Neural Comput & Applic
DOI 10.1007/s00521-012-0983-5

ORIGINAL ARTICLE

Evolving artificial neural network and imperialist competitive algorithm for prediction permeability of the reservoir

Mohammad Ali Ahmadi · Mohammad Reza Ahmadi · Seyed Reza Shadizadeh

Received: 8 November 2011 / Accepted: 22 May 2012

Abstract Permeability is the key parameter of the reservoir. In most reservoirs, permeability measurements are rare, and therefore permeability must be measured in the laboratory from reservoir core samples or evaluated from well test data. However, core analysis and well test data are usually only available from a few wells in a field. Unfortunately, coring every well in large fields is very expensive and uneconomical. In the present paper, the model based on a feed-forward artificial neural network (ANN) to predict permeability of the reservoir is proposed. After that, ANN model was optimized by imperialist competitive optimization (ICA). ICA is used to decide the initial weights of the neural network. The ICA-ANN model is applied to predict permeability in one of the northern Persian Gulf oil fields of Iran reservoir located in Ahwaz, Iran, utilizing available geophysical well log data. The imperialist operators and parameters are carefully designed and set avoiding premature convergence and permutation problems. For an evaluation purpose, the performance and generalization capabilities of ICA-ANN are compared with those of models developed with the common technique of BP. The results demonstrate that carefully designed imperialist competitive algorithm-based neural network outperforms the gradient descent-based neural network.

Keywords Artificial neural network · Imperialist competitive optimization · Permeability · Well log data

Abbreviations

CT	True conductivity
DT	Sonic transit time
RHOB	Bulk density
NPHI	Density tool reading
SGR	Standard gama ray
GA	Genetic algorithm
ANN	Artificial neural network
SA	Simulated annealing
ICA	Imperialist competitive algorithm
MSE	Mean square error

M. A. Ahmadi (✉)
Department of Petroleum Engineering,
Ahwaz Faculty of Petroleum Engineering,
Petroleum University of Technology,
P.O.BOX:63431, Kut Abdollah, Ahwaz, Iran
e-mail: ahmadi6776@yahoo.com

M. R. Ahmadi
Department of Mining Engineering,
Shahrood University of Technology, Shahrood, Iran

S. R. Shadizadeh
Department of Petroleum Engineering, Abadan Faculty
of Petroleum Engineering, Petroleum University of Technology,
Abadan, Iran

1 Introduction

Permeability is one of the most important rock parameters in reservoir engineering that affects fluids flow in reservoir. In most reservoirs, permeability measurements are rare and permeability is determined from rock sample or well testing data. Core analysis and well test data are expensive and time-consuming [1].

Neural networks have been increasingly applied to predict reservoir properties using well log data [2–4]. Moreover, previous investigations [5–7] have indicated that artificial neural networks (ANNs) can predict formation permeability even in highly heterogeneous reservoirs using geophysical well log data with good accuracy.

Published online: 12 June 2012

A soft sensor is a conceptual device whose output or inferred variable can be modeled in terms of other parameters that are relevant to the same process. According to Rallo et al. [8], artificial neural network could be used as soft sensor building approach. The ANN is a popular, nonlinear, nonparametric tool in well log analysis. This technique has been increasingly applied to predict reservoir properties using well log data [3, 9].

To determine the network structure and its relevant parameters such as connecting weights, some evolutionary algorithms such as genetic algorithm (GA) [10], unified particle swarm optimization (UPSO) [11–14], particle swarm optimization [15–17], imperialist competitive algorithm (ICA) [18], hybrid genetic algorithm and particle swarm optimization [19, 20], shuffled frog leaping algorithm (SFLA) [21], stochastic particle swarm optimization (SPSO) [22], pruning algorithm (PA) [23] and back propagation (BP) [24] can be applied.

At the same time, since neural network training can be consider as a type of optimization problem, recently, some evolutionary algorithms inspired by social behavior in the nature are also developed to solve NN training, such as particle swarm paradigm, which simulates swarm behavior of ants or birds. Imperialist competitive algorithm (ICA) was just developed in 2007 (Atashpaz-Gargari et al. [25]). It has become as hot topic involving optimization issues because of ICA's simple structure and easy implementation in practice.

In this paper, the imperialist competitive algorithm was combined with artificial neural network (ANN) algorithm to construct a hybrid learning algorithm and also to optimize the weights of feed-forward neural network. In the hybrid algorithm, distribution of the initial particles is random in the parameter space and the global optimum is resulted from the global searching of the case–study parameters. The parameters for making a network model are designed carefully to prevent early convergence and permutation problems. Results obtained from this study clearly indicate that hybrid imperialist competitive algorithm (ICA) method enhances the reliability and predictability of neural network (NN). According to the simulation results, high potential of the proposed network for permeability prediction can be concluded if the outputs from the approach proposed in this paper and back propagation (BP) neural network are compared using the available well log data for one of the Persian Gulf oil fields in southern Iran. The methodologies used here in this study and the results obtained are discussed in greater details in the next sections.

2 Artificial neural networks

Artificial neural networks are parallel information processing methods that can express complex and nonlinear relationship and use number of input–output training patterns from the experimental data. ANNs provide a nonlinear mapping between inputs and outputs by its intrinsic ability (Hornik and Stinchcombe [26]). The success in obtaining a reliable and robust network depends on the correct data preprocessing, correct architecture selection, and correct network training choice strongly (Garcia et al. [27]).

The most common neural network architecture is the feed-forward neural network. Feed-forward network is the network structure in which the information or signals will propagate only in one direction, from input to output. A three-layered feed-forward neural network with back propagation algorithm can approximate any nonlinear continuous function to an arbitrary accuracy [28]; Hornick and Stinchcombe [29].

The network is trained by performing optimization of weights for each node interconnection and bias terms. Until the values at the output layer neurons are as close as possible to the actual outputs. The mean squared error of the network (MSE) is defined as:

$$\mathrm{MSE} = \frac{1}{2}\sum_{k=1}^{G}\sum_{j=1}^{m}\left[Y_j(k) - T_j(k)\right]^2, \tag{1}$$

where m is the number of output nodes, G is the number of training samples, $Y_j(k)$ is the expected output, and $T_j(k)$ is the actual output.

The data are split into two sets, a training data set and a validating data set. The model is produced using only the training data. The validating data are used to estimate the accuracy of the model performance. In training a network, the objective is to find an optimum set of weights. When the number of weights is higher than the number of available data, the error in fitting the non-trained data initially decreases, but then increases as the network becomes overtrained. In contrast, when the number of weights is smaller than the number of data, the overfitting problem is not crucial.

3 Imperialist competitive algorithm

The imperialist competitive algorithm (ICA) is a new evolutionary algorithm in the evolutionary computation field based on the human's sociopolitical evolution (Atashpaz-Gargari et al. [25]). Like other evolutionary algorithms, the ICA starts with initial populations called countries. There are two types of countries: colony and imperialist (in optimization terminology, countries with the least cost), which together form empires. In the imperialistic competition process, imperialists try to attempt to achieve more colonies. So during the competition, the powerful imperialists will be increased in the power and

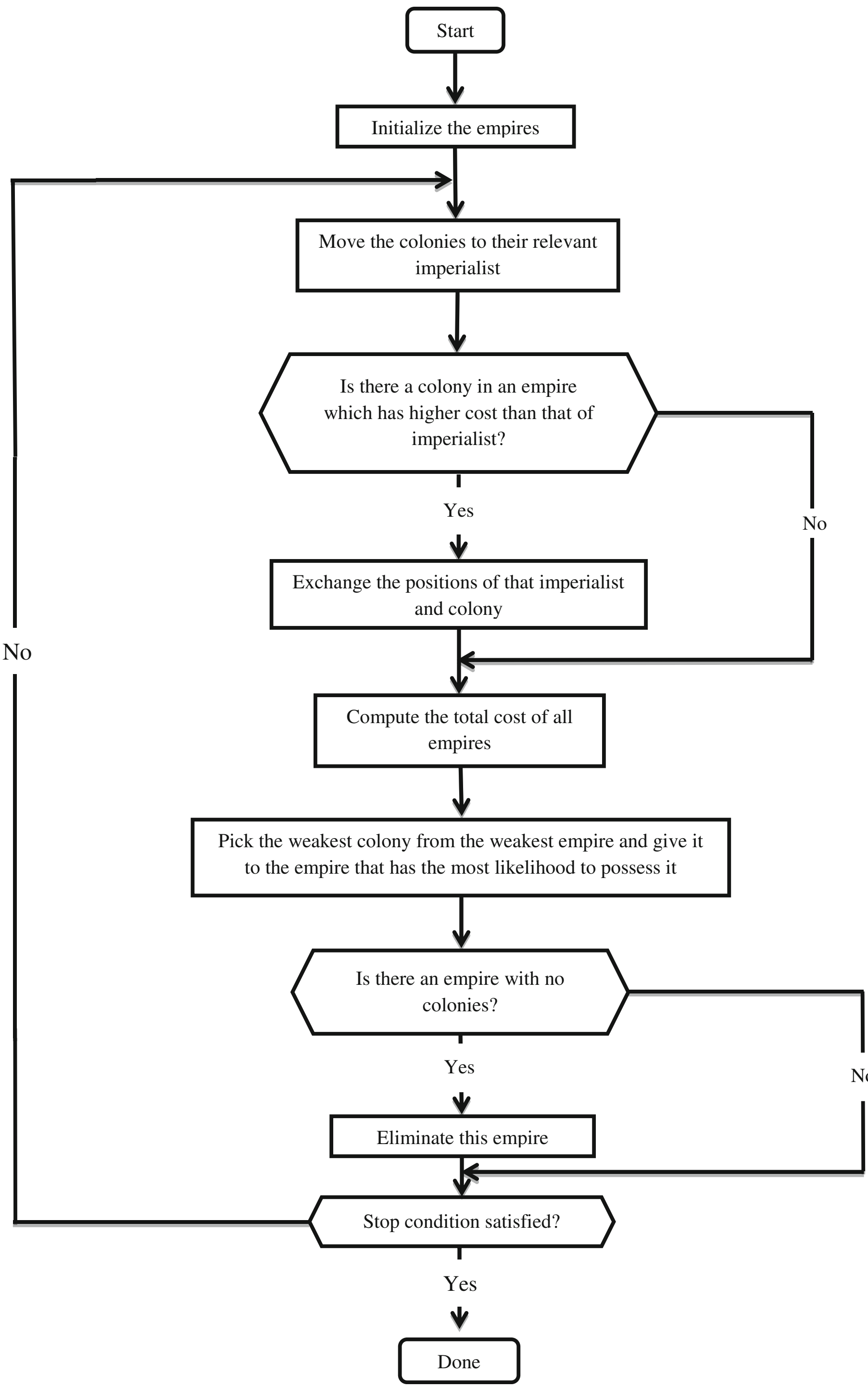

Fig. 1 Flowchart of the ICA algorithm

the weak ones will be decreased in the power. When an empire loses all of its colonies, it is assumed to be collapsed. At the end, the most powerful imperialist will remain in the world and all the countries are colonies of this unique empire. In this stage, imperialist and colonies have the same position and power (Fig. 1).

The implementation procedures of our proposed matching strategy based on ICA are described as follows.

3.1 Generating initial empire

The goal of optimization is to find an optimal solution in terms of variables of the problem. They form an array of variable values to be optimized. In GA terminology, this array is called "chromosome," but there the term "country" is used for this array. In an N_{var}—dimensional optimization problem, a country is an $1 \times N_{var}$ array. This array is defined by:

$$\text{Country} = [P_1, P_2, P_3, \ldots, P_{N_{\text{var}}}] \tag{2}$$

The cost of a country is found by evaluating the cost function f:

$$\text{Cost} = f(\text{country}) = f([P_1, P_2, P_3, \ldots, P_{N_{\text{var}}}]) \tag{3}$$

The algorithm starts with the number of initial countries (N_{country}), number of imperialist (N_{imp}), and number of the remaining countries are colonies that each belongs to an empire (N_{col}). The initial number of colonies of an empire is convenience with their powers. To divide the colonies among imperialists proportionally, the normalized cost of an imperialist is defined by:

$$C_n = c_n - \max_i\{c_i\} \tag{4}$$

where c_n is the cost of nth imperialist and C_n is its normalized cost. Having the normalized cost of all imperialist, the power of each imperialist is calculated by:

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{N_{\text{imp}}} C_i} \right| \tag{5}$$

On the other hand, the normalized power of an imperialist is determined by its colonies. Then, the initial number of an imperialist will be:

$$\text{N.C}_n = \text{round}\{P_n.N_{\text{col}}\} \tag{6}$$

where N.C_n is the initial number of colonies of nth empire and N_{col} is the number of all colonies. To divide the colonies among imperialists, N.C_n of the colonies is selected randomly and assigned them to each imperialist. The colonies together with the imperialist form the nth empire.

3.2 Moving colonies of an empire toward the imperialist

The imperialist countries try to improve their colonies and make them a part of themselves. This fact is modeled by moving all colonies toward their relevant imperialist. Figure 2 shows this movement. In this figure, the colony moves toward the imperialist by x (it is a random variable with uniform distribution) units.

$$x \sim U(0, \beta \times d) \tag{7}$$

where β is a number >1 and d is the distance between a colony and an imperialist. In the moving process, a colony may reach a position with lower cost than that of its imperialist. In this case, the imperialist and the colony change their positions. Then, the algorithm will continue by the imperialist in the new position and then colonies start moving toward this position.

3.3 The total power of an empire

The total power of an empire depends on both the power of the imperialist country and the power of its colonies. This fact is modeled by defining the total cost by:

$$\begin{aligned}\text{TC}_n = {} & \text{Cost}(\text{imperialist}_n) \\ & + \xi \,\text{mean}\{\text{cost}(\text{colonies of impire}_n)\},\end{aligned} \tag{8}$$

where T.C._n is the total cost of the nth empire, and ξ is a positive number that is considered to be <1. A small value for ξ implies that the total power of an empire to be determined by just the imperialist and increasing it will increase the role of the colonies in determining the total power of an empire. The value of 0.1 for ξ is a proper value in most of the implementations.

3.4 Imperialistic competition

All empires try to take the possession of colonies of other empires and control them. The imperialistic competition gradually brings about a decrease in the power of weaker empires and an increase in the power of more powerful ones. This competition is modeled by just picking some (usually one) of the weakest colonies of the weakest empires and making a competition among all empires to possess this colonies.

To start the competition, first, the possession probability of each empire is found based on its total power. The normalized total cost is obtained by:

$$\text{N.T.C.}_n = \text{T.C.}_n - \max_i\{\text{T.C.}_i\} \tag{9}$$

where T.C._n and N.T.C._n are the total cost and the normalized total cost of nth empire, respectively. Having the normalized total cost, the possession probability of each empire is given by:

$$P_{P_n} = \left| \frac{\text{N.T.C.}_n}{\sum_{i=1}^{N_{\text{imp}}} \text{N.T.C.}_i} \right| \tag{10}$$

To divide the mentioned colonies among empires, vector **P** is formed as

$$\mathbf{P} = P_{P_1}, P_{P_2}, P_{P_3}, \ldots, P_{P_{N_{\text{imp}}}} \tag{11}$$

Then, the vector **R** with the same size as P whose elements are uniformly distributed random numbers is created,

$$\mathbf{R} = [r_1, r_2, r_3, \ldots, r_{N_{\text{imp}}}] \tag{12}$$

Then, vector **D** is formed by subtracting **R** from **P**

$$\mathbf{D} = \mathbf{P} - \mathbf{R} = [D_1, D_2, D_3, \ldots, D_{N_{\text{imp}}}] \tag{13}$$

Referring to vector **D,** the mentioned colony (colonies) is handed to an empire whose relevant index in **D** is maximized.

Powerless empire will collapse in the imperialistic competition and their colonies will be divided among other empires. At the end, all the empires except the most powerful one will collapse and all the colonies will be under the control of this unique empire. In this stage, imperialist and colonies have the same position and power.

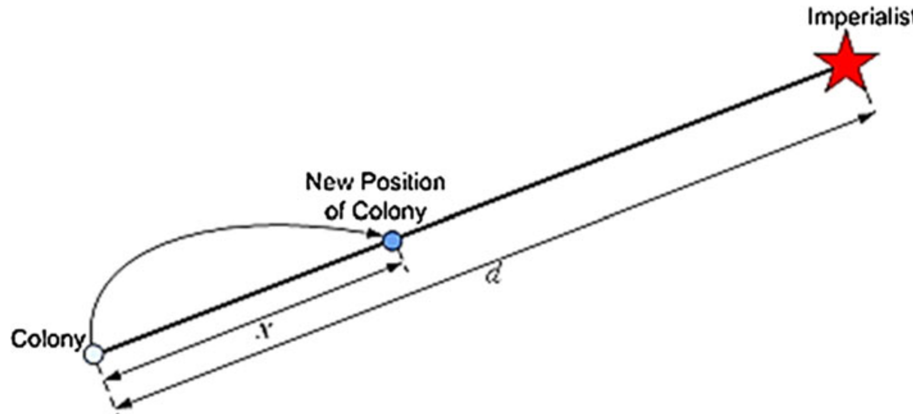

Fig. 2 Movement of colonies toward their relevant imperialist (Atashpaz-Gargari et al. [25])

Table 1 Some of the real data

K (mD)	CT	DT	NPHI	RHOB	GR
0.03	0.0026	51.5319	−0.01003	2.3075	5.2697
0.04	0.003546	52.1627	0.009855	2.3817	9.352
0.05	0.004672	52.5765	0.016694	2.4043	10.7745
0.09	0.007756	52.8514	0.019563	2.4594	12.939
0.116	0.01918	53.2675	0.023121	2.4752	13.1825
0.12	0.019273	53.4376	0.026208	2.4868	13.2099
0.123	0.020457	53.4577	0.028429	2.4934	13.2429
0.128	0.0207	53.575	0.033688	2.5329	13.9082
0.14	0.023538	54.0661	0.038018	2.5442	14.8658
0.15	0.028745	54.4946	0.044901	2.5448	15.1563

4 Case study: an oil field in Iran

In this study, field data are implemented in order to verify the efficiency of the predictions made by the developed algorithm. The oilfield under question is Mansouri oilfield, 30 km in length and 3.5 km in width, which is located 40 km from south of Ahvaz city in south of Iran. Mansouri oilfield is composed of two reservoirs: Asmari and Bangestan. Asmari reservoir is Tertiary carbonate reservoir, a very well-known formation, and reservoir in the Middle East. The type section of the Asmari formation is located on the SW flank of an anticline in the Kuh-e Asmari in the Khuzestan Province of Iran. In the type section, only the middle and upper parts of the limestone are exposed. Pabdeh formation that is a shale formation represents the lower part of the formation. The Asmari consists of creamy- to brown-colored, fractured, dense limestone with little primary porosity. The Asmari reservoir at the Mansouri oilfield consists of 8 different zones each with different geological and petrophysical properties among them zones 1–3 make good pay zones. Data from Zone 2 of the Asamri carbonate reservoir are used for the purpose of this study. Bangestan reservoir in the Mansouri oilfield consists of three different formations: Ilam, Sarvak, and Kazhdomi.

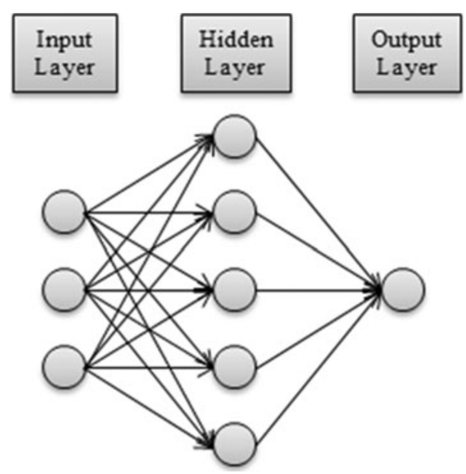

Fig. 3 Architecture of three-layer ANN

5 ICA-ANN-based soft sensor results

According to Eq. (1), the minimizing process of MSE value is the adjusting and optimizing process of weights and thresholds of the ANN. Therefore, the ICA is used to optimize the weights and thresholds of the ANN. The weights optimization is considered in this paper.

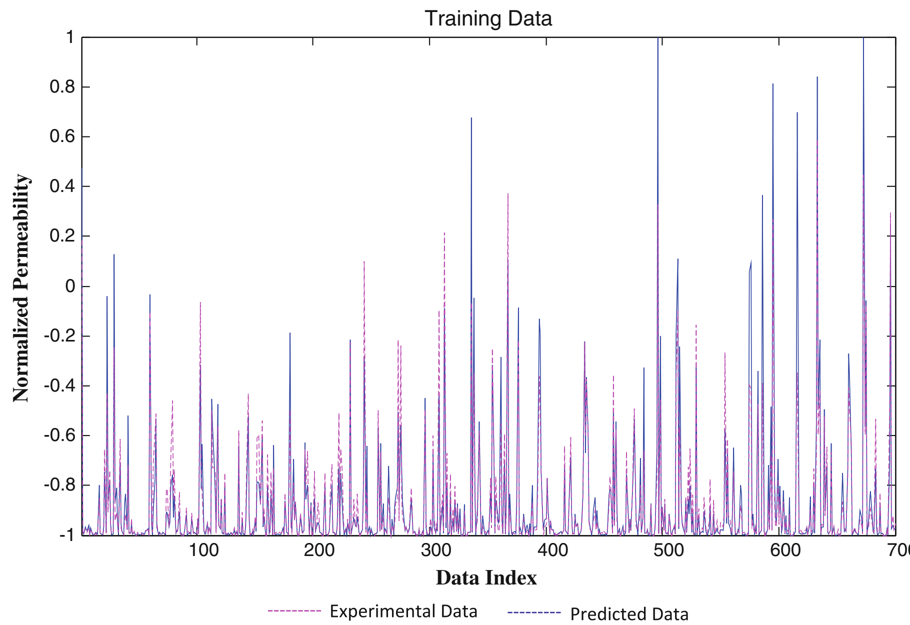

Fig. 4 Comparison between measured and predicted permeability (ICA-ANN) training

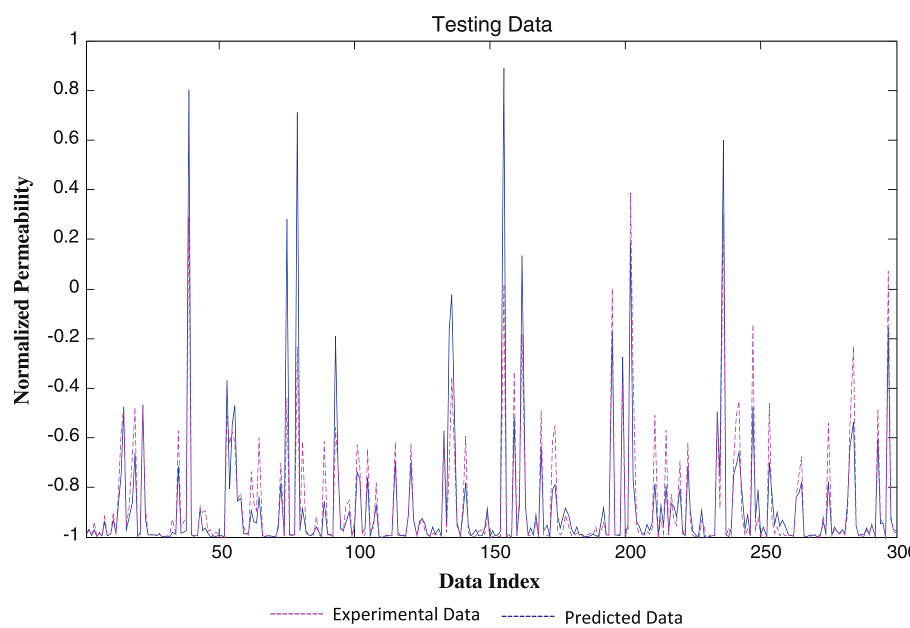

Fig. 5 Comparison between measured and predicted permeability (ICA-ANN) test

5 ANN architectures with different number of hidden layers were examined in order to find the most appropriate architecture, based on the values of mean square error (MSE) and efficiency coefficient (R^2) for training and testing runs. The results obtained at this stage showed that the best ANN architecture is 5-7-1 (5 input units, 7 hidden neurons, and 1 output neuron) for the ICA-ANN method used in this study. The developed ANN model has 7 hidden neurons in the mid-layer and sigmoid and linear activation functions in hidden and output neurons, respectively. Before training and testing, all source data are normalized into the range between -1 and 1, by using the maximum and minimum values of the variable over the whole data sets. The developed ANN model trained with back propagation network (Fig. 2) was trained by the Levenberg–Marquardt least-squares method to predict permeability using five parameters [True Conductivity (CT), Sonic Transit Time (DT), Density Tool Reading (NPHI), Bulk Density (RHOB), and Gamma Ray (GR)] as inputs. Table 1 presents a part of the real data (including the above parameters) employed in this study (Fig. 3).

ICA is used as neural network optimization algorithm, and the mean square error (MSE) used as a cost function in this algorithm. The goal in proposed algorithm is minimizing this cost function. Every weight in the network is initially set in the range of $[-1, 1]$. In these simulations, the

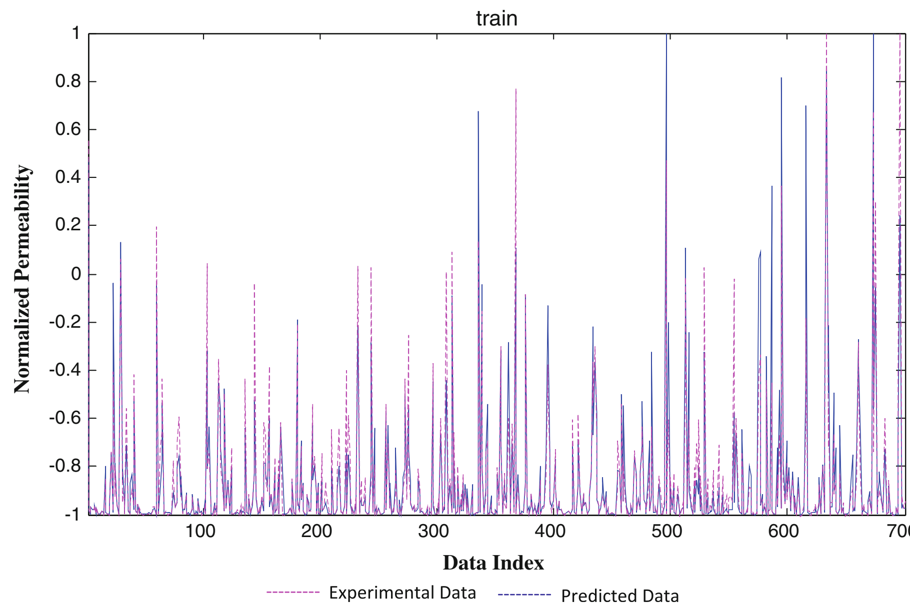

Fig. 6 Comparison between measured and predicted permeability (ANN) training

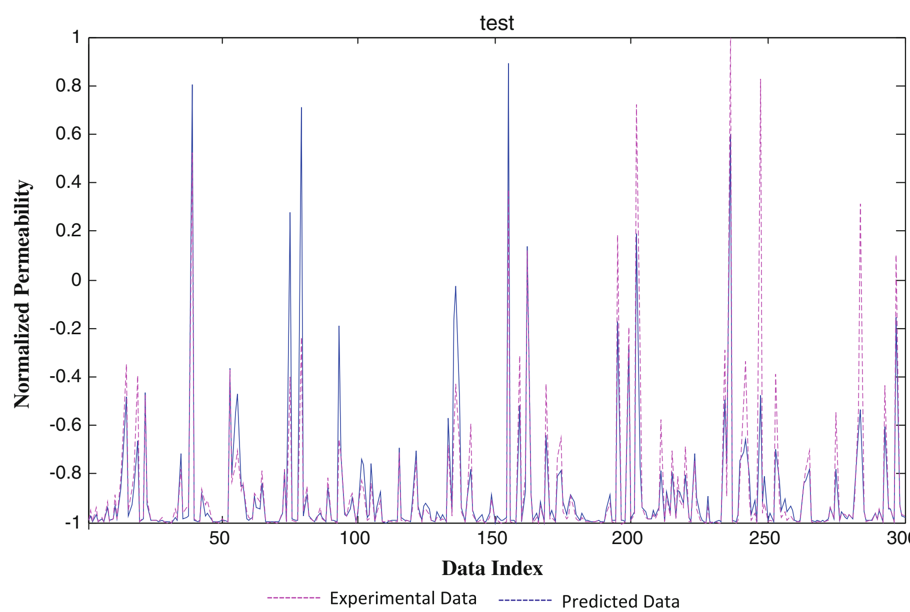

Fig. 7 Comparison between measured and predicted permeability (ANN) test

number of imperialists and the colonies are considered 4 and 40, respectively; parameter β is set to 2. 700 data samples were chosen by a random number generator for network training. The remaining 300 samples were put aside to be used for testing the network's integrity and robustness.

The permeability prediction of the reservoir in the training and test phase is shown in Figs. 4, 5, 6, and 7, respectively. It should be mentioned here that the legend of the vertical axis in Figs. 4, 5, 6 and 7 presents normalized permeability that was calculated using the following expression:

$$\text{Normalized Permeabliity} = \frac{2(k - k_{\min})}{(k_{\max} - k_{\min})} - 1 \tag{14}$$

where $k_{\min}$ and $k_{\max}$ are the minimum and maximum permeabilities of the data used in this study, respectively. In these figures, values of predicted permeability by using ICA-ANN model and ANN model and experimentally measured permeabilities against data indexes are plotted. The simulation performance of the ICA-ANN model was evaluated on the basis of mean square error (MSE) and efficiency coefficient R^2. Table 2 gives the mean square error and R^2 values for the two different models of the validation phases. According to this table, ICA-ANN model with $R^2 = 90634$ has good performance. So it can be observed that the performance of ICA-ANN model is better than ANN model. In general, a R^2 value >0.9 indicates a very satisfactory model performance, while a R^2 value in the range 0.8–0.9 signifies a good performance, and value <0.8 indicates an unsatisfactory model performance. Figures 8 and 9 show the extent of the match between the measured and predicted permeability values by ICA-ANN and ANN models in terms of a scatter diagram. Figures 8 and 9 are regression plot between ICA-ANN and ANN models prediction values against experimentally measured permeabilities. According to these figures and by comparing R^2 value of this plots, we should respect the effectiveness of the ICA-ANN model.

Table 2 Comparison between the performances of ICA-ANN- and ANN-based soft sensor

	ICA-ANN	ANN
MSE	0.0131	0.0258
R^2	0.90634	0.8043

6 Conclusions

In this article, imperialist competitive algorithm evolved neural network has presented. The idea of our algorithm is that each initial point of the neural network is selected by an imperialist competitive algorithm. The fitness of the imperialist competitive algorithm is determined by neural network. The imperialist competitive parameters are carefully designed to optimize the neural network, avoiding premature convergence. The experiment with real well logs

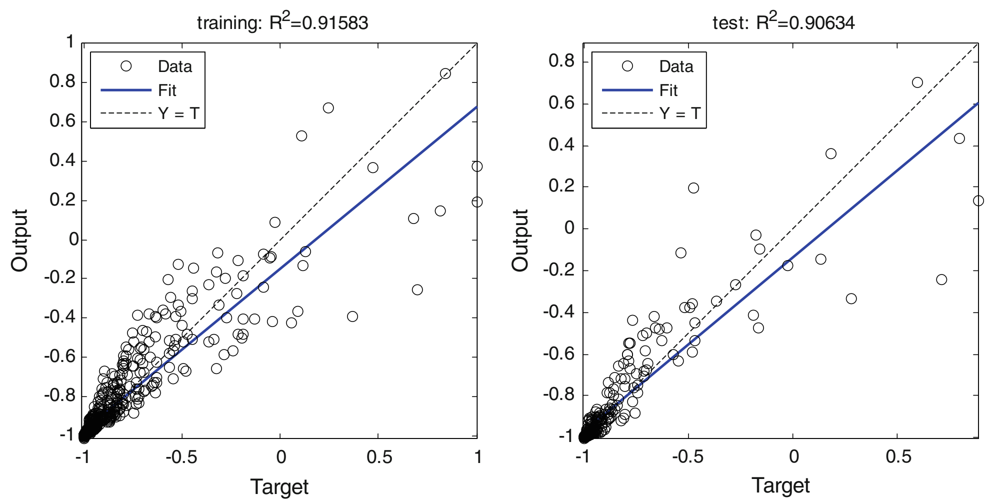

Fig. 8 R^2 ICA-ANN

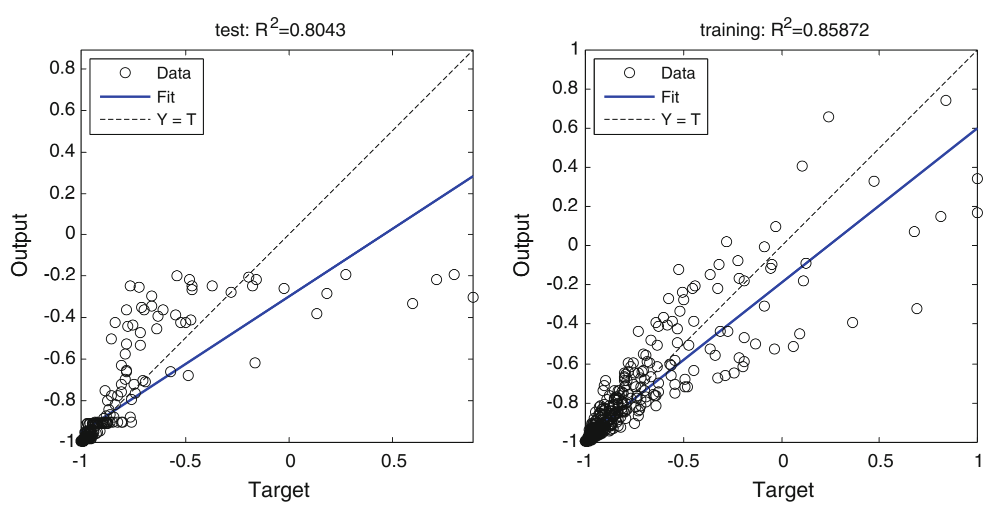

Fig. 9 R^2 ANN

and core measurements data has showed that the predictive performance of the proposed model is better than that of the traditional neural network. This has been supported by the analysis of the changes in connection weights and biases of the neural network. One problem when considering the combination of neural networks and imperialist competitive algorithm for permeability prediction is the determination of the optimal neural network topology. Neural network topology described in this experiment is determined manually. A substitute method is to apply the imperialist competitive algorithm for neural network structure optimization, which will be a part of our future work.

References

1. Weldu T, Ghedan S, Al-Farisi O (2010) Hybrid AI and conventional empirical model for improved prediction of Log-derived permeability of heterogeneous Carbonate reservoir. SPE production and operation conference, Tunis, 8–10 June, SPE No.136127
2. Mohaghegh S, Arefi R, Ameri S, Rose D (1994) Design and development of an artificial neural network for estimation of formation permeability. In SPE 28237, petroleum computer conference, July 31–August 3, Dallas
3. Balan B, Mohaghegh S, Ameri S (1995) State-of-the-art in permeability determination from well log data: part I-a comparative study, model development. In: SPE Eastern regional conference and exhibition, West Virginia, pp 17–21
4. Wiener J (1995) Predict permeability from wireline logs using neural networks. Petroleum Eng Int 67(5):18–24
5. Aminian K, Bilgesu HI, Ameri S, Gil E (2000) Improving the simulation of waterflood performance with the use of neural networks. In SPE 65630, proceeding of SPE Eastern regional conference, October
6. Aminian K, Thomas B, Bilgesu HI, Ameri S, Oyerokun A (2001) Permeability distribution prediction. In SPE paper, proceeding of SPE Eastern regional conference, October
7. Wong PM, Jang M, Cho S, Gedeon TD (2000) Multiple permeability predictions using an observational learning algorithm. Comput Geosci 26(8):907–913
8. Rallo R, Ferre-Gin J, Arenas A, Giralt F (2002) Neural virtual sensor for the inferential prediction of product quality from process variables. Comput Chem Eng 26:1735–1754
9. Doveton JH, Prensky SE (1992) Geological applications of wireline logs: a synopsis of developments and trends. Log Anal 33(3):286–303
10. Qu X, Feng J, Sun W (2008) Parallel genetic algorithm model based on AHP and neural networks for enterprise comprehensive business. Intelligent Information Hiding and Multimedia Signal Processing, Harbin, 15–17 August 2008
11. Ahmadi MA (2012) Neural network based unified particle swarm optimization for prediction of asphaltene precipitation. Fluid Phase Equilib 314:46–51
12. Ahmadi MA, Shadizadeh SR, Goudarzi A (2012) Combining artificial neural network and unified particle swarm optimization for oil flow rate prediction: case study. Neural Comput & Applic. doi:10.1007/s00521-012-0955-9
13. Ahmadi MA, Shadizadeh SR (2012) Neural-network-based unified particle swarm optimization for the prediction of asphaltene precipitation due to natural depletion. Neural Comput & Applic. doi:10.1007/s00521-012-0921-6
14. Ahmadi MA, Shokrollah-Zadeh Behbahani H. New approach for permeability prediction from well log data by using evolutionary algorithm, part A: recovery, utilization, and environmental effects. J Energy Sources. doi:10.1080/15567036.2011.605427
15. Ahmadi MA. Prediction of asphaltene precipitation by using pso-neural network, part A: recovery, utilization, and environmental effects. J Energy Sources. doi:10.1080/15567036.2011.598902
16. Ahmadi MA, Shadizadeh SR. New approach for prediction of asphaltene precipitation due to natural depletion by using evolutionary algorithm concept. J Fuel. doi:10.1016/j.fuel.2012.05.050
17. Zendehboudi S, Ahmadi MA, James L, Chatzis I. Prediction of condensate-to-gas ratio for retrograde gas condensate reservoirs using artificial neural network with particle swarm optimization. Energy Fuels. doi:10.1021/ef300443j
18. Ahmadi MA (2011) Prediction of asphaltene precipitation using artificial neural network optimized by imperialist competitive algorithm. J Petrol Explor Prod Technol 1:99–106. doi:10.1007/s13202-011-0013-7
19. Ahmadi MA, Zendehboudi S, Lohi A, Elkamel A, Chatzis I. Application of hybrid genetic algorithm with particle swarm optimization and neural network for reservoir permeability prediction. J Geophys Prospect. doi:10.1111/j.1365-2478.2012.01080.x
20. Ahmadi MA, Shadizadeh SR (2012) Prediction of asphaltene precipitation by using hybrid genetic algorithm and particle swarm optimization and neural network. Neural Comput & Applic. doi:10.1007/s00521-012-0920-7
21. Ahmadi MA, Shadizadeh SR (2012) Permeability prediction of carbonate reservoir by combining neural network and shuffled frog-leaping. J Amer Sci 8(2):529–533
22. Ahmadi MA, Shadizadeh SR, Hasanvand M (2011) Neural Network based stochastic particle swarm optimization for prediction of minimum miscible pressure. Int J Comput Appl 34(1):15–19
23. Reed R (1993) Pruning algorithms-a survey. IEEE Trans Neural Netw 4:740–747
24. Ahmadi MA, Shirdel MR, Samaee MA. Prediction of asphaltene precipitation by using artificial intelligence. Pet Sci Technol. doi:10.1080/10916466.2011.590834
25. Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, IEEE congress on evolutionary computation, pp 4661–4667
26. Hornik K, Stinchcombe M, White H (1990) Universal approximation of an unknown mapping and its derivatives using multilayer feed forward networks. Neural Netw 3(5):551–560
27. Garcia-Pedrajas N, Hervas-Martinez C, Munoz-Perez J (2003) COVNET: a cooperative co evolutionary model for evolving artificial neural networks. IEEE Trans Neural Netw 14:575–596
28. Brown M, Harris C (1994) Neural fuzzy adaptive modeling and control. Prentice-Hall, Englewood Cliffs
29. Hornick K, Stinchcombe M, White H (1989) Multilayer feed forward networks are universal approximators. Neural Netw 2:359–366