PERGAMON

Expert Systems with Applications 15 (1998) 405–411

Expert Systems with Applications

Ranking importance of input parameters of neural networks

A.H. Sung*

Department of Computer Science, New Mexico Institute of Mining and Technology, Socorro NM 87801, USA

Abstract

Artificial neural networks have been used for simulation, modeling, and control purposes in many engineering applications as an alternative to conventional expert systems. Although neural networks usually do not reach the level of performance exhibited by expert systems, they do enjoy a tremendous advantage of very low construction costs.

This paper addresses the issue of identifying important input parameters in building a multilayer, backpropagation network for a typical class of engineering problems. These problems are characterized by having a large number of input variables of varying degrees of importance; and identifying the important variables is a common issue since elimination of the unimportant inputs leads to a simplification of the problem and often a more accurate modeling or solution.

We compare three different methods for ranking input importance: sensitivity analysis, fuzzy curves, and change of MSE (mean square error); and analyze their effectiveness. Simulation results based on experiments with simple mathematical functions as well as a real engineering problem are reported.

Based on the analysis and our experience in building neural networks, we also propose a general methodology for building backpropagation networks for typical engineering applications. © 1998 Elsevier Science Ltd. All rights reserved

Keywords: Artificial neural networks; Importance ranking; Sensitivity analysis; Fuzzy curves; Change of MSE

1. Introduction

Artificial neural networks have found widespread applications in diverse areas as a practical tool for modeling simulation, control, and prediction. Especially prevalent in various engineering applications are the backpropagation (BPNs) networks, which are typically trained with data or patterns collected during actual operations. After a BPN is adequately trained, it can be used in field operations for the purposes of control and/or prediction of the operational parameters. Many applications of this kind can be found in the literature, for some of the more recent ones, see e.g. Dagli et al. (1997). A common characteristic of such applications where neural networks are used as an alternative of expert systems is that the correctness of answers generated by the system is crucial while the facilities for reasoning and explanation of conventional expert systems are not essential.

An advantage of using neural networks is that they often can be quickly constructed using available data at a very low cost when compared with developing conventional expert systems. The saving in time and cost is achieved by replacing the process of knowledge acquisition and knowledge base construction with the process of training networks. Another, perhaps more significant, advantage is that neural networks can learn from examples and make predictions for new situations. Therefore, neural networks can often be trained to solve a problem once a sufficient amount of representative data becomes available to constitute a good training set, even before the problem is fully understood or before human experts are able to formulate their knowledge in an organized, complete and consistent manner to allow an expert system solution (Hertz et al., 1991; Zurada, 1992).

This paper addresses the issue of identifying important input parameters of a BPN. Since the ability to quickly identify the important inputs and the redundant inputs of a network leads directly to reduced size, faster training, and possibly more accurate results of the network, it is an issue of great practical as well as theoretical importance. We compare three different methods that have been proposed — sensitivity analysis, fuzzy curves, and change of MSE (mean square error) — and analyze their effectiveness on small BPNs based on simple nonlinear functions as well as a production use BPN developed for a petroleum engineering application.

The sensitivity analysis method uses a sensitivity matrix

* Corresponding author. Tel.: 001 505 835 5949; Fax: 001 505 835 5587; E-mail: sung@nmt.edu

0957–4174/98/$ - see front matter © 1998 Elsevier Science Ltd. All rights reserved
PII: S0957-4174(98)00041-4

calculated based on the partial derivatives of outputs of the network with respect to each input. We extend the work of Zurada et al. (1994), and Engelbrecht et al. (1995), and derive the sensitivity formula for the general BPNs comprising three layers.

The fuzzy curves method of Lin and Cunningham (1995) uses a fuzzy logic based technique to identify the importance of input variables of a BPN. For each input, a fuzzy membership function which has exponential form is defined for each training pattern, and then centroid defuzzification is used to produce a fuzzy curve for each input parameter. The importance of inputs are ranked according to the ranges of the fuzzy curves.

The change of MSE (mean square error) method of He et al. (1996, 1997) is an ad hoc method based on measuring the change of MSE when an input is deleted during training. For example, if the deletion of an input results in a larger change of the MSE, the input may be considered more important. This method has been used to build BPNs used for modeling and predicting cement bonding quality in oil wells (He et al., 1996, 1997).

A fast BPN training algorithm — the scaled conjugate gradient (SCG) algorithm of Moller (1993) was implemented to build a collection of networks to experiment with the three methods of identifying input parameter importance.

We also report our experience in developing BPNs for predicting oil well cement bonding quality, which is a problem occurring in petroleum engineering (He et al., 1996, 1997), and propose a methodology for building BPNs for such typical engineering applications.

2. Ranking significance of inputs

In this section, we compare three techniques of identifying important inputs. Sensitivity analysis and change of MSE methods require training neural networks. The fuzzy curves technique only analyzes the relationship between input data and output data. For simplicity, we consider only systems with a single output; multiple-output systems can be analyzed as a collection of single-output systems.

2.1. Sensitivity analysis (SA)

The sensitivity coefficient matrix for a one-hidden-layer BPN was derived (Zurada et al., 1994; Engelbrecht et al., 1995). However, two hidden layers are generally required to approximate arbitrary functions to a desired accuracy (Hertz et al., 1991). In the following, we derive the sensitivity coefficient matrix for a BPN with two hidden layers.

Consider a neural network with two hidden layers of nodes $Z = (z_1,z_2,\ldots,z_I)$ and $Y = (y_1,y_2,\ldots,y_J)$. The input layer is $X = (x_1,x_2,\ldots,x_L)$, and the output layer is $O = (o_1,o_2,\ldots,o_K)$ (Fig. 1). L, I, J, and K are, respectively, the number of nodes of the input layer, two hidden layers and the output layer.

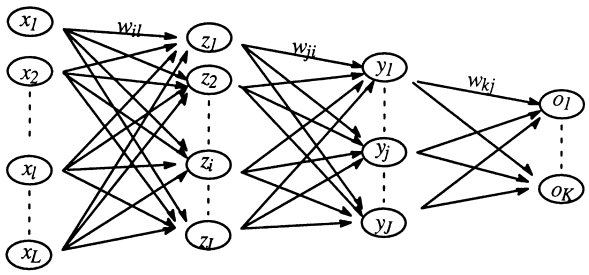

Fig. 1. Neural network with two hidden layers.

Given a training pattern p, the sensitivity of the input x_l with respect to the output o_k can be defined by:

$$\mathbf{S}^p_{k,l} \doteq \frac{\partial o_k}{\partial x_l}$$

$$= o'_k \sum_{j=1}^{J} \left(w_{kj} y'_j \sum_{i=1}^{I} (w_{ji} z'_i w_{il}) \right)$$

Where o',y' and z' are the corresponding derivative values of the activation function. ws are the weights in the neural network (Fig. 1). the sensitivity of any input with respect to any output in a pattern p can be defined by extending the above equation to obtain the following:

$$\mathbf{S}^p \doteq \mathbf{O'W_{KJ}Y'W_{JI}Z'W_{IL}}$$

where $\mathbf{W_{KJ}}$, $\mathbf{W_{JI}}$ and $\mathbf{W_{IL}}$ are the weight matrices. $\mathbf{O}'(K \times K)$, $\mathbf{Y}'(J \times J)$ and $\mathbf{Z}'(I \times I)$ are diagonal matrices defined as:

$$\mathbf{O}' \doteq diag(o'_1, \ldots, o'_K)$$

$$\mathbf{Y}' \doteq diag(y'_1, \ldots, y'_J)$$

$$\mathbf{Z}' \doteq diag(z'_1, \ldots, z'_I)$$

The above equation only defines the sensitivities of inputs for a given training pattern. For a set of training patterns, Zurada et al. define three methods to calculate the sensitivity of each input. In our analysis, we use the absolute value average sensitivity matrix defined as:

$$\mathbf{S}^{avg}_{k,l} \doteq \sum_{p=1}^{p} |S^p_{kl}|/P$$

where P is the total number of the training patterns.

2.2. Fuzzy curves (FC)

The fuzzy curves method of Lin and Cunningham (1995) uses fuzzy logic to establish the relationship between the input variables and an output variable. Suppose a system has N inputs $X = (x_1,x_2,\ldots,x_N)$, and one output $Y = (y)$, and the number of training patterns is P. Let x_{ik} be the i^{th} variable in the $k^{\text{th}}(k \leq P)$ pattern, and y_k is the output value in the k^{th} pattern. The algorithm is briefly described as follows:

- Calculate the fuzzy membership function ϕ_{ik} for each input x_i in the k^{th} pattern. For P training patterns, we have P fuzzy rules for each input.

$$\phi_{ik}(x_i) = \exp\left(-\left(\frac{x_{ik} - x_i}{b} \right)^2 \right),\ k = 1, 2, \ldots, P;\ i = 1, 2 \ldots, I$$

- Use the center of area method for defuzzification to generate a fuzzy curve c_i for each input variable x_i:

$$C_i(x_i) = \sum_{k=1}^{P} \phi_{ik}(x_i) y_k \,/\, \sum_{k=1}^{P} \phi_{ik}(x_i)$$

- Plot the fuzzy curves in the $x_i - C_i\ (x_i)$ space. If the fuzzy curve for an input is flat, this input does not have significant influence on the outcome and so it is not an important input. The importance of inputs can be ranked by their ranges covered by their respective fuzzy curves.

2.3. Change of MSE (COM)

Another method to measure the importance of an input is to measure the change of mean square error (MSE) when that input is deleted from the neural network. The *MSE* is defined as:

$$MSE = \sum_{k=1}^{K} \sum_{p=1}^{P} (t_{kp} - o_{kp})^2 / P$$

where t_{kp} and o_{kp} are the desired output and the calculated output, respectively, of the k^{th} output neuron for the p^{th} pattern; K is the total number of output nodes and P is the total number of patterns.

In the COM method, we retrain the neural network with $N - I$ inputs each time after an input is deleted and observe the change in MSE (relative to the MSE of the neural network trained with all the inputs included), where N is the number of original input parameters. Based on the net change in MSE, we can rank the importance of input variables in several different ways based on different arguments. For instances: we can rank those inputs whose deletion cause the largest change in MSE as the most important since the error is most sensitive to these inputs; or we can rank those whose deletion cause the largest decrease in MSE as the most important since they are most relevant to the construction of a network with a small MSE. We will discuss these two different COM-based ranking methods later.

3. Examples

In order to understand the effectiveness of the three approaches in evaluating the significance of inputs, we first examine two simple nonlinear functions which re modeled by BPNs. Both functions include independent variables x_1 and x_2. Additionally, the inputs of the neural networks can also include a noisy input x_3 and/or a redundant (or dependent) input x_4.

Example 1. The first example employs the following nonlinear equation:

$$Y = (x_1 + 2)^2 + \sin(2\pi x_2)\ 0 \le x_1, x_2 \le 1$$

This is an additive function whose value is the sum of two functions, each of them only involves one variable. Assume that x_1 and x_2 are independent, obviously x_1 is more important than x_2 since $(x_1 + 2)^2$ dominates the value of the function while $\sin(2\pi x_2)$ only generates a small fluctuation. The average sensitivity of x_1 in the entire input domain is 5, and for x_2 its average sensitivity is 4.

A collection of BPNs were then built to model this nonlinear function. In order to accelerate the training, we use the scaled conjugate gradient algorithm of Moller (1993) to train the BPNs. Our experience with previous projects is that the SCG algorithm is a backpropagation training algorithm having superior performance.

The following are the simulation results:

$$\textit{Function}: Y = (x_1 + 2)^2 + \sin(2\pi x_2)\ 0 \le x_1, x_2 \le 1$$

$$x_3:\ \textit{noisy input}.\ 0 \le x_3 \le 1$$

$$x_4:\ \textit{dependent (or redundant) input}.$$

$$\text{Correct importance order}:\ (x_1, x_4) > x_2 x_3$$

The simulation results of sensitivity analysis (SA) and fuzzy curves (FC):

Cases	Input variables	Other constraints	SA Importance order	FC Importance order
1	(x_1,x_2)	none	$x_1 > x_2$	$x_1 > x_2$
2	(x_1,x_2)	$x_1 = x_2$	$x_1 > x_2$	$x_1 = x_2$
3	(x_1,x_2,x_3)	none	$x_1 > x_2 > x_3$	$x_1 > x_2 > x_3$
4	(x_1,x_2,x_4)	$x_4 = 1 - x_1{*}x_1$	$x_1 > x_2 > x_4$	$x_1 > x_4 > x_2$
5	(x_1,x_2,x_3,x_4)	$x_4 = 1 - x_1{*}x_1$	$x_1 > x_2 > x_4 > x_3$	$x_1 > x_4 > x_2 > x_3$
6	(x_1,x_2,x_3,x_4)	$x_4 = 1 - x_1{*}x_2$	$x_1 > x_2 > x_4 > x_3$	$x_1 > x_4 > x_2 > x_3$

The simulation results of the COM method follow:
Case 7: input variables (x_1,x_2,x_3)

Variable deleted	COM after deleting
x_1	1.78×10^{-4}
x_2	4.6×10^{-5}
x_3	1.8×10^{-5}

Importance order: $x_1 > x_2 > x_3$
Case 8: inputs (x_1,x_2,x_4) $x_4 = 1 - x_1^2$

Variable deleted	COM after deleting
x_1	2.19×10^{-4}
x_2	2.6×10^{-6}
x_4	1.36×10^{-4}

Importance order: $x_1 > x_4 > x_2$

Example 2. The second example employs the following

function:

$$Y = ((x_1 + 2)^{1.5} + \sin(2\pi x_2))^2 \quad 0 \le x_1, x_2 \le 1$$

This is not an additive function. The sensitivity of a variable also depends on the other variable (i.e. each of the partial derivatives $\partial Y/\partial x_1$ and $\partial Y/\partial x_2$ is a function of both x_1 and x_2). However, the calculations of sensitivity analysis would indicate that input x_1 is more important than x_2. The average sensitivity of x_1 in the input domain is 43.70, and 35.78 for x_2. Again we train a collection of BPNs which may involve a noisy input and a redundant input; and then use SA, FC and COM to measure the importance of inputs. The following are the simulation results:

Function : $Y = ((x_1 + 2)^{1.5} + \sin(2\pi x_2))^2 0 \le x_1, x_2 \le 1$

x_3 : *noisy input*. $0 \le x_3 \le 1$

x_4 : *dependent input*.

Correct importance order : $(x_1, x_4) > x_2 > x_3$

The simulation results of SA and FC are:

Cases	Input variables	Other constraints	Importance order ranked by SA	Importance order ranked by FC
10	(x_1,x_2)		$x_2 > x_1$	$x_1 > x_2$
11	(x_1,x_2)	$x_1 = x_2$	$x_2 > x_1$	$x_1 = x_2$
12	(x_1,x_2,x_3)		$x_2 > x_1 > x_3$	$x_1 > x_2 > x_3$
13	(x_1,x_2,x_4)	$x_4 = 1 - x_1{*}x_1$	$x_2 > x_1 > x_4$	$x_4 > x_1 > x_2$
14	(x_1,x_2,x_3,x_4)	$x_4 = 1 - x_1{*}x_1$	$x_2 > x_1 > x_4 x_3$	$x_4 > x_1 > x_2 > x_3$
15	(x_1,x_2,x_3,x_4)	$x_4 = 1 - x_1{*}x_2$	$x_2 > x_1 > x_4 > x_3$	$x_1 > x_2 > x_4 > x_3$

The simulation results of the COM method follow:
Case 16: input variables (x_1,x_2,x_3)

Variable deleted	COM after deleting
x_1	2.75×10^{-4}
x_2	2.58×10^{-4}
x_3	3.7×10^{-5}

Importance order: $x_1 > x_2 > x_3$
Case 17: inputs (x_1,x_2,x_4) $x_4 = 1 - x_1^2$

Variable deleted	COM after deleting
x_1	1.06×10^{-4}
x_2	5.5×10^{-5}
x_4	$6.8 \times {}^{-5}$

Importance order: $x_1 > x_4 > x_2$
Case 18: inputs (x_1,x_2,x_3,x_4) $x_4 = 1 - x_1^2$

Variable deleted	COM after deleting
x_1	7.4×10^{-5}
x_2	2.36×10^{-4}
x_3	1.27×10^{-4}
x_4	1.85×10^{-4}

Importance order: $x_2 > x_4 > x_3 > x_1$

4. Results and comparison

From the results of the examples in Section 3, the following observations are drawn:

1. The SA and the FC methods both correctly detect the noisy input x_3 as the least important input (Cases 3 and 12). Since it is hard to discover a relationship between the noisy input and the actual outputs for any of the neural networks trained with good training data, it is not surprising that both methods work well. When the noisy input and the redundant input are not both present in the input set, the COM method can also correctly determine the importance order of inputs (Cases 7, 8, 16 and 17). However, when the noisy input and the redundant input are both present, the COM method incorrectly determines the input importance order (Case 18).
2. For the first function, SA gives the correct conclusion that x_1 is more important than x_2 (Cases 3–6). However, for the second function, it gives a wrong conclusion that x_2 is more important than x_1 (Cases 12–15).
3. The FC method performs very well in identifying the importance of the inputs. One drawback of the FC method, however, is that when the training data are not representative samples of the input space it cannot distinguish the significance of inputs. In Case 2, when x_1 and x_2 have the same value in every sample, the FC method suggests that they are equally important.
4. In Cases 4 and 13, the dependent input x_4 and the relevant input x_1 should be almost equally important. However, the SA method cannot determine this.
5. Correctly determining the importance of all inputs does not imply that we can detect the dependent inputs. Except Cases 2 and 15, FC well characterizes the importance order of dependent inputs in all cases.
6. Finally, it should be noted that other methods have been tried to identify important inputs, see, for example, Bartlett (1994). However, no method can successfully identify dependent inputs in all cases because, generally, a neural network is trained to model a continuous function with only a finite set of training data; so a dependence can be established between any two or among any three or more variables.

The FC method performs better than SA or COM in most cases of our experiments, if the training samples are representative.

5. An engineering application

In oil well drilling, the quality of cement bonding (CBQ) between well casing and formation rock (depicted in Fig. 2), is very important because good CBQ ensures the success of oil and gas exploitation from a reservoir. Since CBQ is affected by as many as 25 operational parameters (see

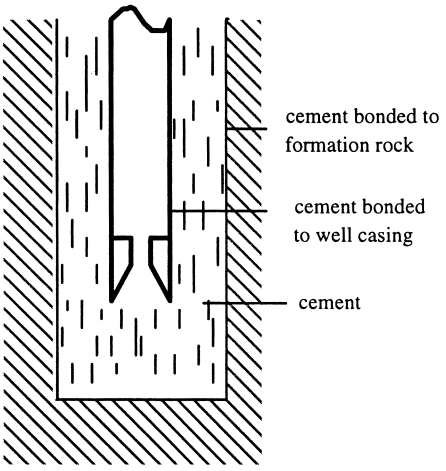

Fig. 2. Cement bonding between well casing and formation rock.

Table 1 for a complete listing), such as formation pressure, well bore pressure, properties of drilling fluid and cement slurry (Smith, 1984), it is essential to control these parameters properly in order to obtain good CBQ. Conventionally, the control of parameters is done according to knowledge and experience obtained from field cementing practices and laboratory evaluation. Although such an ad hoc approach can usually meet practical requirements, it is preferable to have a method that can correctly predict the CBQ from a given set of operation parameters whose values are known or assumed. Once such a method becomes available, by calculating the CBQ for different combinations of operation parameters, the optimal combination can be found (to maximize CBQ or to minimize cost).

Because the relation between CBQ and the operational parameters is evidently complex and nonlinear, and expert system solutions are bound to be time-consuming and costly, therefore, neural networks seem naturally suited to the problem. In He et al. (1996, 1997), a collection of BPNs were trained to model and predict the oil well cement bonding quality. They used a different version of the COM method to identify the least important inputs (those inputs whose deletion cause the largest decrease in MSE are deemed least important). The performance of networks trained with the least important inputs deleted was then compared with networks obtained through a linear combination of networks, and networks whose training data was selected from self-organized clusters using a Kohonen network, etc. The performance of networks with different topology was also compared. The issue of how to partition the data set into training data and testing data was also addressed experimentally.

Here we use SA and FC with the same training data set (as was used in He et al., 1996, 1997) to determine the significance of inputs and compare the results with that of COM. The results are listed in Table 2 (with the 10 most important of the 25 input variables identified by each method shown) and Fig. 3. According to the field expert's opinion, the first 10 variables (x_1–x_{10}) are the most important of all inputs. From Table 1, we observe that the FC method gives a ranking of the importance of inputs that is most consistent with

Table 1
The 25 inputs and one output of the cement-bonding quality modeling and prediction neural network

Input variables	Ranges
x_1: Mud density	1.00–2.00
x_2: Apparent mud viscosity	40.0–50.0
x_3: Slurry density	1.00–2.00
x_4: Apparent slurry consistency	10.0–20.0
x_5: Thickening time	2.00–3.00
x_6: Over-balance pressure	10.0–20.0
x_7: Drilling time	1.00–11.0
x_8: Injection rate	0.00–2.00
x_9: Displacing rate	0.00–2.00
x_{10}: Over slurry return	50.0–90.0
x_{11}: Bit size	7.00–14.0
x_{12}: Calliper	40.0–50.0
x_{13}: Casing size	4.00–8.00
x_{14}: Well depth	800–1200
x_{15}: Stabilizer spacing	20.0–50.0
x_{16}: Mud yield point	20.0–60.0
x_{17}: Mud filtration	3.00–6.00
x_{18}: Mud solid content	9.00–13.0
x_{19}: Slurry water loss	9.00–17.0
x_{20}: Slurry final setting	3.00–6.00
x_{21}: Casing moving velocity	0.00–1.00
x_{22}: Annulus pressure held	50.0–150
x_{23}: Time of running casing	5.00–17.0
x_{24}: Cycles of mud circulation	1.00–4.00
x_{25}: Rate of mud circulation	200–600
Output: CBQ index	0.00–1.00

Table 2
Ranking of input importance by three different methods

Var.	SA(order)	COM(order)	FC(order)
x_1	0.217	0.0017	0.201(6)
x_2	3.541(1)	0.0012	0.195(8)
x_3	0.159	0.0017	0.212(3)
x_4	0.319	0.0026(7)	0.249(1)
x_5	1.001(8)	0.0018	0.240(2)
x_6	0.391	0.0010	0.193(9)
x_7	0.629	0.0023	0.189(10)
x_8	1.230(6)	0.006	0.202(5)
x_9	1.005(7)	0.0007	0.198(7)
x_{10}	0.776(10)	0.0006	0.206(4)
x_{11}	0.964(9)	0.0028(4)	0.005
x_{12}	1.298(5)	0.0030(2)	0.085
x_{13}	0.527	0.0007	0.005
x_{14}	0.561	0.0025(9)	0.048
x_{15}	2.035(2)	0.0030(3)	0.050
x_{16}	0.496	0.0019	0.047
x_{17}	0.068	0.0023	0.002
x_{18}	0.309	0.0024	0.033
x_{19}	0.618	0.0027(5)	0.029
x_{20}	0.326	0.0026(8)	0.006
x_{21}	0.744	0.0027(6)	0.035
x_{22}	1.582(4)	0.0022	0.050
x_{23}	0.504	0.0020	0.054
x_{24}	0.389	0.0035(1)	0.027
x_{25}	1.621(3)	0.0025(10)	0.056

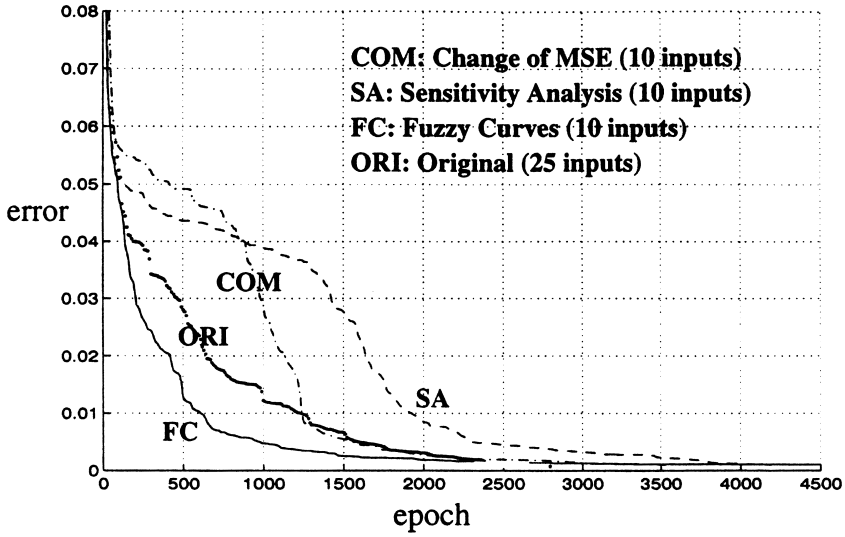

Fig. 3. Plot of error vs time in different training processes.

the expert's opinion. After pruning the 15 unimportant inputs and training new BPNs with a 10-10-4-1 architecture, we found that the new BPNs converge to the same error tolerance as the original BPN; see Fig. 3 and Table 3.

6. Summary and conclusions

The main objective of this paper is to analyze the effectiveness of three methods of ranking the importance of input parameters of BPNs. We derived a sensitivity analysis formula for BPNs with two hidden layers and compare it with the method of fuzzy curves and the method based on MSE changes.

A large class of modeling, control, and prediction problems in chemical, petroleum, and process engineering, etc., among which the CBQ problem appears fairly typical, is characterized by having a large number of input parameters of varying degrees of importance. Neural networks provide a feasible solution to such problems when an analytical model does not exist or is too complex to solve, and expert system solutions are unattainable or too costly. To develop a BPN for this kind of problems, the following guidelines are presented based on the results of this paper and our experience in developing BPNs for oil well CBQ prediction (He et al., 1996, 1997).

1. Apply a method to rank the importance of inputs only if a reduction in the number of inputs is desired.
2. Seek expert opinion regarding the ranking of importance of inputs.
3. Delete those inputs that rank low in importance both by the method of step 1. and by expert opinion. If expert opinion is unavailable, two methods should be used to rank input importance and compared; or assume all inputs are important.
4. Use a self-organizing method, e.g. Kohonen's self-organization map (see Hertz et al., 1991 for details of this neural network), to perform a clustering of the field data after deletion of unimportant inputs. Results reported in He et al. (1996, 1997) indicate that clustering input data before training contributes to the network's achieving a higher accuracy.
5. Select a fixed percentage of representative data from each cluster as training data; the rest will be used as testing data. Train the network.
6. Try a linear combination of trained neural networks only if a single network cannot deliver the required performance; this is because even an optimal linear combination of networks (Hashem and Schmeiser, 1997) may not show a significant improvement of performance over a single network (He et al., 1996, 1997).

Our experience in building the neural networks for the CBQ problem indicated that expert knowledge is highly valuable in building networks of high accuracy. However, expert knowledge may be incomplete or even unavailable in many situations. How to utilize domain expert's partial or incomplete knowledge more effectively in conjunction with analytical and/or experimental tools to identify important input parameters, is an issue worthy of further investigation.

Table 3
Comparison of final results (in MSE) of networks built with different methods

COM	SA	FC	Original
1.19×10^{-3}	3.77×10^{-4}	3.17×10^{-4}	1.53×10^{-3}

Acknowledgements

The author acknowledges the assistance of Mr Feng He and Mr Jun Lin, who helped conduct the numerous experiments reported in this paper. Dr Gordon Guo of the

Petroleum Recovery Research Center of New Mexico Tech, who is now with SCI America Inc., provided invaluable assistance by supplying the training and testing data for the CBQ-modeling and prediction neural networks and serving as the domain expert in that application.

References

Bartlett E.B. (1994). Self determination of input variable importance using neural networks. *Neural, Parallel and Scientific Computations*, 2, 103–114.

Dagli, C. H., Akay, M., Ersoy, O., & Fernandez, B. R. (Eds.). (1997). Smart engineering systems: neural networks, fuzzy logic, data mining, and evolutionary programming. *Intelligent Engineering Systems Through Artificial Neural Networks*, Vol. 7. New York: ASME Press.

Engelbrecht, A. P., Cloete, I., & Zurada, J. M. (1995). Determining the significance of input parameters using sensitivity analysis, From natural to artificial neural computation: *Proceedings of International Workshop on Artificial Neural Networks* (pp. 382–388). Malaga-Torremolinos, Spain: Springer.

Hashem S., & Schmeiser B. (1997). Optimal linear combinations of neural networks. *Neural Networks*, 10 (4), 599–614.

He, F., Sung, A. H., & Guo, B. (1996). Using neural networks to predict oil well cement bonding quality, (Technical Report). Socorro: New Mixico Tech, Computer Science Department.

He, F., Sung, A. H., & Guo, B. (1997). A neural network for prediction of oil well cement bonding quality, *Proceedings of IASTED International Conference on Control* (pp. 417–420). Cancun, Mexico: IASTED-ACTA Press.

Hertz, J., Krogh, A., and Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation*. Redwood City: Addison-Wesley.

Lin Y., & Cunningham G.A. (1995). A new approach to fuzzy-neural system modeling. *IEEE Transactions on Fuzzy Systems*, 3 (2), 190–198.

Moller M.F. (1993). A scaled conjugate gradient algorithm for fast learning. *Neural Networks*, 6, 525–533.

Smith, R. C. (1984). *Journal of Petroleum Technology*, 36(12), 1897–1904.

Zurada, J. M., Malinowski, A., & Cloete, I. (1994). Sensitivity analysis for minimization of input data dimension for feed forward neural network, *Proceedings of IEEE International Symposium on Circuits and Systems*. London: IEEE Press.

Zurada, J. M. (1992). *Introduction to Artificial Neural Systems*. St. Paul, MINN: West Publishing Company.