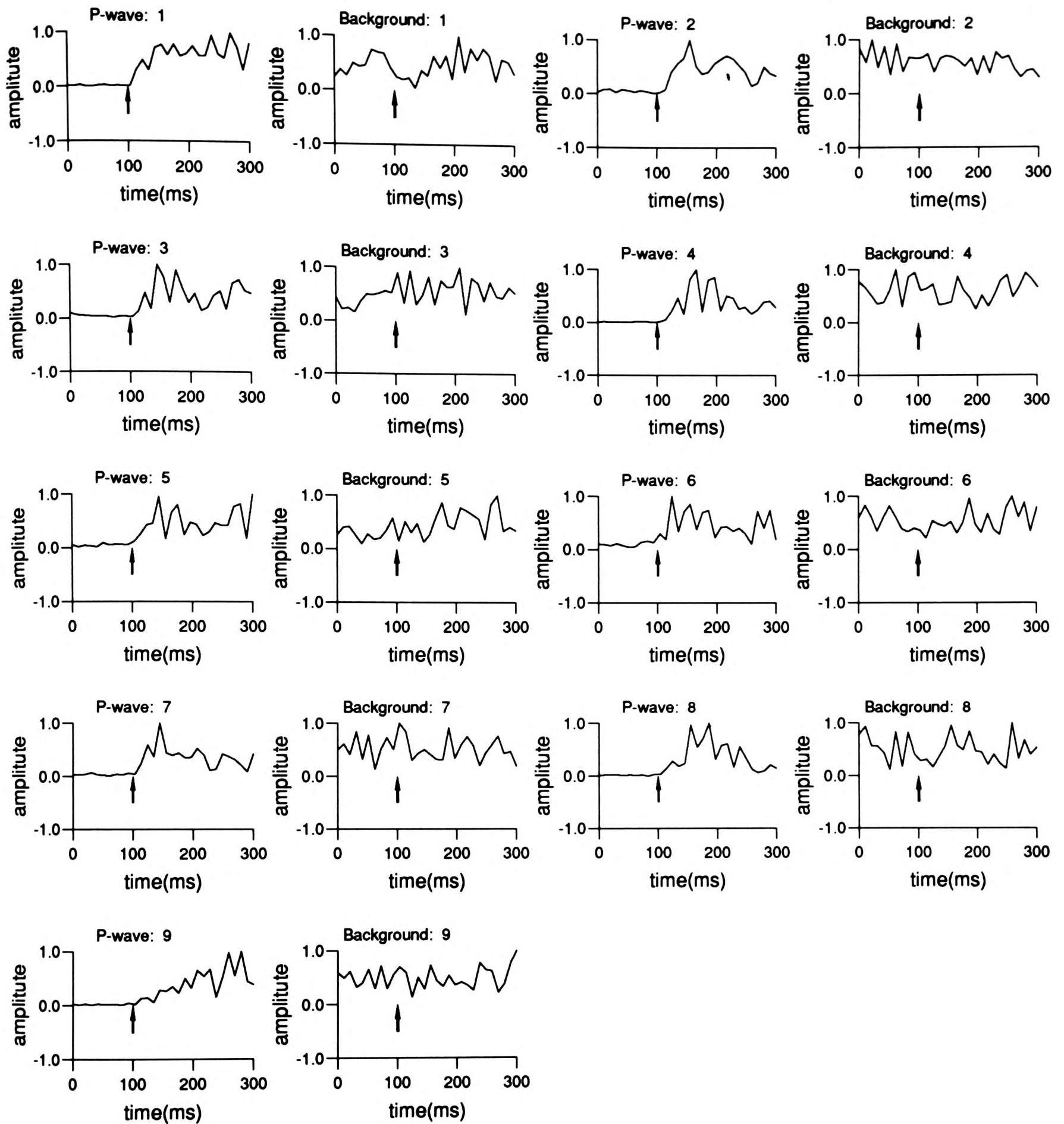


**Figure 3.** The distribution of signal-to-noise ratio (SNR) of *P* waves manually picked in the complete data set which includes 877 recordings. SNR is defined as the ratio of the  $M(t)$  maxima before and after the onset time.



**Figure 4.** As Fig. 3 but for the *S* waves manually picked in the complete data set which includes 877 recordings.





**Figure 5.** Nine *P*-wave segments and nine noise segments used for training the network. Noise segments are extracted prior to the *P*-wave arrival in the same seismogram. Arrows on *P*-wave segments indicate arrival times used to train the network, all are at the tenth sample. These segments are individually normalized before being input into the neural network.

arrivals and noise (Fig. 5). The training procedure takes 498 iterations (less than one minute of CPU time on a VAX4000). The system error reached is  $2.5 \times 10^{-5}$ , with a pattern error of  $10^{-4}$ . After training, the neural network is ready to pick the waves.

### 3.5 Arrival detection

To detect or pick an arrival we first calculate the observed  $M(t)$ , take each windowed segment of this, and then feed it into the trained neural network. We shift the window by one

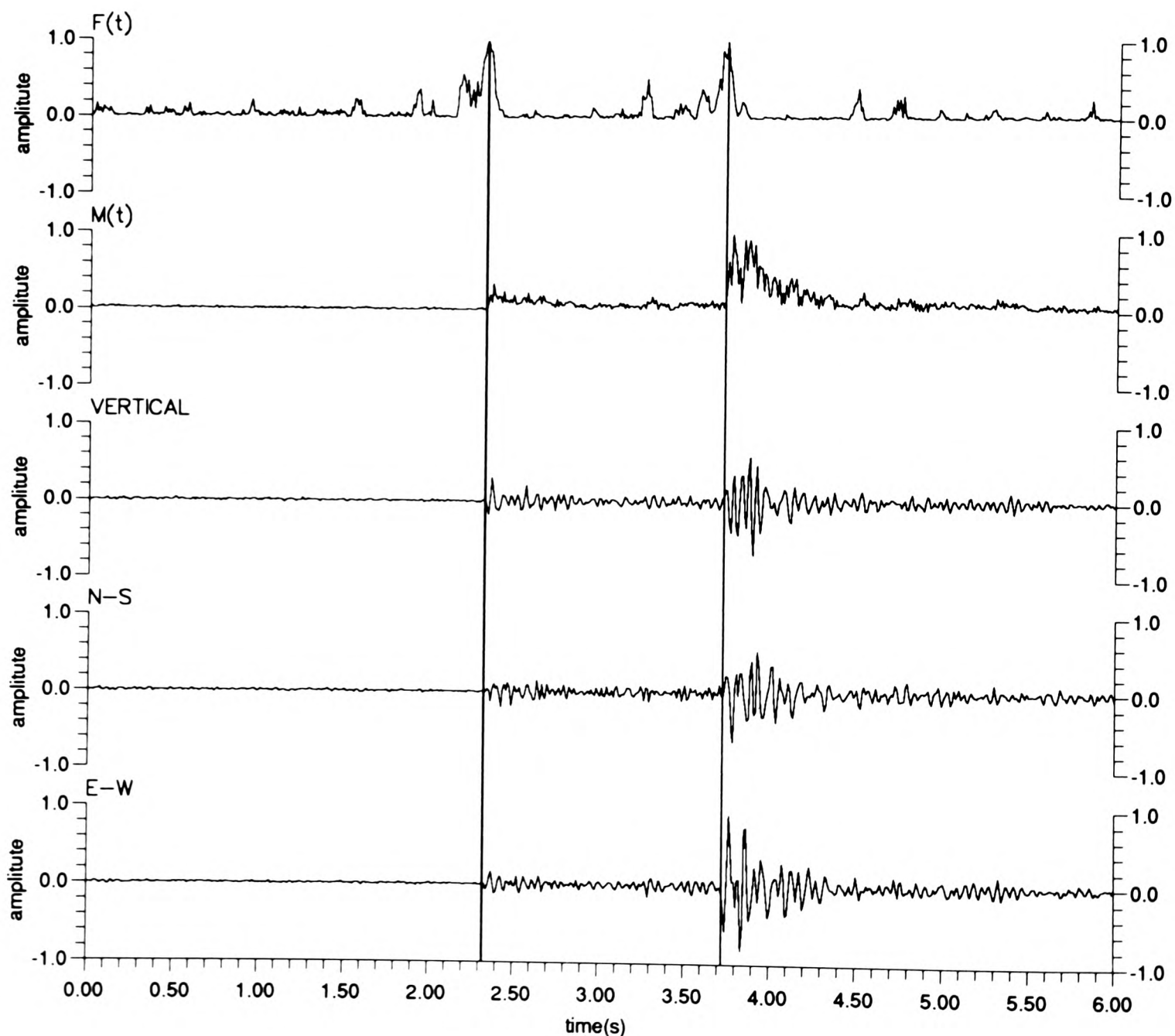
sample at a time and feed each segment into the network, storing the output. The procedure is repeated until the end of the seismogram is reached. In general, the output ( $o_1(t)$ ,  $o_2(t)$ ) lies between the ideal for a signal or for a noise (for example, (0.8, 0.2) or (0.4, 0.6)). To provide a single indication of the onset, we use a function  $F(t)$  which highlights the difference between the actual output and ideal noise:

$$F(t) = \frac{1}{2}[(o_1(t))^2 + (1 - o_2(t))^2]. \quad (1)$$

Figure 6 shows an example of  $F(t)$  for one of the data segments in our chosen data set. The peaks in  $F(t)$

correspond to abrupt changes in  $M(t)$ , 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 are two large peaks corresponding to  $P$  and  $S$  waves. The positions of their maxima occur exactly at the manually chosen onset times of the  $P$  and  $S$  waves. 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. With a threshold of 0.6 it is possible to detect 200  $P$ -wave arrivals (95.2 per cent) and 184  $S$ -wave arrivals (87.6 per cent) from 210 sets of three-component data. If we decrease this

Station: DP  
Date: 1984-07-04  
Start-time: 00h16m19s  
Scale: 546



**Figure 6.** Three-component vector modulus  $M(t)$  and  $F(t)$  function computed from output of the neural network. The vertical lines are automatically picked by the network, and indicate exact picks without error. The maxima correspond to the  $P$  and  $S$  waves, although there are some small maxima in  $F(t)$  prior to the main arrivals which show small changes in  $M(t)$ . There is a small precursor before each main peak in  $F(t)$  which disappears in better trained networks.

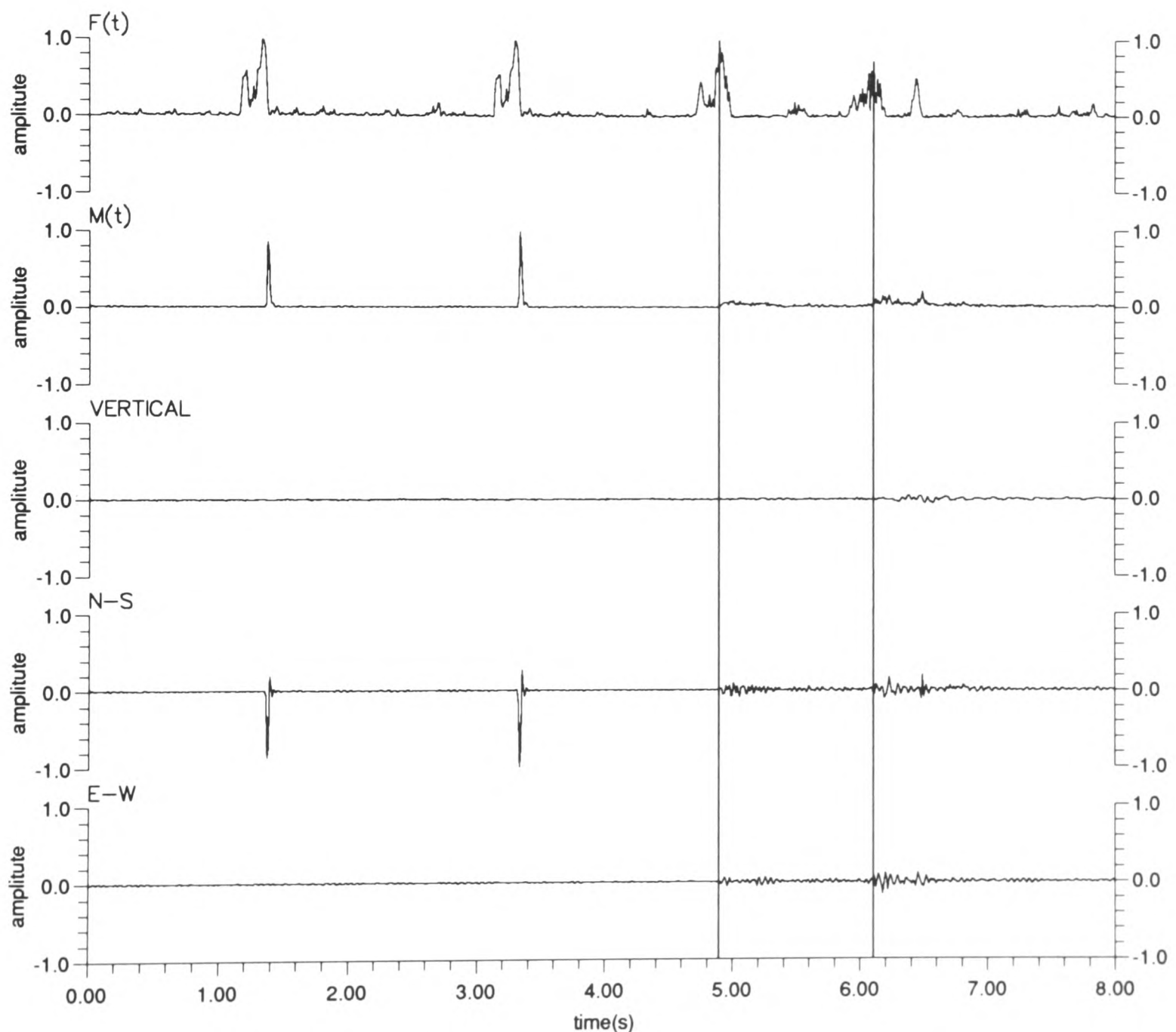
threshold to 0.5, 205 *P*-wave arrivals (97.6 per cent) and 200 *S*-wave arrivals (95.2 per cent) are detected. Given that only seven *P*-waves from station DP were used to train this network, this result is extremely encouraging. It is interesting to note other peaks in this function, which indicate other wave arrivals, spikes or noise bursts. Fig. 7 shows such an example including some spikes in the seismogram. Other techniques are necessary to disregard these signals. If the peaks correspond to spikes which are typically one or two sample points of anomalously large amplitude relative to the background noise, they may be flagged using a conventional algorithm. It is also possible to discriminate false arrivals by examining the peak of  $F(t)$ ,

the arrival being rejected if only one or two points are greater than the threshold and if the  $M(t)$  maximum is less than a pre-specified amount. Otherwise, we confine such discrimination to a secondary stage of our analysis scheme.

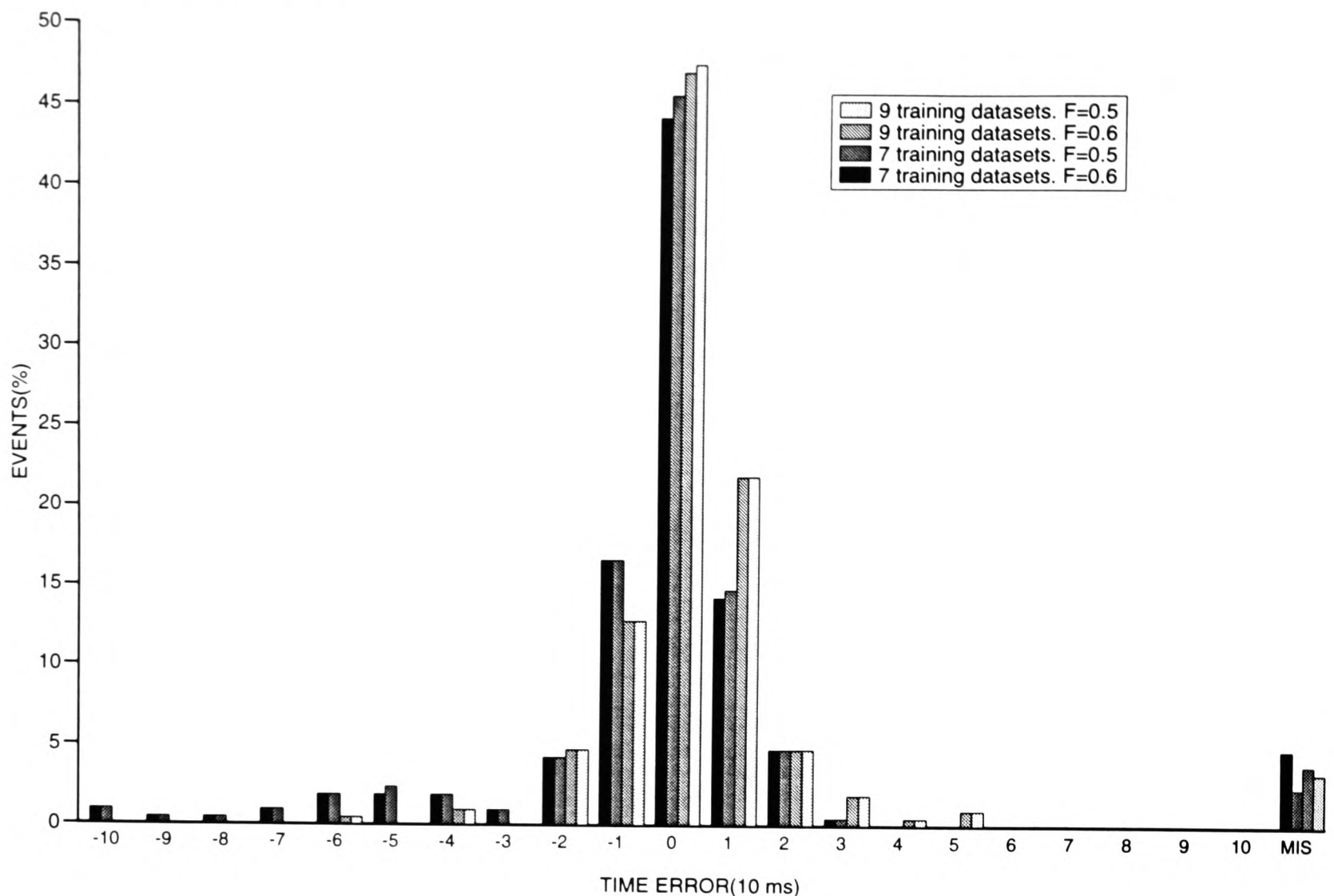
### 3.6 Picking of onset time

As the segments of  $M(t)$  are fed into the trained neural network,  $F(t)$  reaches a maximum when the arrival time is at the tenth point (see Section 3.4). Either side of this maximum,  $M(t)$  is shifted and the network output  $F(t)$  decreases as shown in Figs 6 and 7. This implies that the onset time may be estimated by searching for a local

Station: DP  
Date: 1984-05-17  
Start-time: 12h13m33s  
Scale: 1958



**Figure 7.** Three-component vector modulus  $M(t)$  and  $F(t)$  function computed from output of the neural network. Data have interfering electrical spikes, and  $F(t)$  has some peaks with maxima greater than 0.6 corresponding to them. These spikes may be disregarded by a post-processing routine. *P* and *S* waves are automatically detected by this method.



**Figure 8.** Statistics for *P*-wave picks on a data set of 210 local earthquake records. A negative value indicates a later pick than that given by the visual analysis. MIS refers to all those unpicked arrivals, defined as picks with errors larger than 10 sample increments (100 ms).

maximum after  $F(t)$  exceeds the threshold. The  $F(t)$  maxima exactly indicate the *P*- and *S*-arrival onset times in Figs 6 and 7. Note that each maximum has a small precursor which connects with the main peak, together with a base length equal to the input segment. According to the training procedure, small values of input nodes before the tenth point increase the value of  $F(t)$ . As the arrival onset comes into the input window, the relative values of input nodes before the onset decrease and the  $F(t)$  should increase. For a well trained network, the  $F(t)$  should gradually increase until the onset reaches the tenth point and then the  $F(t)$  should gradually decrease until the onset moves out of the input window. However, for a less well trained network, as the onset moves forward in the window,  $F(t)$  may decrease at some points forming a precursor. We find that as more training data are included, this precursor and the large peak merge into one wide peak (Fig. 14). Figs 8 and 9 show the statistical results of using this method for a network trained with the seven training sets of *P* arrivals and noise (Fig. 5). Here, we can estimate at least 75.2 per cent of the *P* waves and 50.0 per cent of the *S* waves having onset times of less than or equal to one sample increment using a threshold of 0.6, or 77.1 per cent of the *P* waves and 53.8 per cent of the *S* waves with a reduced threshold of 0.5. The onset time is relatively insensitive to the threshold, confirming that it is determined by the local maximum.

Let us now consider the possibility of improving the *P*-wave picks missed by the neural network. The arrivals not picked have no clear first motion, and the change of  $M(t)$  is not visually obvious, with a small corresponding maximum in  $F(t)$ . A more suitable strategy is to retrain the neural network by including this type of data. This approach may be outweighed if too many training data are used, as this increases the training time, and there is the possibility of having to accommodate more subtle variations using a larger neural network structure. To tackle this, we include two extra *P*-wave segments (segments 8 and 9 in Fig. 5) which have different shapes from the other *P*-wave segments and two corresponding noise segments (Fig. 5). The training procedure takes 712 iterations, with a system error of  $2 \times 10^{-5}$  and a maximum pattern error of  $10^{-4}$ . Figs 8 and 9 compare the results for this new trained network, with the previously trained network. The retrained neural network has an improved performance over the previous one. It detects 202 (96.1 per cent) *P* waves and 207 (98.6 per cent) *S* waves using the threshold for  $F(t)$  of 0.6. The estimated onset times are also more accurate, with at least 172 (81.9 per cent) *P* waves and 162 (77.1 per cent) *S* waves (for  $F(t) > 0.6$ ) or 173 (82.3 per cent) *P* waves and 163 (77.6 per cent) *S* waves (for  $F(t) > 0.5$ ) having onset times  $\leq$  one sample increment. Only one *P* wave and three *S* waves have onset times with errors  $\geq$  five sample increments, and only



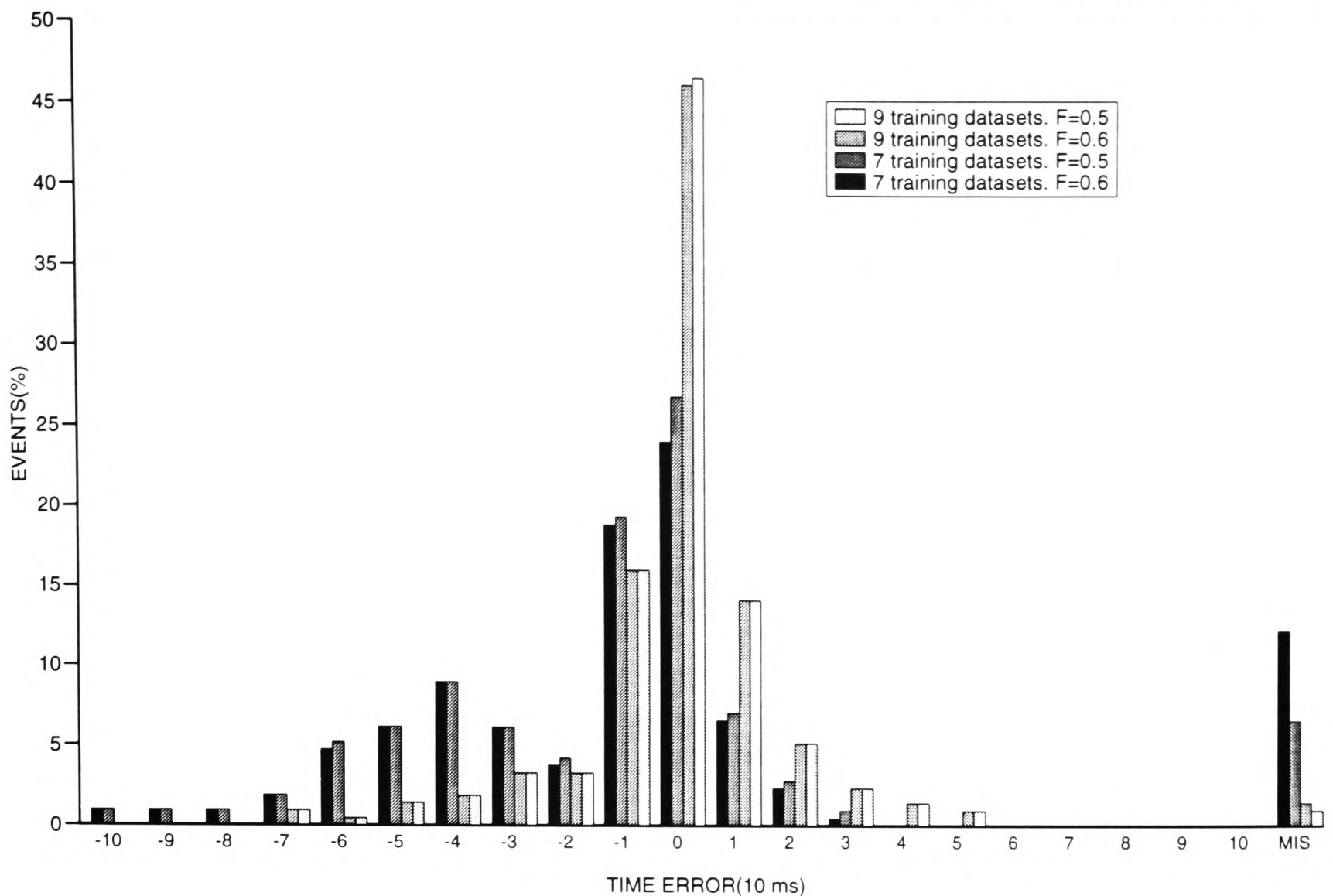


Figure 9. Notation similar to Fig. 8 but for *S*-wave picks.

eight *P* waves and three *S* waves are missed completely. It is interesting to note that only two extra *P* waves are detected but 23 more *S* waves, with a commensurate increase in the picking ability. The *S*-wave picking ability is improved as the additional *P*-wave segments actually resemble many of the *S* waves. Again, decreasing the  $F(t)$  threshold to 0.5 does not significantly improve the picking, with only one more *P* wave and one more *S* wave picked. This comparison substantiates the well-known adaptive behaviour of neural networks, that improvement can always be achieved by judicious choice of the training data sets.

### 3.7 Sensitivity to segment length

The time taken during analysis depends on the neural network structure. The structure can be decreased by decreasing the nodes in the input layer and in the hidden layer. A reduction in the input nodes was tested by reducing from 30 to 20 nodes, keeping the hidden nodes and output nodes the same. We use the nine pairs of *P*-wave and noise segments to train this network. In this case, the training procedure is slower, and it took 2018 iterations to reach a satisfactory convergence point. The results are now worse than before, although the number of unpicked *P* waves remains the same (eight sets of records), with the number of unpicked *S* waves increasing from 3 to 14 events. The onset estimation is worse, with only 57.6 per cent of the *P* waves and 55.2 per cent of the *S* waves having estimates with errors  $\leq$  one sample increment. There is also a larger

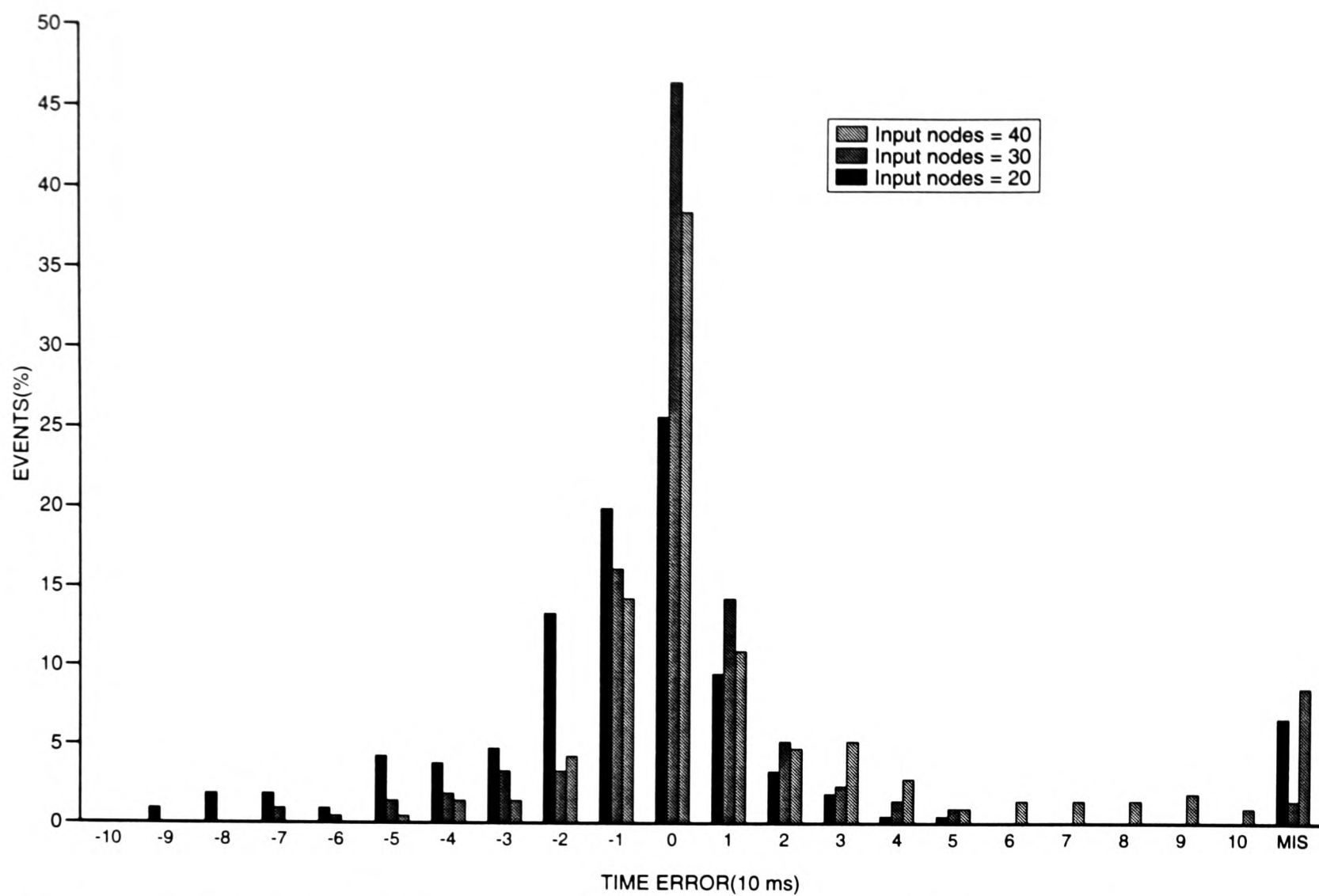
number of spurious picks. Figs 10 and 11 summarize the comparison for this case and the original. The input nodes are also increased to 40 for comparison, with the network now taking 362 iterations to converge. Now, only 67.8 per cent of *P* waves and 63.8 per cent of *S* waves are picked with an error  $\leq$  one sample increment (Figs 10 and 11). There is a larger number of unpicked arrivals.

We should point out that the input segment length depends on the characteristics of the signals. It is suggested that this segment should include several complete cycles of a wavelet. It appears that reducing or increasing the number of input nodes dramatically affects the performance and there appears to be an optimum number of input nodes for our particular configuration. This reflects the general observation that network architecture must be specifically tailored to individual applications. This represents the 'Achilles heel' of ANN applications, and further optimization is required to adapt to particular event types.

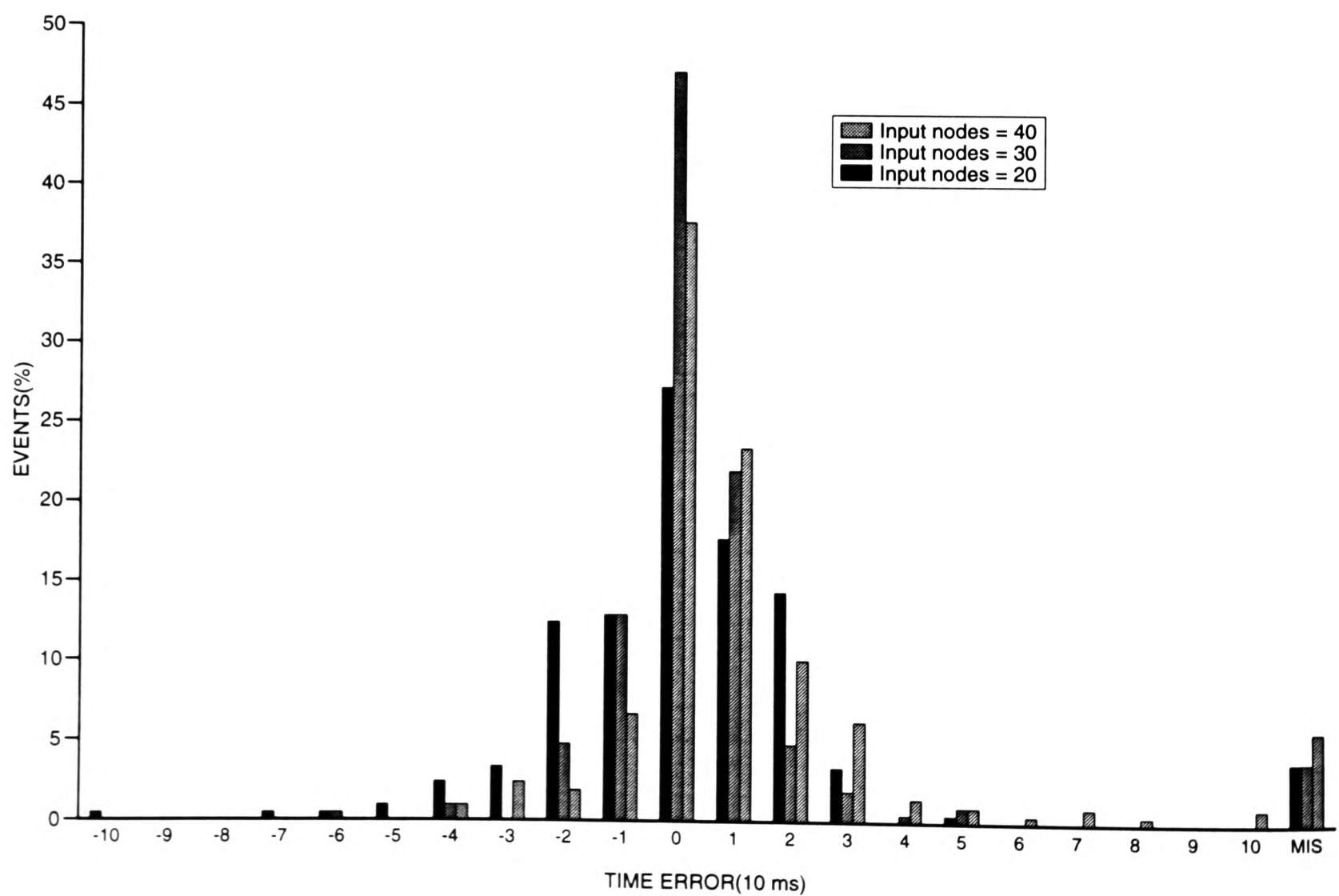
## 4 APPLICATION TO COMPLETE DATA SET WITH A RANGE OF SEISMOGRAM QUALITY

### 4.1 Data character and adaptation of ANN processing

Here, we test our neural network further by incorporating additional recordings from stations DP and AY, which now form the complete data set of 1754 three-component sets



**Figure 10.** Notation similar to Fig. 8. Comparison of *P*-wave picks for three neural networks with 20, 30 and 40 input nodes.

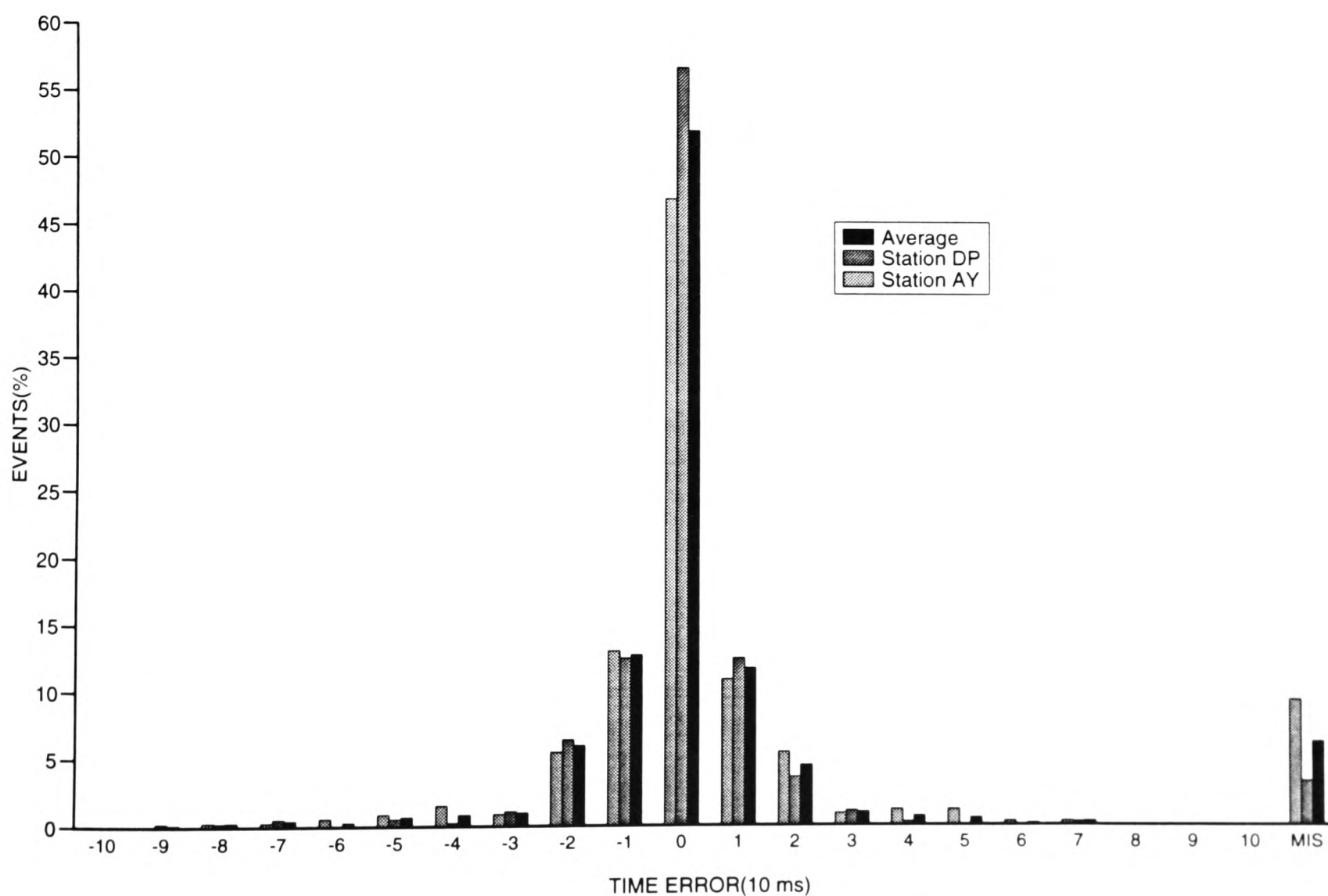


**Figure 11.** Notation similar to Fig. 8. Comparison of *S*-wave picks for three neural networks with 20, 30 and 40 input nodes.

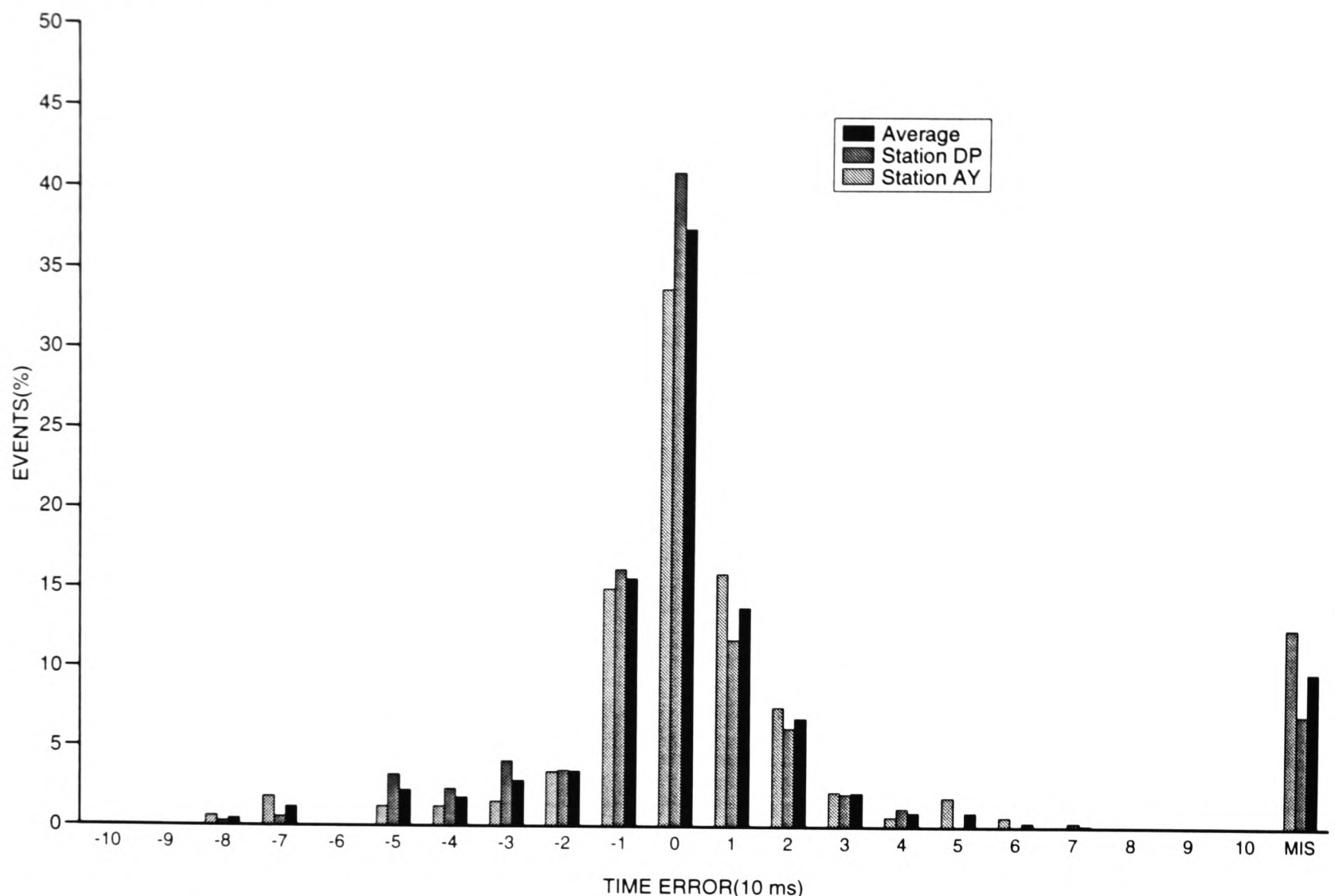
(877 from DP and 877 from AY), from which only the high-quality data were selected in the previous section. In many of these cases the seismometers did not function properly, and either one or two components were inactive or possessed high-amplitude noise so that some of the three-component sets were incomplete. For the recordings in which one or two components do not record seismic signals, the ANN described above can still use  $M(t)$  and analyse them with significant error. The ANN used to accomplish this task has a similar design to the one described above, but uses different threshold parameters for arrival detection. There are also 159 recordings from station DP and 60 recordings from station AY with excessive noise preceding the event or ringing throughout the record. They produce false alarms, and necessitate an additional component to the system to act as a quantity-control procedure. This may be achieved by the use of neural networks, and such a trace editing system is not uncommon; for example, McCormack, Zauha & Dushek (1993) designed an ANN to detect noisy and dead traces in raw surface seismic field data. Here we select the recordings manually, leaving the design of this procedure to a later investigation. Consequently, we must be aware that our statistics will appear more successful than if this procedure had been applied to all the data irrespective of quality. We find that 325 recordings from station DP and 198 recordings from station AY are unusable, leaving 373 and 504

recordings respectively for further processing. These data are reduced further as manual picks for comparison with the ANN results are only possible for 360  $P$  waves and 341  $S$  waves at station DP and 342  $P$  waves and 320  $S$  waves at station AY.

The ANN used in this more extensive test is trained by nine pairs of  $P$  waves and noise segments, and again has 30 input nodes. With an  $F(t)$  threshold of 0.6, the ANN can detect 348  $P$  waves (96.7 per cent) and 317  $S$  waves (93.0 per cent) from Station DP, and 311  $P$  waves (90.9 per cent) and 280  $S$  waves (87.5 per cent) from station AY. Most of the failures arise at the low SNRs of between 1 and 3. The method appears to pick all phases with an SNR > 3. Figs 12 and 13 show the estimation results. For station AY, 234 (68.6 per cent)  $P$  waves and 207 (64.7 per cent)  $S$  waves have onset times within one sample increment (10 ms) of the expected manual values. For station DP, 295 (81.9 per cent)  $P$  waves and 234 (68.4 per cent)  $S$  waves have onset times within the same tolerance. In addition, only 7.7 per cent of  $P$  waves and 11.8 per cent of  $S$  waves have onset times with errors greater than five sample increments or are missed entirely. Once picked, the SNR does not affect the accuracy of the estimate. In the situation where not all  $S$  waves picks are manually possible, a valid comparison with the ANN results cannot be made and they are excluded from our statistics. This produces a small bias in our results as it only affects 10 seismogram sets.



**Figure 12.** Statistics for  $P$ -wave picks on the complete data set of 877 local event recordings, with notation similar to Fig. 8. The success of the ANN relative to manual reference picks is quoted as a percentage.



**Figure 13.** As in Fig. 12 but for *S* waves.

## 4.2 Sensitivity to signal-to-noise ratio

One additional benefit in using a trained neural network is its ability to pick waves in low SNR conditions with only a high SNR training data set. Fig. 14 shows an example in which  $F(t)$  has two peaks, indicating a high SNR seismogram set. The peaks are rectangular in shape, with a width of roughly 30 sample points. Fig. 15 shows an example in low SNR, for which the *P*-wave peak appears sharper. Although the three-component particle motion does not display a significant difference between the signal and the noise for this case,  $M(t)$  does indicate the change in the nature of the signal, and this translates to the narrow peak in  $F(t)$  which now indicates the onset. This reveals the possibility of interpreting the shape of these peaks to extend the ability of the trained neural network beyond the boundaries of the training set. Fig. 16 shows a low SNR example in which the *S* wave (SNR = 1.8) can still be automatically picked by the ANN.

## 5 DISCUSSION AND CONCLUSIONS

### 5.1 ANN performance

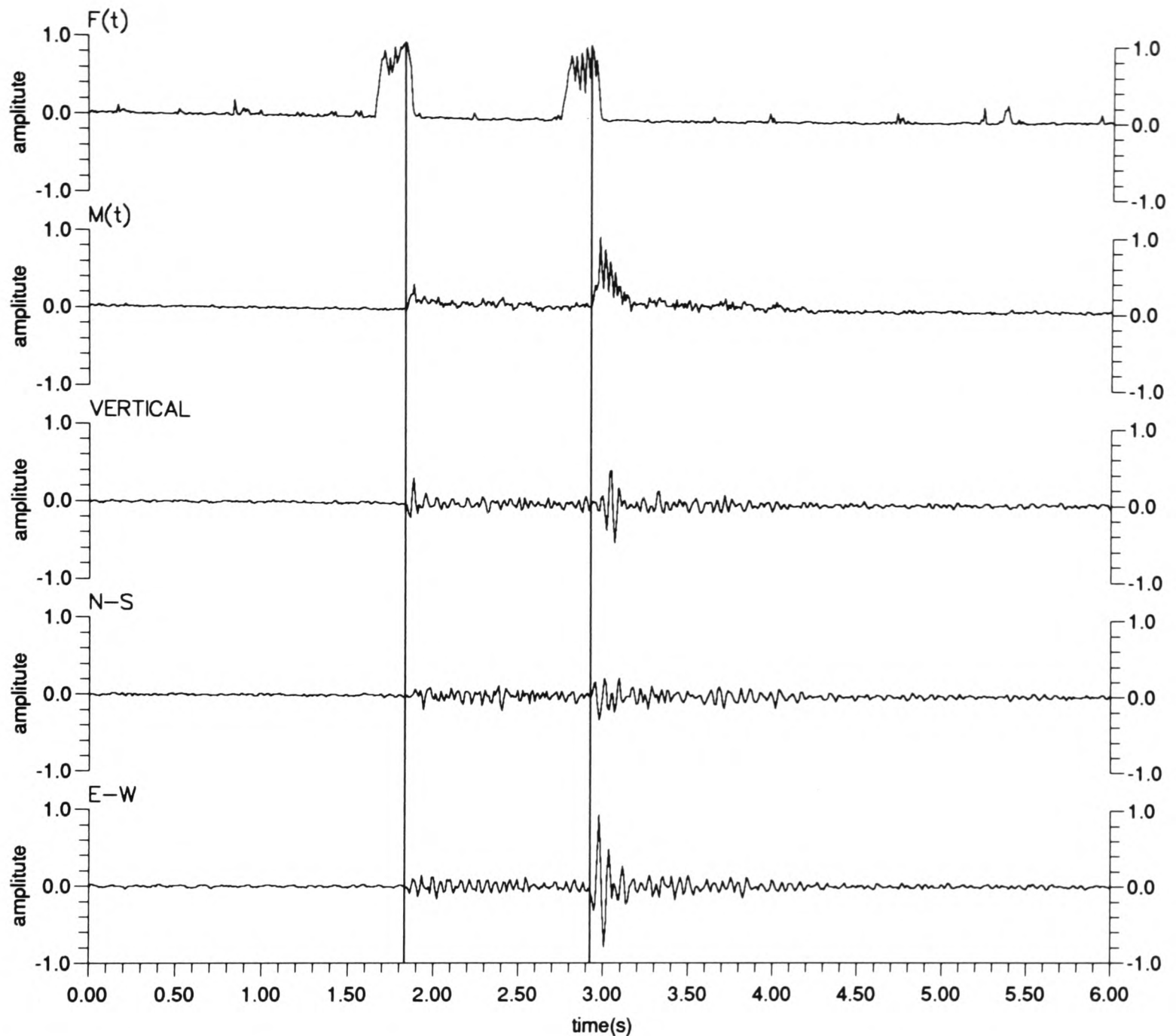
A multilayered neural network is used as a tool to pick *P* and *S* waves from local earthquake data. The neural network input is the vector modulus of each set of three-component records. The results are encouraging, and demonstrate that a neural network trained using a small

subset of the data (only nine *P* waves and commensurate noise segments in this case) can detect 93.9 per cent of the *P* waves and 90.3 per cent of the *S* waves. Using this to further pick the onset times, we find a success rate of 75.8 per cent (with  $F > 0.6$ ) for the *P* waves and 66.7 per cent (with  $F > 0.6$ ) for the *S* waves with an error of one sample increment (10 ms). Although false arrivals and spikes can be discarded by using pre-processing steps, here we chose to include these in the selection, to be discriminated at a later stage. However, a pre-processing stage is required to discard excessively noisy and unusable recordings. They are not included in the statistics which we quote. For the data set in Section 4, 17.4 per cent of the detections include false alarms. Inspection of the seismograms revealed that most of the false arrivals are similar to the *P* or *S* wavelets, and in fact it would not be possible to distinguish them visually if only one segment were available. Additional information is required for this task.

These results, combined with the advantage of not requiring programs to construct special variables and parameters with complicated mathematics, suggest that the ANN is a natural choice for such applications. All that is necessary is to select suitable example arrivals in a training set. The method is adaptive, and training sets can be altered to enhance particular features of different data sets. Adding new training data sets and retraining the network is easy and quick, and can improve the performance of the network. Although the training time can be long, especially as the network architecture becomes involved, once trained the



Station: DP  
 Date: 1984-10-06  
 Start-time: 20h01m42s  
 Scale: 490



**Figure 14.** Example of picking *P* and *S* waves in a high signal-to-noise ratio. Three-component vector modulus  $M(t)$  and  $F(t)$  function computed from output of the neural network trained with nine pairs of *P* waves and noises. Compared with Figs 6 and 7, there are no precursors before the main peaks in  $F(t)$  corresponding to the *P* and *S* arrivals. The peaks have a base length equal to the ANN input segment (290 ms).

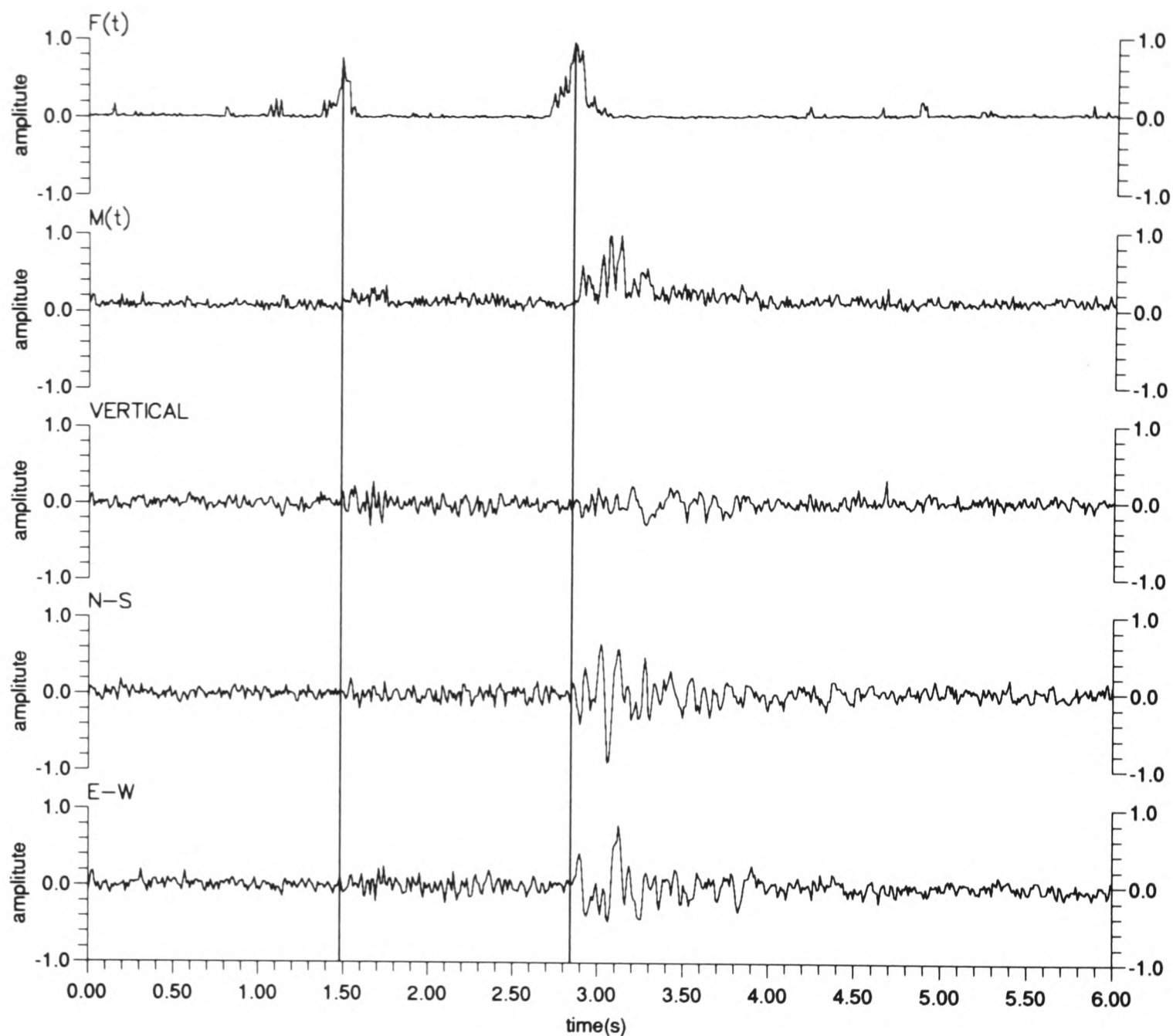
network is sufficiently quick to operate in most real-time applications. However, the network cannot be viewed as all encompassing, as the performance still depends upon the training set and its ability to predict cannot lie too far outside its experience. The exact boundaries of this behaviour have not yet been completely explored. Another limitation is in finding an optimal architecture for a particular application. This is not yet fully understood for multilayered back-propagation networks, although there are certain other network designs where this is possible (Falman & Lebiere 1990; Kusuma & Brown 1992). The

current network design is limited to three-component recordings, but this method has now also been adapted for single-component recordings.

## 5.2 Comparison with other picking algorithms

As discussed earlier, there are many picking algorithms already in use on many seismic networks world-wide. Table 1 gives a comparison of the performance of our technique for local earthquake data and the performance of a few selected techniques in common use. Because articles tend to

Station: AY  
 Date: 1984-05-06  
 Start-time: 21h54m39s  
 Scale: 64

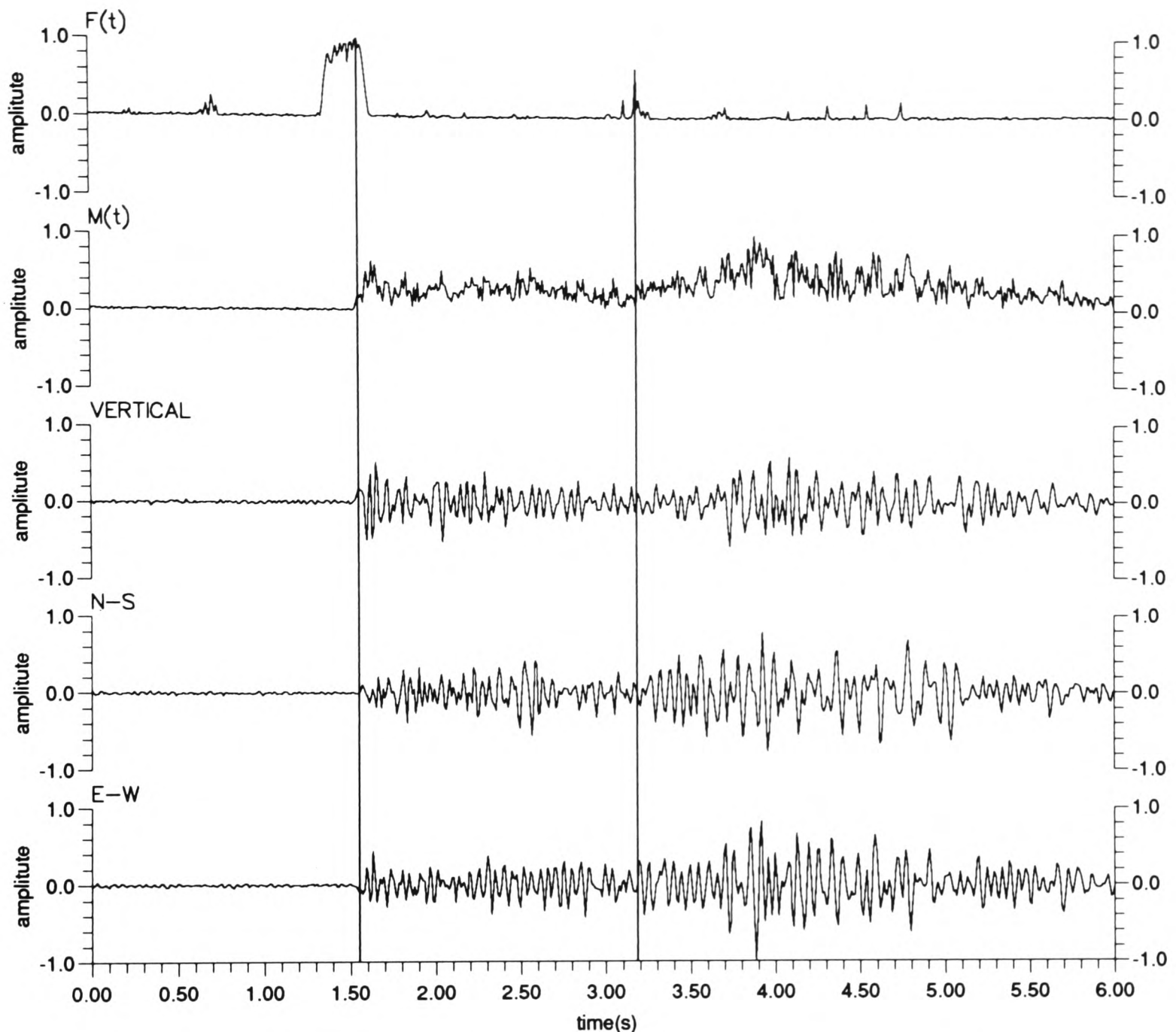


**Figure 15.** Example of picking *P* wave in a low signal-to-noise ratio. Three-component vector modulus  $M(t)$  and  $F(t)$  function computed from output of the neural network trained with nine pairs of *P* waves and noises. Although the three-component particle motion does not display a significant difference between the *P* wave and the noise,  $M(t)$  does indicate the change in the nature of signal, and this translates to a narrow peak in  $F(t)$  which indicates the *P* onset.

describe principles and show a few examples these cannot be directly or wholly compared with our result which is applied to a specific data set of local events, so that this table may not be truly representative of the optimal forms of each technique. As false alarms were not fully treated in our algorithm, we do not suggest without further tests and development that our method is better. However, it does appear that the small estimation error for both *P*-wave and *S*-wave analysis is potentially encouraging for future work. We believe that an additional strength of the neural network

is that it can deal with raw data once it has been trained appropriately. This contrasts with many other techniques, which rely upon pre-processing steps to generate control parameters. The network presented here is relatively quick to train and has been shown to be adaptive to various types of waves. To demonstrate this feature fully, and avoid data-specific results, further tests are to be performed on the network, to extend application to other three-component data sets and single-component data sets. We suggest that it may be possible to process regional or

Station: DP  
 Date: 1984-05-22  
 Start-time: 04h10m09s  
 Scale: 601



**Figure 16.** Example of picking *S* wave in a low signal-to-noise ratio. Three-component vector modulus  $M(t)$  and  $F(t)$  function computed from output of the neural network trained with nine pairs of *P* waves and noises. Due to *P* wave coda it is difficult to indicate the *S* arrival where the SNR is only 1.8. The peak in  $F(t)$  is sharper and its maximum is only 0.64.

teleseismic observations also using this approach. In these cases, however, the network architecture may need to be adjusted to suit the behaviour of these data.

### 5.3 Future work

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. In this analysis we make a concerted effort to develop an approach which avoids any preconceived notions regarding the polarization properties of the individual

arrivals in the earthquake record. This is important as we recognize that a parametric model may not be generally applicable to local and teleseismic events in a heterogeneous crust where the arrivals may be composed of several distinct signal processes (Der *et al.* 1993). Consequently, in future schemes we intend to direct our current approach towards the task of arrival identification using generic polarization characteristics, so that more complex wave trains such as *Sn* and *Lg* may be identified and analysed. Ultimately we hope to integrate other ANN units into a processing flow for record editing and event classification and mechanism determination.



**Table 1.** A summary comparison of selected picking methods.

Author	Input Data	Method	Wave Type	Picking Result	Time Error
Allen (1978)	single trace	STA/LTA	P	60-80%	≤0.05 sec
Bear & Kradolfer (1987)	single trace	modified STA/LTA	P	Local: 65.9% Region: 79.5% Tele: 90%	≤1 sample ≤3 sample
Joswig & Schulte-Theis (1993)	single trace	Master-event correlation	P	80 % for weak events	≤1 sample
Klumpen & Joswig (1993)	3-component recording	generic polarization	P & S	67% for P & S	≤50ms
Kracke (1993)	modulus of 3-component recording	LTA/threshold	P	96.5%	not mentioned
Dai & MacBeth (1994, this paper)	modulus of 3-component recording	neural network	P & S	94% for P 90% for S 75% for P 67% for S	≤100ms ≤100ms ≤10ms ≤10ms

## ACKNOWLEDGMENTS

This research was sponsored by the Global Seismology Research Group (GSRG) at the British Geological Survey (BGS), Edinburgh, and is published with the approval of the Director of the BGS. We thank Chris Browitt, Programmes Director of BGS, David Booth and John Lovell for supplying the earthquake data. We also thank Yun Liu for assisting with the manual picking. Thanks are also extended to the staff and students of GSRG for their support and encouragement with this work.

## REFERENCES

- Allen, R.V., 1978. Automatic earthquake recognition and timing from single trace, *Bull. seism. Soc. Am.*, **68**, 1521-1532.
- Bache, T.C., Bratt, S.R., Wang, J., Fung, R.M., Kobryn, C. & Given, J.W., 1990. The intelligent monitoring system, *Bull. seism. Soc. Am.*, **80**, 1833-1851.
- Bear, M. & Kradolfer, U., 1987. An automatic phase picker for local and teleseismic event, *Bull. seism. Soc. Am.*, **77**, 1437-1445.
- Chiaruttini, C. & Salemi, G., 1993. Artificial intelligence techniques in the analysis of digital seismograms, *Comput. Geosci.*, **19**, 149-156.
- Chiaruttini, C., Roberto, V. & Saitta, F., 1989. Artificial intelligence techniques in seismic signal interpretation, *Geophys. J. Int.*, **98**, 223-232.
- Cichowicz, A., 1993. An automatic S-phase picker, *Bull. seism. Soc. Am.*, **83**, 180-189.
- Crampin, S., Evans, R. & Ucer, S.B., 1985. Analysis of records of local earthquakes: the Turkish Dilatancy Projects (TDP1 and TDP2), *Geophys. J. R. astr. Soc.*, **83**, 1-16.
- Der, Z.A., Baumgardt, D.A. & Shumway, R.H., 1993. The nature of particle motion in regional seismograms and its utilization for phase identification, *Geophys. J. Int.*, **115**, 1012-1024.
- Dowla, F.U., Taylor, S.R. & Anderson, R.W., 1990. Seismic discrimination with artificial neural networks: preliminary results with regional spectral data, *Bull. seism. Soc. Am.*, **80**, 1346-1373.
- Dystart, P.S. & Pulli, J.J., 1990. Regional seismic event classification at the NORESS array: seismological measurements and the use of trained neural networks, *Bull. seism. Soc. Am.*, **80**, 1910-1933.
- Evans, S.A., Beamish, D., Crampin, S. & Ucer, S.B., 1987. The Turkish Dilatancy Project (TDP3): multidisciplinary studies of a potential earthquake source region, *Geophys. J. R. astr. Soc.*, **91**, 265-286.
- Fahlman, S.E. & Lebiere, C., 1990. The cascade-correlation learning architecture, in *Advances in Neural Information Processing Systems 2*, ed. Touretzky, D.S., Morgan Kauffmann.
- Joswig, M., 1990. Pattern recognition for earthquake detection, *Bull. seism. Soc. Am.*, **80**, 170-186.
- Joswig, M. & Schulte-Theis, H., 1993. Master-event correlations of weak local earthquake by dynamic waveform match, *Geophys. J. Int.*, **113**, 562-574.
- Klumpen, E. & Joswig, M., 1993. Automated reevaluation of local earthquake data by application of generic polarization patterns for P- and S-onsets, *Comput. Geosci.*, **19**, 223-231.
- Kracke, D., 1993. A three-component event detector based on waveform analysis, *Comput. Geosci.*, **19**, 117-122.
- Kusuma, T. & Brown, M., 1992. Cascade-correlation learning architecture for first-break picking and automated trace editing, in *Proceedings of SEG 62nd Annual International Meeting*, pp. 10-13, Society of Exploration Geophysics.
- Lomax, A.J. & Michelini, A., 1988. The use of spherical coordinates in the interpretation of seismograms, *Geophys. J.*, **93**, 405-412.
- Lovell, J., 1989. Source parameters of microearthquake swarm in Turkey, *MPhil thesis*, University of Edinburgh, Edinburgh.
- McCormack, M.D., 1991. Neural computing in geophysics, *Geophysics: The Leading Edge of Exploration*, **10**, 11-15, January 1991.
- McCormack, M.D., Zaucha, D.E. & Dushek, D.W., 1993. First-break refraction event picking and seismic data trace editing using neural network, *Geophysics*, **58**, 67-78.
- Murat, M. & Rudman, A., 1992. Automated first arrival picking: A neural network approach, *Geophys. Prospect.*, **40**, 587-604.
- Pao, Y.H., 1988. *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, New York.
- Pisarenko, V.F., Kushnir, A.F. & Savin, I.V., 1987. Statistical adaptive algorithms for estimation of onset moments of seismic phases, *Phys. Earth Planet. Inter.*, **47**, 4-10.
- Poulton, M.M., Sternberg, B.K. & Glass, C.E., 1992. Location of subsurface target in geophysical data using neural networks, *Geophysics*, **57**, 1534-1544.
- Roberts, R.G., Christoffersson, A. & Cassidy, F., 1989. Real-time event detection, phase identification and source location estimation using single station three-component seismic data, *Geophys. J.*, **97**, 471-480.
- Rumelhart, D.E., Hinton, G.E. & Williams, R.J., 1986. Learning internal representation by backpropagating errors, *Nature*, **322**, 533-536.
- Takanami, T. & Kitagawa, G., 1988. A new efficient procedure for the estimation of onset times of seismic waves, *J. Phys. Earth*, **36**, 267-290.
- Takanami, T. & Kitagawa, G., 1993. Multivariate time-series model to estimate the arrival times of S-waves, *Comput. Geosci.*, **19**, 295-301.
- Wang, L.X. & Mendel, J.M., 1992. Adaptive minimum prediction-error deconvolution and source wavelet estimation using neural networks, *Geophysics*, **57**, 670-679.

# Split shear-wave analysis using an artificial neural network?

Hengchang Dai<sup>1,2</sup> and Colin MacBeth<sup>1</sup>

## Introduction

Artificial neural networks (ANNs) are simple models that attempt to simulate the operation of neurons in the brain. Although ANNs are relatively new in seismology, their origins can be traced back to the 1940s when psychologists began developing models of human learning. One of the most exciting developments in ANNs was the advent of the *Perceptron*, the idea that a network of elemental processors arrayed in a manner reminiscent of biological neural networks might be able to learn how to recognize and classify patterns in an autonomous manner. However, in 1969, Marvin Minsky, one of the founding fathers of artificial intelligence, proved mathematically that perceptrons were incapable of solving many simple problems. After over a decade of being in the scientific wilderness, ANNs have once again become a popular tool for many applications requiring algorithms with pattern recognition capability as those mathematical difficulties have been overcome by the introduction of more complex neural network architectures in the 1980s. These new network designs offer increased flexibility and robustness. They are particularly attractive as, unlike conventional methods that incorporate a fixed algorithm to solve a particular problem, ANNs utilize a learning scheme to develop an appropriate general solution, making them flexible and adaptive to different datasets. ANNs are now starting to be used in seismic applications for event picking, or correlation of seismic horizons with sparse well-log data, and are likely to find other applications of great value.

Here, we explore a new area where ANNs may prove of benefit: the recognition of the vector motion of seismic waves. For convenience we will confine ourselves to the identification of the particular patterns of the particle motion recorded in the horizontal plane, which arise from split shear-waves: the interfering fast shear-wave *qS1* and slow shear-wave *qS2*, and investigate whether the pattern recognition capability is sufficiently powerful to be used to extract their characteristic

parameters. Recognition and quantification of these split shear waves in seismic data can provide information about the effective anisotropy of the rockmass from which information regarding the distribution of internal heterogeneity may be determined. This does not restrict the more general relevance of our conclusions to other types of vector particle motion. Our incentive is that the visual analysis of such waves is time-consuming and individual recordings are more difficult to interpret. It is usual to group the recordings according to the acquisition geometry to make the measurements more robust. However, even experienced interpreters may still be subjective and inconsistent in their results and human investigation is still required to inspect the data initially and perform the pre-conditioning. Here, we concentrate on two parameters for estimating this splitting: the polarization direction of the faster shear-wave and the time-delay between split shear-waves. Can the neural network, a natural choice for this type of problem, satisfactorily distinguish between different splitting characteristics, and what parts of the particle motion does it weight in order to do this?

## What are artificial neural networks?

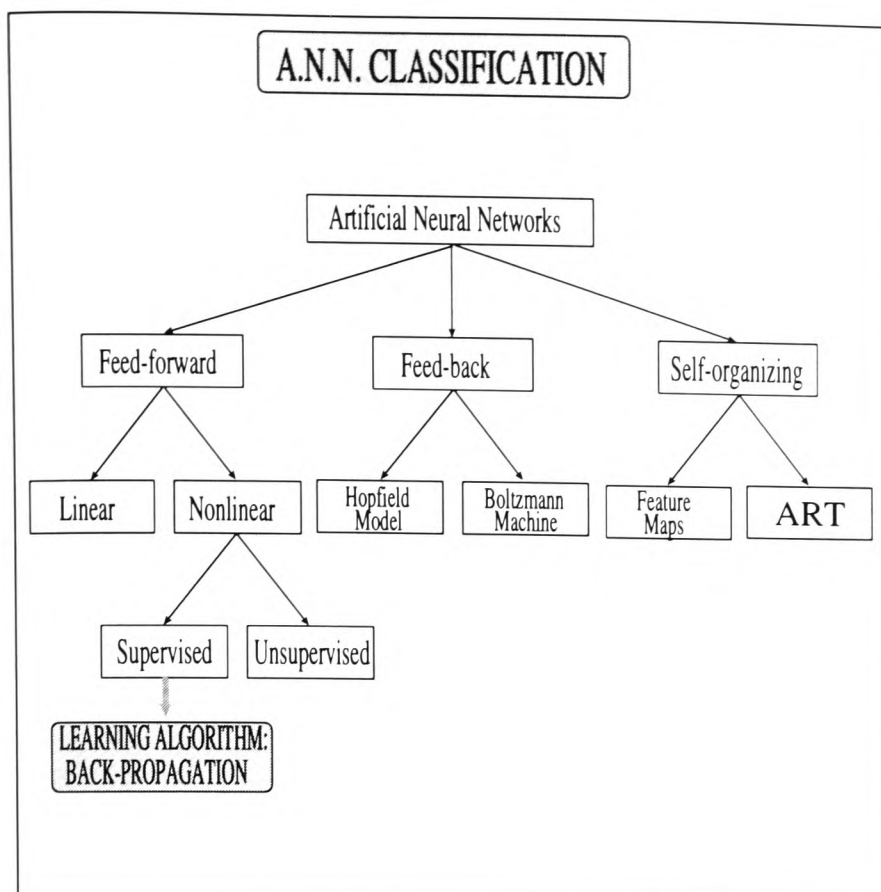
In the brain, a neural cell, or neuron, receives input from many other neurons via interconnections called axons. If the energy level of the combined input exceeds a threshold level, then the neuron transmits an output electrical signal depending on its active function to other neurons through electrical and chemical transport mechanisms. The output signal strength is modified by a special connection called a synapse before entering another neuron. It is believed that certain forms of learning occur when the synapses are trained to assume certain strength or weights by repeated exposure to the same stimulus. An ANN consists of nonlinear processors called nodes (corresponding to neurons) linked to each other by interconnections analogous to the axons. Each interconnection has an associated scalar weight (corresponding to a synapse) whose value can be modified during the learning phase of the neural network.

Although the concept of a neural network appears initially to be quite straightforward, there is a bewildering array of different kinds of ANNs which now exist to solve different problems; Fig. 1 gives a classification.

<sup>1</sup>British Geological Survey, Murchison House, West Mains Road, Edinburgh EH9 3LA, Scotland, UK.

<sup>2</sup>Department of Geology and Geophysics, University of Edinburgh, West Mains Road, Edinburgh EH9 3JW, Scotland, UK.





**Fig. 1.** General artificial neural network classification. The neural networks are divided into three main categories: feed-forward, feed-back and self-organization neural networks. In each category, there are many kinds of neural network, but we only list two to demonstrate the classification.

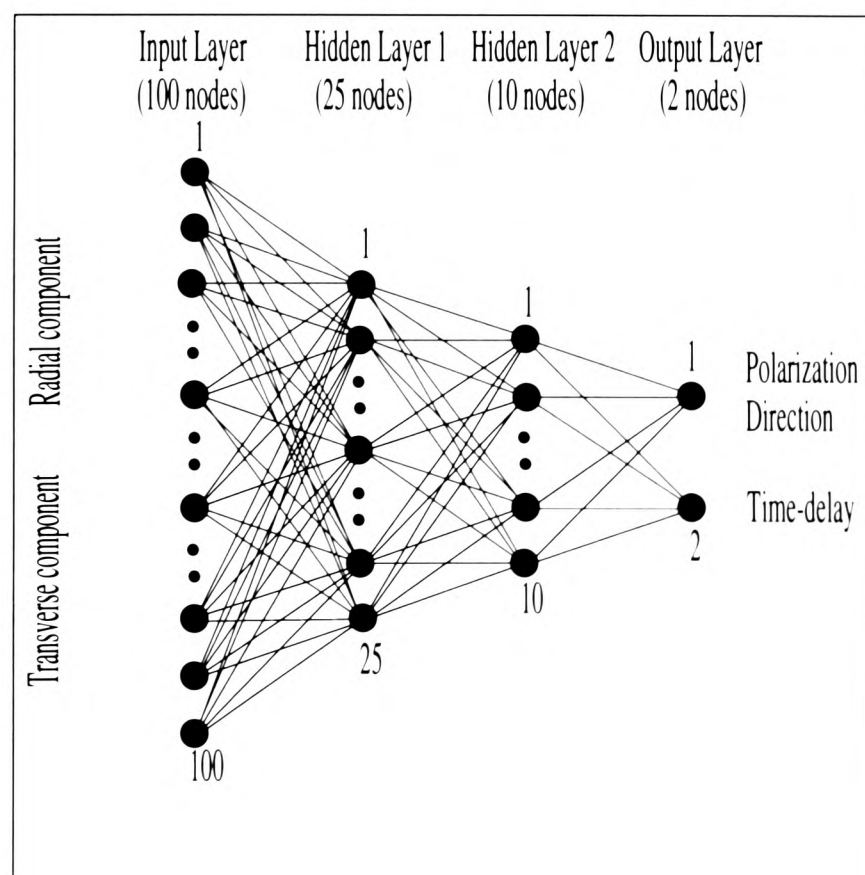
Different ANNs have different topological structures for the neurons and their connections, and use different learning algorithms based on different philosophies. The main categories for this classification are listed below.

- Feed-forward neural networks—the output of each layer feeds the next layer of units;
- Feed-back neural networks—the input information defines the initial activity state of a system. The first output of the system is taken as the new input, which produces a new output and so on;
- Self-organizing neural networks—where neighbouring cells compete in their activation by means of mutual lateral interactions, and develop adaptively into specific detectors of different signal patterns.

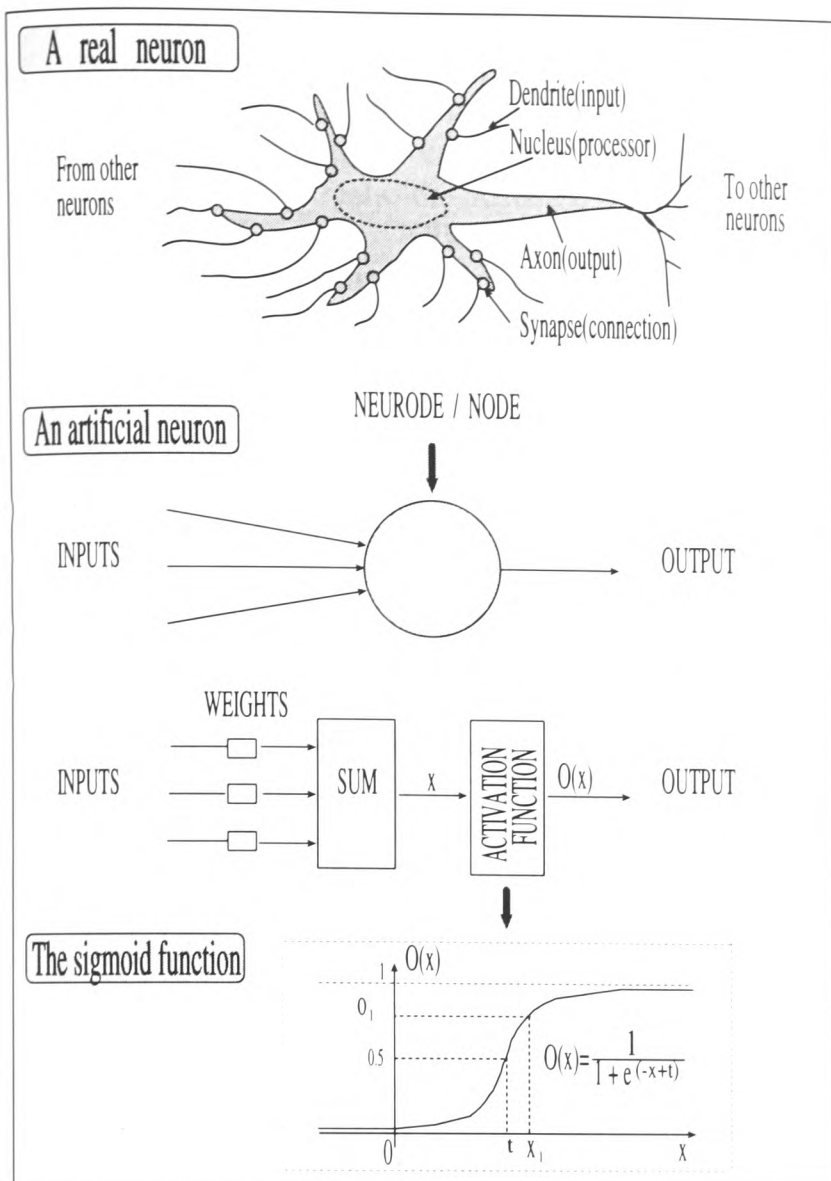
Each of them has applications in solving different types of problem. The ANN used in this study is defined as nonlinear, multilayer, and feed-forward, being trained by the back-propagation of error (Rumelhart *et al.* 1986). This is the most popular type of ANN in use for pattern recognitions, as it is well understood and appears quite versatile. ANNs are used to solve a diversity of geological and geophysical problems. For example, Palaz and Weger (1990) use ANNs to recognize waveforms in seismic data. ANNs have also been used to combine synthetic spontaneous potential (SP) and resistivity logs (McCormack 1991) to estimate

lithology logs. Poulton *et al.* (1992) used ANNs to estimate offset, depth, and conductivity-area product of a conductive target given an electromagnetic image of the target. ANNs also have been used in the first-break picking of surface seismic data (Murat and Rudman 1992; McCormack *et al.* 1993). Moreover, other kinds of neural networks are also used in seismic event classification (Leggett *et al.*, 1993). This kind of network represents our first and simplest choice of network architecture to solve our problem; will it be successful?

Our chosen network is composed of sets of *nodes* arranged in *layers*: one input layer, two intermediate hidden layers, and one output layer. This architecture is illustrated schematically in Fig. 2, where we have drawn nodes as solid circles and interconnections as straight lines. The node forms the basic processing unit of the network, and the various components of this unit are shown in Fig. 3, alongside a real neuron of the human brain. The first (input) layer just passes the signals to the various nodes in the second layer through the interconnections. The nodes in the hidden layers and the output layer receive and sum the signals and output according to their activation functions. As not all nodes need be given the same priority, a weighting is applied to each input signal before a summation procedure. These weights are applied as simple multiplicative scalars and



**Fig. 2.** Neural network structure used for analysing shear-wave splitting in VSP data. Four layers are used: the first layer has 100 nodes and inputs the radial and transverse components in the order of  $r_1, r_2, \dots, r_{50}, t_1, t_2, \dots, t_{50}$ . There are 25 and 10 nodes in the first and second hidden layers respectively. Only two output nodes are present, corresponding to the polarization and time-delay estimates.



**Fig. 3.** Schematic diagram of various constituents making up an artificial node in a neural network, compared to a real neuron in the human brain. The symbol,  $x$  represents the summed input to the node,  $t$  the threshold, and  $O(x)$  the output of the node after being conditioned by a non-linear activation function.

effectively amplify or attenuate the signals. This weighting simulates synaptic connections to the neuron body. With the exception of the nodes in the input layer, the net input to each node is the sum,  $x$ , of the weighted outputs of nodes in the previous layer. This node transmits an output signal,  $O(x)$ , to other nodes when the energy level of the combined input to a particular node exceeds a threshold level,  $t$ , depending upon the characteristic activation function. The activation function may assume some smooth S-shape, which may be represented mathematically by the sigmoidal function (although other functions are available). The outputs of each node in the input layer is a component of the input pattern.

In our example we input two horizontal recordings of seismic shear waves into the network and output the values of two parameters:  $\phi$ , the polarization of the fastest shear-wave, and  $\Delta\tau$ , the differential travel-time of the split shear-waves. No assumption is made about the polarization of the slower wave, the differential amplitude, or the source wavelet shape. The input layer has 100 nodes (50 for radial component input and 50 for transverse component input), the first hidden layer 25

nodes, the second hidden layer 10 nodes, and the output layer 2 nodes. This provides a window size of 196ms, with 4ms sample intervals for the data. The actual numbers of hidden layers and nodes in each layer are somewhat arbitrary, but do depend on external constraints from the physical problem such as the number of input and output nodes and also on the desired system error, pattern error and the nature of the training samples. There is no fixed relationship between these various factors for this type of neural network, although there are certain other network designs where this is the case (Kusuma and Brown 1992). We are guided by the knowledge that generalization is increased and memory is reduced by limiting the number of hidden nodes (Dowla *et al.* 1990). Too few hidden nodes will lead to a long learning process or no convergence. Too many hidden nodes will introduce noise in the output. In this experiment, our final choice was made after trial and error with different configurations.

The choice of a programming language is more critical than in other situations due to the computation demands of neural networks. Whatever language is used, it is advisable to seek a version that has been optimized for numeric data. The C language has become something of a de facto standard for neural network programming. Our neural network was set up using a C program written on a VAX4000 under VAX/VMS<sup>TM</sup> operation system.

### Training the network

In our application, the ANN may be likened to an analyst, and must be trained by presenting many different patterns of shear-wave particle motion recorded by orthogonal horizontal geophones with known parameters. In our case these are provided from synthetic seismograms, but they could also be field data interpreted in an analyst's style or, ideally, several analysts' styles. After training is accomplished, it is hoped that the ANN can estimate the splitting parameters from a variety of field VSP seismograms, consistently mimicking the expert interpretation. The ability of a neural network to recognize new patterns depends on the training patterns used as references. The learning occurs when the weights are trained to assume certain values by repeated exposure to the same stimulus. The most popular way to adjust the weights and threshold values of the sigmoidal functions is to use a back-propagation learning algorithm, or *Delta Rule*. A good mathematical description of this is given by Pao (1988). This method attempts to find the most suitable solution (numerical values of thresholds and weights) for global minimization of the mismatch between the desired output pattern and its actual value for all of the training examples. The degree of mismatch for each input output pair is quantified by solving for the unknown parameters between the hidden and output layer and then by propagating the mismatch backwards through the network to adjust the parameters between



the input layer and hidden layer. The first input pattern is presented to an initially randomized network, and the weights and thresholds adjusted in all the links. Other patterns are then presented in succession, and the weights and thresholds adjusted from the previously determined values. This process continues until all patterns in the training set are exhausted. It is generally accepted that this procedure is independent of the order in which the example patterns are presented. However, a final check can be performed by looking at the *pattern error* which is defined as the square of difference between desired output and neural network output for each pattern and the *system error* which is defined as the average of all pattern errors, to determine whether the final network solution satisfies all of the patterns presented to it within a certain error. The set of weights and thresholds in the network are now specifically tailored to 'remember' each input and output pattern, and can consequently be used to recognize or generate new patterns given an unknown input. The network is now trained, and can be used in subsequent analyses. Figure 4 summarizes the various stages of training and implementation.

The training patterns we used were generated from synthetic seismograms computed for a zero-offset VSP acquisition geometry using the anisotropic reflectivity method (Taylor 1992) because they were simple, regular, noiseless and give known splitting parameters. The horizontal X and Y recordings were computed for shear-waves generated by a horizontal point source along the Y direction, propagating through a uniform anisotropic half-space which simulates a range of  $qS/$  polarization directions from  $-X80^\circ Y$  to  $X80^\circ Y$  with an interval of  $10^\circ$ , and time-delays from 0ms to 40ms in an interval of 4ms. We fed the seismograms, not their

attributes, directly into the ANN as it is difficult to know which attributes give suitable information for solving the problem. The advantages of using seismograms is that we do not need to enforce a preconceived wavefield model. The disadvantage is that not all mathematical functions can be simulated by the ANN, and so we may exclude certain attributes. The radial component  $R(t)=\{r_1,r_2,\dots,r_{50}\}$  and transverse component  $T(t)=\{t_1,t_2,\dots,t_{50}\}$  were fed into the ANN in the order:  $\{r_1,r_2,\dots,r_{50},t_1,t_2,\dots,t_{50}\}$ .

Although only 161 patterns in total were used in the training, the training procedure was slow (4507 iterations in about one and a half hours CPU time on a VAX4000), indicating that the global optimum may not be well-defined. The procedure did converge, however, to an acceptable system error of  $8.02 \times 10^{-6}$ , with all output patterns achieving an error of less than  $10^{-4}$  with their ideal form. As a final check on this solution, the training data were fed back into the trained neural network. The resultant outputs indicated that all, except one pattern, lie within a tolerable maximum polarization error of  $2^\circ$ , and a maximum time-delay error of 1ms. A further check was made by inputting new synthetic data with polarizations differing by  $5^\circ$ ; Fig. 5 displays a portion of these results. The results indicate that the network solution does appear internally consistent, and should in principle be capable of recognizing similar patterns in field data with unknown characteristics, but not lying too far from the experience of the network. One important advantage of ANNs is that once trained, they are very fast in generating the desired output. With

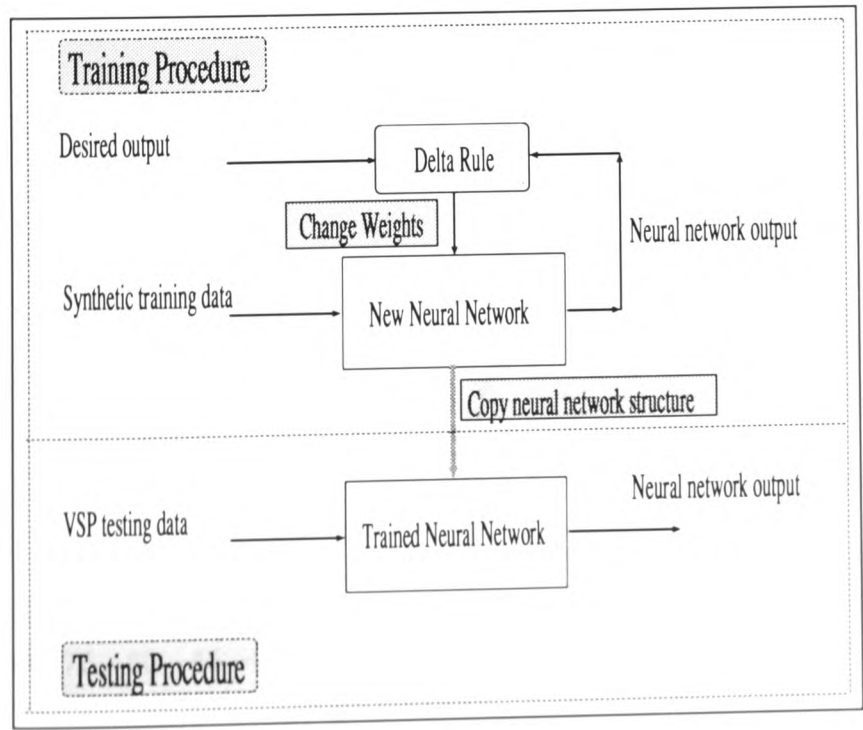


Fig. 4. Various component parts for neural network training and testing

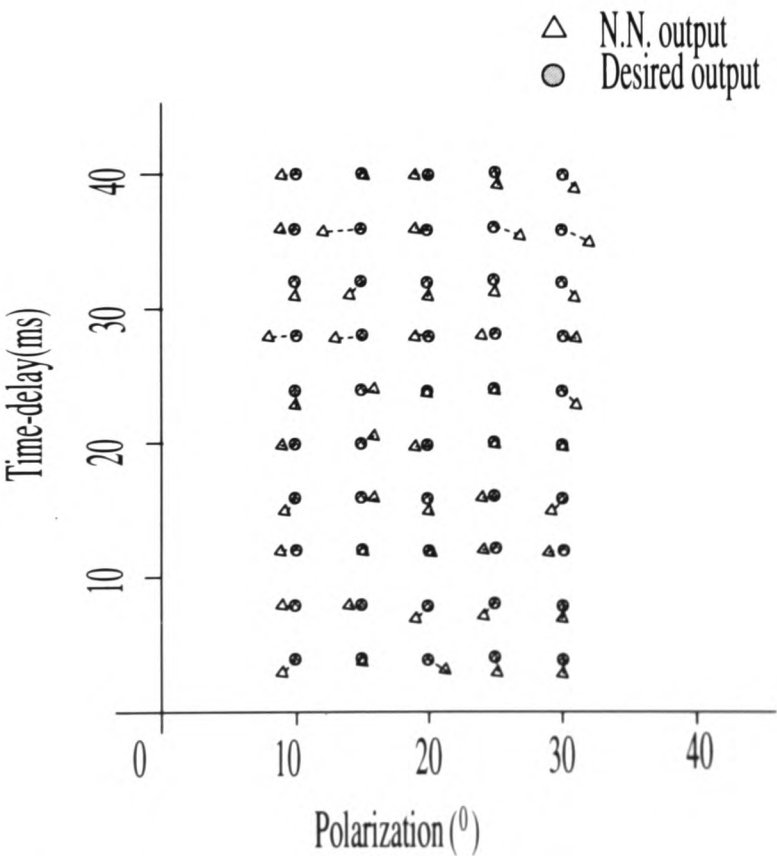


Fig. 5. Comparison of test results for trained network applied to various different synthetic VSP data, demonstrating an apparent internal consistency.



our particular solution, it took only 15 s of interactive CPU time to process the 161 training datasets.

### Is the network successful with real data and what lessons can we learn?

Accepting that the trained neural network could perform its intended task on synthetic data, we now attempted to analyse real data. A three-component shear-wave VSP dataset generated by cross-line horizontal vibrators at a borehole site in the Paris Basin is chosen as it has been studied extensively (Clet and Lefeuvre 1989). The depth surveyed in the borehole ranged from 1100m to 2060m with recording levels every 15m, and for our work we tested recordings of the 272m source offset made between 1325m and 2060m. The horizontal geophone components were re-oriented using the *P*-wave offset data prior to our study. The sampling rate is also reduced from 2ms to 4ms so that we could accommodate a large enough time-window (196ms) for the analysis. After this initial rearrangement of the data, the seismograms were processed by the neural network very quickly, taking only about seven seconds to process 50 recordings. The shear-wave parameters  $\phi$  and  $\Delta\tau$  output from the network compared favourably with those determined by the single source technique of Zeng and MacBeth (1993) (Fig. 6). A selection of five

observed particle motions are shown in Fig. 7, alongside the corresponding synthetic motions chosen by the network as being the closest fit. Perhaps the most noticeable feature of these comparisons was that the direction of the initial onsets of the motions picked by visual examination did not correspond to either the picked polarization estimates for the observed data, or the actual polarization estimates for the observed data, whilst the network results agree with both. This appears to challenge our conceptual understanding of such motions: the initial onset should principally contain energy from the *qS1* wave and hence, depending upon the time-delay, give a direct indication of the polarization direction. In fact, it appears that the overall general character, including mainly the lobe of energy in the region of interference between *qS1* and *qS2*, is matched by the network and is the actual indicator for the choice of particle motion.

Given this disparity between visual examination and the network results, a further insight is required to view the mechanism by which the neural network performs its classification. This is achieved by investigating the magnitudes of the different weights in the completely trained network. Figure 8 displays weight maps of the neural network trained on the synthetic seismograms

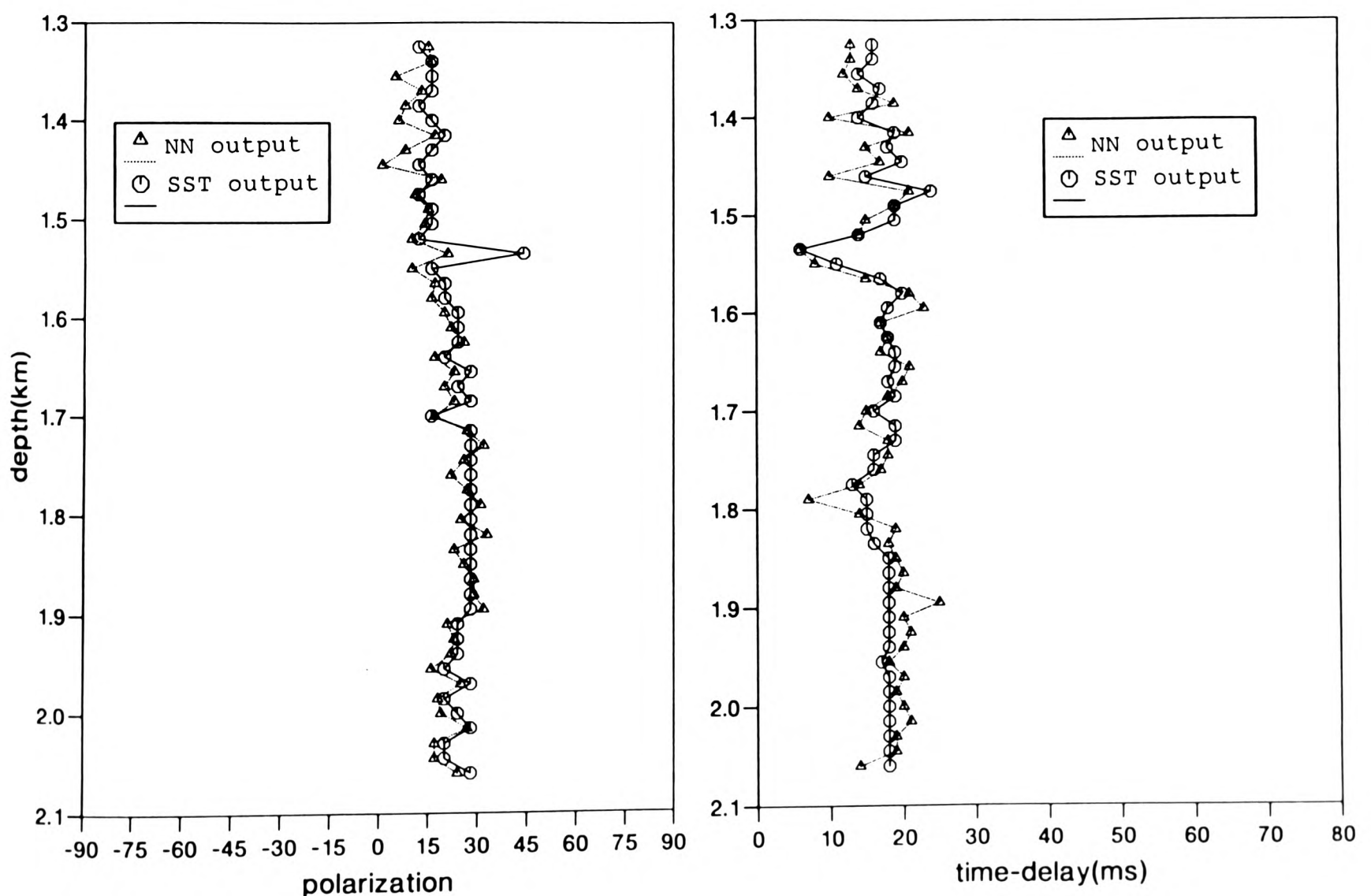


Fig. 6. Test results for Paris Basin VSP data compared with the results determined by the single source technique (SST)

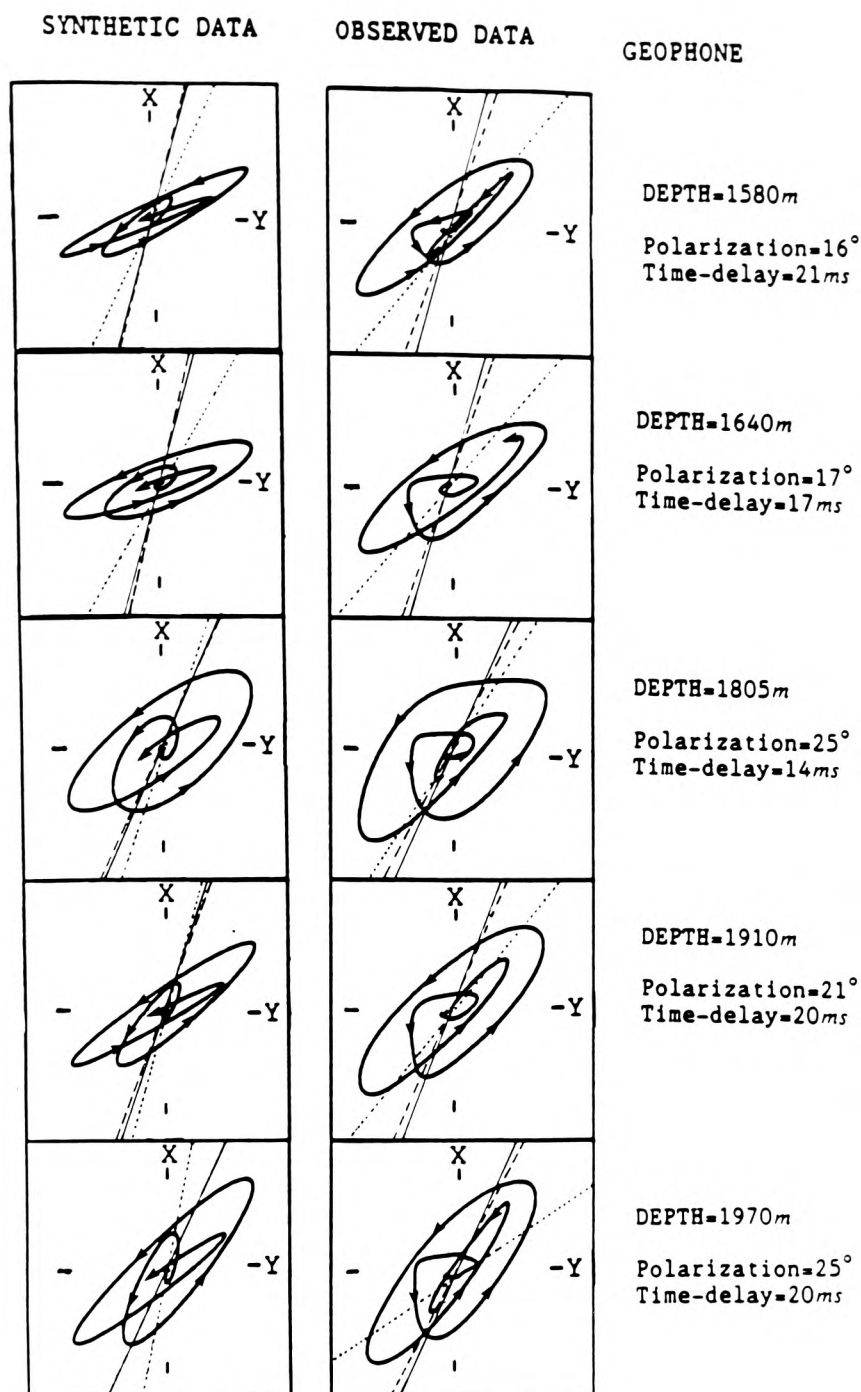


Fig. 7. Selection of five observed particle motions in the horizontal plane and corresponding synthetic particle motions chosen by the network as having the closest fit. Lines on each plot show the polarization azimuth chosen by the network (dashed line), polarization azimuth chosen by the single source technique (solid line on the observed data) and actual model polarization (solid line on the synthetic data), and polarization direction chosen by visual examination (dotted line).

used in this study. These maps show the magnitudes of each weight connecting the different layers in the network in Fig. 2. In each weight map, solid circles are shaded on a grey-scale corresponding to a magnitude range indicated by the key, with the far right row of circles in each map representing the nodal threshold. Although it is not possible to fully understand the logic underlying the network solution by visually inspecting these weight patterns, the weight map may still lend some perspective as to which aspects of the seismic waveform are more relevant to the solution than others, and thus be of benefit to further processing schemes or studies.

Before training, the weights and thresholds are initialized randomly between  $-0.5$  and  $+0.5$ . After training these values range from  $-3.7$  to  $+4.2$ . It appears that the weights connecting the input to first hidden layer concentrate on the portion of the signal between samples 14 to 40 (52ms to 156ms), corresponding to the main energy in the input signal, the  $qS1$ - $qS2$  interference lobe. A curious feature revealed by Fig. 8 is that weights of large magnitude appear only for the radial component. This result suggests that the network is somehow sensing the departure from isotropic behaviour, as in an isotropic medium the radial component would be zero. Large negative and positive weights correspond to negative and positive signal values, yielding a large positive product,  $x$ . Due to the nature of the activation function in Fig. 3, which has in its denominator a negative exponential with the exponent being the summation of the product of weights and signal input, these large values produce in turn a large negative exponent, thus accentuating  $O(x)$ . The input weights separate into two distinct groups, both concentrated around the central lobe, which in turn connect separately (but not totally independently) with the weight distributions for the output polarization and time-delay.

To gain more understanding of the various weighted combinations of the input signal, which ultimately determine  $\phi$  and  $\Delta\tau$ , we trained the network using split shear-waves with a source wavelet which is a single pulse  $\delta(t)$ . In this case, each training example has four non-zero sample values, two of which correspond to delayed secondary arrivals. The training dataset is created for the same range of polarization and time-delay as in the previous example. Figure 9 gives the weight maps for this trained network and correctly shows that there are large values only for the range of signal amplitudes in the training dataset (nodes 20 to 30 and nodes 70 to 80 inclusive) corresponding to the maximum time-delay of 40ms. Note that in Fig. 9, the particular input signal example has a 20ms time-delay corresponding to input nodes 25 and 75. This does not reflect the range of time delay in the network which we initially set by presenting all training examples. It is interesting to note that both components are now strongly weighted. The weight combinations for this map are not easily traced as the parameter estimates appear to be more widely spread amongst the connections, especially for the second hidden layer. It must be concluded that the connections between the input layer, first and second hidden layers appear to be source-wavelet dependent and are probably being trained for the specific dataset. This agrees with our findings that if the source wavelets used in the synthetic training data are too dissimilar from the field data, the neural network fails to recognize the field data.

Several other indicators of these difficulties have also arisen during sensitivity tests of our results. For example, we find the output of the trained network is



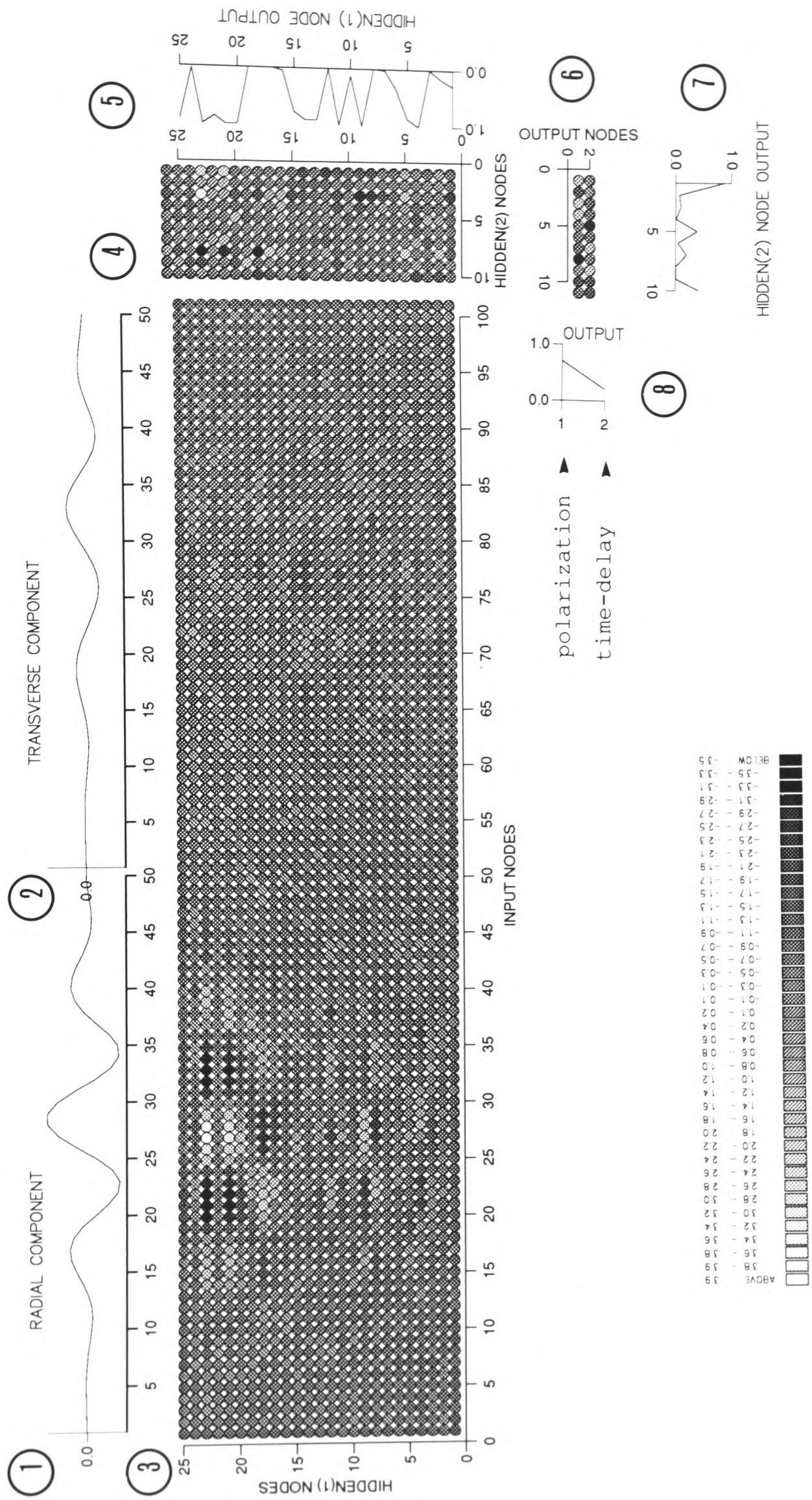


Fig. 8 Weight map for the trained neural network and a sample of synthetic training dataset. (1) and (2) correspond to the input traces ( $r_1, r_2, \dots, r_{50}$  and  $t_1, t_2, \dots, t_{50}$ ); (3) the mapping between the input layer and the first hidden layer (the weights having been reordered); (4) the mapping between the first hidden layer and the second hidden layer; (5) the output from the first hidden layer; (6) the mapping between the second hidden layer and the output layer; (7) the output of the second hidden layer; and (8) the final output. In each weight map, the right hand row of circles represent the node thresholds. The contrast in each circle indicates the value of a weight.

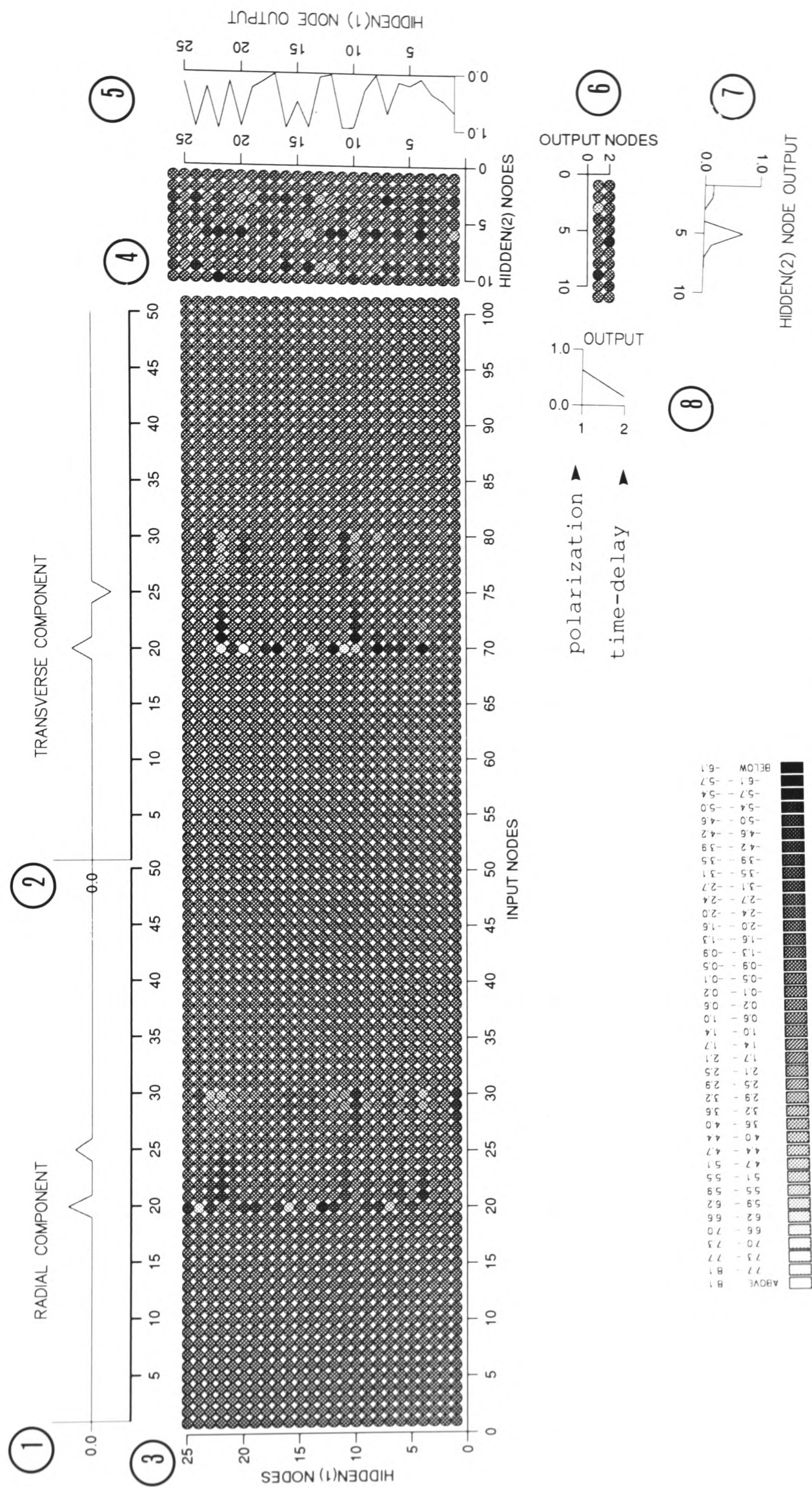


Fig. 9. As in Fig. 8, but for neural network trained by a single delta function  $\delta(t)$  as an input source wavelet.

sensitive to the size and position of the waveforms in the time-window. Shifting the position of the window may occasionally cause large changes in the neural network output. The shear-wave must be carefully picked to ensure accurate processing, which necessitates a pre-analysis algorithm for picking (which has been attempted for both exploration and earthquake data). It appears that the ability of ANN also depends on the type of training data, and a critical factor is the choice of training datasets to match the field data for source and medium characteristics. Consequently, we must inspect the field data first, before selecting suitable parameters to create the training data. To make the neural network more robust, we need to add more training data, exploring different wavelets, coherent noise, source direction and offsets. The major drawbacks with this approach are the excessive increase in training time (which increases exponentially!) and a large and unmanageable network structure.

### Conclusions

It is widely known that neural networks are ideally suited to situations where standard algorithms cannot be used as the mathematical relationship is uncertain, provided a supply of many examples of what is required can be generated. Once trained using the example patterns, it can recognize new patterns using its memory. In this respect, the approach is simple and adaptive to any problem. However, there are some limitations in this philosophy, which we have encountered in our application of our chosen type of network. The performance depends on the training data and its ability to generate solutions cannot lie too far outside its experience—in our case the solution is a highly non-linear function of wavelet and its shape cannot deviate too far from the actual source wavelet. Although, in general, extra training sets should help with this problem, for this application it is too time-consuming to train the network to recognize more complicated wavelets, and pre-processing for the source function appears necessary. Alternatively, we might choose a different network architecture from Fig. 1. On a more positive note, this ANN application has been of value in exploring the emphasis placed on each component part of the seismogram for this particular problem; ANNs have the ability to discover a relationship, no matter how complex, without *prior* statistical concepts.

Consequently, application of a network can often be used as a tool to highlight relationships which exist between certain quantities, and help gain a greater insight into the physical process.

### Acknowledgements

This research was sponsored by Global Seismology Research Group (GSRG), British Geological Survey (BGS) of the Natural Environmental Research Council (NERC), and is published with the approval of the Director of the BGS (NERC). We thank the staff and students of GSRG for their help and thank the French VSP Consortium for the VSP data.

### References

- CLJET, C. and LEFEUVRE, F. 1989. Characterization of the micro-structure of the sedimentary rocks in the Paris Basin from shear-wave splitting analysis. *Presented at the 51st Annual Meeting, EAEG*.
- DOWLA, F.U., TAYLOR, S.R., ANDERSON, R.W. 1990. Seismic discrimination with artificial neural networks: preliminary results with regional spectral data. *Bulletin of the Seismological Society of America*, **80**, 1346–1373.
- KUSUMA, T., and BROWN, M. 1992. Cascade-correlation architecture for first-break picking and automated trace editing. *Proceedings of SEG 62nd Annual International Meeting, Society of Exploration Geophysics*, 10–13.
- LEGGETT, M., SANDHAM, W.A., and DURRANI, T.S. 1993. Seismic event classification using a self organising Kohonen network. *Proc. IEE/IEEE int Workshop on Neural Algorithms in Signal Processing*, 14–16 November 1993, Chelmsford, UK.
- MCCORMACK, M.D. 1991. Neural Computing in Geophysics. *Geophysics: The Leading Edge of Exploration*, January 1991, 11–15.
- MCCORMACK, M.D., ZAUCHA, D.E. and DUSHEK, D.W. 1993. First-break reflection event picking and seismic data trace editing using neural network. *Geophysics* **58**, 67–78.
- MURAT, M. and RUDMAN, A. 1992. Automated first arrival picking: A neural network approach. *Geophysical Prospecting* **40**, 587–604.
- PALAZ, I., and WEGER, R.C. 1990. Waveform recognition using neural network. *Geophysics: The Leading Edge of exploration*, March 1990, 28–31.
- PAO, Y.H. 1988. *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley Publishing Company Inc., New York.
- POULTON, M.M., STERNBERG, B.K. and GLASS, C.E. 1992. Location of subsurface target in geophysical data using neural network. *Geophysics* **57**, 1534–1544.
- RUMELHART, D.E., MCCLELAND, J.L. and the PDP Research Group. 1988. *Parallel Distributed Processing: Exploration in the microstructure of Processing, Volume 1: Foundations*. MIT Press, Cambridge, MA.
- TAYLOR, D.B. 1992. *ANISEIS v4.6 Manual*. Applied Geophysics Inc., Houston, TX.
- ZENG, X. and MACBETH, C. 1993. Algebraic processing techniques for estimating shear-wave splitting in near-offset VSP data: theory. *Geophysical Prospecting* **41**, 1033–1066.



### Statistics of this thesis

	words	Figures	Tables	Pages	Total pages
Abstract:	769			3	3
Acknowledgements:	469			2	2
Contents:	580			5	5
<i>sub total:</i>	<i>1818</i>			<i>10</i>	<i>10</i>
Chapter 1:	2789	2		9	11
Chapter 2:	4612	8		15	23
Chapter 3:	3406	9		12	21
Chapter 4:	9059	17 (24)	3 (4)	27	55
Chapter 5:	5067	15 (24)	2	15	41
Chapter 6:	4447	11 (14)	4	13	31
Chapter 7:	2728			9	
<i>Appendix A:</i>	<i>616</i>			<i>4</i>	
<i>sub total:</i>	<i>32724</i>	<i>62 (81)</i>	<i>9 (10)</i>	<i>104</i>	<i>195</i>
Reference:	2292 (106 item)			10	10
Total	36834	52 (81)	9 (10)	124	215

Thesis submitted on **23 December, 1995**

Passed oral examination on **4, March 1996**