

A counterpropagation fuzzy-neural network modeling approach to real time streamflow prediction

Fi-John Chang^{a,*}, Yen-Chang Chen^b

^aDepartment of Agricultural Engineering, National Taiwan University, Roosevelt Road, Taipei 10770, Taiwan, ROC

^bDepartment of Agricultural Engineering, National Taiwan University, Taipei, Taiwan, ROC

Received 23 February 2000; revised 7 November 2000; accepted 6 February 2001

Abstract

A counterpropagation fuzzy-neural network (CFNN) is the fusion of a neural network and fuzzy arithmetic. It can automatically generate the rules used for clustering the input data. No parameter input is needed, because the parameters are systematically estimated by the approach of converging to an optimal solution. The advantages of the CFNN include the ability to cluster, learn, and construct, and the model presented herein is used to develop a hydrological model. The CFNN can automatically construct a rainfall-runoff model to forecast streamflow. The available streamflow and precipitation data of the upstream of the Da-cha River, in central Taiwan, is used to evaluate the CFNN rainfall-runoff model. A comparison of the results obtained by the CFNN model and ARMAX indicate the superiority and reliability of the CFNN rainfall-runoff model. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Artificial neural network; Counterpropagation neural network; Fuzzy system; Counterpropagation fuzzy-neural network; Rainfall-runoff; Streamflow

1. Introduction

Real time streamflow estimation is always a benchmark problem of hydrologists and water resource engineers, and has received a prominent focus for many decades. One of the difficult tasks of reservoir operation is inflow forecasting for preventing dam failure. Thus, accurate streamflow forecasting is extremely important for on-line reservoir operation. Some sophisticated hydrological models, such as deterministic catchment model (Kraijenhoff and Moll, 1986) and geomorphologic instantaneous unit hydrograph (GIUH) model (Rodriguez-Iturbe and

Valdes, 1979), for describing the rainfall-runoff process are usually very complicated. And a great deal of work, such as field surveying and parameter estimation, should be done before such models can be applied. Obviously, such models show a lack of practicality and are very difficult to use for real time streamflow forecasting in Taiwan, whose characteristics of watersheds are erodible soils, high mountains, steep slopes, subtropical climate, and heavy rainfall during typhoons.

Neural networks and fuzzy systems are created to simulate the nervous system and brain activity. The notion of fuzzy sets was first introduced by Zadeh (Zadeh, 1965) to represent vagueness in linguistics by a mathematical way. Recently, a rapid growth in the use of fuzzy sets in hydrological modeling, such as rainfall forecasting (Yu and Chen, 2000), groundwater

* Corresponding author. Tel.: +886-223639461; fax: +886-223635854.

E-mail address: changfj@ccms.ntu.edu.tw (F.-J. Chang).

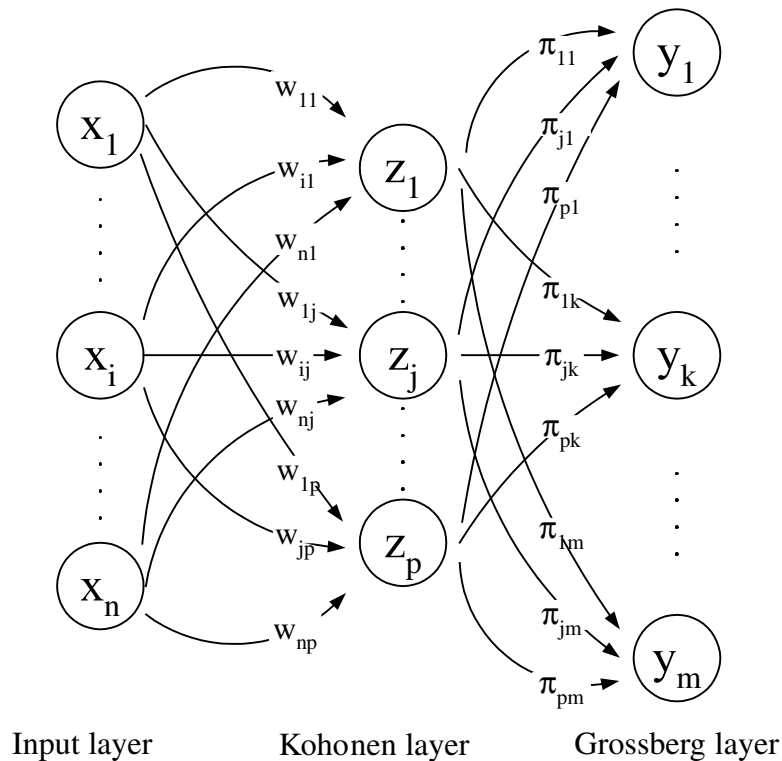


Fig. 1. Structure of the forward-only CPN.

simulation (Dou et al., 1999; Schulz and Huwe, 1997), and drought analysis (Pongracz et al., 1999; Pesti et al., 1996). The prototype of a neural network was first proposed by McCulloch and Pitts (McCulloch and Pitts, 1943). Hebb designed the first learning law for neural networks (Hebb, 1949). After that, a number of neural networks, such as the backpropagation neural network (Rumelhart et al., 1986) and fuzzy-neural network (Nie and Linkens, 1994), were developed to solve a wide variety of problems. Mathematically, neural networks are information processes systems to model the brain as a parallel computation. Neural network is composed of a large number of interconnected processing units (nodes), arranged in an input layer, an output layer, and one or more hidden layers. Each layer consists of several nodes. The input layer contacts with the outside environment, the hidden layer transforms the input or hidden layers to a hidden layer by a nonlinear function, and the output layer is the response of the network. Fig. 1 shows a 3-layer feedforward neural network with n input, p hidden,

and m output nodes. The principal advantage of neural networks is their adaptive nature, which learns from the historical data to automatically adjust parameters, in dealing with nonlinear problems. A physical model is not needed when neural networks and fuzzy systems are applied. They estimate functions from training data and then construct numerical estimators that can be applied to deal with streamflow prediction (Chang and Suen, 1997; Shamseldin, 1997; Chang and Hwang, 1999; Sajikumar and Thandaveswara, 1999, etc.), rainfall estimation (Kuligowski and Barros, 1998; Zhang et al., 1997), and groundwater modeling (Yang et al., 1997). In this study, a counter-propagation fuzzy-neural networks (CFNN), which couples neural network and fuzzy systems is used to develop a rainfall-runoff model for streamflow forecasting during typhoon. The CFNN is, furthermore, compared with a traditional stochastic forecasting model, autoregressive moving average with exogenous variable (ARMAX) model, for showing its impression of performances.

2. Forward-only counterpropagation network structure

The counterpropagation network (CPN), which was first introduced by Hecht-Nielsen (Hecht-Nielsen, 1987a,b, 1988), is of two types, full and forward-only. The CPN was developed to provide an efficient method for approximating a function $y = f(x)$. The full CPN works best only when inverse function f^{-1} exists. To avoid this effort, the forward-only CPN was designed to approximate $y = f(x)$ when f^{-1} is not necessarily needed. The forward-only CPN is adopted to develop the CFNN. The architecture of the forward-only CPN, shown in Fig. 1, shows the information flows in the feedforward direction only. The forward-only CPN consists of three layers: input, hidden (Kohonen), and Grossberg layers. The input layer with n nodes stores the input values, the Kohonen layer with p nodes clusters the input values with a similarity measurement, and the Grossberg layer with m nodes calculates the output by summing the weighted Kohonen layer outputs.

The forward-only CPN has a hybrid learning scheme. The learning of CPN can be split into two stages, unsupervised and supervised. During the learning processes, the weights (or parameters) of the neural network will be adjusted automatically. In supervised learning, each input pattern is associated with a specifically correct target pattern. On the other hand, unsupervised learning involves no critic to oversee the learning process. Unsupervised learning is used during the first stage for clustering the input vectors to separate distinct sets of input data. During the second stage of learning, the weight vectors between the Kohonen and Grossberg layers are adjusted by supervised learning to reduce the errors between the CPN outputs and the corresponding desired targets.

During the first stage, the distances between the input vector $x = (x_1, \dots, x_i, \dots, x_n)^T$ composed of n input nodes and all of the p Kohonen nodes with n dimensions are determined to compete for the winner. The winning node z_j has the weight vector $w_j = (w_{1j}, \dots, w_{ij}, \dots, w_{nj})^T$ closest to the input vector. The winner-take-all operation that permits only the hidden node being the most similar to the input vector to be active at a time is implemented here to train the weight vectors from input layer to Kohonen layer.

The winner's weight vector is updated according to (Rumelhart and Zipser, 1985)

$$\Delta w_{ij} = \begin{cases} \alpha(x_i - w_{ij}) & \text{winning node,} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where α is the learning rate subject to $\alpha > 0$, x_i is the i th node of input vector, and w_{ij} is the weight of the i th input node to the winning node j . The competitive signal, which is a binary variable assuming value 1 for the winner node presented and value 0 for the winner node absented, sent from the winning node to the Grossberg layer is 1, and the competitive signals sent by the other Kohonen nodes are 0s'.

After the weight vectors from the input layer to the Kohonen layer have been determined, the weight vectors between the Kohonen layer and the Grossberg layer are trained by

$$\Delta \pi_{jk} = Z_j \beta (y_k - \pi_{jk}) \quad (2)$$

in which Z_j is the competitive signal, β the learning rate, y_k the target, and π_{jk} is the weight from Kohonen node j to Grossberg node k . The output node k is given by

$$y_k^* = \sum_{j=1}^p \pi_{jk} Z_j \quad (3)$$

where y_k^* is the k th computed output.

The CPN can compress the n input data to p sets where in general $p \ll n$. It classifies the similar input vectors as a single output cluster to build a look-up table. Compared to other neural networks, the learning speed of CPN is extremely fast due to the simple network topology and the efficient learning algorithm. After the CPN is trained, all of the weights will be kept fixing. Only input data is needed for the model to operate when the CPN is used for predicting.

3. Fuzzy arithmetic

Fuzzy arithmetic provides a good approach to dealing with ambiguity and uncertainty. It is a structured numerical estimator and has successfully attracted a growing interest in applications. Fuzzy arithmetic combines rule base and fuzzy control to describe complex nonlinear processes. The rule base is the collection of rules. A rule contains two statements,

		H		
	O			
	Q	162	165	168
	80	50	70	85
	100	60	90	100
	125	75	110	120

Fig. 2. A rule base of reservoir operation.

>the premise and the conclusion. It is a logical implication: IF premise THEN conclusion. For example, Fig. 2 shows the rule base of a reservoir operation. *H*, *Q*, and *O* are gage height, inflow and outflow, respectively. The entry at the center of the rule base defines the reservoir operation rule: ‘IF *H* = 165 and *Q* = 100 THEN *O* = 90.’ The rule base of reservoir operation is composed of nine rules (3 different *H* times 3 different *Q*).

Classical logic theory can only represent one color in back and white (likes binary number 0 and 1). This logic does not accord well in gray (likes binary system containing no other number except 0 and 1). Fuzzy logic, which can be used to represent vague concepts lets elements be represented by degrees of membership. The degree of membership is a positive real number in

the interval [0,1]. A membership function assigns a degree of membership to an element and can be any shape. Using the centroid defuzzification method to map the crisp output from a space of fuzzy control actions, the fuzzy control output can be determined as follows:

$$\text{IF } x \text{ is } M^i \text{ THEN } y^i \tag{4}$$

$$Y = \frac{\sum_{i=1}^n M^i(x)y^i}{\sum_{i=1}^n M^i(x)} \tag{5}$$

in which *Y* is the fuzzy control output, $M^i(x)$ the degree of membership of the *i*th rule, *x* the input, y^i the output of the *i*th rule, and *n* is the number of rule. The fuzzy control is suggested to reduce error when a rule is chosen (Nie, 1989). Fig. 3 shows how the membership function works. It illustrates an example of the reservoir operation according to gage high only. When *H* = 167, the degrees of membership, $M^1(H)$, $M^2(H)$, and $M^3(H)$ are 0, 1/3, and 2/3, respectively. The fuzzy control output of reservoir operation (outflow) becomes $(0 \times 80 + (1/3)100 + (2/3)120)/(0 + (1/3) + (2/3)) = (340/3)$.

4. The counterpropagation fuzzy-neural network model

The CFNN is the fusion of CPN and fuzzy arithmetic. A similar type of network was first proposed by Nie (1993), who further investigated its adapting,

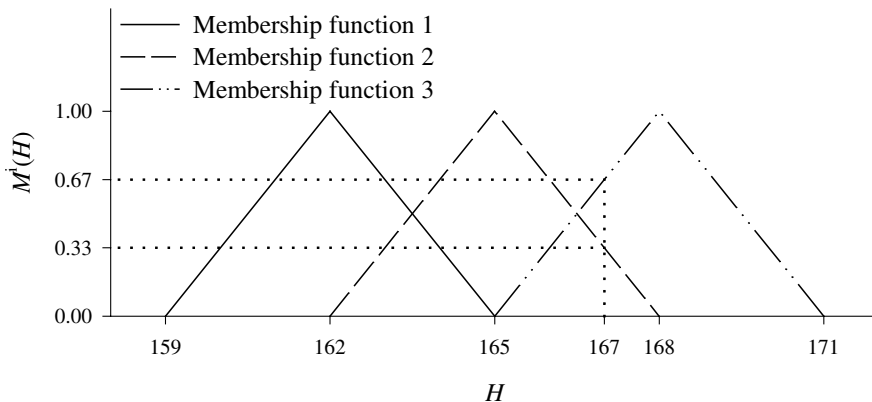


Fig. 3. Membership function graphs of reservoir operation.

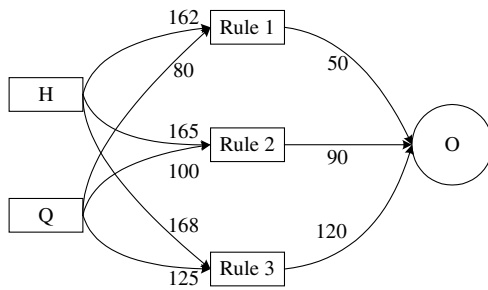


Fig. 4. A schematic diagram of 2-3-1 CPN.

learning, and reasoning capabilities and applied to a problem of multivariable control of blood pressure (Nie and Linkens, 1994). Two modifications have been made to aim at constructing a rainfall-runoff model. First, the type of membership function is changed from triangular to Gaussian function, so that the matching process could be more flexible and the weighted averaging for the fuzzy output would be more reasonable. The Gaussian function is a localized function with the property of $z(x) = \exp(-\|x - w\|^2/2\Delta^2) \rightarrow 0$ as $|x - w| \rightarrow \infty$. Its membership degree symmetrically and sharply decreases as the difference between the x and the node centroid w increases, but the membership degree is always larger than 0. Second, Δ the tolerant interval, using in the original model is a constant that would be gradually amplified as the input data cannot be clustered into any rule during the model's application stage. This improves the model performance, especially for those unusual cases.

The CFNN can be split into two stages, training and

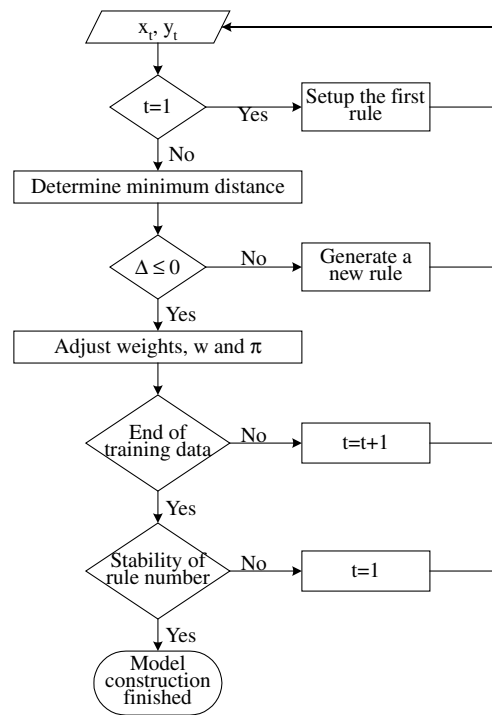


Fig. 6. CFNN learning model.

application. During the first stage, the CPN is used to build the rule base. The fuzzy arithmetic is introduced in the second stage to improve the performances. Fig. 1 also shows the architecture of the CFNN. Each Kohonen node represents a rule. The connections of input and Kohonen layers, w , represent the 'IF' statement of a rule. The connection between Kohonen and Grossberg layers is π that is, the 'THEN' part of a

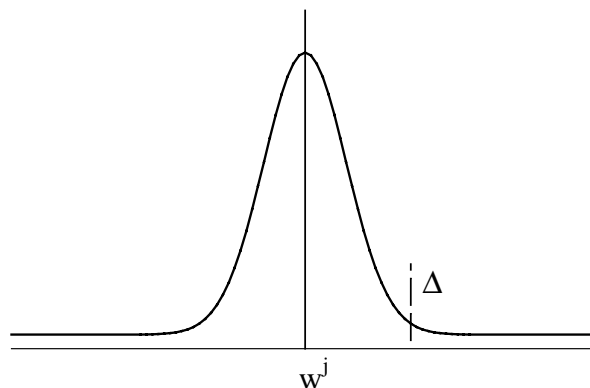


Fig. 5. Gaussian membership function.

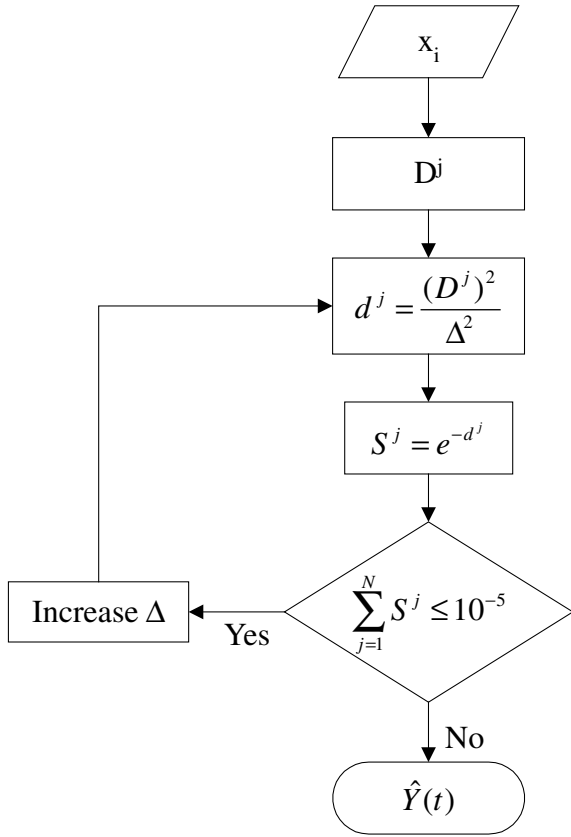


Fig. 7. CFNN estimating model.

rule. Thus the statement of each rule is defined as: IF x is w THEN y is π (Weigend et al., 1991). A 2-3-1 CPN is shown in Fig. 4. The simple network includes three rules. The inputs of the network are gage height and inflow; the only output of the network is outflow. Consider Rule 3, that is: ‘IF $H = 168$ and $Q = 125$ THEN $O = 120$.’

A Gaussian function shown in Fig. 5 is used to represent the degree of membership (Donald et al., 1996). The function is defined as

$$M^j(x) = \exp\left(\frac{-\sum_{i=1}^n (x_i - w_{ij})^2}{\Delta^2}\right) \quad (6)$$

where w_{ij} = mean (center of rule j); Δ^2 = variance (width square of rule j); and n = number of input variables (x_i).

Fig. 6 summarizes the training processes of the CFNN. The Δ is selected before the training of the model. The center of the first rule is the first input. After the first rule is built, the distances between input data and the center of each rule have to be determined as

$$D_j = \left[\sum_{i=1}^n (w_{ij} - x_i)^2 \right]^{(1/2)} \quad (7)$$

where D_j = the distance between input data and rule. The minimum distance between input data and rules is given by

$$D_{\min} = \min_{j=1,p} D_j \quad (8)$$

where D_{\min} = the minimum distance between input data and z_j . If D_{\min} is smaller than Δ , the center of rule, w_{ij} and π_{jk} has to be updated as

$$w_{ij}^{\text{new}} = w_{ij}^{\text{old}} + \alpha[x_i - w_{ij}^{\text{old}}] \quad (9)$$

$$\pi_{jk}^{\text{new}} = \pi_{jk}^{\text{old}} + \beta[y_k - \pi_{jk}^{\text{old}}] \quad (10)$$

where α and β are learning rates within the interval [0,1]. If $D_{\min} > \Delta$ a new rule will be created and $w_{ij}^{\text{new}} = x_i$, $\pi_{ij}^{\text{new}} = y_k$. The learning rates usually will be reduced after the entire training data is presented. The training processes will be iterated until the number of rule is stable.

The CFNN can be applied after training. The application procedure for the CFNN is shown in Fig. 7 and can be summarized as follows: Step 1: Initialize weights that are trained during the training stage. Step 2: Present input vector x . Step 3: Compute the distances between input x and rules. Eq. (7) is used to represent the distances. Step 4: Determine relative distance, d_j , which is given by

$$d_j = \left(\frac{D_j}{\Delta}\right)^2 \quad (11)$$

Step 5: Determine matching degree of rule j that is defined as follows:

$$S_j = \exp(-d_j) \quad (12)$$

Step 6: Find the rule that is closest to the input vector x . The input x can not be clustered, if the sum of matching degree is less than a selected value, i.e. 10^{-5} in our case. In such a condition, Δ will be

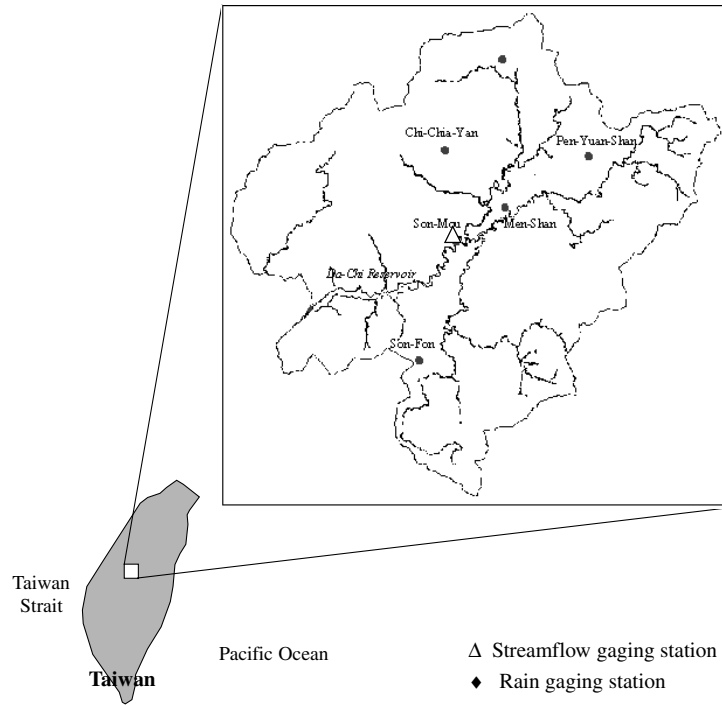


Fig. 8. Locations of study watershed and gage stations.

automatically expanded. Repeat Steps 4 to 6 until the input x can be clustered. Step 7: Compute activation. The fuzzy output is deduced by the weighted averaging, defined as

$$Y = \frac{\sum_{j=1}^p S_j \pi_j}{\sum_{j=1}^p S_j} \quad (13)$$

5. ARMAX model

The autoregressive moving average with exogenous variable (ARMAX) model (Yang et al., 1996) for one-hour ahead flood forecasting is used as a basis for comparison with CFNN. A general input–output system of ARMAX can be written as follows:

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (14)$$

where $y(t)$ is flood at time t ; $u(t)$ is exogenous rainfall input at time t ; $e(t)$ is white noise at time t ; $A(q)$, $B(q)$

and, $C(q)$ are parameters of autoregressive, exogenous, and moving average parts as the following form:

$$A(q) = 1 + a_1q^{-1} + \dots + a_nq^{-n} \quad (15)$$

$$B(q) = b_1 + b_2q^{-1} + \dots + b_mq^{-m+1} \quad (16)$$

$$C(q) = 1 + c_1q^{-1} + \dots + c_rq^{-r} \quad (17)$$

where q^{-1} is back-shift operator; n , m , and r are the order of autoregressive; exogenous; and moving average.

6. Application of the CFNN

The above methodology is applied to the upstream of the Da-Cha River for predicting real time streamflow. The Da-Cha River is located in central Taiwan with a total catchment size of 1236 km². The length of the Da-Cha River is about 140 km and the average channel slope is 1/39. It is the steepest channel in Taiwan. A series of hydraulic structures were constructed to generate power. Locations of the

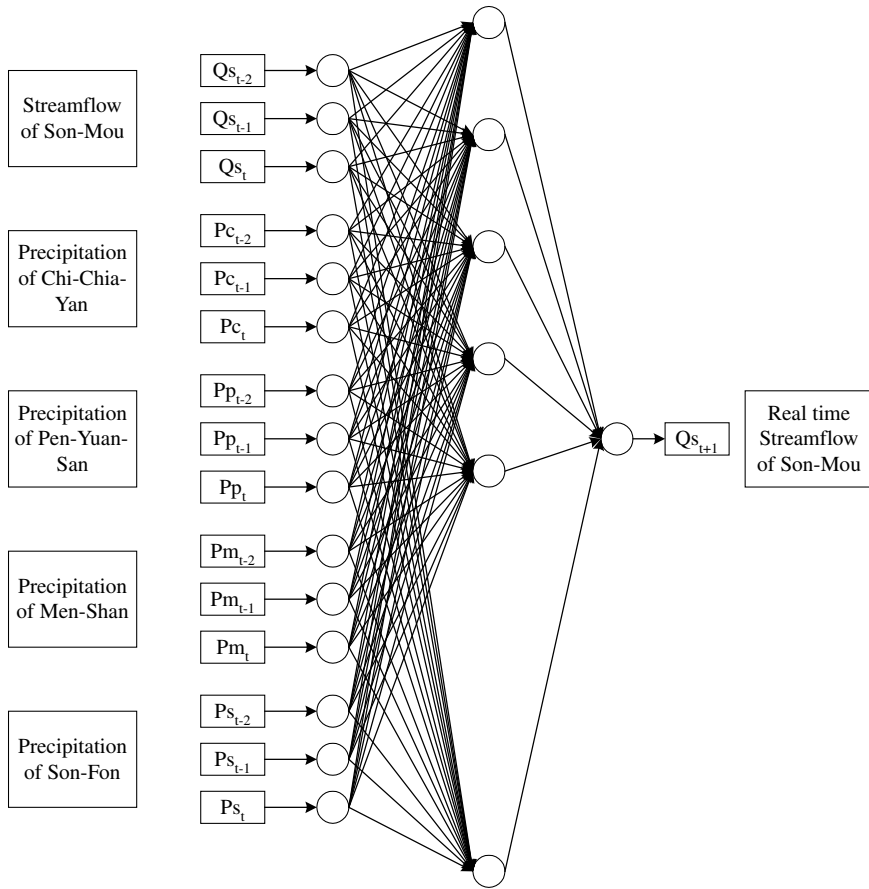


Fig. 9. CFNN structure for real time streamflow forecasting.

studied basin (area = 514 km²) and used gage stations are shown in Fig. 8. A station for the streamflow data is denoted by a triangle and the precipitation stations by circles. Son-Mou gage station was established to measure the inflow of the De-Chi Reservoir, the upmost and pivotal reservoir in the Da-Cha River. Accurate streamflow forecasting is extremely important for the operation of the De-Chi Reservoir. The streamflow (m³/s) and precipitation (mm/h) data used here are gathered from Taiwan Power Company.

A CFNN model is established to model real time streamflow prediction and is shown in Fig. 9. *Q_s* stands for the streamflow of Son-Mou, and *P_c*, *P_p*, *P_m*, and *P_s* for precipitation of Chi-Chia-Yan, Pen-Yuan-Shan, Men-Shan, and Son-Fon, respectively. The subscripts *t-2*, *t-1*, *t*, and *t + 1* represent two-hour-before, one-hour-before, present time, and one-

hour-ahead, respectively. The input layer has three nodes representing the streamflow at Son-Mou, and 12 nodes for the precipitation of the other four rain gage stations. Only one node in the Grossberg layer represents the one-hour-ahead forecasting streamflow of the Da-Cha River at Son-Mou station. The initial number of rules is zero and will be automatically generated to describe the complex hydrological processes. The performances of CFNN and ARMAX are evaluated and compared by the normalized root-mean-square error (NRMSE), mean absolute error (MAE), and relative mean absolute error (RMAE) as

$$NRMSE = \frac{1}{\sigma} \left[\frac{1}{T} \sum_{i=1}^T (Q_i - \hat{Q}_i)^2 \right]^{1/2} \quad (18)$$

Table 1
Number of training data set and number of rule generated for constructing models

Year	Cumulative number of training data sets	Cumulative number of rules
1983–1989	3183	1058
1990	4315	1497
1991	4894	1577
1992	5768	1899
1993	6624	2069
1994	7560	2366
1995	8590	2534

$$MAE = \frac{1}{T} \sum_{i=1}^T |Q_i - \hat{Q}_i| \quad (19)$$

$$RMAE = \frac{MAE}{\bar{Q}} \quad (20)$$

in which σ = standard deviation of measured streamflow; Q_i = measured streamflow; \hat{Q}_i = forecasted streamflow; \bar{Q} = mean of measured streamflow; and T = total number of measured streamflow. NRMSE indicates the closeness of the forecasted streamflow to observed streamflow. MAE and RMAE (Maidment, 1992) to represent the prediction error. This implies perfect matching when MAE = 0 or RMAE = 0.

The streamflow is forecasted under the conditions that any two of four hours and any three of four rain gage stations have precipitation records. The forecasting process is terminated when all of the rain gage stations continue having no precipitation record for 12 h. The initial learning rate α is 1/2. The α is

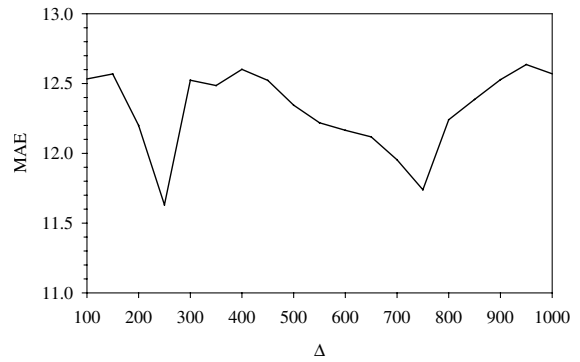


Fig. 10. Effect of Δ on MAE for the rainfall-runoff model.

replaced with a harmonic series, $1/(t + 1)$, until the number of rules stabilizes. In this study, β is not adjusted and is always 1/2. The available data sets from 1986 to 1989 are used for establishing the candidate models. Each set consists of 15 data. Nineteen values of Δ , which is updated from the initial value 100 with increments of 50 until 1000 are used to construct the model. The flood data between June and September 1990 (heavy rainfall season) is used to determine the best Δ for model construction Fig. 10 shown that $\Delta = 250$ has the minimum MAE. Thus $\Delta = 250$ is selected for constructing the rainfall-runoff models and it will be fixed for the forecasting.

After Δ is determined, the streamflow data from 1990 to 1996 is used for verifying the streamflow prediction of model. Owing to the new collected data, the model is updated every year. Table 1 presents the number of training data and the number of rules or hidden nodes generated for constructing models. The training data of every year includes the training data of this year and previous years. The new

Table 2
Summary of the result of the CFNN and ARMAX

Year	\bar{Q} (m ³ /s)	\hat{Q} (m ³ /s)		NRMSE		MAE (m ³ /s)		RMAE	
		CFNN	ARMAX	CFNN	ARMAX	CFNN	ARMAX	CFNN	ARMAX
1990	155.2	159.0	153.1	0.09	0.17	7.09	15.78	0.05	0.12
1991	23.0	24.2	24.6	0.11	0.28	1.38	3.27	0.06	0.17
1992	115.1	119.1	114.7	0.07	0.20	4.05	11.37	0.03	0.12
1993	48.1	49.9	49.0	0.09	0.20	1.46	5.27	0.03	0.12
1994	110.5	114.5	111.1	0.13	0.19	6.80	12.41	0.06	0.14
1995	34.5	35.6	35.8	0.14	0.31	1.40	4.34	0.04	0.14
1996	74.4	77.4	75.2	0.11	0.14	3.81	7.41	0.05	0.12

Table 3
Results of underestimated streamflow

Year	Events	Events underestimated		Percentage of events underestimated	
		CFNN	ARMAX	CFNN	ARMAX
1990	842	33	137	3.92	16.27
1991	370	21	40	5.68	10.82
1992	646	14	118	2.17	18.27
1993	590	14	70	2.37	11.86
1994	693	37	106	5.34	15.30
1995	725	16	98	2.21	13.52
1996	692	24	104	3.47	15.03

incoming data sometimes cannot be clustered into any rule; then a new rule is generated. Thus, the number of rules increases with the number of training data. A rule base with enough rules could accurately cluster the input data. The CFNN and ARMAX are applied for the prediction of streamflow. Table 2 gives the comparative results of two models. Apparently, the CFNN has better performance than the ARMAX in terms of smaller NRMSE, MAE, and RMAE. The results of the streamflow underestimated and overestimated are shown in Tables 3 and 4, respectively. The definitions of underestimation and overestimation are that the differences between the forecasted and observed streamflows are more than 10 and 5%, respectively. These results show the CFNN can predict streamflow more accurate than the ARMAX and reveal that the CFNN can be used to forecast one-hour-ahead streamflow. To provide a impression of

Table 4
Results of overestimated streamflow

Year	Events	Events overestimated		Percentage of events overestimated	
		CFNN	ARMAX	CFNN	ARMAX
1990	842	34	92	4.04	10.93
1991	370	18	52	4.86	14.05
1992	646	24	61	3.72	9.44
1993	590	15	85	2.54	14.41
1994	693	29	96	4.18	13.85
1995	725	21	87	2.90	12.00
1996	692	30	92	4.34	13.36

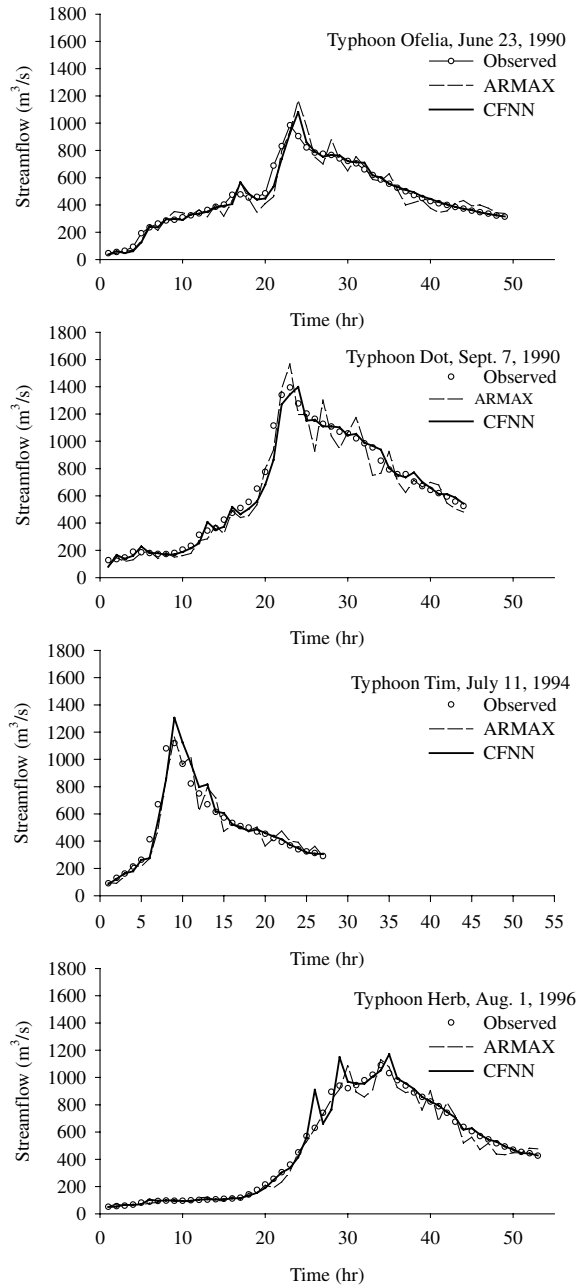


Fig. 11. Comparison of CFNN and ARMAX using four typhoon events.

the accuracy of streamflow prediction using the CFNN, four typhoon events including the largest event have been extracted to show the performance of the CFNN and ARMAX. Fig. 11 shows the

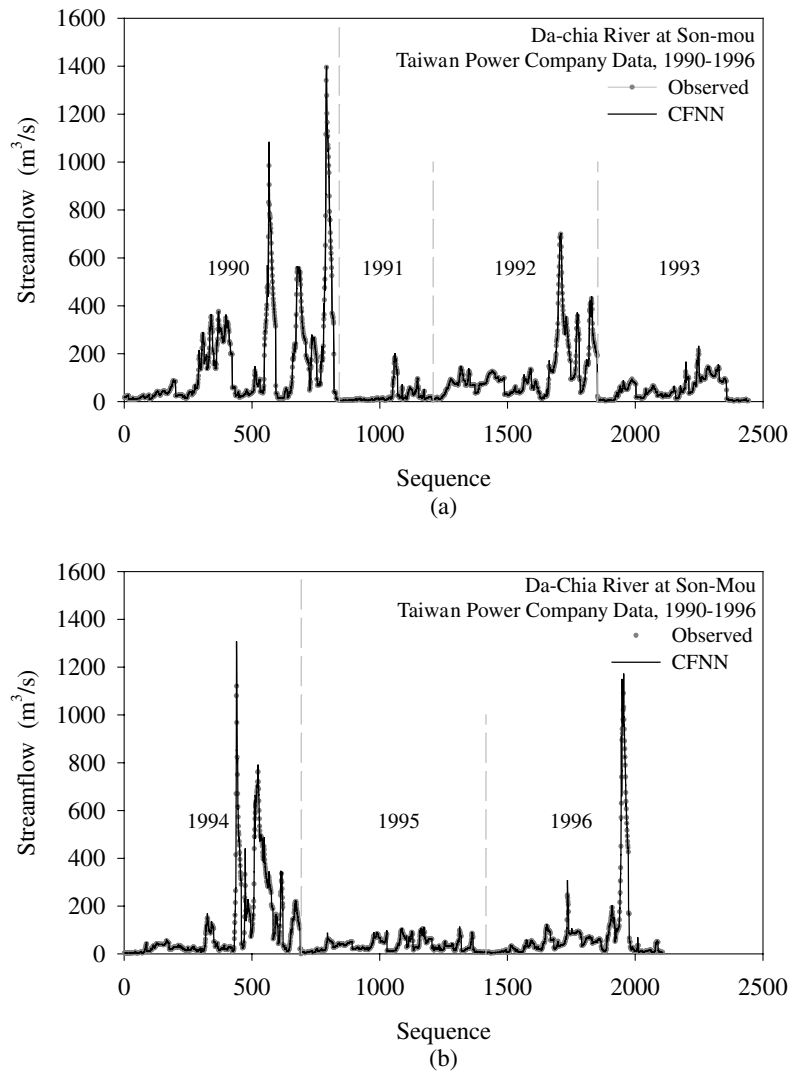


Fig. 12. Accuracy and reliability of forecasted streamflow.

comparison of CFNN and ARMAX during four typhoon events. Fig. 12 shows the accuracy of forecasted streamflow by the CFNN model. All the points in the figure nicely fall around the line of agreement. These results indicate the accuracy and reliability of the model presented herein.

7. Conclusions

Neural networks have a complex connection structure and simple computing elements that can solve

problems with natural mechanisms. The fuzzy arithmetic combines a rule base with fuzzy logic to form a structure of fuzzy if-then rules. This study proposes the CFNN, which is the integration of a neural network and fuzzy arithmetic. The CFNN presents three advantages: the ability to learn, construct, and cluster. The membership function of original CFNN is triangular, which, in this study, is replaced by Gaussian function. The concept of automatically increasing Δ for clustering input data in the application stage is introduced. They improve the model performance, especially the unusual events.

The CFNN rainfall-runoff model is successfully applied to forecast one-hour-ahead streamflow of the Da-Cha River at Son-Mou. Compared to sophisticated hydrological models, such as GIUH, the principal advantages of the proposed method is that the CFNN can automatically construct a rainfall-runoff model and estimate the needed parameters by an approach converging to an optimal solution. The comparative results obtained by the CFNN and ARMAX provide evidences that the CFNN can offer a higher degree of reliability and accuracy than ARMAX in streamflow forecasting.

Acknowledgements

This paper is based on partial work supported by Taiwan Power Company and National Science Council, ROC (Grant no. NSC 89-2313-13-002-041). In addition, the authors are indebted to the reviewers for their valuable comments and suggestions.

References

- Chang, F.-J., Hwang, Y.-Y., 1999. A self-organization algorithm for real-time flood forecast. *Hydrologic Processes* 13, 123–138.
- Chang, F.-J., Suen, J.-P., 1997. A study of the artificial neural network for rainfall-runoff process. *Journal of Chinese Agricultural Engineering* 43 (1), 9–25 (in Chinese).
- Donald, K., Wedding, I., Cios, K.J., 1996. Time series forecasting by combining RBF networks, certainty factors, and the Box–Jenkins model. *Neurocomputing* 10, 149–168.
- Dou, C., Woldt, W., Gogardi, I., 1999. Fuzzy rule-based approach to describe solute transport in the unsaturated zone. *Journal of Hydrology* 220, 74–85.
- Hebb, D.O., 1949. *The Organization of Behavior*. Wiley, New York.
- Hecht-Nielsen, R., 1987a. Counterpropagation networks. *Applied Optics* 26, 4979–4984.
- Hecht-Nielsen, R., 1987b. Nearest matched filter classification of spatiotemporal patterns. *Applied Optics* 26 (10), 892–1899.
- Hecht-Nielsen, R., 1988. Applications of counterpropagation networks. *Neural Networks* 1, 131–139.
- Kraijenhoff, D.A., Moll, J.R., 1986. *River Flow Modelling and Forecasting*. Dordrecht, Netherlands.
- Kuligowski, R.J., Barros, A.P., 1998. Experiments in short-term precipitation forecasting using artificial neural networks. *Monthly Weather Review* 126, 470–482.
- Maidment, D.R., 1992. *Handbook of Hydrology*. McGraw-Hill, New York (pp. 262–263).
- McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133.
- Nie, J., 1989. A-class of new fuzzy control algorithms. *Proceedings of IEEE International Conference on Control and Applications, Israel*.
- Nie, J., 1993. Constructing rule-bases for multivariable fuzzy control by self-learning — Part I: system structure and learning algorithms. *International Journal of System Science* 24, 111–127.
- Nie, J., Linkens, D.A., 1994. Fast self-learning multivariable fuzzy controllers constructed from a CPN network. *International Journal of Control* 6, 369–393.
- Pesti, G., Shrestha, B., Duckstein, L., Gogardi, I., 1996. A fuzzy rule-based approach to drought assessment. *Water Resources Research* 32 (6), 1741–1747.
- Pongracz, R., Bogardi, I., Duckstein, L., 1999. Application of fuzzy rule-based modeling technique to regional drought. *Journal of Hydrology* 224, 100–114.
- Rodriguez-Iturbe, I., Valdes, J.B., 1979. The geomorphologic structure of hydrologic response. *Water Resources Research* 15 (6), 1409–1420.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representation by error propagation. *Parallel Distributed Processing* 1, 318–362.
- Rumelhart, D.E., Zipser, D., 1985. Feature discovery by competitive learning. *Cognitive Science* 9, 75–112.
- Sajikumar, N., Thandaveswara, B.S., 1999. A non-linear rainfall-runoff model using an artificial neural network. *Journal of Hydrology* 216, 32–55.
- Schulz, K., Huwe, B., 1997. Water flow modeling in the unsaturated zone with imprecise parameters using a fuzzy approach. *Journal of Hydrology* 201, 211–229.
- Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modelling. *Journal of Hydrology* 199, 272–294.
- Weigend, A.S., Rumelhart, D.E., Huberman, B.A., 1991. Generalization by weight-elimination with application to forecasting. *Advances in Neural Information Processing System* 3, 875–882.
- Yang, C.C., Prasher, S.O., Lacroix, R., Sreekanth, S., Patni, N.K., Masse, L., 1997. Artificial neural network model for subsurface-drained farmland. *Journal of Irrigation and Drainage Engineering, ASCE* 123, 285–292.
- Yang, H.-Z., Huang, C.-M., Huang, C.-L., 1996. Identification of ARMAX model for short term load forecasting: an evolutionary programming approach. *IEEE Transactions on Power Systems* 11 (1), 403–408.
- Yu, P.-S., Chen, C.-J., 2000. Application of gray and fuzzy methods for rainfall forecasting. *Journal of Hydrological Engineering, ASCE* 5 (4), 339–345.
- Zadeh, L.A., 1965. Fuzzy sets. *Information Contributions* 8, 338–353.
- Zhang, M., Fulcher, J., Scofield, R.A., 1997. Rainfall estimation using artificial neural network group. *Neurocomputing* 16, 97–115.