An Improved Fuzzy Neural Network for Permeability Estimation from Wireline Logs in a Petroleum Reservoir

*Y. Huang, *P.M. Wong and **T.D. Gedeon
*Centre for Petroleum Engineering
**School of Computer Science and Engineering
The University of New South Wales
Sydney NSW 2052, Australia

ABSTRACT: *Reservoir permeability estimation from wireline logs is the most difficult task for petrophysicists. Many studies have shown that the backpropagation neural network (BPNN) is the most promising tool to date, because of its ability to learn and generalise. This paper presents an improved fuzzy neural network (FNN) to solve the same problem. In the example presented, this model is stable with fast convergence and gives smaller error compared to BPNN and previous FNN methods.*

1. INTRODUCTION

1.1 Petroleum reservoir and permeability

A petroleum reservoir is a volume of porous sedimentary rock which has been filled with hydrocarbon, such as oil and gas. Reservoir permeability is one of most important petrophysical properties, and is widely used to determine the well or field production rate of such hydrocarbon.

Permeability is a measure of the mobility of fluid flow through the porous media when subjected to applied pressure gradients. The determination of such properties is complicated because the measurement sites available in the reservoir are limited to isolated well locations. At these locations, measurements take the form of actual rock samples (cores) and wireline log readings.

1.2 Data collection

Rock samples are obtained by using a coring barrel to recover intact cylindrical samples of reservoir rock. These samples are then sent to the laboratory and different petrophysical properties (such as permeability) are measured. Wireline log readings are obtained every 150 mm or so of depth, by lowering various sondes in the drilled wells. These measure formation and fluid properties in and around the wellbore location. Typical sondes generate electrical signals from measurements of radioactive, resistivity, acoustic, and neutron attenuation and scattering properties of the formation and its contained fluids. Because coring is a relatively time consuming and expensive process, much effort is made to relate other measures to the available core permeability measurements so that the transformations developed can be applied to predict permeability data in uncored wells or intervals.

1.3 Correlation methods

Correlation of wireline log readings and core permeability measurements has been widely studied. One of the most commonly used method to date is backpropagation neural networks [3], [7], [8], [13], [14]. This method uses the wireline log readings as inputs, and connection weights are determined by minimising the difference between calculated permeability and target permeability from core measurements.

While backpropagation neural networks (BPNN) have been widely used to model data correlation of various kinds, optimisation of network topology and development of fast training algorithms are still being studied in order to compensate for the deficiencies of such models. During the last few years of research, fuzzy logic has shown great improvements over previous neural network models, such as fuzzy neural networks (FNN) which incorporate fuzziness (cognitive uncertainty) into the neural network framework [1], [2], [4], [5], [6], [9], [10], [12]. The use of FNN in permeability estimation has not been previously studied.

1.4 Objective of present study

The objective of this paper is to compare the performance of BPNN and FNN models in permeability estimation from wireline logs in an oil field. An improved FNN model (IFNN), based on a gradient descent fuzzy algorithm, is also developed. Its performance will be evaluated and compared with BPNN and FNN on the same data set. In the case demonstration, part of the core permeability data set is used in developing the models, and these models are then applied to the rest of the data set in which comparison can be made.

We will first review a FNN model proposed by Lin and Cunningham [6]. We will then introduce our improved FNN (IFNN), followed by results and discussions of a case study.

2. FUZZY NEURAL NETWORKS

Figure 1 shows a schematic diagram of the fuzzy neural network (FNN) developed by Lin and Cunningham [6]. We will use their model to estimate permeability. We have four layers (input, fuzzification, hidden and output. In this diagram, we

0-7803-3679-8/96/$5.00 © 1996 IEEE

have two inputs, three fuzzy rules, and one output neuron. The number of hidden neurons is the same as the number of fuzzy rules. There are 2x3 neurons in the fuzzification layer. Every neuron in this layer represents a fuzzy membership function for each of the input variables. The activation function used in the fuzzification layer is:-

$$A_{ij}(x_j) = \exp(-\left|w_{ij1}x_j + w_{ij0}\right|^l) \qquad (1)$$

where A_{ij} is the value of radial basis (fuzzy membership) function of the j^{th} input variable corresponding to the i^{th} rule. The variable l is in the range $0.5 \le l \le 5$, and w_{ij} is the connection weight. The letter "F" in the diagram represents this operation.

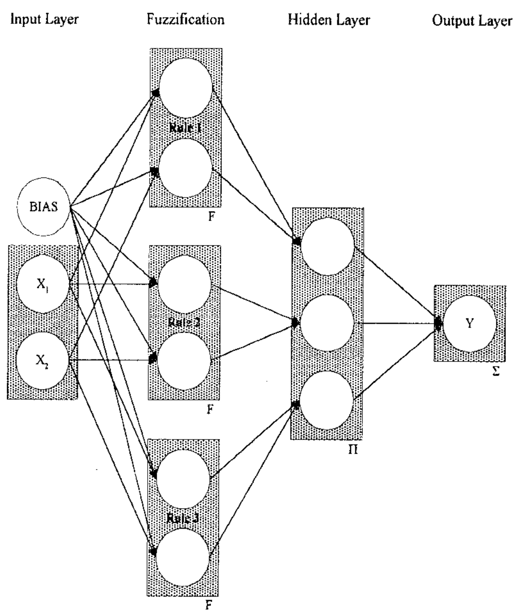


Fig.1. A schematic diagram of a FNN model.

The activation function used in the hidden layer is:-

$$y_i(x_1, x_2, \dots, x_m) = \prod_{j=1}^{m} A_{ij}(x_j) \qquad (2)$$

where m is the number of input variables. The letter "Π" denotes this operation in the figure.

The output layer is labeled "Σ", and performs:-

$$Y = \sum_{i=1}^{n} y_i c_i \qquad (3)$$

where n is the number of fuzzy rules, c_i is the connection weight. Hence, the fuzzy rule is of the form:-

IF x_1 is A_{i1} and x_2 is A_{i2} and ... and x_m is A_{im}
THEN y is c_i

Lin and Cunningham [6] used a fuzzy curve to estimate the number of rules, and the initial weights $\{w, c\}$ were determined according to the ranges of the input and output. The network was trained using the backpropagation algorithm. The authors also claimed that FNN could result in rapid convergence compared to BPNN, but it is sometimes unstable and poor in precision.

3. IMPROVED FUZZY NEURAL NETWORKS

The improved fuzzy neural network (IFNN) is based on the FNN model presented previously, and the fuzzy model suggested by Takagi and Sugeno [11]. The model developed by Takagi and Sugeno [11] is a non-linear model which can be represented by the following rule:-

Rule: IF x_1 is A_{i1} and x_2 is A_{i2} and ... and x_m is A_{im}
THEN:-

$$c_i = b_{i0} + \sum_{j=1}^{m} b_{ij} x_j \qquad (4)$$

and:-

$$Y = \frac{\sum_{i=1}^{n} y_i c_i}{\sum_{i=1}^{n} y_i} \qquad (5)$$

where b_{ij} is the connection weight (see Figure 2).

There are three major differences between IFNN and FNN:-

1) Equation (4) expresses a highly non-linear functional relation using a small number of rules. IFNN iteratively calculates b_{ij}, instead of c_i in FNN, for the determination of Y.
2) Equation (5) approximates any continuous function on a compact set [15]. Y in equation (3) does not have the denominator.
3) y in equation (2) uses the fuzzy MIN (or product) "∧" instead of the "Π" operation.

The iterative equations for updating weights $\{w, b\}$ are presented in the appendix.

4. CASE STUDY

4.1 Objective

The objective of this study is to compare three different methods in predicting permeability from wireline logs: BPNN, FNN and IFNN. The results are analysed and plotted in graphical form.

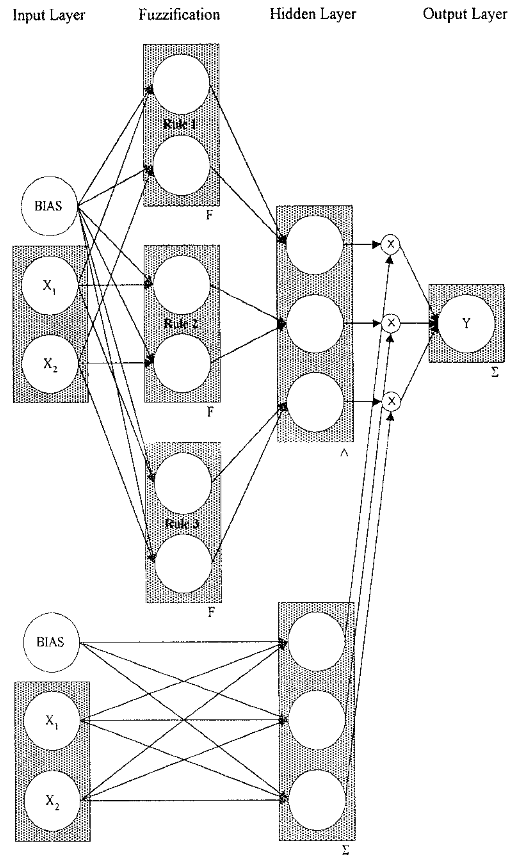


Fig. 2. A schematic diagram of an IFNN model.

4.2. Training and test data

An oil field was used to provide wireline logs and core data. The data set consisted of 230 points with six input logs, namely gamma ray (GR), microspherically-focussed log (MSFL), deep resistivity (LLD), sonic travel times (DT), bulk density (RHOB) and neutron porosity (NPHI). Each of these data points had a core permeability measurement (k) obtained from laboratory analysis. This whole data set was randomly separated into two smaller sets. The first one consisted of 150 patterns for training, and the other one had 80 patterns for testing. The test data set was used to validate the trained network after every iteration. The best model was obtained based on the minimum root mean-square-error (RMSE) on the test set.

Each of the input logs was normalised in the range of (0,1). This is normally done in neural computation as the network will then give comparable magnitudes of weight values. The logarithm of MSFL, LLD and k values were used because the transformed variables are usually normal distributed, and this transformation works better in most prediction models. All the logarithm of k values were normalised in the range of (0.1,0.9) for the BPNN. In order to keep consistency for this study, the same normalised input and output values were used for all models.

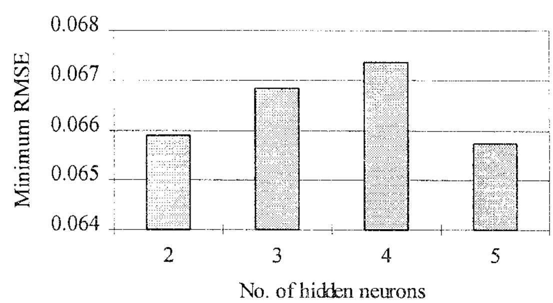
5. RESULTS

In BPNN, the training data set for permeability prediction was composed of 150 patterns with six input logs. Two to five hidden neurons were used to train the model. Five hidden neurons were found to produce the lowest minimum error on the test set. (see Figure 3a). The maximum training epochs was 10,000. The minimum RMSE was 0.0657 on the test set at the maximum epoch. Further training gave give slightly smaller RMSE, but the improvement was insignificant.

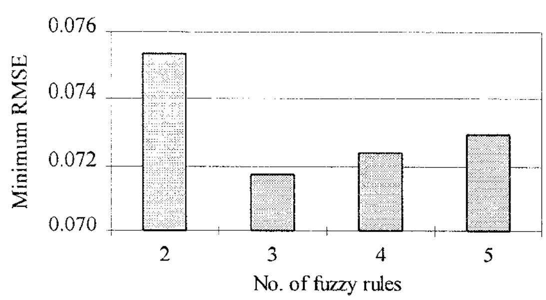
For FNN, two to five fuzzy rules were tried, and the optimal number was three (see Figure 3b). The minimum RMSE on the test set was 0.0717 at 424 training epochs.

For IFNN, two to five fuzzy rules were tried, and the optimal number was three (see Figure 3c). The minimum RMSE on the test set was 0.0655 at 315 training epochs.

a). No. of hidden neurons for BPNN.



b). No. of fuzzy rules for FNN.



c). No. of fuzzy rules for IFNN.

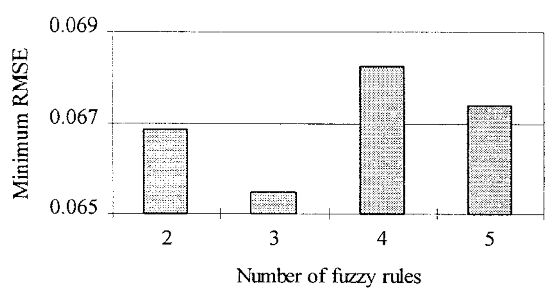


Fig. 3. Sensitivity analysis on model parameters.

6. DISCUSSION

The summary of the results are plotted in Figures 4 and 5. Figure 4 shows the learning profiles of the different models for the test set. Only the models with best performance are shown. The horizontal axis is a logarithmic scale in order to highlight the early behavior of the models. The vertical axis is the RMSE on the test set. IFNN shows the fastest convergence, followed by FNN and BPNN.

Figure 5 shows a cross-plot of training epochs versus minimum RMSE for the best model configurations. Ideally, a good technique should be in the bottom left hand region which means the model requires small training epochs and gives small error. In this figure, BPNN is on the top left hand region, which represents high training epochs and low error, FNN is on the bottom right hand region, which represents low training epochs and high error, and IFNN is on the bottom left hand region, which represents low training epochs and low error. Although the minimum RMSE from BPNN and IFNN were similar, IFNN has an obvious advantage of being faster to converge.

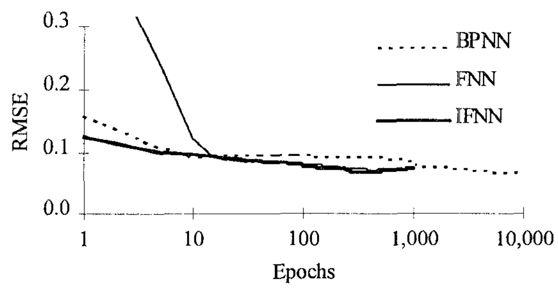


Fig. 4. BPNN, FNN and IFNN learning profiles for the 80 test patterns.

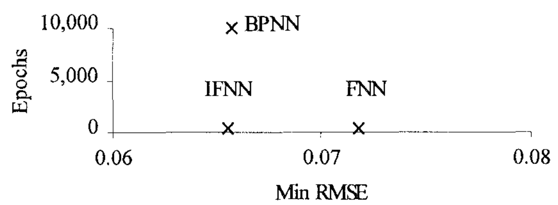


Fig. 5. No. of epochs versus minimum RMSE crossplot.

7. CONCLUSIONS

A technique for permeability prediction from wireline logs is presented using a fuzzy model. The technique is applied to a field data set and is also compared with backpropagation neural networks and fuzzy neural networks. Based on the results obtained from this study, the major findings are:-

1) The improved fuzzy neural networks can be used to estimate permeability from wireline logs.
2) The improved fuzzy neural networks gives smaller error on the test set compared to backpropagation neural networks and the previous fuzzy neural networks. It has the property of rapid convergence.
3) Further work on improving the iterative algorithm for updating weights is currently being undertaken.

8. REFERENCES

[1] V. Gorrini, T. Salome, and H. Bersini, "Self-structuring fuzzy system for function approximation," *IEEE International Conference on Fuzzy Systems*, 1995, pp. 919-926.

[2] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with back-propagation algorithm," *IEEE Trans. on Neural Networks*, vol. 3, no. 5, 1992, pp. 801-814.

[3] Z. Huang, J. Shimeld, M. Williamson, and J. Katsube, "Permeability prediction with artificial neural network modeling in the Venture Gas Field, offshore Eastern Canada," *Geophysics*, vol. 61, no. 2, 1996, pp. 422-436.

[4] J.R. Jang, "Self-learning fuzzy controllers based on temporal back-propagation," *IEEE Trans. on Neural Networks*, vol. 3, no. 5, 1995, pp. 714-723.

[5] R. Langari and L. Wang, "Fuzzy models, modular networks, and hybrid learning," *IEEE International Conference on Fuzzy Systems*, 1995, pp. 1291-1298.

[6] Y.H. Lin and G.A. Cunningham III, "A new approach to fuzzy-neural system modeling," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 2, 1995, pp. 190-197.

[7] H.A. Malki, J.L. Baldwin, and M.A. Kwari, "Estimating permeability by the use of neural networks in thinly bedded shaly gas sands," *SPE Computer Applications*, April 1996, pp. 58-62.

[8] S. Mohaghegh and S. Ameri, "Virtual measurement of heterogeneous formation permeability using geophysical well log responses," *The Log Analyst*, March-April 1996, pp. 32-39.

[9] M.K. Park, S.H. Ji, M.J. Kim, and M. Park, "A new identification method for a fuzzy model". *IEEE International Conference on Fuzzy Systems*, 1995, pp. 2159-2164.

[10] M. Ryoke, Y. Nakamori, and K. Suzuki, "Adaptive fuzzy clustering and fuzzy prediction models," *IEEE International Conference on Fuzzy Systems*, 1995, pp. 2215-2220.

[11] T. Takagi and M. Sugeno, "Fuzzy identification of system and its applications to modeling and control," *IEEE Trans. on System, Man and Cybernetics*, vol. 15, no. 1, 1985, pp. 116-132.

[12] L. Wang and R. Langari, "Building Sugeno-type models using fuzzy discretization and orthogonal parameter estimation techniques," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 4, 1995, pp. 454-458.

[13] P.M. Wong, F.X. Jian, and I.J. Taggart, "A critical comparison of neural networks and discriminant analysis in lithofacies, porosity and permeability predictions," *Journal of Petroleum Geology*, vol. 18, no. 2, 1995, pp. 191-206.

[14] P.M. Wong, I.J. Taggart, and T.D. Gedeon, "Use of neural network methods to predict porosity and permeability of a petroleum reservoir," *AI Applications*, vol. 9, no. 2, 1995 pp. 27-38.

[15] X.J. Zeng and M.G. Singh, "Approximation accuracy analysis of fuzzy system as function approximators," *IEEE Trans. on Fuzzy Systems*, vol. 4, no. 1, 1996, pp. 44-63.

APPENDIX

The iterative equations are developed based on the least mean square (LMS) technique and the gradient descent algorithm. As shown in equation (5), the final global output Y is determined by the weights$\{b,w\}$. In order to update the weights, let Y^* be the target output, then define the error function:-

$$E = \frac{1}{2}(Y - Y^*)^2$$

Using LMS technique and the gradient descent algorithm, we can get the following iterative equations for updating the weights:-

$$w_{ij0}(t+1) = \frac{w_{ij0}(t) - \alpha w_{ij0}(t) + \frac{\beta}{p}\sum_{k=1}^{p}\left[\frac{A_{kij} y_i l(Y^* - Y_k)(c_{ki} - Y_k)|w_{ij1}(t)x_{kj} - w_{ij0}(t)|^{l-1}}{\sum_{r=1}^{n} y_{kr}}\right]}{1-\alpha}$$

$$w_{ij1}(t+1) = \frac{w_{ij1}(t) - \alpha w_{ij1}(t) + \frac{\beta}{p}\sum_{k=1}^{p}\left[\frac{A_{kij} y_i x_{kj} l(Y^* - Y_k)(c_{ki} - Y_k)|w_{ij1}(t)x_{kj} - w_{ij0}(t)|^{l-1}}{\sum_{r=1}^{n} y_{kr}}\right]}{1-\alpha}$$

$$b_{i0}(t+1) = \frac{b_{i0}(t) - \alpha b_{i0}(t) + \frac{\beta}{p}\sum_{k=1}^{p}\left[\frac{A_{kij} y_i (Y^* - Y_k)}{\sum_{r=1}^{n} y_{kr}}\right]}{1-\alpha}$$

$$b_{ij}(t+1) = \frac{b_{ij}(t) - \alpha b_{ij}(t) + \frac{\beta}{p}\sum_{k=1}^{p}\left[\frac{A_{kij} y_i x_{kj} (Y^* - Y_k)}{\sum_{r=1}^{n} y_{kr}}\right]}{1-\alpha}$$

where p is the number of training samples, α and β is usually in the range of (0, 1), and:-

$$Y_k = \frac{\sum_{i=1}^{n} c_{ki} \prod_{j=1}^{m} \exp(-|w_{ij1}(t)x_{kj} + w_{ij0}(t)|^{l})}{\sum_{r=1}^{n} y_{kr}}$$

$$y_{kr} = \prod_{j=1}^{m} \exp(-|w_{rj1}(t)x_{kj} + w_{rj0}(t)|^{l})$$

$$c_{ki} = b_{i0} + \sum_{j=1}^{m} b_{ij} x_{kj}$$

where $k = 1, 2, \ldots, p$, $i = r = 1, 2, \ldots, n$, x_{kj} is the j^{th} component of input variables for the training pattern k and $\wedge$ is fuzzy MIN or product operator.

The initialisation of $\{b_{i0}, b_{ij}, w_{ij0}, w_{ij1}\}$ is as follows: $b_{i0} = 1 - \sum b_{ij}$; $b_{ij} = r_{ij} / m$, where r_{ij} is a random number in the range of (0, 1); the initial weights of w_{ij0} and w_{ij1} is the same as the previous FNN model.