

Neural Network Modelling of Flow in Yinluoxia Station Based on Flow in Zhamashike Station in Heihe River, China

W. Huang

Department Hydraulic Engineering
Tongji University
Shanghai, China

Department of Civil Engineering, Florida State University,
Florida, USA

Y.N. Chao

Department Hydraulic Engineering
Tongji University
Shanghai, China

S.D. Xu

Department of Harbor, Waterway and Coastal Engineering
School of Transportation, Southeast University
Nanjing, China

Y. Cai

Department Hydraulic Engineering
Tongji University
Shanghai, China

F. Teng

Department Hydraulic Engineering
Tongji University
Shanghai, China

B.B. Wang

Department Hydraulic Engineering
Tongji University
Shanghai, China

Abstract—Artificial neural network model was established between two river flow in two hydrological stations, Yinluoxia Station and Zhamashike Station in upper Heihe River basin. Results indicate very good correlations for the general trend of the flow data at two stations with correlation coefficients of 0.86 and 0.94 for 2004 and 2005, respectively. Major differences between model results and observations occur near the peak flow or flood periods. This indicates that other factors, such as local rainfalls, can be included in future study to further improve the model accuracy.

Keywords—artificial neural network; flow; Yinluoxia; Zhamashike; Heihe River

I. INTRODUCTION

The Heihe River basin, located in the northwest China, encompasses Qinghai and Gansu Provinces and Inner Mongolia in China. The Heihe River originates from glacial melt water and a fair amount of precipitation in the Qilian Mountains forming the northern periphery of the Tibetan Plateau. The Heihe River with the length of 821 Km is the second largest river in the interior of Northwest China. It originates from the Qilian Mountains and flows to the Gobi desert. Annual precipitation of the Heihe River basin is approximately 300-500 mm in the upstream mountain area while 30-50 mm in the downstream desert area. The melting of glaciers and snow on those mountains have provided water for irrigation to people living in oasis cities. The extensive overuse of surface water for irrigation in the middle oasis reaches has triggered a series of severe environmental problems such as the disappearances of the river and terminal

lakes and a severe decline of the groundwater level in the lower desert reach. The annual ranges of precipitation in the upper reach is about 300 to 500 mm.

Long term flow data and the number of hydrological measurement are important for water resources study and management in Heihe River basin. However, because of the cost, long term flow data are available only on limited stations, while short term flow data are available in many stations. Establishment of correlation between long term data station and short term data station is a cost effective way to derive more data to support water resources study. There are several hydrological measurement stations in the upper reach of the Heihe River (Figure 1). Yingluoxia Station considered as the outlet of the flow from the mountain area of the upper river reach to the middle reach. Zhamashike Station is located in the mountain area, which can be used to characterize the runoff and snow melt effects on the river flow. The distance along the river between Zhamashike to Yinluoxia stations is more than 200 km.

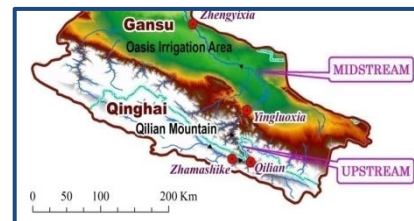


FIGURE 1. LOCATIONS OF HYDROLOGICAL MEASUREMENT STATION AT ZHAMASHIKE AND YINLUOXIA IN THE STUDY AREA OF HEHEI RIVER BASIN, CHINA.

Artificial neural network (ANN) is a recognized powerful tool which has been widely used in multivariate signal processing. It can provide information only at locations where input data are available for model training. In this study, a neural network model was established and tested for predicting river flow in Yinluoxia Station based on the flow data in Zhamashike Station. If it is successful, it will provide a good example and reference for ANN modeling to derive long-term data in other stations where only short term data are available.

II. BACKPROPAGATION ARTIFICIAL NEURAL NETWORK METHODOLOGY

In analogy to the biological nervous system, ANN technology is being applied to solve a wide variety of complex scientific, engineering and business problems. Neural networks are ideally suited for such problems because like their biological counterparts, an ANN can learn, and therefore be trained to find solutions, recognize patterns, classify data, and forecast future events. Hagan and Demuth [1] and Haykin [2] give the details of the theories and engineering applications of the neural networks. In a neural network model, the outputs are correlated to the inputs through the neurons (or nodes) with weights and bias. The behavior of a neural network is defined by the way its individual computing elements are connected and by the strength of those connections or weights. The weights are automatically adjusted by training the network according to a specific learning rule until it performs with the desired error rate.

A. Multiple-Layer Model

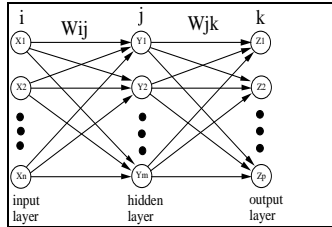


FIGURE II. THREE-LAYER FEED-FORWARD NEURAL NETWORK FOR MULTIVARIATE SIGNAL PROCESSING.

In practical applications, a neural network often consists of several neurons in several layers. A schematic diagram of a three-layer neural network is given in Figure II, where X_i ($i=1, \dots, n$) represents the input variables (such as boundary forcing functions of wind and water levels); Y_i ($i=1, \dots, m$) represents the outputs of neurons in the hidden layer; and Z_i ($i=1, \dots, p$) represents the outputs of the neural network such as water levels and currents in and around coastal inlets. The layer that produces the network output is called the *output layer*, while all other layers are called *hidden layer*. The weight matrix connected to the inputs is called the *input weight* (W_{ij}) matrix, while the weight matrices coming from layer outputs are called *layer weights* (W_{jk}).

B. Standard Network Training Using Gradient Descent Method

Multiple-layer neural network using backpropagation training algorithm is popular in neural network modeling [1]

because of its ability to recognize the pattern and relationship between non-linear signals. The term of backpropagation usually refers to the manner in which the gradients of weights are computed for non-linear multi-layer networks. A neural network must be trained to determine the values of the weights that will produce the correct outputs. Mathematically, the training process is similar to approximate a multi-variable function, $g(X)$, by another function of $G(W, X)$, where $X=[x_1, x_2, \dots, x_n]$ is the input vector, and $W=[w_1, w_2, \dots, w_n]$ the coefficient or weight vector. Therefore, the training task is to find the weight vector 'W' that provides the best approximation of the function $g(X)$ based on the training input $[X]$.

The standard or basic training method is called '*Gradient Descent Method*'; in which weight changes move the weights in the direction where the error declines most quickly. Training is carried out by assigning random initial weights to each of the neurons (usually between 0.1 to 1.0), and then presenting sets of known input and *target (output)* values to the network. The network estimates the output value from the inputs, compares the model predicted output to the target value, and then adjusts the weights in order to reduce the mean squared difference between the network output and the target values. The complete input-output sets are often run through the network for several iterations (or epochs) until either the mean square error is reduced to a given level or reaches a minimum, or until the network has been trained for a given number of iterations.

If we let w_m represent the value of weight w after m -th iteration in a neuron, then

$$w_m = w_{m-1} + \Delta w_m \quad (1)$$

Where Δw_m is the change in the weight w at the end of iteration m . It is calculated by

$$\Delta w_m = -\varepsilon d_m \quad (2)$$

Where ε is the user-specified parameter controlling the proportion by which the weights are modified. The term d_m is given by

$$d_m = \sum_{n=1}^n \left(\frac{\partial E}{\partial w_m} \right) \quad (3)$$

Where N is the total number of examples, and E is the simulation output error.

C. Faster Training by Network Optimization with Conjugated Gradient Algorithm

In practical ANN applications, optimized training algorithms are often used. There are several optimized training algorithms as described by Haykin (1999), such as scaled conjugated gradient backpropagation. One of the optimized methods is the scaled conjugate gradient algorithm (SCG) (Fitch et al., 1991; Moller, 1993; Hagan, 1995). The SCG optimized training algorithm was designed to avoid the time consuming line search. In the conjugate gradient algorithm a search is performed along conjugate directions, which produces faster convergence than steepest descent

directions. It differs from the previously mentioned error backpropagation in gradient calculations and subsequent corrections to weights and bias. Here, a search direction, \bar{d}_k , is computed at each training iteration, k , and the error function is minimized along it using a line search. The gradient descent does not move down the error gradient as in the above backpropagation method, but along a direction that is conjugate to the previous step. The change in gradient is taken as orthogonal to the previous step, with the advantage that the function minimization, carried out in each step, is fully preserved due to lack of any interference from subsequent steps.

The basic iteration process (Fitch, et al., 1991; Hagan 1995) is given as follows:

1. Initialize weight vector, \bar{w} , by using uniform random numbers from the interval $(-0.5, 0.5)$. Calculate error gradient, \bar{g}_0 , at this point. Select initial search direction

$$\bar{d}_0 = -\bar{g}_0 \quad (4)$$

2. For each iteration k , determine constant α_k , which minimizes the error function $f(\bar{w} + \alpha_k \bar{d}_k)$ by line search where \bar{d}_k is the search direction at iteration k . Update the weight vector \bar{w}_k to \bar{w}_{k+1} using:

$$\bar{w}_{k+1} = \bar{w}_k + \alpha_k \bar{d}_k \quad (5)$$

3. If error at this iteration, $k+1$, is acceptable, or if specified number of computations of the function and gradients is reached, terminate the algorithm.

4. Otherwise obtain new direction vector, \bar{d}_{k+1} :

$$\bar{d}_{k+1} = -\bar{g}_{k+1} \quad (6)$$

if $k+1$ is an integral multiple of N , where N is the dimension of \bar{w} . Otherwise,

$$\bar{d}_{k+1} = -\bar{g}_{k+1} + \beta_k \bar{d}_k \quad (7)$$

where

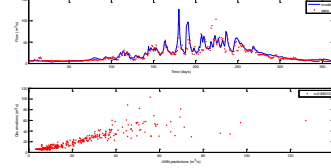
$$\beta_k = \frac{(\bar{g}_k \bar{g}_k^T)}{(\bar{g}_{k-1} \bar{g}_{k-1}^T)} \quad (8)$$

III. ANN MODELING OF FLOW AT YINLUOXIA BY FLOW AT ZHAMASHIKE

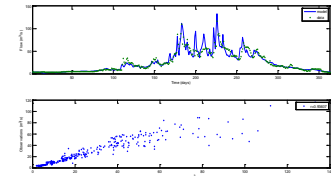
Flow data in 2004 and 2005 were used in ANN model simulations. While flow data in 2004 were used as model training, flow data in 2005 were used as model verification. ANN model was developed under Matlab software environment as described by Huang and Foo [3] and Huang et al. [4].

Comparison between observed data and ANN model simulation for 2004 in the model training period is given in

FigureIII. It shows that ANN model simulations match well with data, with the correlation coefficient of 0.86 and root-mean-square-error (RMSE) of 17.7 m³/s. Comparing to the maximum flow of about 200 m³/s, the relative error is about 8.8 %. Most deviations between model and data occur near peak flow, which indicates that, for short-term flood events, additional factors such as rainfall runoff need to be included.



FIGUREIII. COMPARISON OF FLOW BETWEEN ANN MODEL SIMULATIONS AND OBSERVED DATA IN 2004 DURING MODEL TRAINING PERIOD AT YINLUOXIA STATION.



FIGUREIV. COMPARISON OF FLOW BETWEEN ANN MODEL SIMULATIONS AND OBSERVED DATA IN 2005 DURING MODEL VERIFICATION PERIOD AT YINLUOXIA STATION.

Comparison between observed data and ANN model simulation for 2005 in the model verification period is given in FigureIV. It shows that ANN model simulations match well with data, with the correlation coefficient of 0.94 and root-mean-square-error (RMSE) of 14.0 m³/s. Comparing to the maximum flow of 250 m³/s, the relative error is about 5.0 %. Therefore, the ANN model generally produce reasonable flow predictions of the general trend of river flow at Yinluoxia Station based on the flow data at the upstream station at Zhamashike. However, similar to the situation in 2004, most deviations between model and data occur near peak flows or floods. Considering that floods are often caused by combination of factors of upstream inflow and local rainfall runoff, this indicates that, for short-term flood events, additional factors such as rainfall runoff need to be included.

IV. CONCLUSION

In this study, the artificial neural network model was established between two river flow in two hydrological stations, Yinluoxia Station and Zhamashike Station in upper Heihe River basin. Results indicate very good correlations for the general trend of the flow data at two stations with correlation coefficients equal or above 0.9 for 2004 and 2005 flow data. Major differences between model results and observations occur near the peak flow or flood periods. This indicates that other factors, such as local rainfalls, can be included in future study to further improve the model accuracy. Results from this study can provide a good reference for ANN modeling of river flow in other stations in Heihe River.

ACKNOWLEDGEMENTS

The investigation is supported by the National Natural Science Foundation (No. 91225301), the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (No. 2013-1792) and the Fundamental Research Funds for the Central Universities. Special thanks go to the Center for Water Research of Beijing University, and Cold and Arid Regions Science Data Center at Lanzhou for providing the data.

REFERENCES

- [1] Hagan, M.T., Demuth, H., and Beale, M., Neural Network Design, PWS Publishing Company, 20 Park Plaza, Boston, MA 02116 – 4324, 1995.
- [2] Haykin, S., Neural Network: A Comprehensive Foundation. Prentice Hall, New Jersey, 1999.
- [3] Huang, W. & Foo, S., Neural network modeling of salinity variation responding to multiple forcing functions in Apalachicola River, Water Research, vol. 36, pp. 356–362, 2002.
- [4] Huang, W., Murray, C., Kraus, N. & Rosati, J., Development of a regional neural network for coastal water level predictions, Ocean Engineering, vol. 30, pp. 2275–2295, 2003.