Neural Comput & Applic (2012) 21:409–421
DOI 10.1007/s00521-010-0501-6

REVIEW

Artificial neural networks workflow and its application in the petroleum industry

N. I. Al-Bulushi · P. R. King · M. J. Blunt · M. Kraaijveld

Received: 2 October 2009 / Accepted: 19 November 2010 / Published online: 9 December 2010

Abstract We develop a neural network workflow, which provides a systematic approach for tackling various problems in petroleum engineering. The workflow covers several design issues for constructing neural network models, especially in terms of developing the network structure. We apply the model to predict water saturation in an oilfield in Oman. Water saturation can be accurately obtained from data measured from cores removed from the oil field, but this information is limited to a few wells. Wireline log data are more abundantly available in most wells, and they provide valuable, but indirect, information about rock properties. A three-layered neural network model with five hidden neurons and a resilient back-propagation algorithm is found to be the best design for the saturation prediction. The input variables to the model are density, neutron, resistivity, and photo-electric wireline logs, and the model is trained using core water saturation. The model is able to predict the saturation directly from wireline logs with a correlation coefficient (r) of 0.91 and an error of 2.5 saturation units on the testing data.

N. I. Al-Bulushi (✉)
Petroleum Development of Oman (PDO),
P.O. Box 761, 113 Muscat, Oman
e-mail: nabil.i.bulushi@pdo.co.om

P. R. King · M. J. Blunt
Department of Earth Science and Engineering,
Imperial College London, London SW7 2AZ, UK
e-mail: peter.king@imperial.ac.uk

M. J. Blunt
e-mail: m.blunt@imperial.ac.uk

M. Kraaijveld
Shell International, Rijswijk, Netherlands
e-mail: M.Kraaijveld@shell.com

Keywords Artificial neural networks · Neural network workflow · Generalization · Wireline logs · Water saturation

1 Introduction

Artificial neural networks (ANNs) efficiently provide sophisticated data-processing solutions to complex problems since they can find highly nonlinear relationships [1]. Furthermore, neural networks have the ability to generalize, and they are robust against noise. These abilities make ANNs suitable for solving many problems in the petroleum industry [1]. They have been used to predict porosity, permeability, determine facies, and identify zones [2–4]. Helle and Bhatt [5] demonstrated the successful application of ANNs to predict water saturation using only wireline logs. In previous work, we showed a case study for ANN application for saturation prediction where we briefly discussed the neural network part [6]. In this work, we focus more on the neural network development and we present a comprehensive analysis of the neural network workflow.

In this paper, a systematic workflow is developed to construct the neural networks models. The workflow covered several design issues for developing neural networks models, especially for constructing the structure of the model. We assess the relevance of the statistics of the data and the importance of determining the uncertainties in the original data. The contribution of input variables was determined, and the results were compared with other regression models.

Oil and gas (hydrocarbon) are very important energy sources worldwide. One objective of the petroleum industry when an oilfield is first discovered is to obtain an accurate estimate of the hydrocarbon volume in place (reserve) before any money is invested in production and development. One of the important parameters involved in calculating the reserves is water saturation (S_{w}). Water saturation is defined as the volume fraction of pore space of formation rock that is filled with water, where the rest of the pore space is filled with either oil or gas. There are many methods available in the industry to calculate the water saturation; these include petrophysical evaluation models [7–9]. However, all these methods have many limitations and, more importantly, the input parameters to these models are often not readily available [10–12]. In particular, the presence of shales (low permeability layers) in the formation makes saturation prediction problematic using wireline log data. Water saturation can also be directly determined from core measurements; however, core data are only available in a few wells in the field since they are expensive to obtain. In this paper, ANNs are applied to predict water saturation in shaly formations directly from wireline logs and using core data as training samples.

2 Artificial neural networks

Artificial neural networks have been designed to imitate the function of the biological neurons of the human brain. Their main feature is the ability to find highly complex nonlinear relationships between variables [1]. Furthermore, they have the ability to learn and generalize, where they can produce reasonable results in the hidden testing patterns [13]. Furthermore, the ANNs are not limited by the assumptions of the underlying model [14]. Generally, they are capable of solving several types of problems, including function approximation, pattern recognition and classification and optimization and automatic control [15].

A major element of ANNs is the perceptron or artificial neuron [16]. These neurons mimic the action of the abstract biological neuron. Figure 1 is a schematic representation of an artificial neuron. An ANN is composed of many neurons connected by a line of communication; these are called connections [16]. They can be trained to perform a certain task by adjusting the values of the connection between the elements. The most commonly used ANN architecture is the multi-layer perceptron (MLP). An MLP is a cascade of two or more layers of perceptrons arranged in a layered fashion, and each layer is fully connected to the next [17]. Based on connectivity, there are two main types of MLP: feed-forward and feedback networks [17]. In a feed-forward network, the signals are allowed to travel one way only, from input to output, whereas in feedback type, the network can have signals traveling in both directions; this is achieved by introducing loops in the network. Moreover, the MLP can be categorized into supervised and unsupervised network based on the training method used [16]. Generally, an MLP has three main layers: input, hidden, and output layers. The input layer provides the network with the necessary information from the outside world. The hidden layer is responsible for the main part of the input to output mapping. The output layer does the final processing and outputs the data to the outside world.

The key operation in the development of an ANN is the learning process, by which the free parameters (weight and bias) are modified through a continuous process of stimulation by the environment in which the network is embedded [13]. The learning process continues until a satisfactory error is reached. One complete presentation of the entire training data to the network during the training process is called an epoch or one training cycle [13]. The overall objective of the MLP learning is to optimize the performance function. Optimizing means finding the minimum of the performance function [18]. Figure 2 summarizes the operating system of a multi-layer perceptron. Many learning algorithms, such as back-propagation (BP), BP with momentum, resilient propagation (PROP), conjugate gradient and Levenberg–Marquardt [18], are available to train the network. The most widely used algorithm is BP. BP uses the chain rule to determine the influence of each weight in the network structure with respect to the derivatives of the error function. The process starts by computing the derivatives of the performance function at last layer of the network and then propagating the derivatives backward until the first layer of the network is reached [13, 16]. However, there are many drawbacks associated with it, such as the overall convergence is slow, trapping in local minima and the selection of appropriate learning rate [13, 18]. A method proposed to avoid the problem of the learning rate in BP is to introduce a momentum term to the weight update. Thus, in addition to the weight adaptation due to the error signal (gradient of the error), the weight is also changed by a factor μ of the previous weight adaptation. One obvious problem with this method is that it contains finding the optimal value of two parameters, the

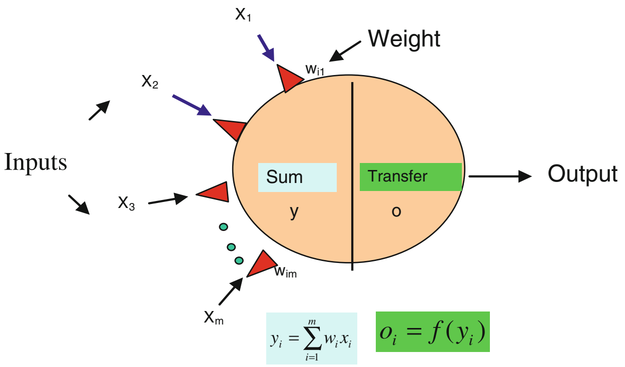

Fig. 1 Schematics of an artificial neuron, after Rojas (1996)

learning rate and the momentum term [17, 18]. In previous adaptive training algorithms (BP and its modification), the size of the actual weight step is dependent not only on the learning rate but also on the partial derivatives. Even though the learning rate is carefully adapted, it is nevertheless drastically disturbed by the unforeseeable behavior of the derivative itself [18]. RPROP is a new efficient learning scheme, designed by Reidmiller and Braun [19]. This technique performs a direct adaptation of the weight step based on local gradient information, in which the adaptation process is based on the sequence of signs of the partial derivatives in each dimension of weight space. The main difference between this method and other developed adaptation techniques is that the weight adaption does not depend on the size of the gradient [18]. This method achieves its adaptive weight by introducing an individual adaptive update value Δ_{ij}, which solely determines the size of the weight update. The Levenberg–Marquardt (LM) learning algorithm is derived from Newton's optimization method [18]. The main difference between the LM algorithm and the BP algorithm is the method in which the derivatives are used to update the weight. The basic concept of Newton's method is:

$$X_{K+1} = X_K - A^{-1} g_K. \qquad (1)$$

Newton's method is based on a second-order Taylor series expansion. A_{k} is the Hessian matrix (second derivatives) of the performance function. Newton's method often converges faster than the steepest descent method. Unfortunately, it is complex and expensive to compute the Hessian matrix for neural network models. The Levenberg–Marquardt algorithm was designed to approach the second-order Newton method training speed without the need to compute the Hessian matrix [18]. The main problem with the LM algorithm is the need for the large storage of some matrices of the free parameters. However, a technique of not computing and storing the whole approximated matrix in LM algorithms was developed to avoid the storage issue.

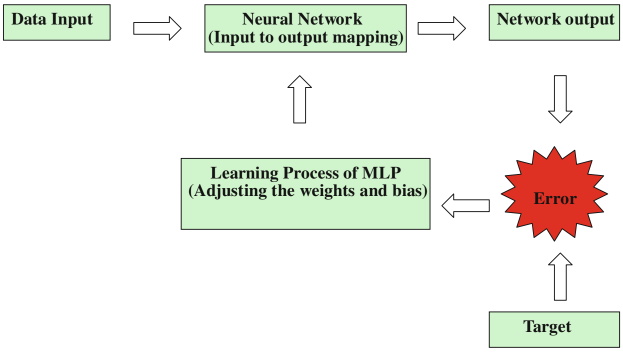

Fig. 2 The operating system of a multi-layer perceptron

The data in the neural network are divided into three main parts: training, validation, and testing subsets [13, 17]. The training data are used to train the network and to adapt its internal structure. The validation data are used during training along with the training data to monitor the performance of the network. They are not used to adapt the network. The testing data are kept aside until the whole training process has been completed. This set is used as a biased to investigate the generalization capability of the trained network on new data. It is basically used to test whether the network captured the general trend and did not memorize (fitted) the noise on the training data (overtraining).

There are two main modes for training [13, 20]: sequential (stochastic) and batch modes. In stochastic mode, the free parameter updating is performed after the presentation of each training example. In batch mode, the weight updating is performed after the presentation of all the training examples that constitute an epoch (one training cycle). The performance function is then the average sum of the square of the error (average of the whole training data samples). The sequential mode is much faster since it requires less local storage of connection weight [13, 20]. On the other hand, the batch mode has the advantage that conditions of the convergence are well understood [20]. Furthermore, many advanced learning algorithms such as conjugate gradients operate only in batch mode.

2.1 Stopping criteria and generalization

The aim of neural network model training is to obtain a low enough error solution for the problem under investigation. The learning network algorithm searches for the global lowest error. The main challenge in neural network modeling is how to set the criteria for network training termination. In other words, how can we stop the network from training before memorization (fitting the noise) takes place, where the lowest error found by the network might not be necessarily the best solution in order to generalize the model [13]. The ability of an ANN to execute well on hidden patterns (testing data subset) is called its ability to generalize [13, 16]. Besides the generalization issue, it is not always certain that the training error converges to a minimum or that it achieves it in a reasonable time. All these issues make the stopping criterion a complex issue in neural network modeling.

Generalization is one of the critical issues in developing an ANN model. It is more significant than the network's ability to map the training patterns correctly (finding the lowest error in the training subset), since the network objective is to solve the unknown case [16, 21]. The generalization is affected by three main factors: (1) the size of the data, (2) network size and (3) the complexity of the problem under investigation [13, 16]. The last factor though is out of our control.

Specifying the network size is an important task. If the network is very small, then its ability to provide a good solution of the problem might be limited. On the other hand, if the network is too big, then the danger of memorizing the data (not being able to generalize) will be high [16, 22]. Hush and Horne [16] pointed out that in general, it is not known what size of network works best for a problem under investigation. Furthermore, it is difficult to specify a network size for a general case since each problem will demand different capabilities. With little or no prior knowledge of the problem, the trial and error method can be used to determine the network size.

There are several approaches that may be used as a trial and error procedure to determine network size. One approach is to start with the smallest possible network and gradually increase the size [16, 23]. The optimal network size is then that point at which the performance begins to level off. After this point, the network will begin to memorize the training data. Another approach is to start with an oversized network and then apply a pruning technique that removes weight/nodes, which end up contributing little or nothing to the solution [20]. In this approach, an idea of what size constitutes a large network needs to be known [16].

Many studies suggest that one hidden layer network is capable of mapping any continuous functions, suggesting that one hidden layer is sufficient [24, 25]. Nevertheless, many other studies have shown that a larger number of hidden neurons are needed to accomplish the task [16, 17, 26–28]. Two hidden layers have benefits, especially when the complexity of the problem increases and when a prohibitive number of neurons are needed in one hidden layer [16, 29, 30]. However, Hush and Horne [16] pointed out that no more than two hidden layers should be used.

The number of neurons in the input and output layer is related to the nature of the problem under investigation, reflecting the number of input and output parameters [31]. In terms of the number of hidden neurons in the network structure, one should never use more hidden neurons than the number of training samples [16, 27, 32, 33]. The more free parameters there are relative to the number of the training cases, the more overfitting of the data will take place. The number of free parameter can be determined using Eq. 2 [34]:

$$\begin{aligned} N_{\mathrm{F}} &= I_{\mathrm{p}} \times H_{\mathrm{n}} + H_{\mathrm{n}} \times O_{\mathrm{p}} + H_{\mathrm{n}} + O_{\mathrm{p}} \\ &= O_{\mathrm{p}} + (I_{\mathrm{p}} + O_{\mathrm{p}} + 1)H_{\mathrm{n}} \end{aligned} \qquad (2)$$

where N_{F} number of free parameters, I_{p} number of inputs, O_{p} number of outputs, H_{n} number of hidden neurons.

N_{F} should be lower than the number of training data samples (N_{t}). In some cases, especially in a very noisy environment, the selection of a good network size is not enough for a good generalization. It is necessary to use other generalization methods besides the optimum model selection such as early stopping [17, 35, 36]. The validation data play a key role in the early stopping method. The validation error will normally decrease during the initial phase of training, along with the training set error. However, when the network starts fitting the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, training is stopped, and the weights and biases at the minimum of the validation error are returned. In good practice, the trained network is saved at the point defined by the early stopping criterion, and training continues solely to check whether the error will fall again. This will ensure that the increase in the validation error is not a temporary event [17, 35, 36]. Figure 3 shows the concept of cross-validation.

Another method of improving the generalization is by using a regularization approach such as the weight decay method [37]. In this approach, a term is added to the performance function in order to reduce the weight size, hence reducing the overall network complexity. Hush and Horn [16] explained the benefit of this approach; they divided the weight in the network into two main categories: weights with large influence on the solution and weights with small or no influence on the solution. The second group was referred to as excess weights. These excess weights can have a wide range of values, and they are not likely to take values near zero unless they are encouraged to do so. These excess weights cause poor generalization. Therefore, the weight decay method encourages these excess weights to take values near zero, thus improving the overall generalization. Furthermore, in addition to improving generalization, this method has another important advantage. After learning with the weight decay, the magnitude of each weight is directly proportional to its influence on the mapping error [16]. However, the drawback with this type of regularization is that it is difficult to determine the optimum value for the weight decay rate.

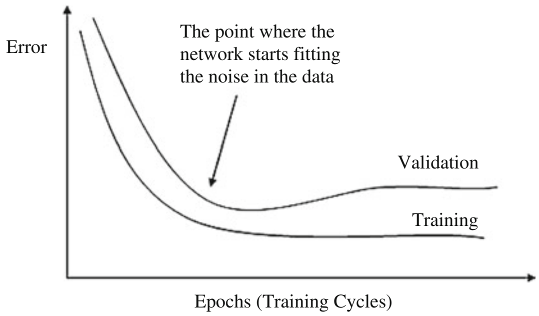

Fig. 3 Cross-validation stopping criterion, after Bishop (1995)

3 Neural network workflow

The overall objective of developing any prediction model is to build a model that solves the problem under investigation to the level of accuracy required. In this study, a systematic workflow or methodology to model a neural network was developed. The first part of the workflow focuses on the analysis of the data in terms of statistics and pre-processing. The second part focuses on different design issues in terms of finding the optimum number of hidden neurons and evaluating different learning algorithms. Finally, the workflow shows the importance of analyzing the relative contribution of the input variables and comparing the results of the neural network with other statistical methods. The workflow is shown schematically in Fig. 4. The different elements of workflow are explained in detail following the case study presented in this paper.

4 Neural network workflow for predicting water saturation in a sandstone formation in Oman

We used an ANN to predict the water saturation in a shaly formation in Oman. This formation was deposited in a braided stream environment and contains baffles produced by shale layers and rip-up mudstone conglomerates. Water saturation is defined as the volume fraction of the pore space of formation rocks filled with water, where the rest of the pore space is filled with either oil or gas. It is one of the most important parameters required in petroleum industry for hydrocarbon volume calculation. Determining the water saturation is not a simple task, especially in complex and heterogeneous reservoirs. The common method for determining the water saturation in industry is by using empirical and semi-empirical petrophysical models. All petrophysical models use information from wireline logs besides other information to determine the water saturation. However, in general, all water saturation models have many limitations, which lead to either the underestimation or the overestimation of the water saturation. These limitations in the water saturation models are the main justification for investigating new models.

Accurate measurement of water saturation can be obtained from core data. The Dean-Stark core data gives accurate measurements of water saturation, provided careful handling and special type of core is selected. However, this method is expensive compared with petrophysical model. Wireline logs are electrical measurements run in most wells of field where hydrocarbon is located. The wireline well log data are more abundantly available in most wells, and they provide valuable, but indirect, information about rock properties. Hence, we try to establish the complex nonlinear relationship between core and wireline log data using ANNs. This will then allow a prediction of water saturation and other reservoir properties in wells where no core data exist.

"Appendix" gives a background on wireline logs and coring in petroleum industry.

4.1 Problem definition

Problem definition includes the following steps:

- Define the property to be predicted
- Determine the data used to train the model
- Evaluate the uncertainties in the data
- Determine suitable model input parameters

The first step in the neural network workflow is to define the problem under investigation. Problem definition includes determining the property to be predicted and the truth data to train the model. Once the truth data are defined to train the model, it is important to evaluate the uncertainty in the original data. This provides a boundary of how much further the trained model needs to be optimized.

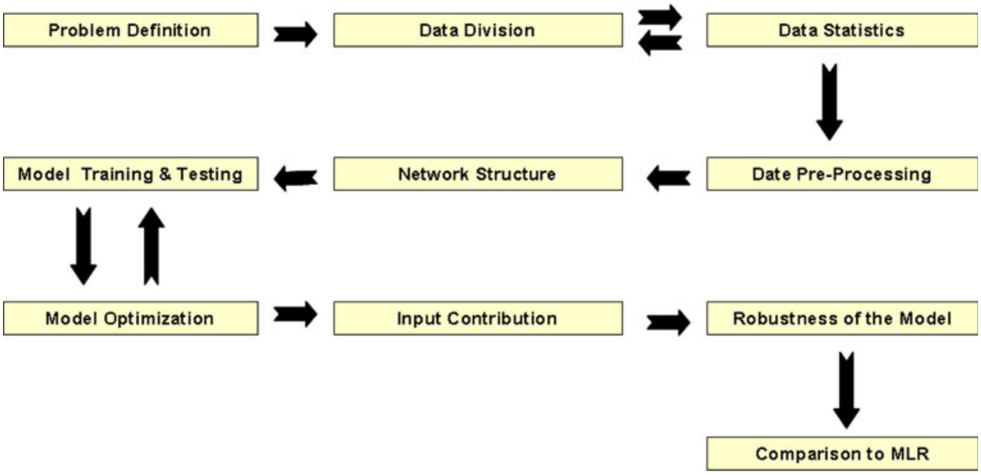

Fig. 4 Neural network workflow

In this case study, the network is used to predict the water saturation directly from wireline well logs, taking the core Dean-Stark water saturation as hard data to train the model. The data in this case are taken from a well that had sponge core water saturation. The sponge core is a special core sampling method where fluids that leak out of the core from pressure release decompression are captured by an oil-wet sponge that surrounds the core. In the laboratory, the total amount of the fluid (in core and sponge) is analyzed. Combined they should provide an estimate of the in situ saturation. The average total saturation of water and oil is found to be equal to 95.5%. This summation must add to 100%. Hence, a 4.5 saturation units (S.U.) uncertainty was estimated in water saturation values. These uncertainties are assumed to be in both water and oil estimates.

Selecting the appropriate input variables is an important issue in ANN modeling. When more inputs than required are selected, this will result in a large network size and consequently this will decrease the learning speed and efficiency of the method and reduce the generalization capability [15]. On the other hand, selecting few parameters might not be enough to model the problem under investigation. There are many approaches available to select the input parameters [38, 39]. These include the following:

- Understanding the physics of the problem under investigation and relating the parameters that have the highest impact. This step requires having a prior knowledge of the problem under investigation. However, there are many complex problems that make it difficult to determine all possible inputs.
- Taking a stepwise approach, training different networks with different combinations of input parameters and then selecting the inputs which produce best model performance.
- Using statistical dependence techniques, such as correlation or principal component analysis.

The potential model inputs to the network in this case are limited; therefore, the approximate physical principles approach is applicable here. The density, neutron, resistivity, and photo-electric (PE) wireline logs were used as input variables to the model. The gamma ray is not taken as input data for this particular case since it is disrupted by the existence of feldspar and mica. More information about physics behind these wireline logs can be found in the “Appendix”.

4.2 Data handling

The ANN is a data-driven model [40]. Therefore, the data play a major role in model design and development. The data are divided into two main subsets: operating data and the testing subset. The operating data are used to train the network and the testing subset to determine how well the model works. The operating data are further divided into training and validation, depending on the nature of the problem and the amount of the data available. There is no rigid rule in terms of selecting the amount of operating and testing data. However, generally the number of operating data is selected to be greater than the testing data. This is in order for the training to capture the overall heterogeneity and variability of the selected sample (this is the case for predicting the water saturation). However, if fewer data samples represent the overall variability in the data, then selecting more training data than the testing might not be visible. The total number of core measurements available was 83 data points. In this case, 14 data points are taken for testing and the rest are taken as operating data.

The statistics of the data are an important aspect in the development of the ANN. It is important that the different data sets (operating and testing) should have comparable characteristics. In most cases, the ANN is unable to extrapolate beyond the range of the training data [31, 41]. Therefore, the testing subset should fall within the range of the operating data in order for the model to capture the range and variation of the testing data. Therefore, the testing data subset should be selected to be as consistent with the operating as possible. Tables 1 and 2 show the statistics of both the operating and testing data for each of

Table 1 Input variables statistics for both operating and testing data

	Density		Neutron		Resistivity		PE	
	O	T	O	T	O	T	O	T
Mean	2.34	2.33	0.23	0.22	30.98	26.34	2.00	1.92
SD	0.02	0.01	0.02	0.01	7.21	1.18	0.20	0.06

	Mean − SD	Mean + SD	Mean − SD	Mean + SD	Mean − SD	Mean + SD	Mean − SD	Mean + SD
O	2.32	2.36	0.20	0.25	23.77	28.20	1.81	2.20
T	2.32	2.34	0.21	0.22	25.16	27.53	1.86	1.98

‘*O*’ stands for operating data and ‘*T*’ stands for testing data. ‘*SD*’ stands for standard deviation

Table 2 Output variables statistics for both operating and testing data

	Core water saturation	
	O	T
Mean	39.82	43.17
SD	13.01	5.61
	Mean − SD	Mean + SD
O	26.81	52.83
T	37.56	48.78

'*O*' stands for operating data and '*T*' stands for testing data. '*SD*' stands for standard deviation

the input and output variables. From a cursory examination of the tables, one can see that the testing data have statistics similar to the operating data.

After selecting the operating and testing data, pre-processing of the data should take place before introducing them to the model [42]. This process will help to improve the training process and ensure that every parameter will receive equal attention by the network [43]. Pre-processing involves two fundamental elements: data scaling and data transformation [17, 44]. In this case, the data were scaled using the mean and standard deviation method having a mean of zero and unit standard deviation. Other types of scaling were investigated in the optimization step. Data transformation involves applying a normal transform to the data (making them more normally distributed). Some studies have shown that the neural network is unlike other statistical methods that require normal transformation of the data in advance, and the probability distribution of the input is not required to be known beforehand [44, 45]. On the other hand, other work, especially in the area of prediction of the time series, found the transformation to be helpful where normalizing the data in advance helps the network to concentrate on the real problem at hand (minimizing the performance function), producing better results [47, 48]. However, it is hard to prove such a conclusion theoretically [48]. In this work, the data were scaled using the mean and standard deviation method. Other typing of scaling was investigated in the optimization step.

4.3 Network structure

Developing a neural network structure is the most difficult step in ANN modeling. The following four main parameters need to be determined:

- The number of neurons in input and output layers
- The number of hidden layers and the number of neurons in these layers
- Selecting the stopping criteria for training
- Selecting the optimization learning algorithm
- The type of activation functions in hidden and output layers

Developing a network structure is entirely problem dependent as different problems require different structures. The number of neurons in the input and output layers are fixed by the nature of the problem. Therefore, they can be determined from the number of input and output parameters. Determining the number of hidden layers and their neurons is the main task in the network structure. One hidden layer and two hidden layers can be used, depending on the problem complexity and the amount of data available to construct the model. There is no rigid rule in terms of finding the optimum number of neurons in the hidden layer. However, it is crucial that the number of free parameters should be less than the number of operating data. The optimum number of hidden neurons can be obtained by a trial and error method [23], which is explained in detail in Sect. 2.1

In this case, a limited amount of data were available for training and testing the model. One hidden layer was selected to construct the model. The optimum number of hidden neurons ($H_{\max}$) was calculated to be five, by considering the amount of operating data available and the number of free parameters (having operating data of at least twice the number of free parameters). However, this selection was investigated in the optimization step. The default learning algorithm chosen in this study is resilient propagation (PROP) [19]. The tan-sigmoid and linear equations were taken as activation functions for hidden and output layers, respectively [18].

As limited data are available, the cross-validation type cannot be used to stop the network training. In this case, the training cycles (epochs) were used as a stopping criterion. Three hundred epochs were used to train the model. The choice of this number is important, since a larger number may lead to a pattern memorization (not being able to generalize). Figure 5 shows the effect of the number of epochs on the root mean square error (RMSE) on the testing data. As the number of epochs increases above 300, the error on the testing data starts to increase. The reason for this increase can be explained by looking at Fig. 6 that

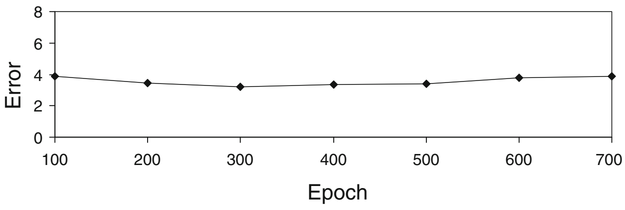

Fig. 5 Error evolution on the testing data with different training cycles

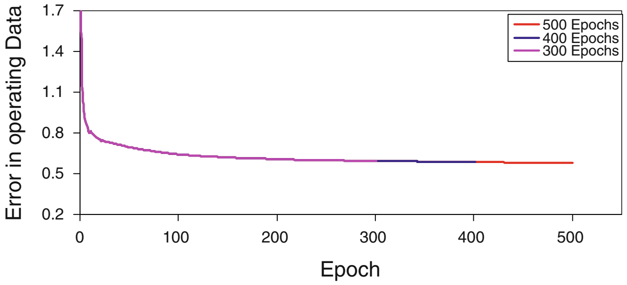

Fig. 6 Error evolution on the operating data with different training cycles

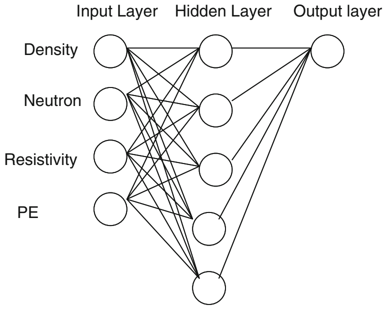

Fig. 7 The structure of the ANN for water saturation prediction

shows the error evolution on the operating data with different training cycles. As the number of epochs increases, the error on the operating data decreases very slowly and the model starts memorizing the noise in the data. Therefore, running the model above 300 epochs will reduce the error on the operating subset, but it will also increase the error on the testing data. Hence, the 300 epochs were used to stop the network from training. The final neural network structure is shown in Fig. 7.

4.4 Model training and results

Once the ANN architecture is determined, the network is ready to be trained and tested. The operating data are used to train the model with the pre-determined optimum number of hidden neurons. Training is performed several times, each with a different weight initialization. This is to ensure starting at a different point in the error surface in order to minimize the effect of local minima. The model is tested with testing data to examine its generalization. The results are analyzed using the root mean square error (RMSE) and the correlation coefficient (r).

The neural network model was able to predict the water saturation with an RMSE of 3.2 S.U. (where saturation is measured in percentage) and a correlation coefficient (r) of 0.83 (between the core measurements and the ANN output). Overall, the ANN is capable of predicting the water saturation with low error, within the uncertainty in the original data in consideration. Figure 8 shows the correlation between the core saturation measurements and the estimated values by the ANN. Most of the data points are located on the unit straight line. Figure 9 shows the comparison between the laboratory measurement and the predicted values of the water saturation using the neural network model. The ANN model estimation closely follows the trend of the core measurements.

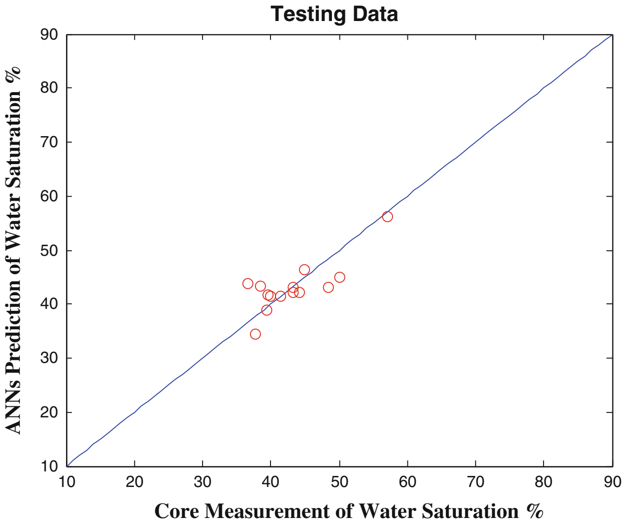

Fig. 8 Correlation between the laboratory measurements (Dean-Stark) and the estimated values from the ANN model

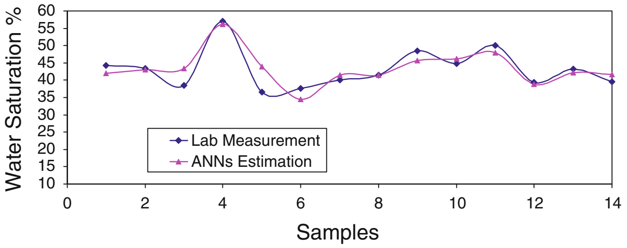

Fig. 9 Comparison between the core measurements (Dean-Stark) and the estimated values from the ANN model

4.5 Model optimization

It is important to run different sensitivity analyses to investigate whether the model can be further optimized. The sensitivity analysis includes the following:

- Testing the optimum number of hidden neurons
- Testing different types of scaling
- Testing the learning algorithm parameters
- Testing different transfer functions
- Testing the stopping criteria

The previous ANN was taken as a base case, and several optimization processes were performed. However, it should

Table 3 The effect of different number of hidden neurons in the ANN for case study 1 and associated error (RSME) and correlation coefficients

Hidden neurons	RMSE	Correlation coefficient (r)
2	4.3	0.6
10	5.6	0.54
50	8.3	0.53
5	3.2	0.83

be taken into consideration that the base case model gave satisfactory results, and this optimization step might not be necessary for this particular case. The optimum number of hidden neurons for the base case was five. The robustness of this selection was investigated by running the model with a different number of neurons in the hidden layer. The model was run with 2, 10, 50 hidden neurons. Table 3 shows the results of these different cases to the testing data. The two hidden neurons produced an error of 4.3 S.U., higher than the base case. As the number of neurons increases compared with those in the base case, the error on the testing data increases. This is because the network starts memorizing the data.

In the base case model, the mean and standard deviation scaling method was used. In the 'min and max' method, the data is scaled to the range of [−1, 1]. This method was also investigated, and it gave similar results to the base case with a RMSE of 3.6 S.U.

Table 4 shows the comparison between the performance of different learning algorithms. LM learning algorithms produced almost the same results as the base case PROP algorithm, followed by conjugate gradient. The normal BP method also gave acceptable results with an RMSE of 4.4 S.U.

The PROP learning algorithm achieves its adaptive weight update through introducing an individual weight update value Δ_{ij}, and this value changes during the training by a certain predefined parameter [19]. A sensitivity analysis was performed on these parameters. A slightly better result than that of the base case was obtained by tuning these parameters (when $\Delta_o = 0.03$ compare 0.07).

Table 4 The error (RSME) on the testing data using different learning algorithms

Learning algorithm	RMSE	Correlation coefficient (r)
PROP (base case)	3.20	0.83
Back-propagation (PB)	4.35	0.5
BP with momentum	4.15	0.68
BP with variable learning rate	3.88	0.73
Conjugate	3.75	0.74
Levenberg–Marquardt	3.30	0.8

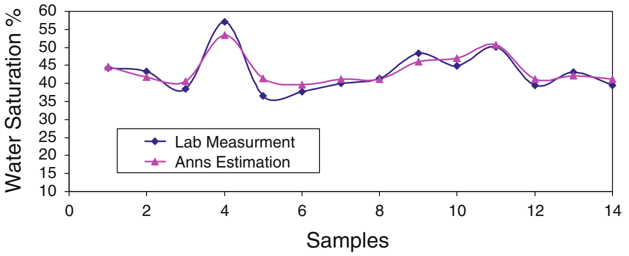

Fig. 10 Comparison between the core measurements (Dean-Stark) and the estimated values from the optimized neural network model (optimized PROP algorithm)

The network was able to predict the water saturation with an error of 2.5 S.U. and a correlation coefficient (r) of 0.91. However, there is not a significant difference in the results for this case compared with the base case, especially with the original uncertainties in the core data. Figure 10 shows a comparison between the core measurements and estimated values by the ANN. The ANN estimation closely follows the trend of the core data. Figure 11 shows the correlation between the core measurement and the estimated values by the ANN model. Most of the values of the ANN estimation are located on a line of unit slop, which shows a good comparison with the real data.

4.6 Contribution of input parameters

Neural networks have the disadvantage of being less transparent compared with other conventional models [39, 46]. To make the ANNs more transparent, it is important to understand the relevance and relative importance of model inputs. There are many methods available to study the

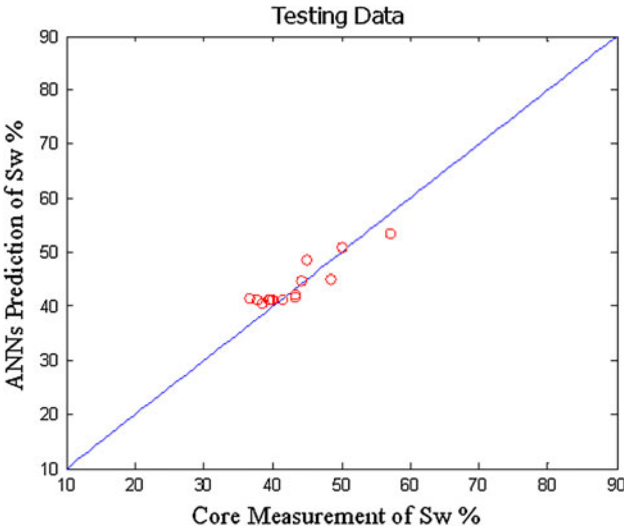

Fig. 11 Correlation between the core measurements (Dean-Stark) and the estimated values from the optimized neural network model (optimized PROP algorithm)

contribution of the variables, such as partial derivatives (PaD) that calculates the partial derivative of the output as a function of the input parameters and the Garson weight method that computes the connection weight between the variables [49–51].

Figure 12 presents the relative contribution of the input parameters from both PaD and Garson method. Both PaD and Garson method resulted in the same conclusion. The resistivity log was the most significant factor for water saturation prediction: oil, rock, and gas do not conduct, while formation brine does—the resistivity is a sensitive measure of the saturation as a consequence. The neutron and density logs give information about the porosity but not saturation directly, and they have almost the same contribution level. Finally, the PE has the lowest impact on the water saturation, without a significant difference from density and neutron contribution levels. Since the PE has the lowest contribution level, a case study was run using only three wireline logs: resistivity, density, and neutron logs. The neural network was still able to predict the water saturation with an error of 3.5 S.U. and correlation coefficient (r) of 0.76.

It is important to check the robustness of the optimized model. One way of doing this is by investigating its capability to predict other reservoir parameters besides the modeled one if the input variables involve other information about the reservoir. Other methods include introducing more noise to the input data and investigating the effect of this on the testing data. Since the input parameters for ANN model in this study include information about lithology, it is expected that the structure trained model for water saturation can also predict the other properties of the formation, in particular the volume of shale. In this step, the optimized ANN model for the water saturation was used to predict the volume of shale (the output this time was to calculate the shale volume). The results showed that the trained neural network model was able to predict the volume of the shale with an error of 2% and a correlation coefficient (r) of 0.84. The generated ANN model has therefore proved its capability to predict water saturation and the volume of shale simultaneously. In this case, the neutron and density logs gave the highest contribution to the model.

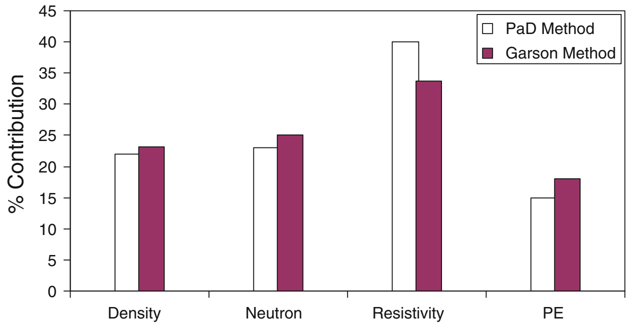

Fig. 12 Input variables contribution to the ANN from Pad and Garson method

4.7 Comparison of the ANN with conventional statistical regression models

It is always important to compare the ANN results with other types of regression model in order to investigate their capability against other simpler techniques. The multiple linear regression method assumes a linear relationship between the variables [52]. On the other hand, the ANNs are known for their ability to process nonlinear relationships. MLR was performed using standard statistical software. Using all four input variables, a correlation coefficient (r) of 0.41 was obtained. By using the stepwise regression, only one variable, the resistivity, was retained by the model, and a correlation coefficient (r) of 0.42 was obtained for this case. Table 5 summarizes the results of the comparison. A nonlinear regression was also tested and did not give satisfactory results. The ANN gives better results than MLR. This is because the relationship between the variables (water saturation and the log data) is highly nonlinear. Figure 13 shows an example of the complex relationship between the neutron log values and core water saturation.

4.8 Using more testing data

In the base case, fourteen data samples were used for testing the model. In this step, the number of the testing

Table 5 Comparison between neural network model and statistical regression method

	Multiple linear regression	Neural network model (ANN)
Correlation coefficient	0.41	0.91

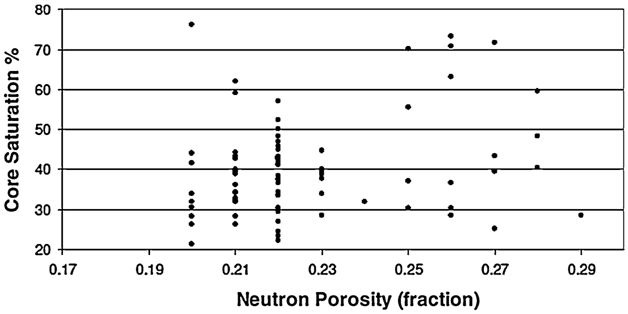

Fig. 13 Relationship between the neutron log and core water saturation

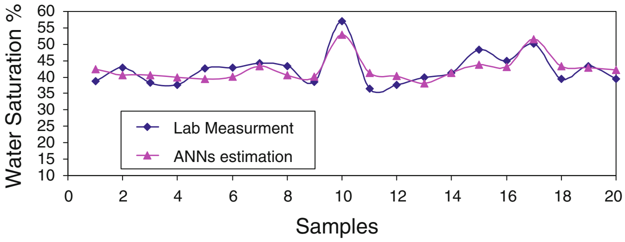

Fig. 14 Comparison between the core measurements and the estimated values from the ANN for new testing data

data was increased to 20 samples (63 samples as operating data). The same development procedure as that of the base case was followed. The network was able to predict the water saturation with an error of 3.6 S.U. and a correlation coefficient (r) of 0.7 (between the core measured and ANN estimation). Figure 14 shows the comparison between the laboratory measurements and the ANN estimation. The model closely follows the trend of the core data. This performance is almost identical to the base case, indicating that sufficient samples were retained for training.

4.9 Conclusions

In this paper, a neural network workflow (methodology) was developed. This workflow covers a range of design issues related to ANN development. The workflow was used to develop an ANN model for water saturation prediction in a petroleum field. Wireline logs are abundantly available in most of the drilled wells in an oilfield. The core data, which gives an accurate determination of water saturation, is only available in specific wells. The ANN was used to find the complex nonlinear relationship between wireline logs and core saturation data.

The results showed that the optimized ANN model successfully predicted the water saturation with a correlation factor (r) of 0.91 (between the core measurement and ANN output) and a root mean square error of 2.5 saturation units to the testing data, this is within the error of the measurements used to train the ANN. Several sensitivity analyses were performed to investigate the robustness of the selected model structure. The analyses included varying the number of hidden neurons, different scaling methods, changing the transfer function and investigating different learning algorithms. The resistivity log was the most important factor in the developed model. Furthermore, the ANN was superior to conventional statistical models (such as multiple regressions).

Appendix [52, 53]

Wireline logs are continuous measurements of downhole formation through electrical instruments. The logging is performed during and after drilling a well where the tool is lowered to the formation through electrical cables. The interpretation of wireline logs provide indirect valuable information of the formation that has been drilled, such as lithology, porosity, saturation, and permeability. There are many tools used for wireline logging, such as gamma ray detector, density and neutron, resistivity measurement, and sonic travel time.

Wireline logs are continuous electrical measurements of downhole formation through electrical instruments. The measurements are performed at each depth of required formation, typically at every 150 cm provided through a long band of paper. The logging is performed during and after drilling a well where the tool is lowered to the formation through electrical cables. The interpretation of wireline logs provide indirect valuable information of the formation which has been drilled, such as lithology, porosity, saturation, and permeability. There are many tools used for wireline logging, such as gamma ray detector, density and neutron, resistivity measurement, and sonic travel time.

The gamma ray tool is a passive logging tool that detects the natural radiation of gamma rays from the formation, which are result of high-energy electromagnetic radiation [53, 54]. The gamma ray tool gives information about the lithology of the formation where high gamma rays are related to shaly environment, whereas low readings are interpreted as clean sands. The density log tool emits gamma ray into the formation that collides with the electrons in the formation [53, 54]. In the process, the gamma rays are attenuated. The counts rates of the scatted gamma ray at a fixed distance from the source are inversely related to the electron density of the formation; consequently, the bulk density of the formation can be calculated. The photo-electric absorption index gives information about the lithology of the formation where the photo-electric measurement primarily response to the rock matrix [53, 54]. The neutron logging tool bombards the formation with high-energy neutron. The high-energy neutron interact with the nucleus of the atoms in the formation where each interaction causes lose of neutron energy [53, 54]. The hydrogen atoms has the same mass of the neutron, hence lowers the speed of the neutron significantly. The slowing down rate is determined by the hydrogen index of all components of the formation and formation fluids that contact a significant fraction of hydrogen. The distance over which the neutrons have traveled before they reach a lower-energy level is related to the amount of the hydrogen atoms present in the formation. A combination of density and neutron tool gives indication of the lithology of the formation besides the gamma ray. The resistivity logging tool basically measures the resistivity of the formation. By measuring the resistivity, the water saturation can be calculated.

The coring and core analysis provide direct measurements of petrophysical properties in the laboratory. The core basically represents a whole section of rock extracted from the drilled formation. In the laboratory, samples from the core are taken for physical measurements such as porosity, saturation, and permeability. The physical properties from the core analysis represent the ground truth, and they are compared to the wireline logs calculated petrophysical properties. However, the core is an expensive method and limited only to few wells in the formation, whereas the wireline logs are abundantly available in most of the wells in the formation. Water saturation can be determined directly from cores taken from a well in the field. The widely used laboratory method to determine the water saturation is the Dean-Stark method. In this method, the fluid saturation is determined by distillation of the water fraction and extraction of the oil fraction from a sample. The process starts by vaporizing the water in the sample by boiling the solvent. This vaporized water is then condensed and collected in a calibrated trap. At this stage, the volume of the water in the sample can be determined. The solvent is also condensed and then flows back over the sample to extract the oil. After the water and oil have been removed, the sample is dried. The weight of oil is calculated by the difference between the total loss in the sample weight and water weight removed from it. The loss in the sample weight is calculated by measuring the weight of the sample before and after extraction.

References

1. Ali JK (1994) Neural network: a new tool for petroleum industry. Proceedings of SPE European petroleum computer conference, UK, paper 27561
2. Olson TM (1998) Porosity and permeability prediction in low-permeability gas reservoirs from well logs using neural networks. Proceedings of SPE rocky mountain regional/low-permeability reservoir symposium and exhibition, USA, paper 39964
3. White AC, Molnar D, Aminian K, Mohaghegh S, Ameri S (1995) The application of ANN for zone identification in a complex reservoir. Proceedings of the SPE Eastern regional conference & exhibition, USA, paper 30977
4. Bhatt A, Helle HB (2002) Determination of facies from well logs using modular neural networks. Pet Geosci 8:217–228
5. Helle HB, Bhatt A (2002) Fluid saturation from well logs using committee neural networks. Pet Geosci 8:109–118
6. Al-Bulushi N, King P, Blunt M, Kraaijveld M (2009) Development of articial neural network for predicting water saturation and fluid distribution. J Petrol Sci Eng 68:197–208
7. Archie GE (1941) The electrical resistivity log as an aid in determining some reservoir characteristics. Trans AIME 146:54–62
8. Waxman MH, Smits LJM (1968) Electrical conductivities in oil-bearing shaly sands. Soc Petrol Eng J 18(2):107–122
9. Leverett MC (1941) Capillary behavior in porous solids. Trans AIME 146:152–169
10. Worthington PE (1985) The evolution of shaly sand concepts in reservoir evaluation. The Log Analyst 26(1):23–40
11. Ipek I (2002) Log-derived cation exchange capacity of shaly sands: application to hydrocarbon detection and drilling optimisation. Doctoral diss., Louisiana State University
12. Rezaee MR, Lemon NM (1996) Petrophysical evaluation of kaolinite-bearing sandstone: Water saturation (S_w), an example of the Tirrawarra sandstone reservoir, Copper Basin. Proceedings of the SPE Asia pacific oil & gas conference, Australia, paper 37023
13. Haykin S (1998) neural networks, a comprehensive foundation. Prentice Hall PTR, 2nd edn. USA
14. Essenreiter R, Karrenbach M, Treite S (1998) Multiple reflection attenuation in seismic data using back-propagation. IEEE Trans Signal Process 46(7):2001–2011
15. Maren AJ, Harston CT, Pap RM (1990) Handbook of neural computing applications. Academic Press, New York
16. Hush DR, Horne BG (1993) Progress in supervised neural networks. IEEE Signal Process Mag 10(1):8–39
17. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press
18. Hagan M, Demuth HB, Beale M (1996) Neural network design. PWS Publishing Company, USA
19. Reidmiller M, Braun H (1993) A Direct adaptive method for faster back-propagation learning: the PROP algorithm. Proceedings of the IEEE international conference on neural network, pp 586–591
20. Cun YLe, Denker JS, Solla SA (1990) Optimal brain damage. Adv Neural Network Process Syst 2:168–177
21. Hush DR, Horne B, Salas JM (1992) Error surfaces for multilayer perceptrons. IEEE Trans Syst Man Cybern 22 (5)
22. Baum EB, Haussler D (1990) What size net gives valid generalization? Neural Comput 1(1):151–160
23. Hush DR (1989) Classification with neural networks: a performance analysis. IEEE Int Conf Syst Eng, pp 227–280
24. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximates. Neural Netw 2:359–366
25. Irie B, Miyake S (1998) Capabilities of there-layered perceptron. Proc IEEE Int Conf Neural Netw, pp 641–648
26. Hecht-Nielsen R (1990) Neurocomputing. Addison-Wesley, New York
27. Huang SC, Huang YF (1991) Bounds on the number of hidden neurons in multilayer perceptrons. IEEE Trans Neural Netw 2(1): 47–55
28. Cheng B, Titterington DM (1994) Neural networks: a review from a statistical perspective. Stat Sci 9(1):2–54
29. Chester DL (1990) Why two hidden layers are better than one. Neural Netw J 1:265–268
30. Flood I, Kartam N (1994) Neural networks in civil engineering. I: Principles and understanding. J Comput Civil Eng 8(2):131–148
31. Maier HR, Dandy GC (2000) Neural Networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. J Environ Modeling Softw 15:101–124
32. Moody JE (1992) The effective number of parameters: an analysis of generalization and regularization in non-linear learning system. Adv Neural Netw Process Syst 4:841–854
33. Rogers LL, Dowla FU (1994) Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. Water Resour Res 30(2):457–481
34. Bann VDM, Jutten C (2000) Neural networks in geophysical applications. Geophysics 65(4):1032–1047
35. Masters T (1993) Practical neural network recipes in C ⋚⋚. Academic Press
36. Amari S, Murata N, Muller KR, Finke M, Yang H (1997) Asymptotic statistical theory of overtraining and cross-validation. IEEE Trans Neural Netw 8(5):985–996

37. Hanson SJ, Pratt L (1989) Comparing biases for minimal network construction with back-propagation. Advances in neural information processing systems 1, (Morgan Kaufmann Publishers Inc, pp 177–185)
38. Lachtermacher G, Fuller JD (1994) Back-propagation in hydrological time series forecasting, stochastic and statistical methodology in hydrology and environmental engineering. In Hipel KW, McLeod AI, Panu US, Singh VP (eds) Kluwer Academic Publisher, Dordrecht, pp 229–242
39. Goda HM, Maier HR, Behrenbruch P (2005) The development of an optimal artificial neural network model for estimating initial, irreducible water saturation- Australian reservoirs. Proceedings of the SPE Asia Pacific Oil & Gas conference and Exhibition, Indonesia, paper 93307
40. Stunder M, Al-Tuwaini JS (2001) How data-driven modeling like neural networks can help to integrate different types of data into reservoir management. Proceedings of the society of petroleum engineers middle east oil show, paper 68163
41. Flood I, Kartam N (1994) Neural networks in civil engineering. I: Principles and understanding. J Comput Civil Eng 8(2):131–148
42. Burden FR, Brereton RG, Walsh PT (1997) Cross-validatory selection of test and validation sets in multivariate calibration and neural networks as applied to spectroscopy. Analyst 122(10): 1015–1022
43. Kaastra I, Boyd MS (1995) Forecasting futures trading volume using neural networks. J Futures Mark 15(8):953–970
44. Tukey JW (1977) Exploratory data analysis. Addison-Wesley, New York
45. Burke LI, Ignizio JP (1992) Neural networks and operations research: an overview. Comput Oper Res 19:179–189
46. Faraway J, Chatfield C (1998) Time series forecasting with neural networks: a comparative study using the airline data. Appl Stat 14(2):109–250
47. Fortin V, Ouarda TBMJ, Bobee B (1997) Comment on ‘‘the use of artificial neural networks for the prediction of water quality parameters’’ by Maier HR and Dandy GC. Water Resour Res 33(10):2423–2424
48. Shi JJ (2000) Reducing prediction error by transforming input data for neural networks. J Comput Civil Eng 14(2):109–166
49. Dimopoulos I, Chronopoulos J, Chronopoulou-Sereli A, Lek Sovan (1999) Neural network models to study lead concentration in grasses and permanent urban descriptors in Athens city (Greece). Ecol Modell 120:157–165
50. Gevery M, Dimopoulos I, Lek Sovan (2003) Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecol Modell 160:249–264
51. Garson D (1991) Interpreting neural network connection weights. AL Expert l6(4): 46
52. Lek S, Delacoste M, Baran P, Dimopulos I, Lauga J, Auglagnier S (1996) Application of neural network to modelling nonlinear relationships in ecology. Ecol Modell 90:9–52
53. Schlumberger (1991) Log interpretation principles/application. Schlumberger educational series seventh printing
54. Schlumberger oil glossary http://www.glossary.oilfield.slb.com