

Application of ELM to predict the coagulant dosing in water treatment plants

Xiaoyan Deng and Canguang Lin

ABSTRACT

Predicting the coagulant dosage is especially crucial to the purification process in water treatment plants, directly affecting the quality of the purified water. Nowadays, several mathematical methods have been adopted for the purification process, but their predictive precision and speed still need to be improved. This study applies a novel neural network called the extreme learning machine (ELM) to predict the coagulant dosage based on certain signification factors of the raw water. Performances are compared between ELM and back-propagation neural networks in this paper. The results show that both neural network algorithms perform well in this application and ELM can realize online prediction due to its short time consumption.

Key words | activation function, artificial neural networks, back-propagation algorithm, coagulant dosing prediction, extreme learning machine, optimization problem

Xiaoyan Deng (corresponding author)
Canguang Lin
 School of Automation Science and Engineering,
 University of South China University of Technology,
 Guangzhou 510640,
 China
 E-mail: dengxy@scut.edu.cn

LIST OF PARAMETERS

Parameters	Explanation
M	The number of training data
MAPE	Mean absolute percent error
N	The dimension of input features
\tilde{N}	The number of hidden neurons
O_k	The k th actual output of the neural network corresponding to the k th input
R	The correlation coefficient
$std(x)$	The standard deviation of x
x_k	The k th training data (input)
y_k	The k th standard value corresponding to the k th input

the quality of the purified water. Nowadays, there are several conventional methods to predict the coagulant dosage in practice, such as the jar test (Wu & Lo 2010), based on the operation's experience, and regression methods (Deng *et al.* 2010), etc. Because of complicated and nonlinear relationships between operational parameters in water treatment processes, some methods cannot capture the underlying relationships (Tashaouie *et al.* 2012) and hardly adjust in time to the proper dosage (Wu & Lo 2010). An accurate prediction method can contribute to reducing the consumption of coagulant dosage, facilitate the purification process and improve the quality of the purified water. Artificial neural networks (ANNs) have been increasingly applied in this area, and they can successfully solve the system's nonlinearity.

An ANN is a nonlinear system, as first developed by McCulloch & Pitts (1943), which is the simulation of biological nervous systems and has the ability to deal with much information. Therefore ANN can be used to solve many complicated problems, and several previous studies have applied

INTRODUCTION

The coagulation process is a very prominent step in water treatment. The precision of controlling the coagulant dosage is closely related to the next several processes and

doi: 10.2166/ws.2016.205

ANN to improve the water treatment system. Mirsepassi *et al.* (1995) used back-propagation (BP) networks to determine the alum and polymer dosing in a water treatment plant on the central coast of NSW. Valentin *et al.* (1999) developed a hybrid system based on the Kohonen self-organizing feature map, and used a multilayer perceptron to predict the coagulant dosing. Yu *et al.* (2000) improved their prediction model using ANN to control the coagulant dosing at a water treatment plant in Taipei, Taiwan. Karama *et al.* (2005) used an ANN model for online prediction of optimal coagulant dosing, and in order to avoid over-fitting, their study adopted the Levenberg–Marquardt method in combination with weight decay regularization.

In all the studies above, the ANN models extract all the significant features from the raw water as their inputs and take the actual coagulant dosing as their outputs. Then the neural networks can be trained through the Levenberg–Marquardt method to determine the optimal coagulant dosage. However, the BP model is still confronted with bottlenecks such as time-consuming training, local minima and necessary stopping methods.

The extreme learning machine (ELM) is a single hidden layer feed-forward neural network (SLFN) proposed by Huang *et al.* (2004, 2006, 2012), and it has been applied in many regression and multiclass classification problems. ELM only needs to designate the number of hidden layer nodes and choose the appropriate activation function. With its input weights (linking the input layer to the hidden layer) and hidden layer bias randomly assigned, the ELM can then analytically determine the output weights (linking the hidden layer to the output layer) through the Moore–Penrose generalized inverse, while plenty of iterations are required to train the BP neural networks. Furthermore, the result of ELM is the smallest norm least-squares solution, which means that it can have the smallest predicting error (Bartlett 1998). In this paper, we apply this novel neural network to enhance the accuracy of the prediction system, and focus on analysis and comparison of two neural networks for solving the coagulant dosage prediction problem. Specifically, the main contributions of this paper lie in the following facts.

(1) Two neural networks: back-propagation neural networks (BPNNs) and ELM are exploited, analyzed and

compared when they are applied to predict the coagulant dosage. To the best of the authors' knowledge, this is the first time that ELM has been applied in this application.

(2) Computer simulations and experiments were conducted to verify the effectiveness of two neural networks.

(3) In addition, both BPNNs and ELM obtain good performance, but ELM is faster than BPNNs when solving the prediction of the coagulant dosage.

Above all, we organize this paper as follows. The second section will briefly introduce the methodology of ELM and BP as these two artificial neural networks. The simulation performances of both networks and further discussion of the distinctions between ELM and BPNNs are presented in the third section Experimental conclusions are given in the final section.

METHODS

An ANN's structure typically consists of an input layer, single hidden layer and an output layer, which is shown in Figure 1, and each layer links to the adjacent layers by weighted connections so that the information can propagate in a feed-forward manner. Differentially, BP propagates the error in a back-forward way through the Levenberg–Marquardt method while ELM obtains its optimal solution using the Moore–Penrose generalized inverse.

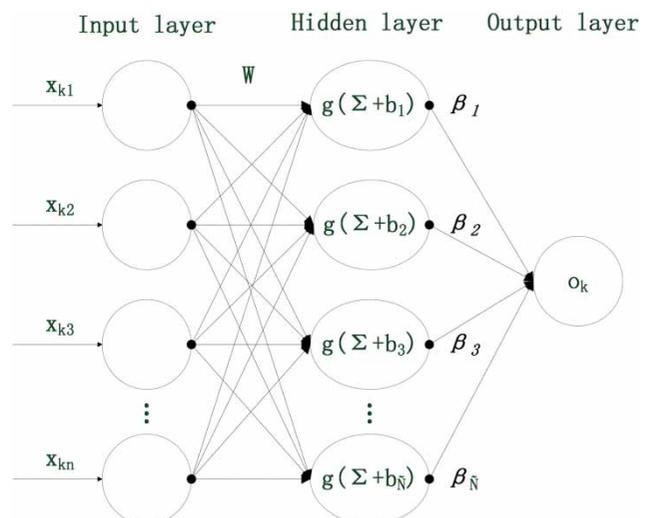


Figure 1 | ELM model.

As follows, the matrix form of ELM is presented and we will apply this neural network to our modeling to predict the coagulant dosage.

Matrix form of ELM

Suppose that we have M training data, that is (x_k, y_k) and $k = [1, 2, \dots, M]$, where $x_k = [x_{k1}, x_{k2}, \dots, x_{kN}]^T \in R^n$ and $y_k \in R$, N is the dimension of input features and the neuron number of the input layer. The input weight vector $W = [w_1, w_2, \dots, w_{\tilde{N}}]$, \tilde{N} is the number of hidden neurons, and $w_j = [w_{j1}, w_{j2}, \dots, w_{jN}]^T$, where w_{ji} denotes the weight that connects the j th hidden neuron and the i th input neuron, here $i = [1, 2, \dots, N]$ and $j = [1, 2, \dots, \tilde{N}]$.

Since the input weights and hidden layer bias, where $b_j \in R$, are randomly assigned, the SLFNs with \tilde{N} hidden neurons and the activation function $g(\sum + b_j)$ can be mathematically modeled as follows (Huang *et al.* 2004, 2006, 2012):

$$\sum_{j=1}^{\tilde{N}} \beta_j g(w_j \cdot x_k + b_j) = y_k, k = 1, 2, \dots, M \tag{1}$$

Given the M equations above, then the matrix form of the ELM algorithm can be written as follows:

$$H\beta = Y \tag{2}$$

where $H = \{g(w_j \cdot x_k + b_j)\}$, and $Y = [y_1, y_2, \dots, y_M]$ is the output vector corresponding to the inputs $[x_1, x_2, \dots, x_M]$. Then the solution can be obtained as follows, where H^\dagger is the Moore–Penrose generalized inverse of H :

$$\hat{\beta} = H^\dagger Y \tag{3}$$

Data preprocessing

According to the modeling above, now we will preprocess our data and apply them to this neural network model. The data in this paper are provided by a water treatment plant in Foshan, China, for 2008. The data set contains

5,100 samples, and the neural network takes its raw-water turbidity, value of pH, flux rate and turbidity of water to be filtered as inputs and takes coagulant dosage as output.

We removed unreasonable data to eliminate the burst noise, and 5,017 valid data remained. Then we randomly divided the valid data set into a training data set (2/3) and a testing data set (1/3).

Normalization

Assume the neural networks take the logic sigmoid as its activation function, which is presented in Figure 2. The logic sigmoid is a bounded continuous differentiable function, whose values are located at $[0, 1]$. While the input variables and output variables (coagulant dosage) have a wide range, the neural networks will saturate and overflow without normalization. Thus we normalized the input variables and output variables respectively as follows:

$$x_{ik} = \frac{x_{ik} - \bar{x}_i}{std(x_i)}, k = 1, 2, \dots, M, i = 1, 2, \dots, N \tag{4}$$

$$y_k = \frac{y_k - \bar{y}}{std(y)}, k = 1, 2, \dots, M \tag{5}$$

Here \bar{x}_i is the mean value of the i th input feature, $std(x_i)$ is the standard deviation of the i th input feature, \bar{y} is the mean value of the label, and $std(y)$ is the standard deviation of the

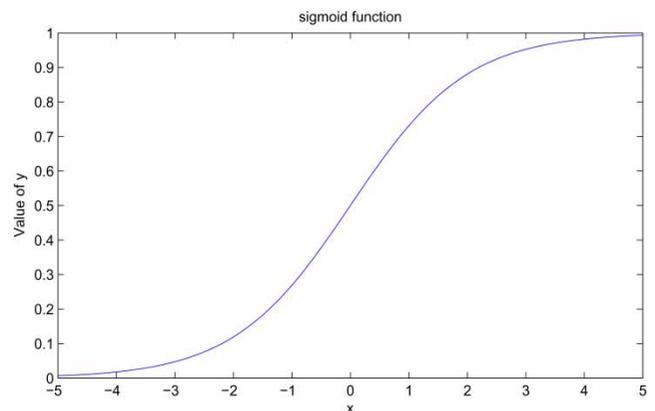


Figure 2 | Sigmoid function.

label. And we unnormalize the normalized data as follows:

$$x_{ik} = x_{ik} \cdot std(x_i) + \bar{x}_i, k = 1, 2, \dots, M, i = 1, 2, \dots, N \quad (6)$$

$$y_k = y_k \cdot std(y) + \bar{y}, k = 1, 2, \dots, M \quad (7)$$

Cost function

To estimate and compare the performances of ELM and BP, their performances are evaluated by the same cost function to calculate the MAPE as its training error and testing error.

The cost function is given as follows, where y_k is the standard dosing, while the input is x_k , and o_k is the actual output of the neural networks:

$$MAPE = \frac{1}{M} \sum_{k=1}^M \left(\left| \frac{y_k - o_k}{y_k} \right| \times 100\% \right) \quad (8)$$

RESULTS AND DISCUSSION

Performance evaluation

For the BP neural networks, the learning rate is fixed as 0.01, and the neuron number and activation functions are

designated respectively. Generally, we set the max iteration as 1,000 for the tansig and logsig functions (the definitions of the functions are formulated as Equations (9) and (10)). The neuron number is assigned from 6 to 20, and the networks are likely over-fitting when the neuron number becomes large. Because BPNNs consume a lot of time in training the networks, each set-up is trained ten times and the average performances are presented in Table 1.

$$\text{tansig} = f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (9)$$

$$\text{logsig} = f(x) = \frac{1}{1 + e^{-x}} \quad (10)$$

For the ELM algorithm, the input weights and hidden layer bias are randomly assigned within the interval $[-1, 1]$. In order to compare the capacity of ELM with BPNNs, we adopt the same hidden layer activation functions (tansig and logsig), and assign the hidden layer neuron number from 10 to 500 respectively (the interval for the neuron number is set as 10). Each set-up is trained 50 times and the average performances are shown in Figure 3.

As the neuron number of the ELM can range over a wide extent to get a good performance, we only list one set-up which has the smallest average testing error. And

Table 1 | The performance of BPNNs with tansig or logsig as the activation function

Activation function	Neuron number	Average iteration number	Average training time (%)	Average training error (%)	Average testing error (%)	Smallest testing error (%)
Tansig	6	591	51.4039	7.09	7.21	6.63
	8	708	57.7563	6.60	6.88	6.28
	10	844	75.7478	5.98	6.29	5.86
	12	951	92.8658	5.81	6.19	5.57
	14	988	98.0263	5.42	5.72	5.32
	16	1,000	111.6640	5.09	5.54	5.20
	18	1,000	117.8120	4.85	5.23	4.84
	20	1,000	127.2032	4.84	5.69	4.73
Logsig	6	534	42.7224	7.16	7.22	6.84
	8	806	66.8199	6.50	6.70	6.12
	10	897	81.2812	6.27	6.52	5.90
	12	986	97.8064	5.71	5.96	5.62
	14	982	97.2682	5.47	5.88	5.44
	16	1,000	106.4332	5.18	5.80	5.37
	18	1,000	113.0742	5.07	5.46	5.07
	20	1,000	128.5261	4.89	5.38	5.03

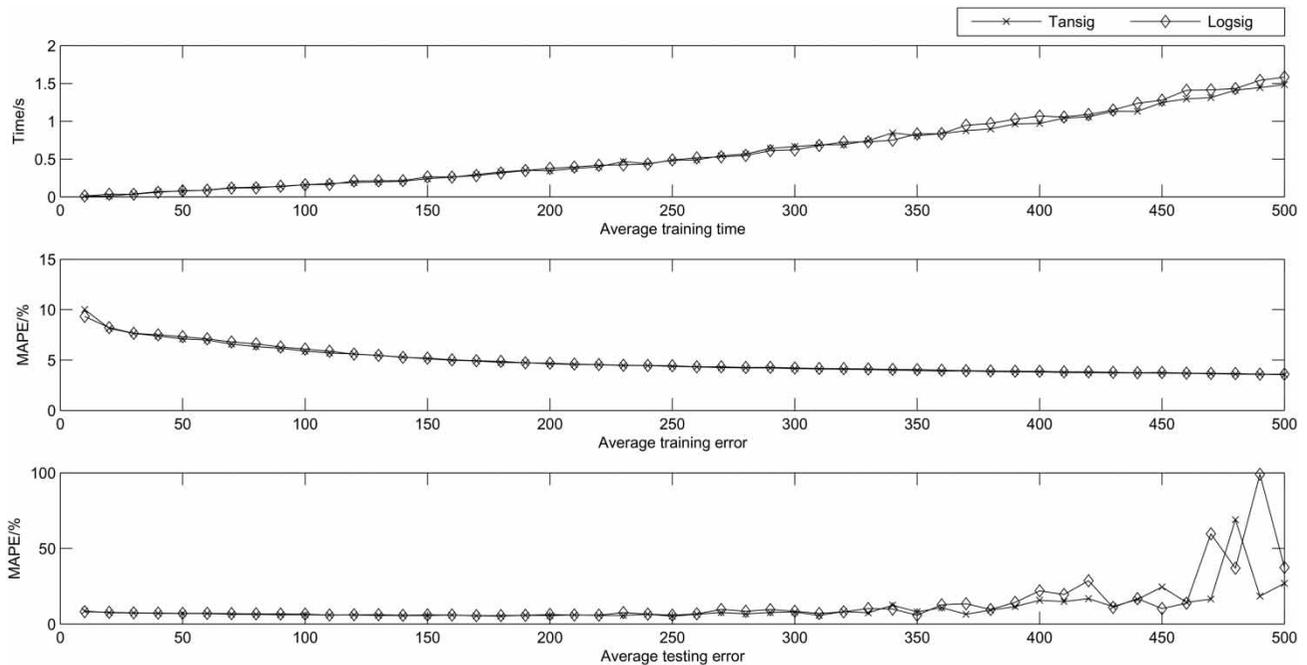


Figure 3 | Performances of different activation functions (tansig and logsig) and neuron numbers.

each set-up of the ELM algorithm is trained 50 times and the average performances are presented in Table 2.

As is shown in Tables 1 and 2 and Figure 3, when the neuron number increases, the training time of both algorithms also increases with the trend, while their training error and testing error decrease gradually. However, the speed of the ELM is much faster than the BPNNs. Similarly, both algorithms will face an over-fitting problem when the neuron number becomes very large, such as tansig and logsig (in Figure 3) producing large testing errors when their neuron number exceeds 350.

In order to enhance the precision of our predictive model, several uncommon activation functions are adopted for these two neural networks (tribas, radbas and satlin); these are bounded nonlinear functions and adapted to these two

neural networks (Hornik 1999). The definitions of these activation functions are formulated as Equations (11)–(13):

$$\text{tribas} = f(x) = \begin{cases} 1 - \text{abs}(x), & \text{if } -1 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$\text{radbas} = f(x) = e^{-x^2} \quad (12)$$

$$\text{satlin} = f(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ x, & \text{if } 0 \leq x \leq 1 \\ 1, & \text{if } 1 \leq x \end{cases} \quad (13)$$

For BPNNs, as the convergence rates of tribas, radbas and satlin are much faster than the former activation functions, some proper stopping-methods are needed (Sun *et al.* 2008) to avoid over-fitting or under-fitting. Here we terminate the training process when the correlation coefficient R reaches near to 0.98 (shown in Figure 4), which denotes that the neural networks can approximate well with both the training data and the testing data. However, the larger the neuron number is, the faster the convergence speed that the BPNNs will reach, and thus the less the iteration number the BPNNs require, whereas the stopping method becomes

Table 2 | The performance of ELM with tansig or logsig as activation function

Activation function	Neuron number	Average training time(s)	Average training error (%)	Average testing error (%)	Smallest testing error (%)
Tansig	120	0.1732	5.53	6.23	5.53
Logsig	80	0.1011	6.55	6.99	6.44

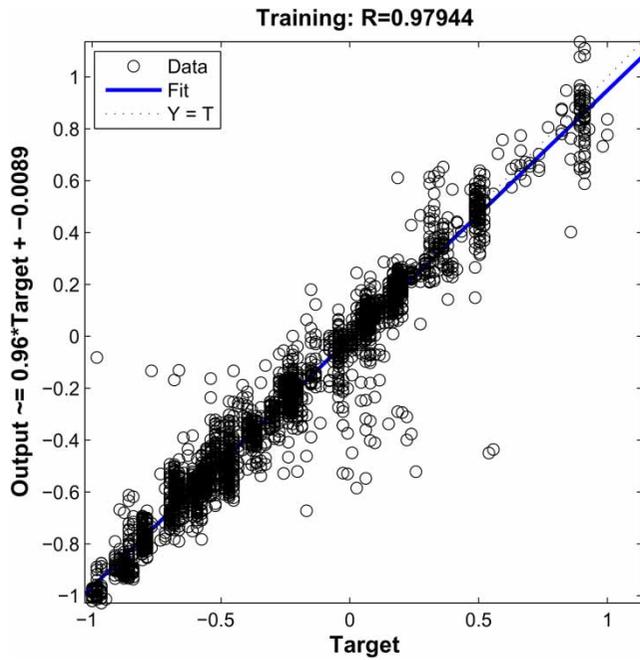


Figure 4 | Correlation coefficient R.

hard to control and the training time becomes longer when the neuron number becomes much larger. Consequently the neuron number was set in the interval [50,200].

The average performances of BPNNs are presented in Table 3. The set-up of the ELM which has the smallest average testing error is presented in Table 4, and its performances for different activation functions and neuron numbers are shown in Figure 5 (the interval of neuron number is set as 10).

Table 4 | The performance of ELM with tribas, radbas or satlin as activation function

Activation function	Neuron number	Average training time(s)	Average training error (%)	Average testing error (%)	Smallest testing error (%)
Tribas	400	0.8263	3.98	4.96	4.65
Radbas	160	0.2150	4.98	6.10	5.34
Satlin	360	0.7107	4.13	4.98	4.60

As is shown in Figure 5, as the neuron number increases, the training time of both algorithms increases, and their training error and testing error decrease. In this situation, radbas will produce a large testing error when its neuron number exceeds 250. Tribas and satlin are adaptive to a wide range of neuron numbers. In general, both algorithms with tribas or satlin and the proper neuron number can obtain better performances than the situation (tansig and logsig) listed above, which means that our predictive model has been improved.

To prove both algorithms can generate good performance, we randomly choose 200 testing data and use them to predict the coagulant dosage using BPNNs and ELM respectively. The performances of the BPNNs are presented in Figure 6, and the performances of the ELM are presented in Figure 7. The activation function of both algorithms is satlin and the neuron number is set as 200 for BP and 360 for ELM.

In the first subplot, the solid curve is the standard dosing corresponding to the inputs, and the dotted curve

Table 3 | The performance of BPNNs with tribas, radbas or satlin as activation function

Activation function	Neuron number	Average iteration number	Average training time (%)	Average training error (%)	Average testing error (%)	Smallest testing error (%)
Tribas	50	60	17.1195	4.60	5.11	4.83
	100	40	23.7122	4.00	4.76	4.55
	150	16	17.2210	3.94	4.83	4.60
	200	10	17.2912	3.98	4.93	4.61
Radbas	50	60	17.7935	4.12	4.82	4.58
	100	20	14.8497	4.09	4.80	4.54
	150	10	17.1336	4.23	4.77	4.59
	200	9	30.7603	3.99	4.74	4.55
Satlin	50	110	29.8102	4.71	5.08	4.77
	100	100	56.9560	4.24	4.77	4.52
	150	30	31.6885	3.85	4.51	4.41
	200	11	19.3847	4.02	4.70	4.48

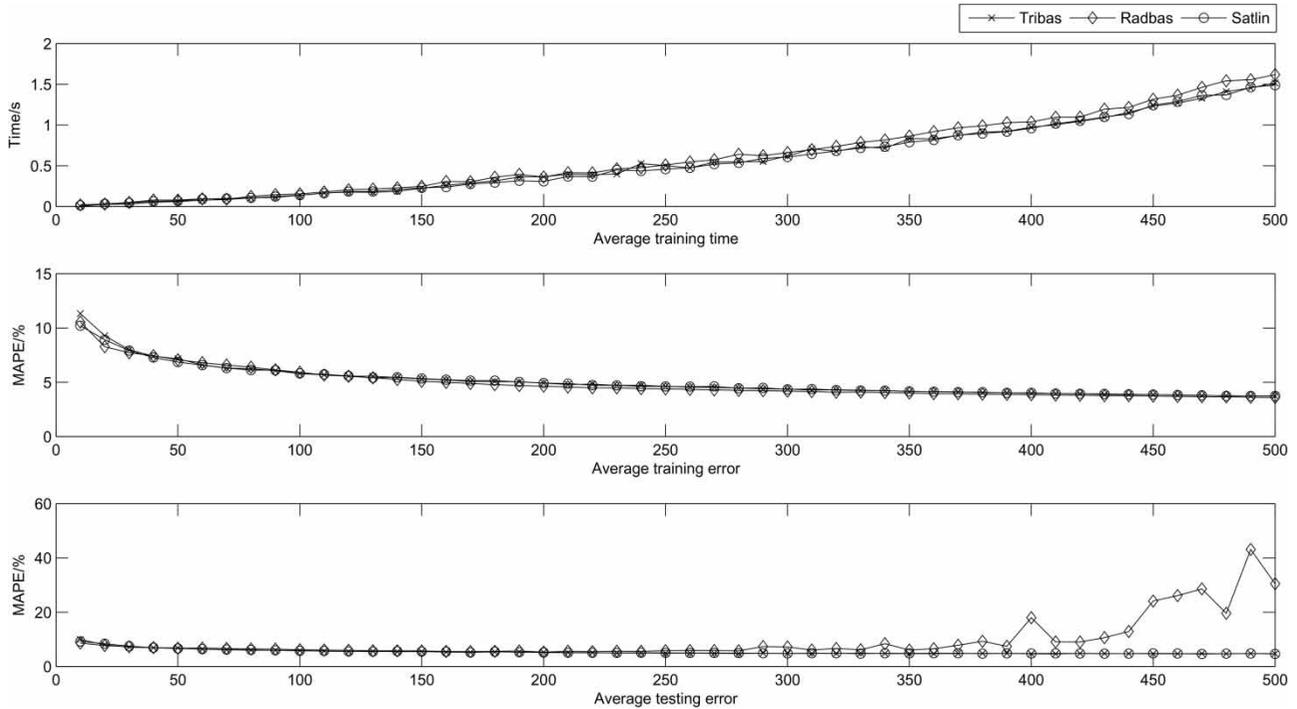


Figure 5 | Performances of different activation functions (tribas, radbas and satlin) and neuron numbers.

is the actual output of the neural networks. In the second subplot, the value of the curve denotes the relative error of each of the testing data, i.e. $(y_k - o_k)/y_k$. Comparing the performances of both algorithms, both of them show similarly good performances, however, the ELM runs

more than 50 times faster than the BPNNs algorithm in this application.

All simulations are conducted in MATLAB running on a personal computer with a dual core CPU (1.80 GHz and 1.80 GHz) and 4G memory.

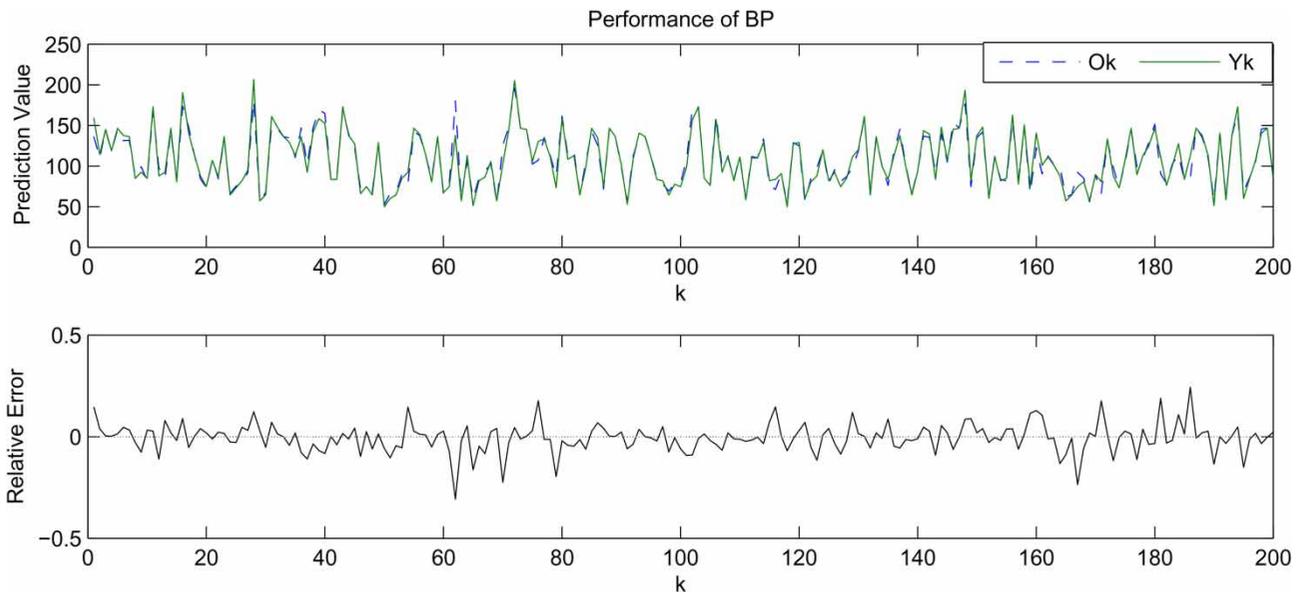


Figure 6 | Performance of BP: training time = 20.7306s; training accuracy = 95.68%; testing accuracy = 95.3%.

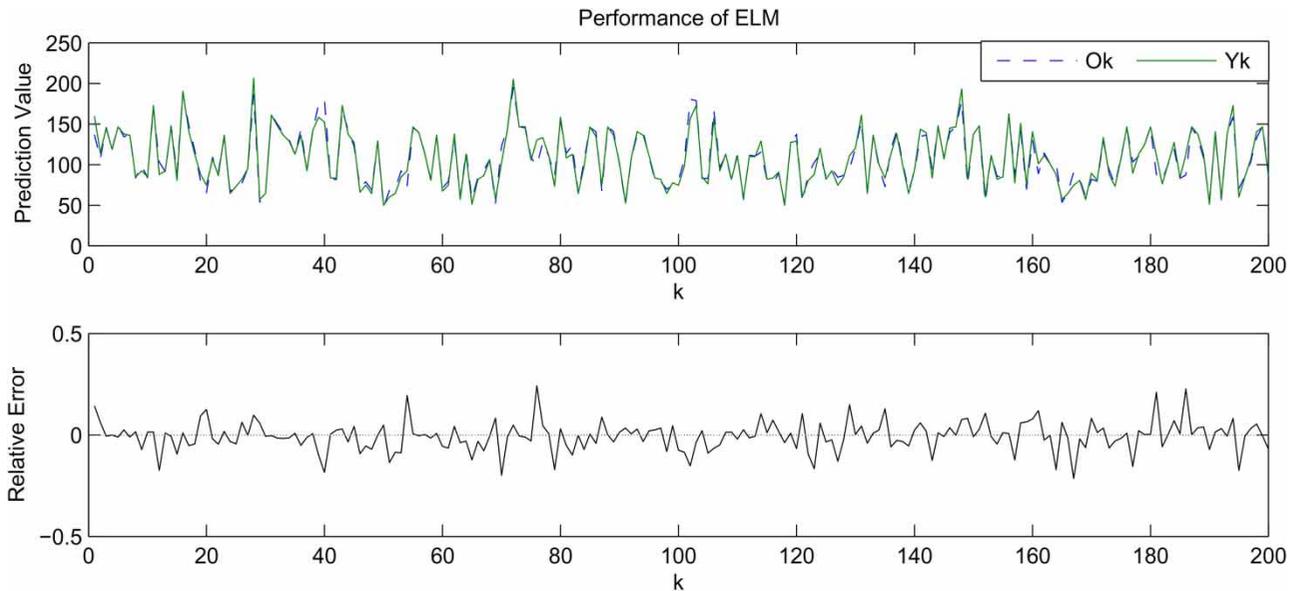


Figure 7 | Performance of ELM: training time = 0.7084s; training accuracy = 95.67%; testing accuracy = 95.32%.

DISCUSSION

In this section, several similarities and distinctions between BP and ELM will be discussed for this application.

- (1) As is shown in Figures 6 and 7, both the BPNNs and ELM can be used to predict the coagulant dosing, and the performances of the BPNNs and ELM are quiet good and even similar. If the proper hidden layer activation function and neuron number are adopted, and the training error of the ELM is 4.33% and the training error can reach as low as 4.68%, the accuracy is more than 95.32%.
- (2) The data sets we used only include raw-water turbidity, pH, flux rate and the turbidity of water to be filtered as inputs, but the factors which influence the coagulant dosing are not limited to these, including, for example, temperature and microorganism content. Considering more factors, maybe it can help to improve the accuracy of prediction in actual practice.
- (3) From the result, we see that the training time of the BPNNs is about 60 seconds or even 120 seconds. In actual practice, this is not allowable because the water treatment process is very fast. But the time for the ELM is less than one second. The speed of the ELM is more

prominent. In most practical engineering applications, the ELM can quickly work out the accurate coagulant dosing corresponding to the changeable input factors, so that it is very helpful for realizing online prediction.

- (4) Both algorithms require taking the bounded nonlinear function as the hidden layer activation function. Their training time also increases with the trend of increasing neuron number, while their training error and testing error decrease with that trend. And both of them face the same issues of over-fitting and under-fitting.
- (5) To obtain better performance, the BPNNs algorithm requires plenty of iterations and time to tune all the input weights, hidden layer bias and output weights, while the ELM algorithm only needs to randomly designate the input weights and hidden layer bias, and then mathematically determines the output weights through the Moore–Penrose generalized inverse.
- (6) The BPNNs algorithm needs to fix many parameters, such as hidden layer activation function, learning rate and hidden layer neuron number. Sometimes it requires an appropriate stopping method to avoid over-fitting and under-fitting. The ELM only needs to designate the activation function and hidden layer neuron number. Thus it is much easier to train the ELM for good performance than to train BPNNs.

- (7) BPNNs and ELM with different activation functions have different performances. For both algorithms to obtain good performance with the same activation function, the ELM generally requires a larger neuron number.

CONCLUSION

In this paper, we apply the ELM to model the coagulant dosage prediction system and compare its performances with BPNNs. This paper has demonstrated that both BPNNs and ELM can perform well when used to predict the coagulant dosing, while ELM is much more prominent in speed. Thus in most engineering applications, the ELM can be very significant for online prediction. The ELM algorithm can also avoid converging at a local minimum, and it does not require some stopping methods to avoid over-training. Therefore, the ELM is adopted in this application. Another interesting line of future research would be a regularization term in the ELM, which can reduce the over-fitting effects and help to select neuron number.

ACKNOWLEDGEMENT

This work was supported in part by the Foshan Cancheng Technology program of China (2008B1034).

REFERENCES

- Bartlett, P. L. 1998 The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory* **44** (2), 525–536.
- Deng, X. Y., Tang, D. C., Zhu, X. F., Huang, D. P., Zou, Z. Y., Li, Z. F. & Luo, Y. H. 2010 Based on BP neural network and the mechanism of the coagulant dosage modeling. *Industrial Instrumentation and Automation Device* **6**, 8–10.
- Hornik, K. 1991 Approximation capabilities of multilayer feedforward networks. *Neural Networks* **4** (2), 251–257.
- Huang, G. B., Zhu, Q. Y. & Siew, C. K. 2004 Extreme learning machine: a new learning scheme of feedforward neural networks. In: *IEEE International Joint Conference on Neural Networks, Volume 2*. IEEE, New York, USA, pp. 985–990.
- Huang, G. B., Zhu, Q. Y. & Siew, C. K. 2006 Extreme learning machine: theory and applications. *Neurocomputing* **70** (1–3), 489–501.
- Huang, G. B., Zhou, H., Ding, X. & Zhang, R. 2012 Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man & Cybernetics, Part B* **42** (2), 513–529.
- Karama, A., Benhammou, A., Lamrini, B. & Le Lann, M. V. 2005 A neural software sensor for online prediction of coagulant dosage in a drinking water treatment plant. *Transactions of the Institute of Measurement & Control* **27** (3), 195–213.
- McCulloch, W. S. & Pitts, W. 1943 A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **5** (4), 115–133.
- Mirsepasi, A., Cathers, B. & Dharmappa, H. B. 1995 Application of artificial neural networks to the real time operation of water treatment plants. In: *IEEE International Conference on Neural Networks, Proceedings, Volume 1*. IEEE, New York, USA, pp. 516–521.
- Sun, Z. L., Choi, T. M., Au, K. F. & Yu, Y. 2008 Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems* **46** (1), 411–419.
- Tashaouie, H. R., Gholikandi, G. B. & Hazrati, H. 2012 Artificial neural network modeling for predict performance of pressure filters in a water treatment plant. *Desalination and Water Treatment* **39** (1–3), 192–198.
- Valentin, N., Denoeux, T. & Fotoohi, F. 1999 An hybrid neural network based system for optimization of coagulant dosing in a water treatment plant. In: *International Joint Conference on Neural Networks, Volume 5*. IEEE, New York, USA, pp. 3380–3385.
- Wu, G. D. & Lo, S. L. 2010 Effects of data normalization and inherent-factor on decision of optimal coagulant dosage in water treatment by artificial neural network. *Expert Systems with Applications* **37** (7), 4974–4983.
- Yu, R. F., Kang, S. F., Liaw, S. L. & Chen, M. C. 2000 Application of artificial neural network to control the coagulant dosing in water treatment plant. *Water Science and Technology* **42** (3–4), 403–408.

First received 17 September 2016; accepted in revised form 14 December 2016. Available online 28 December 2016