Journal of Hydroscience and Hydraulic Engineering
Vol. 19, No.2 November, 2001, 47-55

ENHANCED EFFICIENCY ON THE PARAMETERS ESTIMATION OF MUSKINGUM MODEL USING ARTIFICIAL NEURAL NETWORK

By

C.C. Yang

PhD student, Department of Civil Engineering,
National Chiao Tung University, Hsinchu, Taiwan

and

L.C. Chang

Assoc. Prof., Department of Civil Engineering,
National Chiao Tung University, Hsinchu, Taiwan

SYNOPSIS

The Muskingum Model is the most widely used method on flood routing for hydraulic engineering. However, it uses a subjective means of estimating the physical parameters, and possibly lowers the efficiency of flood discharge calculations. This study presents a novel scheme capable of reducing the complexity associated with the Muskingum model in estimating the parameters by applying an Artificial Neural Network (ANN). The input and output neurons of ANN are designed according to the Muskingum formula. After completing the learning phase of the ANN model, a sensitivity analysis is performed to obtain the required parameters for Muskingum model on flood routing. A case study is also presented to demonstrate the new scheme's effectiveness. Simulation results indicate that the new scheme can reduce the complexity of the Muskingum model when estimating the parameters. Consequently, the proposed scheme can easily estimate the required parameters for the Muskingum model on flood routing in an objective manner.

INTRODUCTION

Among the many models used for flood routing, the Muskingum Method is the most widely used because of its simplicity. The Muskingum flood routing model was developed by the U.S. Corps of Engineers for the Muskingum Conservancy District Flood-Control Project over six decades ago. The following continuity and storage equations are the most commonly used forms of the Muskingum model (1):

$$\frac{dS_t}{dt} = I_t - O_t \tag{1}$$

$$S_t = K\left[X I_t + (1-X) O_t\right] \tag{2}$$

where St, It and Ot denote the simultaneous amounts of storage, inflow, outflow, respectively, at time t; K is storage-time constant for the river reach, which has a value reasonably close the flow travel time through the river reach; and X is a weighting factor usually varying between 0 and 0.5 for reservoir storage. Eq. (1) to (2) may be induced as

$$O_{t+\Delta t} = C_0 I_{t+\Delta t} + C_1 I_t + C_2 O_t \tag{3}$$

$$C_0 = \frac{-KX + 0.5\Delta t}{K(1-X) + 0.5\Delta t} \tag{4}$$

$$C_1 = \frac{KX + 0.5\Delta t}{K(1-X) + 0.5\Delta t} \tag{5}$$

$$C_2 = \frac{K(1-X) - 0.5\Delta t}{K(1-X) + 0.5\Delta t} \tag{6}$$

constrain : $C_0+C_1+C_2=1$ (7)

According to eq. (4) to (7), if eq. (3) can be used, three parameters (C_0, C_1 and C_2) have to conform. In practice, although $\triangle t$ represents the time step and is the given value, K,X are unknown parameters. The conventional procedure for determining the values of K, X is trial and error method. By assuming a value of X, the values of $[X I_t+(1-X)O_t]$ are computed and plotted against the corresponding value of S. The correct value of X corresponds to the plot for which the width of the loop is minimum or the plot approximates a straight line.

Although this trial and error method has been used for several decades, it is time-consuming and is prone to subjective interpretation (2). To improve this method, Mohan (1) proposed the objective approach of genetic algorithm to estimate the parameters of Muskingum routing models. Results indicate that the genetic algorithm approach is much more efficient in estimating the parameters of Muskingum routing models than the conventional estimation methods owing to its ability to prevent the subjective and computational time associated with the conventional estimation methods. On the other hand, Chen C.S., Wang N.B. (3) proposed a modified Muskingum flood routing model to describe the real flood characteristics more effectively. Their model was developed based on the law of conservation of mass so that the effects of the upstream tributaries and the distance from each gauging station of tributary to the downstream control point in a basin could be included. A genetic algorithm was also employed to obtain the parameters in the process.

The discussion above suggests that the genetic algorithm can estimate the parameters of the Muskingum flood routing model. Moreover, similar to the genetic algorithm, Artificial Neural Network (ANN) is a new computing architecture in the area of Artificial Intelligence (AI) and, therefore, may be another good scheme to estimate the parameters of Muskingum flood routing model; In this study, we estimate three parameters (C_0, C_1, C_2) which symbolize the interactive relation between input variables ($I_{t+\Delta t}$, I_t, O_t) and output variable($O_{t+\Delta t}$) according to eq.(3). Based on above meaning a novel scheme (ANN and sensitivity analysis) is proposed. ANN can accurately represent an internally complex relation between input and output variables. In addition, a sensitivity study is applied to the neural network model to extract information from the key input variables that might strongly affect the output variables. ANN and sensitivity analysis were performed to obtain information needed as follows: Zhichao G. and Robert E (4) undertook a nuclear power plant performance study by using the neural network and sensitivity analysis. The thermal performance data obtained from TVA nuclear power plant indicated that the plant probably lost some Megawatts of electric power due to the variation of the heat rate. Analyzing the raw data recorded weekly during the plant operations was difficult due to the fact that a nuclear power plant is an extremely complex system with thousands of parameters. The neural network was set up to function as the internal thermodynamic model of the plant so as to predict the heat rate. Then, a sensitivity analysis was performed on the neural network model to extract information from the key parameters that might strongly affect thermal performance. Another illustration involved the application of ANN to assess voltage stability. A.A. El-Keib and X.ma (5) proposed a multi-layer feed-forward artificial neural network with error back-propagation learning to calculate the voltage stability margin (VSM). Based on the energy method, a direct mapping relation between system loading conditions and VSMs was set up via the ANN. A systematic method for selecting the ANN's input variable was also developed by using a sensitivity analysis. This analysis was performed to ascertain the system's responsive behavior to load changes and to determine more appropriate ANN architectures that could be designed to assess voltage stability.

In light of the above discussion, ANN and sensitivity were applied with success in above two regions but they were never adopted in the parameter estimation of Muskingum method. This investigation, therefore, presents a novel scheme based on ANN and sensitivity analysis to estimate the parameters (C_0, C_1, C_2) of Muskingum linear function. The input and output neurons of Artificial Neural Network are designed according to the Muskingum formula $O_{t+\Delta t}=C_0 I_{t+\Delta t}+C_1 I_t+C_2 O_t$, input neurons are $I_{t+\Delta t}$, I_t, O_t and output neuron is $O_{t+\Delta t}$. After completing the learning phase of the ANN model, the sensitivity analysis of the ANN model is implemented to extract information from practical data to reveal the significance of input neurons. The values of parameters (C_0, C_1, C_2) are then obtained from the significance of input neurons with the limitation of $C_0+C_1+C_2=1$. Finally, the proposed method is compared with trial and error method from C_0, C_1, C_2.

A case study presented herein demonstrates the effectiveness of the scheme which is proposed. Simulation results indicate that this new scheme can reduce the complexity on the parameter estimation of Muskingum model by using an objective means of estimating the parameters. Consequently, the proposed scheme can easily estimate the required parameters for Muskingum model on flood routing.

METHODOLOGY

Artificial Neural Network

ANN consists of many artificial neurons (commonly referred to as processing units or nodes). The output signal is determined by the algebraic sum of the weighted inputs, i.e.,

$$Y_j = f(\sum_i W_{ij} X_i + \theta_j) \quad (8)$$

where, Y_j : output signal of the node j ;

f : transfer function ;

W_{ij} : weights between the node i and j ;

X_i : input signal of the input node i ;

θ_j : bias value of the output node j ;

ANN has two phases of neural processing: (I) Learning process, by which all knowledge in ANN is encoded in the interconnection weights which are determined through learning process from a set of examples, and (II) Recalling process, in which the recalling process attempts to retrieve the information, based on the weights obtained from learning process, and to predict the output data of new examples. In addition, the learning process can be categorized into two types: (1) Supervised learning (also referred to as learning with a teacher). Supervised learning gradually adjusts the weights of the ANN, thereby minimizing the error signal between the known answers and the responses of ANN. (2) Unsupervised learning, which does not rely on an external teacher. Without a known answer, this approach is expected to identify features, categories or class memberships in the input data and associate them with the corresponding outputs.

The multi-layer neural network, is a widely used neural network, and contains three layers (Fig 1): input (receives the input signals from the external world), hidden (represents the relation between input layer and output layer), and output (releases the output signals to the external world) layers. Multi-layer neural network with error back-propagation training gradually adjusts its weights, thereby minimizing the error between the known answers and actual responses (6)(7).

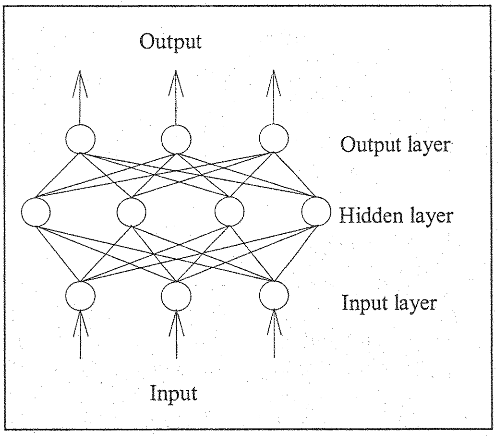

Fig. 1 The structure of a multi-layer neural network

Sensitivity

A successfully trained neural network works essentially as a mapping function, which maps a set of input vectors $\vec{x}$ in n-dimensional space to a set of output vectors y in m-dimensional space. It can be expressed as: $\vec{y} = f(\vec{x})$; where $\vec{x} = (x_1, x_2, ..., x_n)$ and $\vec{y} = (y_1, y_2, ..., y_m)$. The partial derivative, $\partial y_k / \partial x_i$, is the rate change in y_k with respect to a change in x_i. Therefore, $\partial y_k / \partial x_i$ can be used to measure the importance among the input variables, x_i, i=1,2,..n (4).

The multi-layer neural network is a widely used neural network at present. After the learning phase of error back-propagation training is finished, the output O_k can be written as

$$O_k = f(Net_k), \qquad Net_k = \sum_{j_n} o_{j_n} W_{j_n,k} + \theta_k \tag{9}$$

$$O_{j_n} = f(Net_{j_n}), \qquad Net_{j_n} = \sum_{j_{n-1}} O_{j_{n-1}} W_{j_{n-1},j_n} + \theta_{j_n}$$

$$\vdots \qquad\qquad \vdots$$

$$O_{j_1} = f(Net_{j_1}), \qquad Net_{j_1} = \sum_{i} O_i W_{i,j_1} + \theta_{j_1} \tag{10}$$

where, f is transfer function; W denotes a connected weight; θ represents bias value; i , j and k denote input unit, hidden unit and output unit; O_{jn}, O_{jn-1}, O_{jn-2}, ... O_{j1} denote the hidden units in the n, n-1, n-2, ...1 hidden layers. The work of the partial derivative is proceeded by using the following calculation to extract information from the key input variables that might strongly affect the output variables (8).

$$\frac{\partial O_k}{\partial O_i} = \sum_{j_n} \sum_{j_{n-1}} \dots \sum_{j_1} W_{j_n,k} G(Net_k) W_{j_{n-1},j_n} G(Net_{j_n}) W_{j_{n-2},j_{n-1}} G(Net_{j_{n-1}}) \\ W_{j_{n-3},j_{n-2}} G(Net_{j_{n-2}}) \dots\dots W_{i,j_1} G(Net_{j_1}) \tag{11}$$

$$G(Net_k) = f'(Net_k) \tag{12}$$

$$G(Net_{j_n}) = f'(Net_{j_n})$$

$$\vdots$$

$$G(Net_{j_1}) = f'(Net_{j_1}) \tag{13}$$

APPLICATION

Data Set

This study also investigates the application of ANN and sensitivity analysis in estimating the parameters of the Muskingum model, and by using a typical problem as an example. The data set with twenty-two time steps from Wilson (1974) (2) is considered for illustration. There are two reasons for selecting this example. First, this example has been studied previously by Gill (9), Tung (10) and Yoon and Padmanabhan (2) for testing different parameter estimation methodologies; and secondly the volume of inflow hydrograph is equal to the volume of outflow hydrograph.

Table. 1 Data set from Wilson (1974)

time (hour)	inflow I (cms[m³/sec])	outflow O (cms[m³/sec])
0	22	22
6	23	21
12	35	21
18	71	26
24	103	34

30	111	44
36	109	55
42	100	66
48	86	75
54	71	82
60	59	85
66	47	84
72	39	80
78	32	73
84	28	64
90	24	54
96	22	44
102	21	36
108	20	30
114	19	25
120	19	22
126	18	19

Trial and Error Method

The conventional method of trial and error requires accumulated storage to be plotted against weighted flow. The value of X that gives a straight line instead of a loop is considered to be the best X-value, and the storage constant K is estimated from the reciprocal of the slope.

Consequently, the trial and error method plots the accumulated storage versus weighted flow with an overlaid fitted line for a user-specified value of X. The user then chooses the best fitting line by visual judgment, and the program calculates a storage constant K and routed outflows. The trial and error method is therefore subjective (2).

In this example, the optimal estimated values of parameters based on $\triangle t$ of six hours (0.25 days) according to the trial and error method are K is 1.5 days and X is 0.25. Figure 2 shows the relationship between accumulated storage and simultaneous weighted flow when X is 0.25. The values of parameters (C_0, C_1, C_2) can be calculated according to eqs. (4) to (6) after K and X are given, and then the discharging outflows are obtained from eq. (3) by using the above parameters.

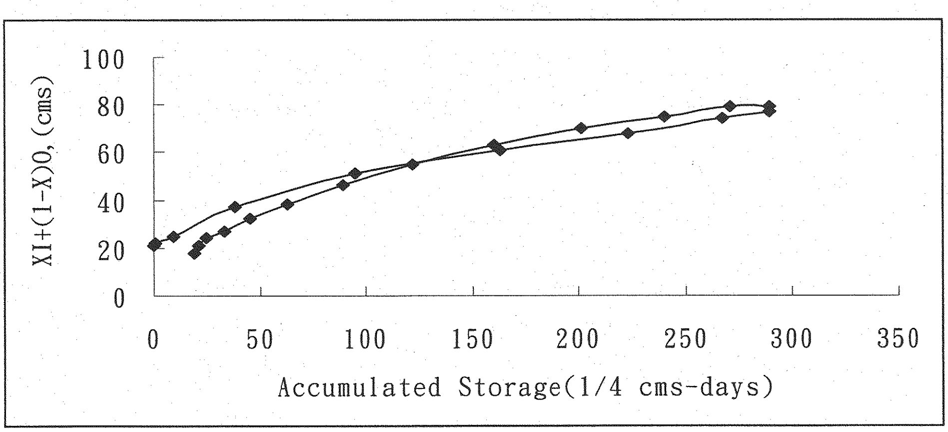

Fig. 2 Accumulated storage versus simultaneous weighted flow when X is 0.25

ANN Model and Sensitivity Analysis

In this study, the three parameters (C_0, C_1, C_2) symbolize the interactive relationship between input variables ($I_{t+\Delta t}$, I_t, O_t) and output variable ($O_{t+\Delta t}$) according to eq. (3), $O_{t+\Delta t}=C_0 I_{t+\Delta t}+C_1 I_t+C_2 O_t$. When the partial derivative of eq.(3), $\frac{\partial O_{t+\Delta t}}{\partial I_{t+\Delta t}}$ 、 $\frac{\partial O_{t+\Delta t}}{\partial I_t}$ 、 $\frac{\partial O_{t+\Delta t}}{\partial O_t}$, is obtained, the values of C_0 、 C_1 、 C_2 are clearly determined since $\frac{\partial O_{t+\Delta t}}{\partial I_{t+\Delta t}}$ = C0, $\frac{\partial O_{t+\Delta t}}{\partial I_t}$ = C1, $\frac{\partial O_{t+\Delta t}}{\partial O_t}$ = C2. Therefore, a novel scheme (ANN and sensitivity analysis) is proposed. The input and output neurons of ANN are designed according to the Muskingum formula $O_{t+\Delta t}=C_0 I_{t+\Delta t}+C_1 I_t+C_2 O_t$; input neurons are represented by $I_{t+\Delta t}$, I_t, O_t and the output neuron is represented by $O_{t+\Delta t}$. On the other hand, the number of hidden layers is set to one and the transfer function is adopted as the sigmoid function. Our stopping criterion is that sum squared errors is less than 0.05 and learning times set is 7000. With ultimate actual accuracy is 0.0288, the learning training process helps us obtain representative weights and biases. The consequent structure of ANN is depicted in Fig 3. After the learning phase of the ANN model is completed, the sensitivity analysis of the ANN model is implemented to determine the average significance of input neurons ($\sum_{t=1}^{n}\frac{\partial O_{t+\Delta t}}{\partial I_{t+\Delta t}}/n$ =A, $\sum_{t=1}^{n}\frac{\partial O_{t+\Delta t}}{\partial I_{t}}/n$ =B, $\sum_{t=1}^{n}\frac{\partial O_{t+\Delta t}}{\partial O_t}/n$ =C; n denotes the number of time steps). Then the ratio of the average significance of each input neurons to the average significance of total input neurons is calculated such that it satisfies the constraint, C0+C1+C2=1. Each ratio gives the value of a parameter (C0 = $\frac{A}{A+B+C}$ 、C1 = $\frac{B}{A+B+C}$ 、C2 = $\frac{C}{A+B+C}$). Finally, the novel method is compared with the trial and error method, and the simulated results of these two methods for estimating parameters are presented in Fig. 4 and Table 2.

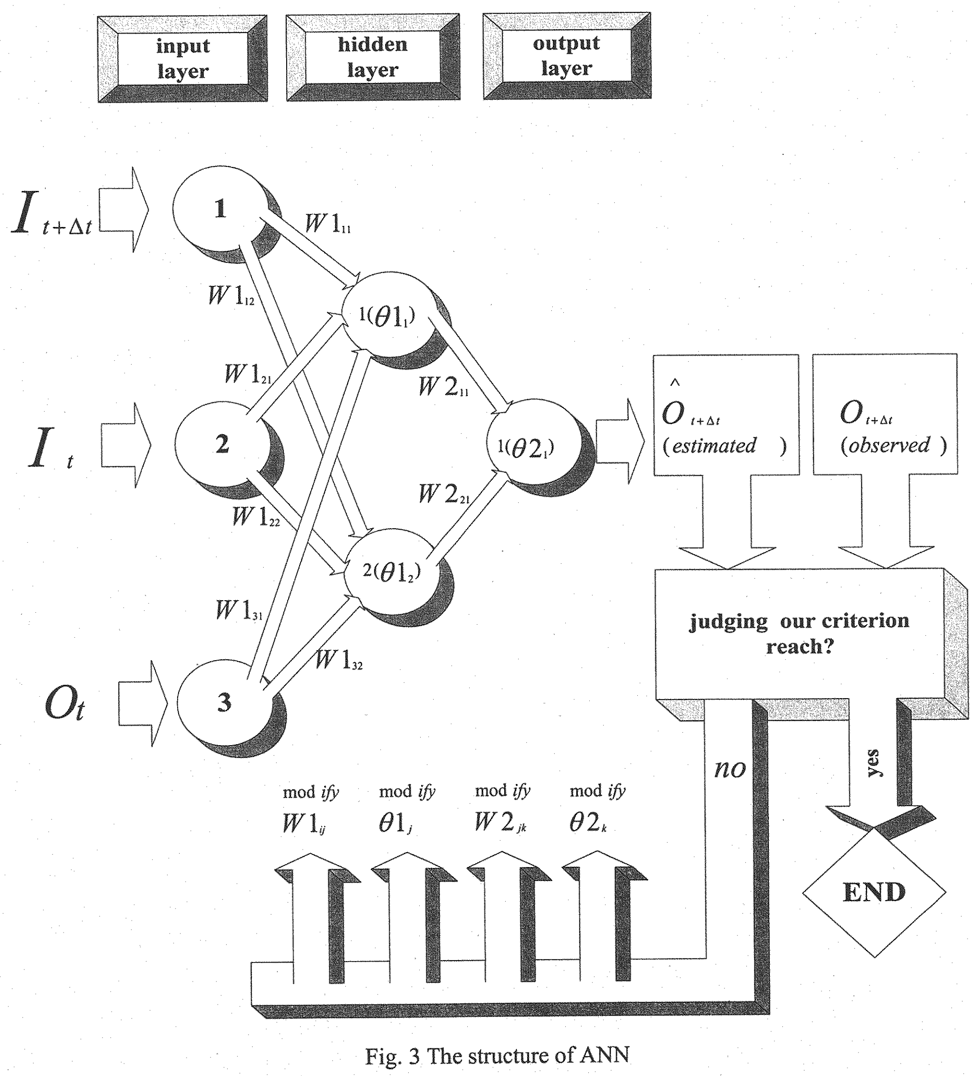

Fig. 3 The structure of ANN

Criteria for Comparing the Models

Three indicators are used to evaluate the accuracy of the proposed model (6):

(1) Coefficient of efficiency, CE

$$CE = 1 - \frac{\sum (Q_{obs} - Q_{est})^2}{\sum (Q_{obs} - \overline{Q}_{obs})^2} \quad (14)$$

Where Q_{est} denotes the estimating flood discharge at each time step (cms); Q_{obs} represents the observed flood discharge at each time step (cms); and $\overline{Q}_{obs}$ is the mean value of the observed flood discharge (cms). The closer value of CE to 1, the more the accurate model.

(2) Error of peak discharge, EQ_P

$$EQ_p = \frac{Q_{pest} - Q_{pobs}}{Q_{pobs}} \quad (15)$$

Where Q_{pobs} and Q_{pest} are the observed and estimated peak discharges of the flood, respectively. A lower absolute value of EQp implies a more accurate model.

(3) Error of time to peak, *ETp*

$$ET_p = T_{pest} - T_{pobs} \quad (16)$$

Where T_{pest} and T_{pobs} denote the estimated and observed times to peak discharge, respectively. A smaller value of ETp implies a more accurate prediction of occurrence of peak discharge.

Table. 2 The simulated results of two methods

method	Co	C1	C2	CE	EQp	ETp 【hr】
ANN and sensitivity analysis	-0.203	0.381	0.822	0.977	-0.026	1
trial and error	-0.2	0.4	0.8	0.980	-0.033	1

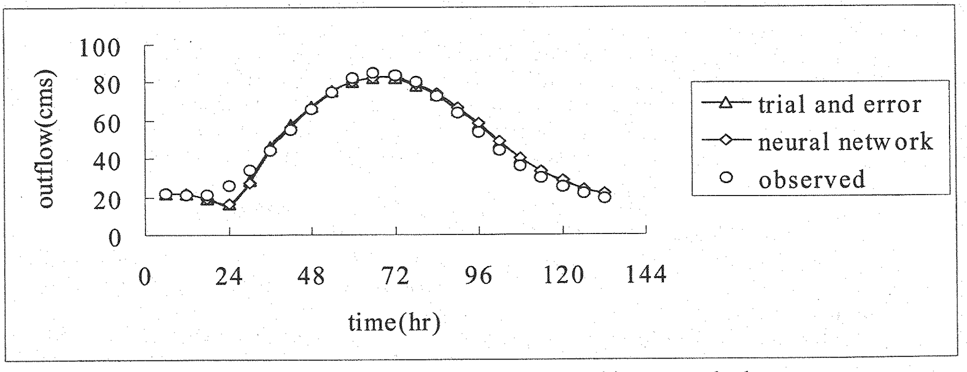

Fig. 4 The outflow estimation of two Muskingum methods

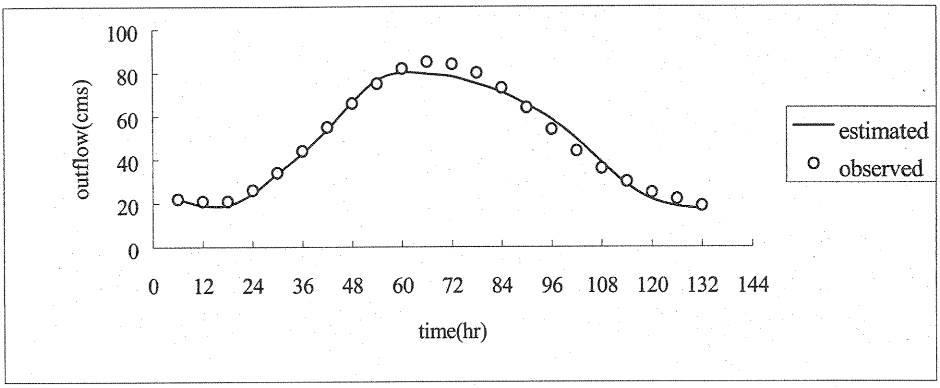

Fig. 5 The outflow estimation by ANN instead of Muskingum model

CONCLUSION

To increase the efficiency of estimating parameters of the Muskingum model, this investigation presents a novel scheme based on ANN. A sensitivity analysis is also performed to estimate the parameters (C_0, C_1, C_2) of the Muskingum linear function. The input and output neurons of ANN are designed according to the Muskingum formula. After completing the learning phase of the ANN model, the sensitivity analysis of the ANN model is performed to obtain the required parameters for Muskingum model on flood routing. The proposed approach is compared with the trial and error method using different criteria for the selected data. In terms of estimating the parameter values (C_0, C_1, C_2), both approaches yield similar results. With respect to the accuracy of flood routing as assessed by the three indicators (CE, EQp, ETp), the novel approach performs better or is at least comparable to the trial and error method. Although these methods estimate accurately the parameters, the trial and error method not only uses a subjective means of estimation owing to the requirement of an initial hypothesis of parameters but is also time consuming due to the lack of an objective selection criteria for the proper values of parameters. Therefore, the proposed scheme can reduce the complexity associated with estimating the parameters of the Muskingum model by using an objective rather than a subjective means of doing so. Therefore, the proposed scheme can easily estimate the required parameters for the Muskingum model on flood routing.

Moreover, this study attempts to calculate the outflow discharge of related data set by ANN instead of Muskingum model. Fig5 displays the well performance (CE = 0.982, EQp = -0.059, Etp = 1) of employing ANN. This finding provides evidence that ANN is also appropriate for application on the outflow discharge estimation of flood routing when physical parameters are unknown. Although ANN has the advantage of mapping or estimating, it is hard to obtain physical parameters directly by depending only on its learning phase. If we try to determine any physical meaning or obtain any physical parameters from the learning phase of ANN, a sensitivity study of ANN is necessary.

REFERENCES

1. Mohan, S. : Parameter estimation of nonlinear muskingum models using genetic algorithm, Journal of Hydraulic Engineering, Vol. 123, No. 2, pp.137-142, 1997.
2. Yoon, J., and Padmanabhan, G. : Parameter estimation of linear and nonlinear muskingum models, Journal of Water Resources Planning Management ASCE, Vol. 119, No. 5, pp.600-610, 1993.
3. Chen, C.S., Wang, N.B. : The modified muskingum flood routing model considering tributaries and lateral flow, Journal of Chinese Soil and Water Conservation, Vol. 27, No. 3, pp.195-204, 1996.
4. Zhichao, G., Robert, E. : Nuclear power plant performance study by neural networks, IEEE Transactions on Nuclear Science, Vol. 39, No. 4, pp.915-919, 1992.
5. El-Keib, A.A., Ma, X. : Application of artificial neural networks in voltage stability assessment, IEEE Transactions on Power Systems, Vol. 10, No. 4, pp.1890-1895, 1995.
6. Yang, C.C., Chen C.S., Chang, L.C. : Modeling of watershed flood forecasting with time series artificial neural network algorithm, 1998 International Water Resource Engineering Conference, pp.903-908, Memphis, August, 1998.
7. Zurada, J.M. : Introduction to artificial neural system, Info Access Distribution Pte Ltd., Singapore, 1992.
8. Shieh, M.F. : Application of neural networks in nonlinear hysteretic structural dynamic analysis, Master Dissertation, Department of Civil Engineering National Chiao Tung University, Taiwan, 1999.

9. Gill, M. A. : Flood routing by the muskingum method, Journal of Hydrology, Amsterdam, The Netherlands, 36, pp.353-363, 1978.
10. Tung, Y. K. : River flood routing by nonlinear muskingum method, J. Hydr. Div. ASCE, 111(12), pp.1447-1460, 1985.

APPENDIX – NOTATION

The following symbols are used in this paper:

K	= storage-time constant for the river reach ;
X	= weighting factor ;
f	= transfer function ;
C_0, C_1, C_2	= the parameters of the Muskingum linear function ;
S_t	= storage of t hour ;
I_t	= inflow discharge of t hour ;
$I_{t+\Delta t}$	= inflow discharge of t+ Δt hour, "Δt" is time step ;
O_t	= outflow discharge of t hour ;
$O_{t+\Delta t}$	= outflow discharge of t+ Δt hour ;
Q_{est}	= estimating flood discharge (cms) ;
Q_{obs}	= observed flood discharge (cms) ;
Q_{pobs}	= the peak discharges of flood of observation ;
Q_{pest}	= the peak discharges of flood of estimation ;
T_{pest}	= the times to peak discharge of estimation ;
T_{pobs}	= the times to peak discharge of observation, respectively ;
W_{ij}	= weights of the node ;
X_i	= input signals ;
Y_j	= output signals ; and
θ_j	= bias value.

(Received September 1, 2000 ; revised May 14, 2001)