CONF-970348--1

Computer Science and Mathematics Division

Oil Reservoir Properties Estimation Using Neural Networks

N. B. Toomarian
Center for Integrated Microsystems
Jet Propulsion Laboratory
California Institute of Technology

J. Barhen
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory

C. W. Glover
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory

F. Aminzadeh
UNOCAL Corporation
Sugarland, TX

Submitted to: The Third International Conference on Neural Networks and Their Applications, Marseilles, France

Research sponsored by the Engineering Research Program, Office of Basic Energy Sciences, U. S. Department of Energy, under Contract No DE-AC05-96OR22464 with Lockheed Martin Energy Research Corp.

DISCLAIMER

Oil Reservoir Properties Estimation Using Neural Networks

N.B. Toomarian[1], J. Barhen[2], C. W. Glover[2], and F.Aminzadeh[3]

This paper investigates the applicability as well as the accuracy of artificial neural networks for estimating specific parameters that describe reservoir properties based on seismic data. Our approach relies on JPL's adjoint operators general purpose neural network code to determine the best suited architecture. We believe that results presented in this work demonstrate that artificial neural networks produce surprisingly accurate estimates of the reservoir parameters.

1. INTRODUCTION

The oil industry acquires and must process large volumes of geoscience data of various type 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 processing time can be extensive. Hence, it is imperative that the efficiency of the data manipulation and data reduction be improved. This is where Artificial Neural Networks (ANNs) may help the oil industry.

The characterization and prediction of reservoir properties is one major area where ANNs may offer improvement. Seismic data is used to characterize large volumes of the Earth's upper crust. However, this data provides only gross structure information on the size and orientation of the various underlying strata. On the other hand, well log data provide a detailed characterization of the strata but only in a very localized region, and it is far more expensive to obtain than the seismic data. To find a reservoir and characterize it, one must associate the detailed well log data with the seismic data. This will allow one to accurately extrapolate detailed information over a large volume. The strength of ANN techniques lies in their ability to capture and approximate nonlinear mappings between two data sets; in this case the relationship between well log and seismic data. Thus, the use of ANN techniques may improve the efficiency and accuracy in the manipulation of geoscience data. The research results reported here address the applicability as well as the accuracy of ANNs in estimating specific parameters that describe reservoir properties based on seismic response information.

To test the effectiveness of ANNs in predicting different reservoir properties, we start with a suite of well logs from a well in a known oil field. The first step is to generate blocked logs that have a limited number of layers with constant layer properties[1]. 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 everyone of these perturbations, we generate a synthetic seismic shot gather. For more details on different prestack attributes used in the oil industry see Ref. [2]. For every gather, we compute different seismic attributes within the time window of interest. The attributes used in this study are:

Reservoir Parameters

SWE	Sand Water Saturation
EFFPOR	Sand Effective Porosity
THICKNESS	Sand Thickness
RATIO	Clay/Sand Ratio within sands

Seismic Attributes

DEPTH / TIME	depth / Two-way Seismic Travel Time
GREF-0	Normalized, Rectified Near Offset Amplitude (AVO)
SLOPE-0	AVO Least Squares Slope of GREF-0
GREF-1	Intercept of Fit to Normalized, Rectified Amplitude Fit
SLOPE-1	Fluid Factor Attribute
GREF-2	(Ro-Rwet) of Unnormalized Intercept
SLOPE-2	(G-Gwet) of Unnormalized Signed Amplitude Fit
GREF-3	Ro Intercept of Unnormalized Signed Amplitude Fit
SLOPE-3	G or Slope of Unnormalized Signed Amplitude Fit

[1] Center for Integrated Microsystems, Jet Propulsion Laboratory, California Institute of Technology
[2] Center for Engineering Systems Advanced Research, Oak Ridge National Laboratory
[3] UNOCAL Corporation, Sugarland, TX

SLOPE-4	(Ro + G)/2 of GREF-3 and SLOPE-3
SLOPE-5	Another combination of the above
SLOPE-6	Ro*G
SLOPE-7	(Ro + G)/2 of Normalized, Rectified Amplitude Fit
VINT	p-wave Velocity of Sands
SVINT	s-wave Velocity of Sands
RHOB	Density of Sands

With various perturbations of the reservoir properties, 152 seismic records were generated. Each record consists of as many as 338 sample points at which the above mentioned attributes were determined. Our immediate goal is to predict, using ANNs, selected reservoir parameters from some or all the 17 seismic attributes. Following validation of the ANN approach, the longer term objective is to apply these techniques to large-scale seismic datasets.

In this paper, presents the results of applying ANNs to two of four main reservoir parameters, i.e., Sand Water Saturation and Sand Effective Porosity. Our methodology for preprocessing the input data, including dimensionality reduction, and data transformation for maximum feature-class separability are described in a companion paper[6]. There, we also report results on predicting another key reservoir parameter, i.e. the sand thickness.

2. APPROACH

At the outset it was decided to use a separate ANN to predict each main reservoir parameter, rather than use a single ANN to predict all parameters simultaneously. This approach offers several advantages. First, it allows us treat the prediction of a each parameter of interest as a separate problem, independent from the others. Thus, we can tailor our approach specifically to a given parameter. For example, in our reservoir analysis, we noticed that the *EFFPOR* and *RATIO* variables take on a continuos range of values. This suggested that the ANN computational model should be a nonlinear mapping algorithm. On the other hand, the *SWE* and *THICKNESS* variables take on discrete values. Hence, each value can be treated as a class. Thus, the appropriate ANN model would here be a classification algorithm. Second, using a different ANN for each parameter allows us to determine and use only those input variables that significantly affect that parameter's prediction. Finally, separate ANNs can always be combined, if necessary, in a hierarchical manner to predict other reservoir parameters.

2.1 Input Data Scaling

Scaling ANN input data equalizes the importance of all input variables and speeds up learning. Raw data can differ by several orders of magnitude. For example, one input attribute may have values on the order of 0.00001 while another is around 100,000. The larger number will quickly saturate the ANN processing elements (nodes). Saturated nodes produce constant output values corresponding to the maximum of their dynamic range. The ANN weight update rule is proportional to the derivative of the node's activation function at the current output value. For a saturated node the derivative is almost zero and its weight stops learning. Thus, in order to alleviate this problem, the data should be scaled to match the node's dynamic range.

We used a linear mapping to scale the range of each input variable to the nodes' dynamic ranges. The input variables' ranges were determined from the full data set. A linear activation function was used for the input nodes; their dynamic ranges were chosen to be [0, 1] or [-1, 1], depending on the particular ANN. Thus, all data were mapped to the appropriate interval.

2.2 Adjoint Operator Based Code

The adjoint operator method[3-4] is used to study the appropriate ANN architecture for this application. This method is implemented in a computer code that allows the user to chose and compare different ANN architectures for the problem at hand. Specifically, our method views an ANN as a set of coupled nonlinear differential equations:

$$\dot{u}_n + u_n = f_n(\sum W_{nm} T_{nm} u_m + I_n) \qquad (1)$$

where u_n represents the output of the n^{th} neuron; $\dot{u}_n$ is the derivative of u_n with respect to time; and T_{nm} is the strength of the synaptic coupling from the m^{th} to the n^{th} neuron. The sigmoidal functions f_n modulate the neural responses; typically, $f(x) = \tanh(x)$. The W_{nm} is the connectivity matrix consisting of binary numbers that indicate whether neuron n receives input from neuron m. When W_{nm} is 1 then neuron n is connected to neuron m. The W_{nm} matrix is an input to the code, hence, the analyst can decide which neurons should be connected. Such an approach allows one easily to select, test, and compare different neural architectures by changing the values of this matrix. I_n is an external input to the n^{th} neuron. The goal in this code is to minimize the difference between the desired and actual output from the network for each training sample, i.e., the error function is:

$$E = \frac{1}{2KN} \sum_k \sum_{n \in output} (d_n - u_n)^2 \qquad (2)$$

where d_n is the desired output for n^{th} neuron, and the summation is performed for the output neurons only, for all K samples. The primary bottleneck in

gradient based iterative approaches for updating the values of the matrix T arises from the cost of evaluating the derivatives $\partial E / \partial T_{nm}$. This we overcome by using adjoint operator techniques. Mathematical details are given in [3-4], and we briefly summarize the essential features in the sequel. Ultimately, the learning rule has the familiar form:

$$T_{nm}^{\tau+1} = T_{nm}^{\tau} - \Delta\eta \frac{\partial E}{\partial T_{nm}} \qquad (3)$$

Here, the superscript τ indicates the iteration number, Δ is a constant step size, and η is the learning rate. In the current code we have the option of keeping this learning rate constant or changing its value adaptively based upon the gradient. In the latter case, if the gradient has a high (low) value, then the learning rate is will be small (large). This is analogous to walking down a hill. If the hill is very steep, one takes small steps. On the other hand, if the slope is moderate, one can run by taking big steps. Specifically, we evaluate η by the following equation:

$$\eta = \left(\frac{\partial E}{\partial T} \right)^{-1/3} \qquad (4)$$

This induces a Terminal Attractor[5] effect into the dynamics of Eq.(3), where the "terminal" (non-Lipschitzian) properties arise from the value of the exponent selected, e.g., here -1/3.

In order to calculate the value of $\partial E / \partial T$ for Eq. (3), in principle one needs to calculate ($\partial E / \partial u$). The latter can be calculated by differentiating the neural dynamics in Eq. (1). This would lead us to a system of coupled linear equations, also known as forward sensitivity equations, the right hand sides of which depend on T_{nm}. These equations have to be solved as many times as there are parameters in the system, i.e., N^2 where in N is number of neurons in the system. An alternative approach exists, based upon the concept of Adjoint Operators, that calculates the value of $\partial E / \partial T$ indirectly, without calculating explicitly the value of $\partial E / \partial u$. This approach reduces the computational cost, and hence, the training time in a fully connected network by $O(N^2)$ [3-4].

This system allowed us to test various neural architectures for estimating the different reservoir parameters. Initially, several preliminary test cases were executed for each parameter. The tests were conducted on similar data (input and output set) with different architectures constructed using the connectivity matrix and the learning rate options. Based upon the training time needed compared to the accuracy of the results obtained for different ANN architectures, we concluded that a multilayer feedforward architecture with one or possibly two hidden layers was sufficient for the application at hand.

3. RESULTS

One of the challenges of this application has been the selection of the input attributes out of the set of 17 presented above. We believe that there are three different approaches to the problem.

a) Use the brute force approach, i.e., use all the attributes as input, and let the network figure out which ones are more important. This is the approach we selected for water saturation parameter estimation.
b) Use engineering judgment for choosing the important attributes and use them as input to the network. Our effective porosity estimation is based upon this approach.
c) Perform some preprocessing (similar to principal component analysis) on the data to find out which of the attributes are dominant. This is the approach we employed for the sand thickness estimation, the results of which are presented elsewhere[6].

3.1 Estimating Effective Porosity

The effective porosity of a reservoir was estimated from seismic data. As stated above, the most cost-effective architecture for this application was determined to be a feed forward three layer network operated in mapping mode. The input layer consisted of 4 neurons. The following 4 attributes were used as input: Depth, 0-Offset, AVO-Slope, and P-Velocity.

The hidden layer involved 10 neurons; the output layer had only one neuron. To train and test the network, we processed 2200 seismic samples. Approximately 200 data points were chosen randomly from this dataset for training the network. Tests with the trained network were conducted on all 2200 samples.

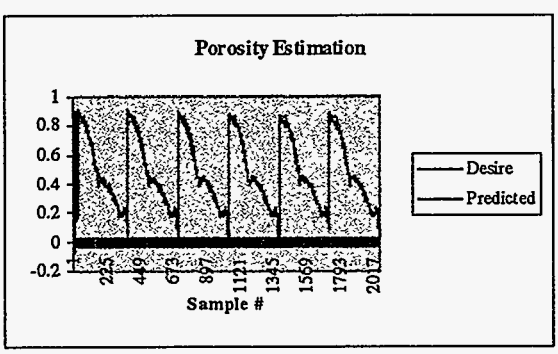

Figure 1. Desired and network predicted values of effective porosity.

The above figure shows the actual porosity values, versus those predicted by the network. The agreement is outstanding.

3.2 Water Saturation Estimation

For each trace, the value of the water saturation is constant. Hence, the problem of estimating the water saturation becomes a classification problem. Since the input attributes vary dramatically over time, to classify the saturation level based upon raw attributes becomes a hard, if not impossible problem. This suggested that it would be advantageous to extract features representative of each trace. To this end, we integrated all the attributes over time for each trace. The advantages of such an approach are rather remarkable. First, the amount of data to be manipulated is reduced dramatically (from over 50,000 to about 150), i.e., one sample per trace. Furthermore, one then needs only a small network to do the classification, which in turn results in a fast training time.

To train a network to classify water saturation, we selected a three layer feed forward network. The input layer consisted of 12 inputs, i.e., Depth, AVO-Offset, AVO-Slope, GREF(3), Slope(3), Slope(4), Slope(7), P&S Velocities, and Density. The hidden layer had 6 neurons, and only one output neuron was used. The output values were allowed to range between -1 and 1. However, for classification purposes a threshold was introduced at zero, i.e., if the network output is less than zero the saturation level was estimated to be at approximately 40%, while an output greater than zero indicated a saturation level of 100%.

The training data consisted of 26 samples that were randomly selected from the pool of 152 samples. All data values were normalized to be between -1 and 1. The network was tested on all 152 samples. Figure 2 shows the network output vs. the actual saturation value. Instances where the network misclassified the saturation level are indicated in figure 3. As can be observed from the figure, only 6 out of 152 levels were misclassified, i.e., a success rate of over 96% was achieved. This result demonstrates a major advance over estimates achieved to date with conventional techniques.

4. CONCLUSIONS

Our research at this stage appears to indicate that ANNs can efficiently be used to capture, based solely on sensed seismic data, the same essential reservoir information as a detailed forward computational model. Oil Industry representatives, part of an ongoing DOE Advanced Computing Technology Initiative project have reviewed our preliminary ANN results. It is their feeling that the proposed approach will help demonstrate that ANNs produce surprisingly accurate estimates of key reservoir parameters. They deemed this aspect of the experiment a success. However, the real value of the ANN paradigm will be measured by their ability to handle the very large scale datasets that characterize field seismic exploration. We plan to conduct such tests in the forthcoming year.

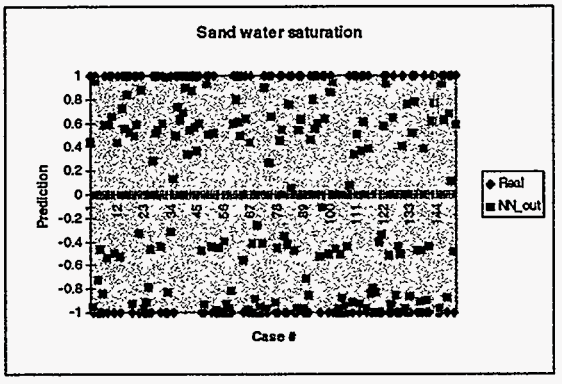

Figure 2. Desired and network predicted values of sand water saturation.

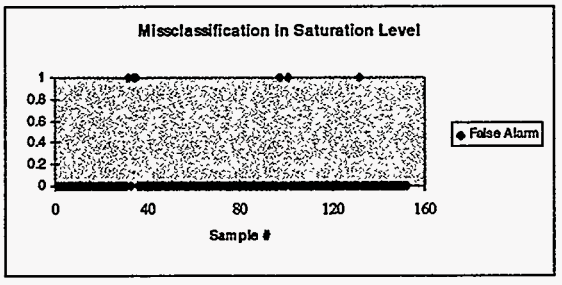

Figure 3. Six instances that the network misclassified the sand water saturation level.

5. REFERENCES

1. Castagna, J. P., Backus, M. M., 1995, AVO, An Overview, ***The Leading Edge of Geophysics***.
2. deFigueiredo, R. , Aminzadeh, F. and Maccato, A., 1996, "A Visualization Tool for Well Log Blocking", ***Seismic Exploration***, *5*(1), pp 79-92.
3. Toomarian, N. B. and Barhen, J., 1992, "Learning a Trajectory Using Adjoint Functions and Teacher Forcing", ***Neural Networks***, *5*(3), pp. 473-484.
4. Barhen, J., Toomarian, N., and Gulati, S., 1990, "Application of Adjoint Operators to Neural Learning", ***Appl. Math Lett***. *3*(3), pp 13-18.
5. Zak, M. 1992, "Terminal Attractors in Neural Networks", ***Neural Networks***, *2*(4), pp. 259-274.
6. Glover, C.W., F. Aminzadeh, J. Barhen and N.B. Toomarian , 1996, "Neural Network Accuracy Measures and Data Transforms Applied to the Seismic Parameter Estimation Problem", NEURAP'97 (submitted).