Inversion of Fracture Parameters by Using the Artificial Neural Network

Hengchang Dai, Xiang-Yang Li, and Colin MacBeth
British Geological Survey,
Murchison House, West Mains Road, Edinburgh EH9 3LA, UK

Abstract: *A BPNN approach is developed to invert the fracture parameters from Thomsen parameters. A BPNN is trained by pairs of fracture parameters and Thomsen parameters which are calculated by a complex relationship. After training, the BPNN can emulate the inverted relationship between them and then to invert fracture parameters from Thomsen parameters. The results are encouraging. This BPNN approach can invert the fracture parameter with high accuracy. It shows that this BPNN approach can be used as a general simulation tool to resolve complicated relationship between sets of input and output. Although this work only applied to the theoretical relationship, it also shows the potential of applying BPNN to inversion problem with experiment data so that a comprehensive solution can be obtained by combining them.*

Keywords: Back-propagation neural network, Inversion, Fracture, Thomsen parameter, Transverse isotropic.

1 Introduction

In exploration seismology, one important aim is to detect the details of reservoir fractures that include the fracture density, the aspect ratio, and the medium's *P*- and *S*-velocities. Although such fracture parameters cannot be directly measured from seismic data, they can be inverted from other parameters determined from the seismic data. Many authors have developed the relationship between the fracture parameters and these parameters. For example, Thomsen[1] introduced three parameters for a transversely isotropic medium. They can be obtained by measuring the elastic wave anisotropy from seismic data. However, their relationships to the fracture parameters are complicated and the fracture parameters are only approximately resolved under some restrictions. There is thus a pressing need to invert the fracture parameter from Thomsen parameters exactly and efficiently. Due to the non-linearity and indeterminacy of the problem, conventional methods do have limitations to resolve it. New methods are necessary to be developed.

Artificial neural networks (ANNs) provide an alternative method to resolve this kind of inversion problem. ANNs are simple models that attempt to simulate the operation of neurons in the human brain. They had been widely used in various areas of geophysics, such as seismic event classification [2][3], lithology log estimation [4], First break or arrival picking [5][6][7][8], earthquake prediction [9], Seismic data inversion [10][11]. The wide range of applications emphasize the particular strength of ANNs over traditional methods incorporating a fixed algorithm to solve a particular problem, as the ANNs utilize a learning scheme to develop an appropriate solution so that the ANNs is flexible and adaptive to different data sets.

The ANN used in the work is the back-propagation neural network (BPNN). It can simulate the complicated relationship between input and output parameter [12][13]. This relationship may be mathematical functions from existing theories, or measured data. In this paper, we will develop a BPNN approach to invert fracture parameters. We first introduce the relationship of Thomsen parameters and fracture parameters for a transversely isotropic medium. The BPNN is then applied to simulate the relationship between the two sets of parameters and finally to invert the fracture parameters from Thomsen parameters.

2 Thomsen parameter and fracture parameters

A transversely isotropic medium can be considered as an isotropic medium with fractures in it [14]. Its properties can be determined by the fracture density, the fracture aspect ratio, and the *P*- and *S*-wave velocities of the isotropic medium, as well as the *P*-wave velocity of the fluid within the fracture. These fracture parameters can be linked with Thomsen parameters, ε, δ, and γ [1]. They can be written as:

$$\varepsilon = \frac{2(1-r^2)\varepsilon_d M}{1-r^2\varepsilon_d M}; \quad \delta = 2\frac{\varepsilon_d M - r^2\varepsilon_d N}{1-r^2\varepsilon_d M};$$

$$\gamma = \frac{\varepsilon_d N}{2(1-\varepsilon_d N)}. \quad N = \frac{16}{3}\frac{1}{3-2r^2}; \tag{1}$$

$$M = \begin{cases} \frac{4}{3} r^2 r_f^2 & \text{for fluid - satuated fracture} \\ \frac{4}{3}\frac{1}{1-r^2} & \text{for dry fracture} \end{cases}.$$

Where $r = \frac{\beta_b}{\alpha_b}$ and $r_f = \frac{\alpha_b\sqrt{\pi\varepsilon_{ar}}}{\alpha_f}$. α_b is *P*-velocity, β_b *S*-velocity, ε_d fracture density, ε_{ar} aspect ratio and α_f fluid velocity.

The advantage of using the Thomsen parameters is that they directly link to the measurable quantities from seismic data. Here, ε and γ are, respectively, the amount of *P* and *S* anisotropy, while δ is related to both the amount of anisotropy and the deviation of wavefront from an elliptical shape. The Thomsen parameters give us a way to detect the fracture parameters by measuring the seismic data. Note that Thomsen parameters depend on combinations of the fracture parameters, not the individuals.

Once the Thomsen parameters are observed from seismic data, the fracture parameters might be inverted from Equation 1. Mathematically, the relationship between Thomsen parameters and fracture parameters can be written as:

$$\varepsilon = F_1(\varepsilon_d, r, r_f)$$
$$\delta = F_2(\varepsilon_d, r, r_f) \tag{2}$$
$$\gamma = F_3(\varepsilon_d, r, r_f)$$

Thus F_1^{-1}, F_2^{-1}, and F_3^{-1} are the inverse functions. Unfortunately, it is not easy to obtain the inverse functions because Equation 1 is non-linear and indeterminate. In order to resolve this, some authors use an approximate method to simplify the problem under restrictions [14]. However, BPNNs can provide an alternative method to simulate the inverse relationship. Here a BPNN approach is developed to invert the fracture parameters from Thomsen parameters. Because, there are no fundamental differences in applying BPNN for the dry fracture and fluid-satuated fracture, for simplicity we only consider the dry fracture case, e.g., inverting only ε_d and *Vs/Vp*.

3 Back-Propagation Neural Network

The BPNN used in this paper is a multi-layer feed-forward neural network trained by the generalized delta rule [15]. It is made up of an input layer, some hidden layers, and an output layer of nodes. Figure 1 shows the BPNN used in this work. Each node has a nonlinear activation function. The outputs of the nodes in one layer are transmitted to nodes in another layer through weighted links, which are applied as simple multiplication scalars, and effectively amplify or attenuate the signals. With the exception of the input nodes, the input to each node is the sum of the weighted outputs of nodes in the previous layer. Each node is then activated in accordance with the summed input using its activation function and a threshold for the

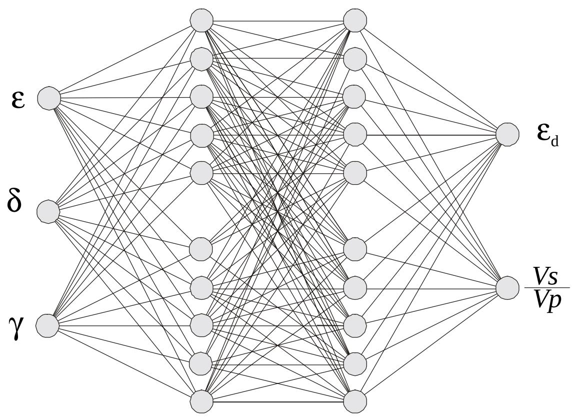

Figure 1. A BPNN structure used for inverting fracture parameters from Thomsen parameters. Three nodes are in the input layer and two nodes in the output layer. Two hidden layers have the same number nodes.

function. The inputs to input nodes are the components of the input pattern. The signals feed through the network only in a forward direction.

The ability of a BPNN to solve a problem comes from emulating the natural learning (or training) procedure. The BPNN is trained by using the generalized delta rule. This method attempts to find the most suitable solution (numerical values of weights and thresholds) for a global minimum in the mismatch between the desired output pattern and its actual value for all of the training examples. The degree of mismatch for each input-output pair is quantified by solving for unknown parameters between the hidden and output layer and then by propagating the mismatch backwards through the network to adjust the parameters between the input layer and hidden layer. After training, the set of weights and thresholds in the network are now specifically tailored to "remember" each input and output pattern and can consequently be used to recognize or generate new patterns given an unknown input.

4 The approach

In this approach, a BPNN will be used to simulate the inverse relationship between the fracture parameters and Thomsen parameters. The training dataset, a set of input-output pairs is created by using Equation 1 to calculate the Thomsen parameters for a set of given fracture parameters. These input-output pairs simulate Equation 1. To obtain the inverse relationship, we set the Thomsen parameters as input and the fracture parameters as the output during training. Once trained, the relationship between the input and output of the BPNN simulates the inverse relationship between the fracture parameter and Thomsen parameters and this trained BPNN can be used to invert fracture parameters from measured Thomsen parameters.

4.1 Selecting the training dataset

The most important thing is to select the suitable training dataset, a set of input-output patterns. The Thomsen parameters, ε, δ and γ are calculated from given fracture parameters, ε_d and *Vs/Vp* by using the Equation 1. Although, mathematically, both ε_d and *r* can have a value ranging between zero and one, the real data cannot reach the boundary values. Here, we select ε_d from 0.04 to 0.40 with increment of 0.04 and *Vs/Vp* from 0.1 to 0.8 with increment of 0.05 Figure 2 shows the training dataset.

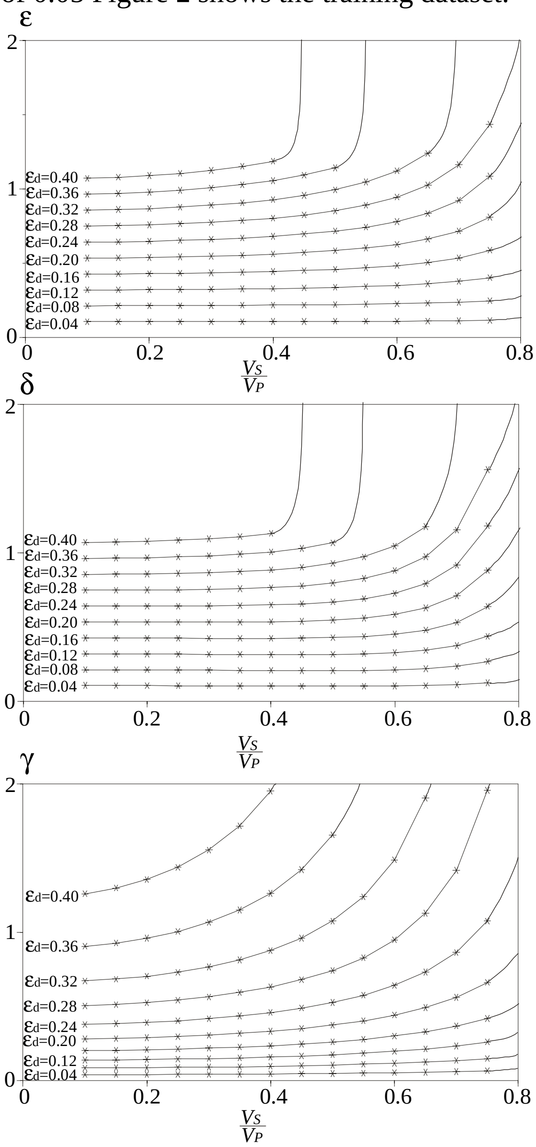

Figure 2. The relationship between the fracture parameters and Thomsen parameters. ε, δ, and γ are calculated by using Equation (1) for given *Vs/Vp* and ε_d. The asterisk (*) indicates the training data pair.

4.2 Selecting the structure of BPNN

The BPNN used has four layers and is trained by generalized delta rule [15]. The inputs to the BPNN are the Thomsen parameters: ε, δ and γ.

The outputs from the BPNN are ε_d and *Vs/Vp*. (Figure 1) This means the input layer has three nodes and the output layer has two nodes. The numbers of hidden nodes depends on various factors such as input nodes, output nodes, training pattern numbers, and the performance of the BPNN. There is no fixed generic relationship between the number and these factors [16]. Their numbers are usually chosen after a process of trial and error with different training runs. Generally, we know that the learning cannot converge with too small hidden nodes and too many hidden nodes result in an over-trained problem. In order to select a suitable structure, we trained several BPNNs with different hidden nodes. Figure 3 shows the training results. From this we can find that for the BPNN with 10 hidden nodes, the error curve is smooth and gradually decreases. For the BPNN with 6 or 8 hidden nodes, the system error is large and not stable. Although the system error is smaller for the BPNN with 12 or 14 hidden nodes than that for the BPNN with 10, the curve oscillating. Even there is a big jump at iteration=165000 for BPNN with 8 and 14 hidden nodes. It shows that there exists a optional structure for this inversion problem. Finally ten nodes are selected for both two hidden layers in this approach.

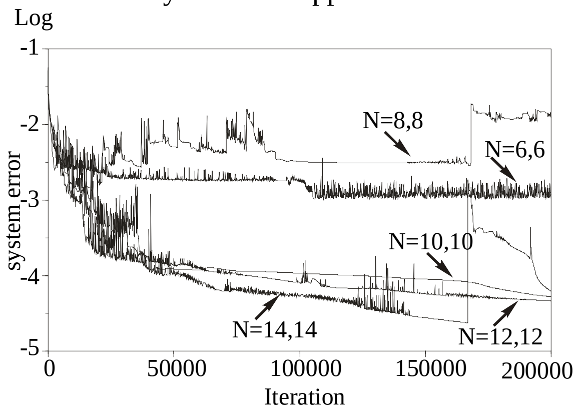

Figure 3. The system error against the iteration number during training procedure for BPNNs with different hidden nodes. Each BPNN has two hidden layer with the same nodes.

5 Results and discussions

5.1 Training Procedure

According the above, we select 126 pairs of input-output patterns (Figure 2) to train the BPNN. The learning rate, 0.7 and the momentum rate 0.9 are selected because the two values commonly used [17]. The training took 200,000 iterations (about two hours on a Sun Ultra 1 station) with system error 0.000052.

5.2 Testing procedure

5.2.1 Testing for training data

After training, we feed the training dataset to the BPNN again to check its performance. Figure 4 shows the performance of the trained BPNN for the training dataset. Each plot in Figure 1 shows the comparison between the desired output and BPNN output for given Thomsen parameters. For ε_d, the BPNN output precisely match the desired output. Only a few of them were in disagreement. For *Vs/Vp,* most are matched well with a few of them mismatch when ε, δ, γ and *Vs/Vp* are small. However the overall performance of the trained BPNN is very good.

The mismatch is due to Equation 1 itself. Theoretically, the errors of fracture parameters are:

$$\Delta(\frac{Vs}{Vp}) \propto \frac{1}{\varepsilon_d(\frac{Vs}{Vp})}[\Delta\varepsilon, \Delta\delta, \Delta\gamma]$$

and (2)

$$\Delta\varepsilon_d \propto [\Delta\varepsilon, \Delta\delta, \Delta\gamma].$$

When ε_d and *Vs/Vp* are small, Δ(*Vs/Vp*) are enlarged. However, for large ε_d and *Vs/Vp,* Δ(*Vs/Vp*) is similar with $\Delta\varepsilon$, $\Delta\delta$, or $\Delta\gamma$. $\Delta\varepsilon_d$ is always similar with $\Delta\varepsilon$, $\Delta\delta$, or $\Delta\gamma$. Because the BPNN is not perfectly trained with small error, this will results large error on *Vs/Vp* for small ε_d and *Vs/Vp.*

5.2.2 Testing for new data

In order to test its generating ability, we create another set of input-output pairs calculated by using Equation 1 with different values of Thomsen parameters (Figure 5) Here, ε_d are from 0.02 with 0.04 increment and *Vs/Vp* are from 0.125 with 0.05 increment. We then feed the Thomsen parameters into the trained BPNN and the trained BPNN generated its output. Figure 6 shows the performance. For ε_d, the error between the BPNN output and desired

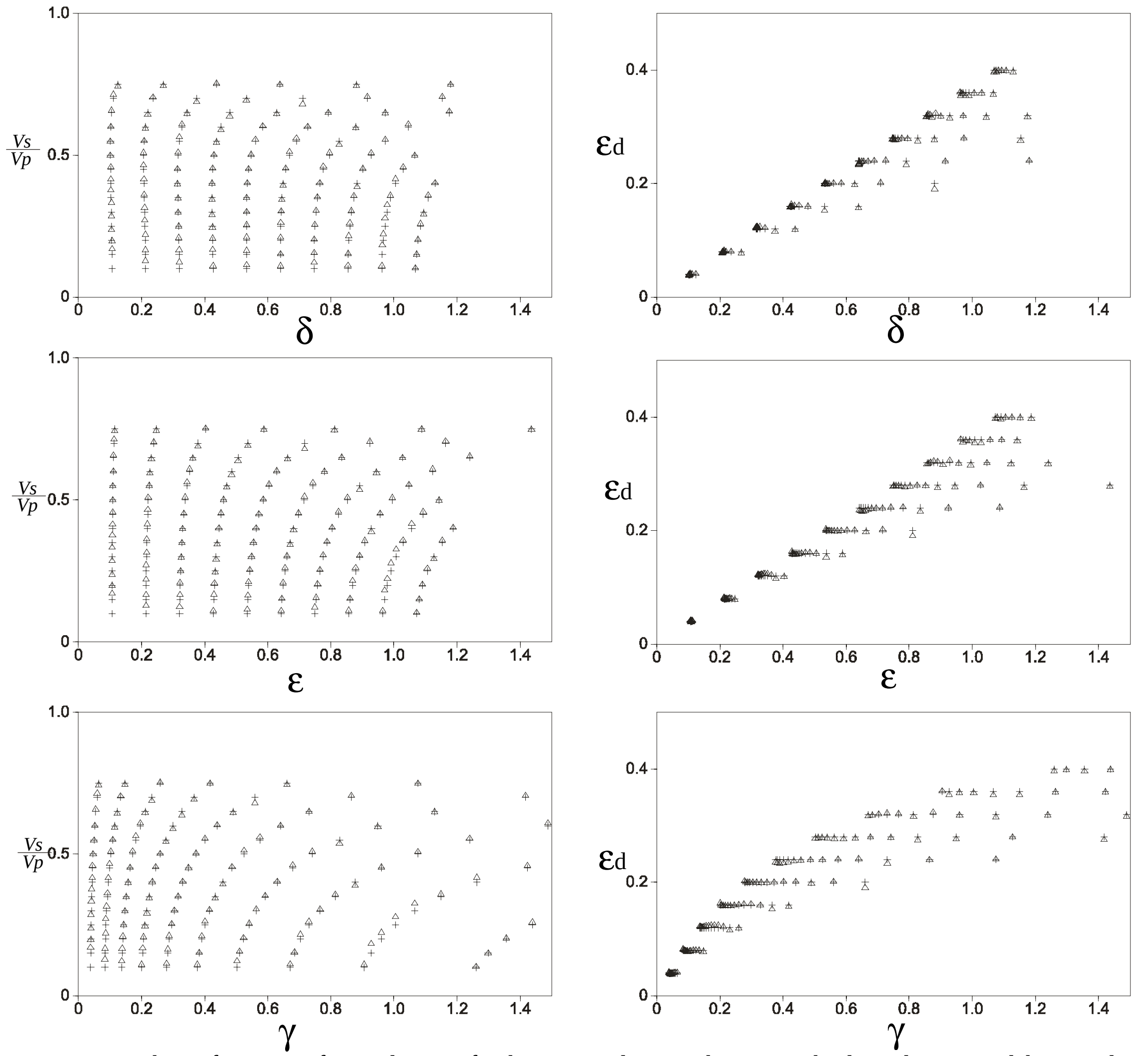

Figure 4. The performance of trained BPNN for the training dataset. The cross is the desired output and the triangle is the BPNN's output

output is similar to the error of training data except few points. For *Vs/Vp,* the error is a little bit larger than the error from the training data, but they are still within the allowed range. We must also note that the error depends on the distribution of training dataset. Although we calculate ε, δ and γ by giving ε_d and *Vs/Vp* the equal increment, the resulting ε, δ and γ do not have the equal increment which is proportional to the value of ε_d and *Vs/Vp,* the resulted density of training data points is not equal. In the area of high density, the BPNN is well trained, and the error is small. Otherwise, the error is large. It shows that the training dataset play an important role in the BPNN performance. This also gives us a clue to improve the BPNN performance by selecting suitable training dataset.

6 Conclusions

We have developed a BPNN approach to invert fracture parameters from Thomsen parameters for the case of dry fracture, The test results show that the trained BPNN can simulate the inverse relationship between the Thomsen parameters and the fracture parameters. This work shows that the BPNN can be used as a general function simulator to define a complicated relationship between sets of input and output.

In this paper, the relationships are formulae from an existing theory. Note that there are different relationships developed by different authors. Using this approach, we can train the BPNN to simulate all of them. We can also train a BPNN to simulate a relationship from experimental data. Because every theory has its limitations, we might combine the results from different theories and the experimental data to train a BPNN. This trained BPNN may overcome their limitations to give a comprehensive solution.

This approach yields encouraging results and shows the potential of using the BPNN for the inversion problem. In further work, we need to investigate factors that affect the BPNN performance. We will also apply this inversion approach to the case of fluid-saturated fractures and to experimental data.

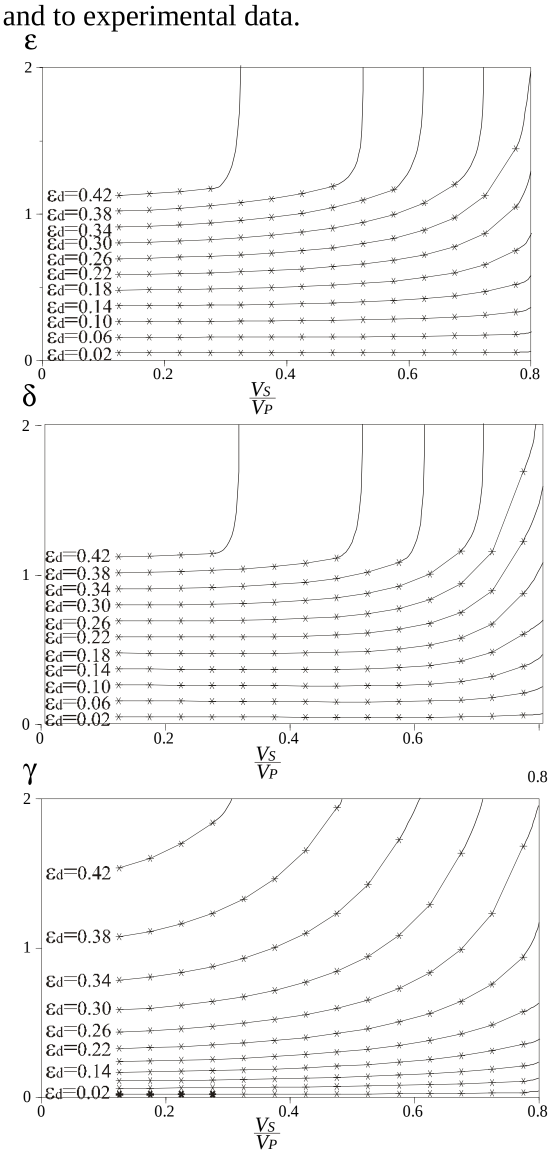

Figure 5. The testing dataset. Notation as Figure 3.

Acknowledgements

This work is presented with approval from the Director of the British Geological Survey and EAP sponsors: Amerada Hess, BG plc, BP-Amoco, Chevron, Conoco, Elf UK, ENI-Agip Fina, Mobil, PGS, Phillips, Saga Petroleum, Schlumberger, Shell, Texaco, and Veritas DGC.

References

[1] Thomsen, L. A., 1986, Weak elastic anisotropy, *Geophysics*, 51, 1954-1966.

[2] Dystart, P. S. and Pulli, J. J., 1990, Regional seismic event classification at the NORESS array: seismological measurements and the use of trained neural network, *Bulletin of Seismological Society of America,* **80**, 1910-1933

[3] Dai, H. C., and MacBeth, C., 1997, Application of back-propagation neural networks to identification of seismic arrival types, *Physics of Earth and planetary Interiors,* **101**, 177-188.

[4] McCormack, M. D., 1991, Neural computing in geophysics, *Geophysics: The leading Edge of Exploration,* **10,** 11-15, January 1991.

[5] Murat, M., and Rudman, A., 1992, Automated first arrival picking: A neural network approach, *Geophysical Prospecting*, **40**, 587-604.

[6] McCormack, M. D., Zaucha, D. E., and Dushek, D. W., 1993, First-break refraction event picking and seismic data trace editing using neural network, *Geophysics*, **58**, 67-78

[7] Dai, H. C., & MacBeth, C., 1995, Automatic picking of seismic arrivals in local earthquake data using an artificial neural network, *Geophysical Journal International*, **120**, 758-774.

[8] Dai, H. C., & MacBeth, C., 1997, The application of back-propagation neural network to automatic pick seismic arrivals from single-component recordings. *Journal of Geophysical Research*, **102, No B7,** 15105-15113.

[9] Feng, X-T, Seto, M, and Katsuyama, K, 1997, Neural dynamic modelling on earthquake magnitude series, *Geophysical*

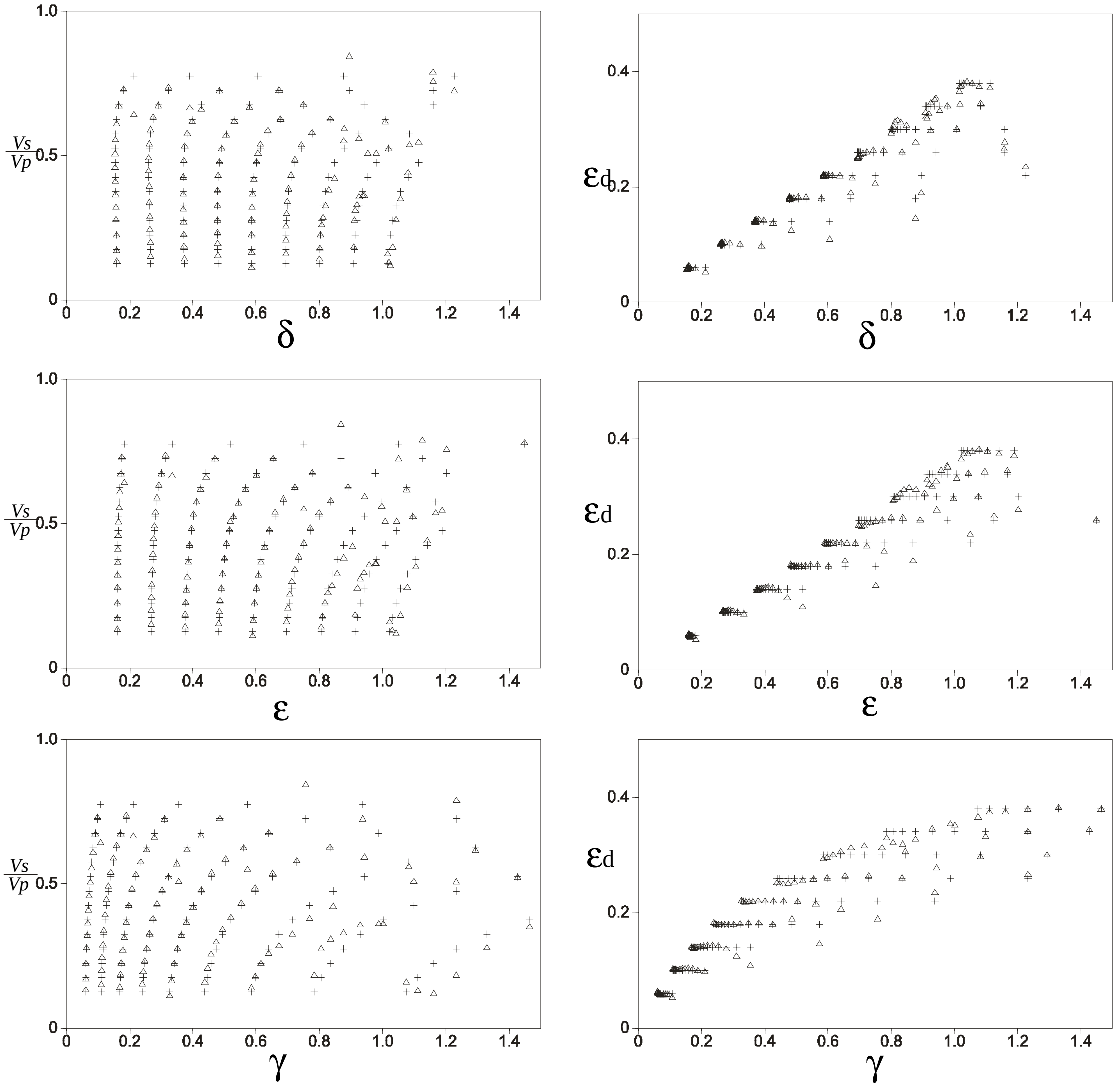

Figure 6. The performance of trained BPNN for the testing dataset. The cross is the desired output and the triangle is the BPNN's output

Journal International, **128**, 547-556

[10] Roth, G., and Tarantola, A., 1994, Neural networks and Inversion of seismic data, *Journal of Geophysical Research*, **99**, No B4, 6753-6768.

[11] Boadu, F., 1998, Inversion of fracture density from field seismic velocities using artificial neural networks, *Geophysics*, **63,** 534-545.

[12] Demuth, H. and Beale, M., 1993, *Neural network toolbox user's guide,* The MathWorks Inc, MA.

[13] Hudson, J.A., 1981, Wave speeds and attenuation of elastic waves in material containing cracks, *Geophysical Journal of the Royal Astronomical Society*, **64**, 133-150.

[14] Li, X. Y., 1997, Viability of azimuthal variation in P-wave moveout for fracture detection: *Expanded Abstracts of 67th International Meeting of Society of Exploration Geophysicists*, 1555-1558.

[15] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, Learning internal representation by backpropagating errors, *Nature*, **323**, 533-536.

[16] Fausett, L., 1994, *Fundamentals of neural networks*, Prentice Hall, Inc, NJ.

[17] Dai, H. and MacBeth, C., 1997, Effects of learning parameters on learning procedure and performance of a BPNN, *Neural networks*, **10,** 1505-1521.