Journal of Urban and Environmental Engineering, v.12, n.1, p.154-159

ISSN 1982-3932
doi: 10.4090/juee.2018.v12n1.154159

Journal of Urban and Environmental Engineering

www.journal-uee.org

DETERMINATION OF DOWNSTREAM FLOOD FLOW CONSIDERING INPUTS FROM DIFFERENT UPSTREAM RIVERS USING ANN

Briti Sundar Sil[1*] and Bahnisikha Das[2]

[1]Faculty of Civil Engineering of Department, NIT Silchar, India
[2] M.Tech Scholar of Civil Engineering of Department, NIT Silchar, India

Received 7 November 2017; received in revised form 16 May 2018; accepted 18 May 2018

Abstract: For estimating and forecasting of flood event, researchers and engineers mostly use the Muskingum flood routing method which is widely used throughout the world. The application of two parameter based Muskingum model is valid only for single inflow flood routing without any lateral inflow into the routing reach. However, normally a river is fed by a number of branch channels or rivulets at various upstream points. So, the single inflow-outflow Muskingum model cannot be applied in such situation. To overcome this problem, artificial Neural Network (ANN) has been applied in a river system considering inflow from various upstream rivers with a common outflow section. A simple static ANN model have been developed using concurrent discharge data. The model is applied in Mississippi River network starting from St. Louis, Montana to downstream section at Thebes, Illinois. In this reach, from St. Louis to Thebes, in the Mississippi river, a total of six lateral inflows confluence to the main river at different locations. Using ANN model, considering water discharge as input from all the upstream sections, water discharge at the most downstream section, Thebes is computed. Statistical performance analysis of the estimated data shows that ANN can be efficiently used for estimation of flood flow considering multiple inflows.

Keywords: flood routing, ANN, Muskingum, multiple inflows

* Correspondence to: Briti Sundar Sil, Tel.: +91 9435579130. E-mail: britisil@gmail.com

INTRODUCTION

Flood routing in river has gain extreme importance to water resource engineering. Muskingum flood routing method is very popular river flood routing procedures. In Muskingum method the weighted sum of the inflow and outflow is proportional to the storage in the reach through which flood is being routed. The original Muskingum method is valid for single channel reach without considering lateral inflows, only bounded by an inflow and outflow gauging sites, into the routing reach. However in natural channels, there is occurrence of lateral inflows and hence the Muskingum method is not applicable in such situation.

In this study a model has been prepared for flood routing in a river considering multiple inputs from the upstream tributaries. In this model all the upstream inflows are aggregated and converted to an equivalent single inflow which represents a hypothetical inflow at a characteristics point at the upstream of the catchment. For the development of the model, artificial neural network was used to estimate the parameters which represent the contribution of the upstream channel inflows for estimating the single equivalent inflows. A statistical performance of the model was analyzed and compared with the observed values. The analysis shows that the developed model can efficiently and successfully be used to estimate the required parameters which best describe the routing model considering multiple inflows.

An artificial neural network (ANN) can perform parallel link, correction of error, and nonlinear activation. It has become an arising tool for the prediction of flow and association of information. Its purpose is to achieve a forecast of system reaction without an attempt to understand in detail about the character of the phenomena which it represents (Rumelhart *et al.,* 1986, Fahlamn 1989, Hornik 1989, Rogers and Lamarsh 1992, Haykin 1994, Fausett 1994). It consists of nonlinear input–output representation, which includes an input layer, output layer, and hidden layer units called neurons. This paper includes the application of ANN to the flow problems in an intricate channel network. The objective is to build up an ANN channel network model for the simulation and prediction of discharge in a channel network and also to express the efficiency and effectiveness of ANN method. The model uses the actual river network which is simulated as ANN architecture and includes the substantial performance and interior situation of the river system. The incorporation of physical behavior in the model and using natural river network for ANN creation make it feasible to have the most favorable architecture for modeling ANN and also make it easier for the water resource engineers to understand the techniques and deduce the result. The model is proposed for real-time forecasting of the flow in an intricate river system with less number of data that is necessary for the information of topography compared to a usual hydrodynamic model without negotiating the precision of modeling.

METHODOLOGY

Overview of ANN

An ANN represents the biological neural network mathematically in an easy way which has the capacity to learn from the examples, identify a pattern in the data, adjust solution over time, and processes the information quickly. The use of ANN to water resource issues is swiftly advancing the popularity because of their huge control and potential in mapping the non linear system data. A water resource system is generally nonlinear and consists of multiple variable and the various variables may have compound relationship between each other. ANNs can be competently used for solving such problems.

An ANN comprises of a number of data processing elements known as neurons or nodes that are arranged in layers. All the neurons present in different layers are directly connected to each other. The neurons in its own layer are not connected within them. The neurons present in the input layer obtain the input values which are then transferred to the subsequent layer of the network and the procedure continues till the values in the output layer are achieved. The data that is passing from one node to other node are multiplied by the weights which manage the power of a transient signal. The weight represents the information that is used by the net for solving a problem. The associated weight is multiplied with the input value for each neuron, the product of which is then added together and thereafter passed through a transfer function to achieve the result. By regulating the weights associated with each node, the desired output is achieved and the error function value is calculated for a definite input which is then back-propagated from a layer to its preceding layer (Rumelhart *et al.,* 1986). This type of network where the data flows in a single direction is called the feed forward network (White 1990, Gallant and White 1992, Rumelhart *et al.,* 1986). Among many ANN structures used in the area of hydrology, the most widely used is the multilayer feed forward network (Rumelhart *et al.,* 1986).

Training of ANN

Training of ANN network is the procedure by which its weights are determined. A set of input and known output data are used for training ANN. At the beginning the initialization of the weights is done with a set of arbitrary values or on the basis of some earlier practice. Then, the weights are altered analytically with the help of learning algorithm so that the dissimilarity between the output obtained by ANN and observed value is very less for a

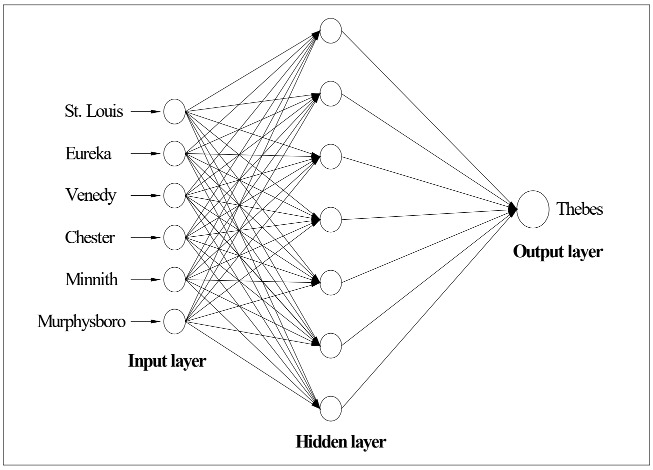

Fig. 1 Three layer Feed-forward ANN.

given input (**Fig. 1**). The learning process of ANN is completed when this variation between them is less than a particular acceptance. Generally, the execution is applied under two circumstances: (i) whenever the weights are simplified a highest number of times and (ii) whenever the errors computed for a different test dataset begins to rise.

The most commonly used learning rule for ANN is the error back-propagation (BP) algorithm developed by Rumelhart *et al.*, (1986) (Bishop 1995, Tchaban *et al.*, 1998). The back-propagation algorithm depends on the delta rule. In back-propagation procedure, the weights are changed for all neurons on the basis of the collected error derivatives in connection with every weight. From the training set, a set of input and output data are chosen and then output is computed by the network from the inputs. From the actual output, this output is deducted for getting the error of the output layer. The error is then back propagated throughout the network to adjust the weights properly. The procedure is sustained until a given error acceptance is achieved. Over the training samples, the mean square error is to be reduced which is the representative objective function. After completing the training process, the validation and implementation of the ANN performance is done for its planned use. An ANN is trained in a superior way with a wider range of input data. If the model is trained with a less number of input values then it may perform badly and hence the failure may limit its application where the accessible data for calibration is insufficient to include all probable situations. The performance of an ANN model was enhanced by Imrie *et al.*, (2000) which were achieved by including supervision to the learning mechanism and adding an easy cross-validation process in training ANN. The system can create models which can generalize well on the fresh data and can extrapolate outside the range of values which is incorporated into the calibration range.

The process of determining the most favorable network architecture is a component of the learning policy (Lebiere and Fahlman 1990). The number of input, output, and hidden layer neurons is problem dependent. There is no definite rule to find out the number of neurons in the intermediate layers. Though, if the neurons in the hidden layer are very less, then the network may not be trained properly with adequate degrees of freedom and if the number is very high, then time requirement will be very high for training and the network may also over fit the data (Karunanithi *et al.*, 1994). The arrangement that provide the least mean square error (MSE) was preferred as the structure of ANN (Chalisgaonkar and Jain 2000). In this paper ANN is utilized for modeling the river system on the basis of the actual structure of the river so that the complexity in finalizing the hidden layer structure can be evaded.

Description of ANN for a River System

A natural river network consists of various lakes, tributaries, channels, and other water bodies linked together. It may be nonlinear and consists of multiple variables which may have compound relationships between each other. ANNs can be utilized in such situation for solving the problems effectively because of the similarities involved between a river system and neural network (**Fig. 2**). Hence, it is reasonable to apply an ANN model in a compound river network to simulate the flow process.

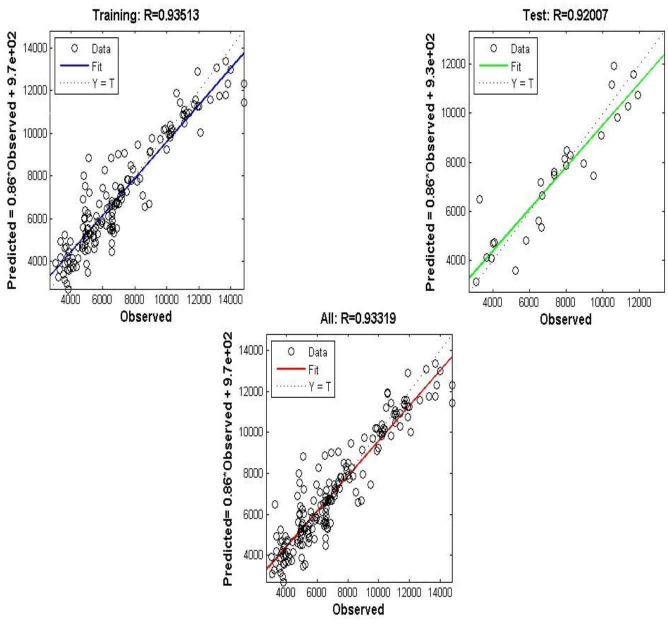

Fig. 2 Observed discharge versus Predicted discharge

Table 1 Statistical parameter of training and testing data

Statistical parameter	Station name						
	St.Louis	Eureka	Venedy	Chester	Minnith	Murphysboro	Thebes
Training data							
Mean	6914.36	166.50	78.36	5206.86	63.79	1.64	7024.10
Standard deviation	2828.68	220.18	44.89	2596.77	79.93	4.30	2869.30
Maximum	13900.00	1160.00	177.00	14200.00	328.00	35.70	14800.00
Minimum	2440.00	26.70	3.40	2050.00	2.75	0.14	3030.00
Testing data							
Mean	6535.93	125.50	74.25	5179.26	73.90	1.24	7470.74
Standard deviation	2498.87	197.42	50.79	2097.34	96.96	1.71	2686.66
Maximum	14400.00	1100.00	176.00	12200.00	314.00	8.16	11900.00
Minimum	2540.00	28.90	3.57	2970.00	2.86	0.19	3090.00

A river network can be conceptualized as per the requirements of the researcher and the correlation and interaction between water bodies. Some assumptions are made for building an ANN model for the flow of water. The network of the river is represented by a system consisting of interrelated nonlinear basins. The model input and output comprises upstream inflows and downstream outflows from the various water bodies. It is assumed that the interaction between reservoirs in the same layer of the network is nil. However, weight is used to represent the relations between reservoirs in the adjoining layers, which is assigned zero if there is no interaction between them that indicates the similarity between the river system and the ANN. The neurons in the initial layer of the network indicate input and that of the last layers serve as output both of which does not have storage capability. The neurons in the hidden layers may have storage functions through which there is exchange of water.

From the various statements and consideration mentioned above, the river network is represented as three basic components: water inflow as input, internal reservoirs in parallel or series, and water outflow as output. The relationship between the input to the first layer and the output from the last neuron is non-linear.

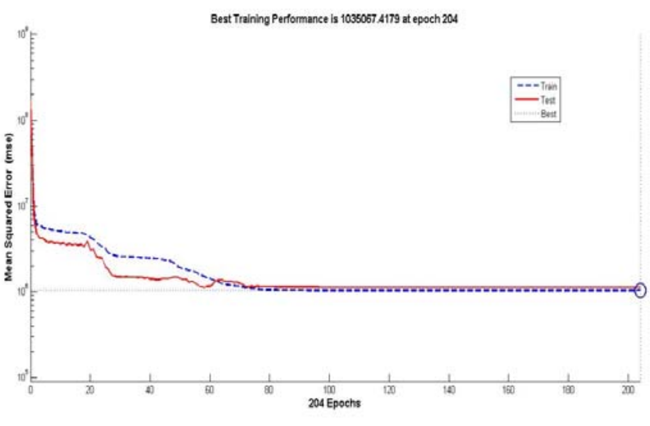

Fig. 3 Performance of ANN model for discharge prediction

Development of ANN model for prediction of discharge

In the present study, an ANN model was developed to predict the discharge of Mississippi River basin in the United States. The river has many tributaries in which the flow from various stations i.e. St. Louis (Montana), Eureka (Montana), Venedy (Illinois), Chester (Illinois), Minnith (Montana), Murphysboro (Illinois) etc meets at station Thebes (Illinois). The flow rate of the various gauging stations starting from April 1, 1981 to September 30, 1981 is collected from USGS station. The ANN model was developed by considering discharge at St. Louis (Montana), Eureka (Montana), Venedy (Illinois), Chester (Illinois), Minnith (Montana) and Murphysboro (Illinois) stations as input and discharge at Thebes (Illinois) station as output. A multilayer feed-forward perceptron with one hidden layer was adopted in the present study (**Table 1**). The transfer function at the hidden layer was tan hyperbolic tangent whereas transfer function at the output layer was pure linear. To find the optimum number of neurons in the hidden layer, the number of neurons in the hidden layer were varied from four (4) to fifteen (15). Before presenting to the network the data were normalized to fall in the range [-1, +1]. Data division was carried out by using 70% data for training and rest 30% for testing. A total of 183 data were used for developing ANN model for discharge better generalization to the training data. The ANN model was implemented in MATLAB R2013a environment. The performance function was set to be mean squared error (MSE). However the networks performance was also reported via two other statistical parameters namely mean absolute percentage error (MAPE) and linear co-relation coefficient (R). Statistical parameters *MSE*, MAPE and R are defined as follows:

Table 2 Performance of ANN model with various number of neurons in the hidden layer

Number of neurons in the hidden layer	MSE × 10^6			MAPE			R		
	All	Training data	Testing data	All	Training data	Testing data	All	Training data	Testing data
4	1.61	1.34	3.16	12.28	11.19	18.78	0.895	0.913	0.793
5	1.54	1.34	2.67	11.43	10.90	14.10	0.900	0.910	0.845
6	1.37	1.21	2.26	10.61	10.51	11.19	0.912	0.922	0.866
7	1.04	1.03	1.12	9.45	9.42	9.62	0.933	0.935	0.920
8	0.92	0.64	2.56	8.53	7.29	15.20	0.942	0.961	0.821
9	0.95	0.58	3.09	7.97	6.98	11.97	0.940	0.959	0.875
10	0.69	0.28	3.09	5.68	4.27	13.80	0.958	0.984	0.798
11	0.86	0.37	3.69	7.18	5.22	16.08	0.946	0.976	0.808
12	1.01	0.29	5.20	5.21	4.01	12.92	0.939	0.982	0.756
13	3.17	0.08	21.05	7.11	2.41	27.81	0.853	0.996	0.566
14	0.60	0.18	3.01	4.90	3.14	14.16	0.964	0.989	0.874
15	0.77	0.26	3.67	5.74	4.37	13.59	0.952	0.984	0.782

Table 3 Weights and biases of ANN model with seven neurons in the hidden layer

Hidden neuron		1	2	3	4	5	6	7
		Weights						
Station name	St.Louis	-0.4242	-0.1375	-0.3856	-0.0604	-0.2837	-1.7506	0.0475
	Eureka	1.8813	-2.5743	-1.1114	0.0648	2.5084	0.0208	3.1853
	Venedy	-0.8685	1.2929	-2.2101	1.5598	-0.3801	-0.3262	1.5660
	Chester	0.6017	2.0765	0.6184	2.2197	-0.6666	-1.7561	-0.6660
	Minnith	-0.7666	1.0701	1.7490	-0.4829	-0.9462	-0.7424	-0.5708
	Murphysboro	-0.6323	0.0056	0.4111	0.6828	0.0101	1.8463	-1.0195
	Thebes	-1.3247	1.7743	-1.5386	-0.9376	2.6868	-0.9197	-2.4968
		Biases						
Hidden layer		0.1924	-0.8983	-0.6043	0.6729	0.6844	-1.0723	1.8239
Output layer		-0.1946						

$$MSE = \frac{1}{N}\sum_{i=1}^{n}\left(x_m - x_o\right)^2 \tag{1}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{n}\frac{\left|x_m - x_o\right|}{x_o} \tag{2}$$

$$R = \frac{n\left(\sum x_m x_o\right) - \left(\sum x_m\right)\left(x_o\right)}{\sqrt{\left\{n\left(\sum x_m^2\right) - \left(\sum x_m\right)^2\right\}\left\{n\left(\sum x_o^2\right) - \left(\sum x_o\right)^2\right\}}} \tag{3}$$

where x_o and x_m are the observed data and predicted data respectively and n is the total number of data.

RESULTS AND DISCUSSION

The performance of the ANN networks with different number of neurons in the hidden layer is presented in **Table 2**. From the **Table 2**, it may be observed that MSE, MAPE & R value of testing data is optimum with seven (7) neurons in the hidden layer (highlighted in bold & Italic). MSE, MAPE & R values of training and testing data of the network with seven neurons in the hidden layer were found to be 1.03×106, 9.42% and 0.935 &, 1.12×106, 9.62% and 0.920 respectively. MSE, MAPE & R values of both training and testing data implied that the ANN model has successfully learnt and predicted the discharge. The weights and biases of the ANN model with seven hidden layers are presented in **Table 3**. **Figure 2** shows the plot of observed discharge against the predicted discharge for training, testing and all data including training and testing. The performance of ANN model is presented in **Fig. 3**. It may be observed from **Fig. 3** that MSE value started at a large value and decreased as the number of iteration (Epoch) increased. It is also observed from **Fig. 3** that by the 62nd iteration there was no over fitting in the data which suggests that Bayesian Regularization has successfully prevented over fitting of testing data.

CONCLUSIONS

Flood routing of a river system having multiple inflows cannot be carried out by basic Muskingum model since it is valid only for single inflow-single outflow flood routing system. As because there exists a similarity in input-output relationship between the river system and the neural network, so adoption of artificial neural network for the development of simulating model defining flood routing on a river network is convenient. The benefit of utilizing ANN in this study is its ability to model the behavior of unsteady flow in an intricate river network. The model has been developed by considering the water flow from St. Louis, Eureka, Venedy, Chester, Minnith and Murphysboro as input and water flow from Thebes as output of the Mississippi river. The performance analysis shows that the model is working well and matches with the observed data with R2 more than 0.90. So, from the study it is concluded that the developed model can describe the flow at a common downstream section in a river reach considering multiple inflows from the upstream tributaries.

REFERENCES

Bishop, C. M. (1995). Neural networks for pattern recognition. Clarendon Press, Oxford.

Fahlman, S. E., and C. Lebiere. (1990). The cascade-correlation learning architecture, Technical report CMU-CS-90–100, School of Computer Science Carnegic-Mellon University, Pittsburgh.

Fausett, L. (1994). Fundamental of neural networks, Prentice- Hall, Englewood Cliffs, New Jersey.

Gallant, A. R., and H. White. (1992). On learning the derivatives of an unknown mapping with multiplayer feed forward networks. *Neur. Networks* **5**(1):129–138.

Haykin, S. (1994). Neural networks: a comprehensive foundation, Macmillan, New York.

Hornik, K. (1989). Multilayer feed forward networks are universal approximations. *Neur. Networks* **2**(4):359–366.

Imrie, C. E., Durucan, S., and Korre A. (2000). River flow prediction using artificial neural networks: generalization beyond the calibration range. *J. Hydrol.,* **233**(1–4): 138–153.

Jain, S. K., and Chalisgaonkar D. 2000). On setting up stage discharge relations using ANN. *J. Comp. Civil Enging.,* **5**(4):428–433.

Karunanithi, N., W. J. Grenney, D. Whitley, and K. Bovee. (1994). Neural networks for river flow prediction. *J. Comp. Civil Enging.,* **8**(2):201–220.

Rogers, J. L., and W. J. Lamarsh. (1992). Application of a neural network to simulate analysis in an optimization process' Proceedings, Artificial Intelligence in Design '92, Kluwer Academic, Boston, 739–754.

Rumelhart, D. E., J. L. McLelland, and PDP Research Group. (1986). Parallel distributed processing, explorations in the micro structure of cognition, Vol. I: Foundations, MIT Press, Cambridge, Massachusetts.

Tchaban, T., M. J. Taylor, and J. P. Griffin. (1998). Establishing impacts of the inputs in a feed forward neural network. *Neur. Comp. Appl.* **7**(3):309–317.

White, H. (1990). Connectionist nonparametric regression: multiplayer feed-forward networks can learn arbitrary mappings. *Neur. Networks* **3**(6):535–549.