



Available online at www.sciencedirect.com



Journal of Hydrology 289 (2004) 1–8

Journal
of
Hydrology

www.elsevier.com/locate/jhydrol

A non-linear rainfall-runoff model using radial basis function network

Gwo-Fong Lin*, Lu-Hsien Chen

Department of Civil Engineering, National Taiwan University, Taipei 10617, Taiwan

Received 13 January 2003; revised 17 September 2003; accepted 31 October 2003

Abstract

In this paper, the radial basis function network (RBFN) is used to construct a rainfall-runoff model, and the fully supervised learning algorithm is presented for the parametric estimation of the network. The fully supervised learning algorithm has advantages over the hybrid-learning algorithm that is less convenient for setting up the number of hidden layer neurons. The number of hidden layer neurons can be automatically constructed and the training error then decreases with increasing number of neurons. The early stopping technique that can avoid over-fitting is adopted to cease the training during the process of network construction. The proposed methodology is finally applied to an actual reservoir watershed to find the one- to three-hour ahead forecasts of inflow. The result shows that the RBFN can be successfully applied to build the relation of rainfall and runoff.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Artificial neural network; Radial basis function; Flow forecasting; Fully supervised learning algorithm

1. Introduction

For many years, hydrologists have attempted to understand the transformation of precipitation to runoff. Dependable forecasts of streamflow are essential to many aspects of water resources projects. In Taiwan, owing to non-uniform temporal and spatial distributions of precipitation and the presence of high mountains and steep channels all over the island, the hydrologic systems are very complex. Thus, flood forecasting is always a benchmark problem for hydrologists and water resources engineers. Neural networks devised via imitating brain activity are

capable of modeling non-linear and complex systems and thus provide an alternative approach for modeling hydrologic systems.

Artificial neural networks were first developed in the 1940s. Generally speaking, neural networks are information processing systems. In recent decades, considerable interest has been raised over their practical applications, because the current algorithms can overcome the limitations of early networks. Bypassing the model construction and parameter estimation phases adopted by most of the conventional techniques, the neural networks can automatically develop a forecasting model through a simple process of the historic data. Such a training process enables the neural system to capture the complex and non-linear relationships that are not

* Corresponding author. Fax: +886-223-631-558.

E-mail address: gflin@ntu.edu.tw (G.-F. Lin).

easily analyzed by using conventional methods. Based on the structure of the neural networks and the learning algorithm, various neural network models are frequently proposed and studied to solve different problems. The back-propagation network is the popular one. The term ‘back propagation’ means that the parameters of the network are determined by back-propagating the error signals through the network, layer by layer (Haykin, 1991).

Regarding the flood forecasting, Zhu and Fujita (1994) compared the performance of the fuzzy reasoning and the feed-forward neural network model in predicting the 3-h ahead runoff. Hsu et al. (1995), Marina et al. (1999) and Komda and Makarand (2000) used the back-propagation network model for river flood forecasting. However, the back-propagation network has some disadvantages. For example, it tends to yield local solutions, the learning rate is slow and the network structure is difficult to develop. Radial basis function network (RBFN) has been widely used for non-linear system identification because of its simple topological structure and its ability to reveal how learning proceeds in an explicit manner. RBFN was first introduced to solve the real multivariate interpolation problem (Powell, 1987). Broomhead and Lowe (1988) were the first ones who exploited the use of radial basis function in the design of neural networks. Poggio and Girosi (1990) developed regularization networks from approximation theory with RBFN as a special case. More recently, RBFN has been employed in non-linear systems identification and time series prediction (Moody and Darken, 1989; Broomhead and Lowe, 1988). Park and Sandberg (1991) studied the universal approximation problem using the RBFN. Recent studies focus on the problem of time series prediction using RBFN. The rainfall-runoff relationship is one of the most complex hydrologic phenomena to comprehend due to the tremendous spatial and temporal variability of watershed characteristics and precipitation patterns, and the number of variables involved in the modeling of the physical processes. In this paper, RBFN is used to construct a rainfall-runoff model. The model is then applied to an actual reservoir watershed to forecast the one- to three-hour ahead runoff.

2. Radial basis function network

2.1. Architecture

A RBFN, which is multilayer and feedforward, is often used for strict interpolation in multi-dimensional space. The term ‘feedforward’ means that the neurons are organized in the form of layers in a layered neural network (Haykin, 1991). The basic architecture of a three-layered neural network is shown in Fig. 1. An RBFN has three layers including input layer, hidden layer and output layer. The input layer is composed of input data. The hidden layer transforms the data from the input space to the hidden space using a non-linear function. The output layer, which is linear, yields the response of the network. The argument of the activation function of each hidden unit in an RBFN computes the Euclidean distance between the input vector and the center of that unit. The network has a better accuracy only when the early stop technique is appropriately used.

In the structure of RBFN, the input data X is an I -dimensional vector, which is transmitted to each hidden unit. The activation function of hidden units is symmetric in the input space, and the output of each hidden unit depends only on the radial distance between the input vector X and the center for the hidden unit. The output of each hidden unit, h_j , $j = 1, 2, \dots, J$, is given by

$$h_j(x) = \phi(\|x - c_j\|) \quad (1)$$

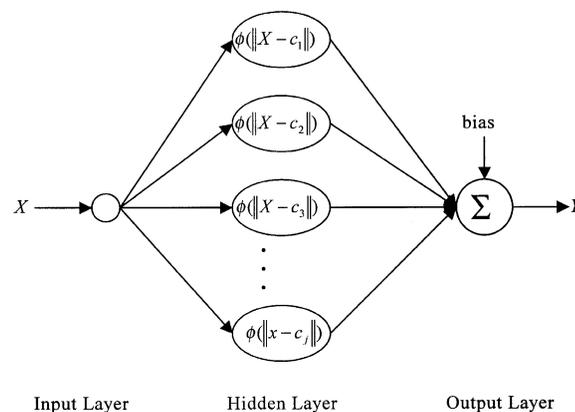


Fig. 1. The structure of RBFN.

where $\| \cdot \|$ is the Euclidean Norm, c_j is the center of the neuron in the hidden layer, and $\phi(\cdot)$ is the activation function, which is a non-linear function and has many types, for example, Gaussian, multiquadric, thin-spline and exponential functions. If the form of the basis function is selected in advance, then the trained RBFN will be closely related to the clustering quality of the training data towards the centers. We use the popular Gaussian function as the transform function in the hidden layer. The Gaussian activation function can be written as

$$\phi_j(x) = \exp\left[-\frac{\|x - c_j\|^2}{2\rho^2}\right] \quad (2)$$

where x is the training data, and ρ is the width of the Gaussian function. A center and a width are associated with each hidden unit in the network. The weights connecting the hidden and output units are estimated using least mean square method. Finally, the response of each hidden unit is scaled by its connecting weights to the output units and then summed to produce the overall network output. Therefore, the k th output of the network \hat{y}_k is

$$\hat{y}_k = w_0 + \sum_{j=1}^M w_{jk} \phi_j(x) \quad (3)$$

where $\phi_j(x)$ is the response of the j th hidden unit, w_{jk} is the connecting weight between the j th hidden unit and the k th output unit, and w_0 is the bias term.

2.2. Learning strategy

To construct a network system, one can use the I vectors to define I radial basis functions. This will lead to the number of hidden neurons equals to that of the data points. When the number of data points is high, the computational time is expensive. Another drawback is that the network may be over-trained. In the neural network, it is often assumed that the number of basis functions is significantly less than that of data points. Therefore the determination of the number of neurons is an important problem for the construction of an RBFN. The placement of neurons and the calculation of the adjusted parameters within the neural system are also important. There are different learning strategies that we can follow in the design of an RBFN, depending on how the centers of

the radial basis functions of the network are specified. Two learning strategies are used herein, namely, hybrid-learning algorithm (Moody and Darken, 1989; Musavi et al., 1992) and fully supervised learning algorithm (Chen et al., 1990 and 1991).

Regarding these two learning algorithms, the hybrid-learning algorithm offers computational efficiency and convergence speed. However, the trained RBFN is closely related to the clustering quality of the training data towards the centers. The K -means clustering is often used in the hybrid-learning algorithm. The values of seeds have a great influence on the quality of clustering using a K -means clustering or a related technique. When the number of clusters is too large, there is probably no training data in the cluster. Moreover, the absence of a method for determining the number of clusters is another drawback. If the fully supervised learning algorithm is used, the hidden unit can be automatically added and the training errors will decrease with increasing number of neurons. Therefore, RBFN is used herein to construct a rainfall-runoff model.

In this paper, the fully supervised learning algorithm is presented for the parametric estimation of the network. First, the network begins with no hidden units. As input and output data are received during training, they are respectively used for generating new hidden units. The location of the first center may be chosen from the training data set, and the standard deviation ρ (i.e. width) of the j th neuron is

$$\rho = \sqrt{\frac{d_{\max}^2}{j+1}} \quad (4)$$

where d_{\max} is the maximum distance between the training data.

When the above steps are finished, a training data point is chosen as the new hidden unit. Then the single hidden layer RBFN linearly combines the output of the hidden units. After output of the network has been obtained, the relative root mean square error (RRMSE) can be calculated. The RRMSE is defined as

$$\text{RRMSE} = \sqrt{\frac{1}{P} \sum_{k=1}^P \left[\frac{\hat{y}_k - y_k}{y_k} \right]^2} \quad (5)$$

where k is the dummy time variable, P is the number of data elements in the period for which computations are made, and y_k and \hat{y}_k are the observation and the forecast at time k , respectively. In a like manner, other data points are chosen sequentially. The training data point having the minimum value of RRMSE is added to the hidden layer until all training data points are tested.

However, a distinct possibility that the model with the best-performing parameter values so selected may end up overfitting the validation data. The network becomes well trained, but its generalization ability is not good. That is, it does not learn enough about the past to generalize in the future. For this reason, the early stopping technique is used to stop the training process. A standard tool in statistics known as cross-validation is used herein as a stopping criterion (Haykin, 1991) in which training, validation and testing data sets are needed. The training data are used to find an optimal set of connecting weights, and the testing data are used to choose the best network configuration. Once an optimal network has been found, a validation is required in order to test the true generalization ability of the model.

In the process of structuring the network, the early stopping technique is used in the following manner. When the RRMSE's of the latter two consecutive neurons are larger than that of the former neuron, then the training of the network is stopped. Using the RRMSE's of two consecutive neurons can avoid the fluctuation of the RRMSE of the validation data. Moreover, the decision as to whether an input-output (x_k, y_k) data should give rise to a new hidden unit depends on the novelty in the data that is decided using the following two conditions

$$\text{num}_{\text{unit}} \leq \text{num}_{\text{max}} \quad (6)$$

$$\text{RRMSE} \leq e_{\text{min}} \quad (7)$$

where num_{unit} is the number of units in hidden layer. Both num_{max} and e_{min} are thresholds to be appropriately selected. If the above two conditions are satisfied, then the data are deemed to have the novelty and a new hidden unit is added. The first condition means that the number of units in the hidden layer must be limited and the second condition means that the error between the network

output and the target output must be significant. The value of e_{min} represents the desired approximation accuracy of the network output. The training procedures of RBFN are summarized in Fig. 2.

3. Application of RBFN

The above methodology is applied to the Fei-Tsui Reservoir Watershed in northern Taiwan for predicting real time streamflows. The length of the main river is 21 km, and the area of the watershed is 303 km². The annual average flow and rainfall are about 50 m³/s and 3,700 mm, respectively. Fig. 3 shows the study area and the locations of six rain gauges (Tai-Ping, Pi-Hu, Pin-Lin, Chi-U-Chiu, Shi-San-Ku, and Fei-Tsui) and one water-level station (Fei-Tsui). The Fei-Tsui water-level station was installed to measure the inflow to the Fei-Tsui Reservoir. Accurate streamflow forecasts are extremely important for the operation of the Fei-Tsui Reservoir. Nine typhoon events are available (see Table 1). The nine typhoon events are divided into three categories: training, validation, and testing. In total, there are 290 training data, 245 validation data and 314 testing data. The time step used for modeling is one hour. Because the inflow of the Fei-Tsui Reservoir cannot be obtained from the water-level station immediately, it can be determined using the following equation

$$I = O + \frac{\Delta S}{\Delta t} \quad (8)$$

where

$$S = f(h) \quad (9)$$

I is the inflow rate (m³/s), O is the outflow rate (m³/s), S is the storage (m³), and h is the water level (m).

The RBFN constructed herein includes 21 nodes in the input layer and one node in the output layer. Among the 21 nodes, three nodes represent the past two-hour and past one-hour, and present inflow rates, and 18 nodes refer to past two-hour, past one-hour, and present rainfall depths of the six rain gauges. In the output layer, there is only one node that represents the 1-h or 2-h or 3-h ahead forecast of streamflow. Four criteria used to evaluate the performance of flow forecasting model are

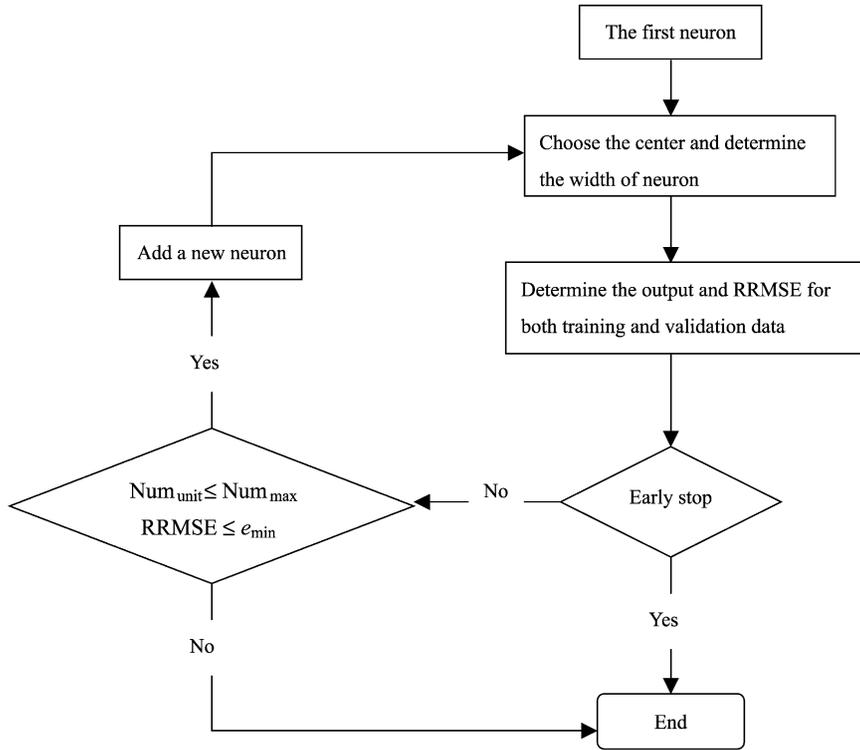


Fig. 2. The flowchart of the RBFN using the fully supervised learning algorithm.

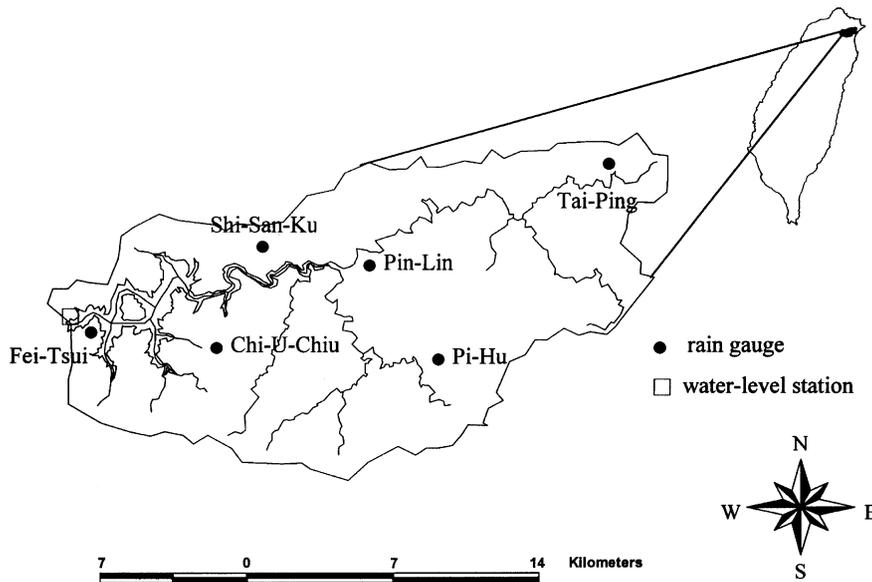


Fig. 3. The Fei-Tsui Reservoir Watershed in northern Taiwan.

Table 1
Description of typhoon events used in the modeling

Number	Name	Date	Duration (hr)	Peak Discharge (m ³ /s)	Remark
1	Polly	1992/08/26	134	970.43	Training
2	Herb	1996/07/30	94	2586.39	Training
3	Gradys	1994/08/31	62	1449.33	Training
4	Tim	1994/07/09	58	673.61	Validation
5	Fred	1994/08/19	93	718.30	Validation
6	Ted	1992/09/20	94	922.00	Validation
7	Ruth	1991/10/27	127	828.36	Testing
8	Seth	1994/10/08	93	1456.78	Testing
9	Doug	1994/08/06	94	535.83	Testing

(1) Error of time to peak discharge:

$$ET_p = \hat{T}_p - T_p \quad (10)$$

where \hat{T}_p and T_p are time to peak for forecasts and observations, respectively.

(2) Error of peak discharge

$$EQ_p = (\hat{Q}_p - Q_p) / Q_p \times 100\% \quad (11)$$

where \hat{Q}_p and Q_p are peak discharges for forecasts and observations, respectively.

(3) Error of total runoff volume

$$EV = \left(\sum_{i=1}^n \hat{Q}_i - \sum_{i=1}^n Q_i \right) / \sum_{i=1}^n Q_i \times 100\% \quad (12)$$

where \hat{Q}_i and Q_i are discharges at time i for forecasts and observations, respectively.

(4) Coefficient of efficiency

$$CE = 1 - \sum_{i=1}^n (Q_i - \hat{Q}_i)^2 / \sum_{i=1}^n (Q_i - \bar{Q})^2 \quad (13)$$

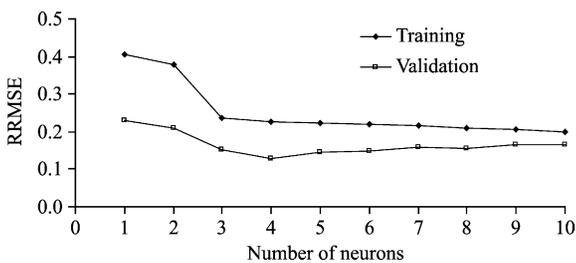


Fig. 4. Variation of RRMSE with the number of neurons for training and validation data.

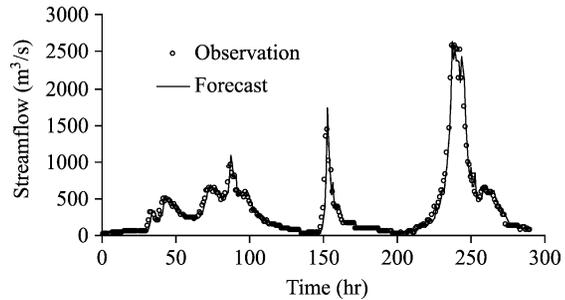


Fig. 5. Comparison of observations and 1-h ahead forecasts for training data.

where \bar{Q} is the average observed discharge; and n is the number of hours of streamflows.

4. Results and discussions

First, the topological structure of RBFN is developed to find the one-hour ahead forecasts of inflow using 290 training data and 245 validation data. Fig. 4 shows the variation of RRMSE with the number of neurons for training data and validation data. As shown in Fig. 4, when the neurons are added serially in the hidden layer, the RRMSE for training data reduces gradually. However, the RRMSE for validation data rises when the number of neurons in the hidden layer is five, and the RRMSE also rises when the sixth neuron is added in the hidden layer. Therefore, training is stopped when the fourth neuron is added. That is, the RBFN has four neurons in the hidden layer.

Fig. 5 compares the observations and 1-h ahead forecasts for training data. The comparison of

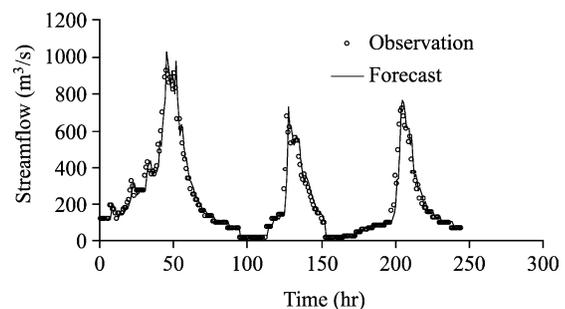


Fig. 6. Comparison of observations and 1-h ahead forecasts for validation data.

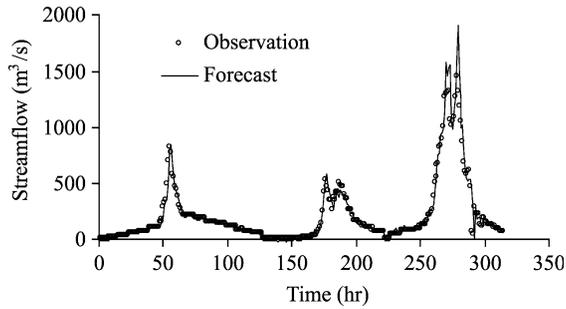


Fig. 7. Comparison of observations and 1-h ahead forecasts for testing data.

observations and 1-h ahead forecasts for validation data is given in Fig. 6. Fig. 7 presents the comparison of observations and 1-h ahead forecasts for testing data. Figs. 5–7 show that the shape as well as the tendency of runoff hydrographs can be reasonably forecasted using the developed model.

The performance of the model in forecasting the 1-h ahead runoff using the fully supervised learning algorithm is summarized in Table 2. According to values of the error of time to peak discharge, ET_p , the forecasts of time to peak are accurate for training and validation data, but those for testing data has a lag time of one hour. The errors of peak discharge, EQ_p ,

Table 2
Performance of the forecasting model using the fully supervised learning algorithm

Number	Name	Criterion											
		ET_p (h)			EQ_p (%)			EV (%)			CE		
		1-h	2-h	3-h	1-h	2-h	3-h	1-h	2-h	3-h	1-h	2-h	3-h
1	Polly	0	0	2	13.8	13.8	21.7	1.8	5.0	11.7	0.96	0.86	0.80
2	Herb	0	0	0	2.2	2.5	4.2	-0.6	-3.1	-5.0	0.98	0.97	0.93
3	Gradys	0	0	2	13.6	14.0	13.2	-2.9	-2.2	-9.3	0.98	0.82	0.70
4	Tim	1	2	2	8.1	12.1	10.2	2.6	5.2	7.1	0.96	0.83	0.76
5	Fred	0	0	1	5.9	10.4	16.5	4.9	4.1	4.3	0.97	0.94	0.92
6	Ted	0	1	2	11.1	14.6	18.2	1.4	0.7	2.7	0.97	0.92	0.81
7	Ruth	1	1	1	2.0	7.2	9.2	-4.9	3.3	8.0	0.96	0.86	0.86
8	Seth	1	2	3	20.4	23.3	27.1	8.6	10.0	12.9	0.95	0.93	0.88
9	Doug	1	1	1	7.1	10.1	21.0	-8.1	3.1	5.9	0.96	0.88	0.79

Table 3
Performance of the forecasting model using the hybrid-learning algorithm

Number	Name	Criterion											
		ET_p (h)			EQ_p (%)			EV (%)			CE		
		1-h	2-h	3-h	1-h	2-h	3-h	1-h	2-h	3-h	1-h	2-h	3-h
1	Polly	2	2	3	9.0	10.9	-6.0	1.4	4.1	6.8	0.89	0.83	0.76
2	Herb	1	2	2	29.8	-7.6	-6.4	-6.3	-4.9	-8.6	0.87	0.83	0.82
3	Gradys	2	2	2	19.4	28.1	25.4	22.0	8.1	15.2	0.70	0.67	0.61
4	Tim	1	2	3	-14.1	-14.7	-4.4	3.4	7.2	12.4	0.89	0.81	0.64
5	Fred	2	2	3	-17.6	14.1	15.8	-0.4	10.8	16.7	0.94	0.89	0.76
6	Ted	2	2	3	9.1	8.5	-5.3	3.6	3.5	4.4	0.91	0.89	0.78
7	Ruth	2	2	3	-14.6	-14.6	-15.0	0.5	0.5	16.6	0.87	0.80	0.65
8	Seth	2	2	3	20.9	21.5	32.3	18.2	18.2	20.2	0.78	0.78	0.72
9	Doug	1	2	3	-7.6	-9.9	-10.3	12.3	12.3	23.9	0.86	0.86	0.61

for nine typhoon events are around 10%. The errors of total runoff volume values, EV, are all smaller than 10%. Moreover, the coefficients of efficiency, CE, are very high for all typhoon events. These results demonstrate the accuracy and reliability of the proposed forecasting model to forecast 1-h ahead runoff. In addition to the 1-h ahead forecasts, Table 2 also gives the performance of the model in forecasting 2-h and 3-h ahead runoff. As indicated in Table 2, the forecasting model using the fully supervised learning algorithm is a reliable and accurate tool for finding the one- to three-hour ahead forecasts of inflow.

For comparisons, another network, which has four neurons in hidden layer, is constructed using the hybrid-learning algorithm. First, the training data set is clustered using a *K*-means clustering. Then, the centers and widths for each neuron in hidden layer are computed. Finally, the network weights can be obtained using the least mean square method. In a like manner, the forecasting model is used to find the one- to three-hour ahead forecasts of inflow. Table 3 summarizes the performance of forecasting model using the hybrid-learning algorithm. From Tables 2 and 3, it is clear that the forecasting model using the fully supervised learning algorithm is a better model. The model using the hybrid-learning algorithm is worse, because the outlier data in the cluster has a great influence in determining the centers of neurons in hidden layer. The centers will have errors and hence the accuracy of the network will decrease.

5. Summary and conclusions

In this paper, the RBFN is used to construct the rainfall-runoff relation. The fully supervised learning algorithm is presented for the parametric estimation of the network. The proposed methodology has been applied to an actual reservoir watershed to forecast the one- to three-hour ahead runoff. Based on our study data, the results show that the RBFN can be

successfully applied to build the relationship between rainfall and runoff. Moreover, the proposed network trained using the fully supervised learning algorithm provides better training and testing accuracy than the network trained using the hybrid-learning algorithm does. The proposed network also gives better forecasts.

References

- Broomhead, D.S., Lowe, D., 1988. Multivariable functional interpolation and adaptive networks. *Complex System* 2, 321–355.
- Chen, S.S., Billings, A., Grant, P.M., 1990. Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *International Journal of Control* 55, 1051–1070.
- Chen, S.C., Cowan, F.N., Grant, P.M., 1991. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks* 2 (2), 302–309.
- Haykin, S., 1991. *Neural Networks: A Comprehensive Foundation*. Prentice Hall Inc., Engle wood, pp. 213–214.
- Hsu, G.L., Gupta, H.V., Sorooshian, S., 1995. Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research* 31, 2517–2530.
- Komda, T., Makarand, C., 2000. Hydrological forecasting using neural networks. *Journal of Hydrologic Engineering* 5 (2), 180–189.
- Marina, C., Paolo, A., Alfredo, S., 1999. River flood forecasting with a neural network model. *Water Resources Research* 35 (4), 1191–1197.
- Moody, J., Darken, C., 1989. Fast learning in networks of locally-tuned processing units. *Neural Computation* 4, 740–747.
- Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., Hummels, D.M., 1992. On the training of radial basis function classifiers. *Neural Network* 5 (4), 595–603.
- Park, J., Sandberg, I.W., 1991. Universal approximation using radial basis function network. *Neural Computation* 3, 246–257.
- Poggio, T., Girosi, F., 1990. Networks for approximation and learning. *Proceedings of the IEEE* 78 (9), 1481–1497.
- Powell, M.J.D., 1987. *Algorithms for Approximations*. Oxford University Press, 143–167.
- Zhu, M.L., Fujita, M., 1994. Comparisons between fuzzy reasoning and neural network methods to forecast runoff discharge. *Journal of Hydroscience and Hydraulic Engineering* 12 (2), 131–141.