WATER SATURATION AND POROSITY PREDICTION USING BACK-PROPAGATION ARTIFICIAL NEURAL NETWORK (BPANN) FROM WELL LOG DATA

S. Mahmoudi[1*], A. Mahmoudi[2]

[1]Mining Engineering Department, Isfahan University of Technology, Isfahan, Iran

[2]Computer & Electrical Engineering School, University of Tehran, Tehran, Iran

ABSTRACT

Porosity and water saturation are two fundamental parameters in reservoir characterization. In this study, for predicting both mentioned parameters artificial neural network was used as intelligent technique. Five variables include neutron log, effective porosity, caliper log, bulk density, and sonic log were used from 3 wells from one of the Iranian oil fields. After normalizing data Seventy percent of data were used as training dataset and remainder for testing the network. Several feed –forward neural networks were operated to obtain best performance of different algorithms to train the network. Levenberge-Marquardt back-propagation algorithm was chosen as the training algorithm which had the best performance and was faster than other algorithms. Optimum neurons in the hidden layer for porosity and water saturation were obtained respectively. Results shown that Back-propagation artificial neural network (BPANN) has a high ability to predict porosity and water saturation which correlation between real output and predicted output using BPANN were 0.82 and 0.93 respectively.

***KEYWORDS**: Porosity; water saturation; neural network; back-propagation algorithm*

1.0 INTRODUCTION

Reservoir characterization is a process for quantitatively assigning reservoir prospects, such as porosity, permeability, and fluid saturations, while recognizing geologic information and uncertainties, in special variability. The immediate application of reservoir characterization is in reservoir modeling and simulation and any primary and/or enhanced recovery design process in petroleum and natural gas industry (Mohaghegh et al., 1996).

* Corresponding author email: sh.mahmoudi1989@gmail.com

Mathematical models used to calculate porosity from density log responses yield the absolute or total porosity of the formation. Absolute porosity is not useful for reservoir characterization because, due to the heterogeneity of the reservoir, some of the pores may be isolated and not connected to the pore network of the reservoir. These isolated pores neither contribute to the reserves of oil and gas nor to the production (Mohaghegh et al., 1996). Porosity and water saturation are two crucial parameters in the petroleum reservoirs. In the recent years, many novel techniques were suggested for determining these parameters (Mohaghegh et al., 1996; Nickravesh & Aminzadeh, 2001; Wong et al., 1995a; Nickravesh, 2004). The common methods that has been used is Back-Propagation Neural Network which has a high ability in prediction of these parameters (Nikravesh, 1998; Wong et al., 1995b; Lim, 2003; 2005, Mohaghegh & Ameri, 2000). In this paper, the BPNNs are created using different training algorithms for selecting the best technique for predicting water saturation and porosity.

1.1 Back-Propagation Neural Network (BPNN)

The back-propagation algorithm neural network (BPNN) trained by the generalized delta rule (Rumelhart et al., 1986) is successfully used in many fields like petroleum and mining science problems.

To train a BPNN, the first input pattern is presented to an initially randomized BPNN, and the weights (including thresholds of nodes) adjusted in all the connection. Other pattern is then presented in succession, and the weights were adjusted from the previously determined values. This process continues until all patterns in the training set are exhausted (an iteration). The final solution is generally independent of the order in which the example patterns are presented. However, a final check can be performed by looking at the pattern error, E_p, the sum of the squares of the difference between the desired output and BPNN output for each pattern, and system error, E_s, the average of all pattern errors, to determine whether the final BPNN solution satisfies all of the patterns presented to it within a certain error (Dai and MacBeth, 1997).

The generalized delta rule which is used to train BPNNs can be mathematically written as:

$$\Delta w_{ijk}(n+1) = \eta \delta_{ik} o_{ij} + \alpha \Delta w_{ijk}(n), \quad (1)$$

Where η is the learning rate and α the momentum rate. $\Delta wijk$ is the change of the weighted connection between n_{ij} (n_{ij} is defined as the j^{th}

node in the i^{th} layer) and $n_i+_1\ k$, o_{ij} the output of n_{ij}, and δ_{ik} change in erroe as a function of the change in the network input to the $n_i+_1\ k$. The quantity $(n+1)$ indicates the $(n+1)^{th}$ step. This equation means the change of weights at the $(n+1)^{th}$ step should be similar to the change of weights undertaken at the nth step (Dai & MacBeth, 1997).

2.0 ARTIFICIAL NEURAL NETWORK PREDICTION

In this paper, a neural network is proposed to predict porosity and water saturation with feed-forward network. The input variable is a neutron log, effective porosity, caliper log, bulk density, and sonic log. A feed - forward consists of one or more hidden layer and one input layer and one output layer. All input and output data should be transformed into a range of zero and one for entering to determine activation function.

The neural network is defined as a system of simple processing elements, called neurons, which are connected to a network by a set of weights. The operated structure of neural networks in this study has been demonstrated in the Figure 1. A two-layer ANN with a tan-sigmoid transfer function for the hidden layer and a linear transfer function for the output layer was used.

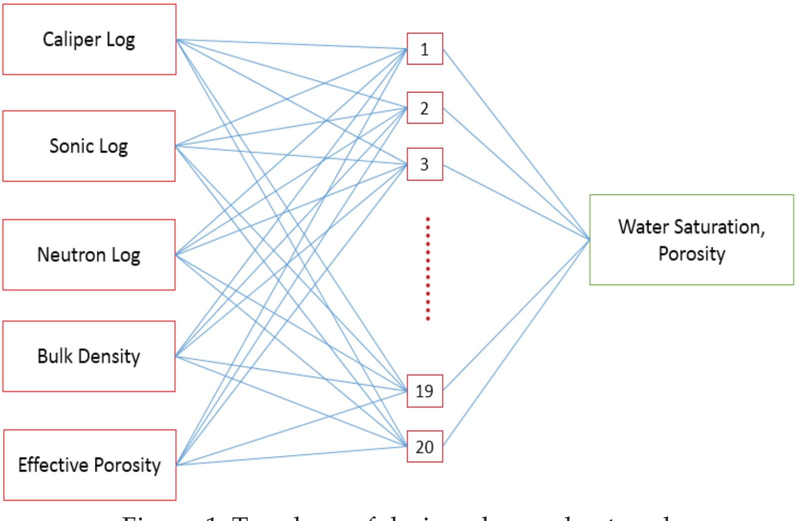

Figure 1. Topology of designed neural network

Learning is the main process in neural network operation because the simulation process depends on that (Jeirani & Mohebbi, 2006). Mathematically, learning is the process by which a set of weights is found that produces the expected output when a net is presented within an input. Therefore, ANNs learn tasks by changing the weight of links between nodes (Bean & Jutten, 2000; Bhat & Helle, 2002; Helle et al., 2001; Mohaghegh et al., 1994). Back-Propagation algorithms use input vectors and corresponding target vectors to train ANN.

In this paper, 70% of data were chosen for training and remain for testing process. At the start of training, the output of each node tends to be small. Consequently, the derivatives of the transfer function and changes in the connection weights are large with respect to the input. As learning progresses and the network reaches a local minimum in error surface, the node outputs approach stable values. Consequently, the derivatives of the transfer function with respect to input, as well as changes in the connection weights, are small.

Thirteen training algorithms in this study were used which basically utilize back-propagation algorithm consist quasi-Newton (BFG), Bayesian regulation (BR), Conjugate gradient with Powell-Beale restarts (CGB), Conjugate gradient with Fletcher-Reeves updates (CGF), Conjugate gradient with Polak-Ribiére updates (CGP), Gradient descent (GD), Gradient descent with adaptive learning rate (GDA), Gradient descent with momentum (GDM), variable learning rate backpropagation (GDX), One-step secant (OSS), Resilient backpropagation (RP), Scaled conjugate gradient (SCG), Levenberg-Marquardt (LM). More details about these algorithms are found in the MATLAB toolbox guide for neural network (Demuth & Beale, 2002).

There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode all of the inputs are applied to the network before the weights are updated. All of the algorithms in this study operate in the batch mode. Performance of all back-propagation algorithms are provided in the Table 1 according to correlation coefficient (R) and mean square error (MSE).

Table 1. Comparison of Back-propagation algorithms for predicting porosity and water saturation of the reservoir

Back-Propagation Algorithm	Porosity		Water Saturation	
	R-value	MSE	R- value	MSE
BFG	0.7789	0.0199	0.9154	0.0124
BR	0.7961	0.0186	0.9225	0.0100
CGB	0.7533	0.0219	0.8959	0.0151
CGF	0.7629	0.0212	0.8973	0.0149
CGP	0.7704	0.0206	0.8931	0.0155
GD	0.4990	0.0476	0.5634	0.0581
GDA	0.5778	0.0348	0.7343	0.0356
GDM	0.3812	0.0505	0.6379	0.0477
GDX	0.6751	0.0276	0.7968	0.0280
LM	0.8011	0.0182	0.9264	0.0095
OSS	0.7528	0.0219	0.8942	0.0154
RP	0.7617	0.0213	0.9059	0.0137
SCG	0.7510	0.0221	0.8932	0.0155

According to obtained results Levenberge-Marquardt had the best performance among other algorithms. Levenberg-Marquardt algorithm is specifically designed to minimize sum-of error function, of the form.

$$E = 1/_2 \sum k(e_k)^2 = 1/_2 ||e||^2 \quad (2)$$

Where e_k is the error in the k^{th} exemplar or pattern and e is a vector with element e_k. if the difference between the pervious weight vector and the new weight vector is small, the error vector can be expanded to first order by means of Taylor series.

$$e\,(i+j)= e(j) + \partial e_k / \partial_{wi}(w(j+1)-w(j)) \quad (3)$$

As a consequence, the error function can be expressed as

$$E= 1/_2 \,||\, e(j) + \partial e_k / \partial_{wi}(w(j+1)-w(j))||^2 \quad (4)$$

Minimizing the error function with respect to the new weight vector, gives

$$w\,(j+1) = w(j) - (Z^T Z)^{-1} Z^T e(j) \quad (5)$$

where

$$(Z)_{ki} = \partial e_k / \partial_{wi} \quad (6)$$

Since the hessian for the sum-of-square error function is

$$(H)_{ij} = \partial^2 E / \partial_{wi} \partial_{wj} = \sum\left\{\left(\frac{\partial e_k}{\partial w_i}\right)\left(\frac{\partial e_k}{\partial w_i}\right) + e_k \partial^2 e_k / \partial_{wi} \partial_{wj}\right\} \quad (7)$$

Neglecting the weights, therefore, involves the inverse Hessian or an approximation thereof for nonlinear networks. The Hessian is relatively easy to compute, since it is based on first order derivatives with respect to the network weights that are easily accommodated by back propagation. Although the updating formula could be applied iteratively to minimize the error function, this may result in a large step size, which would invalidate the linear approximation on which the formula is based.

In the Levenberg-Marquardt algorithm, the error function is minimized, which the step size is kept small in order to ensure the validity of the linear approximation. This is accomplished by use of a modified error function of the form:

$$E= \tfrac{1}{2} \,||\, e(j) + \partial e_k / \partial_{wi}(w(j+1)-w(j))||^2 + \lambda ||w(j+1)-w(j)||^2 \quad (8)$$

Where λ is a parameter governing the step size. Minimizing the modified error with respect to w(j+1) gives

$$w\,(j+1) = w(j) - (Z^T Z + \lambda I)^{-1} Z^T e(j) \quad (9)$$

Very large values of λ amount to standard gradient descent, while very small values λ of amount to the Newton method (Sapna et al., 2012).

For selecting optimum neurons in the hidden layer, the network should be trained with different neurons while other parameters should be constant. After operating different neurons in the hidden layer, the network with twenty two and twenty five neurons were selected for porosity and water saturation respectively regarding to high correlation and minimum mean square error. The final regression between real value and predicted porosity and water saturation is depicted in Figure 2.

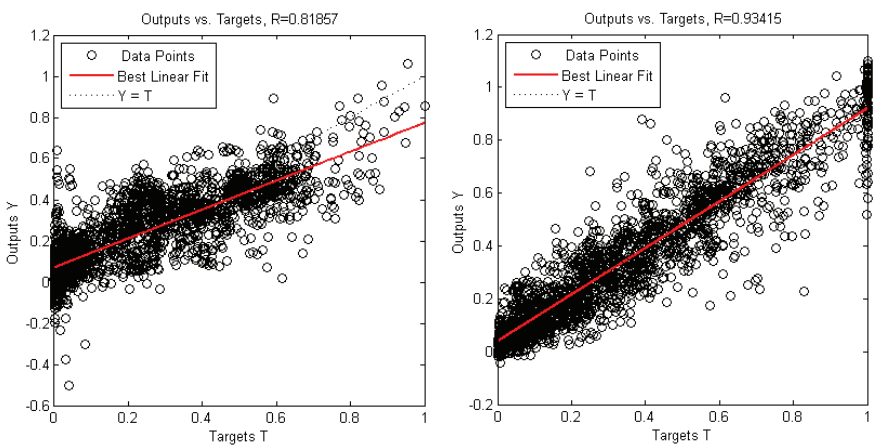

Figure 2. Correlation between output value and predicted value for porosity (left) and water saturation (right)

3.0 CONCLUSION

Porosity and water saturation for three wells in one of the Iranian oil fields were predicted using artificial neural network. Thirteen training functions were used to determine which one has better performance. After training with constant neurons in the hidden layer, Levenberge-Marquardt had a best performance for both porosity and water saturation. Training network with optimum neurons in the hidden layer for porosity and water saturation were 22 and 25 respectively. The correlation coefficient between real output and predicted outputs for porosity and water saturation were 0.82 and 0.93 respectively. The obtained results indicates that the artificial neural network with Levenberge-Marquardt Back-propagation algorithm had a reliable ability to predict porosity and water saturation of oil and natural gas reservoirs.

REFERENCES

Bean, M., & Jutten, C., (2000). Neural networks in geophysical applications. *Geophysics, 65*, 1032-1047.

Bhatt, A., Helle, H.B., (2002). Committee neural networks for porosity and permeability prediction from well logs. *Geophysical Prospecting, 50*, 645-660.

Dai, H., & MacBeth, C., (1997). The effect of learning parameters on learning procedure and performance of a BPNN, *Neural Networks, 10*(8), 1505-1521.

Helle, H.B., Bhatt, A., Ursin, B., (2001). Porosity and permeability study. *Geophysical Prospecting, 49*, 434-444.

Demuth, H., Mark Beale., (2002). Neural network toolbox; For use with MATLAB, Version 4, The MathWorks.

Jeirani, Z., & Mohebbi, A., (2006). Estimating the initial pressure, permeability and skin factor of oil reservoirs using artificial neural networks, *Journal of Petroleum Science and Engineering, 50,* 11-20.

Lim, J.S. (2003). Reservoir permeability determination using artificial neural network: *Journal of Korean Society of Geosystems Engineering, 40,* 232–238.

Lim, J. (2005). Reservoir properties determination using fuzzy logic and neural networks from well data in offshore Korea: *Journal of Petroleum Science & Engineering*, 182-192.

Mohaghegh, S., Arefi, R., Ameri, S., Rose, D., (1994). Design and development of an artificial neural network for estimating permeability. *Society of Petroleum Engineers*. SPE 28237.

Mohaghegh, S., Arefi, R., Ameri, S., Amini, K., Nutter, R., (1996). Petroleum reservoir characterization with the aid of artificial neural networks. *Journal of Petroleum, Science and Engineering, 16*, 263–274.

Mohaghegh, S., & Ameri, S., (2000). Artificial neural network as a valuable tool for petroleum engineers. West Virginia University.

Nikravesh, M., & Aminzadeh, F., (2001). Mining and fusion of petroleum data with fuzzy logic and neural network agents. *Journal of Petroleum Science and Engineering, 29*(3/4), 221–238.

Nikravesh, M. (2004). Soft computing-based computational intelligent for reservoir characterization. *Expert Systems with Applications, 26,* 19-38.

Nikravesh, M. (1998). A Neural network knowledge-based modeling of rock properties based on well log databases. B SPE 46206, 1998 SPE Western Regional Meeting, Bakersfield, California, 10–13 May.

Rumelhart, D., Hinton, G.E., Williams, R.J., (1986). Learning representations by back-propagation errors. *Nature, 323,* 533-536.

Sapna, S., Tamilarasi, A., Parvin Kumar, M., (2012). Backpropagation Learning Algorithm Based On EVENBERG MARQUARDT Algorithm, *Computer Science & Information Technology(CS & IT), 2,* 393-398.

Wong, P.M., Jiang, F.X., Taggart, I.J., (1995a). A critical comparison of neural networks and discrimination analysis in lithofacies, porosity and permeability prediction. *Journal of Petroleum Geology, 18,* 191.

Wong, P. M., Gedeon, T. D., Taggart, I. J., (1995b). An improved technique in prediction: a neural network approach: *IEEE Transactions on Geoscience and Remote Sensing, 33*, 971.