Seismic Signal Classification Using Perceptron With Different Learning Rules

Kou-Yuan Huang, *Senior Member, IEEE*, Jiun-Der You, and Fajar Abdurrahman

***Abstract*—Perceptron is adopted to classify the Ricker wavelets and to detect the seismic anomaly in a seismogram. Three learning rules are used in the training of perceptron to solve the decision boundary. The optimal learning-rate parameter is derived. The lower and upper bounds of the learning-rate parameter are derived. It can provide that the learning can converge when the parameter is within the range. The normalized learning rule is derived also. Combining learning rules, a fusion learning rule is proposed. In the experiments, these rules are applied to the detection of a seismic anomaly in the simulated seismogram and to compare the convergence speed. The fusion learning rule has the fastest convergence and is applied to the real seismogram. The seismic anomaly can be detected successfully. It can improve the seismic interpretation.**

***Index Terms*—Learning-rate parameter, learning rule, perceptron, Ricker wavelet, seismic anomaly, seismogram.**

I. Introduction

PATTERN recognition methods have ever been used to analyze seismic exploration data, earthquake data, and geophysical events [1]–[18]. The methods included statistical methods, syntactic methods, and neural networks. In the 1970s, two major oil companies were successful in predicting the occurrence of gas sand zone from offshore seismic reflection data [19]. The prediction was based on a seismic anomaly in the high amplitude of reflection that was referred as the bright spot. There are many hydrocarbon indicators in the seismic data. The major physical indicators are high amplitude due to high reflection coefficient and low-frequency content at the reflection of the top of the gas sand zone. And there is a polarity reversal in the wavelet for the negative reflection coefficient [19], [20]. Those three attributes are the seismic anomaly.

Most of the learning rules had the weight adjustment in neural networks, for example, the backpropagation learning rule in a multilayer perceptron [21], [22] and in convolutional neural network (CNN) of deep learning [23], [24]. The property of the learning-rate parameter in the learning rules had no further analysis.

Manuscript received February 28, 2020; revised June 7, 2020 and August 15, 2020; accepted September 12, 2020. Date of publication September 23, 2020; date of current version October 9, 2020. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 107-2221-E-009-146. *(Corresponding author: Kou-Yuan Huang.)*

Kou-Yuan Huang and Jiun-Der You are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: kyhuang@cs.nctu.edu.tw; jeundery@gmail.com).

Fajar Abdurrahman is with the EECS International Graduate Program, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: fajarabdrhmn@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2020.3026011

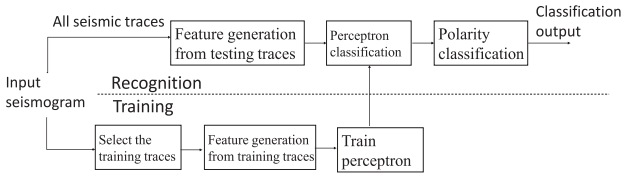


Fig. 1. Block diagram of a seismic pattern recognition system using a perceptron.

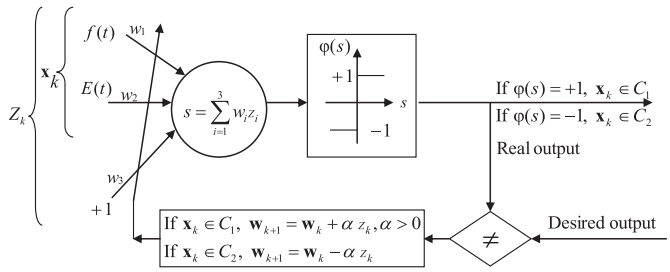


Fig. 2. Perceptron model.

In order to analyze the seismic patterns, a pattern recognition system using a perceptron is presented in Fig. 1. It is composed of two major parts: training and recognition. The training part consists of the selection of training traces, feature generation, and training by perceptron. The recognition part consists of the same feature generation and the classification by perceptron. The features are envelope and instantaneous frequency. Then, after the classification, the polarity is calculated. The result can show the detection of the seismic anomaly.

Perceptron is a supervised classification method in pattern recognition and neural networks [25]–[29]. The model is shown in Fig. 2. It can classify two classes. Here, the perceptron is adopted for the detection of a seismic anomaly. The major properties of the seismic anomaly are high amplitude and low frequency. Complex signal analysis can be used to extract such anomaly [10], [12], [30]. The envelope can describe the outer shape of the signal. The instantaneous frequency can extract the internal frequency of the signal. They are generated from the seismic signal as the features and used in classification.

Three learning rules are used in the training of perceptron. They are the fixed-increment, absolute correction, and fractional correction learning rules [25], [27]. The learning-rate parameter of each learning rule is in the analysis. Except these rules, the optimal learning-rate parameter is derived. The lower bound and the upper bound of the learning-rate parameters are derived. A

normalized learning rule is derived also. Combining the learning rules, a fusion learning rule is proposed here.

In the experiments, these rules are applied to the detection of a seismic anomaly in the simulated seismogram and the convergence speeds are compared. Then, the learning rule with the fastest convergence speed is applied to the detection of a seismic anomaly in the real seismogram.

Ricker wavelets are usually used in the seismic data analysis [10], [12], [19], [31]. The major part of the energy of wavelets in real seismic data is close to the central part of the zero-phase Ricker wavelet [10], [12]. So, it is used in the simulated seismogram. Here, a 20-Hz zero-phase Ricker wavelet is used to simulate the abnormal class. There may be phase changes in the wavelets in seismic data. But through the envelope and instantaneous frequency processing, the central part of the zero-phase Ricker and the front part of the minimum-phase wavelet can have the same abnormal properties. Considering frequency attenuation, the 30-Hz and 25-Hz zero-phase Ricker wavelets are used to simulate the normal class. The 25-Hz zero-phase Ricker wavelet is at the deeper layer. At first, Ricker wavelets are classified in a seismic trace. Then, they are classified in the simulated seismogram.

II. Feature Generation in Seismic Signal

A. Envelope and Instantaneous Frequency

Envelope and instantaneous frequency are generated from the original signal through complex signal analysis [10], [12], [30].

The complex signal has a real part that is the original signal $s(t)$ and an imaginary part $\hat{s}(t)$ that the signal $s(t)$ passes through the Hilbert transform. The complex signal is denoted as $\Psi(t)$

$$\Psi\ (t) = \ s\,(t) + j\hat{s}\,(t)\,.$$

Envelope and instantaneous frequency are denoted as $E(t)$ and $f(t)$.

$$\mathrm{ENV}\ \{s\,(t)\} = \ E\ (t) = |\ \Psi\,(t)| \ = \sqrt{s^2\,(t) + \hat{s}^2\,(t)}$$

$$f\ (t) = \frac{1}{2\pi}\ \frac{d\theta\,(t)}{dt} = \frac{1}{2\pi}\ \frac{d}{dt}\tan^{-1}\frac{\hat{s}\,(t)}{s\,(t)}$$

$$= \frac{1}{2\pi}\ \frac{s\,(t)\,\frac{d\hat{s}(t)}{dt} - \hat{s}\,(t)\,\frac{ds(t)}{dt}}{s^2\,(t) + \hat{s}^2\,(t)}.$$

The instantaneous frequency may become large if the denominator is an approach to zero. In order to avoid this case, a threshold is set to 10^{-4}. If the denominator is less than the threshold, the instantaneous frequency is set to zero.

B. Polarity

We can sum the amplitudes of the central part of the zero-phase Ricker wavelet as the polarity. The polarity is related to the reflection coefficient [12], [19]. If a zero-phase Ricker wavelet has duration N points with $a_1, a_2, ..., a_N$, the polarity is calculated from the central part of the wavelet as follows:

If $\sum_{i=1}^{i=N} a_i < 0$, then the polarity is positive.

If $\sum_{i=1}^{i=N} a_i > 0$, then the polarity is negative.

C. Example of Feature Generation, Labeling, and Classification

1) Seismic Signal and Feature Generation: A seismic trace in Fig. 3(a) is from the 36th trace of simulated seismogram in the experiment later. The sampling rate is 0.004 s. There are 512 points. Ricker wavelets of 20 Hz, 25 Hz, and 30 Hz, and Gaussian band 10–59.7-Hz random noise are generated. The four pattern classes are as follows.

Type 1: $s(t) =$ 20-Hz Ricker wavelet $+ n(t)$.
Type 2: $s(t) =$ 25-Hz Ricker wavelet $+ n(t)$.
Type 3: $s(t) =$ 30-Hz Ricker wavelet $+ n(t)$.
Type 4: $s(t) = n(t)$ Gaussian noise.

The first wavelet at 0.3 s and the second wavelet at 0.64 s in Fig. 3(a) are the 30-Hz zero-phase Ricker wavelets. The third wavelet at 0.85 s is the 20-Hz zero-phase Ricker wavelet that has the properties of high amplitude, low-frequency content, and polarity reversal. Its front part is down due to the negative reflection coefficient, and the energy is concentrated at the central part. Then, two 25-Hz zero-phase Ricker wavelets are at 0.98 and 1.04 s. From the sequence of 30 and 25-Hz wavelets, the frequency attenuation due to wave propagation in the earth is considered. The others are Gaussian noise.

Envelope $E(t)$ and instantaneous frequency $f(t)$ are generated as the two features and shown in Fig. 3(b) and (c). Each time point, or said each pattern, has two features. $E(t)$ is multiplied by 200 in order to have the same magnitude order as $f(t)$. The scatter diagram of two features associated with each time point is shown in Fig. 3(d). The patterns in the feature space are classified into two classes. One is the 20-Hz Ricker wavelet and the other is the 30 and 25-Hz Ricker wavelets and noise.

2) Labeling the Seismic Anomaly in Wavelet: The time points of the seismic trace are labeled as the training patterns of abnormal and normal classes. Labeling the seismic anomaly can be done in the feature space and back to the time points of the original seismic trace interactively until there is a linearly separable case in the feature space. In the feature space in Fig. 3(d) and (f), six feature points in the high envelope and low instantaneous frequency are associated with six time samples of the seismic trace denoted by dark dots at Fig. 3(e). The associated six time samples are at the same wavelet and can be labeled as the abnormal class. Six points in feature space in Fig. 3(f) are labeled as the abnormal class. The two classes in feature space are linearly separable.

3) Classification: We use two steps in the classification. The first step is that we get a linear classifier by perceptron learning to classify two classes in the feature space. Using the fixed-increment learning rule with learning-rate parameter $\alpha = 0.1$, a linear classifier is plotted in Fig. 3(f). Then, the second step is the computation of polarity. The final result is the location of the 20-Hz Ricker wavelet and shown in Fig. 3(g).

III. Perceptron Learning Rules

Perceptron can classify two-class patterns in the feature space for a linearly separable case. In a seismic trace, each time (each pattern) can have two features: envelope and instantaneous frequency. The symbols of the input pattern and the weight vector are given as follows. The pattern $\boldsymbol{x}_k$ is expanded to the

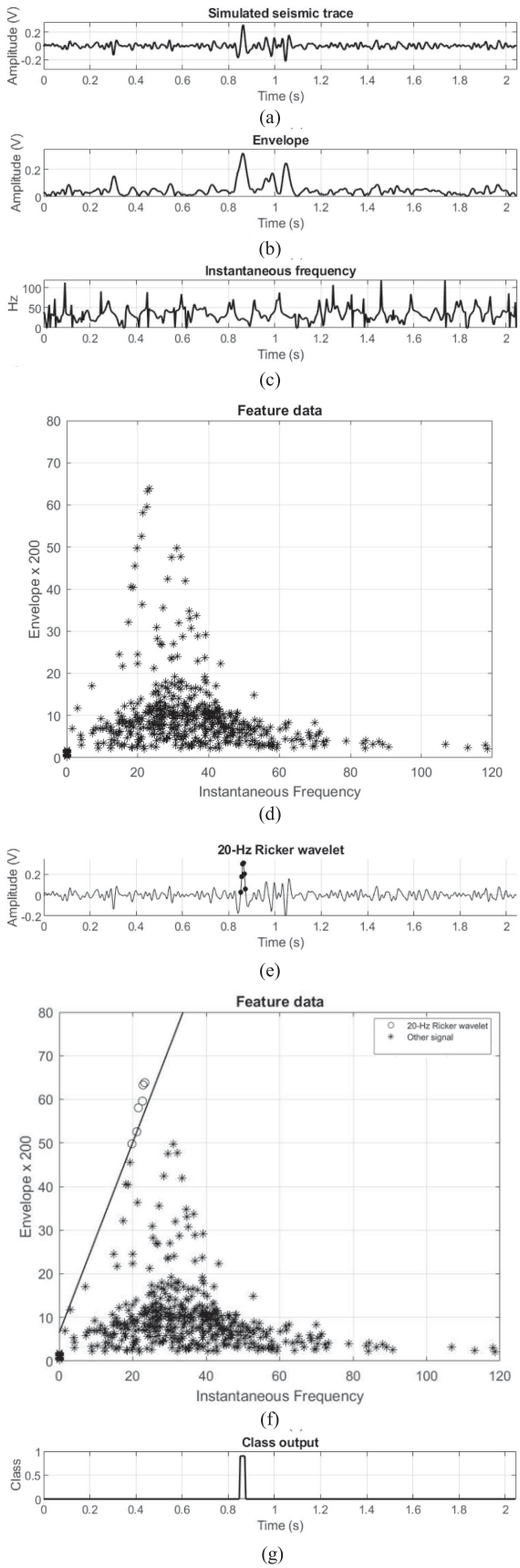


Fig. 3. (a) Simulated seismic trace. (b) Envelope. (c) Instantaneous frequency. (d) Patterns in feature space. (e) Training data of 20-Hz Ricker wavelet. (f) Training patterns of anomaly class and decision line. (g) Location of 20-Hz Ricker wavelet.

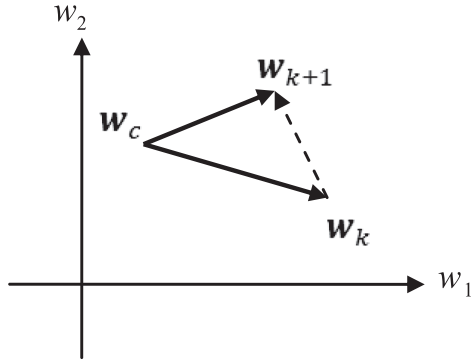


Fig. 4. Weight vectors in a weight space.

augmented vector $\boldsymbol{z}_k$

$$\boldsymbol{z}_k = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} f(t) \\ E(t) \\ 1 \end{bmatrix} \quad \boldsymbol{w}_k = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

where *k* is the index of the pattern and weight vector.

The computation from input to output is

$$\varphi\ (s) = \varphi\left(\boldsymbol{w}_k^T \boldsymbol{z}_k\right).$$

The activation function is the hard limiter denoted as φ.

$$\varphi\ (s) = \begin{cases} 1, & \text{if } s > 0 \\ 0, & \text{if } s\ =\ 0 \\ -1, & \text{if } s < 0 \end{cases}.$$

There are training patterns in classes C_1 and C_2. C_1 has a desired output $+1$ and C_2 has a desired output -1. We want to get a linear classifier with a convergence weight vector $\boldsymbol{w}_c$ such that ${\boldsymbol{w}_c}^T \boldsymbol{z}_k > 0$ for $\boldsymbol{z}_k$ in C_1 and ${\boldsymbol{w}_c}^T \boldsymbol{z}_k < 0$ for $\boldsymbol{z}_k$ in C_2. Input each training pattern $\boldsymbol{z}_k$ to the perceptron, the weight vector $\boldsymbol{w}_k$ must be adjusted if the real output is not equal to the desired output. The learning rules are in the following:

if $\boldsymbol{z}_k$ in C_1 and $\boldsymbol{w}_k^T\ \boldsymbol{z}_k \leq 0$, then $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k\ + \alpha \boldsymbol{z}_k$;

if $\boldsymbol{z}_k$ in C_2 and $\boldsymbol{w}_k^T \boldsymbol{z}_k \geq 0$, then $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k\ - \alpha \boldsymbol{z}_k$ where $\boldsymbol{w}_k$ is the old weight vector and $\boldsymbol{w}_{k+1}$ is the new weight vector, and α is a learning-rate parameter and $\alpha > 0$. For a linearly separable case, the learning process will be terminated until $\boldsymbol{w}_k$ is no longer adjusted, that is convergence.

The weight vector has its weight space, as shown in Fig. 4. By the learning rule, $\boldsymbol{w}_{k+1}$ is more close to the convergence weight vector $\boldsymbol{w}_c$ than $\boldsymbol{w}_k$.

We analyze the conventional learning rules, derive the lower and upper bounds of learning-rate parameter and propose new learning rules in the following.

A. Fixed-Increment Learning Rule

In the fixed-increment learning rule, the learning-rate parameter is a positive constant.

$$\alpha = \text{a positive constant.}$$

The convergence speed for a constant-rate perceptron varies greatly, depending on the choice of the learning-rate parameter. If it is too small, the convergence speed will be very slow. If it is too large, the learning can cause an overshooting problem.

B. Absolute Correction Rule

The absolute correction rule is to select α large enough to obtain $\boldsymbol{w}_{k+1}^T \boldsymbol{z}_k > 0$ for $\mathbf{z}_k$ in C_1 and $\boldsymbol{w}_{k+1}^T \boldsymbol{z}_k < 0$ for $\mathbf{z}_k$ in C_2 after adjustment of the old weight vector $\boldsymbol{w}_k$.

For $\boldsymbol{z}_k$ in C_1, the derivation of the learning-rate parameter is

$$\boldsymbol{w}_{k+1}^T\, \boldsymbol{z}_k = (\boldsymbol{w}_k + \alpha \boldsymbol{z}_k)^T\ \ \boldsymbol{z}_k = \boldsymbol{w}_k^T\ \boldsymbol{z}_k + \alpha \|\boldsymbol{z}_k\|^2 > 0$$

$$\alpha > \frac{-\boldsymbol{w}_k^T \boldsymbol{z}_k}{\|\boldsymbol{z}_k\|^2}$$

where $\boldsymbol{w}_k^T \boldsymbol{z}_k \leq 0$.

For $\boldsymbol{z}_k$ in C_2, it has a similar derivation.

Because of the value of $\boldsymbol{w}_k^T \boldsymbol{z}_k$in C_1 and C_2, the learning-rate parameter α becomes

$$\alpha > \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}.$$

α is taken to be the smallest integer greater than $\frac{|\boldsymbol{w}_k^T \boldsymbol{z}_k|}{\|\boldsymbol{z}_k\|^2}$.

$\alpha =$ smallest integer greater than $\frac{|\boldsymbol{w}_k^T \boldsymbol{z}_k|}{\|\boldsymbol{z}_k\|^2}$.

C. Fractional Correction Rule

The fractional correction learning rule is the modification of the absolute correction rule. The learning-rate parameter is

$$\alpha\ =\ \beta \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}\ \ (\beta > 1)\,.$$

We have a modification on β. The analysis is in the following.

In Fig. 4, $\boldsymbol{w}_{k+1}$ is more close to $\boldsymbol{w}_c$ than $\boldsymbol{w}_k$**.** After the weight vector adjustment, $\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_c\|$ is smaller than $\|\boldsymbol{w}_k - \boldsymbol{w}_c\|$.

$$\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_c\|^2 - \|\boldsymbol{w}_k - \boldsymbol{w}_c\|^2 < 0.$$

For $\mathbf{z}_k$ in C_1, the learning rule is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k\ + \alpha\ \boldsymbol{z}_k = \boldsymbol{w}_k\ + \beta \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2} \boldsymbol{z}_k.$$

Expand $\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_c\|^2 - \|\boldsymbol{w}_k - \boldsymbol{w}_c\|^2 < 0$, and substitute $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k\ + \beta \frac{|\boldsymbol{w}_k^T \boldsymbol{z}_k|}{\|\boldsymbol{z}_k\|^2} \boldsymbol{z}_k$ into it, we can get

$$\left(\beta \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}\right) \left(2\boldsymbol{w}_k^T \boldsymbol{z}_k - \beta \boldsymbol{w}_k^T \boldsymbol{z}_k - 2\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right) < 0$$

$$\left(\boldsymbol{w}_k^T \boldsymbol{z}_k\,(2-\beta) - 2\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right) < 0.$$

Because $\boldsymbol{w}_k{}^T \boldsymbol{z}_k < 0$ and $\boldsymbol{w}_c{}^T \boldsymbol{z}_k > 0$, must $(2-\beta) \geq 0$**.** So, $\beta \leq 2$. From the previous $\beta > 1$, we can get $1 < \beta \leq 2$.

For $\mathbf{z}_k$ in C_2, it has a similar derivation.

The learning-rate parameter becomes

$$\alpha\ =\ \beta \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}\ \ (1 < \beta \leq 2)\,.$$

In [25], the β in fractional correction rule was $0 < \beta \leq 2$. We have a modification.

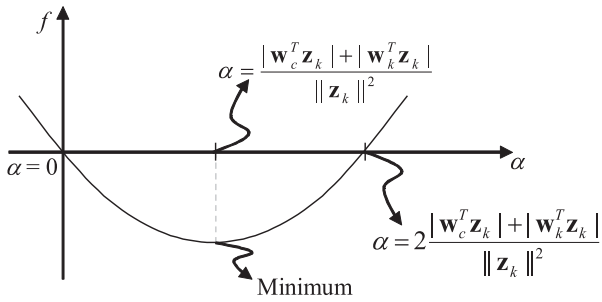


Fig. 5. Function $f(\alpha)$, learning-rate parameter at a minimum of $f(\alpha)$, and the lower bound and upper bound of the learning-rate parameter.

D. Normalized Learning Rule

We can set that the weight vector becomes normalized at each training step. $\|\boldsymbol{w}_{k+1}\|^2 = \|\boldsymbol{w}_k\|^2 = 1$.

For $\boldsymbol{z}_k$ in C_1, $\boldsymbol{w}_k^T \boldsymbol{z}_k \leq 0$

$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k\ + \alpha \boldsymbol{z}_k$

$$\|\boldsymbol{w}_{k+1}\|^2 = \|\boldsymbol{w}_k + \alpha \boldsymbol{z}_k\|^2 = \|\boldsymbol{w}_k\|^2 + \alpha^2 \|\boldsymbol{z}_k\|^2 + 2\alpha \boldsymbol{w}_k^T \boldsymbol{z}_k$$

$$\|\boldsymbol{w}_{k+1}\|^2 -\ \|\boldsymbol{w}_k\|^2 =\ 0\ = \alpha^2\ \|\boldsymbol{z}_k\|^2 + 2\alpha \boldsymbol{w}_k^T \boldsymbol{z}_k$$

$$\alpha\ = \frac{-2\boldsymbol{w}_k^T \boldsymbol{z}_k}{\|\boldsymbol{z}_k\|^2}\ .$$

For $\boldsymbol{z}_k$ in C_2, it has a similar derivation.

From the derivation in classes 1 and 2, the learning-rate parameter becomes

$$\alpha\ = \frac{2\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}.$$

It is corresponding to $\beta = 2$ in the fractional correction learning rule.

E. Optimal Learning-Rate Parameter

We derive the optimal learning-rate parameter. For a linearly separable case, there exists a convergence weight vector $\boldsymbol{w}_c$ such that $\boldsymbol{w}_c^T \boldsymbol{z}_k > 0$ for all $\boldsymbol{z}_k$ in class 1 and $\boldsymbol{w}_c^T \boldsymbol{z}_k < 0$ for all $\boldsymbol{z}_k$ in class 2.

For $\boldsymbol{z}_k$ in C_1, $\boldsymbol{w}_k^T \boldsymbol{z}_k \leq 0$, $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k\ + \alpha \boldsymbol{z}_k$

$$\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_c\|^2 = \|\boldsymbol{w}_k + \alpha \boldsymbol{z}_k - \boldsymbol{w}_c\|^2 = \|\boldsymbol{w}_k - \boldsymbol{w}_c + \alpha \boldsymbol{z}_k\|^2$$

$$\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_c\|^2 = \|\boldsymbol{w}_k - \boldsymbol{w}_c\|^2 + 2\alpha(\boldsymbol{w}_k - \boldsymbol{w}_c)^T \boldsymbol{z}_k + \alpha^2 \|\boldsymbol{z}_k\|^2.$$

From Fig. 4

$$\|\boldsymbol{w}_{k+1} - \boldsymbol{w}_c\|^2 - \|\boldsymbol{w}_k - \boldsymbol{w}_c\|^2 = \alpha^2 \|\boldsymbol{z}_k\|^2 + 2\alpha(\boldsymbol{w}_k - \boldsymbol{w}_c)^T \boldsymbol{z}_k < 0.$$

$$\text{Let } f\ (\alpha) = \alpha^2\ \|\boldsymbol{z}_k\|^2 + 2\alpha(\boldsymbol{w}_k - \boldsymbol{w}_c)^T \boldsymbol{z}_k \quad (1)$$

$f(\alpha)$ is the difference between the squared distance of updated $\boldsymbol{w}_{k+1}$ to convergence $\boldsymbol{w}_c$ and the squared distance of the current $\boldsymbol{w}_k$ to convergence $\boldsymbol{w}_c$. $f(\alpha)$ is a quadratic function of α and is shown in Fig. 5. $f(\alpha) < 0$, i.e., $\boldsymbol{w}_{k+1}$ is closer to $\boldsymbol{w}_c$ than $\boldsymbol{w}_k$. f

(α) has a minimum value that indicates that $\boldsymbol{w}_{k+1}$ can reach to $\boldsymbol{w}_c$ fastest. In order to get the minimum of $f(\alpha)$, let $f'(\alpha) = 0$

$$f'\ (\alpha) = \ 2\alpha\|\boldsymbol{z}_k\|^2 + 2(\boldsymbol{w}_k - \boldsymbol{w}_c)^T\ \boldsymbol{z}_k = \ 0$$

$$\alpha\ = \frac{-(\boldsymbol{w}_k - \boldsymbol{w}_c)^T \boldsymbol{z}_k}{\|\boldsymbol{z}_k\|^2}\ = \frac{\boldsymbol{w}_c{}^T \boldsymbol{z}_k - \boldsymbol{w}_k{}^T \boldsymbol{z}_k}{\|\boldsymbol{z}_k\|^2}.$$

For $\boldsymbol{z}$ in C_2, it has a similar derivation.

From the derivation in classes 1 and 2

$$\alpha = \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right| + \left|\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}. \tag{2}$$

However, this optimal learning-rate parameter is not applicable because the convergence weight vector $\boldsymbol{w}_c$ is unknown before training. But it can be compared with other learning rules.

F. Lower and Upper Bounds of Learning-Rate Parameter

Here, we derive the lower and upper bounds of the learning-rate parameter. From (1), for $\boldsymbol{z}_k$ in C_1

$$f\ (\alpha) = \alpha^2\ \|\boldsymbol{z}_k\|^2 + 2\alpha(\boldsymbol{w}_k - \boldsymbol{w}_c)^T \boldsymbol{z}_k.$$

In Fig. 5, $f\ (\alpha) < 0$, the learning-rate parameter α exists between the left- and right-hand sides of $f\ (\alpha) = \ 0$.

For $f\ (\alpha) = \ 0$

$$\alpha^2\|\boldsymbol{z}_k\|^2 + 2\alpha(\boldsymbol{w}_k - \boldsymbol{w}_c)^T\ \boldsymbol{z}_k = 0$$

$$\alpha\left(\alpha\|\boldsymbol{z}_k\|^2 + 2(\boldsymbol{w}_k - \boldsymbol{w}_c)^T \boldsymbol{z}_k\right) = 0.$$

$\alpha\ =\ 0$, a lower bound, or $\alpha^2\|\boldsymbol{z}_k\|^2 + 2\alpha(\boldsymbol{w}_k - \boldsymbol{w}_c)^T\ \boldsymbol{z}_k = 0$.

$$\alpha = \frac{-2(\boldsymbol{w}_k - \boldsymbol{w}_c)^T \boldsymbol{z}_k}{\|\boldsymbol{z}_k\|^2}$$

$$\alpha = 2\frac{\boldsymbol{w}_c{}^T \boldsymbol{z}_k - \boldsymbol{w}_k^T \boldsymbol{z}_k}{\|\boldsymbol{z}_k\|^2}.$$

For $\boldsymbol{z}_k$ in C_1, $\boldsymbol{w}_k^T \boldsymbol{z}_k \le 0$ and $\boldsymbol{w}_c{}^T \boldsymbol{z}_k > 0$.

$$\alpha\ =\ 2\ \frac{\boldsymbol{w}_c{}^T \boldsymbol{z}_k - \boldsymbol{w}_k^T \boldsymbol{z}_k}{\|\boldsymbol{z}_k\|^2} =\ 2\frac{\left|\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right| + \left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}.$$

It is an upper bound.

For $\boldsymbol{z}_k$ in C_2, it has a similar derivation.

From the derivation in classes 1 and 2, the lower bound of the learning-rate parameter is $\alpha\ =\ 0$. The upper bound is

$$\alpha\ =\ 2\frac{\left|\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right| + \left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}.$$

We set $\alpha > 0$. The range of the learning-rate parameter is

$$0 < \alpha < 2\frac{\left|\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right| + \left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}.$$

For the learning-rate parameter within this range, the learning can have a convergence.

G. Fusion Learning Rule

We propose a fusion learning rule by combining optimal learning-rate parameter, the normalized property of $\boldsymbol{w}_c$, and fractional correction parameter. Although the optimal learning-rate parameter in (2) is not applicable, we can make a modification. In optimal learning-rate parameter

$$\alpha\ = \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right| + \left|\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2}$$

$$\alpha\ = \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right| + |\|\boldsymbol{w}_c\|\|\boldsymbol{z}_k\|\cos(\theta_c)|}{\|\boldsymbol{z}_k\|^2}$$

where θ_c is the angle between $\boldsymbol{z}_k$ and $\boldsymbol{w}_c$.

Assume $\boldsymbol{w}_c$ can be normalized, $\|\boldsymbol{w}_c\| = \ 1$, then

$$\alpha\ = \frac{\left|\boldsymbol{w}_k{}^T \boldsymbol{z}_k\right| + |\|\boldsymbol{z}_k\|\cos(\theta_c)|}{\|\boldsymbol{z}_k\|^2}.$$

$\boldsymbol{w}_c{}^T \boldsymbol{z}_k > 0$ for $\boldsymbol{z}_k$ in $C_1$and $\boldsymbol{w}_c{}^T \boldsymbol{z}_k < 0$ for $\boldsymbol{z}_k$ in C_2. So, $\cos(\theta_c)$ has the range $0 < \cos(\theta_c) \le 1$ for $\boldsymbol{z}_k$ in C_1 and $-1 \le \cos(\theta_c) < 0$ for $\boldsymbol{z}_k$ in C_2.

$$\alpha\ = \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right| + |\|\boldsymbol{z}_k\|\cos(\theta_c)|}{\|\boldsymbol{z}_k\|^2}\ = \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right| + \|\boldsymbol{z}_k\|\,|\cos(\theta_c)|}{\|\boldsymbol{z}_k\|^2}.$$

We change $|\cos(\theta_c)|$ into a constant δ between 0 and 1. Then

$$\alpha\ = \frac{\left|\boldsymbol{w}_k{}^T \boldsymbol{z}_k\right| + \delta\|\boldsymbol{z}_k\|}{\|\boldsymbol{z}_k\|^2}.$$

The numerator may be greater than $2|\boldsymbol{w}_k{}^T \boldsymbol{z}_k|$. We have α in the following:

$$\alpha = \begin{cases} \frac{\left|\boldsymbol{w}_k{}^T \boldsymbol{z}_k\right| + \delta\|\boldsymbol{z}_k\|}{\|\boldsymbol{z}_k\|^2},\ if\ \delta\|\boldsymbol{z}_k\|^2 < \left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right| \\ \frac{2\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\|\boldsymbol{z}_k\|^2},\ otherwise \end{cases}$$

$$\alpha\ = \frac{\min\left(\delta\|\boldsymbol{z}_k\| + \left|\boldsymbol{w}_k{}^T \boldsymbol{z}_k\right|, 2\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|\right)}{\|\boldsymbol{z}_k\|^2}\ ,\ 0 < \delta \le 1.$$

The case $\alpha\ = \frac{2|\boldsymbol{w}_k^T \boldsymbol{z}_k|}{\|\boldsymbol{z}_k\|^2}$ is corresponding to $\beta\ =\ 2$ in the fractional correction learning rule and the normalized learning rule.

H. Summary of Learning-Rate Parameters

The learning-rate parameters are summarized in Table I. All learning-rate parameters are in the range of lower and upper bounds. There is a relation in the following:

$$\begin{aligned} 0 < \frac{\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\boldsymbol{z}_k{}^2} < \alpha \le \frac{\left|\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right| + \left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\boldsymbol{z}_k{}^2} \\ \le \frac{\min\left(\delta\boldsymbol{z}_k + \left|\boldsymbol{w}_k{}^T \boldsymbol{z}_k\right|, 2\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|\right)}{\boldsymbol{z}_k{}^2} \le \frac{2\left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\boldsymbol{z}_k{}^2} \\ < 2\frac{\left|\boldsymbol{w}_c{}^T \boldsymbol{z}_k\right| + \left|\boldsymbol{w}_k^T \boldsymbol{z}_k\right|}{\boldsymbol{z}_k{}^2}. \end{aligned}$$

TABLE I
SUMMARY OF LEARNING-RATE PARAMETERS

	Learning-rate parameter
Fixed increment rule	$\alpha = c > 0$
Absolute correction rule	$\alpha =$ smallest integer greater than $\frac{\lvert \boldsymbol{w}_k^T \boldsymbol{z}_k \rvert}{\lVert \boldsymbol{z}_k \rVert^2}$
Fractional correction rule	$\alpha = \beta \frac{\lvert \boldsymbol{w}_k^T \boldsymbol{z}_k \rvert}{\lVert \boldsymbol{z}_k \rVert^2}$ $(1 < \beta \leq 2)$
Normalized learning rule	$\alpha = 2 \frac{\lvert \boldsymbol{w}_k^T \boldsymbol{z}_k \rvert}{\lVert \boldsymbol{z}_k \rVert^2}$
Optimal learning-rate parameter	$\alpha = \frac{\lvert \boldsymbol{w}_c{}^T \boldsymbol{z}_k \rvert + \lvert \boldsymbol{w}_k^T \boldsymbol{z}_k \rvert}{\lVert \boldsymbol{z}_k \rVert^2}$
Lower and upper bounds	$0 < \alpha < 2 \frac{\lvert \boldsymbol{w}_c{}^T \boldsymbol{z}_k \rvert + \lvert \boldsymbol{w}_k^T \boldsymbol{z}_k \rvert}{\lVert \boldsymbol{z}_k \rVert^2}$
Fusion learning rule	$\alpha = \frac{min(\delta \lVert \boldsymbol{z}_k \rVert + \lvert \boldsymbol{w}_k{}^T \boldsymbol{z}_k \rvert, 2 \lvert \boldsymbol{w}_k^T \boldsymbol{z}_k \rvert)}{\lVert \boldsymbol{z}_k \rVert^2}$ $(0 < \delta \leq 1)$

The proposed fusion learning-rate parameter is closer to the optimal learning-rate parameter than the other learning-rate parameters.

IV. EXPERIMENTS IN SEISMOGRAMS

A. Experiment in the Simulated Seismogram

A geological model is in Fig. 6(a). It has the gas and oil sand zones. They are trapped beneath the shale layer, the third layer. Each layer has the density and p-wave velocity that can calculate the reflection coefficient. The reflection coefficients can convolve with the zero-phase Ricker wavelets that generate the primary reflection synthetic seismogram in Fig. 6(b). It has 64 seismic traces. Each trace has 512 sampling points. The sampling time is 0.004 s. The reflection at the top of the gas sand zone is 20-Hz zero-phase Ricker wavelet with high amplitude, low frequency, and polarity reversal. For each trace, the signal value is positive at the right-hand side and negative on the left-hand side. Fig. 6(b) contains four types of data.

Type 1: $s(t) =$ 20-Hz Ricker wavelet $+ n(t)$.

Type 2: $s(t) =$ 25-Hz Ricker wavelet $+ n(t)$.

Type 3: $s(t) =$ 30-Hz Ricker wavelet $+ n(t)$.

Type 4: $s(t) = n(t)$ Gaussian noise with standard deviation 0.05 and bandpass 10–59.7 Hz.

In Fig. 6(b), the 30-Hz Ricker wavelets are at the first and second layers. Then, there are 20-Hz Ricker wavelets, as the seismic anomaly between 0.8 and 1.0 s on trace number 20–44 that has the properties of high amplitude, low-frequency content, and polarity reversal. After that, there are 25-Hz Ricker wavelets.

The selection of training patterns is important in seismic pattern recognition. It is selected from the training traces. And the training traces are selected from the seismogram. We have

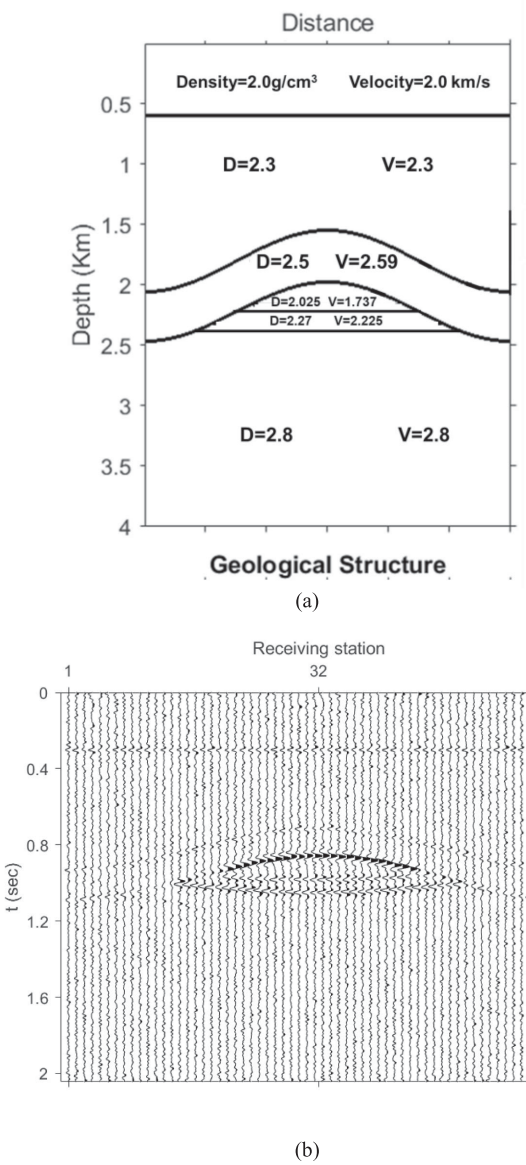
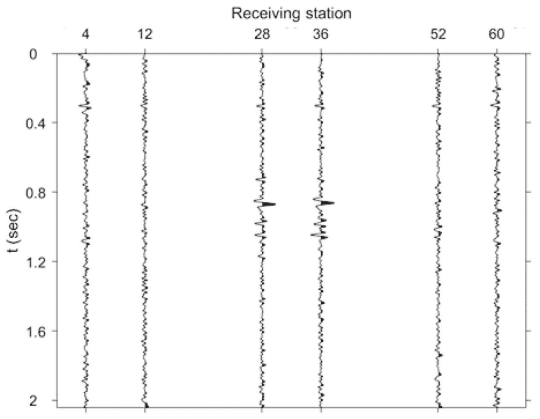


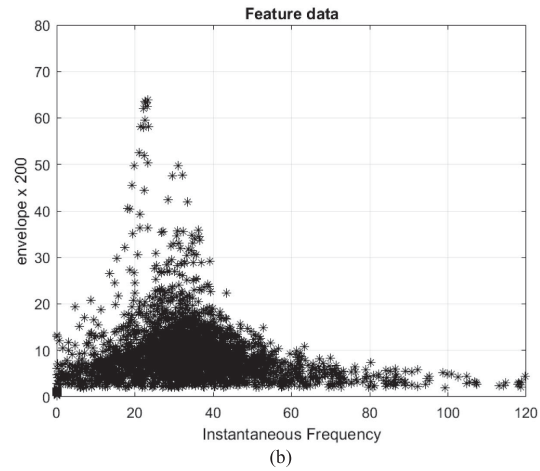
Fig. 6. (a) Geological model. (b) Simulated seismogram.

two cases to select the training traces. For case 1, training traces are selected uniformly. There are eight selected training traces from Fig. 6(a): 4th, 12th, 20th, 28th, 36th, 44th, 52nd, and 60th. However, there are the mixed wavelets at 20th and 44th traces, so these two traces are not selected. As a result, there are six training traces. It is shown in Fig. 7(a). The total time points are $512 \times 6 = 3072$. One time point is one pattern that has two features: envelope and instantaneous frequency. The feature data are shown in Fig. 7(b). After the labeling process, 13 patterns are the class of seismic anomaly and 3059 patterns are the other class. The labeled feature data and classifier are shown in Fig. 7(c). The classifier is from the result of the fastest fusion learning rule with parameter $\delta = 1$.

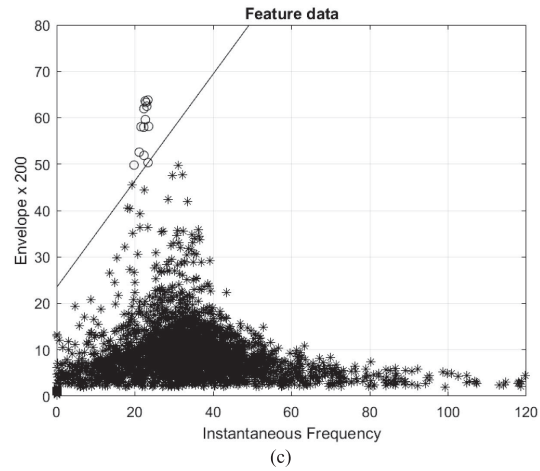
For case 2, the selection of training traces has more traces that have 20-Hz Ricker wavelets related to seismic anomaly. Therefore, 24th, 32nd, and 40th traces are added to the training



(a)

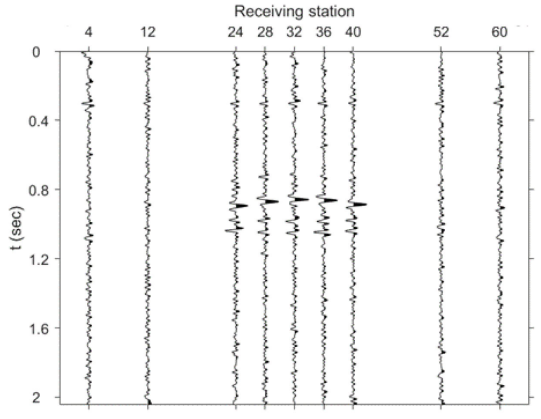


(b)

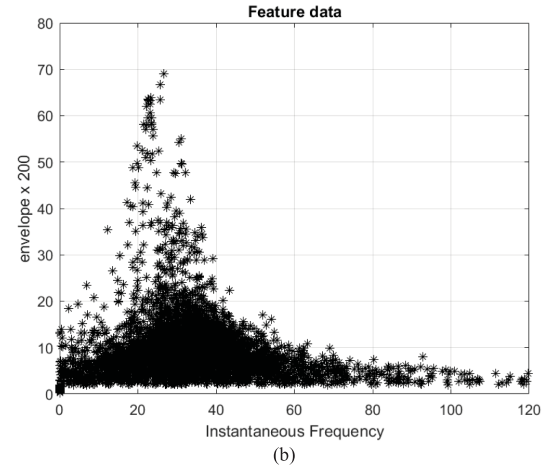


(c)

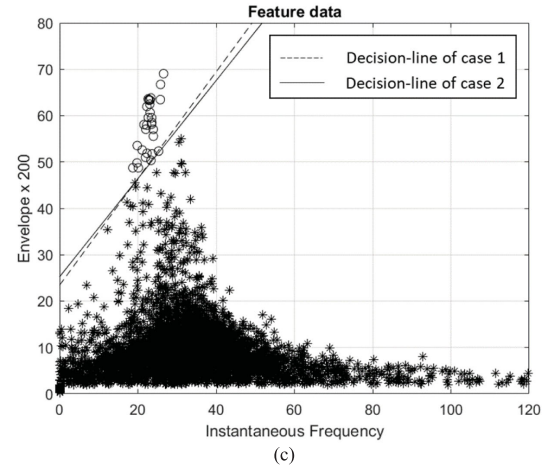
Fig. 7. Training patterns from case 1 of selecting training traces. (a) Training traces. (b) Feature data. (c) Training patterns and classifier.



(a)



(b)



(c)

Fig. 8. Training patterns from case 2 of selecting training traces. (a) Training traces. (b) Feature data. (c) Training patterns and classifier.

traces of case 1. So, there are nine selected training traces and shown in Fig. 8(a). It has $512 \times 9 = 4608$ training time points (patterns). The feature data are shown in Fig. 8(b). After the labeling process, 30 patterns are the class of seismic anomaly and 4578 patterns are the other class. The labeled feature data and classifier are shown in Fig. 8(c). Fig. 8(c) has more training patterns than Fig. 7(c). The regions of the training patterns are wider. The classifier is from the result of the fastest fusion learning rule with parameter $\delta = 1$.

Compare with the classifiers in Figs. 7(c) and 8(c), the classifier in Fig. 8(c) is shifted to the right because there are more training patterns at the boundary affecting the classifier. More

TABLE II
PERFORMANCE FOR LEARNING RULES IN THE SIMULATED SEISMOGRAM

Learning rule	Parameters	Number of iterations	CPU training time (s)
Fixed increment rule	$\alpha = 10^{-3}$	15225	196.537
	$\alpha = 10^{-2}$	1360	14.33
	$\alpha = 0.1$	921	9.4
	$\alpha = 1$	4921	65.97
Absolute correction rule	α = Smallest integer greater than $\frac{\lvert \mathbf{w}_k^T \mathbf{z}_k \rvert}{\lVert \mathbf{z}_k \rVert^2}$	4891	60.36
Fractional correction rule	$\beta = 1.2$	4892	60.51
	$\beta = 1.5$	2501	57.24
	$\beta = 2.0$	433	6.655
	$\beta = 2.1$	No convergence	No convergence
Fusion learning rule	$\delta = 0.01$	1638	51.23
	$\delta = 0.1$	825	7.53
	$\delta = 1$	422	5.62
SVM (Linear kernel)	$\eta = \frac{1}{K(\mathbf{x}_k, \mathbf{x}_k)}$	5676	874.38
1-D CNN	$\alpha = 5\text{x}10^{-4}$	447	127.21
	$\alpha = 0.001$	237	58.20
	$\alpha = 0.01$	39	12.18
	$\alpha = 0.05$	No convergence	No convergence

training patterns can be more representative and the patterns at the boundary can affect the classifier. The classifier in Fig. 3(f) is from one seismic trace. Compare with the classifiers in Figs. 3(f) and 8(c), the classifier in Fig. 8(c) is obviously shifted to the right. The selection of training traces has more training patterns and affects the classifier.

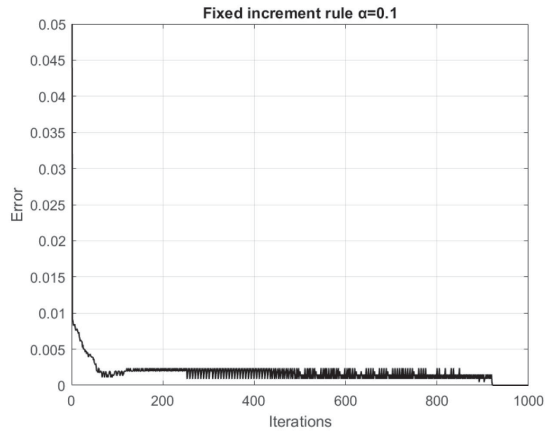
In the learning process, recognition error must be calculated. Entering all training patterns into perceptron for learning is one iteration. Then, the recognition error is calculated. It is defined as the number of the misclassified training patterns k divided by the total number of training patterns N

$$\text{Error } = \frac{k}{N}.$$

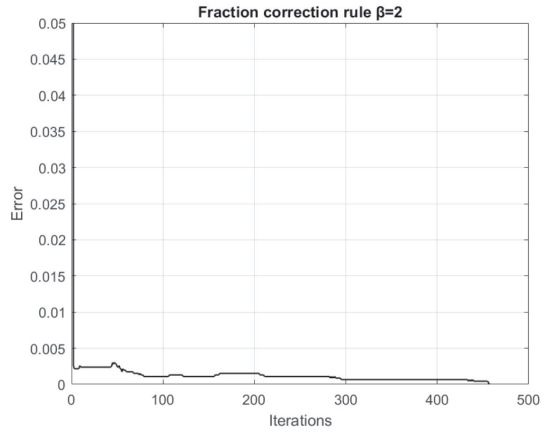
The learning is convergence if the recognition error is 0.

The experiments are in Fig. 8(c) to find the classifiers by the fixed-increment rule, fractional correction rule, and fusion learning rule. The error versus iteration is plotted in Fig. 9(a)–(c), respectively. The learning performances for more experiments are listed in Table II. In order to do the comparison, the initial weight vectors are random but same for all experiments. For small learning-rate parameter, the number of iteration is large. The convergence speed is low. Then, increasing the learning-rate parameter, the number of iterations becomes smaller. But for the large learning-rate parameter, the iteration number is large, even no convergence, because the classifier will become vibration for large adjustment of the weight vector.

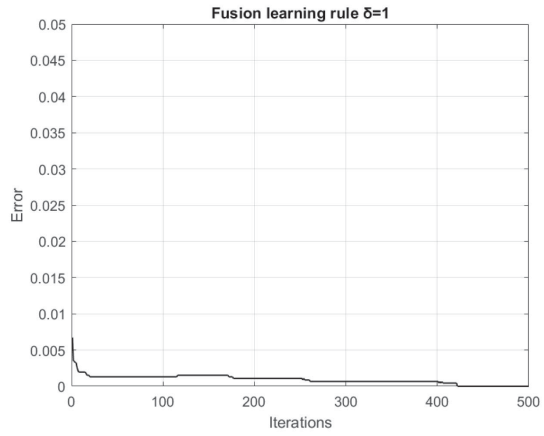
After the classification of Fig. 8(c) using the fastest fusion learning rule with $\delta = 1$, the detected portions at time points are through the calculation of the polarity. Then, the detected results of a seismic anomaly in the training traces and the whole seismogram are shown in Fig. 10(a) and (b).



(a)



(b)

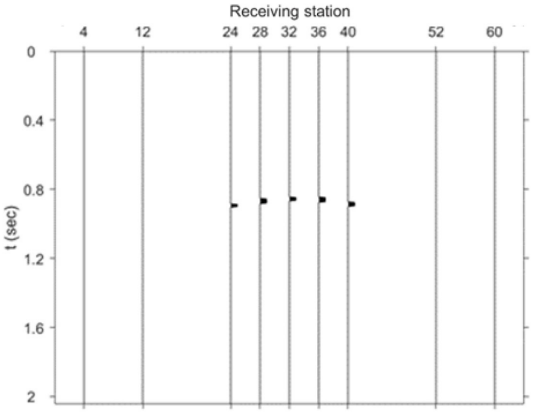


(c)

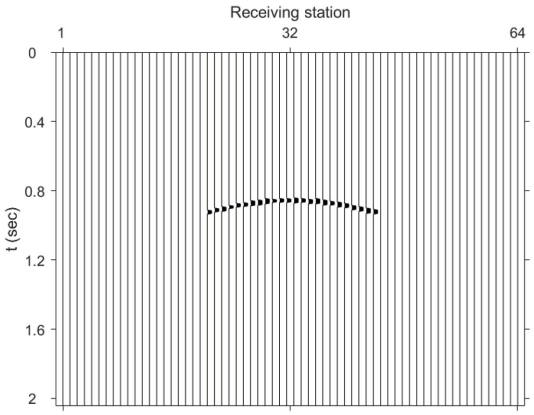
Fig. 9. Error versus iteration of perceptron learning rules. (a) Fixed-increment rule ($\alpha = 0.1$). (b) Fractional correction rule ($\beta = 2$). (c) Fusion learning rule ($\delta = 1$).

B. Experiment in the Real Seismogram

We also do the experiment in the real seismic data. The real seismogram at Mississippi Canyon is shown in Fig. 11. We follow the same procedures as in the simulated seismogram. In the selection of training traces, we have two cases. For case



(a)



(b)

Fig. 10. (a) Detected seismic anomaly in training traces. (b) Detected seismic anomaly in the seismogram.

1, we select eight training traces uniformly. They are 4th, 12th, 20th, 28th, 36th, 44th, 52nd, and 60th traces. Then, we remove the 28th trace to avoid a mixed wavelet. The training traces are in Fig. 12(a). It has $512\times7=3584$ time points. After the feature generation, we can plot the feature data in Fig. 12(b). We take the patterns with high envelope and low instantaneous frequency as the abnormal class from Fig. 12(b). Then, we map the patterns to the time points in the original signal interactively. The time points must be at the same wavelet. Except that the wavelets must be in a horizon layer in the seismogram. If the pattern is not on a horizon, it will be deleted. The selected training patterns are the circle symbols in Fig. 12(c). Through training pattern selection, there are 21 patterns for abnormal and 3563 for normal.

For case 2, we add three training traces with a high amplitude. They are 30th, 32nd, and 34th. Fig. 13(a) shows the training traces. Fig. 13(b) shows the feature data. Fig. 13(c) shows the training patterns of two classes. There are $512\times10=5120$ time points in total. It has 36 points for abnormal and 5084 for normal.

The classifiers on Figs. 12(c) and 13(c) are from the fastest fusion learning rule with $\delta=1$. Two classifiers are identical because the boundary patterns are the same. Learning in Fig. 13(c),

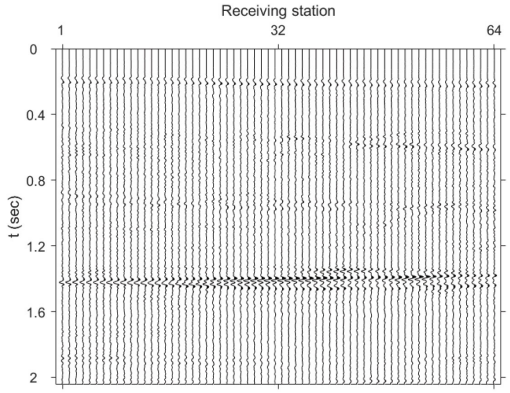


Fig. 11. Real seismogram at Mississippi Canyon.

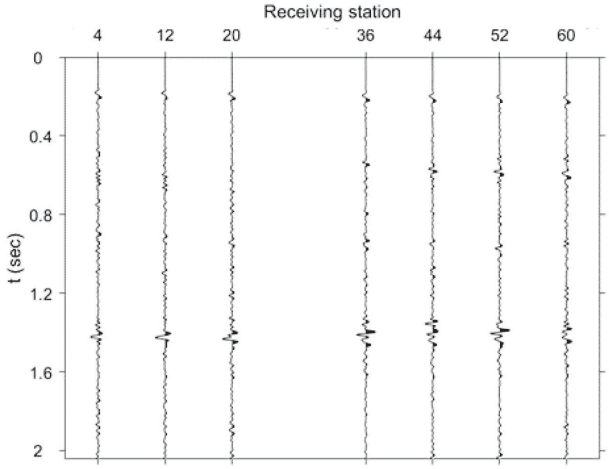
the error versus iteration is shown in Fig. 14. The learning can converge in 144 iterations with CPU time 2.98 s.

After the classification of Fig. 13(c) using the fastest fusion learning rule with $\delta=1$, the detected portions at time points are through the calculation of the polarity. Then, the detected results of a seismic anomaly with black color in the training traces and the whole seismogram are shown in Fig. 15(a) and (b). The black and white colors are different in polarity in Fig. 15(b). The black color is for a seismic anomaly. And the white color is for a normal class.

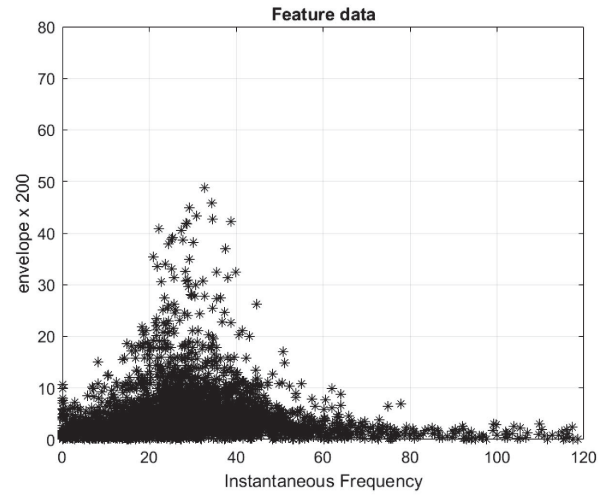
C. Discussion of Comparison With Other Methods

Other statistical pattern recognition methods were ever used in the detection of seismic anomaly [10], [12]. Tree classification was used in [10]. The first step was using instantaneous frequency, and the second was using an envelope. It was equivalent to set vertical line classifier and horizontal line classifier in the feature space. Here, from the distribution of feature data of real seismic data in Fig. 13(c), the classifier is slant. It is not good to use the tree classification, i.e., not vertical and horizontal lines. It will cause an error. The decision-theoretical methods were used in [12]. They were linear classification, Bayes classification, and quadratic classification. But there was no real data experiment. Here, the perceptron method is applied to the real seismic data at Mississippi Canyon.

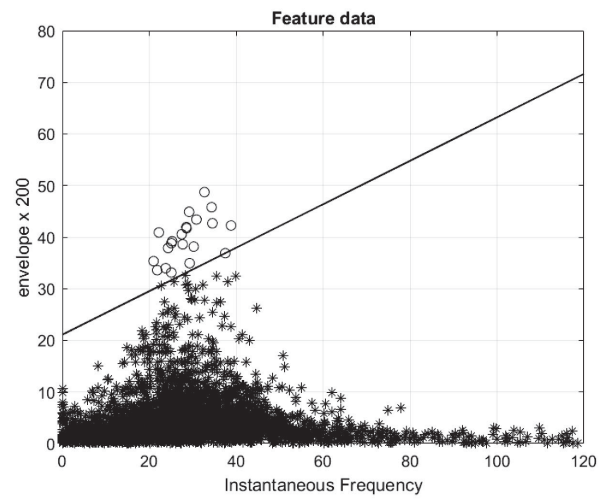
Other syntactic pattern recognition methods were ever used in the detection of seismic anomaly [32], [33]. In [32], it was a supervised classification to detect bright spots. Two major steps were used in the processing. The first step extracted the 1-D candidate wavelets of a bright spot. Levenshtein distance [34] was used to calculate the distance between two strings. The second step was to test whether or not the candidate bright spot wavelets represent a continuous reflection layer. In [33], it was an unsupervised classification to detect a bright spot. The dendrogram was constructed for hierarchical clustering. The optimal number of clustering was 6. For both syntactic methods in [32] and [33], the horizontal layer connection in the wavelets of the bright spot was not long. That is because the segment recognition of Freeman chain code is sensitive to noise. Here, the detected black portions of bright spots in the



(a)

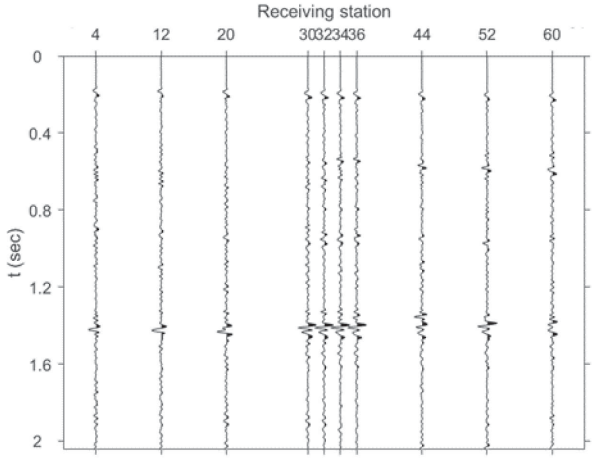


(b)

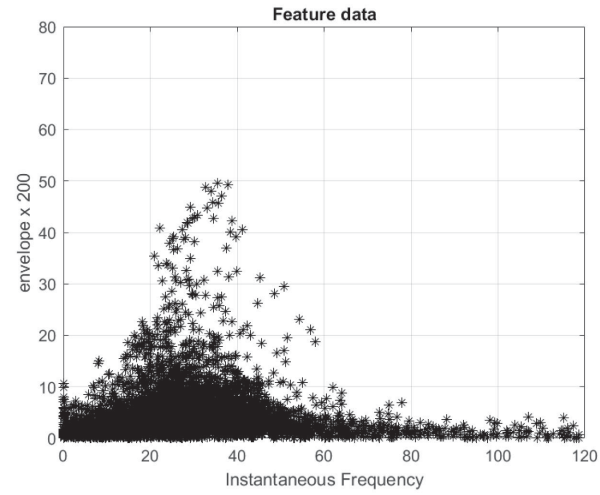


(c)

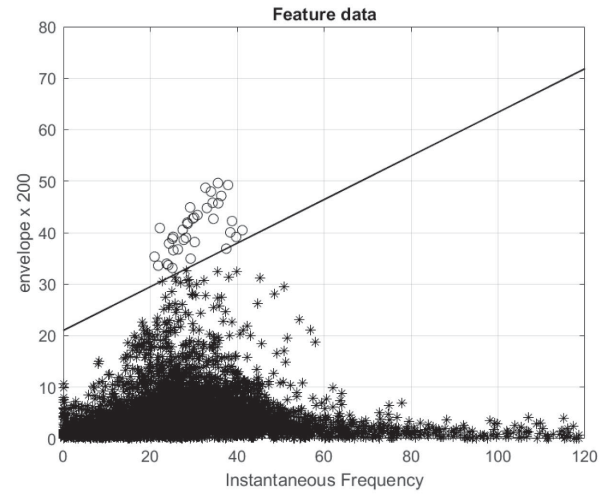
Fig. 12. Training patterns of real seismogram from case 1 of selecting training traces. (a) Training traces of real seismogram. (b) Feature data. (c) Training patterns and classifier.



(a)



(b)



(c)

Fig. 13. Training patterns of real seismogram from case 2 of selecting training traces. (a) Training traces of real seismogram. (b) Feature data. (c) Training patterns and classifier.

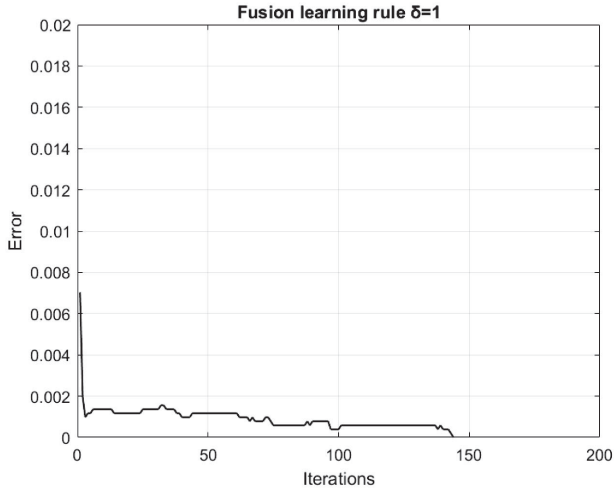
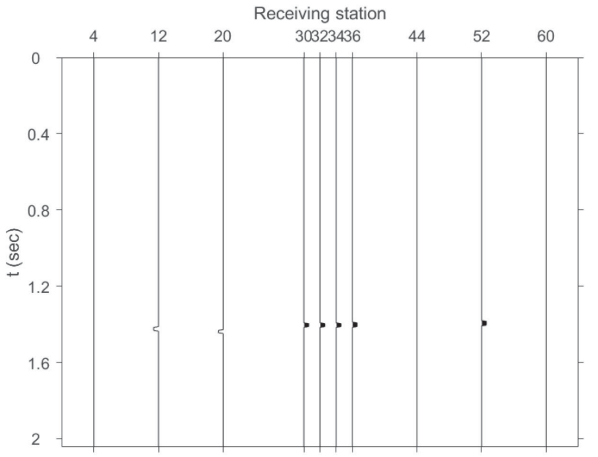
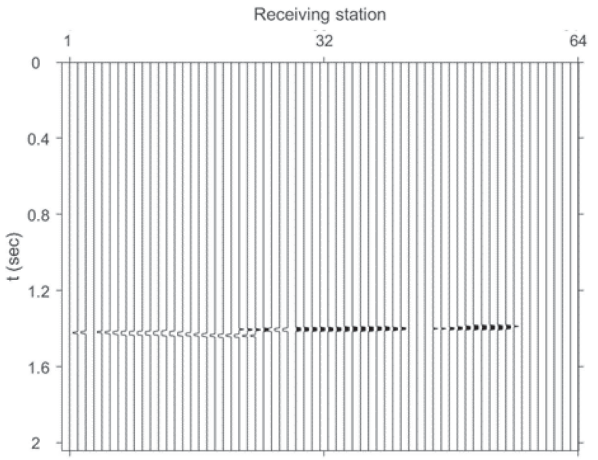


Fig. 14. Error versus iteration of training process using fusion learning rule in case 2 of the real seismogram.



(a)



(b)

Fig. 15. (a) Detected seismic anomaly in training traces. (b) Detected seismic anomaly in a real seismogram.

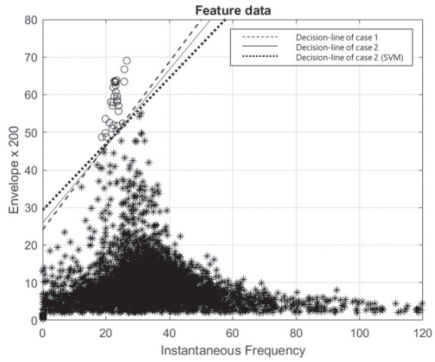


Fig. 16. Comparison of classifiers using perceptron and SVM in feature data of the simulated seismogram.

horizontal direction in Fig. 15(b) are long. The continuity of the horizon is important in the layer detection. So, the result of the perceptron classification is better.

For the comparison, we also use a support vector machine (SVM) [35] to the feature data of the simulated seismogram in Fig. 8(c). SVM is to find the classifier that the marginal distance between two-class support vectors is largest. The classifier is in the middle of the two-class support vectors. The two-class training patterns $\{\boldsymbol{x}_k\}_{k\,=\,1,\,2,\,\ldots,\,N}$ has the desired output $d_k = +1$ for C_1 and $d_k =$ -1 for C_2. Each $\boldsymbol{x}_k$ is expanded to the high-dimensional feature vector $\phi(\boldsymbol{x}_k)$.

Using the Lagrange method with Karush–Kuhn–Tucker conditions in the primal problem to minimize the Lagrange function, we can get the weight vector of the classifier

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i d_i \phi\left(\boldsymbol{x}_i\right)$$

where multipliers $\boldsymbol{\alpha} = \{\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_N\}$, and the classifier is

$$g\left(\phi\left(\boldsymbol{x}\right)\right) = \boldsymbol{w}^T\ \phi\ \left(\boldsymbol{x}\right) = \sum_{i=1}^{N} \alpha_i d_i \phi^T\left(\boldsymbol{x}_i\right)\ \phi\ \left(\boldsymbol{x}\right)$$

$$= \sum_{i=1}^{N} \alpha_i d_i K\left(\boldsymbol{x}, \boldsymbol{x}_i\right) = 0 \tag{3}$$

where K is the kernel function

$$K\ \left(\boldsymbol{x}, \boldsymbol{x}_i\right) = \phi^T\ \left(\boldsymbol{x}\right)\phi\ \left(\boldsymbol{x}_i\right) = \phi^T\ \left(\boldsymbol{x}_i\right)\phi\ \left(\boldsymbol{x}\right) = \ K\left(\boldsymbol{x}_i, \boldsymbol{x}\right).$$

The multipliers $\boldsymbol{\alpha} = \{\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_N\}$ can be solved in the dual problem to maximize the Lagrange function L

$$L\ \left(\boldsymbol{\alpha}\right) = \sum_{i=1}^{N} \alpha_i\ - \frac{1}{2} \sum_{i\,=\,1}^{N} \sum_{j\,=\,1}^{N} \alpha_i \alpha_j d_i d_j K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right).$$

The gradient ascent method is used to adjust α_k such that the function L becomes maximum

$$\alpha_k\ \left(t+1\right) = \alpha_k\ \left(t\right) + \Delta\alpha_k\ \left(t\right) = \alpha_k\ \left(t\right) + \eta \frac{\partial L\left(\boldsymbol{\alpha}\right)}{\partial \alpha_k\left(t\right)}$$

$$= \alpha_k\ \left(t\right) + \eta \left(1 - d_k \sum_{i\,=\,1}^{N} \alpha_i d_i K\left(\boldsymbol{x}_i, \boldsymbol{x}_k\right)\right). \tag{4}$$

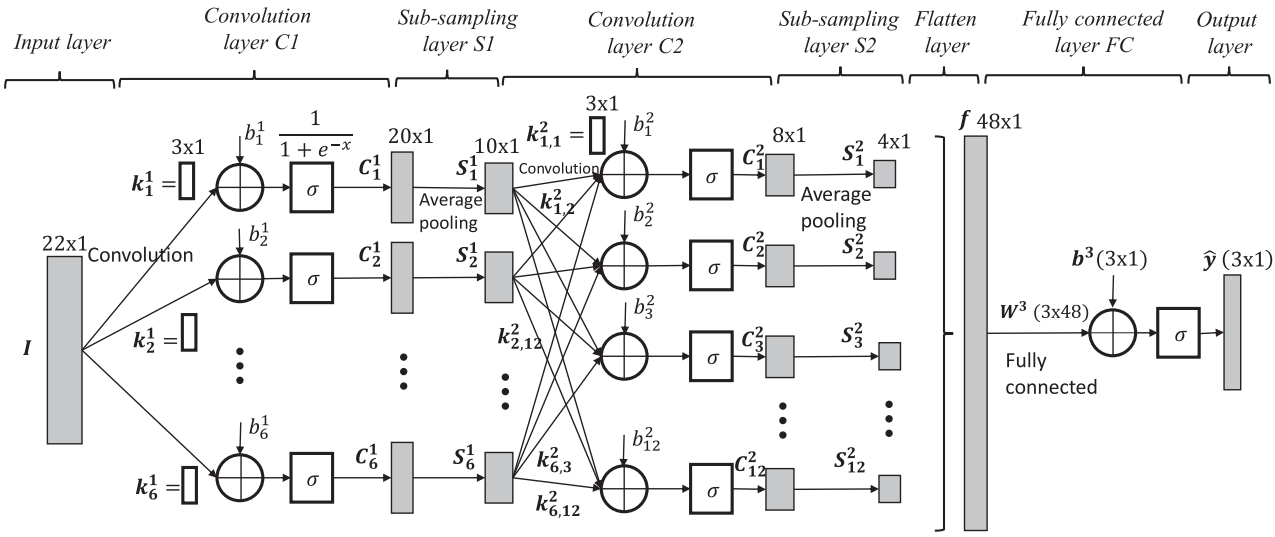


Fig. 17. 1-D CNN model for wavelet classification.

The change of L function, ΔL, due to the change of one α_k can be maximized and get the parameter η

$$\eta = \frac{1}{K\left(\boldsymbol{x}_k, \boldsymbol{x}_k\right)}. \tag{5}$$

So, we can use the Kernel–Adatron learning algorithm [36] to find each α_k. It means that we can input each training pattern $\boldsymbol{x_k}$ and by the calculation of (5) and (4) until convergence to get α_k, $k = 1, 2, ..., N$. Then, the classifier is in (3).

SVM is applied to find the classifier in Fig. 8(c). The used linear kernel function is

$$K\ (\boldsymbol{x}_i,\ \boldsymbol{x}) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}\right)\ = \phi^T\ (\boldsymbol{x}_i)\, \phi\,(\boldsymbol{x})\,.$$

The line classifier using SVM is plotted in Fig. 16. It is in the middle of the support vectors of two classes. Also, the line classifier using a perceptron is plotted in Fig. 16. It is close to the marginal patterns (vectors) of two classes. So, the SVM has a better classifier than that of the perceptron. But from the performance in Table II, the SVM takes more number of iterations and longer CPU time. Because the SVM has N Lagrange multipliers, $\alpha_k, k = 1, 2, ..., N$, but perceptron has only one two-feature weight vector and the associated learning-rate parameter α to be adjusted. So, to find a solution, perceptron is faster several times than SVM.

We ever used CNN in deep learning to classify Ricker wavelets in a seismogram [37]. Usually, 2-D CNN was used to classify 2-D image patterns. The 2-D CNN [38] was reduced to 1-D CNN that was used to classify the waveforms into Ricker wavelets with different amplitudes, frequencies, and polarities. The model is shown in Fig. 17. It has two convolution layers and one fully connected layer. Each convolution layer includes one convolution and one subsampling.

It was a supervised classification. The input was a waveform. It was from seismic data between two zero crossings. There were three classes at the output of 1-D CNN: 20-Hz Ricker wavelet, 30-Hz Ricker wavelet, and noise. For learning, we must design all possible waveforms of different amplitudes, frequencies, and polarities. That was a problem. The design of noise was a problem too. The weights in the model were trained in the backward propagation using a gradient descent method. The computation was complex.

The learning rule at the fully connected layer was

$$W^{3'}\ (i,j) = W^3\ (i,j) - \ \alpha \frac{\partial E}{\partial W^3\,(i,j)}$$

where $i = 1, ..., 3$, and $j = 1, ..., 48$, and L was the error function between the desired outputs and real outputs.

The learning rule at the second convolution layer was

$$k_{p,q}^{2'}\,(u) = k_{p,q}^2\ (u) - \ \alpha \frac{\partial E}{\partial k_{p,q}^2\,(u)}$$

where $p = 1, ..., 6$, and $q = 1, ..., 12$, and k was the coefficient in the filter.

The learning rule at the first convolution layer was

$$k_p^{1'}\,(u)\ \ = k_p^1\ (u) - \alpha \frac{\partial E}{\partial k_p^1\,(u)}$$

where $p = 1, ..., 6$.

The 1-D CNN was applied to the simulated seismogram for classification of 20 and 30-Hz Ricker wavelets and noise. The system performance is listed in Table II. Because of more coefficients to be adjusted, it requires longer CPU time. And for large learning-rate parameter α, it will be no convergence. The detected wavelets are not good in continuous horizon testing due to the noise effect in the waveform. But the results of our perceptron method are good in continuous horizon testing.

V. Conclusion and Discussion

In a seismogram, there exists a seismic anomaly that has the properties of high amplitude, low-frequency content, and polarity reversal in the wavelets. The envelope and instantaneous frequency can be generated through a complex signal analysis.

The envelope can describe the outer shape of the signal. And the instantaneous frequency can extract the internal frequency of the signal. They are used as features.

The perceptron is adopted to detect the seismic anomaly. Three conventional learning rules are used in the training of perceptron. They are the fixed-increment rule, absolute correction rule, and fractional correction rule. The optimal learning-rate parameter is derived. The lower and upper bounds of the learning-rate parameter are derived. It provides that the learning can converge when the parameter of the learning rule is within the range. The normalized learning rule is derived also. Combining learning rules, a fusion learning rule is proposed. Its learning-rate parameter is closer to the optimal learning-rate parameter than that of the other rules.

In the experiments, these rules are applied to the detection of a seismic anomaly in the simulated seismogram and to compare the convergence speed. The fusion learning rule has the fastest convergence and is applied to the real seismogram. The seismic anomaly can be detected successfully. It can improve the seismic interpretation.

It is a human handling to determine the training patterns from feature space to signal in time domain back and forth. Especially, the training patterns of two classes must be linearly separable. But the distribution of training patterns in the feature space may not be linearly separable. We may consider nonlinear classification methods to solve the problem.

The number of training patterns in a seismic abnormal is a few compared with that in the normal. It becomes an unbalanced sample problem.

The analysis of the learning-rate parameter may be used in the backpropagation learning rule of the multilayer perceptron and of CNN to get the fast convergence.

In seismic exploration, for the amplitude attenuation problem, the gain control can recover the weak to normal signal in the seismic data acquisition and data processing. For the frequency attenuation problem, the low-frequency content of the reflection wavelets at the top of the gas sand zone is still more significant than its neighboring layers. The low-frequency content plus the high amplitude and polarity reversal can have the three conditions of a seismic anomaly for detection.

ACKNOWLEDGMENT

The authors would like to thank K. Barry of the Teledyne Exploration for providing real seismic data.

REFERENCES

[1] C. H. Chen, "Seismic pattern recognition," *Geoexploration*, vol. 16, pp. 133–146, 1978.
[2] K. R. Anderson, "Automatic analysis of microearthquake network data," *Geoexploration*, vol. 16, pp. 159–175, 1978.
[3] P. Bois, "Autoregressive patterns recognition applied to the delimitation of oil and gas reservoirs," *Geophys. Prospecting*, vol. 28, pp. 572–591, 1980.
[4] C. H. Chen, "A review of geophysical signal analysis and recognition," in *Proc. 2nd Int. Symp. Comput.-Aided Seismic Anal. Discrimination*, 1981, pp. 144–152.
[5] K. R. Anderson, "Syntactic analysis of seismic waveforms using augmented transition network grammars," *Geoexploration*, vol. 20, pp. 161–182, 1982.
[6] H.-H. Liu and K. S. Fu, "A syntactic approach to seismic pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 2, pp. 136–140, Mar. 1982.
[7] K. Y. Huang and K. S. Fu, "Decision theoretic pattern recognition for the classification of Ricker wavelets and the detection of bright spots," in *Proc. 52nd Annu. Meeting Soc. Exploration Geophys.*, Dallas, TX, USA, 1982, pp. 222–224.
[8] C. H. Chen, "Pattern analysis of acoustical and seismic events," in *Proc. IEEE 3rd Int. Symp. Comput.-Aided Seismic Anal. Discrimination*, Jun. 15–17, 1983, pp. 114–118.
[9] P. Bois, "Some applications of pattern recognition to oil and gas exploration," *IEEE Trans. Geosci. Remote Sens.*, vol. GE-21, no. 4, pp. 416–426, Oct. 1983.
[10] K.-Y. Huang and K.-S. Fu, "Detection of bright spots in seismic signals using tree classifiers," *Geoexploration*, vol. 23, pp. 121–145, 1984.
[11] F. Aminzadeh and S. Chatterjee, "Applications of clustering in exploration seismology," *Geoexploration*, vol. 23, pp. 147–159, 1984.
[12] K.-Y. Huang and K.-S. Fu, "Decision-theoretic approach for classification of Ricker wavelets and detection of seismic anomalies," *IEEE Trans. Geosci. Remote Sens.*, vol. GE-25, no. 2, pp. 118–123, Mar. 1987.
[13] Y. Shimshoni and N. Intrator, "Classification of seismic signals by integrating ensembles of neural networks," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1194–1201, May 1998.
[14] K.-Y. Huang, "Neural networks for seismic principal components analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 297–311, Jan. 1999.
[15] K.-Y. Huang, K.-J. Chen, J.-D. You, and A.-C. Tung, "Hough transform neural network for pattern detection and seismic applications," *Neurocomputing*, vol. 71, pp. 3264–3274, Oct. 2008.
[16] K.-J. Huang, K.-Y. Huang, I.-C. Chen, and L. K. Wang, "Simulated annealing for sequential pattern detection and seismic applications," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4849–4859, Dec. 2014.
[17] C. H. Chen, Ed., *Handbook of Pattern Recognition and Computer Vision*, 6th ed. Singapore: World Scientific, Apr. 2020.
[18] K. Y. Huang and W.-H. Hsieh, "Cellular neural network for seismic pattern recognition," in *Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, Ed., 6th ed. Singapore: World Scientific, Apr. 2020.
[19] M. B. Dobrin and C. H. Savit, *Introduction to Geophysical Prospecting*, 4th ed. New York, NY, USA: McGraw Hill, 1988.
[20] C. E. Payton, Ed., *Seismic Stratigraphy—Applications to Hydrocarbon Exploration*, AAPG Memoir 26, Tulsa, OK, USA: Amer. Assn. Petroleum Geologists, 1977.
[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.
[22] D. E. Rumelhart, G. E. Hilton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
[25] N. J. Nilsson, *The Mathematical Foundations of Learning Machines*. California, CA, USA: Morgan Kaufmann, 1990.
[26] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Boston, MA, USA: Academic, 1972.
[27] K. S. Fu, *Syntactic Pattern Recognition Application*. New Jersey, NJ, USA: Prentice-Hall, 1982.
[28] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. New Jersey, NJ, USA: Pearson Education, 2010.
[29] M. Basu and Q. Liang, "The fractional correction rule: A new perspective," *Neural Netw.*, vol. 11, pp. 1027–1039, Aug. 1998.
[30] M. T. Tanner, F. Koehler, and R. E. Sheriff, "Complex seismic trace analysis," *Geophysics*, vol. 44, pp. 1041–1063, 1979.
[31] A. Gholamy and V. Kreinovich, "Why Ricker wavelets are successful in processing seismic data: Towards a theoretical explanation," in *Proc. IEEE Symp. Comput. Intell. Eng. Solutions*, 2014, pp. 11–16.
[32] K.-Y. Huang and K.-S. Fu, "Syntactic pattern recognition for the recognition of bright spots," *Pattern Recognit.*, vol. 18, no. 6, pp. 421–428, 1985.
[33] K.-Y. Huang and D.-R. Leu, "Syntactic pattern recognition for wavelet clustering in seismogram," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 7, pp. 2453–2461, Jul. 2019.

[34] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys. Dokl.*, vol. 10, pp. 707–710, 1966.
[35] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 50th Annu. Workshop Comput. Learn. Theory*, Jul. 1992, pp. 144–152.
[36] T.-T. Friess, N. Cristianini, and I. C. G. Campbell, "The Kernel-Adatron algorithm: A fast and simple learning procedure for support vector machines," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 188–196.
[37] K. Y. Huang and F. Abdurrahman, "1D convolutional neural network for wavelet classification in seismogram," in *Proc. 32nd IPPR Conf. Comput. Vision, Graphics, Image Process.*, Aug. 25–27, 2019, Art. no. OT-0019.
[38] Z. Zhang, "Derivation of backpropagation in convolutional neural network (CNN)," Tech. Rep., Univ. Tennessee, Knoxville, TN, USA, Oct. 2016.

Kou-Yuan Huang (Senior Member, IEEE) received the B.S. degree in physics and the M.S. degree in geophysics from the National Central University, Chungli, Taiwan, in 1973 and 1977, respectively, and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1980 and 1983, respectively.

He is currently a Professor with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. He has authored four books: *Syntactic Pattern Recognition for Seismic Oil Exploration* (World Scientific, vol. 46, 2002), *Neural Networks and Pattern Recognition* (Weikeg, 2nd ed., 2003), *Neural Networks* (Chuan-Hwa Publishing, 4th ed., 2018), and *Syntactic Pattern Recognition* (Chuan-Hwa Publishing, 2018). His research interests include deep learning, machine learning, neural networks, pattern recognition, fuzzy logic, evolutionary computation, signal processing, image processing, and applications, especially to geophysical exploration data analysis and well log data inversion.

Jiun-Der You, biography not available at the time of publication.

Fajar Abdurrahman was born in Bandung, West Java, Indonesia, in 1994. He received the B.A.Sc. (SST) degree in electrical engineering from the Bandung State Polytechnic, Bandung, Indonesia, in 2016. He is currently working toward the master's degree with National Chiao Tung University (NCTU), Hsinchu, Taiwan.

During his college study, he took an internship in one of the government research institutions in Indonesia. From 2016 to 2018, he was an R&D Engineer in PT.Hariff DTE, which is a leading technology company in Indonesia. He is currently a Research Assistant of Neural Network and Pattern Recognition Laboratory, NCTU, Taiwan.