USC–SIPI REPORT #221

First Break Refraction Event Picking Using Fuzzy Logic Systems

by

Jerry M. Mendel and Chung-Kuang Chu

September 1992

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 400
Los Angeles, CA 90089-2564 U.S.A.

Contents

List of Figures

List of Tables

Abstract

First break picking is a pattern recognition problem in seismic signal processing, one that requires much human effort and is difficult to automate. Our goal is to reduce the manual effort in the picking process and accurately perform the picking.

Recently, feedforward neural network first break pickers have been developed using back propagation training algorithms applied either to an encoded version of the raw data [7] or to derived seismic attributes which are extracted from the raw data [14]. In this report we summarize a study in which we applied a recently developed back-propagation fuzzy logic system (BPFLS) to first break picking. In our work we used derived seismic attributes as features, and we also took inter-trace continuity into account by using the distance to a piecewise linear guiding function as a new feature.

Experimental results indicate that the BPFLS achieves about the same picking accuracy as a feedforward neural network that is also trained using a back propagation algorithm; however, the BPFLS is trained in a much shorter time, because of the very good way in which the initial parameters of the BPFLS can be chosen, versus the random way in which the weights of the neural network are chosen.

1 Introduction

The first break(FBR) (or first arrival(FAR)) at each seismic detector (geophone) is indicated on the corresponding seismic trace by a pronounced rise in amplitude above the background level, after which the level of ground motion decreases somewhat but still remains much more active than before [2]. The significance of obtaining an accurate estimate of first-break-time lies in the static corrections which utilize this first-break-time estimate. Static-time corrections are important steps in refraction prospecting. In order to obtain the greatest amount of information possible from seismic data, it is essential to correct reflection times for predictable anomalies not associated with structure at the depth of interest. The weathered layer just below the earth's surface is a source of irregularity about which we usually have the least amount of information. The goal of first-break picking is to compensate for such kinds of irregularities.

First-break picking is an easy task for a human; it does not need much geophysical knowledge; however, it becomes tedious for a human to do it for huge amounts of seismic data, e.g., 3-D survey data. Consequently, automatic picking algorithms have been developed which emulate the human perception process.

FBR picking is a pattern recognition problem which is easy for human perception but difficult for computers. Usually, manual picking has higher accuracy than machine picking. On the other hand, machine picking can provide higher consistency. A mixed approach composed of machine picking and interactive manual picking, seems to be a plausible solution using current technologies. The interactive manual picking part, which should be a small fraction of the entire task, can serve as the injection of human intelligence into the system to maintain the desired accuracy.

Two different neural network approaches, based on different data features, are reviewed in Sections 2 and 3. In Section 4, we review the fundamentals of fuzzy logic and back-propagation fuzzy logic system (BPFLS); describe the procedure of FBR picking using a BPFLS; Compare the performance between the BPFLS and back-propagation neural network (BPNN) via extensive simulations; and compare the convergence behaviors of a BPFLS and a BPNN. Discussions and

conclusions are stated in Section 5. Future work is summarized in Section 6.

2 First-Break Picking Using a Neural Network Based on Trace Pixel-Images

McCormack, et al [7] applied a back-propagation neural network approach to the first break picking problem. They used a two-layer network containing close to 5,000 input neurons and two output neurons. The seismic traces are represented as a two-dimensional pixel image with only two grey levels (0 and 1). One segment of the seismic data and the corresponding pixel image is shown in Fig. 1. The seismic data is quantized into 5 levels; hence, the seismic shot records displayed in their paper [7] do not look very smooth. The typical window size is 9 to 11 traces by 100 samples in time; hence, there are around 5,000 input neurons (if there are 11 traces in the window, five of them are above the current trace and the other five are below, so that $11 \times 100 \times 5 = 5500$ input neurons are needed). Two zero-one output neurons are used to indicate whether the current peak located at the window center is or is not the FBR peak. In their training procedure the criterion for setting the desired outputs is shown in Fig. 2. In a given trace the outputs for those peaks starting from the first potential FBR peak up to the manual picked FBR peak are set equal to $(1, 0)$ and the outputs for those succeeding peaks are set equal to $(0, 1)$. McCormack, et al [7] started the neural network training procedure with a two-layer network, which had around 5,000 input neurons and two output neurons. One or more hidden layers were added to enhance the approximation capabilities when the training procedure had difficulty converging. After the training procedure was completed the seismic shot records were processed in a batch mode.

The basic data unit is a window containing around 5,000 binary values and having a candidate FBR peak located at the center of the window. The samples (windows) are processed trace by trace. The criterion for classifying the FBR peak is as follows. The FBR peak appears at the first of two peaks (centers of two consecutive windows) whose outputs are $(1, 0)$ and $(0, 1)$; i.e., the FBR peak is located at the transition between two opposite outputs of the neural network. A reliability

factor, R, was defined as [7]

$$R = (|O_1 - O_2|^k + |O_2 - O_1|^{k+1})/2 \tag{1}$$

where O_1 and O_2 are the outputs of neurons 1 and 2 of the output layer, and the superscripts represent the k^{th} and $(k+1)^{th}$ consecutive peaks on the trace at which the state changes. The reliabilty factor R provides a measure of the presence of multiple probable FBR peaks on a single trace. In addition, R is used as an indicator of reliability for quality control purposes.

A post-processor was employed to reduce the effects of noise, waveform character changes and signal amplitude variations. The lateral continuity (linear moveout trend) and reliability factors were considered in order to adjust the neural network FBR pickings so that the most probable FBR trajectory was determined.

McCormack, et al also proposed several interesting strategies to accelerate the rate of convergence and escape from a local minimum. It seems inevitable that such difficulties occur when the network size is large. In the next section we review an alternative approach [14] based on a back-propagation neural network which has much fewer neurons.

3 First-Break Picking Using a Neural Network Based on Hilbert Attributes

Veezhinathan, et al [14] solved the first-break (FBR) picking problem by using a neural network based on four signal attributes of the seismic trace. Their neural network classifies the peaks in a trace as FBR peaks or non-FBR peaks. Each peak (or trough) is characterized by a set of signal attributes obtained from a variable length window containing three consecutive peaks where the designated peak is the central peak of the three. The following four attributes were used for characterizing a peak:

1. Maximum amplitude of the peak (or trough);

2. Mean power level in the five-sample window. If $f(t)$ denotes a seismic trace and $g(t)$ its Hilbert transform, then we compute the envelope $E(t) = \sqrt{f^2(t) + g^2(t)}$ and from this envelope determine the mean power level (MPL) as

$$MPL(t) = \frac{1}{5}\sum_{t-2}^{t+2} E^2(t)\Delta t; \tag{2}$$

3. Power ratio (PR) between a forward and reverse sliding of the five sample window. PR is given by $PR(t) = MPL(t+2)/MPL(t-2)$;

4. Envelope slope (ES) at the peak value. ES is given by $ES(t) = \Delta E(t)/\Delta t$, where $\Delta E(t) = E(t) - E(t-1)$.

Some of the above attributes which are related to the Hilbert transform of the seismic trace $f(t)$, and are referred to as Hilbert attributes, were proposed by Taner, et al [12] to analyze seismic traces. The preceding four attributes are considered to have good discriminating capabilities for differentiating FBR peaks and non-FBR peaks [13].

Veezhinathan et al's network processes three consecutive peaks at a time in order to decide if the central peak is a FBR peak. Two adjacent peaks provide the central peak with more spatial correlation to differentiate the FBR peak. Their back-propagation neural network has 12 input neurons (4 attributes per peak times three peaks), 5 hidden units (obtained after experimentation) and a single output neuron. The function of this network is to pick those most likely FBR peaks trace by trace; it does not consider intertrace correlations; hence, a post-processing procedure is performed.

A least-squares line is fit through all the peaks identified by the network as FBR peaks for a shot record. This least-squares line lets the linear trend of those FBR peaks among consecutive traces be taken into account so as to determine more accurate estimates of FBR peaks. For example, Veezhinathan, et al [14] claim that some false alarms can be eliminated, and, estimated positions of FBR peaks can be obtained for those traces which have no FBR peaks detected in the neural network processing phase. It is common to have a multistage processing strategy in seismic signal

processing because there is no explicit optimal criterion we can refer to during processing.

Veezhinathan, et al [14] subsequently included a fifth attribute, the distance to a *reduced traveltime curve* ("t_r is the difference between the observed refraction travel time and the travel time which would have been observed if only one head-wave velocity (V) had been involved: $t_r = t - x/V$, where x = offset distance. The result yields the intercept time (source delay time + geophone delay time) if the assumed velocity is the refractor velocity." defined in [11]) , as another input feature for each peak. This fifth feature replaces the post-processing step and attains comparable results to that obtained by post-processing. The distance to a reduced traveltime curve serves as an indicator of the linear trend of FBR peaks in consecutive traces.

Compared with the approach of McCormack, et al [7], the present method seems simpler because it extracts some derived signal attributes [3] from the recorded seismic data and the neural network operates on these attributes (i.e., features). It dramatically reduces the data required for the FBR picking problem. If the four (or five) features are capable of retaining *sufficient* information about the original data, this second method appears to be computationally simpler than [7].

4 First-Break Picking Using Fuzzy Logic Systems

First-break picking is a pattern recognition problem. It is easy to perform by a human but is difficult to implement automatically. The reasons are that seismic signals are not easy to model accurately for different environments, and rules governing the exploration process are usually subjective [1]. In the preceding two sections we reviewed two model-free neural network approaches that can do a better first-break picking job than can statistical and parametric methods [7, 14]. The only disadvantage of the neural network type algorithms is that they can not incorporate subjective decision rules explicitly, i.e., neural networks can only learn rules from a numerical training sample. In case we do not have rich training samples, then these rules can not be well learned during the training phase. An alternative approach, namely fuzzy logic systems (FLS), can not only learn from training samples but can also incorporate useful linguistic rules obtained from human experts [15,

16, 17]. This seems useful for seismic exploration problems because sometimes subjective linguistic rules from experts play an important role in seismic decision processes.

4.1 Review of Fuzzy Sets and Fuzzy Logic Systems

The basic concepts and terminologies of fuzzy sets are briefly summarized in this section. A more detailed discussion may be found in [4] and [19].

Let U be a collection of objects denoted by u, which may be discrete or continuous. U is called the universe of discourse and u represents an element of U. A fuzzy set F in a universe of discourse U is characterized by a membership function μ_F which takes values in the interval [0,1], namely, μ_F: $U \rightarrow [0,1]$. A fuzzy set can be viewed as a generalization of the concept of a crisp set whose membership function only takes two values {0,1}; thus, a fuzzy set F in U may be represented as a set of ordered pairs of an element u and its grade of membership function, i.e., $F = \{(u, \mu_F(u)) \mid u \in U\}$. When U is continuous, a fuzzy set F can be expressed as

$$F = \int_U \mu_F(u)/u. \tag{3}$$

When U is discrete, a fuzzy set F is represented as

$$F = \sum_{i=1}^{n} \mu_F(u_i)/u_i. \tag{4}$$

The representations in Eqs. (3) and (4) mean that each element $u \in U$ has its associated degree of membership $\mu_F(u)$ with respect to fuzzy set F.

The set theoretic operations of union, intersection and complement for fuzzy sets are defined via their membership functions. Let A and B be two fuzzy sets in U with membership functions μ_A and μ_B, respectively. The membership function $\mu_{A \cup B}$ of the union is pointwise defined for all $u \in U$ by

$$\mu_{A \cup B}(u) = max\{\mu_A(u), \mu_B(u)\}. \tag{5}$$

The membership function $\mu_{A\cap B}$ of the intersection is pointwise defined for all $u \in U$ by

$$\mu_{A\cap B}(u) = min\{\mu_A(u), \mu_B(u)\}. \tag{6}$$

The membership function $\mu_{\overline{A}}$ of the complement of a fuzzy set A is pointwise defined for all $u \in U$ by

$$\mu_{\overline{A}}(u) = 1 - \mu_A(u). \tag{7}$$

The basic configuration of the fuzzy logic system is shown in Fig. 3 [5, 6]. There are four components in a fuzzy logic system (FLS), namely: fuzzification interface, fuzzy rule base, fuzzy inference machine and defuzzification interface. As described in [5], the fuzzification interface

1. measures the values of input variables,
2. performs a scale mapping that transfers the range of values of input variables into corresponding universes of discourse, and
3. performs the function of fuzzification that converts input data into suitable linguistic values.

The fuzzy rule base comprises the knowledge of the application domain. It consists of a "data base" and a "linguistic rule base." The fuzzy inference machine has the capability of simulating human decision-making based on fuzzy concepts. The defuzzification interface performs

1. a scale mapping, which converts the range of values of output variables into corresponding universes of discourse, and
2. defuzzification, which produces a nonfuzzy decision output.

4.2 Review of Back-Propagation Fuzzy Logic System

The back-propagation fuzzy logic system (BPFLS) proposed by Wang and Mendel [16, 17] is used in our FBR picking procedure. The basic structure and operations of the BPFLS are described here. As stated in [16, 17], consider a multi-input single-output fuzzy system, e.g. n-input single-output

fuzzy system. The rules in the fuzzy rule base have the following form:

$$R_j : \text{IF } x_1 \text{ is } A_1^j \text{ and } x_2 \text{ is } A_2^j \text{ and} \ldots \text{and } x_n \text{ is } A_n^j, \text{ THEN } z \text{ is } B^j,$$

where $x_i, i = 1, 2, \ldots, n$, are the inputs to the fuzzy system, z is the output, A_i^j and B^j, $j = 1, 2, \ldots, K$, are linguistic terms, and K is the number of rules used in the fuzzy rule base. The fuzzy rules may come from training samples and human experts. The membership functions have the *Gaussian* form:

$$\mu_{A_i^j}(x_i) = \exp[-\frac{1}{2}(\frac{x_i - \overline{x}_i^j}{\sigma_i^j})^2], \tag{8}$$

where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, K$. Product inference is used in the fuzzy inference machine of the FLS. Centroid defuzzification is used for defuzzifying so that the output of the BPFLS is a number.

The fuzzy system f with product inference, centroid defuzzification and Gaussian membership functions has the form:

$$f(\underline{x}) = \frac{\sum_{j=1}^{K}(\overline{z}^j \prod_{i=1}^{n} \mu_{A_i^j}(x_i))}{\sum_{j=1}^{K}(\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}, \tag{9}$$

where $f : U \subset R^n \mapsto R$, $\underline{x} = (x_1, x_2, \ldots, x_n) \in U$, K is the number of rules in the fuzzy rule base, $\mu_{A_i^j}(x_i)$ is the Gaussian membership function in (8) and $\overline{z}^j$ is the point in the output space R at which μ_{B^j} attains its maximum. The fuzzy system f in (9) can be viewed as a three-layer feedforward network [16, 17].

The training procedure for a BPFLS [16, 17] is described next. For given N training input-output pairs $(\underline{x}^p, d^p)$, $\underline{x}^p \in U \subset R^n$ and $d^p \in R$, $p = 1, 2, \ldots, N$, design the fuzzy system (9) such that

$$e^p = \frac{1}{2}[f(\underline{x}^p) - d^p]^2 \tag{10}$$

is minimized.

The parameters to be determined for the BPFLS in (9), during the training phase, are:

1. output regional centers: $\overline{z}^j$, $j = 1, 2, \ldots, K$; and,

2. input regional centers and standard deviations: $\overline{x}_i^j$ and σ_i^j, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, K$,

where K is the number of rules in the fuzzy rule base and n is the dimension of input vector $\underline{x}$. For notational simplicity, let $f = a/b$, $a = \sum_{j=1}^{K}(\overline{z}^j y^j)$, $b = \sum_{j=1}^{K} y^j$ and $y^j = \prod_{i=1}^{n} \mu_{A_i^j}(x_i^p)$. The updating equations are [16, 17]

$$\overline{z}^j(k+1) = \overline{z}^j(k) - \alpha \frac{f - d^p}{b} y^j \tag{11}$$

where $j = 1, 2, \ldots, K$ and $k = 0, 1, 2, \ldots$ is the update (iteration) index;

$$\overline{x}_i^j(k+1) = \overline{x}_i^j(k) - \alpha \frac{f - d^p}{b}(\overline{z}^j - f) y^j \frac{x_i^p - \overline{x}_i^j(k)}{\sigma_i^{j\,2}(k)}, \tag{12}$$

where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, K$ and $k = 0, 1, 2, \ldots$; and,

$$\sigma_i^j(k+1) = \sigma_i^j(k) - \alpha \frac{f - d^p}{b}(\overline{z}^j - f) y^j \frac{(x_i^p - \overline{x}_i^j(k))^2}{\sigma_i^{j\,3}(k)}, \tag{13}$$

where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, K$ and $k = 1, 2, \ldots$. The initial conditions can be chosen by the method proposed in [16]. As stated in [16], suppose there are N training sample pairs, $(\underline{x}^p, d^p), p = 1, 2, \ldots, N$, and suppose that $N = m \times K$, where K is the number of rules in the BPFLS and m is an integer. First sort the pairs such that $d^i \leq d^{i+1}, i = 1, 2, \ldots, N-1$ for the sequence of pairs $[(\underline{x}^1, d^1), (\underline{x}^2, d^2), \ldots, (\underline{x}^N, d^N)]$. Then choose

$$\overline{z}^j(0) = \frac{1}{m} \sum_{k=1}^{m} d^{k+(j-1)m}, \tag{14}$$

and,

$$\overline{x}_i^j(0) = \frac{1}{m} \sum_{k=1}^{m} x_i^{k+(j-1)m}, \tag{15}$$

where $j = 1, 2, \ldots, K$, and $i = 1, 2, \ldots, n$. Next let $x_{imax} = max(x_i^1, x_i^2, \ldots, x_i^N)$ and $x_{imin} =$

$min(x_i^1, x_i^2, \ldots, x_i^N)$, $i = 1, 2, \ldots, n$, and choose

$$\sigma_i^j(0) = \frac{x_{imax} - x_{imin}}{K}, \tag{16}$$

where $j = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, n$.

4.3 Pre-Processing for FBR Peak Picking: Candidate FBR Peak Picking

First, we have to locate those peaks which are most likely FBR peaks. Then, the attributes of those candidate peaks can be fed into the fuzzy system to decide whether or not they are FBR peaks. Our procedure for picking candidate FBR peaks is:

1. Determine search starting points

   In the refraction prospecting shot record there is a *quiet* period before significant refraction energy arrives. The candidate FBR peaks should be located after some starting point. The candidate FBR peaks will include some peaks which are caused by noise (not by refraction energy) if we search these peaks from the first time point of a given trace record; therefore, it is necessary to exclude those peaks caused by noise. The search starting point is used as the starting time index for candidate FBR peak picking.

   It is possible to determine search starting points using several approaches. For example, they can be determined manually. It is not necessary to get very accurate estimates of them; what we need are approximate estimates of them. Veezhinathan, et al [14] mention that the search is started from the refraction waveform region; there is no detailed descriptions of what this means in their paper. McCormack, et al [9] use the criterion that the accumulated energy changes significantly along a trace to determine the search starting points.

   Our approach is described next for a given set of data which contains many shot records. In our simulations the seismic data are normalized to have unity maximum amplitudes. We first examine the *quiet* region before the refraction waveform arrives. For example in Fig. 5 the starting point for searching for candidate peaks is equal to $n = 10$. We adopted the same

criterion used by McCormack's group to estimate the search starting points. A threshold value (2.5 is used in our experiment) was chosen to test the ratio of slope change of the accumulated energy along the trace, because there should be a prominent increase of the accumulated energy when the refraction waveform arrives. This criterion works well when the background noise level is low; therefore, we also have to compute the noise levels of the traces before the refractive signal arrives. Only low-noise traces are used to determine the search starting points; then a least-squares line is employed to extrapolate the starting points for the high-noise traces. Those bad traces, whose background noise levels are much higher than those of normal traces, should be excluded from the FBR peak-picking procedure.

2. FBR peak picking

   The criteria used for locating candidate FBR peaks are:

   (a) the candidate FBR peak is a local maximum, and

   (b) its value is greater than a threshold (0.1 is the threshold in the trace with unity maximum magnitude).

   The flow chart for determining the search starting points and candidate FBR peaks is given in Fig. 4.

   Candidate FBR peak picking by the above algorithms for a shot record is shown in Fig. 5, where the candidate picked peaks are marked by small vertical bars above those peaks.

4.4 Discussion on Hilbert Attributes

According to Taner's paper [13], *power ratio* has very good FBR peak discriminating capability. We performed an experiment to observe the relationship between power ratio and FBR peaks. Our experiment is described as follows. Shot record # 1571 was chosen and is used in the simulations described in Section 4.6. Ten training traces were chosen for ploting the normalized power ratio curves. The result is shown in Fig. 6. In this set of test traces, there is an offset between the FBR

peak and the maximum of the power ratio curve. This suggests that we should modify the program, taking this offset into account; however, the picking-performance degraded when the shifted power ratio was used in our program. There are two possible reasons for this. First, correct offsets are not very easy to determine if the signal-to-noise ratio is low, because the optimal offsets are different for different traces. Second, the unshifted power ratio might still have significant magnitude for the FBR peak; hence, it could provide sufficient classification capability.

We are studying other Hilbert attributes in order to correctly utilize these features in our processing.

4.5 Intertrace Information Extraction

Intertrace information is taken into account by establishing a *piecewise linear guiding function* through manually-picked FBR peaks. For example, the piecewise linear guiding functions are shown in Figs. 7 and 8 for shot record # 1581 with two (traces # 29 and 48) and four (traces # 22, 29, 36 and 47) training traces. In Fig. 7 the guiding function contains two line segments; the first is a horizontal line segment from traces # 1 to 29 passing through the desired FBR peak in trace # 29 ; the second from traces # 29 to 48 passing through the manually picked FBR peaks in trace # 29 and 48. In Fig. 8 the guiding function consists of five line segments; the first is a horizontal line segment connecting traces # 1 to 22 passing through the desired FBR peak in trace # 22 ;the second to fourth are constructed by passing through the desired FBR peaks in traces # 22, 29, 36 and 47; the fifth is a horizontal line segment connecting traces # 47 and 48 passing through the desired FBR peak in trace # 47. The distance between a candidate peak and the guiding function is a fifth attribute fed into our BPLFS. The effects of this attribute on FBR peaks are examined in Section 4.7.

4.6 Relative Importance of Input Features

In FBR picking using Hilbert attributes, the four Hilbert attributes have different discriminating capabilities [13]. Some attributes are more concentrated than others if we observe their distributions

(see Figures 10 to 15 in [13]). This leads to the following questions:

1. Will the unconcentrated attributes (i.e., unimportant attributes) lead to negative effects in FBR classification?

2. Can we ignore the unimportant attributes without markedly reducing performance? Some simulation results are given in Section 4.8.

3. Is it possible to get better performance if we give different weights to the Hilbert features? If the answer is yes, how is this done?

We are presently considering how to adjust the shapes (widths) of the membership functions and the number of linguistic terms, so that we can weight attributes differently.

4.7 Simulation Results

We implemented the BPFLS [16, 17] using the features cited in [14] (see Fig. 9) for the reasons that our network size is more feasible than that of McCormack, et al [7], and, the Hilbert attributes seem easier to be transformed to linguistic rules. The procedures implemented in this section for FBR picking using BPFLS are summarized as follows.

1. Normalize the trace amplitudes for each trace so that they have unity maximum amplitudes.

2. Pick the candidate FBR peaks. This step is illustrated in Fig. 4.

3. Select the training traces, which have representative FBR peak waveforms. The continuity trend of the FBR curve could also be embedded into the proper choice of training traces.

4. Construct the piecewise-linear guiding function passing through the manually picked FBR peaks.

5. Compute the derived attributes for each training sample (three-peak group). In this Section we consider five attributes, namely peak amplitude, mean power level, power ratio, envelope slope and the distance to the piecewise-linear guiding function; hence, the dimension (n) of

the attribute vector is 15 (5 attributes times 3 peaks), as shown in Fig. 9. A training sample consists of the 15 attributes and their associated output (i.e., first break or no first break).

6. Let the desired output be 1 if the first peak of the training sample is claimed to be the FBR peak; otherwise the desired output is 0.

7. Choose the number of rules [K in (9)] used in the BPFLS. We used the following rule of thumb to do this. First, choose the number of rules equal to a fraction (e.g., 1/4 or 1/3) of the number of training samples. If the training procedure had difficulty converging we increase the number of rules to provide the BPFLS more freedom to approximate the input-output mapping characteristics of the training samples.

8. Determine the initial parameters (initial output regional centers, input regional centers and standard deviations) of the BPFLS by reordering the training samples according to the the values of the desired outputs and using Eqs. (14)- (16).

9. Update the parameters ($\overline{z}^j$, $\overline{x}_i^j$ and σ_i^j, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, K$) of the BPFLS by Eqs. (11) to (13). Suppose there are N training samples. The updating procedure is performed on an ordered set of the N training samples. One *sweep* of training contains N iterations (for N training samples); we update the parameters ($\overline{z}^j$, $\overline{x}_i^j$ and σ_i^j, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, K$) once per iteration.

10. Compute the sum of squared errors between the desired outputs and the actual outputs. If the error is less than a preset threshold (e.g., 0.01), the updated parameters are stored for later processing of all remaining traces; otherwise, the training procedure in step 9 is repeated.

11. The whole shot record is processed by the trained BPFLS. The sample (3-peak group) having the maximum output among all the samples of the current trace is claimed to have the FBR peak as its first peak.

Example 1. First Break Picking for Shot Record # 1571.

A first simulation was performed for a single shot record (shot record #1571), which contains 48

traces (Fig. 5). Five attributes, namely peak amplitude, mean power level, power ratio, envelope slope, and distance to the piecewise-linear guiding function, were considered in the processing. Two traces (# 7 and 45), three traces(# 7, 43 and 45) and four traces(# 7, 30, 43 and 45) were chosen as the training traces. Each training trace contains three 3-peak groups. For example in Fig. 5 there are six candidate FBR peaks (marked by small vertical bars above the peaks) in trace # 7. The first three-peak group contains the first, second and third marked peaks. The second three-peak group contains the second, third and fourth marked peaks. In a similar way we can select the remaining three-peak groups in other training traces as our training samples. In this example the FBR peak picked were: the first marked peak in trace # 7, the first in trace # 30, the second in trace # 43 and the third in trace # 45. The desired output of the BPFLS for the training sample is 1 if one of the above manually picked FBR peaks is the first peak in the training sample (three-peak group); otherwise, the desired output is 0 for the training sample. The picking results for these three cases are shown in Figs. 10, 11 and 12, respectively.

From the discussion (right after Eq. (10)) of unknown parameters to be determined in the training phase, the number of parameters (N_{para}) for the BPFLS is

$$N_{para} = K + 2 \times n \times K, \tag{17}$$

where K is the number of rules and n is the dimension of the input vectors fed into the BPFLS. In Eq. (17) the first term on the right-hand side denotes the output regional centers and the second term the input regional centers and standard deviations. In this example n equals 15. The number of training samples, number of rules and number of unknown parameters for the three cases of this example are summarized in Table 1.

From Figs. 10 to 12 we see that FBR picking by the BPFLS shows acceptable accuracy. The picking results for above three cases (Figs. 10 to 12) look exactly the same. The reason for this may be there is adequate information for FBR picking of this single shot record by providing the BPFLS with two training traces. The information from the third and fourth training traces is

Table 1: Summary of the numbers of parameters in *Example 1.*

no. of training traces	no. of training samples	no. of rules (K) (prespecified)	no. of parameters (N_{para})
2	6	2	62
3	9	9	279
4	12	6	186

redundant. In the three training traces case we used nine rules in the BPFLS, i.e., the number of rules equals the number of training samples. In this situation there is no training needed to match the input-output relationships. We are always able to construct a BPFLS to match the training samples if we have the same number of rules as the number of training samples. In fact the rules could be directly derived from the training samples in a one-to-one mapping manner. In this case the training time is very short (only need to determine the initial parameters); however, the processing time will be longer than that of a BPFLS with fewer rules. For example, in the four training traces case, six rules are used and the total number of design parameters is 186, which is less than that of the three training traces case (279). The processing time is determined by the number of rules. There is a trade off between training time and processing time.

The next example demonstrates the relationship between picking accuracy and the choice of training samples.

Example 2. First Break Picking for Shot Record # 1581.

The second experiment is performed by the same fuzzy logic system but for shot record # 1581 (see Fig. 13). In this simulation all five attributes were also incorporated in the processing. The picking results are shown in Figs. 14, 15 and 16 (training trace numbers are: #29, 48; # 29, 36, 42; and # 22, 29, 36, 47.). Each training trace contained four 3-peak groups. The construction of the training samples is as described in *Example 1.* The FBR peaks, manually picked for the training samples, are all the first marked peaks in the corresponding training traces. The results look acceptable except for some minor picking errors in Figs. 14 (trace # 34) and 15 (trace # 15). The numbers of

parameters in the BPFLS for different training conditions were computed by Eq. (17). The number of training samples, number of rules and number of unknown parameters for these three cases are summarized in Table 2.

Table 2: Summary of the numbers of parameters in *Example 2.*

no. of training traces	no. of training samples	no. of rules (K) (prespecified)	no. of parameters (N_{para})
2	8	2	62
3	12	6	186
4	16	8	248

In this example there is a picking error (trace no. 34 in Fig. 14) when only two training traces are used. Possible sources for this error are: the BPFLS does not have sufficient rules to properly describe the input-output relationship; the BPFLS does not have enough training samples to learn the mapping characteristics; or, the preprocessing algorithm (candidate FBR-peak picking) is not robust, i.e., it did not reject the first marked peak in trace no. 34 of Fig. 13. Another picking error occurs when we chose three training traces (# 29, 36 and 42). The picking result is adequate, however, when we used four training traces (# 22, 29 , 36 and 47). From this example we can see the effects of number of training samples and number of rules on the approximation capabilities. In fact, the minor picking errors are easy to correct by another post-processing procedure, which imposes the continuity constraint on the FBR curve.

Example 3. Effect of the Distance to Guiding Function on FBR Pickiing.

In this example we performed simulations for shot record # 1581 to observe the effect of the fifth attribute on FBR picking. The result is shown in Fig. 17 when the fifth attribute (distance to the guiding function) is ignored. The picking result is clearly better when the fifth attribute is used in the classification (compare Figs. 14 and 17); therefore, the fifth attribute, distance to the guiding function, plays an important role in FBR picking.

Example 4. Linguistic Information for FBR Picking.

In this example we demonstrate how the linguistic rules in a BPFLS help improve its performance. The simulations were also for shot record # 1581. The first break picking results shown in Fig. 17 did not provide adequate *lateral continuity.* The lateral continuity of a FBR curve can be transformed into the following simple linguistic rule:

If the distance between the candidate FBR peak and the guiding function is small, then it is likely to be a FBR peak.

The fuzzy set *FBR peak* is characterized by the following membership function:

$$\mu_{FBR}(d) = \begin{cases} 1 & \text{if } 0 \leq d \leq 3 \\ (-0.5)(d-5) & \text{if } 3 < d < 5 \\ 0 & \text{if } d \geq 5 \end{cases}$$

where d is the distance to the guiding function with the unit of the number of sampling intervals. In this case, the input-output relationship of the BPFLS (Eq.(9)) becomes

$$f(\underline{x}, d) = \frac{\sum_{j=1}^{K}(\overline{z}^j \prod_{i=1}^{n} \mu_{A_i^j}(x_i)) + (0.7)\mu_{FBR}(d)}{\sum_{j=1}^{K}(\prod_{i=1}^{n} \mu_{A_i^j}(x_i)) + (0.7)\mu_{FBR}(d)},$$

where $K = 2$, $n = 12$, d is the distance between the first peak of a 3-peak group and the guiding function; and 0.7 is the center of the membership function associated with "likely to be a FBR peak". The picking result is shown in Fig. 18, which indicates that a simple linguistic rule can improve the picking accuracy when we don't choose the important attribute, distance to the guiding function, as one of the attributes in the learning phase. Compare Figs. 14 and 18 to see that the two ways for treating the guiding function (attribute versus linguistic rule) give comparable results.

4.8 Sensitivity Analysis for BPFLS

McCormack, et al [7] performed a sensitivity analysis for their back-propagation neural network, the purpose of which was to understand the relative significance of each input on the output. They

established a sensitivity matrix S for a trained network. The $i-j^{th}$ entry of S is $\frac{\partial O_i}{\partial I_j}$, where O_i is the i^{th} output and I_j is the j^{th} input of the network. The jth input can be ignored without changing much of the mapping characteristics if the l_1 norm (sum of the absolute values) of the jth column of S is much smaller than those of other columns. The computation burden (in serial simulation) can be decreased if a significant fraction of the inputs can be eliminated.

A similar approach can be applied to our BPFLS in order to determine which inputs (i.e., features) can be ignored so as to reduce computation. For convenience, we follow the notation used in Section 4.2 just before Eq. (11). The sensitivity vector $\underline{s}$ is defined as

$$\underline{s} = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}]^T \tag{18}$$

where

$$\frac{\partial f}{\partial x_i} = \sum_{j=1}^{K} \frac{\partial f}{\partial y_j} \frac{\partial y_j}{\partial x_i} \tag{19}$$

$$\frac{\partial f}{\partial y_j} = \frac{\overline{z}^j - f}{b} \tag{20}$$

and

$$\frac{\partial y_j}{\partial x_i} = -y^j \frac{x_i - \overline{x}_i^j}{\sigma_i^{j2}}; \tag{21}$$

therefore,

$$\frac{\partial f}{\partial x_i} = \sum_{j=1}^{K} \frac{\overline{z}^j - f}{b} (-y^j) \frac{x_i - \overline{x}_i^j}{\sigma_i^{j2}} \tag{22}$$

where $i = 1, 2, \ldots, n$, and K is the number of rules.

We present two examples to illustrate the sensitivity analysis of our FBR picker.

Example 5. Sensitivity Analysis for Shot Record # 1571.

For shot record #1571, twelve training samples (four training traces, each containing three 3-peak groups) were selected to determine the parameters (centers of Gaussian membership functions for input and output variables, and standard deviations of membership functions for input variables) of the BPFLS; five attributes (peak amplitude, mean power level, power ratio, envelope slope and

the distance to the guiding function) were used in the training phase. Two rules were used. The dimension of the sensitivity vector in this example is 15 (dimension of the input vector to the BPFLS is 15 (see Fig. 9)). There are twelve sensitivity vectors for the twelve training samples here. The averaged sensitivity vector $\underline{\bar{s}}$ is obtained by averaging the absolute values of the sensitivity vectors for the twelve training samples ($N = 12$), and is:

$$\begin{aligned}
\underline{\bar{s}} &= \frac{1}{N}\sum_{p=1}^{N}|\underline{s}^p| \\
&= [\overline{\frac{\partial f}{\partial x_1}}, \overline{\frac{\partial f}{\partial x_2}}, \ldots, \overline{\frac{\partial f}{\partial x_{15}}}] \\
&= [0.40, 0.55, 2.75, 0.75, 0.06, \\
&\quad 0.57, 0.44, 1.11, 0.12, 3.46, \\
&\quad 0.10, 0.16, 1.05, 0.01, 0.36]
\end{aligned}$$

where $\underline{s}^p$ denotes the sensitivity vector for the *pth* training sample and $|\cdot|$ denotes the absolute values of the components of a vector.

Observe that the 3rd, 8th and 13th components of $\underline{\bar{s}}$ are large. These correspond to the power ratio feature, and suggest that, perhaps, power ratio, by itself, can do the job. The FBR picking results obtained by using all five attributes is shown in Fig. 12. A comparable result is shown in Fig. 19, when only one attribute, power ratio, is used in classification. We performed this simulation because in Taner's paper [13] he mentions that power ratio has very good discriminating capabilites for the FBR peak-picking problem. Observe, from Fig. 19, that results are very degraded as compared with those in Fig. 12; therefore, a single attribute (power ratio) does not work well. This fact is also mentioned in Taner's paper [13]. Subsequently, we performed the experiment by considering two attributes, power ratio and distance to the guiding function (a piecewise linear function that passes through the manually picked FBR peaks) (note that the 10th component in $\underline{\bar{s}}$ is very large). The result is shown in Fig. 20, which is very similar to the results obtained when five attributes were used (Fig. 12). The reason why the results in Figs. 12 and 20 are similar may be

that the other three attributes (peak amplitude, mean power level and envelope slope) only contain redundant information for FBR peak-picking, i.e., this information may also be contained in the power ratio.

Through the sensitivity analysis, the dimension of the input vector can be reduced, i.e., some components can be ignored without degrading the approximation capability, if these components are much less sensitive than other components. In this example all fifteen components in the averaged sensitivity vector have similar magnitude; hence, it appears that all five features are needed. Interestingly enough, performance is not degraded when we only used two attributes. The reason might be that the 5 attributes are highly correlated; therefore, some (rather than all) of them are sufficient for classification. Consider the following trivial example: if the dimension of the input vector is two and these two components are identical, then one of them is enough, the other one is redundant.

Example 6. Sensitivity Analysis for Shot Record # 1581.

For shot record #1581, eight training samples (two training traces, each containing four 3-peak groups) were selected to determine the parameters of the BPFLS; five attributes (peak amplitude, mean power level, power ratio, envelope slope and the distance to the guiding function) were used in the training phase. Two rules were used. We performed the same sensitivity analysis as in *Example 5.* The dimension of the sensitivity vector in this example is 15 (dimension of the input vector to the BPFLS is 15 (see Fig. 9)). There are eight sensitivity vectors for the eight training samples. The average sensitivity vector $\bar{\underline{s}}$ is obtained by averaging the absolute values of the sensitivity vectors for the eight training samples, and is

$$\begin{aligned}\bar{\underline{s}} &= [\overline{\frac{\partial f}{\partial x_1}}, \overline{\frac{\partial f}{\partial x_2}}, \ldots, \overline{\frac{\partial f}{\partial x_{15}}}] \\ &= [0.23, 0.21, 0.28, 0.19, 0.82 \\ &\quad 0.47, 0.27, 0.26, 0.04, 0.47, \\ &\quad 0.14, 0.10, 0.25, 0.12, 0.29]\end{aligned}$$

The FBR picking results obtained by using the five attributes is shown in Fig. 14. A comparable result is shown in Fig. 21, when only one attribute, power ratio, is used in classification. Observe, from Fig. 21, that results are also degraded as compared with those in Fig. 14; therefore, a single attribute (power ratio) again does not work well. Subsequently, we continued this experiment by considering two attributes, power ratio and distance to the piecewise linear guiding function. The result is shown in Fig. 22, which is similar to the results obtained when five attributes were used (Fig. 14). The reason why the results in Figs. 14 and 22 are similar is the same as mentioned in *Example 5.* Once again the performance is undegraded when we only used two attributes.

4.9 Comparison Between BPNN and BPFLS

The back-propagation neural network described in [14] was also implemented to solve the FBR picking problem. This network (see Fig. 23) has three layers. It has fifteen inputs (first layer), which contain the Hilbert attributes and the distances to the piecewise linear guiding function of a three-peak group; five neurons in the second layer (this number is suggested in [14]); and only one output z, which is used to indicate whether or not the first peak of this three-peak group is the FBR peak.

The back-propagation training procedure is summarized next. The forward operation of the network is to compute

$$z = f(\sum_{i=1}^{5} v_i y_i), \tag{23}$$

where v_i is the connection weight between y_i and z, y_i is the output of the i^{th} neuron in the second layer, and

$$f(s) = \frac{1}{1+e^{-s}} \tag{24}$$

$$y_i = f(\sum_{j=1}^{15} w_{ij} x_j), \tag{25}$$

in which x_j is the input to the j^{th} neuron of the first layer, w_{ij} is the connection from the j^{th} neuron

of the first layer to the i^{th} neuron of the seond layer. The objective function to be minimized is

$$E = \frac{1}{2}(t - z)^2 \tag{26}$$

where t is the desired output and z is the actual output. By using the chain rule, we can obtain the following updating formulas for the connection weights:

$$\begin{aligned} v_i(k+1) &= v_i(k) - \alpha \frac{\partial E}{\partial v_i} \\ &= v_i(k) + \alpha(t - z)z(1 - z)y_i \end{aligned} \tag{27}$$

and

$$\begin{aligned} w_{ij}(k+1) &= w_{ij}(k) - \alpha \frac{\partial E}{\partial w_{ij}} \\ &= w_{ij}(k) + \alpha(t - z)z(1 - z)v_i y_i(1 - y_i)x_j \end{aligned} \tag{28}$$

where $i = 1, 2, \ldots, 5, j = 1, 2, \ldots, 15, k = 1, 2, \ldots$, and α is the learning rate.

Example 7. FBR Picking by BPFLS and BPNN for Shot Record # 1581.

In this example we compare the performance of a FBR BPFLS and BPNN. We used shot record # 1581 (see Fig. 13), and two training traces, # 29 and 48. Each training trace contains four training samples (four three-peak groups); therefore, there are eight training samples. We picked the first peak of the first three-peak group in these two training traces as the FBR peak; hence, the desired output for the first training sample in each of above two traces is 1 and the desired outputs for the remaining six training samples are 0. The desired outputs for these eight samples are

$$\underline{t} = [1, 0, 0, 0, 1, 0, 0, 0]$$

The stopping criterion used in training is

$$error = \sum_{p=1}^{8}(t_p - z_p)^2 < 0.001$$

where the subscript p denotes the p^{th} training pattern.

1. Training Phase

   The output values for the BPNN and BPFLS after training, are:

   (a) BPNN

   $$output = [0.9828, 0.0095, 0.0020, 0.0034, 0.9841, 0.0023, 0.0114, 0.0145]$$

   final error = 9.9966×10^{-4}

   CPU time(SUN/4): 54 min 25 sec

   number of weights = $(15)(5) + (5)(1) = 80$

   (b) BPFLS (eight training samples and four rules)

   $$output = [1.0000, 0.0000, 0.0000, 0.0000, 1.0000, 0.0000, 0.0000, 0.0000]$$

   final error = 7.325×10^{-22}

   CPU time(SUN/4): 6 sec

   number of parameters = $15 \times 4 \times 2 + 4 = 124$ ($15 \times 4 \times 2$ comes from $\overline{x}_i^j, \sigma_i^j$, where $i = 1, 2, \ldots, 15, j = 1, 2, \ldots, 4$ and 4 comes from $\overline{z}^j$, $j = 1, 2, \ldots, 4$)

   Actually, no training was needed for the BPFLS of this example. After initial parameters were set, the sum of squared error is 9.063×10^{-22}, which is already less than the threshold in the stopping criterion (0.001); hence, no training is required. Here we did let the training proceed for one iteration, and the sum of squared error was further reduced to 7.325×10^{-22}.

2. Processing Phase

The outputs for the forty eight trace shot record(# 1581) are shown in Figs. 14 and 24. The BPFLS and BPNN exhibit almost the same results.

The processing times for BPNN and BPFLS to process a 48-trace record are

(a) BPNN: 1 min 43 sec.

(b) BPFLS: 1 min 51 sec.

These processing times did not involve any optimized coding. Clearly, the big advantage of the BPFLS over the BPNN is in the training time.

4.10 Study of the Generalization Capabilities of BPFLS and BPNN

The key issue in reducing human efforts in first break picking is to reduce the work of picking training samples for supervised learning type algorithms, e.g. BPFLS and BPNN. McCormack [8] reported that the performance for a shot record processed by his 2-D pixel image BPNN is usually degraded if the shot record does not contain any training samples used in the training phase.

We are studying whether the BPFLS suffers from this problem. One obvious problem with our approach is how to set up the piecewise linear guiding function if there are no training samples in the shot record. From our previous experiments, we know that the guiding function can help improve the picking results. The guiding function for a shot record which has no training samples can be obtained by averaging the guiding functions before and after the current shot record (in the order of the shot records), which were constructed from training samples. This procedure remains to tested.

We are also trying to incorporate linguistic information into the FBR picking as described in [15]. In case there aren't sufficient numerical training samples, linguistic information might help; however, linguistic information is not easy to obtain in the FBR picking problem. We are studying this.

4.11 Study of the Convergence Behaviors of BPNN and BPFLS

Convergence during training seems to occur much faster for the BPFLS than for the BPNN. Why? Possible answers are: (1) we have an excellent initialization procedure for the BPFLS, whereas the BPNN weights are initialized randomly, or (2) the BPFLS optimization problem is better conditioned than BPNN optimization problem. According to Dr. John Hauser (EE-Systems Dept., USC) to study the second possibility we should examine the Hessian matrices in gradient-type optimization procedures. Here we perform two experiments to do that.

Example 8. Effects of the Initializations of BPFLS and BPNN on Supervised Training for Shot Record # 1581.

We used shot record # 1581 with two training traces, # 29 and 48. Two rules and five attributes were used. There are eight training samples. The desired outputs for these eight samples are

$$\underline{t} = [1, 0, 0, 0, 1, 0, 0, 0]$$

The stopping criterion used in training is

$$error = \sum_{p=1}^{8} (t_p - z_p)^2 < 0.01$$

where the subscript p denotes the p^{th} training pattern.

1. Training Phase

   The output values for the BPNN and BPFLS after training, are:

   (a) BPNN (randomly chosen initial weights; one realization)

   $$output = [0.9452, 0.0294, 0.0076, 0.0116, 0.9457, 0.0082, 0.0333, 0.0425]$$

   final error = 0.01

   CPU time (SUN/4): 13 min 50 sec

number of weights = $(15)(5) + (5)(1) = 80$

The sum of squared errors in the training procedure is shown in Fig. 25; it took approximately 680 iterations for the training phase to be completed.

(b) BPFLS (eight training samples and two rules, and the initialization described in subsection 4.2)

$$output = [0.9338, 0.0061, -0.0192, -0.0180, 0.9392, -0.0052, 0.0168, 0.0041]$$

final error = 0.0091

CPU time (SUN/4): 21 sec

number of parameters = $(15)(2)(2) + 2 = 62$

The sum of squared errors in the training procedure is shown in Figure 26; it took 7 iterations for the training phase to be completed.

2. Condition Numbers of Hessian Matrices of Objective Functions for BPFLS and BPNN

(a) BPFLS

The objective function to be minimized by a BPFLS is shown in Eq. (10). For convenience it is shown here again;

$$e^p = \frac{1}{2}[f(\underline{x}^p) - d^p]^2 \tag{29}$$

where in this example $p = 1, 2, \ldots, 8$, and e^p is a function of the parameters of the BPFLS, namely centers and variances of the membership functions of the input variables and centers of the membership functions of the output variable. This parameter vector is denoted by $\underline{a}(k)$, where $k = 0, 1, 2, \ldots$, is the time index in the iterative minimization procedure, and $\underline{a}(0)$ is the initial value of the parameter vector, which is determined by the algorithm in Section 4.2.

The Hessian matrix, $H_p(\underline{a}(0))$ for the *pth* training sample, evaluated at $\underline{a}(0)$, is

$$H_p(\underline{a}(0)) = [\frac{\partial^2 e^p(\underline{a}(k))}{\partial a_i \partial a_j}]\Big|_{\underline{a}(k)=\underline{a}(0)} ; i,j = 1,2,\ldots,N_{para}, \tag{30}$$

where N_{para} is the number of parameters in BPFLS. The condition number of a matrix is the ratio between its largest singular value and its smallest one [10]. The condition number of the Hessian matrix provides information about the shape of the quadratic hypersurface around the point $\underline{a}(0)$ (or any other point at which $H_p(\cdot)$ is evaluated). The larger the condition number is, the more slender the quadratic hypersurface is, i.e., the slower the convergence rate will be in a gradient-descent algorithm.

The Hessian matrices $H_p(\underline{a}(0))$ were computed using a finite difference method for $p = 1,2,\ldots,8$; then, singular value decompositions were performed for these eight Hessian matrices. The condition numbers of $H_p(\underline{a}(0)), p = 1,2,\ldots,8$, are

$$\begin{aligned}\{cond(H_p(\underline{a}(0)))\} &= \{2.27\times 10^6, 4.45\times 10^7, 1.00\times 10^8, 1.34\times 10^7,\\ &\quad 1.17\times 10^6, 6.83\times 10^6, 7.91\times 10^6, 1.68\times 10^8\}.\end{aligned}$$

(b) BPNN

A similar procedure was also applied to our BPNN to observe its condition numbers; the results are

$$\begin{aligned}\{cond(H_p(\underline{a}(0)))\} &= \{1.02\times 10^{26}, 3.19\times 10^{25}, 8.13\times 10^{10}, 3.43\times 10^{10},\\ &\quad 7.51\times 10^{31}, 3.11\times 10^{11}, 1.45\times 10^{11}, 4.89\times 10^{11}\}.\end{aligned}$$

From the condition numbers of the two sets of Hessian matrices, it is obvious that the Hessian matrices of the BPNN are, in general, much larger than those of BPFLS. This may, therefore, be one of the reasons that the BPFLS converges faster than the BPNN.

3. Initial Condition Settings

Next, we turn our attention to the influence of initial conditions on the training procedures. In this example there were eight training samples to be approximated. The error evolution for the BPNN is shown in Fig. 25; the error evolution for the BPFLS (with two rules) is shown in Fig. 26. From this example, we see that the parameter initialization schemes described in Eqs.(14) to (16) for the BPFLS provides us with a good to excellent starting point for the training procedure. On the other hand, we don't have a good initialization for the BPNN, other than the random initial connection weights.

If four rules are used in the BPFLS (instead of two rules), the sum of squared errors is 9.063×10^{-22} immediately after using Eqs.(14) to (16) to set the initial conditions. In this case no training is needed to satisfy the training stopping criterion (error less than 0.01). This is surprising, because using four rules to approximate eight training samples, we expected to need training; however, the stopping criterion was satisfied just by using our initial conditions. This may, therefore, be a second reason why the BPFLS converges faster than the BPNN in the FBR picking problem.

4. Processing Phase

The outputs for the forty eight trace shot record(# 1581) are shown in Figures 14 and 27, from which we see that the BPFLS and BPNN exhibit similar results. The processing times for BPFLS and BPNN to process a 48-trace record are

(a) BPFLS: 1 min 41 sec.

(b) BPNN: 1 min 42 sec.

Once again we that the big advantage of the BPFLS FBR picker over the BPNN FBR picker is in the training time.

Example 9. Monte Carlo Simulation of the Effects of the Initializations of BPFLS and BPNN on Supervised Training for Shot Record # 1581.

In this example we seek an explanation for the faster convergence during training of the BPFLS versus the BPNN. Three experiments are described. In the first experiment, we use the subsection 4.2 parameter initialization procedure. In the second and third experiments we initialize the BPFLS and BPNN parameters randomly. In this way we will be able to test the hypothesis that the BPFLS converges much more rapidly than the BPNN because of the good initialization procedure for the former. This example was also for shot record # 1581, and used two training traces (Nos. 29 and 48) with two rules and two attributes (power ratio and distance to the piecewise linear guiding function). Objective function (10), which is minimized in BPFLS, is a function of the parameters of the BPFLS, namely centers and variances of the membership functions of the input variables and centers of the membership functions of the output variable. This parameter vector is denoted by $\underline{a}(k)$, where $k = 0, 1, 2, \ldots$, is the time index in the iterative minimization procedure, and $\underline{a}(0)$ is the initial value of the parameter vector, which is determined by the algorithm in Section 4.2. In this example the number of parameters to be determined in the training procedures of the BPFLS and BPNN are 26 and 35, respectively. If the condition numbers of the Hessian matrices with random initial parameters are not much larger than those obtained by our subsection 4.2 initialization scheme, then the faster rate of convergence of the BPFLS does not come from our initialization algorithm. Additionally, if the condition numbers of the BPFLS Hessian matrices are smaller than those of the BPNN, then we can conclude that the BPFLS training problem is better conditioned than the BPNN training problem.

1. BPFLS with proper initial parameters setting

   The initial parameters in the BPFLS are set by the algorithm in Section 4.2. The condition numbers of the Hessian matrices for 8 training samples are

$$\begin{aligned}\{cond(H_p(\underline{a}(0)))\} &= \{8.96 \times 10^4, 6.70 \times 10^4, 3.26 \times 10^6, 4.32 \times 10^6, \\ &\quad 3.11 \times 10^5, 1.80 \times 10^5, 2.42 \times 10^5, 1.62 \times 10^5\}\end{aligned}$$

2. BPFLS with random initial parameters setting

A Monte Carlo experiment was performed to observe the condition numbers of the Hessian matrices of the BPFLS objective functions. A Monte Carlo of 100 realizations was used. Fifty nine realizations had condition numbers with infinite values. We excluded these realizations and then computed averages and standard deviations of the remaining condition numbers. The average values and the standard deviations of the condition numbers for 8 training samples are

$$\begin{aligned}\{cond(H_p(\underline{a}(0)))\} = \{&6.04 \times 10^{25} \pm 3.73 \times 10^{26}, 7.59 \times 10^{14} \pm 3.63 \times 10^{15},\\ &1.99 \times 10^{29} \pm 1.26 \times 10^{30}, 8.55 \times 10^{13} \pm 5.20 \times 10^{14},\\ &7.39 \times 10^{23} \pm 4.67 \times 10^{24}, 6.54 \times 10^{29} \pm 4.14 \times 10^{30},\\ &5.64 \times 10^{25} \pm 2.59 \times 10^{26}, 4.30 \times 10^{34} \pm 2.72 \times 10^{35}\}.\end{aligned}$$

This Monte Carlo experiment demonstrates that our BPFLS initialization algorithm gives much better convergence results than random initialization.

3. BPNN with random initial connection weights

A Monte Carlo experiment was performed to observe the condition numbers of the Hessian matrices of the BPNN objective functions. A Monte Carlo of 100 realizations was used. The average values and the standard deviations of the condition numbers for 8 training samples are

$$\begin{aligned}\{cond(H_p(\underline{a}(0)))\} = \{&5.20 \times 10^{12} \pm 3.62 \times 10^{12}, 1.10 \times 10^{5} \pm 1.37 \times 10^{5},\\ &1.09 \times 10^{5} \pm 9.84 \times 10^{4}, 1.04 \times 10^{5} \pm 9.21 \times 10^{4},\\ &5.85 \times 10^{12} \pm 5.66 \times 10^{12}, 1.04 \times 10^{5} \pm 1.14 \times 10^{5},\\ &1.07 \times 10^{5} \pm 9.63 \times 10^{4}, 1.15 \times 10^{5} \pm 8.56 \times 10^{4}\}.\end{aligned}$$

This Monte Carlo experiment indicates that, on average, the BPNN training problem with random initial weights is more poorly conditioned than the BPFLS training problem when our Section 4.2 initialization scheme is used. On the other hand, the BPFLS with random initial parameters is, on average, more poorly conditioned than the BPNN with random initial weights. Although the condition numbers we computed in this example only inform us about the shapes of the quadratic hypersurfaces related to the optimization problems just after the initializations of the BPFLS and BPNN, they do help explain our earlier experimental results.

5 Discussions and Conclusions

The back-propagation fuzzy logic system (BPFLS) has been successfully applied to the first-break (FBR) picking problem, which is a pattern recognition problem in seismic data processing. The BPFLS can achieve the same level of accuracy as that of the back-propagation neural network (BPNN) for FBR picking. Furthermore, the BPFLS has much faster convergence rate in supervised learning than that of the BPNN. This suggests that a BPFLS should be the first candidate for supervised learning problems, especially in cases when training time is limited.

Five attributes, namely peak amplitude, mean power level, power ratio, envelope slope and distance to a guiding function, are inputs of our BPFLS. In some examples, similar performance can be attained when only two attributes (power ratio and distance to the guiding function) are used. This suggests that we should carefully study other attributes, e.g., instantaneous frequency and phase, which might carry essential information for FBR picking.

An approach which utilizes the original seismic signal instead of the derived attributes is not as flexible as feature-based approaches. There is potential to improve the performance of feature-based systems, because we can try lots of different features to improve the performance and to obtain more insights about seismic signals. Finally, our BPFLS is also capable of incorporating linguistic

rules to simplify the decision process, something we are presently exploring.

6 Future Work

In this section, we provide a list of additional work that needs to be done on the FBR picking problem using BPFLS.

1. Extend the simulations to multiple shot records

   The training samples will be chosen from different shot records and the trained BPFLS will be used to process the remaining shot records. According to Dr. Michael McCormack's suggestion [8], we should establish the BPFLS by selecting training samples from a limited number of shot records, then apply the trained system to a large number of shot records to test the generalization capability of BPFLS. Such experiments are presently underway. We expect to incorporate some helpful linguistic rules into the BPFLS to enhance the classification capability, in case there might not be sufficient numerical training samples or when the shot record does not include training samples [15]. In fact, there is much redundancy in the seismic survey data; how to extract sufficient useful information in the shot record for FBR picking is under study.

2. Perform experiments to test the robustness of BPFLS.

   According to Dr. McCormack's suggestion we should test our algorithm under different signal to noise ratio situations to test the robustness of BPFLS. In our feature based BPFLS there is a problem in the preprocessing step. The determination of those candidate peaks highly depends on the signal-to-noise ratio. The reliability of the preprocessing step plays a very important role in the succeeding processing steps leading to the classification of FBR peaks. We are trying to design a more robust preprocessing step to address this problem.

3. Multi-trace formulation of the fuzzy system

   How to efficiently utilize inter-trace correlations is the key point to improving the performance

of a fuzzy logic system for FBR picking. Our current approach is to utilize the piecewise linear guiding function that passes through manually picked FBR peaks. For those shot records which do not have training traces, the construction of guiding functions has to rely on a priori information and an approximation procedure that uses some known guiding functions from other shot records. We are also trying to design an efficient second stage processor to correct the wrong picks determined in the first stage of processing.

4. Determine the optimal number of rules used in BPFLS.

   The determination of the number of rules in our BPFLS is set in advance without meeting any optimality criterion. For example, at most N rules are needed to approximate N input-output pairs. Our current strategy for setting the number of rules in the BPFLS is to first let the number of rules equal $N/2$; if the training procedure can not converge, then we increase this number to N; otherwise, this number can be reduced again(e.g., $N/4$). Our goal is to find the smallest number of rules to approximate the N given training samples in order to reduce the computation time in the processing phase. We are trying to apply the orthogonal least squares method to solve this problem.

5. Develop some helpful linguistic rules to enrich the fuzzy system

   We need to understand more geophysical aspects of refraction prospecting and try to design experiments to obtain some *subjective* experiences for this problem. Furthermore, we need to carefully examine the relationships between the Hilbert attributes and FBR peaks. This will help us to derive linguistic rules that can be included in the FBR FLS system.

6. Try to produce an *artificial* expert to choose those training samples.

   No matter how the neural networks or BPFLS's work, human experts are required in the training phase. The ultimate goal of the FBR picking problem is to boost the ratio of automatic processing to human processing. The best situation is one in which we can have an unsupervised learning scheme, which is fully automatic.

7. Compare BPFLS and BPNN for FBR picking.

   We plan to implement the back-propagation neural network for FBR peaks picking to compare the performance with the BPFLS. We need to investigate the relationship between the number of training samples needed to get acceptable performance and the number of connection weights in the BPNN. The same problem also occurs in the BPFLS; we have to understand the relationship between the number of training samples needed and the number of parameters used in the BPFLS.

8. Weight the features arrording to their relative importance.

   We plan to study how to adjust the relative weights of different attributes (i.e., according to the relative importance of these features) in order to enhance the classification capability. Intuitively, this problem is related to the shape of membership functions. We could also adjust the dynamic ranges of different attributes to achieve this goal. We ignore the attribute when the dynamic range of the attribute is zero, i.e., this attribute is not considered in the decision making. We are trying to develop a systematic way to adjust the dynamic ranges for various attributes to reflect the relative importances for the attributes in decision making.

9. Test other data sets.

   We plan to test the BPFLS by using another set of data. Doing this will let us establish whether or not the results and conclusions obtained in our study to-date are data dependent.

References

[1] F. Aminzadeh, "Where are we now and where are we going?" in *Expert Systems in Exploration,* F. Aminzadeh and M. Simaan, Eds., Society of Exploration Geophysicists, Tulsa, OK., 1991, pp. 3-32.

[2] M. Dobrin and C. Savit, *Introduction to Geophysical Prospecting.* 4th ed., New York: McGraw-Hill, 1988.

[3] J. H. Justice, D. J. Hawkins and G. Wong, "Multidimensional attribute analysis and pattern recognition for seismic interpretation," *Pattern Recognition,* vol. 18, No. 6, pp. 391-407, 1985.

[4] G. Klir and T. Folger, *Fuzzy Sets, Uncertainty, and Information.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[5] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller, part I," *IEEE Trans. Syst., Man, and Cybern.,* vol. SMC-20, No.2, pp.404-418, 1990.

[6] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller, part II," *IEEE Trans. Syst., Man, and Cybern.,* vol. SMC-20, No.2, pp.419-435, 1990.

[7] M. McCormack, D. Zaucha and D. Dushek, "First break refraction event picking and seismic data trace editing using neural networks," submitted for publication.

[8] M. McCormack, *Private communications,* 1992.

[9] J. Mendel, "ARCO trip and follow-up telephone call record," Oct. 1991.

[10] *PC-MATLAB User's Guide*, South Natick, MA: The MathWorks, Inc., 1989.

[11] R. E. Sheriff, *Encyclopedic Dictionary of Exploration Geophysics*, third ed., Society of Exploration Geophysicists, Tulsa, OK., 1991.

[12] M. Taner, F. Koehler and R. Sheriff, "Complex seismic trace analysis," *Geophysics,* Vol. 44, pp. 1041-1063, 1979.

[13] M. Taner, "The use of supervised learning in first break picking," in *Proc. Symposium Geophysical Society of Tulsa,* E. Bielanski, Ed., 1988.

[14] J. Veezhinathan, D. Wagner and J. Ehlers, "First break picking using a neural network," in *Expert Systems in Exploration,* F. Aminzadeh and M. Simaan, Eds., Society of Exploration Geophysicists, Tulsa, OK., 1991, pp. 179-202.

[15] L. Wang and J. Mendel, "Generating fuzzy rules from numerical data, with applications," USC-SIPI Report 169, Dep. Elec. Eng.-Syst., Univ. Southern California, Los Angeles, CA, 1990; also, *IEEE Trans. System, Man, and Cybernetics,* vol. SMC-22, Sept. 1992.

[16] L. Wang, "Analysis and design of fuzzy systems," Ph.D. dissertation, USC SIPI Report 206, Dep. Elec. Eng.-Syst., Univ. Southern California, Los Angeles, CA, 1992.

[17] L. Wang and J. Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identifier," *Proc. IEEE International Conference on Fuzzy Systems,* San Diego, CA, 1992, pp.1409-1418.

[18] L. Wang and J. Mendel, "Fuzzy basis functions , universal approximation and orthogonal least squares learning," *IEEE Trans. Neural Network,* vol. , Sept., 1992.

[19] L. A. Zadeh, "Fuzzy sets," *Information and Control,* vol. 8, pp. 338-353, 1965.

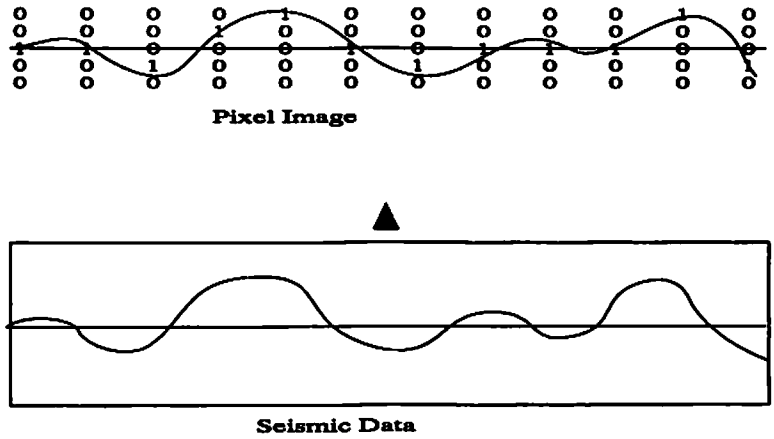

Figure 1: Pixel image representation of seismic data.

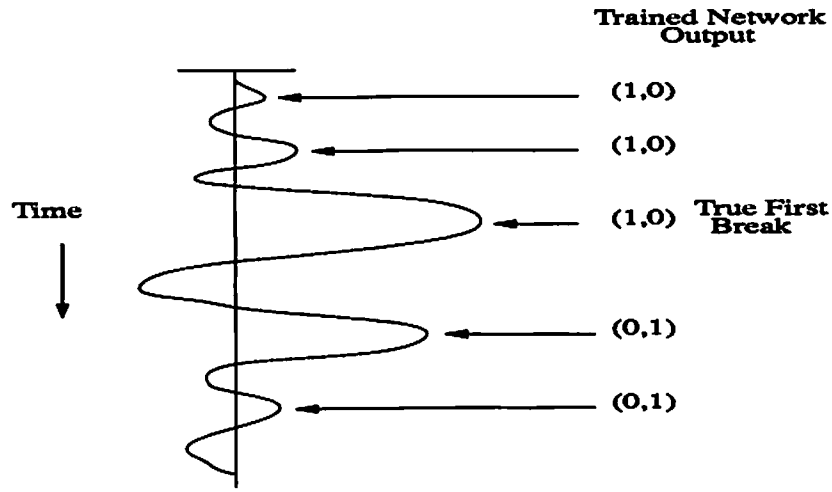

Figure 2: The desired outputs for trained potential first-break peaks.

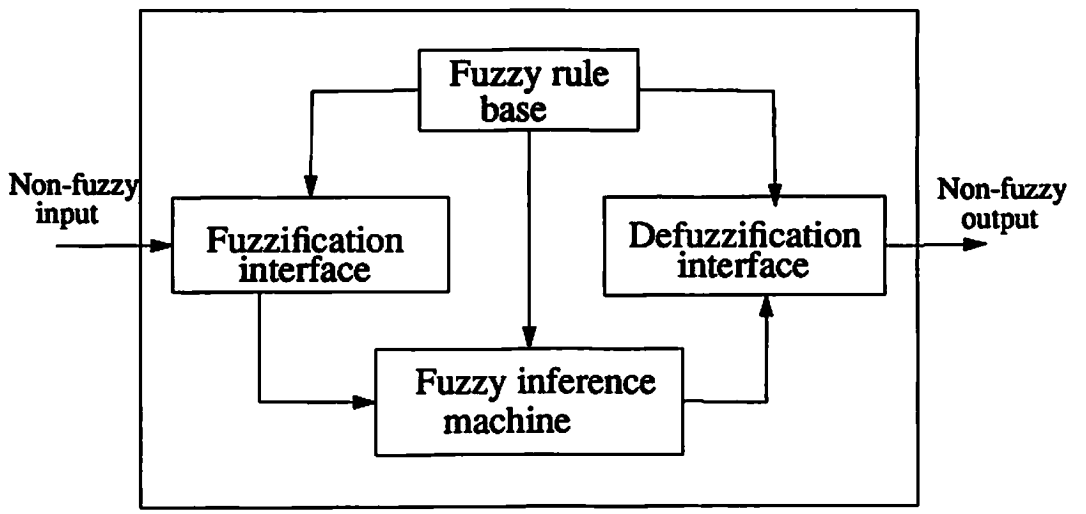

Figure 3: Basic configuration of fuzzy logic system.

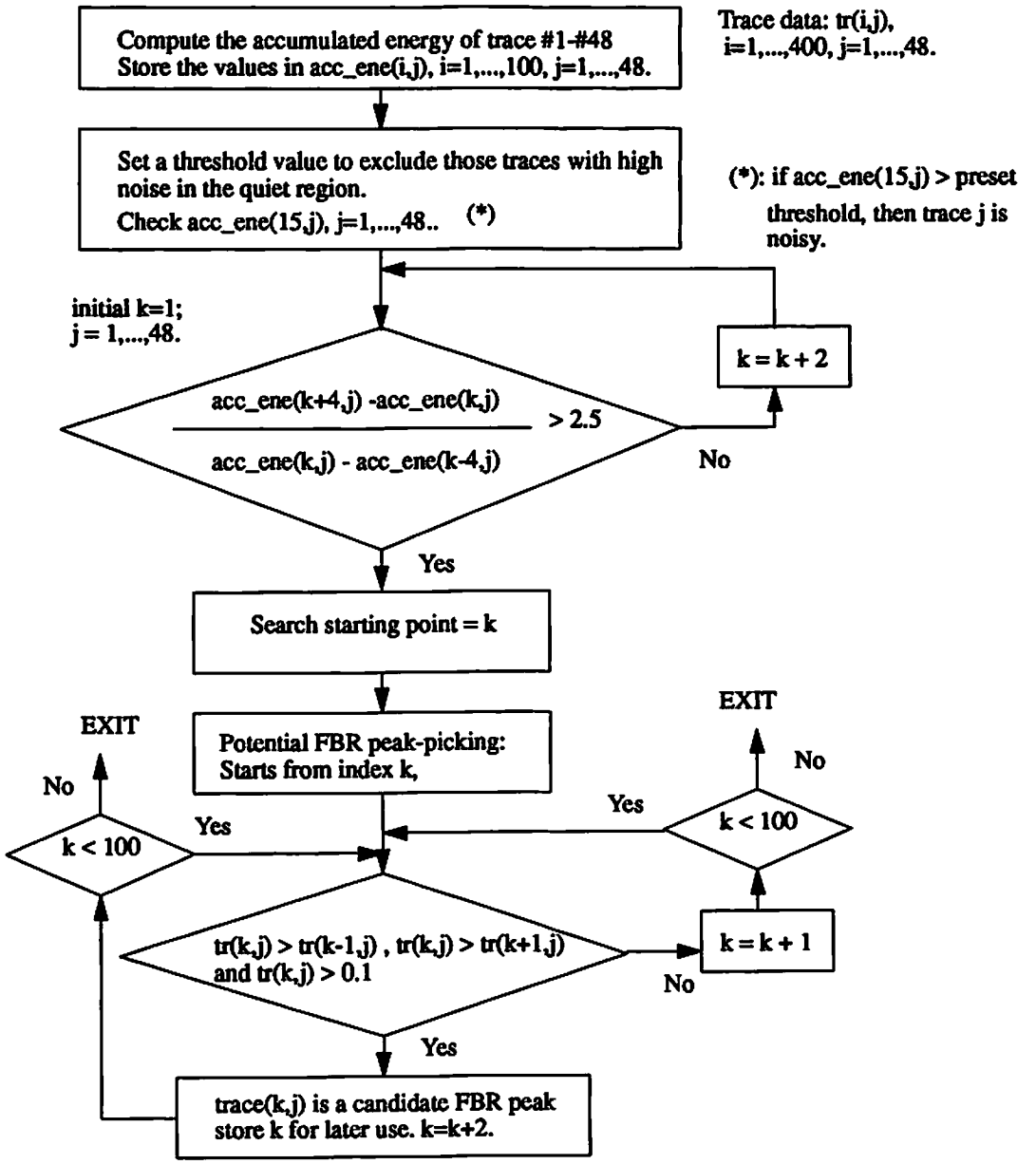

Figure 4: Pre-processing procedure for FBR peak picking. $acc\_ene(i,j) = \sum_{k=1}^{i} tr(k,j)^2$.

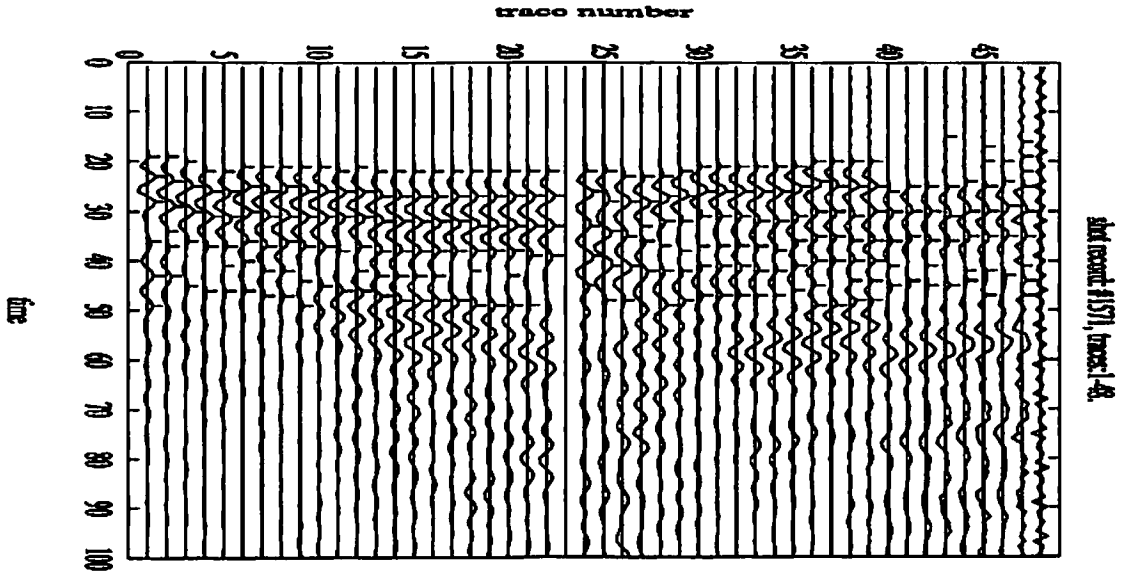

Figure 5: Pre-processing output of shot record # 1571.

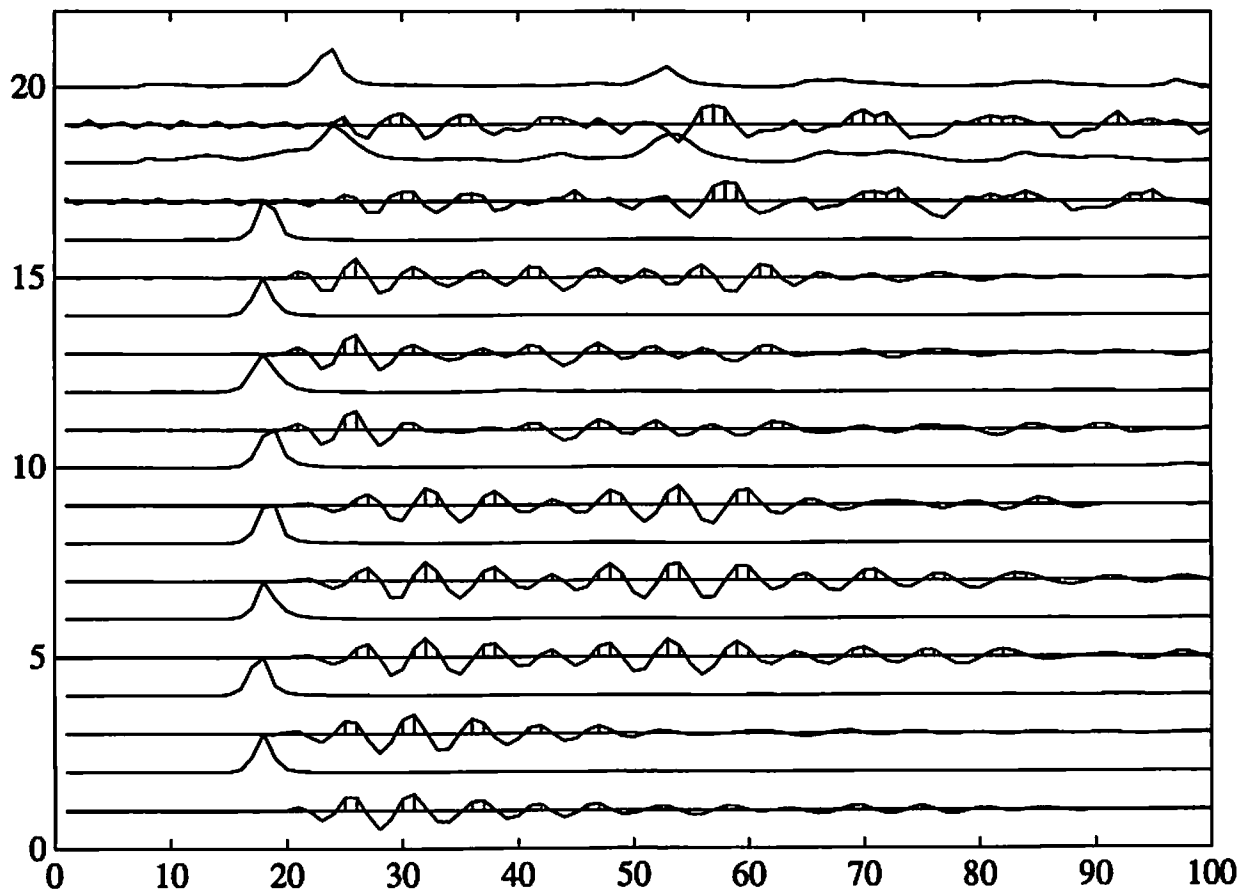

Figure 6: Offset between maximum of power ratio and FBR peak position. Power ratio plots appear directly above a seismic trace.

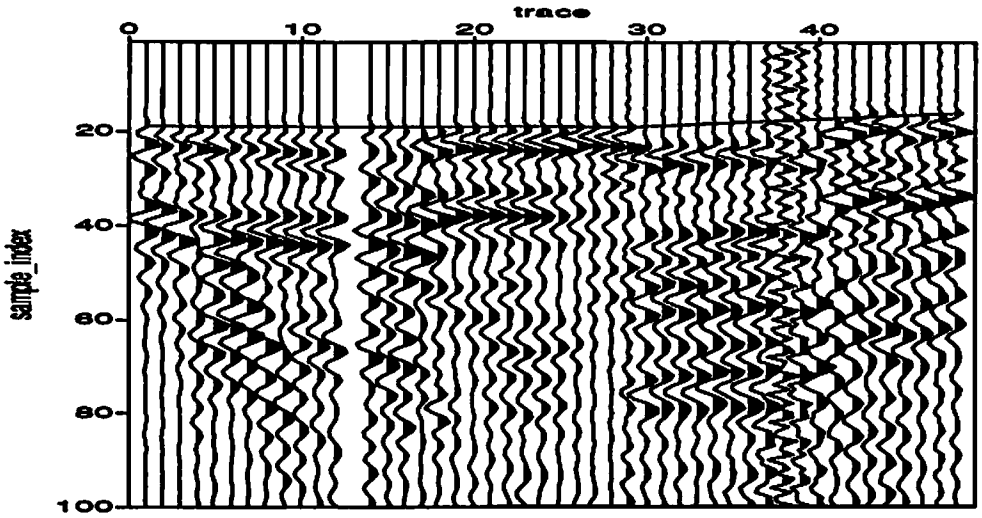

Figure 7: Piecewise linear guiding function for shot record # 1581 with two training traces (# 29 and 48).

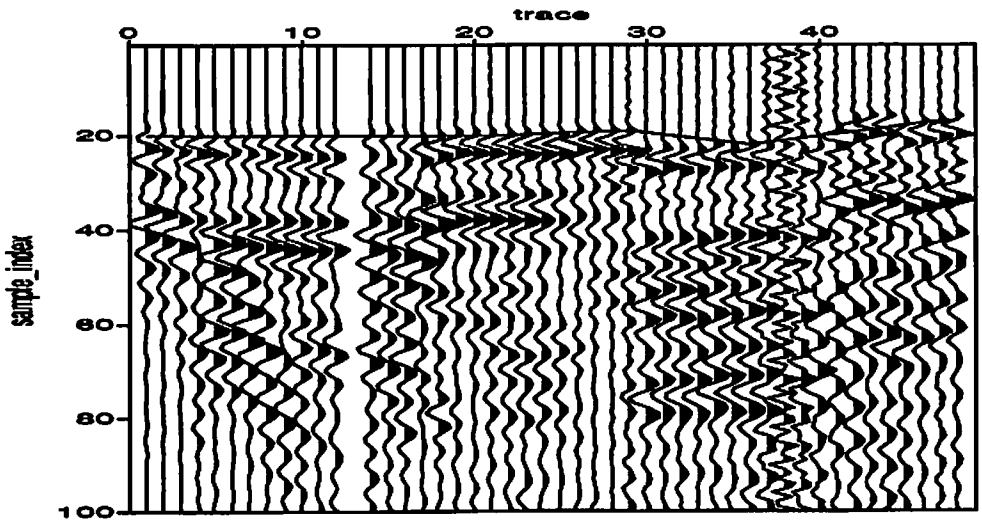

Figure 8: Piecewise linear guiding function for shot record # 1581 with four training traces (# 22, 29, 36 and 47).

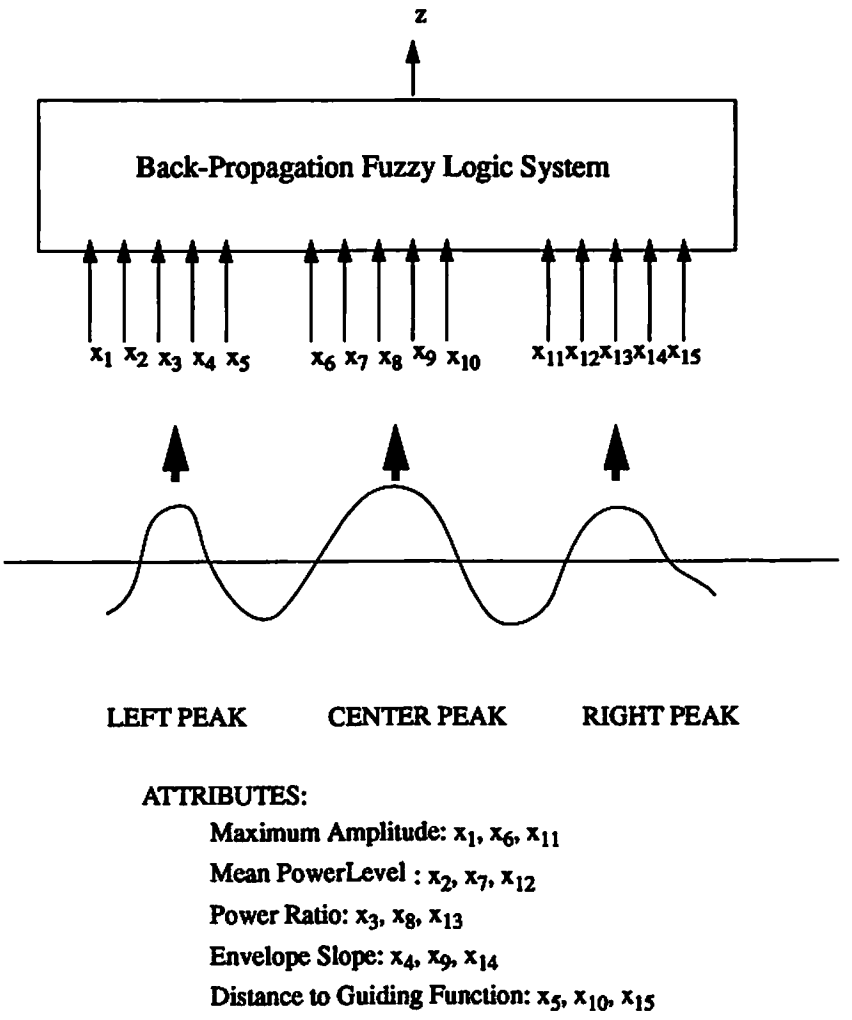

Figure 9: First break picking using fuzzy system based on Hilbert attributes and guiding function.

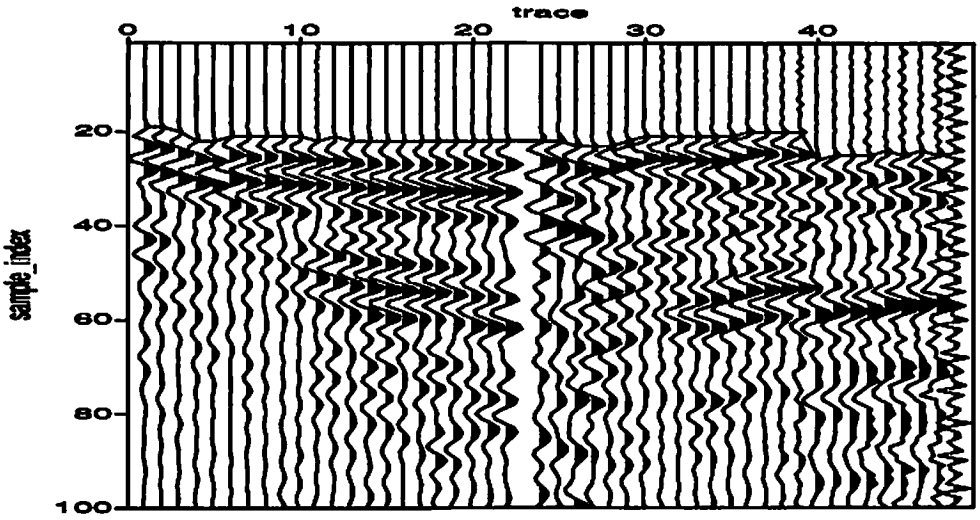

Figure 10: First break picking for shot #1571 by BPFLS with two training traces(#7,45). Five attributes and two rules were used.

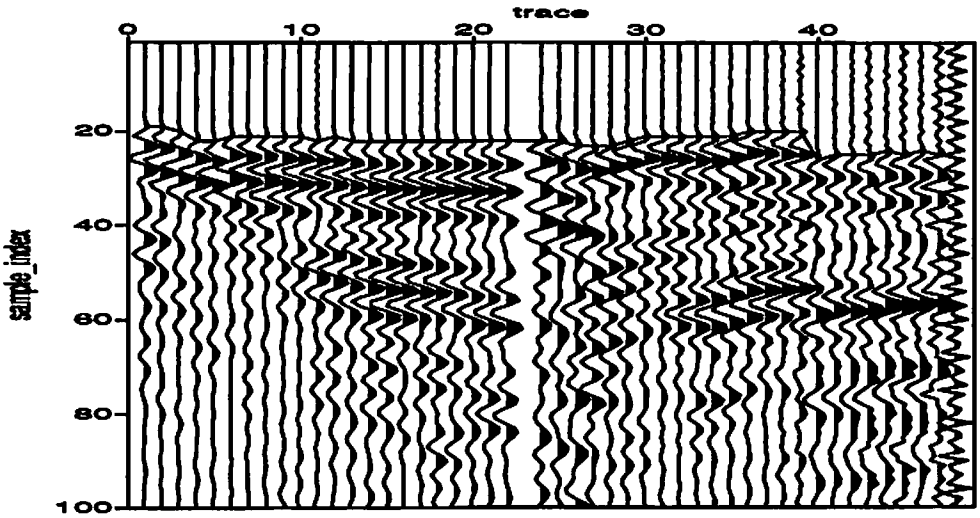

Figure 11: First break picking for shot #1571 by BPFLS with three training traces(#7,43,45). Five attributes and nine rules were used.

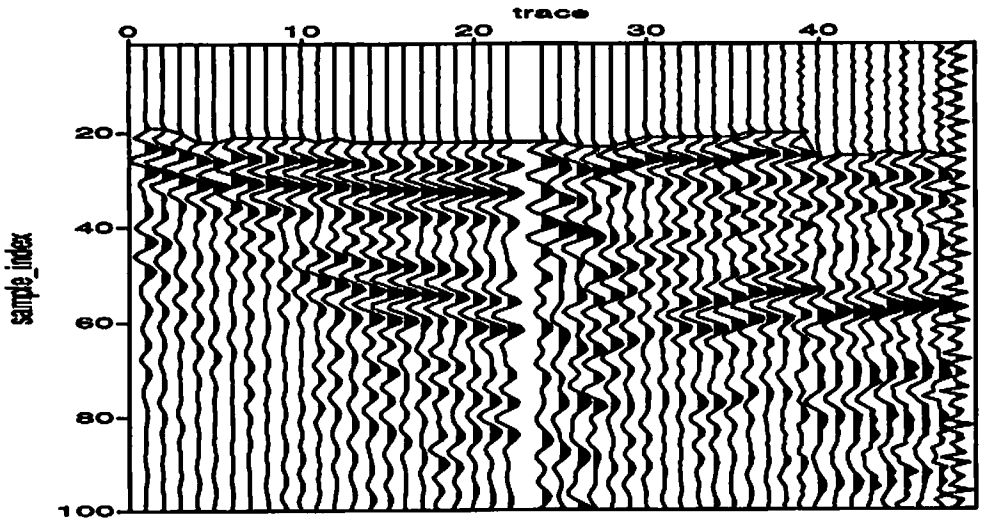

Figure 12: First break picking for shot #1571 by BPFLS with four training traces(#7,30,43,45). Five attributes and six rules were used.

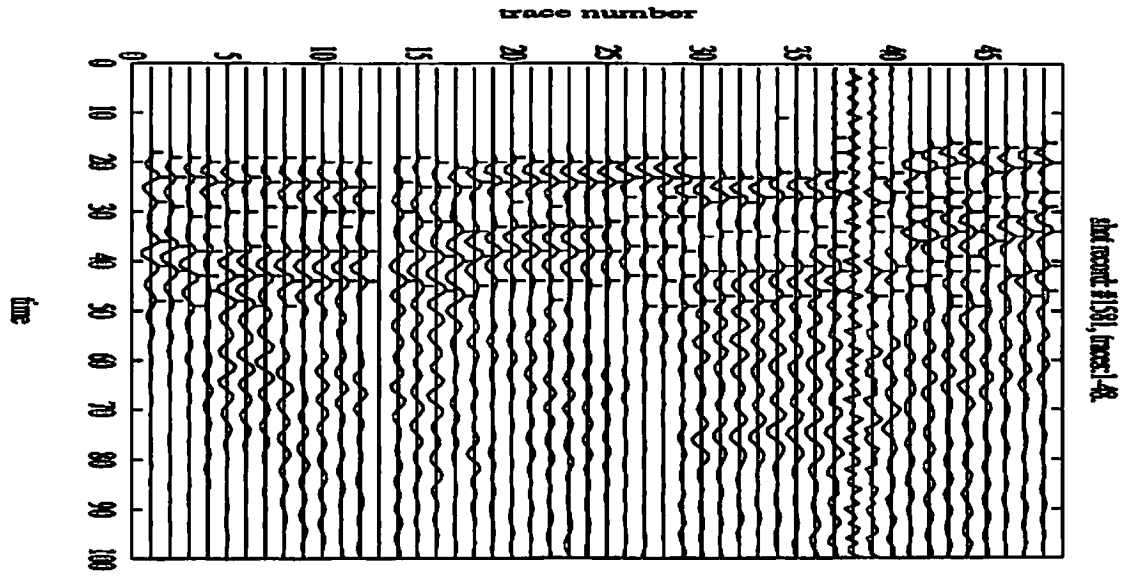

Figure 13: Pre-processing output of shot record # 1581.

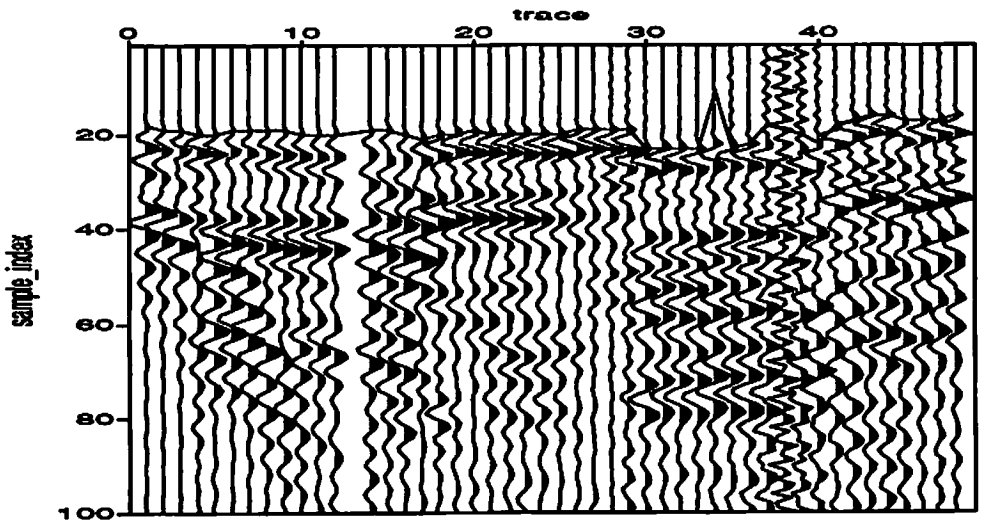

Figure 14: First break picking for shot #1581 by BPFLS with two training traces(#29,48). Five attributes and two rules were used.

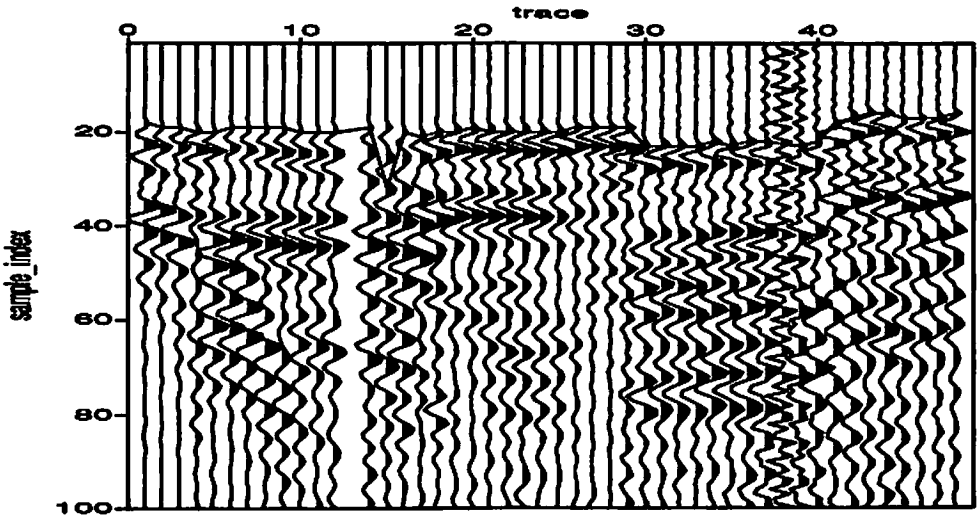

Figure 15: First break picking for shot #1581 by BPFLS with three training traces(#29,36,42). Five attributes and six rules were used.

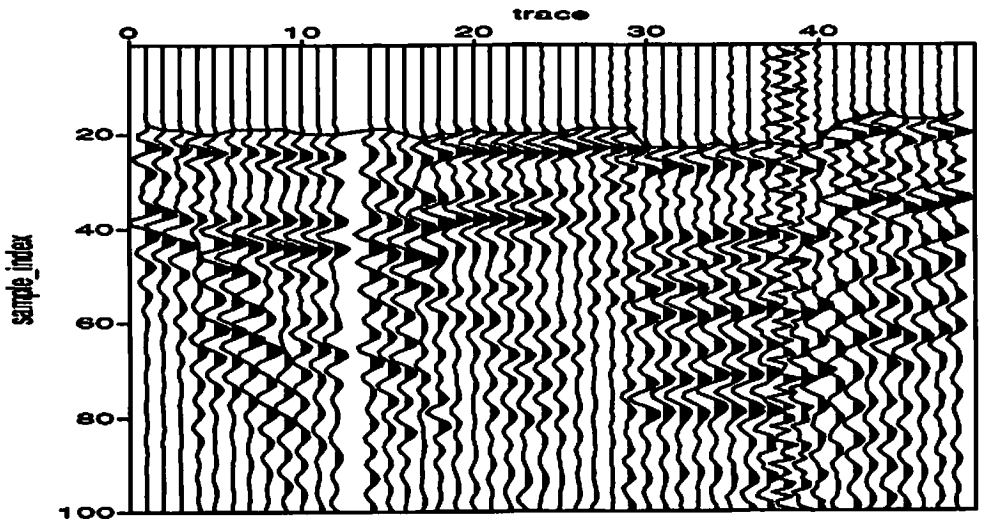

Figure 16: First break picking for shot #1581 by BPFLS with four training traces (#22,29,36,47). Five attributes and eight rules were used.

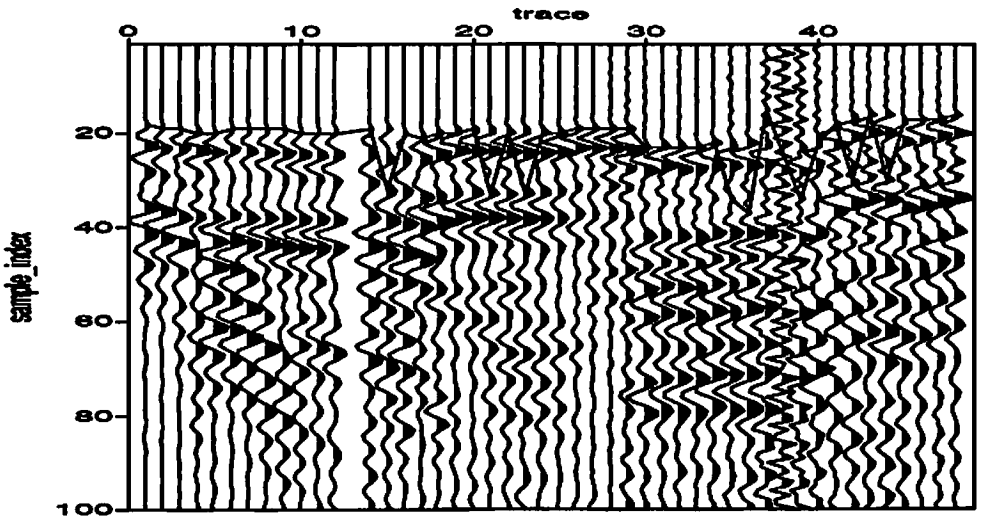

Figure 17: First break picking for shot #1581 by BPFLS with two training traces(#29,48). First four features were used (distance to the guiding function is ignored) in classification. Two rules were used.

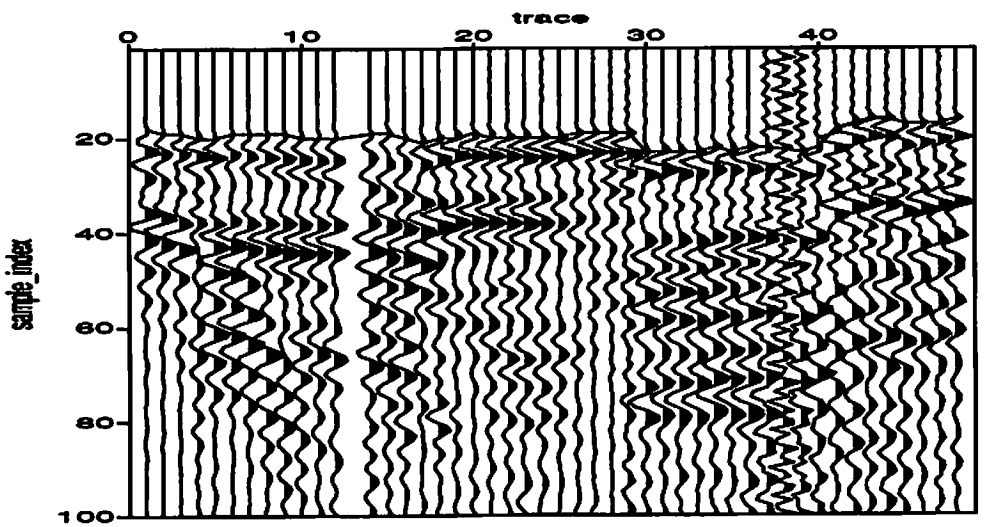

Figure 18: First break picking for shot #1581 by BPFLS with two training traces(#29,48). Only the first four attributes were used (without the distance to guiding function). Two rules were generated from the training samples. One linguistic rule (lateral continuity) was also incorporated.

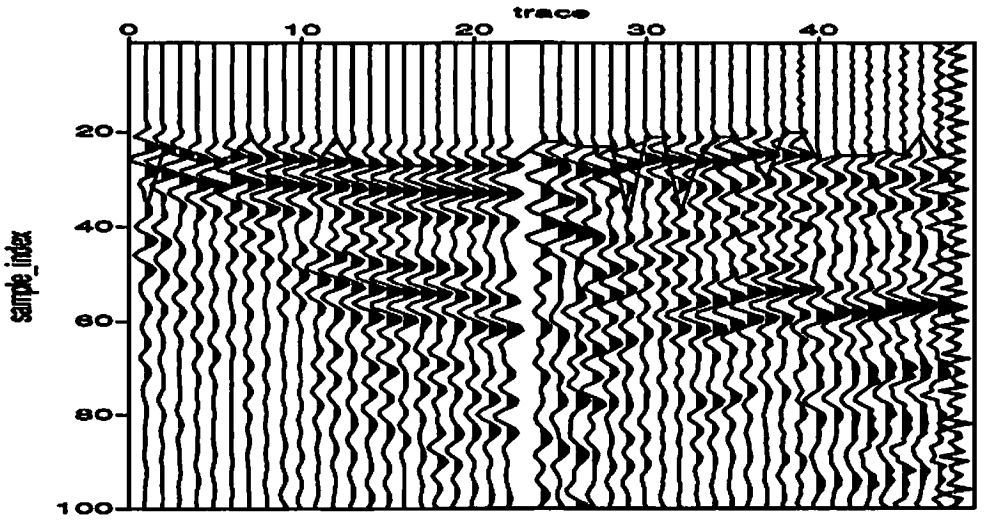

Figure 19: First break picking for shot #1571 by BPFLS with four training traces(#7,30,43,45). Only power ratio was used in classification. Six rules were used.

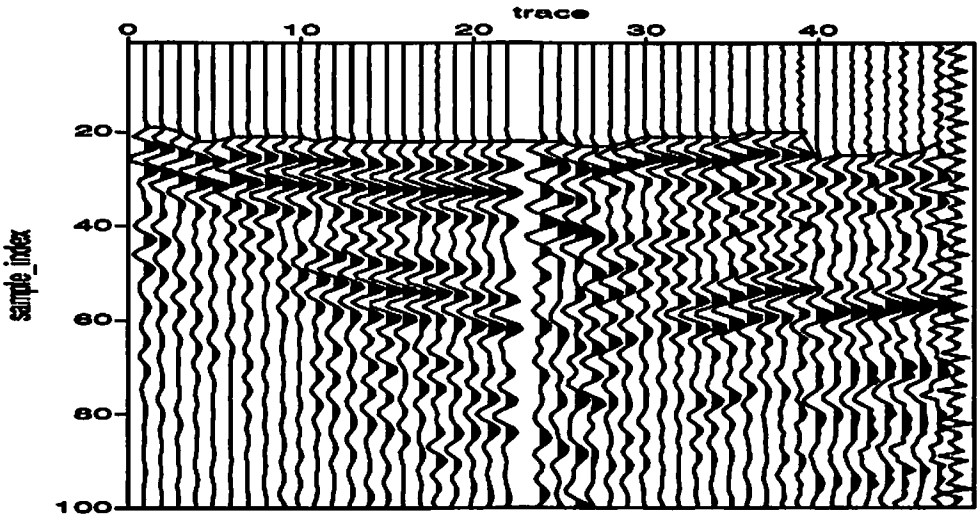

Figure 20: First break picking for shot #1571 by BPFLS with four training traces(#7,30,43,45). Two features, power ratio and distance to the guiding function, were used in classification. Two rules were used.

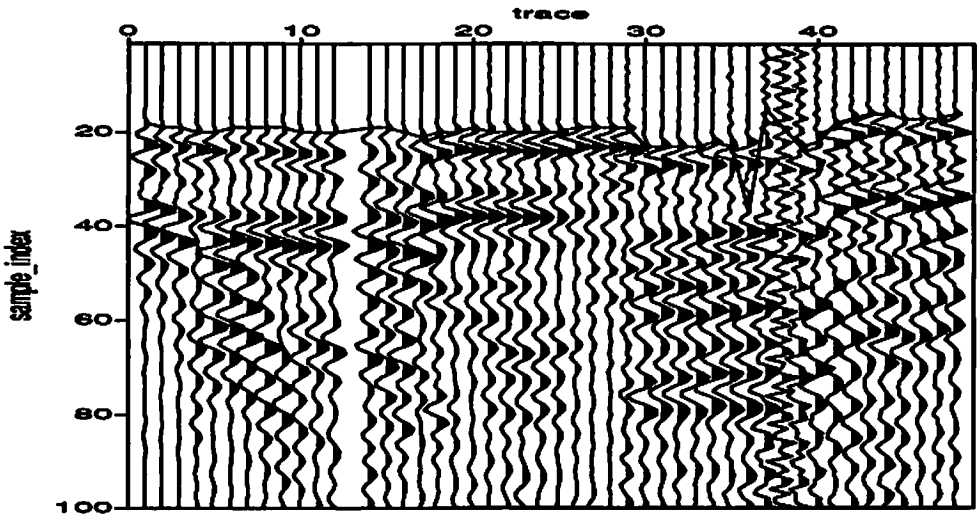

Figure 21: First break picking for shot #1581 by BPFLS with two training traces(#29,48). Only power ratio was used in classification. Two rules were used.

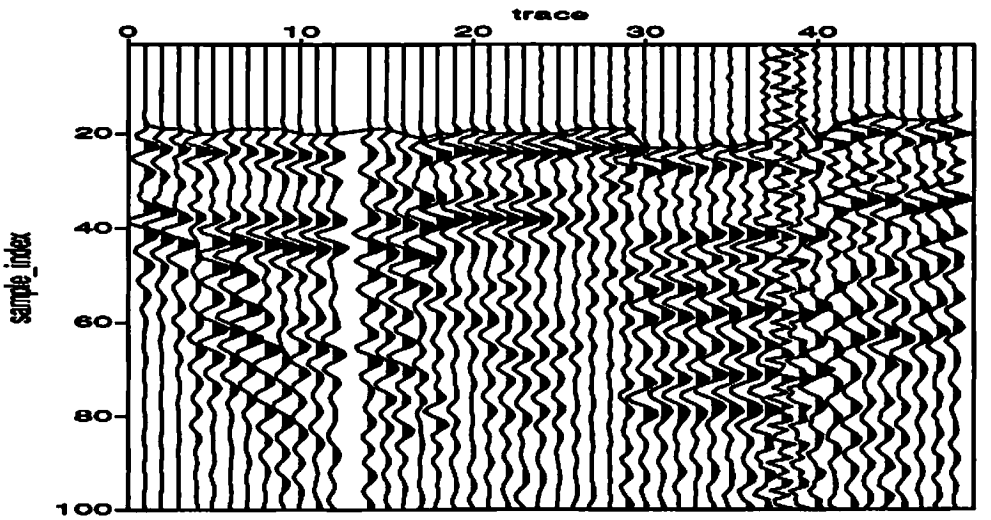

Figure 22: First break picking for shot #1581 by BPFLS with two training traces(#29,48). Two features, power ratio and distance to the guiding function, were used in classification. Two rules were used.

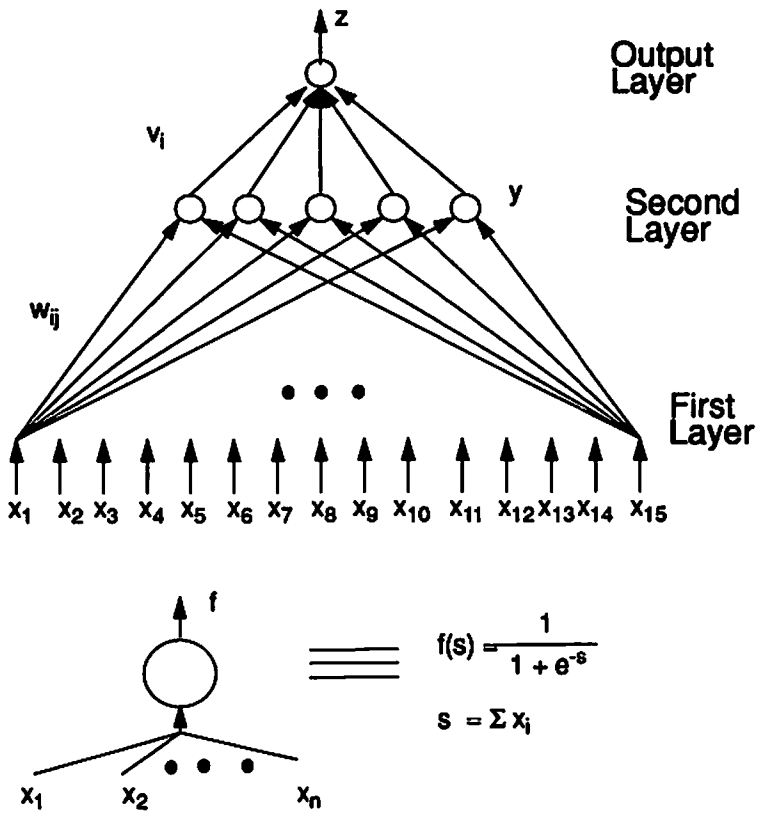

Figure 23: First break picking using back-propagation neural network.

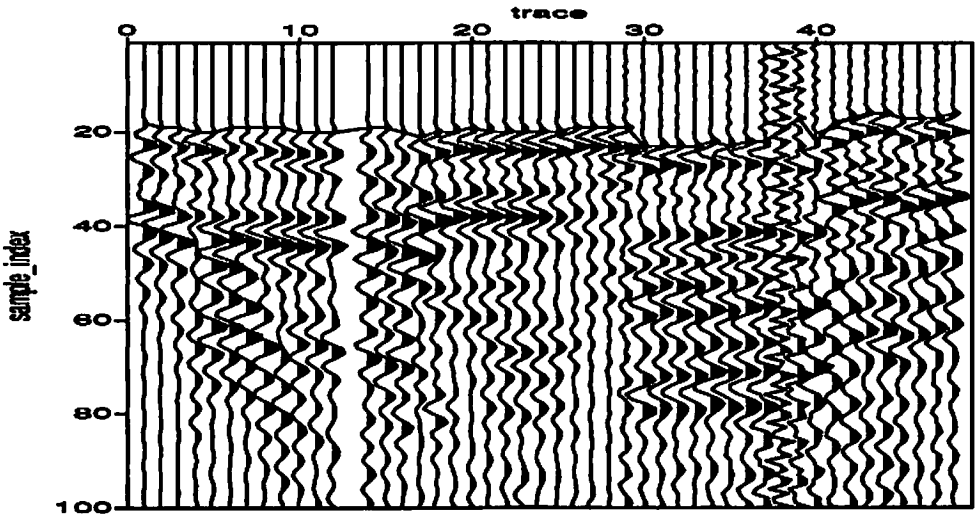

Figure 24: First break picking for shot #1581 by BPNN with two training traces(#29,48). Five attributes were used. Training stopped when sum of squared errors was less than 0.001.

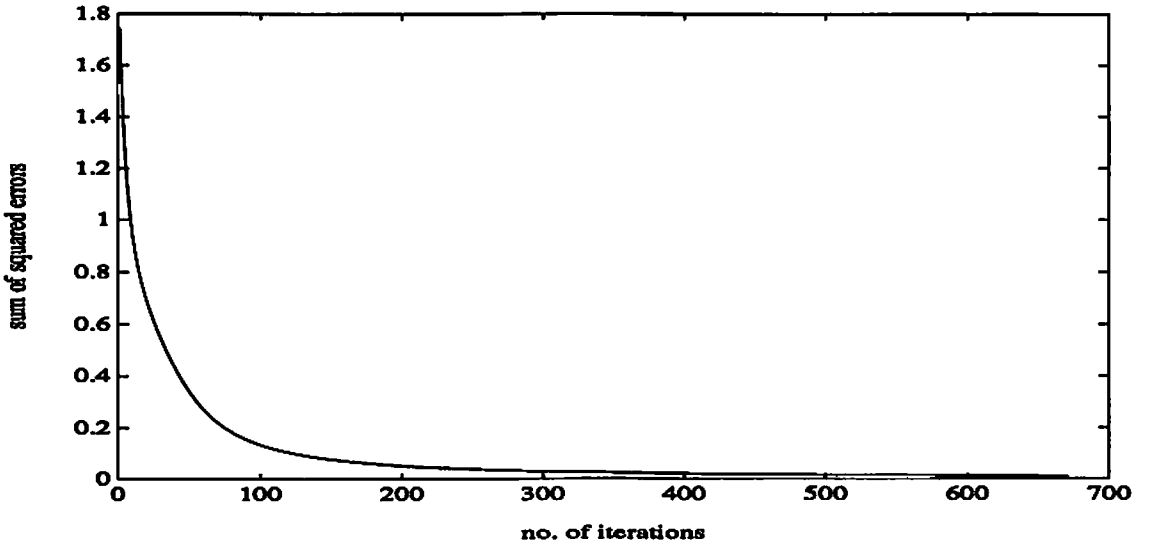

Figure 25: Sum of squared errors in training procedure for shot #1581 by BPNN with two training Traces(#29,48). Five attributes were used. Training stopped when sum of squared errors was less than 0.01.

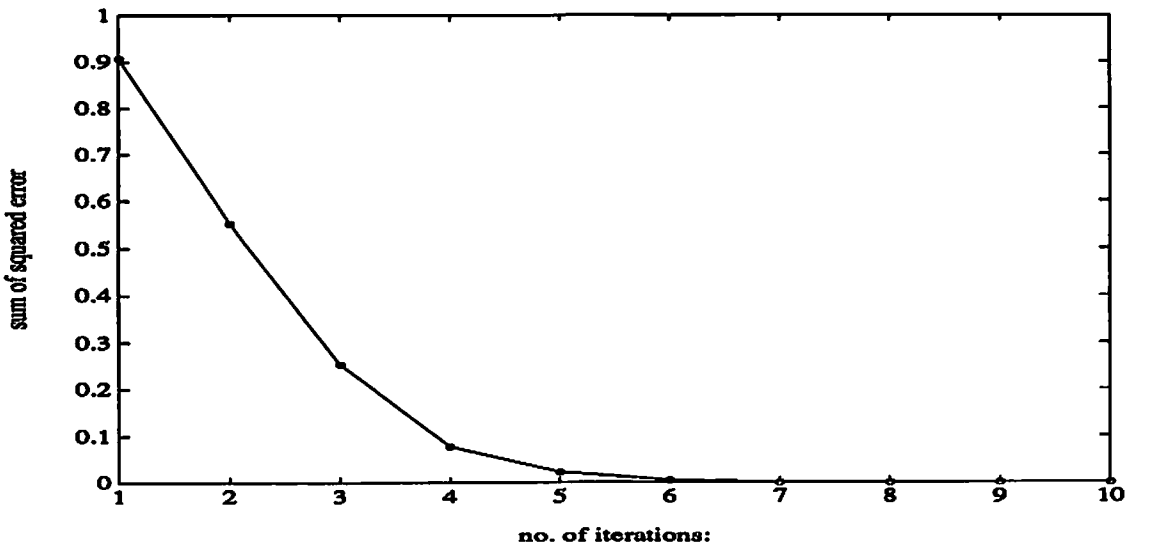

Figure 26: Sum of squared errors in the training procedure for shot #1581 by BPFLS with two training traces(#29,48) and two rules. Five attributes were used. Training stopped when sum of squared errors was less than 0.01.

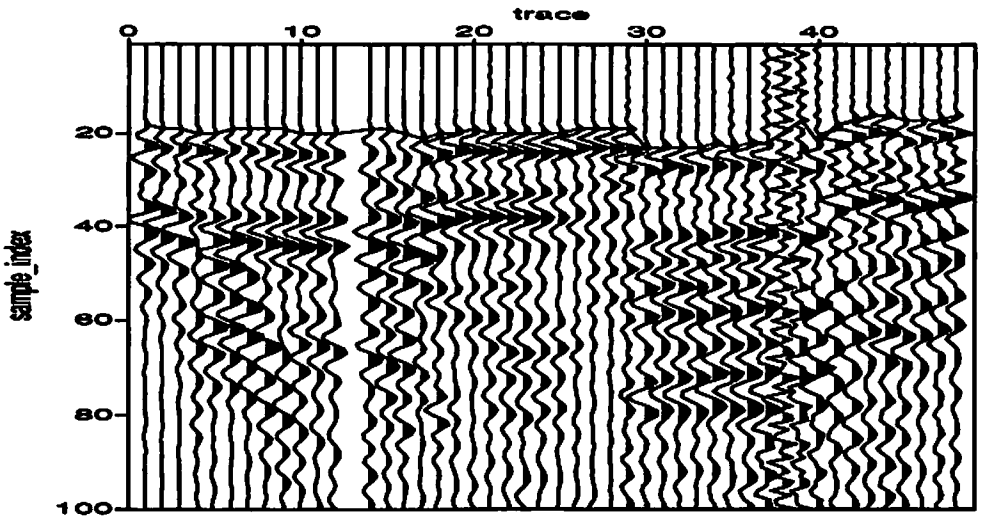

Figure 27: First break picking for shot #1581 by BPNN with two training traces(#29,48). Five attributes were used. Training stopped when sum of squared errors was less than 0.01.