PERGAMON

Computers & Geosciences 26 (2000) 869–875

COMPUTERS & GEOSCIENCES

Reservoir parameter estimation using a hybrid neural network

F. Aminzadeh[a,*], Jacob Barhen[b], C.W. Glover[b], N.B. Toomarian[c]

[a] *dGB-USA and FACT Inc., 14019 SW Freeway, Suite 301-230, Sugarland, TX 77478, USA*
[b] *Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory, USA*
[c] *Jet Propulsion Laboratory, California Institute of Technology, USA*

Received 15 March 1999; received in revised form 9 September 1999; accepted 9 September 1999

Abstract

The accuracy of an artificial neural network (ANN) algorithm is a crucial issue in the estimation of an oil field's reservoir properties from the log and seismic data. This paper demonstrates the use of the *k*-fold cross validation technique to obtain confidence bounds on an ANN's accuracy statistic from a finite sample set. In addition, we also show that an ANN's classification accuracy is dramatically improved by transforming the ANN's input feature space to a dimensionally smaller, new input space. The new input space represents a feature space that maximizes the linear separation between classes. Thus, the ANN's convergence time and accuracy are improved because the ANN must merely find nonlinear perturbations to the starting linear decision boundaries. These techniques for estimating ANN accuracy bounds and feature space transformations are demonstrated on the problem of estimating the sand thickness in an oil field reservoir based only on remotely sensed seismic data.

Keywords: Neural networks; Oil exploration; Reservoir characterization; Seismic attributes

1. Introduction

Here, we demonstrate the viability of Artificial Neural Network (ANN) algorithms in estimating oil field reservoir parameters from remotely sensed seismic data. To accomplish this one must first develop a technique to obtain an unbiased accuracy measure from a finite sample set. This measure will allow for meaningful comparisons between different ANN methods and architectures, as well as meaningful comparisons between the multitude of different data preprocessing methods. The objectives of this paper are twofold: (1) to demonstrate the use of the *k*-fold cross validation technique to obtain confidence bounds on an ANN's accuracy statistic from a finite sample set; and (2) to demonstrate that an ANN's classification accuracy is dramatically improved by transforming the data into a feature space that maximizes the linear separation between classes.

Usually, the performance of a classifier is reported as the accuracy it achieves on a test set of data. This accuracy measure by itself is meaningless. It represents a single instantiation of the random variable representing the accuracy statistic. What we seek is an estimate of the classifier's accuracy on all possible samples (the population) from an unknown probability distribution.

* Corresponding author. Fax: +1-281-265-2512.
E-mail address: Faz@dgbUSA.com (F. Aminzadeh).

0098-3004/00/$ - see front matter
PII: S0098-3004(00)00027-3

The data set (training plus test sets) represents a single, finite-sized sample set from the population. The question we need to answer is: How representative is the classifier's sample accuracy of the classifier's population accuracy? The k-fold cross validation approach is used to answer this question and is described in Section 3, and it is applied to all results reported here.

Armed with a meaningful measure of a classifier's performance, we can now investigate the effects of different ANN architectures and data preprocessing methods on the accuracy in estimating oil field reservoir parameters. It is often said that the art in obtaining accurate estimates lies in the data preprocessing techniques. In Section 4, we present a preprocessing technique that transforms the ANN's input feature space to a dimensionally smaller, new input space. The new input space represents a feature space that maximizes the linear separation between classes. We show in Section 4 that this improves the ANN's convergence time and accuracy.

We begin this paper by outlining the parameter estimation problem for oil field reservoirs, and describing the data set used here to illustrate our approaches to accuracy estimation and data transformation. Other previous publications on the use of neural network for analyzing seismic data are, Aminzadeh et al. (1994), de Groot (1999) and Wang et al. (2000).

2. Reservoir parameter estimation problem

The overall objective of our research is to demonstrate the effectiveness ANN computing techniques in providing an accurate estimation of petrophysical parameters that describe reservoir properties. Specifically, can ANNs accurately obtain functional relationships between perturbations of subsurface rock properties (e.g., sand layer thickness, effective porosity, ratio of clay to sand, and saturation) and seismic response information, for example, depth, s- and p-wave velocities, amplitude-versus-offset (AVO) and zero offset amplitude?

The oil industry acquires and processes large volumes of seismic data. In conjunction with various types of geoscience data (e.g., log, core, geochemical, gravity, magnetic, remote sensing data), attempts are made to locate prospective places for oil and gas reservoirs. This data is extensively manipulated before it is analyzed and interpreted. Every data manipulation step is important and data processing can be time consuming and expensive. It is imperative that the efficiency of the data manipulation and data reduction be improved. Thus, it is expected that ANNs can help the oil industry in two regards: (1) improved efficiency in data processing; (2) characterization and prediction of reservoir properties by training the ANN with known data.

The characterization and prediction of reservoir properties is an important application of ANNs in the oil industry. The input data to the prediction problem are usually processed and interpreted seismic and log data and/or a set of attributes derived from the original data set. Historically, many "hydrocarbon indicators" have been proposed to make such predictions. Among some of the statistical approaches are: seismic clustering analysis (Aminzadeh and Chatterjee, 1984) and fuzzy pattern recognition (McCormack, 1990). Many of the ANNs developed for this purpose are built around the earlier techniques for establishing a relationship between the raw data and physical properties of the reservoirs.

The research reported here investigates the accuracy of ANN computing techniques in estimating petrophysical parameters that describe reservoir properties based on seismic response information. This information is derived by perturbing each of the different reservoir properties. The perturbations in the data set were obtained from oil field measurements. This data set will represent a known lithology and will be used to judge the ANN's estimation accuracy. Once ANN techniques are developed for this problem and their accuracy determined, they will be applied to areas with more limited information.

To test the effectiveness of our ANNs in predicting different reservoir properties, we start with a suite of well logs (seismic velocity and density) from a well in a known oil field. The first step is to "block" the logs, to generate blocked logs that have a limited number of layers with constant layer properties, for example using the method described in de Figueiredo et al. (1996). Then we generate a suite of elastic models by perturbing the following reservoir properties in the depth interval of interest: water saturation, effective porosity, sand thickness, and sand/shale ratio. For every one of these perturbations we generate the synthetic seismic shot gather. For every gather, we compute different seismic attributes within the time window. The attributes used in this study are described below. The reason for describing the attributes in the form of "a measure of" is the fact that in most cases a particular approximation or transformation of the raw data has been used to obtain best results. The data used in this study is the same as those in Patnaik et al. (1995).

2.1. Seismic attributes

A1 a measure of average velocity from the surface to the reflector position

A2 a measure of average reflection coefficients for the "near offset traces"
A3 a measure of slope of the amplitude variations from near to far offsets
A4 a measure of zero offset amplitude
A5 a measure of nonzero offset amplitude
A6 a measure of compressional velocity of sands
A7 a measure of shear velocity of sands
A8 a measure of density of sands

With various perturbations of the reservoir properties 160 seismic records were generated. Each record represents a time window with as many as 360 sample points for each seismic attribute.

Typical reservoir parameters to be estimated are: sand water saturation; effective sand porosity; ratio of sand/clay; and sand thickness. In this paper we will demonstrate our results by predicting sand thickness, to be referred to as B. The input to the neural network will be parameters A1–A8 described above.

Instead of attempting to estimate the exact value of B we will estimate the closest value of B to one of the pre-determined ranges (e.g., 5, 10, 25, 100 ft). Thus, each range represents a class in a pattern classification problem. The objective here will be to determine the B classes (range) from the seismic attributes.

3. Accuracy measure: k-fold cross validation

We want to estimate the ANN performance on all future samples presented to it after training. This is clearly impossible unless the underlying probability distribution that the training samples were drawn from is exactly equal to the probability distribution from which the future examples are drawn. The final application's probability distribution must also be "similar" to the training probability distribution for a measure of accuracy to have any meaning. This statement is true for any regression and/or pattern recognition method.

Even if this caveat is true and the probability distributions are equal, we can only provide an estimate P of a population parameter p. We must bound our estimate with a confidence interval because the training set is finite and P is a random variable. One method to estimate the confidence interval is to use the de Moivré–Laplace theorem and the assumptions it entails.

3.1. de Moivré–Laplace theorem

This theorem states that the probability that P is within $\pm z\sigma_P$ of p with a confidence δ is given by

$$Pr[-z\sigma_P < P - p < z\sigma_P] \approx \delta$$

where σ_P^2 is the variance of P. Only in the limit as we train on all population samples does the confidence approach one. Thus, any estimate of a statistic describing the accuracy of an ANN is a random number and by itself is meaningless unless it is accompanied by a corresponding confidence interval.

The proportion of successful predictions an ANN produces on the population, p, is the population parameter we would like to estimate from a finite sample set as our measure of the ANN's accuracy. The proportion of successful predictions P the ANN produces on a test set D_{tst} after its weight parameters, $\boldsymbol{w}$, have been fixed to $\boldsymbol{w}^*$ by training is given by

$$P = \frac{1}{n} \sum_{\langle \boldsymbol{x}_i, \boldsymbol{y}_i \rangle \in D_{\text{tst}}}^{n} \varphi[\, f(D_{\text{trn}}, w^*, \boldsymbol{x}_i \in X)\boldsymbol{y}_i],$$

where φ is an indicator function that produces a value of one when a sample $\boldsymbol{x}_i$ is drawn from a test set D_{tst} and the ANN yields the correct output vector $\boldsymbol{y}_i$, or otherwise its zero. Simply put, this is the accuracy of the ANN on the test set D_{tst}.

3.2. Blind test (holdout method)

The holdout method is an unbiased first-order technique to validate a population parameter p from a finite sample set. The holdout method consists of $|$ samples drawn from a set D with an unknown probability distribution to obtain a sample set $D|$. The sample set $D|$ is divided into a training set D_{trn} and a test set D_{tst}. The bootstrap and k-fold-cross-validation techniques are the two main variations of the holdout method.

The bootstrap method randomly draws members for the training and test sets by sampling from $D|$ with replacement. The ANN is trained from D_{trn} and tested on D_{tst}. This process is repeated with different training and test sets to build an estimate of P and its variance σ_P^2. The confidence interval is determined from σ_P^2 accordingly.

The k-fold-cross-validation technique partitions the data in k mutually exclusive partitions. The first k-1 partitions are used for D_{trn} and the kth for D_{tst} where the ANN accuracy is computed. Next, partitions $\{1, 2, \ldots, k-2, k\}$ are used D_{trn} and the $k-1$ partition is used for D_{tst}; then partitions $\{1, 2, \ldots, k-3, k-1, k\}$ are used for D_{trn} and the $k-$ partition is used for D_{tst}; and so on until k accuracy measures have been obtained along with their variance.

Both of these methods are unbiased in the limit that training set consists of $|-1$ samples. The test set is the single remaining example. The accuracy estimate and variance are computed from all $|-1$ permutations of

these sets. This is the leave-one-out holdout method. The unbiased guarantee disappears if these conditions are not met. In practice, all $|-1$ permutation sets are seldom used because of the labor involved in evaluating ANN algorithms for a large number of training and test sets. Usually, most researchers divide the data into one training and one test set with an equal number of samples in each set, or at best a small number of training and test sets. Thus, one is left with a potentially highly biased ANN accuracy measure with a tight confidence bound that may not be remotely close to the population's accuracy. The researcher may therefore be led into a false sense of security about the performance of the ANN's results.

Kahavi (1995) compared the bias and variance tradeoff between the bootstrap and k-fold-cross-validation techniques as a function of the number of samples in training/test sets. He found that the bootstrap method has a smaller variance than k-fold-cross-validation, but the bias is much larger. For this reason, Kahavi concluded that k-fold-cross-validation may provide a better operational estimate of a classifier's accuracy than bootstrap. In addition, he showed for the k-fold-cross-validation technique that ten or more partitions are sufficient for the sample accuracy with a 95% confidence interval to enclose the population's accuracy. For this reason, we employed the k-fold-cross-validation technique with a $k = 10$ partition to estimate our ANN's accuracy and confidence intervals.

4. Data transformations and dimensionality reduction

Much of the success with any regression or classification algorithm comes in the analysis and preparation of the input data. Rather than blindly use all seismic attributes for inputs to the ANN, we carried out an analysis to determine those attributes that carry a significant amount of useful information for our problem. This reduces the input space's dimension. Next, we scaled all attributes to the same range of values so that no one parameter dominates the ANN training phase. These steps, discussed below, are crucial in achieving the best performance from the ANN.

4.1. Dimensionality reduction

In theory we can use all seismic attributes as an input to the ANN and it should perform satisfactorily. The problem with this, in practice, is that the training time is significantly increased without much performance improvement. The ANN spends time learning meaningless correlations amongst a large number of input variables due to idiosyncrasies in individual training cases. Reducing the number of input variables allows the ANN to focus on only significant correlations. The problem is to find those variables involved in the "significant correlations". We used standard statistical measures to determine the "significant variables".

A principal component analysis (PCA) is traditionally used to determine whether any of the variables are highly correlated and should be combined. It also indicates those combinations of variables that contain large spreads in the data on average. Also, it indicates those variables that are on average approximately constant and can be dropped as a "significant variable".

However, for our pattern classification problem, those "insignificant variables" may represent the greatest separation between the classes, whereas the "significant variables" may represent highly overlapping class distributions.

4.2. Data transformation

The ANN estimation of the B parameter was treated as a pattern classification problem because it took on only six discrete ranges of sand thickness. Each range was treated as a class. To reduce the dimensionality of the ANN's input space, we seek a nonsingular transform that preserves the class separation while it transforms the original input space $\boldsymbol{X}$ into a lower dimensional space $\boldsymbol{Y}$. (This section follows the notation of Fukunaga, 1990.)

We seek a solution to a criteria function J that rewards maximal separation among the classes. There are many criteria functions that may be chosen, the one we use is given by

$$J = \mathrm{tr}(S_{\mathrm{w}}^{-1} S_{\mathrm{b}})$$

where the within-class scatter S_{w} of samples from class ω_{t} around their expected class mean vectors $\boldsymbol{M}_i$ are expressed by

$$S_{\mathrm{w}} = \sum_{i=1}^{L} P_i E\{(\boldsymbol{X} - \boldsymbol{M}_i)(\boldsymbol{X} - \boldsymbol{M}_i)^{\mathrm{T}}\} \mid \omega_i,$$

the between-class scatter S_{b} is expressed by

$$S_{\mathrm{b}} = \sum_{i=1}^{L} P_i (\boldsymbol{M}_i - \boldsymbol{M}_0)(\boldsymbol{M}_i - \boldsymbol{M}_0)^{\mathrm{T}},$$

and the expected vector of the mixture distribution is

$$\boldsymbol{M}_0 = E\{\boldsymbol{X}\} = \sum_{i=1}^{L} P_i \boldsymbol{M}_i.$$

We are looking for a transform A from an n-dimensional space $\boldsymbol{X}$ to an m-dimensional space $\boldsymbol{Y}$ ($m < n$);

$$\boldsymbol{Y} = A^{\mathrm{T}} \boldsymbol{X}.$$

The optimization of the criteria function is given by the following three steps:

Express J in terms of the m-dimensional $\boldsymbol{Y}$-space,

$$J(m) = \mathrm{tr}(S_{\mathrm{w}Y}^{-1} S_{\mathrm{b}Y}) = \mathrm{tr}\{(A^{\mathrm{T}} S_{\mathrm{w}X} A)^{-1} (A^{\mathrm{T}} S_{\mathrm{b}X} A)\}$$

take the derivative with respect to A and set it to zero,

$$\frac{\partial J(m)}{\partial A} = -2 S_{\mathrm{w}X} A S_{\mathrm{w}Y}^{-1} S_{\mathrm{b}Y} S_{\mathrm{w}Y}^{-1} + 2 S_{\mathrm{b}X} A S_{\mathrm{w}Y}^{-1} = 0$$

and solve for the optimum nonsingular transform A,

$$(S_{\mathrm{w}X}^{-1} S_{\mathrm{b}X}) A = A (S_{\mathrm{w}Y}^{-1} S_{\mathrm{b}Y}).$$

We want to diagonalize the above equation to those eigenvalues and their associated eigenvectors that contribute the most to the separation of the classes. Diagonalizing this equation is equivalent to simultaneously diagonalizing the two matrices S_{w} and S_{b}. To see this we choose a transform A such that

$$A^{\mathrm{T}} S_{\mathrm{w}X} A = \mathrm{I} \quad \text{and} \quad A^{\mathrm{T}} S_{\mathrm{b}X} A = K$$

is satisfied. In general, K is not diagonal. The first equation above is satisfied if it is chosen to be

$$A = \Lambda_{\mathrm{A}}^{-1/2} \Phi_{\mathrm{A}}$$

where Λ_{A} and Φ_{A} are the eigenvalue and eigenvector matrices for $S_{\mathrm{w}X}$. Next, we seek to diagonalize K by solving for its eigenvalue and eigenvector matrices, Λ_K and Φ_K,

$$\Phi_K^{\mathrm{T}} K \Phi_K = \Lambda_K \quad \text{and} \quad \Phi_K^{\mathrm{T}} I \Phi_K = I.$$

Substituting in for K above yields

$$A^{\mathrm{T}} S_{\mathrm{b}X} A \Phi_K = A^{\mathrm{T}} S_{\mathrm{w}X} A \Lambda_K \Phi_K$$

or

$$S_{\mathrm{w}X}^{-1} S_{\mathrm{b}X} [\Lambda_{\mathrm{A}}^{-1/2} \Phi_{\mathrm{A}}] \Phi_K = [\Lambda_{\mathrm{A}}^{-1/2} \Phi_{\mathrm{A}}] \Lambda_K \Phi_K.$$

This transforms and projects the input space $\boldsymbol{X}$ onto the m eigenvectors of $S_{\mathrm{w}X}^{-1} S_{\mathrm{b}X}$ that optimize J. This projection is closely related to Fisher's Linear Discriminant. The linear discriminant boundaries are perpendicular to the line connecting $\boldsymbol{M}_i$ and $\boldsymbol{M}_j$ for ij in this m-dimensional space.

We will transform all the input vectors for the estimation of B according to the transformation, $[\Lambda_{\mathrm{A}}^{-1/2} \Phi_{\mathrm{A}}] \Phi_K$.

5. Results

The B parameter took on only six range values (5, 10, 25, 50, 100, 200 ft) for a very wide range of input parameter values. We treated each of the six values as members of six different classes and used the ANN to predict the B class given its input values. The adjoint analysis (Toomarian and Barhen, 1992) indicated that a fully connected, feed-forward, multi-layered ANN is an appropriate architecture for this pattern classification problem.

The ANN input space initially consisted of A1, A2, A3, A4, A5, A7 and A8 seismic parameters. A transformation to an eigenspace (as discussed in Section 4) reduced the number of inputs to five eigenvectors. The entire data set was projected into this space and linearly scaled to the [0, 1] interval.

The ANN's output consisted of six nodes, one for each class. All hidden layer nodes used the tanh activation function with a response range of [−1, 1], while the output nodes used a sigmoid activation function with a range of [0, 1].

The entire transformed and scaled data was randomly partitioned into 10 subsets. These partitions were combined according to the k-fold-cross-validation procedure into 10 training and testing pairs, such that

Training set No. 1 = partitions {1, 2, ..., 9} and test set No. 1 = partition {10}; training set No. 2 = partitions {1, 2, ..., 8, 10} and test set No. 2 = partition {9}; ... ; training set No. 10 = partitions {2, 3, ..., 9} and test set No. 10 = partition {1}.

These sets were used to obtain all accuracy estimates and their associated confidence intervals.

The number of hidden layers and the number of nodes per hidden layer in the ANN's architecture was determined experimentally. Initially, a single hidden layer was used with 10 nodes. All weights were initialized with values between ±0.1 and training set No. 1 shuffled for presentation to the ANN. The ANN's error on test set No. 1 was monitored along with the training set's error as a function of the number of epochs (1 epoch = 1 full presentation of the training set's examples). As the number of epochs increases

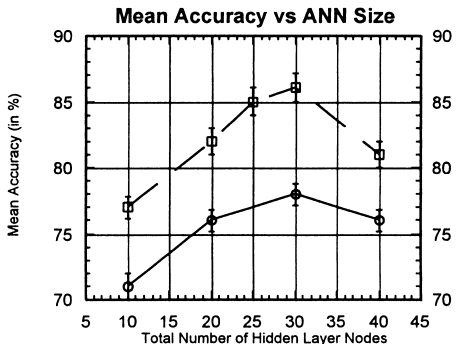

Fig. 1. Mean accuracy is displayed along with its 95% confidence intervals for ANNs with single (solid) and two (dashed) hidden layers. No transformation was applied to input data.

both the training set error and test error decrease, up to a point where the training set error continues to decrease and the test set error starts to increase. Just prior to this point we extract the error on test set No. 1 and use this value in our accuracy estimate. This process is repeated for the remaining sets. The 10 accuracy estimates are then averaged, variances extracted, and their 95% confidence intervals are computed according to the formulas outlined in Section 3. This process is repeated for each ANN of a given size.

Fig. 1 displays the mean accuracy and 95% confidence intervals for ANNs with one and two hidden layers and with a differing number of nodes in the hidden layers. The input space used to generate these points was *not* transformed to the eigenspace, but was the seven seismic parameters discussed above. The lower curve represents the mean accuracy for an ANN with 10, 20, 30, and 40 nodes in a single hidden layer. The upper curve represents the mean accuracy for an ANN with 10 (5×5), 20 (15×5), 25 (15×10), 30 (20×10), and 40 (25×15) nodes in two hidden layers.

The two layer ANN architecture provides a higher accuracy than a single hidden layer ANN.

Fig. 2 displays the mean accuracy and confidence intervals for ANN's two hidden layers and where the *input data were transformed into the eigenspace*.

These results show that the transformed data consistently yields a higher accuracy than the untransformed data and that the ANN size is smaller. In the course of this investigation, we also found that the number of iterations needed to reach convergence is reduced by about a factor of 2/3. This is due to the fact that the ANN started from an eigenspace that already linearly separates the classes and the ANN provides nonlinear perturbations to the linear decision boundaries.

Our results indicate that a two hidden layer ANN with 15 nodes in the first hidden layer and 10 nodes in the second will be approximately 94% $\pm$ 1% accurate on any future examples drawn from the probability distribution that was used to generate the training/test data. Thus, one can expect roughly similar accuracy for any B estimates generated a similar population probability distribution as those used in this study.

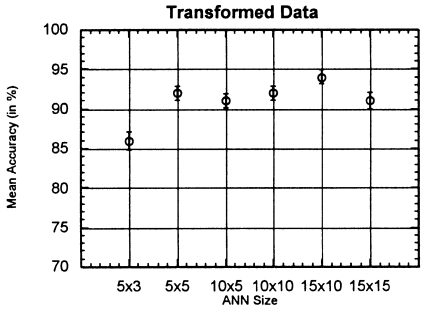

Fig. 2. Mean accuracy is displayed along with its 95% confidence intervals for two hidden layer ANNs. Number of nodes in first and second layers are given along abscissa. Data were transformed according to discussion in Section 4.2.

6. Conclusions

Our approach focuses on quantitative measures to determine the accuracy of ANNs in obtaining functional relationships between reservoir properties and seismic response data. A simulation program provided precise and unambiguous data for this study. The seismic response data which is used as input to the ANN is transformed so that different features do not dominate and bias the ANN's results. The k-fold-cross-validation method is used to estimate the accuracy and its confidence interval for all ANNs used in this study. The number of free parameters (size) in the ANN are determined by confidence interval measures.

We also show that an ANN's classification accuracy is dramatically improved by transforming the ANN's input feature space to a dimensionally smaller, new input space. The new input space represents a feature space that maximizes the linear separation between classes. Thus, the ANN's convergence time and accuracy are improved because the ANN must merely find nonlinear perturbations to the starting linear decision boundaries. These techniques for estimating ANN accuracy bounds and feature space transformations are demonstrated on the problem of estimating the sand thickness in an oil field reservoir based only on remotely sensed seismic data.

References

Aminzadeh, F., Chatterjee, S.L., 1984. Application of clustering in exploration seismology. Geoexploration 23, 147–159.

Aminzadeh, F., Katz, S., Aki, K., 1994. Transactions IEEE Geoscience and Remote Sensing 32 (6), 1139–1143.

de Figueiredo, R.J.P., Aminzadeh, F., Maccato, A., 1996. A visualization tool for well log blocking. Seismic Exploration 5 (1), 79–92.

de Groot, P.F.M., 1999. Volume transformation by way of neural network. Proceedings of European Association of Geoscientists & Engineers meeting. Helsinky, Finland. Chapter 3–37.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, San Diego, 591 pp.

Kahavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, p. 1137.

McCormack, M.D., 1990. Neural computing in geophysics.

Geophysics: The Leading Edge of Exploration 10 (1), 11–15.

Patnaik, G., Aminzadeh, F., Wrolstad, K.H., 1995. Prediction of lithology and saturation using artificial neural networks. In: Proceedings of IUGG/SEG symposium on AI in Geophysics, 12 July, Denver.

Toomarian, N.B., Barhen, J., 1992. Learning a trajectory using adjoint functions and teacher forcing. Neural Networks 5, 473–484.

Wang, P., Tamhane, D., Aminzadeh, F., 2000. A soft computing Approach to Integrate Well logs and geological clusters for Petrophysical Prediction. Proceedings 3rd International Conference on Petroleum Geophysics. New Delhi, India.