

# Enforced self-organizing map neural networks for river flood forecasting

Fi-John Chang,<sup>1\*</sup> Li-Chiu Chang<sup>2</sup> and Yan-Shiang Wang<sup>1</sup>

<sup>1</sup> Department of Bioenvironmental Systems Engineering, National Taiwan University, Taipei, Taiwan, ROC

<sup>2</sup> Department of Water Resources and Environmental Engineering, Tamkang University, Tamsui, Taipei Hsien, Taiwan, ROC

## Abstract:

Self-organizing maps (SOMs) have been successfully accepted widely in science and engineering problems; not only are their results unbiased, but they can also be visualized. In this study, we propose an enforced SOM (ESOM) coupled with a linear regression output layer for flood forecasting. The ESOM re-executes a few extra training patterns, e.g. the peak flow, as recycling input data increases the mapping space of peak flow in the topological structure of SOM, and the weighted sum of the extended output layer of the network improves the accuracy of forecasting peak flow. We have investigated an ESOM neural network by using the flood data of the Da-Chia River, Taiwan, and evaluated its performance based on the results obtained from a commonly used back-propagation neural network. The results demonstrate that the ESOM neural network has great efficiency for clustering, especially for the peak flow, and super capability of modelling the flood forecast. The topology maps created from the ESOM are interesting and informative. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS self-organizing map (SOM); artificial neural network; rainfall-runoff process; flood forecasting

Received 17 March 2005; Accepted 22 November 2005

## INTRODUCTION

Modelling the flood process over a watershed is inherently complex, highly non-linear and temporally and spatially non-uniform. In the last century, hydrological systems were mainly explored through hydrological models, and a great deal of research has been devoted to the modelling of the rainfall–runoff process. These include black-box/stochastic models and physical/conceptual models (e.g. Salas *et al.*, 1985; Hydrological Engineering Center, 1990). Although physically based models are useful for understanding the mechanisms involved in the process, the black-box models are easier to implement. Yet, with their inherently simple assumptions and/or lack of verified data/parameters, these existing models might not be always satisfying and/or might be difficult to implement for real-time flood forecasting.

The aforementioned difficulties have led hydrologists to explore the artificial neural networks (ANNs) as a way of obtaining models based on historical data. Recently, applications of ANNs have provided a promising alternative for rainfall–runoff modelling and streamflow forecasting (Imrie *et al.*, 2000; ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a,b; Chang *et al.*, 2004; Rajurkar *et al.*, 2004) and for reservoir operation in water resources management (Jain *et al.*, 1999; Coulibaly *et al.*, 2000; Neelakantan and Pundarikanthan, 2000; Chang and Chang, 2001).

For modelling a flood forecast, an ANN is usually established by using several preceding steps' (hours or days) data of the upstream rainfall and streamflow gauges as input. A major problem in constructing an ANN is that if the dimension of the input is great, e.g. using a time-delay pattern of several rainfall and/or flow gauges as input, the number of hidden neurons will be huge and the network will be complex (Bowden *et al.*, 2002). To reduce the complexity, clustering ANNs or clustering techniques have been applied for hydrological systems (e.g. Del Genio and Kovari, 2002; Luchetta and Manetti, 2003; Moradkhani *et al.*, 2004; Chang *et al.*, 2005). Clustering networks exhibit remarkable properties of self-organization. These properties are closely related to the formation of knowledge representation in information theory.

In this study, we investigate one of most famous clustering networks, i.e. the self-organizing map (SOM) for flood forecasting. The SOM algorithm was originally introduced by Kohonen (1982). From the biological viewpoint, it has several properties as essential capabilities of the brain, such as lateral interactions and topological structure. An SOM generates lower dimensional topological ordered maps of input data through learning, which is very useful of analysing high-dimensional data. Subsequently, it has been widely applied in a range of applications from pattern recognition, classification, signal processing, speech processing and reverse engineering of geometric models (Barhak and Fischer, 2002). Once a SOM is determined, the output of the network to the input vectors can be recalled from the classifying results memorized in the network. However, this way

\* Correspondence to: Fi-John Chang, Department of Bioenvironmental Systems Engineering and Hydrotech Research Institute, National Taiwan University, Taipei, Taiwan, ROC. E-mail: changfj@ntu.edu.tw

is only good for pattern recognition problems and cannot be effectively used for continuous function approximation (such as flood discharge). To be more generalized in extended the use of the topological structure of SOM for flood forecasting, the hidden mapping nodes are extended to an output layer and their connected weights are adjusted through a supervised learning algorithm so that the rainfall–runoff process can be well established.

A detailed review of the SOM algorithm is first presented in the following section. Then the enforced training procedure of the original SOM learning process, the enforced SOM (ESOM), is presented and the effect of the new training procedure is illustrated with an example. A brief description of the least-mean-squares (LMS) algorithm for output weights optimization is then given, and this is followed by a summary of the back-propagation neural network (BPNN) algorithm. We then apply these models (BPNN, SOM, and ESOM) to forecast flood events in the Da-Chia River, Taiwan. The results and conclusion are then given in the final two sections.

THE SELF-ORGANIZING MAP ALGORITHM

Figure 1 shows the architecture of an SOM neural network, which consists of three layers, i.e. (1) input layer ( $n$  neural units), (2) hidden mapping layer (two-dimensional matrix), and (3) output layer. An SOM, also called a Kohonen SOM, is a two-layer network that can map the multidimensional space onto a two-dimensional (2-D) space, preserving the original order as well. The output layer is extended to provide the specific value for a giving input pattern (discussed in the ‘The least-mean-squares algorithm for optimizing output weights’ section). The SOM algorithm is an unsupervised classification that uses a competitive learning strategy to adjust the connected weights between input and hidden layers and to form a topographically ordered map in the hidden layer. Different from other clustering methods for unsupervised data, the SOM can be highly non-linear, directly showing the similar input vectors in the source space by points (neurons) close in the 2-D target space. Ham and Kostanic (2001) summarized the learning

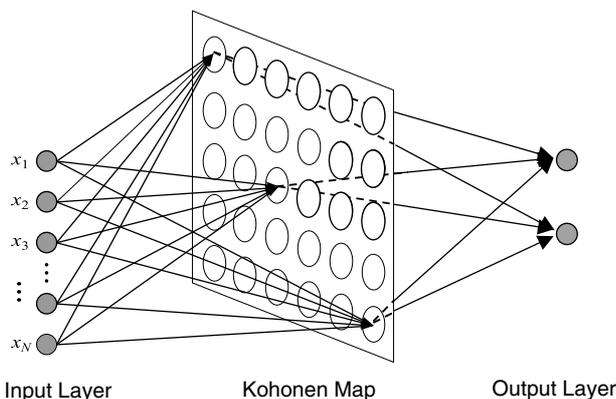


Figure 1. The architecture of an SOM neural network

algorithm of training connected weights in an SOM as follows.

For simplicity, we assume that the number of a 2-D array of neurons is  $m$  and the external input vector  $X$  is  $n$ -dimensional:

$$X = [x_1, x_2, \dots, x_n]^T \tag{1}$$

The synaptic weight vector connecting input vector to the hidden neuron  $i$  is denoted by

$$W_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T \quad i = 1, 2, \dots, m \tag{2}$$

In competitive learning networks, the neurons compete among themselves to determine a winner (or a winning neuron), and then the network simply adjusts the winner and its neighbourhood neurons. Thus, a winner  $q$  is defined as the one whose weight vector is the closest to the input vector  $X$ , i.e.

$$q(X) = \min_i \|X - W_i\| \quad i = 1, 2, \dots, m \tag{3}$$

The topological neighbourhood function of the winner  $q$  is considered to decrease monotonically with the number of iterations  $k$  shown in Figure 2 and defined as

$$\eta_{qi}(k) = \eta(k) \exp \left[ -\frac{\|r_i - r_q\|^2}{R(k)^2} \right] \tag{4}$$

where the vectors  $r_q$  and  $r_i$  are the coordinates of the neurons  $q$  and  $i$  respectively;  $\eta(k)$  and  $R(k)$  are chosen as decreasing functions of time. The learning rule is given by

$$W_i(k+1) = \begin{cases} W_i(k) + \mu(k)\eta_{qi}(k)[X(k) - W_i(k)] & i \in \eta_{qi}(k) \\ W_i(k) & i \notin \eta_{qi}(k) \end{cases} \tag{5}$$

where  $0 < \mu(k) < 1$ , the learning rate parameter, should decrease with  $k$ .

The above mapping procedure can be described briefly as follows:

1. Initialize the network weight vectors.
2. Randomly chose an input vector from the input space.
3. Determine the winning neuron by calculating the Euclidean distance between the input vector and the

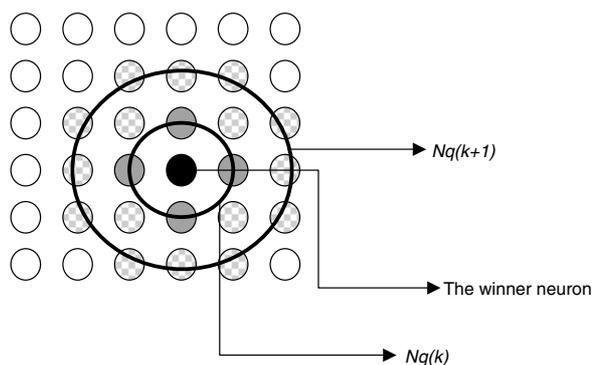


Figure 2. Topological neighbourhood around the winner

weight vectors of all neurons in the hidden layer (Equation (3)).

4. Adjust the weight vector of the winner and the weight vectors of its neighbouring neurons according to the learning rule (Equation (5)).
5. Iterate the procedures from (2) until the weight vectors stabilize.

After a large number of iterations, each input vector is mapped onto a specific neuron in the hidden layer in such a way that the weight vector of the neuron is closer to the input vector.

#### THE PROPOSED ENFORCED SELF-ORGANIZING MAP MODEL: ENFORCED TRAINING PROCEDURE

Although the SOM algorithm is a powerful learning process for clustering the input patterns into groups of similar patterns, it cannot always satisfy diverse characteristics of data distribution. For example, Figure 3 shows a set of 2-D input vectors corresponding to five classes. These points are generated from a uniform distribution in  $[0, 1]$  and are classified as five classes according to their positions; most of them, located in the centre of the data space, are classified as class  $C_1$ ; the top, the bottom, the right and the left parts of the boundary are classified as  $C_2$ ,  $C_3$ ,  $C_4$  and  $C_5$  respectively. Figure 4 shows that the neurons of the SOM map categorize a uniform distribution of input patterns. It is easy to tell that those points, which are located in the boundary of the input space, are not well represented by the constructed neurons. The major reason is that the SOM, like other cluster methods, can only represent the distribution of the major parts of input points and ignores the small parts of input points. In a real-world implementation of a model, the events that rarely happen, however, might be the main concern or the key issue, such as the peak flow in a flood.

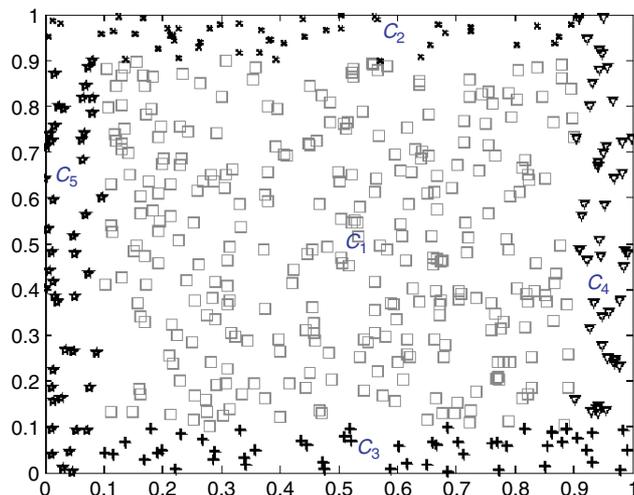


Figure 3. Uniform distribution of data points with five clustering subsets

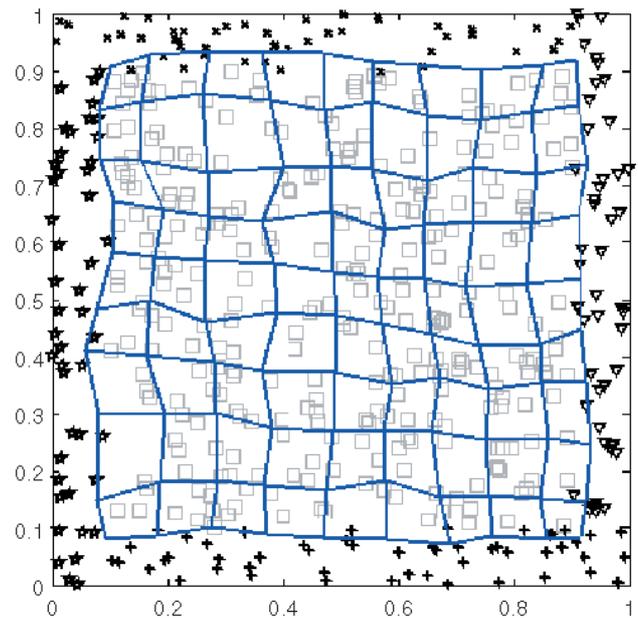


Figure 4. The topological map formation in the simulation of a  $10 \times 10$  SOM network

To fix this shortcoming, the important information (inputs) should be emphasized. Hsu *et al.* (1999) described a way to highlight important patterns, but which were less frequently sampled in training data, by screening the data through a regular spacing grid filter and demonstrated that the extremely huge volume of bits of information in a remotely sensed image data could be more effectively represented and efficiently managed. Another simple way to solve the problem in cluster analysis (or cluster neural network) is through repeatedly implementing those specific patterns. The idea is similar to the human learning process: when we try to learn and memorize an important and specific phenomenon, we practise it repeatedly. The proposed ESOM is based on this knowledge learning process; it enforces a few extra training iterations for the converged SOM network by using a specific data group, namely a set of enforced data. Because the SOM is a competitive learning network, in which only one neuron is 'on' at a time, retraining the SOM network by using the enforced data will only cause the related cluster centres to be moved in the direction of these specific groups. In the above example, we choose the points in four classes,  $C_2$ ,  $C_3$ ,  $C_4$  and  $C_5$ , in Figure 3 as the set of enforced data to follow a few training iterations to the above converged SOM. Figure 5 shows the neurons of the ESOM are moved in the direction of the boundary of the input space, and many more neurons represent classes of  $C_2$ ,  $C_3$ ,  $C_4$  and  $C_5$  and fewer neurons represent class  $C_1$ , in comparison with the SOM in Figure 4.

To build up a flood forecast model, the peak flow data are the most valuable part, but these data are rare. Our challenge is to obtain an appropriate topological neural network for flood forecasting through improving the clustering process of the SOM and enforcing its training process by reusing high flow data. In this way, we

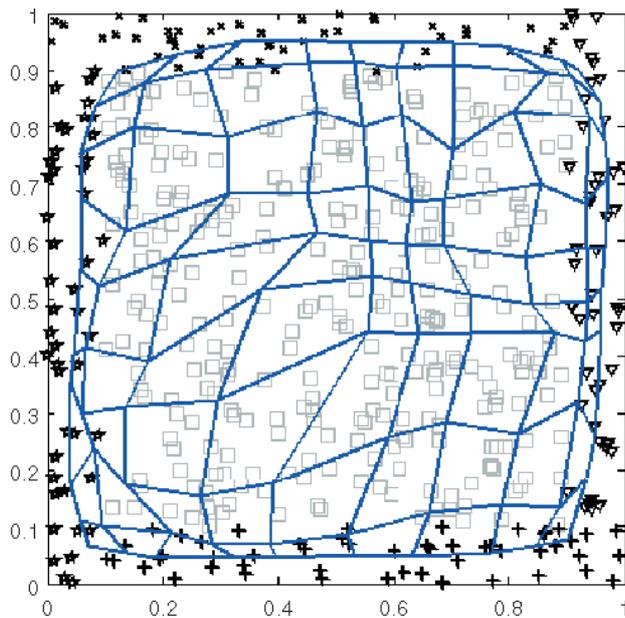


Figure 5. The topological map formation in the simulation of a 10 × 10 ESOM network

expect this would not only increase the mapping spaces of peak flow in the topological structure of the SOM, but also improve the performance of flood forecasting at high flows. Figure 6 shows the flowchart of the SOM and ESOM training procedure.

THE LEAST-MEAN-SQUARES ALGORITHM FOR OPTIMIZING OUTPUT WEIGHTS

The SOM uses an unsupervised learning whereby similar input patterns are clustered into a neuron without using the corresponding output information. In the real world, similar input patterns, however, could have various outputs; consequently, we face the challenge of determining the output of each hidden neuron, or more specifically the output of the network when we give an input pattern. One easy way to determine the output for a given input pattern is to use the average output value for the clustered input patterns to the corresponding neuron and then directly use the output of the closest (most similar) neuron for the given input pattern. By combining the SOM and learning vector quantization, an adaptive pattern classification system can be developed (Kohonen, 1990). Hsu *et al.* (2002) used a linear regression function to estimate the output for each selected input cluster. To be more generalized in using the topological structure of the SOM/ESOM, we propose that the output of the network is computed as a weighted sum of the hidden-layer outputs. In this way all the values of the topological matrix ( $n \times n$  neurons) corresponding to an input vector can be effectively represented (computed by Equation (7)) and the output of the network for the input pattern can be efficiently estimated (computed by Equation (6)). This method is similar to the strategy that is commonly used in the radial basis

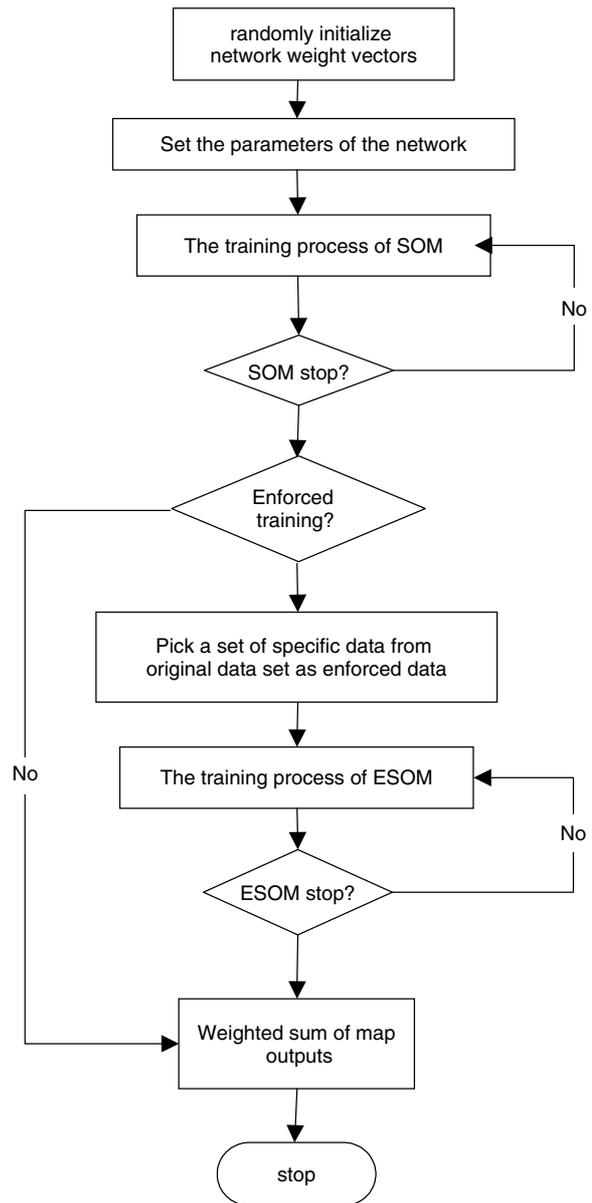


Figure 6. The flowchart of the SOM and ESOM neural network training procedure

function neural networks. The learning (training) algorithm for the output-connected weights is summarized as follows.

Once the centres are determined from the SOM or ESOM, the output of the network to the input vectors can be computed as

$$\hat{y}(q) = \sum_{i=1}^m \alpha_i \phi(X(q), W_i) \quad q = 1, 2, \dots, Q \quad (6)$$

$$\phi(X(q), W_i) = \phi(\|X(q) - W_i\|_2) \quad q = 1, 2, \dots, Q \quad (7)$$

$$\phi(a) = \exp(-a^2)$$

where  $\hat{y} \in \mathbb{R}^{1 \times 1}$  is the actual network output (estimation),  $X \in \mathbb{R}^{n \times 1}$  is the input vector,  $Q$  is the number of samples,  $\phi_i(\cdot)$  is a Euclidean function designed for each neuron of the SOM hidden layer,  $\alpha_i$  are the connected weights in the output layer,  $m$  is the number of neurons in the hidden

layer, and  $W_i \in \mathfrak{R}^{n \times 1}$  are the SOM or ESOM centres in the hidden layer.

The optimal set of connected weights minimizes the performance measures

$$J(w) = \frac{1}{2} \sum_{q=1}^Q [y_d(q) - \hat{y}(q)]^2 \quad (8)$$

where  $y_d \in \mathfrak{R}^{1 \times 1}$  denotes the desired network output (observation). The connected weights (i.e.  $\alpha_i$ ) can be obtained by using the training sets (input–output patterns) through the LMS algorithm. After the training procedure is completed, the network output for a give input vector (in the training or testing set) can then be easily obtained by using Equation (6).

### BACK-PROPAGATION NEURAL NETWORKS

The back-propagation neural network (BPNN) is the most popular and widely used neural network in use today. It is trained by using supervised learning, so the goal of this algorithm is to decrease global error. To obtain the optimal values of the connected weights such that the energy function is a minimum, the standard back-propagation (BP) algorithm searches the error surface by using the steepest descent method. The connected weights are adjusted by moving a small step in the direction of the negative gradient of the energy function at each iteration. Details on the standard BP algorithm can be found in the literature (Rumelhart *et al.*, 1986).

The advantage of this neural network is that it can easily be implemented because of its simple algorithm and/or reasonably priced (or free) commercial software. Nonetheless, the network also has several drawbacks for some applications: (1) it may fail to produce a satisfactory solution because the training data are insufficient in size; (2) it easily falls into a local minimum and the speed of convergence of the network can be very slow when the number of inputs in the data set is large (Chiang *et al.*, 2004). The performance of the BPNN is investigated through a different number of hidden nodes and a great number of initial settings for the connecting weights. These results are used as the base for evaluating the performance of our proposed ESOM neural network.

### APPLICATION

In this paper, the proposed ESOM coupled with BP networks is applied to the Da-Chia River for flood forecasting. Located in central Taiwan, the Da-Chia River is about 140 km long with an average channel slope of 1/39 and a total watershed size of 1236 km<sup>2</sup>. With the steepest channel in Taiwan, the Da-Chia River has a series of hydraulic structures for power generation. The locations of the study basin (area: 514 km<sup>2</sup>) and the gauging stations used are shown in Figure 7. The Son-Mou gauging station was established to measure the



Figure 7. Location of gauging stations upstream of the Da-Chia River

inflow of the De-Chi Reservoir, the most pivotal reservoir in the Da-Chia River. During flood periods, efficient flood forecasting is extremely important for dam safety and/or water resource management. Above the Son-Mou streamflow gauge (denoted by a triangle) there are nine rainfall gauges (denoted by circles). From 1981 to 2000, hourly data were collected from these gauging stations, consisting of 88 storm events with 4231 observations. These events are divided into training and testing subsets, which have roughly the same statistical properties (mean and variance). The training data has 78 flood events with 3696 observations. Among them, the upper 20% of flow (about 760 observations) is picked as the enforced data set to retrain the converged SOM; consequently, the mapping spaces of peak flow in the topological structure of the SOM would be increased and the performance of flood forecasting at high flows could be improved. The testing subset data has 10 flood events with 535 observations, which are used to test the goodness of fit of the SOM networks built.

### Error measure indices

The criteria of mean-square error (MSE), mean absolute error (MAE) and the goodness of fit with respect to the benchmark  $G_{\text{bench}}$  (Nash and Sutcliffe, 1970; Seibert, 2001) are used for evaluating the performance of the models. The MSE provides a good measure of the goodness of fit at high flows, and the MAE is suitable for measuring the accuracy of the whole flow data (Dawson and Wilby, 2001). These are defined as

$$\text{MSE} = \frac{\sum_{i=1}^n (\hat{Q}_i - Q_i)^2}{n} \quad (9)$$

$$MAE = \frac{\sum_{i=1}^n |\hat{Q}_i - Q_i|}{n} \tag{10}$$

$$G_{\text{bench}} = 1 - \frac{\sum_{i=1}^n (\hat{Q}_i - Q_i)^2}{\sum_{i=1}^n (Q_i - \tilde{Q}_i)^2} \tag{11}$$

where  $Q_i$  is the observed value in step  $i$ ,  $\hat{Q}_i$  is the forecasted value in step  $i$ , and  $\tilde{Q}_i$  is the previous observed value in step  $i$ , e.g.  $\tilde{Q}_i = Q_{i-1}$  for 1-h-ahead and  $\tilde{Q}_i = Q_{i-2}$  for 2-h-ahead forecasting in our case.

*Determining the structure of the models*

The architecture of the SOM neural networks includes three layers: input layer, clustering layer and output layer. Because one of the aims of this study is to demonstrate that the SOM is an effective tool for abstraction of high-dimensional data, we used time-delay information, i.e. the past  $p$  values, of the runoff and nine rainfall gauge stations as input variables. The time-delay information is roughly estimated according to the geographical distribution and the various travel times of the flow from the rainfall gauge stations to the Song-Mao streamflow gauge station. The output of the models is 1-h- or 2-h-ahead streamflow of the Song-Mao gauge station. The number of input variables, summarized in Table I, is 24.

*Finding the optimal size of the networks*

Determining the appropriate size of clustering neural networks is very important for cluster validity and efficiency. Too large a network size would cause each neuron to memorize one of the input patterns, and thereby waste the computational time, but without significant improvement of the forecasting results. On the other hand, if the network size is too small, then many different data groups might be lumped into the same category and not suitably represented by any neuron. Since there is no systematic or standard method for finding the optimal number of clusters in the clustering algorithms, the optimal network size is obtained using trial and error. We vary the network sizes from  $12 \times 12$ ,  $16 \times 16$  to  $20 \times 20$ . In Table II, the  $20 \times 20$  network has the best results in the training phase, but does not have the best

Table I. Input variables of the networks investigated

Time	Input variables <sup>a</sup>
$t - 1$	R1( $t - 1$ ), R2( $t - 1$ ), R3( $t - 1$ ), R4( $t - 1$ ), R5( $t - 1$ ), R6( $t - 1$ ), R7( $t - 1$ ), R8( $t - 1$ ), R9( $t - 1$ ), $Q(t - 1)$
$t - 2$	R1( $t - 2$ ), R2( $t - 2$ ), R3( $t - 2$ ), R4( $t - 2$ ), R6( $t - 2$ ), R7( $t - 2$ ), R8( $t - 2$ ), R9( $t - 2$ ), $Q(t - 2)$
$t - 3$	R1( $t - 3$ ), R2( $t - 3$ ), R3( $t - 3$ ), R8( $t - 3$ ), $Q(t - 3)$

<sup>a</sup> Rainfall gauge stations: Ping-Yan-Shan (R1), Zhi-Jia-Yang (R2), Si-Yuan (R3), Men-Shan(R4), Song-Mao (R5), Tao-Shan (R6), Li-Shan(R7), Wu-Ming-Shan (R8), Huan-Shan (R9).

Table II. The performance of the different-sized SOM networks

Network size	Phase	MSE	MAE	$G_{\text{bench}}$
$12 \times 12$	Training	206	6.16	0.58
	Testing	290	7.55	0.58
$16 \times 16$	Training	177	5.79	0.62
	Testing	<b>265</b>	<b>7.10</b>	<b>0.61</b>
$20 \times 20$	Training	159	5.81	0.66
	Testing	274	7.11	0.60

Table III. The performances of the BPNN with the various numbers of hidden nodes

No. of nodes	Training			Validation		
	MSE	MAE	$G_{\text{bench}}$	MSE	MAE	$G_{\text{bench}}$
1	407	8.15	0.50	582	8.84	0.29
2	410	8.39	0.50	594	9.02	0.27
3	411	8.76	0.50	560	9.12	0.31
4	402	8.87	0.51	525	9.17	0.36
5	405	8.94	0.50	539	9.25	0.34
6	393	9.04	0.52	500	9.22	0.39
7	410	9.28	0.50	515	9.24	0.37
8	412	9.52	0.49	512	9.47	0.37
9	427	9.64	0.48	529	9.64	0.35
10	423	9.83	0.48	508	9.49	0.38

results in the testing phase. The results suggest that it might not be the optimal network structure because it needs more computational time and may be overtrained or overfitting. We find that the results of the  $16 \times 16$  network are good in the training phase and are among the best in the testing phase; thus, the  $16 \times 16$  network was chosen for later use.

To determine the optimal BPNN for the given input-output patterns, we also investigated several network architectures. Because the input vector is set the same as previously, i.e. 24, and a 1-h ahead forecast is performed, i.e. the output vector is unity, only the number of nodes in the hidden layer need to be determined. Networks with various numbers of hidden nodes, from 1 to 10, were examined; furthermore, each network was executed with 50 initial sets of connected weights to make sure the final results were reasonable. Because some of the given initial sets for the networks might not be converged after a number of searches, 2500 epochs in our case, we only record the best 30 results and drop the other 20 results for each network. The average of the best 30 results (MSE, MAE, and  $G_{\text{bench}}$ ) for each network is shown in Table III and Figure 8. It appears that when the hidden layer has six nodes, both training and validation sets have the best performances. These results are used as a base to evaluate the adequacy of our proposed ESOM neural networks.

RESULTS AND DISCUSSION

The size of the SOM network determined in the previous section is  $16 \times 16$  grids. Based on this trained network,

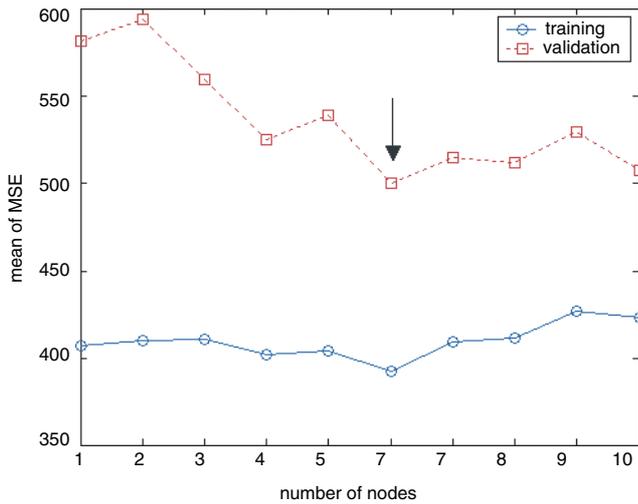


Figure 8. The selection of hidden nodes for a standard BPNN

Table IV. Comparison of one-step-ahead streamflow forecasting among the standard BPNN, SOM and ESOM models

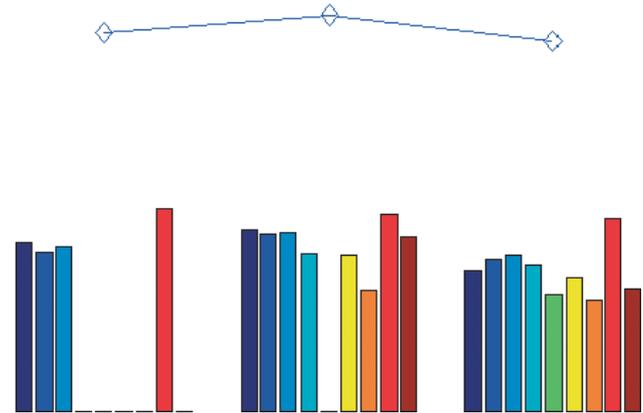
Model	Training			Testing		
	MSE	MAE	$G_{\text{bench}}$	MSE	MAE	$G_{\text{bench}}$
BPNN (mean)	393	9.04	0.52	500	9.11	0.39
BPNN (minimum)	336	7.56	0.59	388	8.21	0.52
SOM network	177	5.79	0.62	265	7.10	0.61
ESOM network	97	4.97	0.79	241	7.15	0.65
SOM network ( $>180 \text{ m}^3 \text{ s}^{-1}$ )	698	14.70	0.66	1497	26.31	0.62
ESOM network ( $>180 \text{ m}^3 \text{ s}^{-1}$ )	339	11.77	0.84	1360	25.73	0.66

the ESOM algorithm is performed by using the 760 selected data sets (i.e. higher flow observations) to execute a few extra training iterations to obtain a new topology.

The results of the BPNN, SOM, and ESOM are summarized in Table IV. The results show that (1) the SOM networks could adequately produce one-step ahead streamflow forecasting with small MAE and high  $G_{\text{bench}}$  value ( $>0.6$ ), and (2) the ESOM does improve the forecasting ability of the SOM, which results in lower forecasting error and a higher  $G_{\text{bench}}$  value compared with the SOM. As we investigate the model performance in the training phase during the high flow (i.e. flow  $>180 \text{ m}^3 \text{ s}^{-1}$ ), the MSE values of the SOM and ESOM are  $698 \text{ m}^3 \text{ s}^{-1}$  and  $339 \text{ m}^3 \text{ s}^{-1}$  respectively. A great improvement has been achieved. In the testing phase, however, the improvement is less significant than in the training phase. This is mainly because there are only a few flow data (87 among 535 tested observations) that are greater than upper 20% flow (i.e.  $180 \text{ m}^3 \text{ s}^{-1}$ ) in the testing set. To investigate the applicability of the SOM and ESOM networks further on 2-h-ahead forecasting, the above procedure was re-executed (the topology map remained the same and only the weights of the output layer need to be retrained); the results are shown in Table V. Again, we can easily find that both the SOM

Table V. Comparison of two-steps-ahead streamflow forecasting between the SOM and ESOM models

Model	Training			Testing		
	MSE	MAE	$G_{\text{bench}}$	MSE	MAE	$G_{\text{bench}}$
SOM network	593	10.84	0.62	887	13.04	0.62
ESOM network	404	9.84	0.74	813	12.53	0.65
SOM network ( $>180 \text{ m}^3 \text{ s}^{-1}$ )	2396	28.23	0.66	5020	48.74	0.63
ESOM network ( $>180 \text{ m}^3 \text{ s}^{-1}$ )	1499	24.87	0.79	4581	47.12	0.67



Bar Chart: rainfall  
Line plot: streamflow

Figure 9. The 24 weights (variables) in an SOM (or ESOM) neuron

and ESOM networks are well trained, in terms of small MSE and MAE values and reasonably good  $G_{\text{bench}}$  value in the training phase, and can suitably make a 2-h-ahead forecast (high  $G_{\text{bench}}$  value in the testing phase). Furthermore, it appears that the ESOM network, again, has much smaller MSE and MAE values and a higher  $G_{\text{bench}}$  value than the SOM network, especially in the training phase.

One of the most significant characteristics and contributions of using an SOM to construct the rainfall–runoff flood forecasting process is that the results obtained by the SOM can be visualized from its topology map. As mentioned in the previous section, our input vector includes 24 variables; consequently, each SOM node has 24 weights, corresponding to each of the 24 input variables. To illustrate the distribution of weights across the SOM matrix, Figure 9 was constructed consisting of three groups of nine vertical bars (representing the rainfall inputs at nine stations for three time steps ( $t - 2, t - 1, t$ )) and three line-connected diamonds (representing three streamflow inputs ( $Q(t - 2), Q(t - 1), Q(t)$ )). The length of the vertical bar is proportional to the strength of the rainfall amount, and the relative vertical position of the diamond represents the strength of the streamflow contribution. The ESOM topology map is shown in Figure 10. The map has  $16 \times 16$  neurons and each neuron represents a cluster of input vectors (as shown in Figure 9). Figure 10 shows that: (1) the upper-left side gathers most

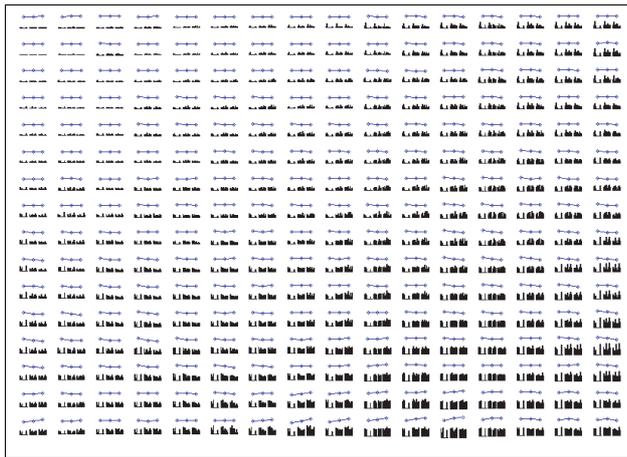


Figure 10. The topology maps of the ESOM (16 × 16 neurons)

of the lowest input values and the lower-right side has most of the highest input values; (2) the nearby neurons are quite similar, whereas an increasing trend of input amount from upper left to the lower right can also easily be detected. Hsu *et al.* (2002) used daily rainfall and streamflow data (six variables) in a river station and obtained a quite impressive relationship between five classification regions in their 15 × 15 SOM units and rainfall–runoff process histogram. The topology map

obtained in this study, however, does not look like Hsu *et al.*'s map. This is mainly because our input vector includes 24 variables (the time-delay information from nine rainfall stations and one flow station). It appears that the time and space variability of rainfall dominate the clustering process; consequently, only the general tendency of the rainfall amount at all the rainfall stations (as well as discharge at the flow station) could be obtained (detected).

Figure 11 shows a scatter plot of observed versus one-step ahead forecasting of the ESOM network in the training phase. One can easily tell that the forecast values of the ESOM are much closer to the ideal line than the SOM. Figure 12 represents the 2-h-ahead forecast results in the testing phase, where both models fit the ideal line reasonably, and the SOM has a slightly wider spread from the ideal line than the ESOM.

CONCLUSIONS

ANNs have been successfully applied in many hydrological and water resource problems, and usually they can provide more promising results than with other, conventional models. Nevertheless, it is well known that the ANN is considered as a non-linear black-box model, and it is not unusual for it to be criticized as not enhancing

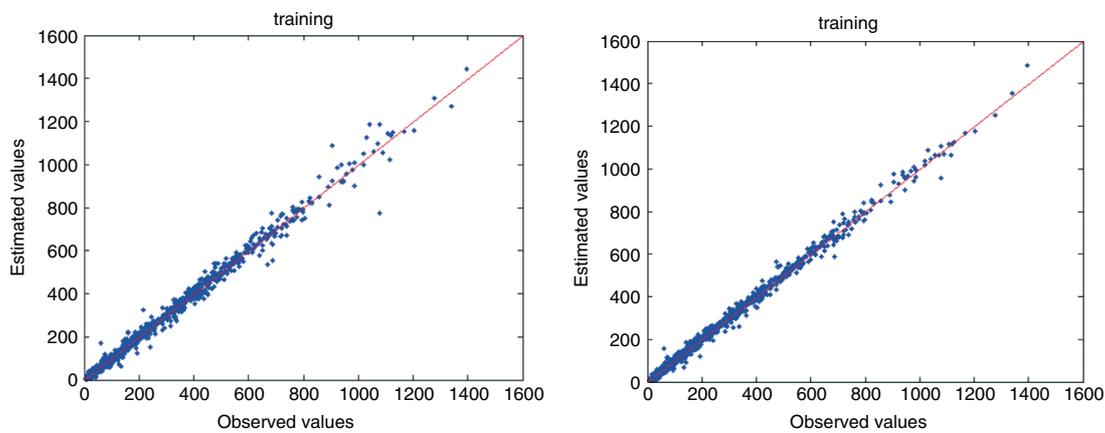


Figure 11. Comparison of observed and one-step-ahead estimated values of the (a) SOM and (b) ESOM models in the training set

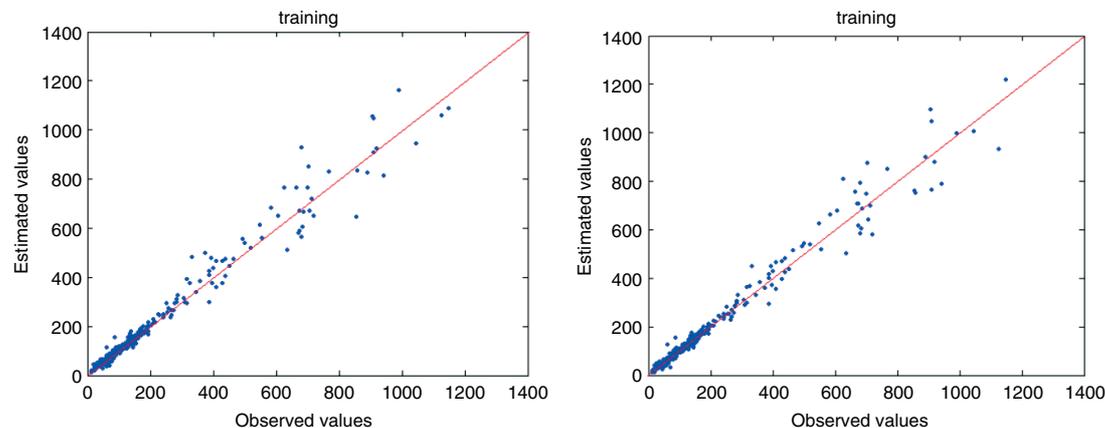


Figure 12. Comparison of observed and two-steps-ahead estimated values of the (a) SOM and (b) ESOM models in the testing set

our understanding of the physical mechanisms involved in river flow dynamics. The SOM is an effective tool for visualization and/or abstraction of high-dimensional data and has been very successfully accepted in various fields. Our aim in this study is to present the applicability of the SOM for constructing the rainfall–runoff process and to propose an ESOM neural network for enhancing the prediction accuracy, especially during a flood period. ESOM re-executes a few extra high-flow training patterns to increase the mapping space of peak flow in the topological structure of the SOM, and the weighted sum of the extended output layer of the network improves the accuracy of forecasting peak flow. The connected weights of the output layer are obtained through the LMS algorithm.

To evaluate the model ability and accuracy for flood forecasting, we executed the commonly used BPNN with various sizes of hidden nodes and a larger number of initial sets of weights for each network (size) to provide a base for the evaluation. From the results obtained in this study, we conclude that: (1) SOM neural networks can adequately produce streamflow forecasting; (2) by means of recycling the high-flow data to retraining the SOM network, the proposed ESOM network not only increased the mapping spaces of peak flow in the topological structure of the SOM, but also improved the performance of flood forecasting at high flows.

To illustrate the distribution of weights across the SOM matrix ( $16 \times 16$  neurons), we designed a figure that can represent the variability of all 24 input variables in each clustered neuron. Owing to the great number of input variables and the natural time and space variability of rainfall, each clustered neuron could only represent the lumped effect of the input variables, and the topology map shows an increasing trend of input amount from upper left to the lower right. It is well recognized that the prediction capability of the resulting ANN model is highly dependent on the selection of the input variables. The SOM matrix will be investigated further to determine what input variables should be included in a watershed rainfall–runoff process and what information of system behaviour can be detected (visualized).

#### ACKNOWLEDGEMENTS

This paper is based on partial work supported by the National Science Council, Taiwan, ROC (grant no. NSC90-2313-B002-261).

#### REFERENCES

- ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. 2000a. Artificial neural networks in hydrology I: preliminary concepts. *Journal of Hydrologic Engineering, ASCE* **5**(2): 115–123.
- ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. 2000b. Artificial neural networks in hydrology II: hydrologic applications. *Journal of Hydrologic Engineering, ASCE* **5**(2): 124–137.
- Barhak J, Fischer A. 2002. Adaptive reconstruction of freeform objects with 3D SOM neural network grids. *Computers & Graphics* **26**: 745–751.
- Bowden GJ, Maier HR, Dandy GC. 2002. Optimal division of data for neural network models in water resources applications. *Water Resources Research* **38**(2): 2-1–2-11.
- Chang LC, Chang FJ. 2001. Intelligent control for modelling of real time reservoir operation. *Hydrological Processes* **15**: 1621–1634.
- Chang LC, Chang FJ, Chiang YM. 2004. A two-step ahead recurrent neural network for streamflow forecasting. *Hydrological Processes* **18**: 81–92.
- Chang LC, Chang FJ, Tsai YH. 2005. Fuzzy exemplar-based inference system for flood forecasting. *Water Resources Research* **41**: W02005. DOI: 10.1029/2004WR003037.
- Chiang YM, Chang LC, Chang FJ. 2004. Comparison of static-feedforward and dynamic-feedback neural networks for rainfall–runoff modeling. *Journal of Hydrology* **290**: 297–311.
- Coulbaly P, Anctil F, Bobee B. 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology* **230**: 244–257.
- Dawson CW, Wilby RL. 2001. Hydrological modeling using artificial neural networks. *Progress in Physical Geography* **25**(1): 80–108.
- Del Genio AD, Kovari W. 2002. Climatic properties of tropical precipitating convection under varying environmental conditions. *Journal of Climate* **15**(18): 2597–2615.
- Ham FM, Kostanic I. 2001. Principles of neurocomputing for science & engineering. McGraw-Hill: New York.
- Hsu KL, Gupta HV, Gao X, Sorooshian S. 1999. Estimation of physical variables from multichannel remotely sensed imagery using a neural network: application to rainfall estimation. *Water Resources Research* **35**(5): 1606–1618.
- Hsu KL, Gupta HV, Gao X, Sorooshian S, Imam B. 2002. Self-organizing linear output map (SOLO): an artificial neural network suitable for hydrologic modeling and analysis. *Water Resources Research* **38**(12): 10–26.
- Hydrological Engineering Center. 1990. *HEC-1 flood hydrograph package. Program users manual*. US Army Corps of Engineers: California, USA.
- Imrie CE, Durucan S, Korre A. 2000. River flow prediction using artificial neural networks: generalization beyond the calibration range. *Journal of Hydrology* **233**: 138–153.
- Jain SK, Das A, Srivastava DK. 1999. Application of ANN for reservoir inflow prediction and operation. *Journal of Water Resources Planning and Management* **125**(5): 263–271.
- Kohonen T. 1982. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics* **43**: 59–69.
- Kohonen T. 1990. Improved versions of learning vector quantization. IEEE International Joint Conference on Neural Networks, vol. I, 545–550.
- Luchetta A, Manetti S. 2003. A real time hydrological forecasting system using a fuzzy clustering approach. *Computers and Geosciences* **29**(9): 1111–1117.
- Moradkhani H, Hsu KL, Gupta HV, Sorooshian S. 2004. Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. *Journal of Hydrology* **295**: 246–262.
- Nash JE, Sutcliffe JV. 1970. River flow forecasting through conceptual models, part I—a discussion of principles. *Journal of Hydrology* **10**: 282–290.
- Neelakantan TR, Pundarikanthan NV. 2000. Neural network-based simulation-optimization model for reservoir operation. *Journal of Water Resources Planning and Management* **126**(2): 57–64.
- Rajurkar MP, Kothiyari UC, Chaube UC. 2004. Modeling of the daily rainfall–runoff relationship with artificial neural network. *Journal of Hydrology* **85**: 96–113.
- Rumelhart DE, Hinton GE, Williams RJ. 1986. Learning internal representations by error propagation. *Parallel Distributed Processing* **1**: 318–362.
- Salas JD, Tabios GQ, Bartolini P. 1985. Approaches to multivariate modeling of water resources time series. *Water Resources Bulletin* **21**(4): 683–708.
- Seibert J. 2001. On the need for benchmarks in hydrological modeling. *Hydrological Processes* **15**: 1063–1064.