Estimation of Thomsen's anisotropy parameters using NMO equations and neural networks

Amber C. Kelter and John C. Bancroft

ABSTRACT

Many models in exploration seismology naively presume that the earth is isotropic, that is, seismic velocities do not vary with direction. Yet individual crystals and most common earth materials are observed to be anisotropic with elastic parameters that vary with orientation (Shearer, 1999). Thus, it would be surprising if the earth was entirely isotropic. Further, it is now commonly accepted that most upper crustal rocks are anisotropic to some extent (Crampin, 1981) and more recently it has become apparent that anisotropy is evident in many other parts of the earth (Shearer, 1999). In addition, alternating layering of high and low velocities where the thickness of the layer is less than the wavelength of the seismic signal will also appear anisotropic (Backus, 1991).

Although seismic processors have been aware of anisotropy, it was ignored because its effect was successfully absorbed into the stacking velocity when processing horizontally layered media. However, erroneous assumptions of an isotropic velocity lead to flawed images and thus incorrect interpretations where targets can appear shifted both laterally and vertically (Isaac et al., 2004).

NEURAL NETWORKS

An artificial neural network (ANN) is an information processing algorithm that is inspired by the way biological nervous systems process information. In the simplest sense a neural network is a mathematical algorithm that can be trained to solve a problem. The key element is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems (Haykin, 1999). Artificial neural networks, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. The learning process in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANN's as well. In the work that follows ANN's are referred to simply as neural networks or NN.

Multilayer feedforward network

The architecture of a multilayer feedforward network is illustrated in FIG. 1. A layer consists of a weight matrix, a bias vector, neuron(s) and an output vector.

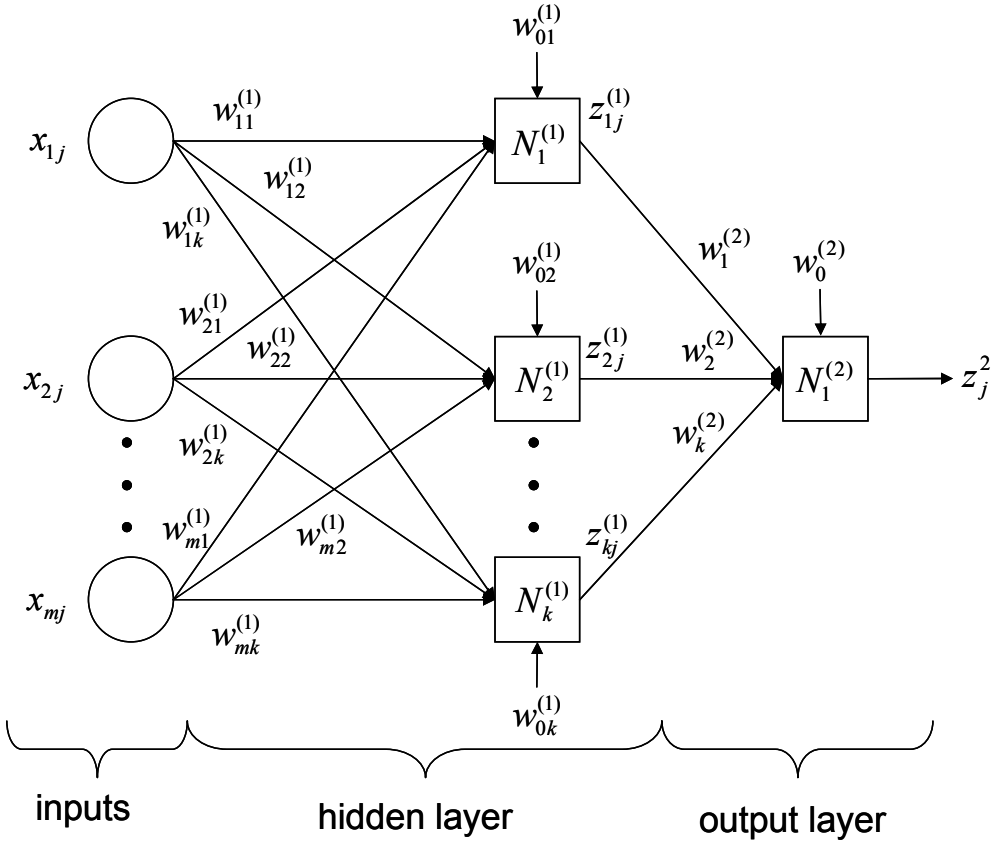

FIG. 1. Flowchart of a multilayer feedforward network with m inputs, k neurons and 1 output (adapted from Russell, 2005).

A hidden layer resides between the input and output layer. There can be any number of 'hidden' layers, however it has been shown that networks with biases, a single sigmoid layer (hidden layer) and a single linear layer (output layer) are capable of approximating any function with a finite number of discontinuities provided that there are enough neurons (Higham and Higham, 2000). A single hidden layer is used in all of our applications. To distinguish between the weights, outputs, etc., for the different layers we append the number of the layer as a superscript to the variable of interest.

From FIG. 1 and following Russell (2004) the input to the multilayer feedforward network is a vector, x, of m attributes where $j = 1,...,N$ is indicative of the number of seismic samples. Weights for the first layer connect inputs and neurons and are written as w_{ik}^1, where i represents the input attribute number and k represents the neuron number. Each neuron, N_K^1, consists of weights, a summation step, and a transfer function. The weighting and summation step in the first layer is written as

$$y_{Kj}^{(1)} = \sum_{i=0}^{m} w_{Ki}^{(1)} x_{ij}, K = 1,...,k, j = 1,...,N \,. \quad (1)$$

In Equation 1, a bias term, w_{0K}, has been included by letting $x_{oj} = 1$. The sum of the weighted inputs and the bias forms the input to the transfer function *f*. The transfer function is written as

$$z_{Kj}^{(1)} = f^{(1)}(y_{Kj}^{(1)}) \,. \quad (2)$$

The output from layer 1 is then fed into layer 2. Again a bias term is incorporated by letting $z_{0j}^{(1)} = 1$, thus the input into the output layer will have k+1 weights and can be expressed as

$$y_j^{(2)} = \sum_{K=0}^{k} w_K^{(2)} z_{Kj}^{(1)}, l = 1,...,N \,. \quad (3)$$

The output is written as

$$z_j^{(2)} = f^{(2)}(y_j^{(2)}) \,. \quad (4)$$

Two commonly used non-linear transfer functions are the tan-sigmoid and log-sigmoid functions. A tan-sigmoid function is defined as $f(n) = \frac{2}{1 - e^{-2n}} - 1$ and a log-sigmoid function is defined as $f(n) = \frac{1}{1 + e^{-n}}$.

Backpropagation

Backpropagation is used to train feedforward networks. In the backpropagation algorithm input vectors and the corresponding output target vectors are used to train a network until it can approximate a function. During training the weights and biases of the network are iteratively adjusted to maximize the network minimizing the difference between estimated and known values.

Data are trained in batch mode where the weights and biases of the network are updated only after the entire training set has been applied to the network. In the application of the backpropagation algorithm two distinct passes of computation are performed; first a forward pass and second a backward pass (Haykin, 1999; Freeman and Skapura, 1991). In the forward pass the weights remain unaltered throughout the network and are used to compute the output of the network on a neuron by neuron basis. The output is then compared with the desired response to obtain an error signal. The backward pass, on the other hand, starts at the output layer by passing the error signals leftward through the network, layer by layer, and recursively computing the local gradient in weight space for each neuron such that a correction is computed for the weights that is proportional to the partial derivatives. The basic procedure is embodied by Freeman et al. (1991) as:

1. Apply an input vector to the network and calculate the corresponding output values
2. Compare actual outputs with the correct outputs and determine a measure of the error
3. Determine in which direction to change each weight in order to reduce the error
4. Determine the amount by which to change the weight
5. Apply the corrections to the weights

Repeat 1-5 with all training samples until the error is reduced to an acceptable value.

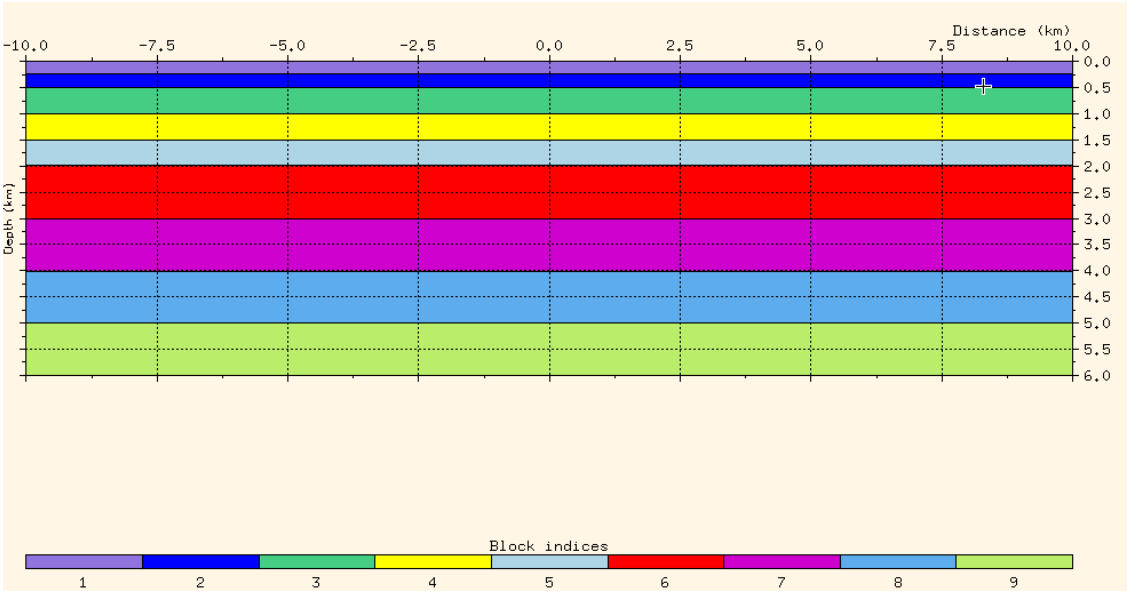

FIG. 2. Geological model used to create synthetic seismic section. Model consists of 9 horizontal layers each with unique material properties

Table 1. Material properties of the synthetic model

Layer	P-wave Velocity	S-wave Velocity	Density	ε	δ
	(m/s)	(m/s)	(kg/m^3)	(unitless)	(unitless)
1	1000	500	1.1	0	0.2
2	1200	600	1.2	0.05	0.25
3	1500	750	1.3	0.1	0.3
4	2000	1000	1.5	0.15	0.1
5	2500	1250	1.7	0.2	0.15
6	3000	1500	1.9	0.25	0.2
7	4000	2000	2.2	0.3	0.25
8	5000	2500	2.4	0.2	0.3
9	6500	3250	2.6	0.1	0.3

SYNTHETIC MODELLING

Synthetic seismograms were generated using NORSAR2D. The geologic model consists of nine horizontal layers each with its own unique material properties. The model was 20.0 km long and 6 km deep. The thinnest layer was 0.25 km and the thickest 1.0 km. The material properties of P-wave velocity, S-wave velocity, density, ε and δ, were assigned for each layer as listed in Table 1.

Ray tracing was performed on the model using NORSAR2D's anisotropic ray tracer. Two seismic surveys were simulated. Velocities were estimated from short offset semblance analysis and combined with vertical well-log velocity information to estimate the anisotropic parameters ε and δ. Note that if anisotropy is present these velocity values will differ; the amount of difference is dependant on the degree of anisotropy. Velocities used in these algorithms are interval velocities that are derived from the RMS velocities.

A summary of the parameters used in the neural network analysis is provided in Table 2. Data type refers to the type of data used to acquire interval velocities i.e. P-wave data (PP), PS-wave data (PS) or P- and PS- wave data (PP & PS), output is the anisotropic parameter that is being estimated, number of neurons is the number of neurons in the hidden layer, training/inputs are the inputs to the network and transfer function is the type of transfer function used in each layer. The first type listed is the transfer function in the hidden layer and the second the type listed is the transfer function used in the output layer. The number of neurons in the output layer is always equivalent to the number of outputs being estimated.

Table 2. Summary of parameters used in neural networks

Data Type	Output	Number of Neurons	Training/Inputs	Transfer Functions
PP	δ	29	Vp(0), Vs(0), Vp_int	Tan-sigmoid, Linear
PP	δ,ε	88	Vp(0), Vs(0), Vp_int	Tan-sigmoid, Linear
PP	ε	32	Vp(0), Vs(0), Vp_int, δ	Linear, Linear
PS	δ	9	Vp(0), Vs(0), Vps_int	Tan-sigmoid, Linear
PS	δ,ε	3	Vp(0), Vs(0), Vps_int	Tan-sigmoid, Linear
PS	ε	4	Vp(0), Vs(0), Vps_int, δ	Tan-sigmoid, Linear
PP & PS	δ	2	Vp(0), Vs(0), Vp_int, Vps_int	Log-sigmoid, Linear
PP & PS	δ,ε	30	Vp(0), Vs(0), Vp_int, Vps_int	Log-sigmoid, Linear
PP & PS	ε	62	Vp(0), Vs(0), Vp_int, Vps_int, δ	Linear, Linear

The number of neurons and type of transfer functions were chosen by varying the combination of transfer functions as:

1. Tan-sigmoid, Linear
2. Tan-sigmoid, Tan-sigmoid
3. Log-sigmoid, Linear
4. Log-sigmoid, Log-sigmoid
5. Linear, Linear

and the number of neurons from 1 to 150 and selecting the combination that gave the lowest root mean squared (RMS) error.

As an example for the first network listed in Table 2 the interval velocities are obtained from P-wave data and the network estimates δ. The input is a 3 x 8 matrix where $V_p(0)$, $V_s(0)$ and V_{p_int} have been established for all 8 layers of the geological model. $V_p(0)$, $V_s(0)$ are the vertical P- and S-wave velocities respectively found, theoretically, from VSP or well log data. V_{p_int} is the interval velocity found from semblance analysis. The hidden layer has 29 neurons and a tan-sigmoid transfer function. The output layer has 1 neuron and a linear transfer function. The output of the network is a 1 x 8 vector that estimates δ for all 8 layers of the geological model. An illustration of this network is displayed in FIG. 3.

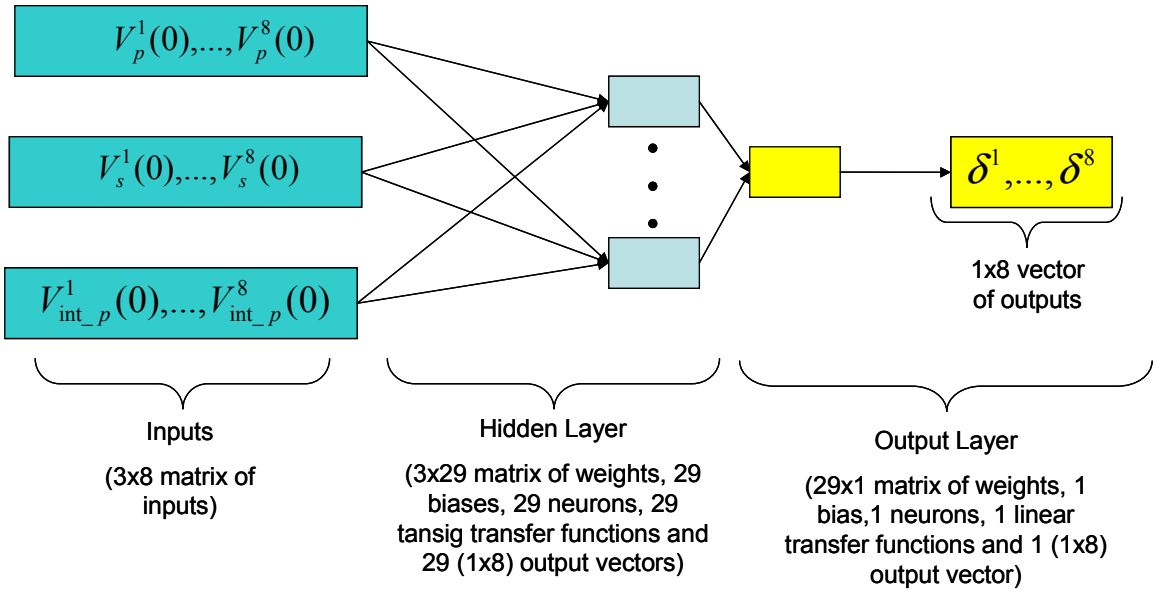

FIG. 3. Estimation of δ from P-wave data using neural networks.

Validation was done to ensure that the neural networks were running properly, data from layers 1, 3, 5 and 7 of the geological model were chosen as training data and a simulation performed all 8 layers. Results from this training and simulation are seen in FIG. 4, where the resultant δ for each geological layer is plotted. Trained results are those from training the network and simulated from applying the trained network to the data. Data used for training is a set of 'true' values and data used for simulation are values found from modeling. Discrepancies between trained and simulated values are a result of velocity picking errors or the difference between the actual and picked velocities.

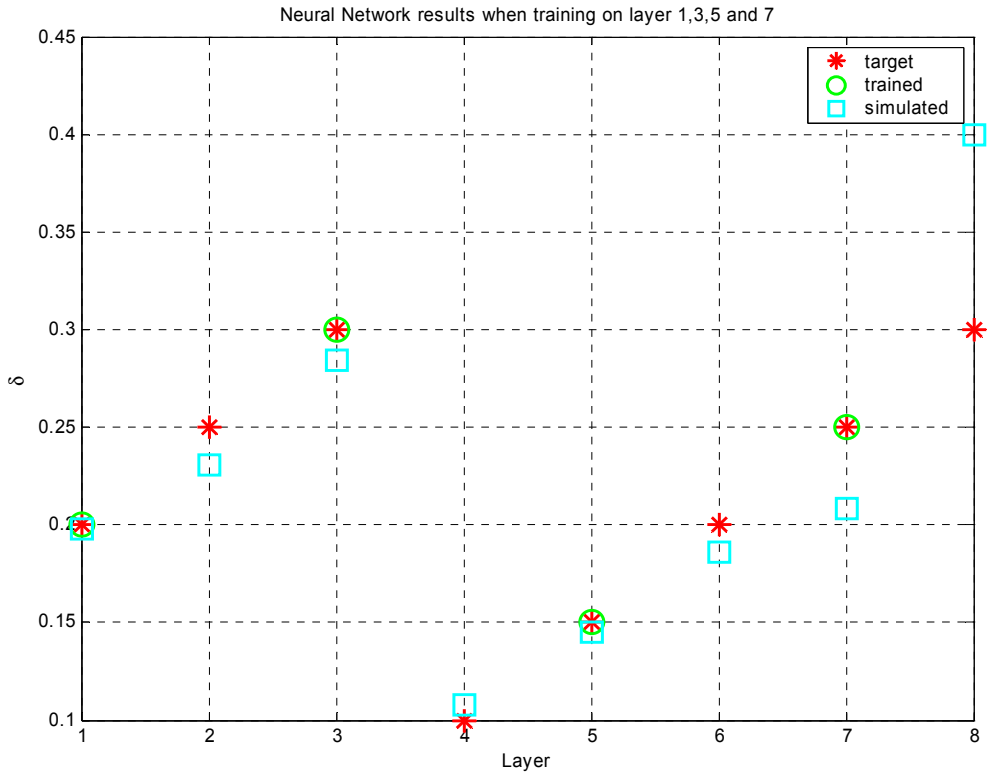

FIG. 4. A QC check when the network is trained on layers 1, 3, 5 and 7 and then incorporated to recover results for all layers.

Solving for δ from PP data

For the PP survey, δ values were found from three different methods

1. applying NMO equation *(PP NMO δ)*
2. neural network inversion estimating δ *(PP NN estimating δ)*
3. neural network inversion estimating ε and δ *(PP NN estimating δ and ε)*

Results are listed in Table 3.

Table 3. True and estimated values of δ after applying P-wave inversion methods.

Layer	Interval Velocity	True δ	PP NMO δ	PP NN Estimating δ	PP NN Estimating δ and ε
1	1180.9	0.2	0.197	0.198	0.198
2	1458.9	0.25	0.239	0.242	0.242
3	1880.1	0.3	0.286	0.288	0.290
4	2178	0.1	0.093	0.096	0.092
5	2835	0.15	0.143	0.146	0.142
6	3523.8	0.2	0.190	0.195	0.191
7	4794.8	0.25	0.218	0.237	0.233
8	6382.3	0.3	0.315	0.305	0.303

Table 4. Root mean squared errors of P-wave inversion methods used to estimate δ.

Method	PP NMO δ	PP NN Estimating δ	PP NN Estimating δ and ε
RMSE	0.0148	0.0076	0.0091

Solving for ε from PP data

The parameter ε was estimated from P-wave data in two different ways

1. neural network inversion estimating ε *(PP NN estimating ε)*
2. neural network inversion estimating ε and δ *(PP NN estimating δ and ε)*

Numerical results are listed in
Table **5**.

Table 5. Results for ε using P-wave inversion methods.

Layer	True ε	PP NN Estimating ε	PP NN Estimating ε and δ
1	0	-0.010	0.00
2	0.05	0.046	0.051
3	0.1	0.094	0.103
4	0.15	0.143	0.146
5	0.2	0.226	0.196
6	0.25	0.263	0.247
7	0.3	0.342	0.311
8	0.2	0.170	0.169

Table 6. Root mean squared errors of P-wave inversion methods used to estimate ε.

Method	PP NN Estimating ε	PP NN Estimating ε and δ
RMSE	0.021	0.012

Solving for δ from PS data

Solving for δ from converted wave data was accomplished with

1. neural network inversion estimating δ *(PS NN estimating δ)*
2. neural network inversion estimating ε and δ *(PS NN estimating δ and ε)*

Table 7. True and calculated δ values from PS-wave inversion methods.

Layer	Interval Velocity (m/s)	True δ	PS NN Estimating δ	PS NN Estimating ε and δ
1	513.9	0.2	0.205	0.197
2	629.2	0.25	0.246	0.246
3	762.2	0.3	0.304	0.321
4	1448.6	0.1	0.113	0.103
5	1788.7	0.15	0.154	0.151
6	2072	0.2	0.298	0.226
7	2798.4	0.25	0.245	0.2497
8	3068.7	0.3	0.376	0.274

Table 8. RMS errors for PS-wave inversion methods used to estimate δ.

Method	PS NN Estimating δ	PS NN Estimating ε and δ
RMSE	0.0441	0.0152

Solving for ε from PS data

Solving for ε from PS-wave data was performed using three methods:

1. applying NMO equation *(PS NMO ε)*
2. neural network inversion estimating *ε (PS NN estimating ε)*
3. neural network inversion estimating ε and δ *(PS NN estimating ε and δ)*

The PS NMO ε method uses δ estimated from the P-wave NMO equation, equation (2.51a)

Table 9. True and calculated ε values from PS-wave inversion methods.

Layer	True ε	PS NMO ε	PS NN Estimating ε	PS NN Estimating ε and δ
1	0	-0.000	-0.007	-0.186
2	0.05	0.049	0.0431	0.132
3	0.1	0.072	0.079	-0.049
4	0.15	0.138	0.114	0.153
5	0.2	0.190	0.190	0.201
6	0.25	0.214	0.189	0.276
7	0.3	0.262	0.270	0.300
8	0.2	0.243	0.212	0.303

Table 10. RMS errors PS-wave inversion methods estimating ε.

Method	PS NMO ε	PS NN Estimating ε	PS NN Estimating ε and δ
RMSE	0.0266	0.0288	0.0968

Solving for δ from joint PP and PS data

Both compressional wave and converted wave data were used in combination to investigate if better results for δ could be achieved. Two types of neural networks were invoked for this task;

1. neural network inversion estimating δ *(PP PS NN estimating δ)*
2. neural network inversion estimating ε and δ *(PP PS NN estimating ε and δ)*

Results are displayed below in
Table **11**.

Table 11. True and calculated δ values from joint PP and PS-wave inversion methods.

Layer	True δ	PP PS NN Estimating δ	PP PS NN Estimating δ and ε
1	0.2	0.208	0.190
2	0.25	0.286	0.209
3	0.3	0.213	0.333
4	0.1	0.118	0.066
5	0.15	0.149	0.152
6	0.2	0.187	0.234
7	0.25	0.250	0.250
8	0.3	0.544	0.305

Table 12. RMS error for joint PP and PS-wave inversion methods estimating δ.

Method	PP PS NN Estimating δ	PP PS NN Estimating ε and δ
RMSE	0.0929	0.0256

When both wave types are considered a neural network that simultaneously solves for both parameters gives optimal results.

Solving for ε from joint PP and PS data

Similarly, ε was determined by doing a joint inversion on compressional and converted wave data. Two types of neural networks were tested;

1. neural network inversion estimating ε *(PP PS NN estimating ε)*
2. neural network inversion estimating ε and δ *(PP PS NN estimating ε and δ)*

Results are provided in Table 13.

Table 13. True and calculated ε values from joint PP and PS-wave inversion methods.

Layer	True ε	PP PS NN Estimating ε	PP PS NN Estimating ε and δ
1	0	0.007	0.00
2	0.05	0.041	0.057
3	0.1	0.077	0.101
4	0.15	0.140	0.107
5	0.2	0.193	0.203
6	0.25	0.223	0.298
7	0.3	0.284	0.300
8	0.2	0.201	0.169

Table 14. RMS errors for joint PP and PS-wave inversion methods used to estimate ε.

Method	PP PS NN Estimating ε	PP PS NN Estimating ε and δ
RMSE	0.0148	0.0255

The best estimation of ε was found from a neural network that estimates only ε.

The best method to estimate δ was a neural network applied to P-wave data that also estimates δ. It was able to delimit δ to within 5% of the true value. Conversely the method deemed to best determine ε was a neural networks applied to P-wave data that recovered both δ and ε. It was able to delimit ε to within 16% of the true value.

FIELD DATA

Inversion methods that were found to give optimal results for the synthetic data are applied to the Blackfoot study for estimation of ε and δ. Based on these results, neural network inversion is applied to compressional wave data for the recovery of the anisotropic parameters.

Evidence of anisotropy at Blackfoot

A test for anisotropy in the Blackfoot dataset was performed on the 1999 radial component by varying the values of the effective V_p/V_s ratio. No obvious anisotropic effects were observed, which is normal when analyzing a flat data set. However the best image of the channel was found using $\gamma_{eff} = 0.9$. Thus, there is anisotropy in the area but it is weak (Lu and Margrave, 2001). Similar results were found by Cary and Lu (1999) who learned from a similar analysis that a great deal of resolution is lost when the effects of layered anisotropy are ignored. Hasse (1998) also explained non-hyperbolic moveout through anisotropy after discounting other potential explanations. Estimates of the

anisotropic parameters are compared with those obtained by Elapavuluri (2003), who used a Monte Carlo inversion.

Optimal inversion methods are applied to the Blackfoot data to recover ε and δ. Following Elapavuluri (2003) anisotropy parameters will be estimated in the formations listed in Table 16. Acronyms used to describe these formations are also listed in Table 16.

Table 15. Formation naming convention.

Acronym	Formation Name
BFS	Base of Fish Scales
MANN	Blairmore-Upper Mannville
COAL	Coal Layer
GLCTOP	Glauconitic Channel
MISS	Shunda Mississippian

Inversions results are listed in Table 16 and Table 17. For reference results from Elapavuluri (2003) are also tabulated.

Table 16. Calculated δ values from Blackfoot P-wave neural networks and Elapavuluri (2003).

Formation	δ (estimated)	δ (Elapavuluri)
BFS	0.269	0.230
MANN	-0.005	0.040
COAL	0.284	0.240
GLCTOP	0.057	0.060
MISS	-0.121	0.000

Table 17. Calculated ε values from Blackfoot P-wave neural networks and Elapavuluri (2003).

Formation	ε (estimated)	ε (Elapavuluri)
BFS	0.232	0.060
MANN	-0.020	0.008
COAL	0.192	0.120
GLCTOP	0.030	0.006
MISS	0.010	0.001

Attempts to quantify anisotropy in the Blackfoot field are limited. Comparing our results with those obtained by Elapavuluri (2003) showed a reasonable match for the estimation of δ. A similar trend is observed in the estimation of ε. The worst correlation is found at the MISS for δ and at the BFS for ε. The best correlation is found in the producing formation, GLCTOP, for both parameter estimations. Results are consistent with Thomsen (1986) where the sands show little anisotropy and the coals and shales display stronger anisotropy.

CONCLUSIONS

A simple and robust algorithm is proposed for the estimation of Thomsen's anisotropy parameters, ε and δ. A neural network that estimates δ had 29 neurons, used vertical compressional and shear wave velocities and compressional interval velocity information for training and incorporates a tan sigmoid and linear transfer function will give optimal results for the estimation of δ. Similarly, a neural network that estimates ε and δ, has 88 neurons, use vertical compressional and shear wave velocities and compressional interval velocity information for training and incorporates a tan sigmoid and linear transfer function will give optimal results for the estimation of ε.

REFERENCES

Backus, G.E., 1962, Long-wave elastic anisotropy produced by horizontal layering: J. Geophys. Res., **70**, 3429.

Cary, P. W. and Lu, H., 1999, Layered anisotropic CCP stacking of the Blackfoot 3C, 3D survey: CREWES Research Report, **11**.

Crampin, S., 1981, A review of wave motion on anisotropic and cracked elastic media: Wave Motion, **3**, 343-391.

Elapavuluri, P. K., 2003, Estimation of Thomsen's anisotropic parameters from geophysical measurements using equivalent offset gathers and the shifted-hyperbola NMO equation, MSc Thesis University of Calgary.

Freeman, J. A. and Skapura, D. M., 1991, Neural Networks Algorithms, Applications and Programming Techniques: Addison-Wesley, Massachusetts.

Hasse, A., Nov. 1998, Non-hyperbolic moveout in plains data and the anisotropy question: CSEG Recorder, 20-34.

Haykin, S., 1999, Neural networks a comprehensive foundation: Prentice-Hall Inc., New Jersey.

Higham, J. D. and Higham, N. J., 2000, MATLAB Guide 2nd edition: Soc for Industrial & Applied Math, Philadelphia, PA.

Isaac, J.H. and Lawton, L.C., 2004, A practical method for estimating effective parameters of anisotropy from reflection seismic data: Geophysics, **69**, 681-689.

Lu, H. and Margrave, G., 2001, Reprocessing the 1999 Blackfoot 3C-3D seismic data and anisotropy analysis for the radial component: CREWES Research Report, **13**, 509-517.

Shearer, P. M., 1999, Introduction to Seismology: Cambridge University Press, United Kingdom.

Russell, B., 2004, The application of multivariate statistics and neural networks to the prediction of reservoir parameters using seismic attributes, PhD Thesis University of Calgary.

Thomsen, L., 1986, Weak Elastic Anisotropy: Geophysics, **51**, 1954-1966.