CHEMICAL ENGINEERING TRANSACTIONS Volume 17, 2009
Editor Sauro Pierucci
Copyright © 2009, AIDIC Servizi S.r.l., ISBN 978-88-95608-01-3 ISSN 1974-9791
DOI: 10.3303/CET0917207

An Empirical Model of a Multiphase Reactor Based on Artificial Neural Network

Barbara Zakrzewska*, Zdzisław Jaworski
Chemical Engineering Faculty, West Pomeranian University of Technology, Szczecin
Al. Piastów 42, 71-065 Szczecin, Poland, email: zakrzewska@zut.edu.pl

An empirical model of a multiphase reactor based on artificial neural network has been developed. The multilayer feedforward network with one hidden layer has been used. The effect of the number of neurons in the hidden layer on the process parameters has also been examined. The model determines the system response to changes in the inlet variables. An optimum network architecture possessing good generalization ability has been determined in this study. The reactor model obtained from the network training process can be used in industrial practice for controlling the reactor in real time.

1. Introduction

Artificial neural network (ANN) have often been used in chemical and process engineering (Himmelblau, 2008; Bulsari, 1995). For instance, Molga (2003) presented the scope of application of artificial neural networks in chemical engineering and biotechnology. Those networks have often been used for control of chemical process systems.
The present study covers the first stage of a model development for controlling process parameters of a multiphase reactor. The reactions occurring in the reactor and geometry of the reactor was similar to that presented by Haut et al. (2004). The model was built based on a feedforward neural network, which has a static structure. Nevertheless, such the model is required to respond quickly, below 1 second, to changes in input parameters of the reactor. It should also help estimate the direction of altering the input parameters in order to shorten the time of reaching the target conditions.

2. Model

The starting point for the model development of the reactor that works in a real chemical plant, was a set of process parameters measured on-line such as flow rate, inlet and outlet temperatures, temperature and pressure inside the reactor, stream compositions. The real value of the output process parameter from the network, for which the model was constructed, was denoted by X. From the perspective of the process control, it is crucial to keep this quantity at a constant level. Due to confidentiality reasons, all measured values are reported here in normalized forms, p_j, according to equations (1) and (2), with p_{min} and x_{min} both equal to -1 and p_{max} and x_{max} equal to 1. Capital letters, P and X, stand for the values measured in the real reactor.

$$p_j = \frac{(p_{max} - p_{min})(P_j - P_{j,min})}{(P_{j,max} - P_{j,min})} + p_{min} \tag{1}$$

$$x = \frac{(x_{max} - x_{min})(X - X_{min})}{(X_{max} - X_{min})} + x_{min} \tag{2}$$

The total available data set used in the model development, consisted of 4390 samples of the normalized values of ten process variables, p_j. The samples were taken at consecutive constant time intervals. The total data set was divided in random mode into three subsets, i.e. the training, validation and test sets. The training set comprised 60% of the available data and both the validation and test sets had 20% of the whole available data each.

A multilayer feedforward network with one hidden layer was used. The scheme of the employed network is shown in Fig. 1. Ten neural elements, each of them corresponding to the selected process parameter, were used as the input set to the network, p_j ($j = 1, 2,\ldots, 10$). The hidden layer was composed of $i = 1, 2,\ldots, N$ neurons and the single neuron in the output layer corresponded to the process parameter, y, which was the controlled one. The input and output layer weights were respectively denoted by $w_{1,i,j}$ and $w_{2,1,i}$, and $b_{1,i}$ and $b_{2,1}$ stand for the bias values of the hidden and the output layers, respectively.

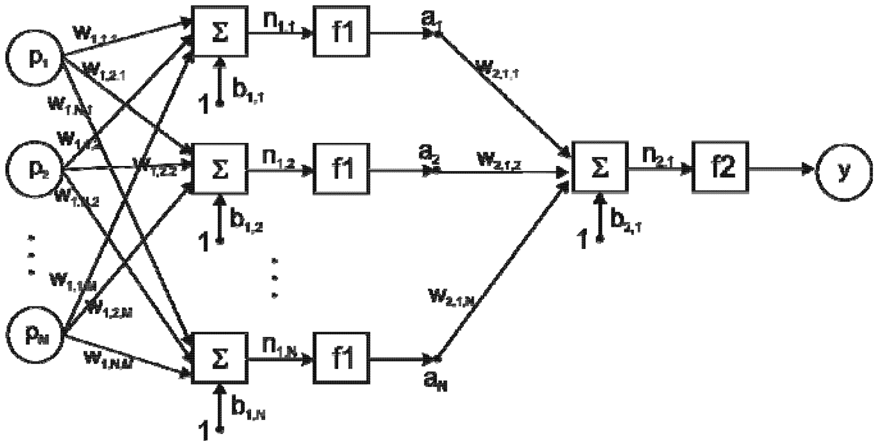

Fig. 1. Topology of multilayer feedforward network with one hidden layer

The output signal from the network, y, was computed from equation (3) in two steps; first by applying the tan-sigmoid transfer function in the hidden layer, **f1**, and then by linear transfer function in the output layer, **f2**. Equation (4) represents the empirical form of equation (3) applied in this study.

$$y = \mathbf{f2}\left(b_{2,1} + \sum_{i=1}^{N} w_{2,1,i}\mathbf{f1}\left(b_{1,i} + \sum_{j=1}^{10} w_{1,i,j} \cdot p_j\right)\right) \tag{3}$$

$$y = b_{2,1} + \sum_{i=1}^{N} w_{2,1,i} \cdot \left[\frac{2}{1 + \exp\left(-2 \cdot \left(b_{1,i} + \sum_{j=1}^{10} w_{1,i,j} \cdot p_j\right)\right)} - 1\right] \tag{4}$$

The proposed model can concisely be presented in equation (5), showing that the output signal is a function of the input vector of normalized variables, **p**, the weight matrix, **w,** and the bias vector, **b**, of the network.

$$y = f(\mathbf{p}, \mathbf{w}, \mathbf{b}) \tag{5}$$

For training the network, the backpropagation training algorithm using the Levenberg-Marquardt optimization method was applied. During the training period the weights and biases of the network were iteratively adjusted to minimize the network performance measure, e. The chosen measure represents the training error, calculated as the mean sum of squares of the difference between the output signal from the network, y_k, and the corresponding experimental value, x_k. Equation (6) was used with the k index denoting individual samples, which ranged from 1 to L and covered 60% of the total data set.

$$e = \frac{1}{L}\sum_{k=1}^{L}(y_k - x_k)^2 \tag{6}$$

The optimum network architecture with good generalization ability and a good rate of learning was determined in this study. Therefore, training with the option of the early stopping method was used in order to eliminate the overfitting problems. After each of the iterations executed during the network training, the network ability of generalization was tested by performing simulations with the validation data set. The agreement between the simulated y_k values, obtained for a given $\mathbf{p}_k$ vector of input variables, with the corresponding experimental x_k values in the validation data was determined by means of the same measure, e, equation (6). Should the e value increased or remained constant during 5 consecutive iterations, while the training error decreased, the network training was stopped.

Important factors affecting the optimum network architecture are the number of neurons in the hidden layer and the number of learning samples. There are no specific recommendations in the literature regarding those numbers. It is only possible to estimate roughly that the size of the learning sample should exceed by at least 10 times the total number of neural weights (Molga, 2003). Another method of choosing optimum network architecture is starting from a low number of neurons in the hidden

layer and then gradually increasing the number (Molga, 2003). Such the procedure was applied in this study using 1, 5, 10, 15, 20, 25, 30, 50, 100 neurons in the hidden layer. Each of those neural networks was tested with the help of the test data set. All the simulated output data, y, were compared with the experimental counterparts, x.
The output signal from the network, y, was denormalized according to equation (7), where $y_{min} = -1$, $y_{max} = 1$, while X_{min} i X_{max} had the same values as before normalization.

$$Y = \frac{(X_{max} - X_{min})(y - y_{min})}{(y_{max} - y_{min})} + X_{min} \tag{7}$$

In order to determine the optimum network architecture, the network accuracy, E, (eq. 8) was calculated. The same criterion was applied both in the case of training and testing the neural network.

$$E = \frac{1}{L}\sum_{k=1}^{L}\frac{|Y_k - X_k|}{X_k}\cdot 100\% \tag{8}$$

3. Results

Nine networks with N=1, 5, 10, 15, 20, 25, 30, 50, 100 neurons in the hidden layer were investigated. Training of the networks was carried out with the help of the early stopping method. After completion of the training period, the output signal, y_k, was compared with the corresponding process value, x_k, from measurements. An example of agreement between the analyzed values predicted from ANN, y_k, and those form measurements, x_k, is shown in Fig. 2 for the training set of 2636 data samples and 20 hidden neurons.

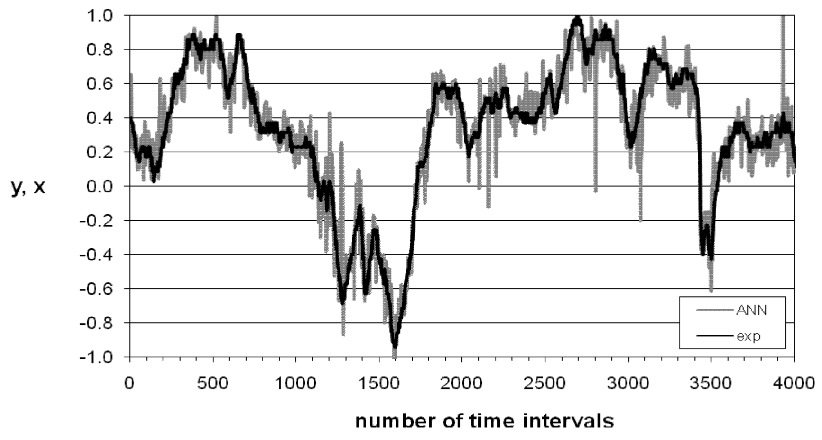

Fig. 2 Comparison of the output signal from the network (ANN) with experimental data (exp) for the training data set

Every trained ANN underwent testing with the help of the test data set that contained 877 testing samples (20%). The output variable values, y, which were computed by means of the trained network were then compared with the experimental data, x, both obtained for a randomly selected input vector, **p**. Graphical comparison of those data is presented in Fig. 3 by plotting them against the corresponding number of time intervals. The test samples and their experimental counterparts are shown there by squares and circles, respectively. For the sake of clarity, only the first half of the time intervals is shown, again for the case of 20 neurons in the hidden layer.

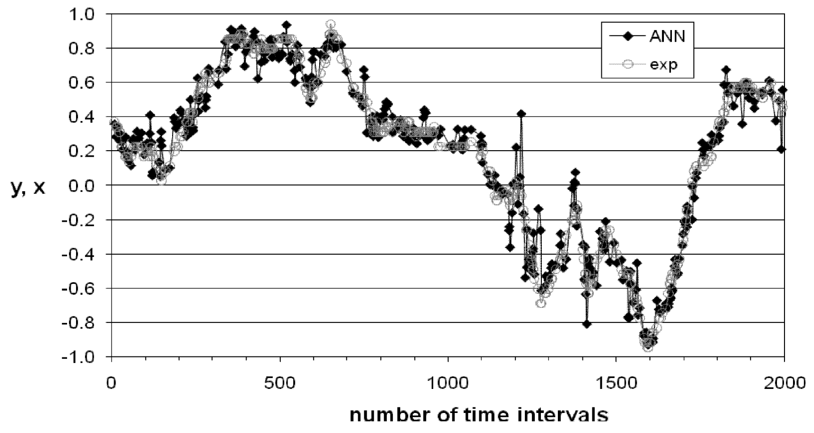

Fig. 3 Comparison of the output signal from the network (ANN) and the corresponding experimental data (exp) for half of the testing data set

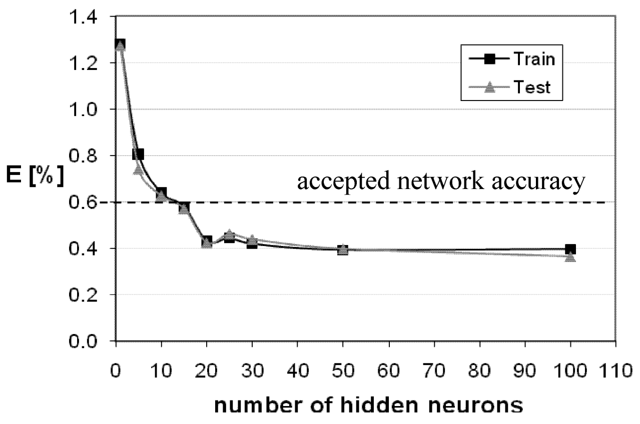

Fig. 4. Network accuracy vs. number of hidden neurons

From the perspective of the process control, it is crucial to keep the value of the output process parameter at a constant level. The network accuracy, E, was therefore computed from equation (8) for every tested ANN. In the critical analysis of the target process variable, Y, it was assumed that an acceptable accuracy should be $E \leq 0.6\%$ of the real value of the tested variable both for the training and test data set. Among the tested networks, those with 15 neurons or more in the hidden layer onward (Fig. 4) fulfilled that criterion. It should be reminded that effect can also result from an excess number of hidden neurons and then the ANN reacts simply like an interpolation device (Osowski, 2000). Therefore, in this work the assumption proposed by Molga (2003) for choosing an optimum network topology was adopted, i.e. the number of the training samples should by about 10 times larger than the total number of the network parameters. Since the number of training samples is 2636 in this study, therefore in our case that assumption is met by a network with 20 neurons and such the network has its total number of parameters equal to 241.

4. Summary and conclusions

An empirical reactor model based on artificial neural network was proposed. It was found that the multilayer feedforward network with one hidden layer of 20 neurons showed the best abilities of reproducing and generalizing the output variable of the tested reactor. Thus, equation (4) with the corresponding set of network parameters (21 values of b and 220 w values) can be regarded as the empirical reactor model.

The reactor model obtained from the network training process can practically be used for controlling the reactor in real time. The model should be able to determine the system response to changes in the inlet variables almost instantly, i.e. quicker than in 1 second. In that case, it will be possible to predict which controlled variable and how much should it be changed to achieve the desired output value. Furthermore, verification of the system response to changes of the state variables at the inlet is expected by means of the model.

The proposed model can also be used as an auxiliary tool at conventional reactor control. In order to enhance the model, much longer working time of the reactor should be taken into account. In addition, to analyze dynamic performance of the reactor, another ANN should be considered, e.g. the recurrent neural network (Al Seyab, 2008).

References

Al Seyab, R.K., Cao, Y., 2008, Differential recurrent neural network based predictive control, Comp. Chem. Eng. 32, 1533–1545.

Bulsari, A.B., Ed., 1995, Neural networks for chemical engineers, Elsevier, Amsterdam.

Haut, B., Halloin, V., Cartage, T., Cockx A., 2004, Production of sodium bicarbonate in industrial bubble columns, Chem. Eng. Sci., 59, 5687-5694.

Himmelblau, D.M., 2008, Accounts of experiences in the application of artificial neural networks in chemical engineering, Ind. Eng. Chem. Res. 47, 5782–5796.

Molga, E.J., 2003, Neural network approach to support modelling of chemical reactors: problems, resolutions, criteria of application, Chem. Eng. Proc. 42, 675-695.

Osowski, S., 2000, Neural networks for information processing, Oficyna Wydawnicza PW, Warszawa (in Polish).