Neural Network Approach to Predict Average Reservoir Pressure

Saber Elmabrouk, Ezeddin Shirif and Rene Mayorga*

Abstract: The average reservoir pressure is an important parameter in petroleum engineering which is utilized in almost all reservoir and production engineering calculations. Normally, the average reservoir pressure is obtained from build-up test and can be measured only when the well is shut in. Besides that, it requires an update from time to time. It also has a great economic impact caused by shutting in the well during the entire test. However, the objective of this study is to establish a neural network model that can map certain relationship that controls previous production performance of the reservoir to predict current average reservoir pressure without shutting in the wells.

Keywords: Reservoir, Pressure, Average, Neural Network, Model.

INTRODUCTION

STraditionally, the average reservoir pressure is obtained from build-up test when the well is shut in by measuring the long-term build-up pressure. The build-up pressure eventually builds up to the average reservoir pressure over a long enough period of time. This time period depends mainly on the reservoir permeability. It may take long in low permeability reservoirs. Because reservoir pressure changes as fluids are produced from a reservoir, the average reservoir pressure should be updated from time to time. In addition, it has a great economic impact caused by shutting in the wells during the entire test.

The current state-of-the-art in production average reservoir pressure, specifically in cases of application of artificial intelligent techniques, is quite limited. In this work we established a neural network model that can map certain relationship that control previous oil, gas and water production performance of the reservoir to predict current average reservoir pressure without needing to close the wells. This method is suitable for both constant and variable flow rates. After obtaining the average reservoir pressure, it can be utilized in almost all reservoir and production engineering studies. Some of these studies include computing rock and fluid characteristics, estimating hydrocarbons in place, establishing the value of water influx, and predicting future reservoir behaviour in primary/secondary recovery and pressure maintenance projects.

Artificial neural networks (ANNs): ANNs are relatively new computation tools that have found extensive utilization in solving many complex engineering problems. A basic network shown in Fig. 1 has three layers of processors: an input layer, an output layer, and a one hidden layer in between. Each layer has a number of neurons or nods.

At first, both input data and corresponding desired output data are given to the network. As the network starts training, the input layer receives the input signals, and then starts processing the data through the hidden layers until reaching the output layer yielding the resulted outputs. These outputs are then compared with the desired outputs computing the error, which is back propagated through the system causing it to adjust the weights,

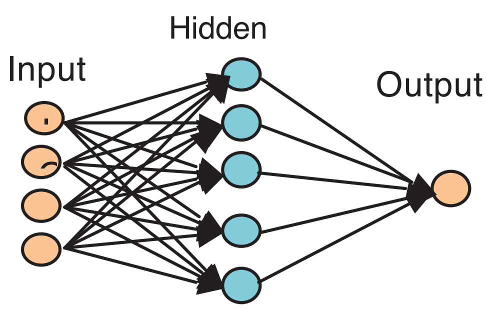

Fig. 1. A generalized artificial neural network. It has an input layer with four neurons, one hidden layer with five neurons and an output layer with one neuron.

*University of Regina, 3737 Wascana Parkway, Regina, SK, S4S 0A2

which control the network. Once a neural network is trained to satisfactory level, it may be then used as an analytical tool on other data.

The feedforward error-backpropagation is the most famous procedure for training ANNs. It is based on searching an error as a function of ANNs weights. Each iteration in the backpropagation constitutes two sweeps: forward activation to produce a solution and a backward propagation of the computed error to modify the weights (Basheer and Hajmeer 2000).

Modeling with neural networks is not an easy task. This poses considerable challenges for engineers particularly in terms of the requirement to realize sufficient robust models. The major challenges could be summarised as following:

Choosing the model structure: Once the number of input and output nods is defined by the problem, the number of hidden layers and nodes in each layer is far from clear. Therefore, the first decision we will need to make is: how many hidden layers and how many nodes in each hidden layer? In most situations, there is no way to determine the best number of hidden units without training several networks and estimating the generalization error of each. If we have too few hidden units, we will get high training error and high generalization error due to under-fitting and high statistical bias. If we have too many hidden units, we may get low training error but still have high generalization error due to over-fitting and high variance (Geman, *et al* 1992).

Selecting Training Algorithm: After the number of hidden layers and number of nods in each layer has been selected, the next step is selecting the training algorithm. The objective of the training algorithm is to determine the global minimum of the error surface. In this process, the network weights must be set so as to minimize the prediction error made by the model. This is the role of the training algorithms. The historical cases (inputs and outputs) are used to automatically adjust the weights in order to minimize this error. This process is equivalent to fitting the model represented by the network to the training data available. The error of a particular configuration of the network can be determined by running all the training cases through the network, comparing the actual output generated with the target outputs. The differences are combined together by an error function to give the network error. The most common error functions (cost functions) are the sum squares error, where the individual errors of the output units on each case are squared and summed together. Despite the wide range of learning algorithms being available to train the ANNs model, there is no single learning algorithm that works best on all learning problems. In fact, the choosing of training algorithm is based on the characteristics of the problem; consequently, training a neural network model essentially means selecting one model from the set of allowed models that minimizes the cost function.

Activation function: The specific activation function (also known as a transfer function) used at the active nodes of the feedforward ANNs is fixed before the model is trained. Activation functions are needed to introduce nonlinearity into the network. Without nonlinearity, hidden units would not make nets more powerful than just plain perceptions. In other words, the neural networks pass the output of their layers through activation functions. These activation functions scale the output of the neural network into proper ranges. There are many activation functions that we can choose from and each one has its own special virtues. We may notice that, it would be possible to use a different activation function for each layer of the neural network. As a result, we have to choose an activation function suited to the distribution of input and output values. For more details of various activation functions see Bulsari (1995).

Ultimately, the selection of the architecture of the neural network will come down to trial-and-error procedure. If the model architecture and learning algorithm are selected appropriately, the resulting ANNs can be extremely robust.

The proposed model: In this study, the ANNs model is designed to estimate the current average reservoir pressure without needing to close the wells. The network is supervised, feedforward with backpropagation. After trying several models, the [4-5-1] network was found to be the best extremely robust model. The input layer received the following parameters; cumulative oil production (Np), gas flow rate (Qg), water flow rate (Qw) and number of wells on production. The output layer has one neuron which is average reservoir pressure (Pavg). The selected model contains one hidden layer with five neurons. The error function of the proposed model was sum-of-squares.

Data preparation and acquisition: A Libyan oil reservoir located in Sirt Basin consisting of 56 oil wells was utilized in this study to develop a model to estimate the average reservoir pressure. The reservoir started its production in March 1970, and a total of 49 average reservoir pressure data points were recorded during a 39 years of production (from March 1979 to May 2009). The range of this dataset is illustrated in Table 1.

To build a robust model, the last two recorded average reservoir pressure (June 2007 and May 2009) was put beside in order to test its prediction performance after it trained properly. The rest of the dataset (47 points) were used to build the network. Besides, to avoid over fitting (overtraining), it is necessary to use early stopping technique. In early stopping, the training data points are split randomly into three sets, training, validation and test sets. The validation set (6 points) is used in learning to decide when to stop. This illustrated in Fig. 2. The Figure shows a typical plot of how the sum-of-squares error changes with the number of iterations. If the validation error increases while the training error steadily decreases then a situation of over fitting may have occurred. When the performance with the validation test stops improving, the algorithm halts. After that, the test set (7 points) was used to evaluate how good an ANNs performs on data that it has not seen before. The ranges of the aforesaid datasets are shown in Table 2.

Table 1. Range of All Used Data

Model parameter	Max	Min	Average
Pavg, psi	1188	820	924
Np, STB	492MM	492M	352MM
Qw, bbl/d	155036	3572	73280
Qg, scf/d	21902	1732	7969
No. of wells	56	12	40

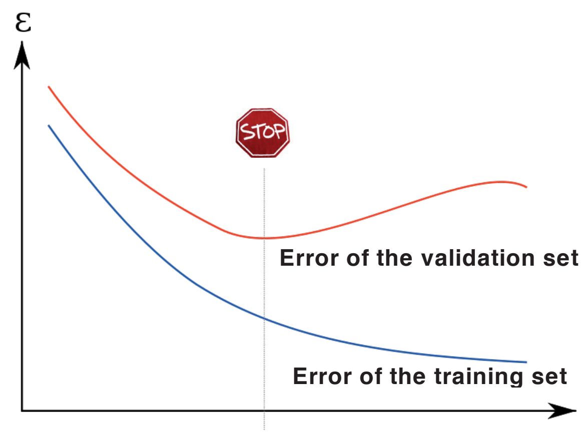

Fig. 2. Error as a function of training time. The red curve is the error on the validation set. The blue curve is the error of the training set.

Neural network architecture: The architecture of the neural network used in this study is multilayered with 4 input nodes, one hidden layer with 5 nodes, and 1 output node, [4-5-1]. The number of hidden nodes, training algorithm and activation function are determined through trial-and-error procedure.The network was trained using feedforward backpropagation with Quasi-Newton training algorithm. The neurons in the backpropagation used hyperbolic tangent as input activation function and logistic activation function in the output. A hyperbolic tangent activation function gives values that range from -1 to 1. A logistic activation function yields an output varies from 0-1.

Consequently, the proposed network provides the best minimum error for training, validation and test of the network Fig. 3, 4 and 5 respectively. It is obviously from the Figures that the network provides results very close to actual average reservoir pressure. This indicates an excellent agreement between the actual and the calculated average reservoir pressure. The statistical parameters for the prediction capability

Table 2. Range of Data using in Training Process

	Training Set			Validation Set			Test Set		
	Max	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.
Pavg, psi	1188	833	933	1072	841	931	1081	853	906
Np, STB	487MM	492M	372M	387MM	618MM	378MM	393MM	563M	364MM
Qw, bpd	155036	3572	71590	114499	31453	667588	120272	22544	85388
Qg, scf/d	21902	1732	8149	17832	2458	8976	18118	1748	7965
Wells	53	12	39	49	15	36	55	37	43

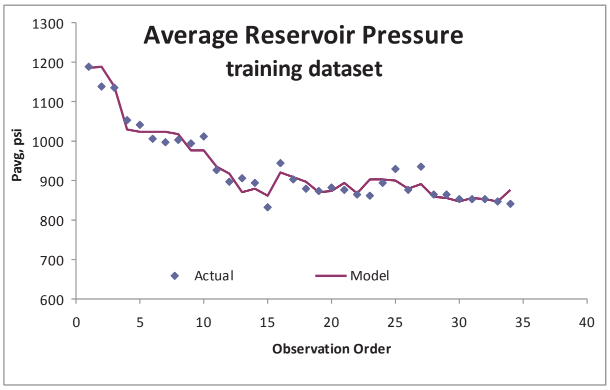

Fig. 3. Network performance, training dataset

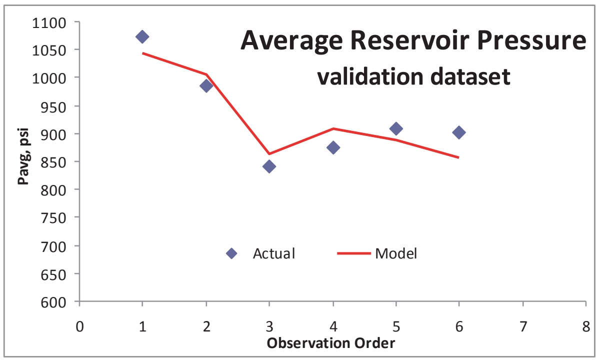

Fig. 4. Network performance, validation dataset

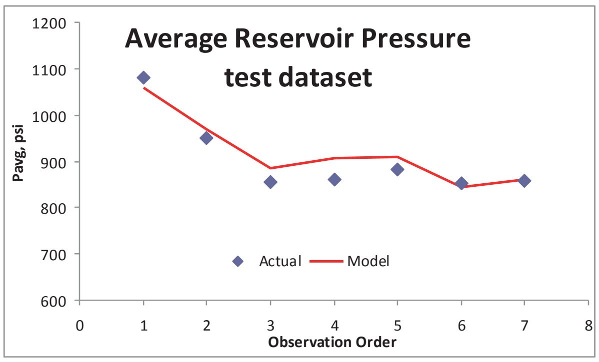

Fig. 5. Network performance, test dataset

obtained from the network are summarized in Tables 3. The proposed network provides prediction values of average reservoir pressure with an average absolute error (AAE) of 18.816 and average absolute relative error (AARE) of 2.033% indicating that the model describes the data well.

To study the prediction performance of the proposed network, two unseen points have been used to evaluate its performance Fig. 6. Table 4 summarizes the statistical error analysis. The analysis shows a small error indicating that the model is robust.

Table 3. Statistical Parameters of the Proposed Network Model

	AAE	AARE, %	MAARE, %
Training	16.814	1.792	4.48
Validation	28.23	3.061	5.027
Test	22.44	2.494	5.416
All	18.816	2.033	5.416

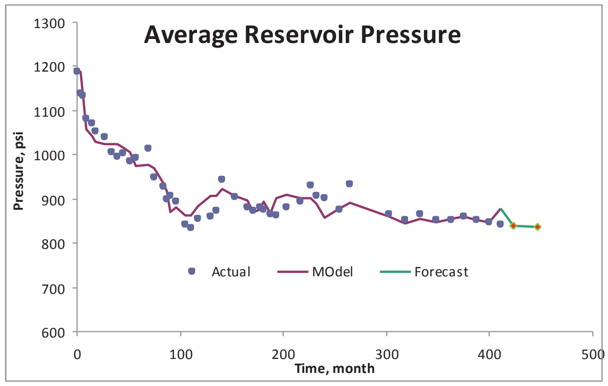

Fig. 6. Prediction of average reservoir pressure.

Table 4. Statistical Analysis for Prediction Performance

Time	wells	Model, psi	Actual, psi	AE	RAE
6/2007	56	839.32	832	2.356	0.287
5/2009	56	830.51	820	1.017	0.122

CONCLUSIONS

A neural network model was developed to predict current average reservoir pressure without needing to close the wells. The model is suitable for both constant and variable flow rates. The data in use were a set of 49 measured average reservoir pressure collected from a Libyan oil reservoir located in the Sirte Basin. The network consisted of one hidden layer with five neurons. The input layer received four parameters; cumulative oil production, water production rate, gas production rate and number of wells on production. The network was trained using feedforward backpropagation with Quasi-Newton training algorithm.

The results reflect that the application of neural network models is feasible for prediction of the current average reservoir pressure of oil reservoirs without the need to close the wells. Nevertheless, the neural network model has the ability to predict current average reservoir pressure accurately using previous reservoir production data. Further study can proceed to construct the most suitable neural network models for other oil fields.

ACKNOWLEDGMENT

We wholeheartedly thank and appreciate FGSA, University of Regina for their generous financial support.

REFERENCES

Basheer, I. A., Hajmeer, M. (2000). Artificial Neural Networks: Fundamentals, Computing, Design, and Application. Jornal of Microbiological Methods, 43: 3–31.

Bulsari, A. B. (1995). Neural Networks for Chemical Engineers, Elsevier, Amsterdam, 000p.

Geman, S., Bienenstock, E. and Doursat, R. (1992). Neural Networks and the Bias Variance Dilemma. Neural Computation, 4: 1-58.

Suchatita Gropal. (1998). Artificial Neural Networks for Spatial Data Analysis http://www.ncgia.ucsb.edu/giscc/units/u188/u188.html