Transp Porous Med (2008) 71:273–288
DOI 10.1007/s11242-007-9125-4

Integration of genetic algorithm and a coactive neuro-fuzzy inference system for permeability prediction from well logs data

Mohsen Saemi · Morteza Ahmadi

Received: 15 September 2006 / Accepted: 13 March 2007 / Published online: 14 April 2007

Abstract Permeability is one of the reservoir fundamental properties, which relate to the amount of fluid contained in a reservoir and its ability to flow. These properties have a significant impact on petroleum fields operations and reservoir management. The most reliable data of local permeability are taken from laboratory analysis of cores. Extensive coring is very expensive and this expense becomes reasonable in very limited cases. Thus, the proper determination of the permeability is of paramount importance because it affects the economy of the whole venture of development and operation of a field. In this study, we introduce a new hybrid network based on Coactive Neuro-Fuzzy Inference System (CANFIS). CANFIS is a dependable and robust network that developed to identify a non-linear relationship and mapping between petrophysical data and core samples. Then to improve the system performance, genetic algorithm (GA) was integrated in order to search of optimal network parameters and decrease of noisy data in training samples. An Iranian offshore gas field is located in the Persian Gulf, has been selected as the study area in this paper. Well log data are available on substantial number of wells. Core samples are also available from a few wells. It was shown that the new proposed strategy is an effective method in predicting permeability from well logs.

Keywords Permeability · Coactive neuro-fuzzy inference systems · Genetic algorithm · Petrophysical data · Artificial intelligence

1 Introduction

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. This physical character is usually obtained from well

M. Saemi (✉) · M. Ahmadi
Mining Engineering Department, Faculty of Engineering, Tarbiat Modares University, Tehran, Iran
e-mail: mo_saemi@yahoo.com

test data interpretation, core data, or well log data. The first method provides a reliable in situ measure of average permeability and the second method measure core permeability in a laboratory. Both well testing and coring operations are very expensive and consequently perform only in some wells of the field.

Well log data is another tool, which can be used to estimate permeability if limited core data are also available. In most cases well logging is performed almost in all wells in a field. Otherwise, many correlations have been widely studied to estimate permeability using well log data with the aid of available core-measured permeability. Permeability is often determined from log-derived parameters such as porosity. Based on the Kozeny–Carmen model (Carman 1937), several conventional empirical formulas have been proposed (Wyllie and Rose 1950). However, these correlation formulas are not universally applicable. Their suitability should be checked against field studies.

Soft computing is an ensemble of various intelligent computing methodologies or systems which include neurocomputing, fuzzy logic, and evolutionary computing. Unlike the conventional or hard computing, it is tolerant of imprecision, uncertainty, and partial truth. It is also, effective and inexpensive method. In reservoir characterization, these intelligent techniques can be used for uncertainty analysis, risk assessment, data fusion, and data mining which are applicable to feature extraction from seismic attributes, well logging, reservoir mapping and engineering (Nikravesh 2004).

In recent years, artificial intelligent (AI) have been increasingly applied to predict reservoir properties using well log data. Comparison investigations (Nikravesh 2004; Xie et al. 2005) demonstrated that artificial intelligent have a great potential for generating more-accurate predictions in reservoir studies.

Genetic algorithms (GAs), fuzzy logic (FL), and artificial neural networks (ANNs) are frequently used artificial intelligence (AI) techniques. Since these three methods are complementary rather than competitive, many researchers have hybridized GAs, FL, and ANNs to develop a better performance model. GAs perform a stochastic searching process based on the mechanisms of natural selection and natural genetics (Holland 1975); FL simulates human decision-making process in a high-level manner; and ANNs model the way the brain performs functions.

In the last few years, several articles (Ghezelayagh and Lee 1999; Jagielska et al. 1999; Gorzalczany and Gradzki 2000) have been devoted to the study of fusing FL, ANNs, and GAs to derive a better model performance than those using a single conventional method. In this work, we fuse GAs, FL, and ANNs to develop a new and powerful tool in order to determine the reservoir permeability distribution model from petrophysical data. CANFIS is one of the most powerful tools that processes the advantages of both artificial neural network and fuzzy logic. CANFIS model can avoid the subjectivity to the definitions of membership function and rules in fuzzy logic. GA is an effective approach to conquer the drawbacks of CANFIS. Therefore, the CANFIS employs GA to simultaneously search for feature selection, and to find optimum parameters of this model. It is the goal of this study to extract formation permeability values from information provided by well logs. It will be shown that this new methodology is able to generate permeability distribution model with accuracies comparable to actual measurements.

2 Theoretical background

Permeability measurements have a significant impact upon field operations and reservoir management. The common correlation used to predict permeability is obtained from the graph of the logarithm of core permeability vs. core porosity. However, sometimes the correlation coefficients are not good, often less than 0.6. This correlation assumes that the permeability is only a function of porosity. Investigations (Timur 1968) have shown that permeability is not only a function of porosity, but also of true resistivity, irreducible water saturation, hydrocarbon density, and rock type. Prediction of permeability using those variables without actual core measurements has been a fundamental problem for petroleum engineers and geoscientists.

In the last few years, tremendous effort has been expended in the generation of a large variety of approaches to estimating permeability utilizing well logs. Intelligent techniques such as ANN, FL, and GA have shown great potential for this purpose. From the viewpoint of human-based problem solving, each method lacks benefits of the other techniques. The hybridization of GAs, FL, and ANNs provides a promising direction to solve real-world problems, since their hybridization quite resembles the way a human solves problems (Tang et al. 2005).

2.1 Artificial neural network

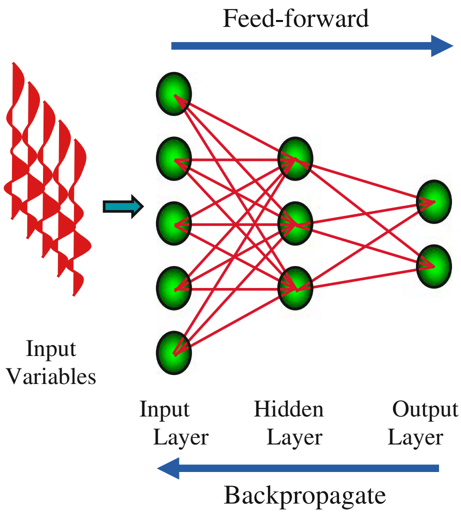
An Artificial neural network (ANN) is an information processing technique developed by inspiration by the way biological nervous systems, such as the brain, process information. ANNs are composed of a large number of highly interconnected processing elements, or neurons, usually arranged in layers. These layers generally include an input layer, a number of hidden layers, and an output layer. Signals that are generated from the input propagate through the network on a layer-by-layer basis in the forward direction. Neurons in hidden layers are used to find associations within the input data and extract patterns than can provide meaningful outputs. A neural network system uses the human-like technique of learning by example to resolve problems. Just as in biological systems, learning involves adjustments to the synaptic connections that exist between the neurons. The output of each neuron, responding to a particular combination of inputs, has an influence (or weight) on the overall output. Weighting is controlled by the level of activation of each neuron, and the strength of connection between individual neurons. Patterns of activation and interconnection are adjusted to achieve the desired output from the training data. Corrections are based on the difference between actual and desired output, which is computed for each training cycle. If average error is within a prescribed tolerance the training is stopped, the weights are locked in and the network is ready to use (Bishop 1995; Javadi et al. 2005).

The network architecture or topology (including: number of neurons in hidden layers, network connections, initial weight assignments, and activation functions) plays a very important role in the performance of the ANN, and usually depends on the problem at hand. Figure 1 shows a Multi-Layer-Perceptron network and its constituents that is one of the simplest types of neural networks.

Generally, there are two main types of learning: supervised and unsupervised. In a supervised learning, the external environment also provides a desired output for each one of the training vectors whereas in unsupervised learning, the external environment does not provide the desired network output.

In most cases, setting the correct topology is a heuristic model selection. Whereas the number of input and output layer nodes is generally suggested by the dimensions of the input and the output spaces, selecting the network complexity or regularization is yet again very important. Too many parameters lead to poor generalization (over fitting), and too few parameters result in inadequate learning (under fitting) (Duda et al. 2001). Moreover Heckerling et al. (2004), indicated that to constrain the magnitude of the connection weight updates during network training, ANNs should be trained with a *learning rate* and to modify weight updates by a proportion of the updates from the previous training case, *momentum factor* must be used.

Fig. 1 A simple Multi-Layer-Perceptron (MLP)



2.2 Fuzzy logic

Fuzzy logic was first developed by Zadeh in 1960s for representing uncertain and imprecise information. It provides approximate but effective descriptions for highly complex, ill-defined, or difficult to analyze mathematical systems. Fuzzy logic is considered to be appropriate to deal with the nature of uncertainty in system and human error, which are not included in current reliability theories. Unlike classical logic which is based on crisp sets of "true and false," fuzzy logic views problems as a degree of "truth," or "fuzzy sets of true and false" (Nikravesh 2004).

Fuzzy logic starts with the concept of a *fuzzy set*. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. *Fuzzy inference* is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves all of the pieces that are described in the following:

- *Membership function (MF)*, is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, a fancy name for a simple concept.
- *If-then rules*, If-then rule statements are used to formulate the conditional statements that comprise fuzzy logic. A single fuzzy if-then rule assumes the form if x is A then y is B where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively. The if-part of the rule "x is A" is called the antecedent or premise, while the then-part of the rule "y is B" is called the consequent or conclusion.
- *Fuzzy logic operators*, If there are multiple parts to the antecedent, apply fuzzy logic operators (AND, OR, and NOT) and resolve the antecedent to a single number between 0 and 1. This is the degree of support for the rule. Using AND, OR, and NOT functions, we can resolve any construction using fuzzy sets and the fuzzy logical operation.

A general Fuzzy Inference System (FIS) contains four major components: fuzzifier, fuzzy if-then rules base, inference engine, and defuzzifier. There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic Toolbox: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined. Prototype publications for interested researchers to descriptions of these two types of fuzzy inference systems are (Mamdani and Assilian 1975; Kaufmann and Gupta 1985; Jang and Sun 1997).

Despite the advantages of FIS over traditional approaches such as deterministic mathematical models or probability methods, there remain a number of drawbacks such as configuration of distributions of MFs, determination of composition operator, and acquisition of appropriate fuzzy rules in applications. Difficulty in determining those parameters increases human inclination to use of hybrid systems such as neuro-fuzzy systems or genetic fuzzy systems in order to improve the modeling performance in various engineering disciplines.

2.3 Genetic algorithms (GAs)

Many investigators (Golderg 1989; De Jong 1992; Michalewicz 1992) reveal that genetic algorithms have proven to be useful in optimization of such non-convex, multi-modal, and noisy problems because of their ability to efficiently use various information to obtain new solutions with enhanced performance and a global nature of search supported there. Genetic algorithms are also theoretically and empirically proven to support robust search in complex search spaces (Oh et al. 2003). The GA simulates the mechanics of population genetics by maintaining a population of knowledge structure, which is made to evolve the problems must be represented in a suitable form to be handled by the GA. The GA often works with a form of binary coding. If the problems are coded as chromosomes, the population is initialized. Each chromosome within the population is gradually evolved by biological operations. Once the population size is chosen, the initial population is randomly generated. After the initialization step, each chromosome is evaluated by the fitness function. According to the value of the fitness function, the chromosomes associated with the fittest individuals will be reproduced more often than those associated unfit individuals (Bauer 1994).

The GA works with three operators that are iteratively used. The *selection* operator determines which individuals may survive. The *crossover* operator allows the search to fan out in diverse directions looking for attractive solutions and permits chromosomal material from different parents to be combined in a single child. In addition, the *mutation* operator arbitrarily alters one or more components of a selected chromosome. It provides the means for introducing new information into the population. Finally, the GA tends to converge on optimal or near-optimal solutions (Hertz and Kobler 2000).

The GA is usually employed to improve the performance of artificial intelligence techniques. For ANN, the GA was applied to the selection of neural network topology including optimizing a relevant feature subset, determining the optimal number of hidden layers and processing elements. In addition, some researchers searched the connection weights of ANN using the GA instead of local search algorithms including a gradient descent algorithm. They (Gupta and Sexton 1999; Kim and Han 2000; Sexton et al. 1998) suggested that global search techniques including the GA might prevent ANN from falling into a local optimum.

3 Methodology

From the viewpoint of human-based problem solving, each method lacks benefits of the other techniques. The hybridization of GAs, FL, and ANNs provides a promising direction to solve real-world problems, since their hybridization quite resembles the way a human solves problems.

In the triumvirate of GAs, FL, and ANNs, GAs are frequently used to search for the optimal interconnections, synaptic weights, architectures of ANNs, or to search for the distribution of membership functions (MFs); FL to define system inputs/outputs by fuzzy sets or to make inference via the inference principle; and ANNs to tune the shapes of MFs or to extract fuzzy rules from training patterns (Tang et al. 2005).

The objective of the present research work is to fuse GAs, FL, and ANNs to develop a permeability estimator model based on neuro-fuzzy system that simultaneously searches for optimal input selection, and to adjust the network parameters.

In general, the main reason GA_CANFIS outperforms FL and ANNs is that the model automatically identifies all parameters required in FL and ANNs. This advantage overcomes the difficulties faced in applying FL and ANNs. Also, this characteristic enhances the efficiency for problem solving as well as promotes the applicability of the proposed model. Recently used AI techniques such as FL, ANNs, and GA require a lot of efforts in trial-and-error experiment, questionnaire preparation, survey and analysis, interviews with experts, development of similarity index, development of objective objection, etc.

These methods may consume 1 month or even half a year on developing solutions. On the contrary, applying GA_CANFIS for problem solving requires no effort in questionnaire preparation, questionnaire survey, etc. Both prediction accuracy and time requirement for solution development would be much improved by the proposed method.

In this study, the process of network designing by NeuroSolutions was managed for Excel Release 5.00 software. This software which incorporates various types of ANN and GA, produced by NeuroDimension, Inc., and used to obtain the optimal feature and network parameters in the permeability estimation model.

3.1 Coactive neuro-fuzzy inference system (CANFIS)

Neuro-fuzzy hybrid systems combine the advantages of fuzzy systems, which deal with explicit knowledge which can be explained and understood, and neural networks which deal with implicit knowledge which can be acquired by learning. Neural network learning provides a good way to adjust the expert's knowledge and automatically generate additional fuzzy rules and membership functions, to meet certain specifications and reduce design time and costs.

A very promising paradigm in intelligent techniques is constituted by the neuro-fuzzy approach, in which fuzzy logic and neural networks are combined. The CANFIS model integrates adaptable fuzzy inputs with a modular neural network to rapidly and accurately approximate complex functions. Fuzzy inference systems are also valuable, as they combine the explanatory nature of rules (MFs) with the power of neural networks. These kinds of networks solve problems more efficiently than neural networks when the underlying function to model is highly variable or locally extreme (Camps-Valls et al. 2004).

The fundamental component of CANFIS is a fuzzy axon, which applies membership functions to the inputs. The output of a fuzzy axon is computed using the following formula:

$$f_j(x, w) = \min \forall_i (\mathrm{MF}(x_i, w_{ij})), \tag{1}$$

where i = input index, j = output index, x_i = input i, w_{ij} = weights (MF parameters) corresponding to the jth MF of input i and MF = membership function of the particular subclass of the fuzzy axon.

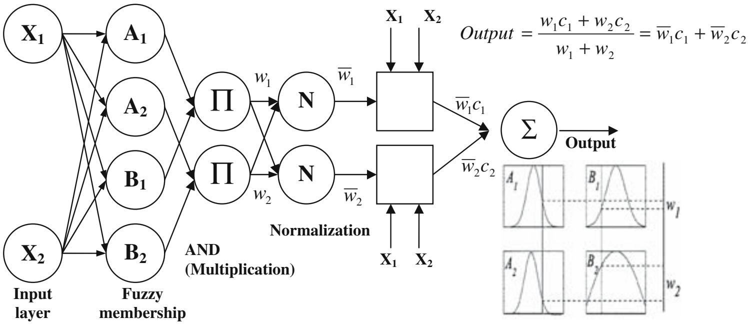


Fig. 2 A prototype two-input one-output CANFIS network and output calculation

This system can be viewed as a special three-layer feed forward neural network. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables. The CANFIS architecture used in this study is shown in Fig. 2.

3.2 CANFIS architecture

Consider a CANFIS structure with n inputs and one output. For model initialize, suppose a common rule set with n inputs and m IF-THEN rules as follows (Burrows et al. 1998):

Rule 1: If z_1 is A_{11} and z_2 is A_{12} and ... and z_n is A_{1n} then $u_1 = p_{11}z_1 + p_{12}z_2 + \cdots + p_{1n}z_n + q_1$
Rule 2: If z_1 is A_{21} and z_2 is A_{22} and ... and z_n is A_{2n} then $u_2 = p_{21}z_1 + p_{22}z_2 + \cdots + p_{2n}z_n + q_2$
$\vdots$
Rule m: If z_1 is A_{m1} and z_2 is A_{m2} and ... and z_n is A_{mn} then $u_m = p_{m1}z_1 + p_{m2}z_2 + \cdots + p_{mn}z_n + q_m$

The corresponding CANFIS structure is illustrated in Fig. 2. All layers in CANFIS structure are either adaptive or fixed. The function of each layer is described as follows:

Layer 1 (Premise Parameters): Every node in this layer is a complex-valued membership function ($\mu_{A_{ij}}$) with a node function:

$$O_{1,ij} = \left|\mu_{A_{ij}}(z_i)\right| \angle \mu_{A_{ij}}(z_i) \text{ for } (1 \le i \le n, 1 \le j \le m). \tag{2}$$

Each node in layer 1 is the membership grade of a fuzzy set (A_{ij}) and specifies the degree to which the given input belongs to one of the fuzzy sets.

Layer 2 (Firing Strength): Every node in this layer is product of all the incoming signals. This layer receives input in the form of the product of all the output pairs from the first layer:

$$O_{2,j} = w_j = \mu_{A_{i1}}(z_1)\mu_{A_{i2}}(z_2), \ldots, \mu_{A_{i1n}}(z_n) \quad \text{for } (1 \le i \le m). \tag{3}$$

Layer 3 (Normalized Firing Strength): Every node in this layer calculates rational firing strength:

$$O_{3,j} = \bar{w_j} = \frac{w_j}{\sum_{j=1}^{m} wj} \quad \text{for } (1 \le j \le m). \tag{4}$$

Layer 4 (Consequence Parameters): Every node in this layer is multiplication of Normalized Firing Strength from the third layer and output of neural network:

$$O_{4,j} = \bar{w}_j u_j = \bar{w}_j (P_{J1} Z_1 + P_{J2} Z_2 + \cdots P_{Jn} Z_{2n} + q_j) \quad \text{for } (1 \le j \le m). \tag{5}$$

Layer 5 (Overall Output): The node in this layer computes overall output of CANFIS network:

$$O_{5,1} = \sum_j \bar{w}_j u_j. \tag{6}$$

Basically, two membership function types can be used (Gaussian or generalized bell). The bell fuzzy axon used in this study is a type of fuzzy axon that uses a bell-shaped curve as its membership function. Each membership function takes three parameters stored in the weight vector of the bell fuzzy axon (Eq. 7):

$$\mathrm{MF}(x, w) = \frac{1}{1 + \left| \frac{x - w_2}{w_0} \right|^{2w_1}}, \tag{7}$$

where x = input and w = weight of the bell fuzzy axon.

Fuzzy axons are valuable because their MF can be modified through backpropagation during network training to expedite the convergence. A second advantage is that fuzzy synapses help in characterizing inputs that are not easily discretized. The powerful capability of CANFIS stems from the pattern-dependent weights between the consequent layer and the fuzzy association layer.

The second major component of CANFIS is a modular network that applies functional rules to the inputs. The number of modular networks matches the number of network outputs and processing elements in each network corresponding to the number of MFs.

Two fuzzy structures are mainly used: the Tsukamoto model and the Sugeno (TSK) model. Finally, a combiner is used to apply the MF outputs to the modular network outputs. The combined outputs are then channeled through a final output layer, and the error is backpropagated to both the MF and the modular network (Roger et al. 1997).

3.3 Genetic optimization

In order to improve the learning of the *CANFIS*, quicker training and enhance its performance, we use genetic algorithms to search for the best number of MF for each input, and optimization of control parameters such as learning rate, and momentum coefficient. This approach also is useful to select the most relevant features of the training data which can produce a smaller and less complicated network, with the ability to generalize on freshly presented data, due to the removal of redundant variables.

The GA combines selection, crossover, and mutation operators with the goal of finding the best solution to a problem by searching until the specified criterion is met. The solution to a problem is called a chromosome, which is composed of a collection of genes. In hybrid neuro-fuzzy-genetic applications, genes are the *CANFIS* parameters to be optimized. The GA creates an initial population and then evaluates this population by training a network for each chromosome. It then evolves the population through multiple generations in the search for the best network parameters.

GAs cause the initial population to evolve towards a population that is expected to contain the best solution (Yanga et al. 2006). We use the following reproduction evaluation cycle for each iteration-referred to as a generation. Chromosomes (individuals) from the current

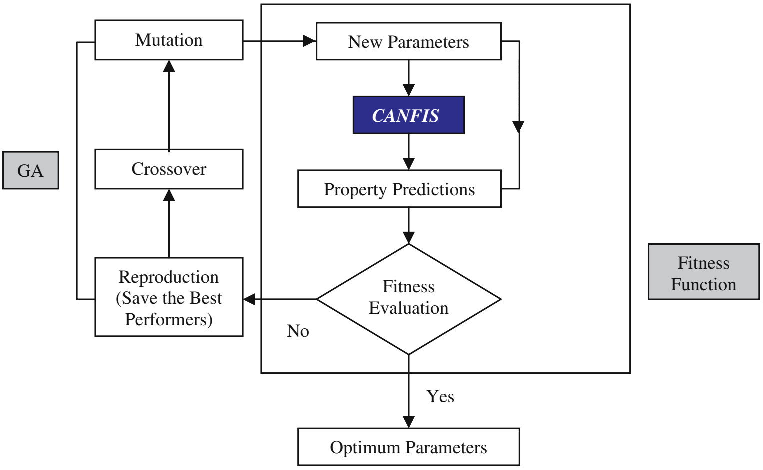


Fig. 3 The CANFIS/genetic algorithm cycle for optimization

population are selected with a given probability; and copies of these chromosomes (individuals) are created. The selection of chromosomes is based on their fitness relative to the current population; that is, the stronger chromosomes will have a higher probability of being copied. The fitness is a function of the *CANFIS* model's response. Selected chromosomes are subjected to mutation and to crossover. Figure 3 shows the CANFIS/genetic algorithm cycle for search of optimum parameters of the model.

These mathematical chromosomes could be operated upon by quasi-genetic processes of crossing over and mutation. To implement crossovers, chromosomes were randomly paired, and segments of paired chromosomes between two randomly determined breakpoints were swapped. Crossovers could be implemented either across genes, so that gene boundaries might potentially be breached by the exchange of genetic material; or within genes, so that gene boundaries would be preserved. Inversions could also be modeled, so that exchanged genetic material could be inverted before becoming incorporated into the recipient chromosome. Mutations were implemented by flipping a bit at a binary locus, so that a "0" bit was converted to a "1," or a "1" bit was converted to a "0."

In this paper, for the optimization of the CANFIS model, GA used the serial method of binary type, roulette-wheel in the selection operator, tow-point crossover in the crossover operator, and boundary in the mutation operator.

Automatically determination of the chromosomes length used to optimal search is one of the most important capabilities of the NeuroSolution software. Thus, all the chromosomes were automatically set in this software so that they consisted of the number of input neurons and membership functions, learning rate, and momentum. NeuroSolution also automatically produced their initial values.

4 Application

The prototype field is located in the Persian Gulf, Iranian offshore. With the data acquired, the reservoir 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

Table 1 Ranges of the used database for all the petrophysical groups

	Well 1	Well 2	Well 3	Well 4
Saturation	0.036–0.994	0.005–0.891	0.010–0.99	0.015–0.994
Gamma ray (GAPI)	20.32–57.69	7.142–39.01	4.395–58.79	2.747–88.46
Neutron log	−0.01–0.33	0.004–0.348	−13.5–1.48	−0.01–0.280
Bulk density (gr/cm^3)	2.131–2.972	2.191–2.912	2.180–2.978	2.163–2.945
PEF (B/cm^3)	1.978–19.23	8.743–19.89	0.43–19.4	16.48–20.10
DT (US/F)	44.3–75.16	49.34–80.65	44.3–81.20	43.84–74.61
Resistivity (MH/M)	0.005–0.489	0.002–0.42	0.019–0.486	0.002–0.32
Depth (m)	2,860–3,265	2,900–3,105	2,650–3,090	2,790–3,205
Permeability (md)	2.3–86.78	0.91–103.5	0.5–109.84	1.36–62.37

exploration and appraisal wells. The gas-bearing reservoirs of the field belong to the Kangan formations of Triassic age. 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.

4.1 Data collection

During the few last years, primary geological investigations demonstrate the degree of heterogeneity of this formation and show the chaotic status of the information that we were faced with. The total number of wells drilled at the time of this study added up to 43 wells. All the above wells have been logged except three wells, which do not have sufficient number of logs needed for petrophysical evaluation. Therefore, the petrophysical data of almost all of the 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. Among these vertical wells of the mentioned field, only six wells have been cored in the reservoir layer. Therefore, it should be noted that cores data are often only available from few wells in a 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.

All, 625 available data points which collected from four wells, were split randomly into three separate groups: training, testing, and validation. Table 1 shows the ranges of the used database for all petrophysical groups. The training set consists of a set of examples used only for learning (i.e., to fit the weights of the network). The validation data 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 was stopped, at each run, once the error performance of the network began to deteriorate, based on the training and validation set errors, when the training error fails to improve over a given number of epochs and the validation error starts to increase.

The test set is a set of examples used only to assess the generalization performance of a trained neural network, and used to evaluate the accuracy of the newly trained network by providing the network a set of data it has never seen. Typically 80% of the data is used for training and validation purposes. The other 20% of the data is categorized as verification.

Table 2 GA parameters setup

Levels	1	2	3
Crossover rate	0.30	0.60	0.90
Mutation rate	0.02	0.10	0.18
Population size	20	40	60

4.2 Simulation

The *CANFIS* architecture used in this study, in which values of geophysical well logs mentioned above are used as inputs, and core permeability data are used as output in order to create of the model that is composed of three layers. Each of the features was normalized, dividing each row by its absolute maximum value and keeping it within ±1 for better speed and success of the network training. A second scheme of normalization with zero mean and a standard deviation of 1 for each feature set was attempted. Another normalization scheme was also examined by making the features zero mean and then normalizing by the absolute maximum value. The results comparing the effectiveness of these normalization schemes indicated that the use of absolute maximum in magnitude normalization scheme exploits the large peaks present in the fault signal lowering the normal rotational components. This changes the relative statistics of the signals with and without faults, leading to better predictions success.

In this work, as the membership function, the bell-shaped curve and as the fuzzy model, TSK were used. The bell function is a little more flexible than the gaussian to adjust of MF parameters and the TSK fuzzy model is generally more popular than the Tsukamoto fuzzy model because of better performance in simulation. Training performed with the batch learning, momentum learning algorithm was used and axon was chosen as the transfer function. For small to medium-sized data sets, the number of membership functions assigned to each network input, will usually be between 2 and 4. These membership functions are then combined together to perform the inferencing operation. For this purpose, GA is used in order to optimal determine of MFs for each input.

CANFIS input variables and training parameters were represented by haploid chromosomes consisting of “genes” of binary numbers. Each chromosome had four genes: the first gene represented the number of neurons in the first layer of the network (feature selection), ranging from 0 to 8 nodes. The second and third genes were used to optimize the learning rate and momentum of the network being trained. These parameters have a lower bound of zero and an upper bound of one. Finally, the fourth gene was applied to find the number of MFs for each input variable, ranging from 2 to 5.

Because the GA solution may be sensitive to its parameters setup, 27 experiments were performed to represent the combinations of different levels for crossover rate, mutation rate, and population size, as shown in Table 2. The values in Table 2 are based on computational experiences and a literature review.

The GAs consistently reported the same quality score for all of the 28 experiments. Since the present problem is not sensitive to its GA parameters design, we choose the middle level from each parameter as the final GA parameters. They are 0.6 for crossover rate, 0.10 for mutation rate, and 40 for population size.

Thus the genetic algorithm was started with 40 randomly generated chromosomes, with gene structures as described above. The genes were decoded, and networks with architecture and learning parameters represented by the decoded genes were trained on the training cohort.

After a new generation of offspring is obtained, the fitness of all chromosomes (both parents and offspring) is evaluated, and the ones with the highest fitness are carried to the next generation for the next genetic cycle. Phenotypic fitness of each network was measured in one ways: as its mean-square error (MSE), which is the squared difference between network output and permeability status averaged across all cases in the cohort. Networks with lower MSE, which implied greater accuracy, were considered to be more fit.

The fitness of every chromosome was evaluated by measuring the MSE, which is the estimated result on a cross-validation data set as defined in Eq. 8:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} (O_i - T_i)^2}{n}, \tag{8}$$

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.

Table 3 summarizes the report of the best fitness and the average fitness values for training data. It also displays the minimum and the final MSEs, and the generation of the minimum MSE across all the generations.

Corresponding plot which resulted from above table are shown in Fig. 4. Figure 4 shows the most fit member at each generation.

Table 3 Optimization summary of the CANFIS training by using GA

Optimization	Best fitness	Average fitness
Generation	33	33
Minimum MSE	0.001913	0.001951
Final MSE	0.001913	0.003102

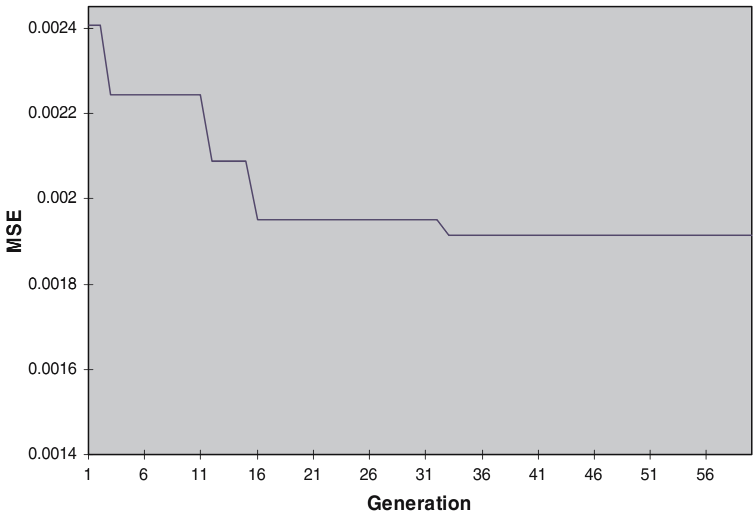


Fig. 4 Best fitness (MSE) vs. generation cycles

5 Results and discussion

According to the strategy based on the rank of their MSE, 40 networks were probabilistically propagated to the subsequent generation proportionate to their fitness. The chromosomes of propagated networks were then modified by crossing over and mutation, and the modified chromosomes were decoded to provide new parameters for the next round of network evolution. This process of phenotypic fitness measurement, selection, crossover recombination, and mutation was iterated through 60 generations; and the network with the lowest error in the 33 generation was designated as the optimal evolved network.

Results using a baseline strategy of recombinant crossovers across gene boundaries, mutation at the gene level, a mutation rate of 0.1, and selection according to rank of MSE, the optimal evolved network had six nodes in the first layer, a learning rate of 0.5 and a momentum of 0.5. The MSE of this optimal network was 0.001913 in the training cohort. Although the optimal network was not represented by any of the network chromosomes in the initial (0th) generation, by the 33th generation it was represented by 18 of the 40 network chromosomes.

5.1 Evaluation of the model performance

In order to test the performance of genetically optimized *CANFIS* after training, the test data point was presented to the network. Each predicted value was compared against the actual observed value to measure of network performance. For this purpose, another group of 104 random input–output data points (20% of all data) that described in Sect. 4.1 was used. Then normalized mean square error (NMSE), mean square error (MSE) and mean average error (MAE) are computed. On the lower values of NMSE and MAE, the network predicts the obesity more truly. The correlation coefficient r gives information about the training of network, having a value in the between $(-1, 1)$. If the correlation coefficient is close to (1), it shows how much the learning is successful. MSE is used to determine how much the network has reached to desired output values. The superiority of the estimation and generalization ability of the proposed GA-CANFIS hybrid network is seen in Table 4.

In addition, the result of final model predictions compared to the actual laboratory measurements and the results are shown in Fig. 5. On the other hand, Fig. 5 shows the experimental and corresponding simulative values that were compared only for unseen data set (test data set).

Figure 5A shows the plot of the network output and the desired network output for each test data set. In this figure, each desired output or core permeability plotted as a solid line and the corresponding network output was a dashed line. Figure 5B reveals that an acceptable agreement (linear correlation coefficient is 99.8%) between the predicted and experimental data can be achieved.

As can be seen from Table 4 and Fig. 5, the predictive ability of the GA-CANFIS model is generally satisfactory. The prediction capability of this hybrid network is found to be acceptable for the locations where well logs are available and core samples are unavailable.

Table 4 Performance of GA_CANFIS for test data set

Factor	Performance
MSE	2.38249726
NMSE	0.006834609
MAE	0.48467543
r	0.99873113

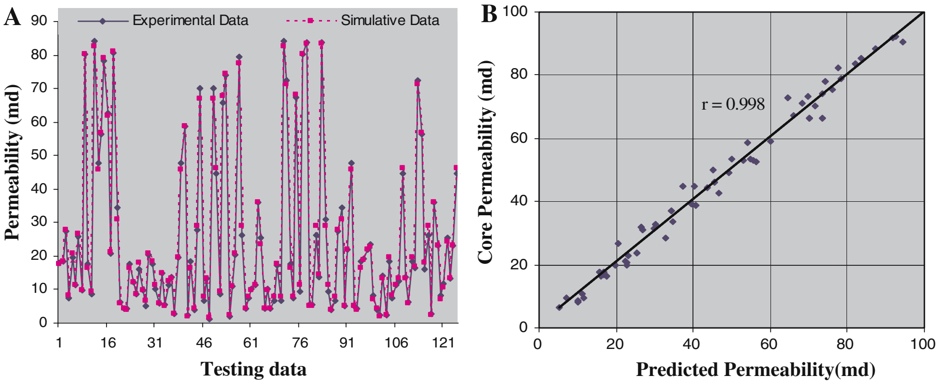


Fig. 5 Core measurements vs. network predictions for test data set

Feature selection is one of the most important our purposes to use the GA to select a subset of features that will be used to train the networks. The GAs consistently selected approximately six features from all eight input variable. The input variables, which were selected by the genetic algorithm, are gamma-ray, water saturation, density, sonic porosity, photoelectric factor, and resistivity log. Therefore, it is found that neutron porosity and depth logs have the lowest impact upon ability of the fluid conductivity in reservoir formations.

In general, this work has many advantages and useful effects in petroleum industry:

- Permeability of the formation is usually determined from the cores and/or well tests. It is worth noting that the cores and well test data are often only available from few wells in a reservoir, because both are very expensive. Therefore, evaluation of permeability from the well log data, available from the majority of the wells, represents significant technical as well as economic advantages. In this study, we proposed an inexpensive and fast method for the prediction of permeability from petrophysical data.
- Through this methodology, the researchers and engineers will be able to characterize reservoir heterogeneity using readily available geophysical well logs. It was shown that a carefully designed CANFIS is able to predict rock permeability with accuracies comparable to the actual laboratory core measurements for the locations in which core samples are unavailable.
- Despite the wide range of applications and flexibility of the Neuro-Fuzzy systems, there is still no general framework or procedure through which the appropriate network for a specific task can be designed. Design and optimization of these networks is still highly dependent upon the designer's experience. This is an obvious barrier to their wider applications. To mitigate this problem, in the present study, a new method was used based on genetic algorithm (GA).
- From reservoir engineering, reservoir management, and enhanced recovery design point of view, knowledge of rock permeability and its spatial distribution throughout the reservoir is of utmost importance. Thus, in the plan of field development and exploration investigations, GA_CANFIS as a powerful tool can be strongly effective to determine optimal locations of the next drillings.
- The novelty of this research is the use of GAs to simultaneously search for all the parameters required in a CANFIS.

6 Conclusion

In this work, an attempt has been made to develop a hybrid methodology for permeability estimation in a reservoir with high degree of heterogeneity. Coactive Neuro-fuzzy modeling was proposed as a dependable and robust method developed to identify a non-linear relationship and mapping between the petrophysical data and core samples. It has been shown that using of GA is a very useful technique for auto-tuning of the CANFIS parameters and to select of an optimal feature set.

The performance of the optimized model with respect to the predictions made on the test data set is shown that this model will be able to adequately estimate the permeability reservoir with high correlation coefficient. It was shown that using this methodology researchers and engineers will be able to predict rock permeability with accuracies comparable to actual laboratory core measurements for the locations that core samples are unavailable.

References

Bauer, R.J.: Genetic Algorithms and Investment Strategies. Wiley, New York (1994)

Bishop, C.: Neural Network for Pattern Recognition. Clarendon Press, Oxford (1995)

Burrows, W.R., Walmsley, J., Montpetit, J., Faucher M.: CART-NEURO-FUZZY statistical data modeling, Part 2: Results. In: Preprints, 14th Conference on Probability and Statistics in the Atmospheric Sciences, Phoenix, AZ, pp. 160–167. American Meteorological Society, Boston, MA (1998)

Camps-Valls, G., Chova, L.G., Maravilla, J.C., Guerrero, J.D., Olivas, E.S., Chorda, L.A., Moreno, J.: Robust support vector method for hyperspectral data classification and knowledge discovery. Trans. Geosci. Rem. Sens. **42**(7), 1530–1542 (2004)

Carman, P.C.: Fluid flow through granular beds. Tran. Inst. Chem. Eng. **15**, 150–166 (1937)

De Jong, K.A.D.: Are genetic algorithms function optimizers? In: Manner, R., Manderick, B. (eds.) Parallel Problem Solving from Nature 2. Elsevier, Amsterdam (1992)

Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience Publications, New York (2001)

Ghezelayagh, H., Lee, K.Y.: Training neuro-fuzzy boiler identifier with genetic algorithm and error back-propagation. IEEE Pow. Eng. Soc. Summer Meeting **2**, 978–982 (1999)

Golderg, D.E.: Genetic Algorithm in Search, Optimization & Machine Learning. Addison-Wesley, Reading, MA (1989)

Gorzalczany, M.B., Gradzki, P.A.: Neuro-fuzzy-genetic classifier for technical applications. Proc. IEEE Int. Conf. Ind. Technol. **1**, 503–508 (2000)

Gupta, J.N.D., Sexton, R.S.: Comparing backpropagation with a genetic algorithm for neural network training. Omega **27**(6), 679–684 (1999)

Heckerling, P.S., Gerber, B.S., Tape, T.G., Wigton, R.S.: Use of genetic algorithms for neural networks to predict community-acquired pneumonia. Art. Int. Med. **30**, 71–84 (2004)

Hertz, A., Kobler, D.: A framework for the description of evolutionary algorithms. J. Oper. Res. **126**(1), 1–12 (2000)

Holland, J.H.: Adaptation in Neural and Artificial Systems. The University of Michigan Press, Ann Arbor, Michigan (1975)

Jagielska, I., Matthews, C., Whitfort, T.: An investigation into the application of neural networks, fuzzy logic, genetic algorithms, and rough sets to automated knowledge acquisition for classification problems. Neurocomposites **24**, 37–54 (1999)

Jang, J.S.R., Sun, C.T.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Englewood Cliffs, NJ (1997)

Javadi, A.A., Farmani, R., Tan, T.P.: A hybrid intelligent genetic algorithm. Adv. Eng. Inf. **19**, 255–262 (2005)

Kaufmann, A., Gupta, M.M.: Introduction to Fuzzy Arithmetic. V.N. Reinhold, New York (1985)

Kim, K.J., Han, I.: Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Exp. Sys. Appl. **19**(2), 125–132 (2000)

Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Machine Stud. **7**(1), 1–13 (1975)

Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Berlin (1992)

Nikravesh, M.: Soft computing-based computational intelligent for reservoir characterization. Exp. Syst. Appl. **26**, 19–38 (2004)

Oh, S.K., Pedrycz, W., Park, H.S.: Hybrid identification in fuzzy-neural networks. Fuzzy Sets Syst. 138:399–426 (2003)

Roger, J.J.S., Chuen-Tsai, S., Eiji, M.: Neuro-Fuzzy and Soft-Computing. Prentice-Hall, Englewood Cliffs, NJ (1997)

Sexton, R.S., Dorsey, R.E., Johnson, J.D.: Toward global optimization of neural networks: a comparison of the genetic algorithm and backpropagation. Dec. Supp. Syst. **22**(2), 171–185 (1998)

Tang, A.M., Quek, C., Ng, G.S.: GA-TSKfnn: parameters tuning of fuzzy neural network using genetic algorithms. Exp. Sys. Appl. **29**, 769–781 (2005)

Timur, A.: An investigation of permeability, porosity and 1273 residual water saturation relationships. Transactions SPWLD, 1274 Paper K (1968)

Wyllie, M.R.J., Rose, W.D.: Some theoretical considerations related to the quantitative evaluations of the physical characteristics of reservoir rock from electric log data. Trans. Am. Inst. Mining Eng. (AIME) **189**, 105–118 (1950)

Xie, D., Wilkinson, D., Yu, T., Ramon, S.: Permeability estimation using a hybrid genetic programming and fuzzy/neural inference approach. In: SPE 95167, Proceeding of SPE, Annual Technical Conference, Dallas, Texas, U.S.A., October (2005)

Yanga, T., Linb, H.C., Chena, M.L.: Metamodelling approach in solving the machine parameters optimization problem using neural network and genetic algorithms: a case study. Rob. Comp-Int. Manuf. **22**, 322–331 (2006)