Optimization of neural network using genetic algorithm in order to predict reservoir permeability

Mohsen. Saemi[i]; Morteza. Ahmadi[ii]; and Seyed Omid Gilani[iii]

ABSTRACT

Permeability prediction in hydrocarbon reservoirs is probably the most challenging issue geologists, petrophysicists and reservoir engineers have to deal with. In particular, to understand reservoir performance and in case of reservoir management and development requires accurate knowledge of permeability. Attempts were made to utilize artificial neural networks (ANNs) for identification of the relationship which may exist between the well log data and core permeability. Despite the wide range of applications and flexibility of the ANNs, there is still no general framework or procedure through which the appropriate network for a specific task can be designed. Design and structural optimization of the neural networks is still very dependent upon the designer's experience. This is an obvious barrier to the wider applications of neural network. To mitigate this problem, a new method for the auto-design of neural network has been used, which is based on genetic algorithm (GA). The new proposed method is evaluated by a case study in South Pars gas field in the Persian Gulf. Design of the topology and parameters of the neural network as decision variables is done first by trial and error, and then using the genetic algorithms in order to improve the effectiveness of forecasting when ANN applied to a permeability predicting problem from well logs. It is shown that a carefully designed neural network is able to predict rock permeability with accuracies comparable to actual measurements.

KEYWORDS

Well log data; neural network; genetic algorithms; permeability; reservoir rock; hydrocarbon reserviors.

1. INTRODUCTION

Permeability is a key parameter associated with the characterization of any hydrocarbon reservoir. Knowledge of rock permeability and its spatial distribution throughout the reservoir is of utmost importance. Permeability is generally measured in the laboratory (direct measurement) on cored rock taken from the reservoir or can be determined by analyzing well test data (indirect measurement). The well testing and coring are expensive and time-consuming in comparison to the electronic survey techniques which most commonly used in all wells. Moreover, in a typical oil or gas field, all wells are "logged" using electrical tools to measure geophysical parameters such as porosity and density, while both well test data and core data are available only for a few wells.

Alternatively, the neural networks have been increasingly applied to predict reservoir properties using well log data [1]-[3]. Moreover, previous investigation [4], [5], [6] indicated that artificial neural networks (ANNs) can predict formation permeability even in highly heterogeneous reservoirs using geophysical well log data with good accuracy.

The major reason for this rapid growth and diverse application of neural networks is their ability to approximate virtually any function in a stable and efficient way. By using ANNs, it is possible to create a platform on which different models can be constructed. In spite of the wide range of applications, neural networks are still designed through a time-consuming iterative trial and error approach. Hence, the time and effort required for network design are totally dependent on the nature of the task and designer's experience. This leads to a significant amount of time and effort being expended to find the optimum or near optimum structure for a neural network

i M. Saemi is with the Department of Mining Engineering, Tarbiat Modares University, Tehran, Iran (e-mail: mo_saemi@yahoo.com).
ii M. Ahmadi is with the Department of Mining Engineering, Tarbiat Modares University, Tehran, Iran (e-mail: mosami@modares.ac.ir).
iii S.O. Gilani is with the Department of Mining Engineering, Tarbiat Modares University, Tehran, Iran (e-mail: omidgilani@modares.ac.ir).

for the desired task. In order to mitigate these deficiencies, various methods for auto-design of neural networks have been proposed [7]. However, these methods have been applied only for the design of neural networks used for simple tasks and not for more complex problems typically encountered in hydrocarbon reservoir management.

The purpose of this study is to apply genetic algorithms (GAs) to determine the number of neurons in the hidden layers, the momentum, and the learning rates for minimizing the time and effort required to find the optimal architecture and parameters of the back-propagation algorithm. It also focused on improving the accuracy of permeability estimation and verifying the performance and validity of optimizing both the neural network size and its parameters using GA.

The attempts [2,3] to predict permeability from well log data have generally been in the form of empirical correlations between permeability, porosity, and water saturation. This technique has been used with some success in sandstone and carbonate reservoirs; however, the existing correlations are mainly for homogeneous formations that have fairly constant porosity and grain size. The prediction of permeability in heterogeneous formations from well log data poses a difficult and complex problem. A comprehensive approach [2] for correlating permeability with geophysical well log data in heterogeneous formations was previously developed, but these cannot predict rock permeability adequately.

Previous investigations [8] revealed that neural network is a powerful tool for identifying the complex relationship among permeability, porosity, fluid saturations, depositional environments, lithology, and well log data. Back-propagation-type neural networks and genetic algorithms are introduced very briefly in this paper, since extensive information already exists in the literature.

2. INTELLIGENT SYSTEMS

2.1. Neural networks

Back-propagation-type neural networks have an input, an output and, in most of the applications, have one hidden layer. The number of inputs and outputs of the neural network are determined by considering the characteristics of the application. In most of the cases, one hidden layer is satisfactory.

Each neuron of a layer is generally connected to the neurons in the proceeding layer. Repeating forward-propagating and backward-propagating steps performs the learning required. When a pattern is given to the input pattern, the forward propagation step begins. The activation levels are calculated and results propagate forward through the following hidden layers until they reach the output layer. Every processing unit sums its respective inputs and then applies a function to compute its output. Sigmoid is the most commonly used function [9].

The output of the network is created at the output layer. The bias units of input and hidden layer add a constant term in the weighted sum, which improves convergence. After the network's output pattern is compared with the target vector, error values for the hidden units are calculated and their weights are changed. The backward propagation starts at the output layer and moves backward through the hidden layers until it reaches the input layer [9]. Figure 1 shows a summary of the network topology illustration.

Any attempt to build a predictive model based on artificial neural networks generates the need to investigate how appropriate the network's architecture and computational paradigm are for the task that it is supposed to handle. For neural networks based on supervised learning, the data are usually split into a training set, a training validation set, and a test set.

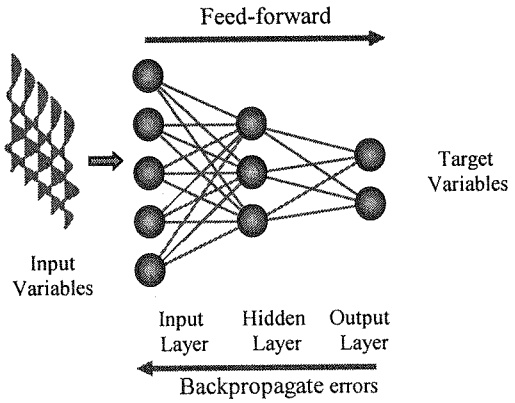

Figure 1: Back-propagation multilayer ANN with one hidden layer.

The training set consists of a set of examples used only for learning, (i.e., to fit the weights of the network). The training validation set is a set of examples used to adjust the network parameters such as network architecture (for example number of hidden layers and neurons, or number of training cycles). The test set is a set of examples used only to assess the generalization performance of a trained neural network, (external valuation). Various networks are trained by minimization of an appropriate error function defined with respect to the training set. The performance of the networks is then compared by evaluating the error function using the training validation set, and the network having the smallest error with respect to the training validation set is selected [10].

The cross-validation data's errors were measured by mean square-error (MSE) as defined in Eq. (1):

$$MSE = \frac{\sum_{i=1}^{n} (O_i - T_i)^2}{n}, \qquad (1)$$

Where: O*i* is the desired output for training data or cross-validation data *i*, T*i* is the network output for training data or cross-validation data *i*, and n is the number of data in the training data set or cross-validation data set.

2.2. Genetic algorithm

In 1975, Holland introduced an optimization procedure that mimics the process observed in natural evolution called genetic algorithms. GA is a stochastic search algorithm inspired by the mechanics of natural evolution, including survival of the fittest, reproduction, cross-over, and mutation. GAs are based on a Darwinian survival of the fittest strategy, and work with a population of individuals, each of which represents a potential solution to a given problem. Each individual, or candidate, solution in the population is generally represented as a linear string analogous to chromosomes in the GAs. The basic algorithms in GAs are selection (reproduction), cross-over, and mutation operators (called genetic operators).

As originally proposed, a simple GA usually consists of three processes selection, genetic operation and replacement. A typical GA cycle is shown in Figure 2. The population comprises a group of chromosomes that are the candidates for the solution. The fitness values of all chromosomes are evaluated using an objective function (performance criteria or a system's behavior) in a decoded form (phenotype). A particular group of parents is selected from the population to generate offspring by the defined genetic operations of crossover and mutation. The fitness of all offspring is then evaluated using the same criterion and the chromosomes in the current population are then replaced by their offspring, based on a certain replacement strategy.

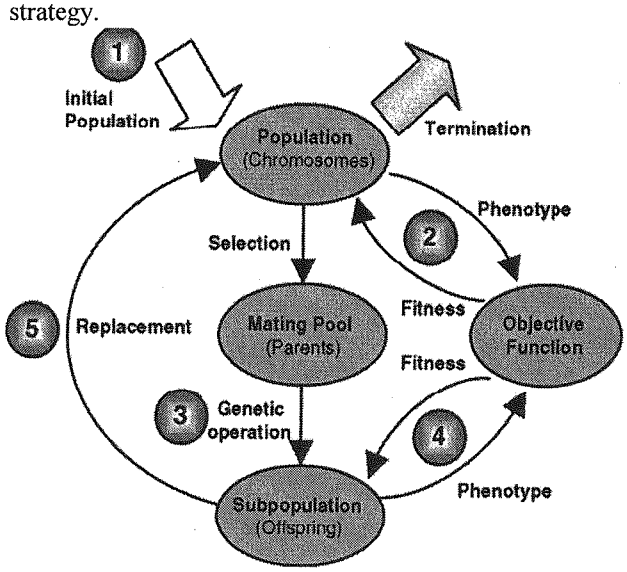

Figure 2: Genetic algorithm cycle and a simple top level description.

Such a GA cycle is repeated until a desired termination criterion is reached. If all goes well throughout this process of simulated evolution, the best chromosome in the final population can become a highly evolved and more superior solution to the problem [11].

In this paper, we implement a genetic algorithm to optimize one or more parameters within the neural network. The most common parameters to optimize are the input vectors, the number of hidden Processing Elements (PEs), the learning rates, and the momentum coefficient.

2.3. Neural network incorporating genetic algorithm in permeability prediction

The architecture of ANNs has considerable influence on their performance. If the number of neurons is too small, they cannot remember the input and output patterns in training. If the number of neurons is too large, performance may be improved, but the large number of connections extends the training time and increases computation burden. It is therefore desirable to establish networks, which are as simple as possible, provided their errors are within the tolerance limits.

Many researchers [12] have therefore addressed these needs by applying GAs in optimizing ANNs' parameters. In particular, GAs are found to be quite useful and efficient when the exploration space of the ANN is extensive. Research by Rooiji et al [13] and Vonk et al [14] proposed using evolutionary computation, such as GAs, in the field of ANNs to generate both the ANN architecture and its weights. Those [15], [16], [17] who supported the proposal were in favour of optimizing the connection weights and the architecture of ANNs by using GAs. In addition, research on permeability estimation from well logs [18], [19], showed that it is highly effective to apply integrated GAs to ANNs in permeability prediction. However, their work did not cover the optimization of ANN parameters using GAs. In our study, GA- optimized ANN predicts the reservoir permeability more accurately than a network in which the ANN calibration is done by trial-and-error approach.

3. APPLICATION

3.1. Geological discription

The South Pars gas field is located in the Persian Gulf, at some 100km from shore. Four huge condensate rich gas-bearing reservoirs have been identified (K1, K2, K3 and K4) in this field. With the data acquired, the Kangan-Upper Dalan (K1 to K4) reservoirs can be described as structurally complex reservoirs on which some geological and reservoir unknowns still exist. Several reservoirs '*oil and gas*' were identified by the exploration and appraisal wells. The oil reservoirs of South Pars are in Khami and Bangestan groups formations. The gas-bearing reservoirs of the field belong to the Kangan and Dalan formation of Triassic and Permian age, respectively. These carbonate formations were deposited in a shallow marine environment during a general marine transgression which began in the middle of Permian and lasted until the early Triassic.

3.2. Modelling data

The total number of wells drilled at the time of this study added up to 43 vertical wells. All the wells have

been logged but three of them do not have sufficient number of logs needed for petrophysical evaluation. Therefore, the petrophysical data of almost wells were available. Wireline logs obtained from these wells are gamma-ray, water saturation, density, neutron porosity, sonic porosity, depth, photoelectric factor and resistivity log. Only 6 of the wells have been cored in the reservoir layer. Therefore, as usual cores data are only available from few wells in reservoir while the well logs are available from the majority of the wells. Thus, the evaluation of permeability from well log data represents a significant technical as well as economic advantage.

3.3. Data analysis

The data were analyzed to establish relevant relationships between log and core data with the aim of evaluating heterogeneity in a hydrocarbon reservoir formation. Heterogeneity is referred to non-uniform, non-linear spatial distribution of rock properties in a hydrocarbon reservoir. In order to demonstrate the degree of heterogeneity of this formation and to show chaotic status of the information, a cross plot of permeability versus gamma ray log responses is presented in Figure 3. In addition, Figure 4 shows permeability variation with respect to bulk density well log responses.

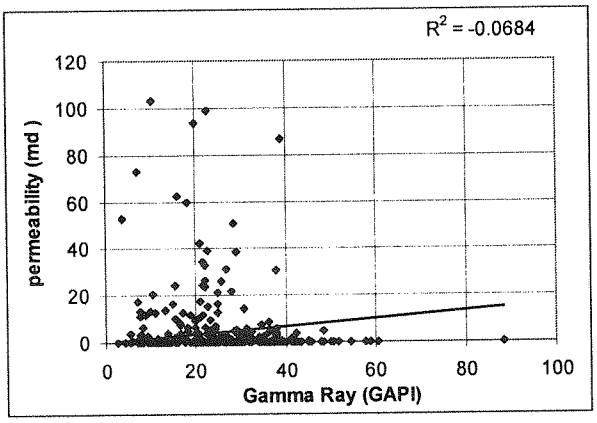

Figure 3: Permeability vs. Gamma ray log responses for wells.

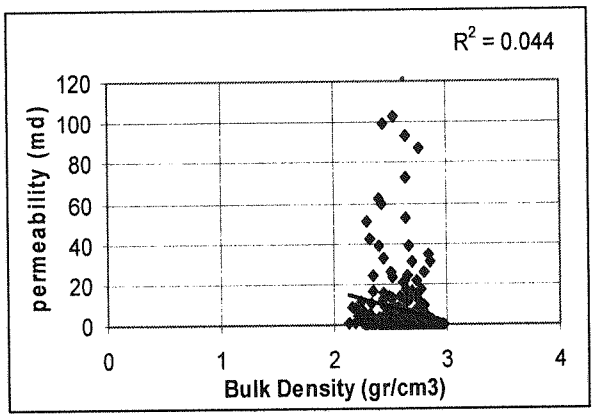

Figure 4: Permeability vs. Bulk density log responses for wells.

As Figures 3 and 4 demonstrated, there are no apparent relationships between logs and core permeability data, because the correlation coefficients (R2) between logs and core permeability data are very poor (where one number is the best fitness). Since bulk density may be regarded as a symbol of porosity rock, (regardless of how linear or non-linear this relationship might be) it may contribute to the permeability of the formation. Gamma ray log response is an indication of shalyness or clay content of formation. This property of the rock may have some impacts on its ability to conduct fluid.

3.4. Network designing by using Genetic Algorithms

In this study, the process of designing of the network by NeuroSolutions for Excel Release 4.2 software is being managed. This software which incorporates ANN and GA, produced by NeuroDimension, Inc., and used to obtain the optimal network size and parameters in the ANN permeability estimation model. In this study, a GA application process for optimizing the parameters (the number of neurons in the hidden layer, the coefficient of the learning rate and the momentum) of ANNs is shown in Figure 5.

All chromosomes were automatically initialized in NeuroSolution by values equal to number of hidden neurons, learning rate, and momentum. NeuroSolution also can automatically produce these initial values. These values for number of the hidden layer neurons were set in a range between a lower bound of m/2 where m is the number of input neurons and an upper bound of 3m, whilst the step size had a lower bound of 0 and an upper bound of 1, and the momentum had a lower bound of 0 and an upper bound of 1.

Population size and number of generations affect processing time because the fitness value must be calculated for every chromosome in every generation [20]. Also, Kim et al, indicate that the best population size and generation number were set to 100 and 50, respectively. Hence the initial population pool value is set to 100 chromosomes, which had at least one different value for the arrayed ANNs parameters. Every chromosome in a population evolved into new chromosomes for 50 generations.

The back-propagation training module shown in Figure 5 was used to evaluate all chromosomes, that is, after the parameter values for each chromosome were translated into the predefined ANN, the network was trained on the training data set and the cross-validation data set was used to test if the stopping criteria were satisfied. In this study, the training process of the network stopped after a maximum of 5000 epochs or until there was no improvement of the MSE for 182 epochs on cross-validation data set. The fitness of every chromosome was evaluated by measuring the MSE, which is the estimated result on a cross-validation data set. A better network has a lower training error but requires a higher fitness value. The cross-validation data were not used to train the ANN model but were used to test the ANN model in the training

stage.

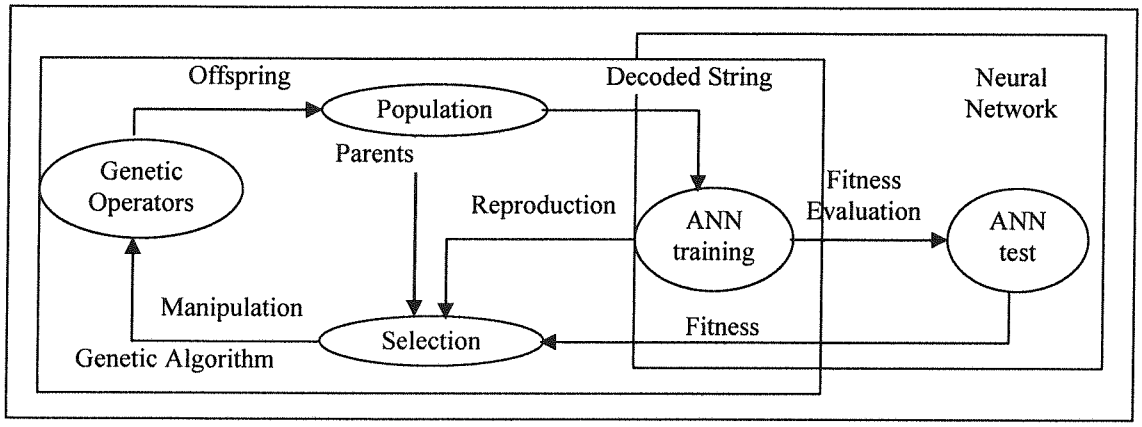

Figure 5: Combination of Genetic Algorithm and Neural network.

After evaluating all chromosomes, an intermediate population was created by extracting chromosomes from the current population using the reproduction operator. In this study, the roulette wheel selection based on ranking algorithm was applied for the reproduction operator.

The chromosomes were organized in the order of their fitness; that is, the chromosome with the lowest fitness received a rank of 0, and the next worst chromosome received a rank of 1, and so on. Chromosomes were selected in quantities according to their relative fitness after ranking in the roulette wheel operator and placement into the intermediate population [20]. Finally, the population of the next generation was formed by applying the cross-over and mutation operator to the chromosomes of the intermediate population.

Then, the new chromosomes reproduced by selection, cross-over, and mutation operators were evaluated, and this procedure for evaluation and reproduction of all chromosomes was repeated until the stopping criterion is satisfied. The basis of GA is the continual improvement of the fitness of the population by means of genetic operators, as individuals are passed from one generation to the next. After a number of generations, the population evolves to a solution close to optimal. In this study, uniform cross-over and uniform mutation operators were used and the probabilities of cross-over and mutation operators were adjusted in 0.9 and 0.01, respectively.

3.5. Simulation Data

All, 280 available data points are organized from all selected wells as input and output. The database to be introduced to the neural network is broken down into three groups: training, cross- validation, and verification. The network is trained using the training set data. The actual output of the training set data is used to develop the weights in the network. At established intervals, the test set is used to evaluate the predictive ability of the network. The cross- validation set also insures that network would not memorize the data which means a tendency for all new data to be regarded as identical to the training data. Training continues as long as the computed error between the actual and predicted outputs for the test set is decreasing. Typically 80% of the data is used for training and validation purposes. The other 20% of the data is categorized as verification. The verification set is used to evaluate the accuracy of the newly trained network by providing the network a set of data it has never seen.

There is possibility of using the current network weights or use the best network weights saved during a genetically training trial run. Note if a cross validation set is used during training, the best network weights are the ones that give the minimum cross validation error. Otherwise, the best network weights are the ones that give the minimum training error. During testing, the learning is turned off and the chosen data set is fed through the network. The network output is collected and a report is then generated showing the testing results.

4. Results and Discussions

4.1. Trail & Error desighning

Well log responses were inputs to the network and permeability values were the outputs. The optimal number of neurons of a single hidden layer network using trial and error method is shown in Table 1. The training result for the ANN on cross-validation data showed the lowest MSE when the number of hidden neurons was 22. It has been found the 8–22–1 architecture was the best model in terms of MSE, which meant 8, 22, and 1 neurons in the input, hidden and output layers, respectively. More than a single hidden layer ANNs also were used throughout this stage, but the results was unsatisfied.

TABLE 1: ANN DESIGNING WITH DIFFERENT NEURON IN THEIR MIDDLE LAYER.

Number of ANN	Number of PEs in hidden layer	Cross-validation error (MSE)
1	6	0. 568
2	10	0. 362
3	12	0. 178
4	16	0. 0853
5	20	0.0769
6	22	0.0565
7	24	0.0894
8	26	0.0621
9	28	0.209

The performance efficiency of the network was evaluated using the measured and ANN estimated values. Table 2 reports the ANN performances in terms of: mean-squared error (MSE), normalized mean-squared error (NMSE), mean absolute error (MAE), minimum absolute error and maximum absolute error and the linear correlation coefficient (r) between core permeability and neural network output. The mathematical expressions of r, MAE, NMSE, and MSE are defined in Sahoo and Ray (1996) [21]. In brief, the ANN predictions are optimum if r, MAE, NMSE, and MSE are found to be close to 1, 0, 0, and 0, respectively. In the present study, MSE is used only for the estimation of network training performance, whereas r, MAE, NMSE are used to measure the prediction performance of ANN on the validation data set.

TABLE 2: PERFORMANCE OF ANN FOR TEST DATA SET THAT TRAINED BY TRAIL AND ERROR.

Performance	KH
MSE	0.254302137
NMSE	0.196715244
MAE	0.216216873
Min Abs Error	0.001150153
Max Abs Error	3.86565017
R	0.53858819

4.2. GA method

The number of hidden layers and the number of input and output neurons in GA-ANN were defined 1 (2, and 3), 22, and 1, respectively. It is found that, the best numbers of hidden layer to determine of the GA-ANN model topology were obtained two hidden layers. On the other hand in this case, the GA-ANN model with two hidden layers by optimizing the ANNs parameters according to the various numbers of hidden layers had the best fitness value. This means that the best ANN estimating model had a 8–16–6–1 architecture.

Table 3 summarizes a report orthe best fitness and the average fitness values for all data. Also, corresponding plots which resulted from this table are shown in Figures 6 & 7. For each of these plots, across all generations the minimum MSE, the generation of this minimum and the final MSE are displayed.

TABLE 3: TRAINING AND CROSS-VALIDATION ERROR OBTAINED FROM TRAINED NETWORK.

Optimization Summary	Best Fitness	Average Fitness
Generation #	14	9
Minimum MSE	0.01103432	0.013683575
Final MSE	0.01103432	0.0245063997

The fitness function is an important factor for the convergence and the stability of genetic algorithm. The collision avoidance and the shortest distance should be considered in path planning. Therefore, smallest fitness value is used to evaluate convergence behavior of the GA. Figure 6 demonstrated the best fitness value versus the number of generation.

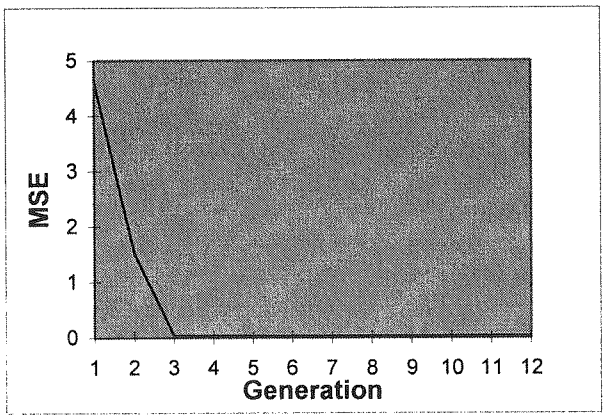

Figure 6: Best Fitness (MSE) versus Generation.

In Figure 7, the average fitness achieved during each generation of the optimization is illustrated. The average fitness is the average of the minimum MSE (cross validation MSE) taken across all of the networks within the corresponding generation.

TABLE 4: PERFORMANCE OF GA-ANN FOR TEST DATA SET.

Performance	**KH**
MSE	0.116714547
NMSE	0.038365831
MAE	0.047358659
Min Abs Error	0.000110548
Max Abs Error	1.055398535
R	0.999304048

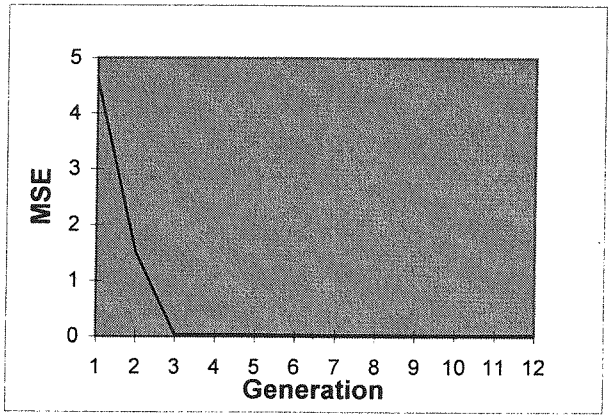

Figure 7: Average Fitness (MSE) versus Generation.

In this case, to evaluate of the hybrid model generalization on the chosen data set or test data points, the performance of each model was reported in Table 4. Figure 8 shows the plot of the network output and the desired network output for each test data set. In this figure, color and the corresponding network output was a dashed line of the same color.

Figure 8 shows the actual permeability values (KH) that were measured in the laboratory (and were never seen by the network during genetic training) in comparison with the network's estimation/prediction (KH output) for each sample.

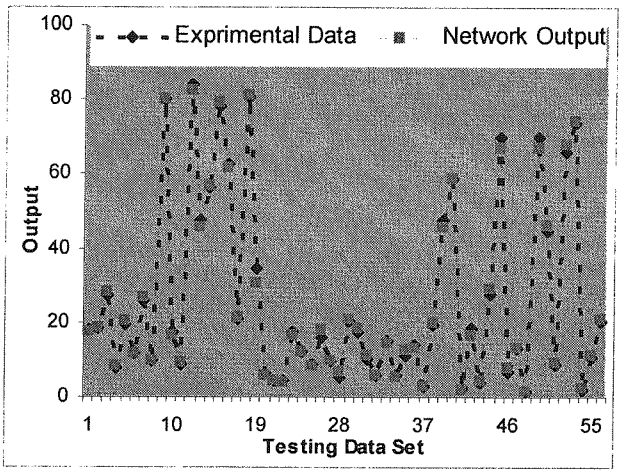

Figure 8: Desired Output and Actual Network Output.

Comparing the results presented in Table 2 and Table 4 reveals that the correlation coefficient between core permeability and ANN output is 0.85 while this parameter between core permeability and GA_ANN output is 0.99. Accordingly, genetically trained network is able to predict/estimate permeability comparable to that of actual core measurements. On the other hand, the capabilities of optimized neural networks in pattern recognition is established.

In Figure 9, the core permeability values versus GA-ANN predictions to verification data points are shown.

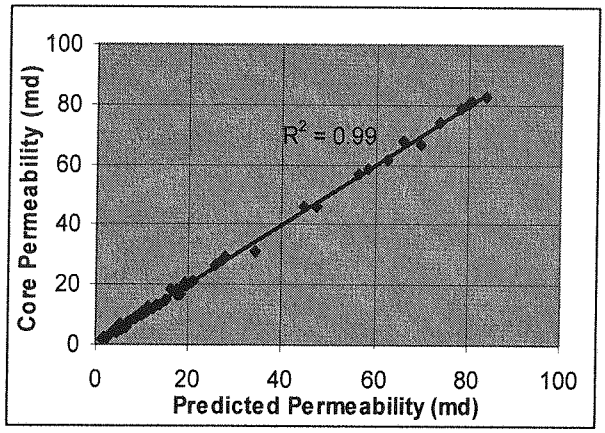

Figure 9: Core measurements versus network predictions for test data set.

This figure reveals that an acceptable agreement (linear correlation coefficient is 99%) between the predicted and experimental data can be achieved.

5. CONCLUSIONS

In this work, an attempt has been made to develop a methodology for designing of neural network architecture using genetic algorithm. GA is used to determine the number of neurons in the hidden layers, the momentum, and the learning rates in order to minimizing the time and effort required to find the optimal architecture and parameters of the back-propagation based ANN.

Comparing the prediction performance efficiency of the GA optimized ANN model with that of trial-and-error approach calibrated ANN model, shows that GA-ANNs significantly outperforms to trial-and-error calibrated ANNs. On the other hand, GA is found to be a good alternative over the trial-and-error approach to determine the optimal ANN architecture and internal parameters quickly and efficiently. The performance of the nets with respect to the predictions made on the test sets shows that the neural network model incorporating a GA was able to adequately estimate the permeability reservoir with high correlation coefficient.

Using this methodology researchers and engineers will be able to characterize reservoir heterogeneity using readily available geophysical well logs. It was shown that a carefully designed neural network is able to predict rock permeability with accuracies comparable to actual laboratory core measurements.

6. REFERENCES

1. J. Wiener, "Predict Permeability From Wireline Logs Using Neural Networks," Petroleum Engineering International, pp. 18–24, May. 1995.
2. S. Mohaghegh, R. Arefi, S. Ameri, and D. Rose, "Design and Development of an Artificial Neural Network for Estimation of Formation Permeability," SPE 28237, Petroleum Computer Conference, Dallas. 1994.
3. S. Mohaghegh, B. Balan and S. Ameri, "State-Of-The-Art in Permeability Determination From Well Log Data: Part 2-Verifiable, Accurate Permeability Predictions, the Touch-Stone of

All Models," SPE 30979, Eastern Regional Conference and Exhibition, Sept. pp. 17–21, West Virginia. 1995.
4. P.M. Wong, M. Jang, S. Cho and T.D. Gedeon, "Multiple permeability predictions using an observational learning algorithm," Computers & Geosciences, vol. 26, pp. 907–913. 2000.
5. K. Aminian, H.I. Bilgesu, S. Ameri and E. Gil, "Improving the simulation of waterflood performance with the use of neural networks," SPE 65630, Proceeding of SPE Eastern Regional Conference, October. 2000.
6. K. Aminian, B. Thomas, H.I. Bilgesu, S. Ameri and A. Oyerokun, "Permeability distribution prediction," SPE Paper, Proceeding of SPE Eastern Regional Conference, October. 2001.
7. R.B. Boozarjomehry, W.Y. Svrcek, "Automatic design of neural network structures," J. Computers and Chemical Engineering, vol. 25, pp. 1075–1088. 2001.
8. S. Mohaghegh, S. Ameri, K. Aminian, "A methodological approach for reservoir heterogeneity characterization using artificial neural networks," J. Pet. Sci. Eng., Vol. 16, pp. 263– 274. 1996.
9. M. Bean, C. Jutten, "Neural networks in geophysical applications," Geophysics, vol. 65, pp. 1032-1047. 2000.
10. S.P. Niculescu, "Artificial neural networks and genetic algorithms in QSAR," J. Molecular Structure (Theochem), vol. 622, pp. 71–83. 2003.
11. A. Saxena, A. Saad, "Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems," J. Applied Soft Computing, pp. 1568-4946. 2006.
12. T. Hegazy, O. Moselhi and P. Fazio, "Developing practical neural network applications using back-propagation," Microcomputers in Civil Engineering, vol. 9, pp. 145–59. 1994.
13. A.J.F. Rooij, L.C. Jain and R.P. Johnson, "Neural network training using genetic algorithms," World Scientific Publishing Co. Pvt. Ltd, Singapore. 1996.
14. E. Vonk, L.C. Jain and R.P. Johnson, "Automatic generation of neural network architecture using evolutionary computation," World Scientific Publishing Co. Pvt. Ltd, Singapore. 1997.
15. G. Miller, P. Todd and S. Hedge, "Designing neural networks using genetic algorithms," Proceeding of the Third International Joint Conference on Genetic Algorithms, pp. 379–384. 1989.
16. S.J. Marshall, R.F. Harrison, "Optimization and training of feedforward neural networks by Gas," Proceeding of IEE Second International Conference on Artificial Neural Networks, pp. 39–43. 1991.
17. S. Bornholdt, D. Graudenz, "General asymmetric neural networks and structure design by genetic algorithms," Neural Networks, vol. 5, pp. 327–34. 1992.
18. Y. Huanga, T. Gedeonb and P. Wongc, "An integrated neural-fuzzy-genetic-algorithm using hyper-surface membership functions to predict permeability in petroleum reservoirs," J. Pet. Sci. Eng, vol. 14, pp. 15-21. 2001.
19. C. Chena, L. Lina, "A committee machine with empirical formulas for permeability prediction," Computers & Geosciences, vol. 32, pp. 485–496. 2006.
20. G.H. Kim, J.E. Yoon, S.H. An, H.H. Cho and K.I. Kanga, "Neural network model incorporating a genetic algorithm in estimating construction costs," Building and Environment, vol. 39, pp. 1333 – 1340. 2004.
21. G.B. Sahoo, C. Ray, "Flow forecasting for a Hawaiian stream using rating curves and neural networks," J. Hydrol., vol. 317, pp. 63–80. 2006.