



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

River Flow Forecasting Using Neural Networks and Auto-Calibrated NAM Model with Shuffled Complex Evolution

^{1,2}M. Zakermashfeh, ¹M. Ghodsian, ¹S.A.A. Salehi Neishabouri and ³M. Shakiba

¹Department of Civil Engineering, Tarbiat Modarres University, Tehran, Iran

²Department of Civil Engineering, Jundi-Shapur University of Technology, Dezful, Iran

³Department of Electrical Engineering, K.N. Toosi University of Technology, Tehran, Iran

Abstract: River flow forecasting is required to provide important information on a wide range of cases related to design and operation of river systems. Since there are a lot of parameters with uncertainties and non-linear relationships, the calibration of conceptual or physically-based models is often a difficult and time consuming procedure. So it is preferred to implement a heuristic black box model to perform a non-linear mapping between the input and output spaces without detailed consideration of the internal structure of the physical process. In this study, the capability of artificial neural networks for stream flow forecasting in Kashkan River in West of Iran is investigated and compared to a NAM model which is a lumped conceptual model with shuffled complex evolution algorithm for auto calibration. Multi Layer Perceptron and Radial Basis Function neural networks are introduced and implemented. The results show that the discharge can be more adequately forecasted by Multi Layer Perceptron neural network, compared to other implemented models, in case of both peak discharge and base flow forecasting.

Key words: River flow forecasting, multi layer perceptron, radial basis function, hydrologic model, shuffled complex evolution, Kashkan river

INTRODUCTION

Estimation or forecasting the magnitude of hydrological variables is required in case of extreme events risk management, water resources planning and management and river structures operation. Flood warning, drought forecasting and optimal operation of the reservoirs and power plants can be performed by appropriate river flow forecasting.

Many approaches have been used to hydrologic forecasting in the last few decades. These models are grouped into three categories (Babovic and Bojcow, 2001).

Lumped conceptual models: These types of models approximate the general internal sub-processes and physical mechanisms which govern the hydrologic cycle. The implementation and calibration of lumped conceptual models is encountered with some difficulties, requires significant amount of calibration data and depends on the expertise and experience of the user.

Distributed physically based models: There are not so many models of this kind, suitable for research. This type of modeling systems needs a large amount of data. They utilize many parameters in their operation in a direct

relation to topology, soil, vegetation and geology characteristics of catchments. In the other hand, many of these parameters are not directly measured everywhere in the studied basin.

Empirical black box models: These kinds of models do not use any explicit well-defined representation of the physical process and governing equations of the phenomena. The accuracy of these models is highly dependent on the quality of the observed data used to calibrate or train them. However, it has been shown that these models are useful operational tools and they are the only option in case where there are not enough meteorological data available.

In this regard, Artificial Neural Networks are non-linear, distributed, heuristic and adaptive learning models constructed from many simple processing elements, called Neuron.

Artificial Neural Network (ANN) is a mathematical tool, which tries to represent low-level intelligence in natural organisms (Gorzalczy, 2002) and it is a flexible structure, capable of making a non-linear mapping between input and output spaces.

Neural network processing is based on performance of many simple processing units called neuron. Each neuron in each layer is connected to all elements in

the previous and the next layer with links, each has an associative weight. The general ability of an ANN is to learn (Karayiannis and Venetsanopoulos, 1993) and to simulate the natural and complex phenomena.

The application of the ANN in hydrology, especially in rainfall-runoff modeling, has become very popular in recent years. The particular advantage of ANN is that, even if the exact relationship between sets of input and output data are unknown but is acknowledged to exist, the network can be trained to learn that relationship, requiring no a prior knowledge of the watershed characteristics (Pang *et al.*, 2007). Such applications can be found in many papers published recently (French *et al.*, 1992; Zhu *et al.*, 1994; Hsu *et al.*, 1995; Smith and Eli, 1995; Minns and Hall, 1996; Shamseldin, 1997; Campolo *et al.*, 1999; Gautam *et al.*, 2000; Imrie *et al.*, 2000; Chang and Chen, 2001; Shamseldin and O'Connor, 2001; Xiong and O'Connor, 2002; Sivakumar *et al.*, 2002; Campolo *et al.*, 2003; Wilby *et al.*, 2003; Jain *et al.*, 2004; Rajurka *et al.*, 2004; Xiong *et al.*, 2004; Chau *et al.*, 2005; Chau, 2006; Dawson *et al.*, 2006). From all these works, ANN models can be divided into updating mode and simulation mode. When the network input includes the runoff or the water level for the previous time period, such ANN models are in updating mode. It is the most popular mode of recent ANN applications and applied with varying degrees of success. Shamseldin (1997) and Rajurka *et al.* (2004) have developed the ANN models with simulation mode. Comparing with other empirical black-box models, the ANN models of simulating mode also show advantages in flood simulation and forecasting. According to the two modes, The ANN models are regarded as black-box models that cannot provide any physically realistic structure and parameters to represent the hydrological processes in watersheds, even though they are capable of identifying complex nonlinear relationships between rainfall and runoff time series. Hsu *et al.* (1995) have showed that the ANN approach does provide a viable and effective alternative to the ARMAX time series approach for developing input-output simulation and forecasting models in situations that do not require modeling of the internal structure of the watershed.

MATERIALS AND METHODS

Case study and dataset: The aim of this study is to predict the hourly discharge (in 12 h steps) for Kashkan River at Poledokhtar hydrometry station, located in the Karkheh River Basin in the South-West of Iran which is showed in Fig. 1. Multi Layer Perceptron (MLP) and Radial Basis Function (RBF) neural networks are used and compared to the NAM hydrologic model which utilizes a Shuffled

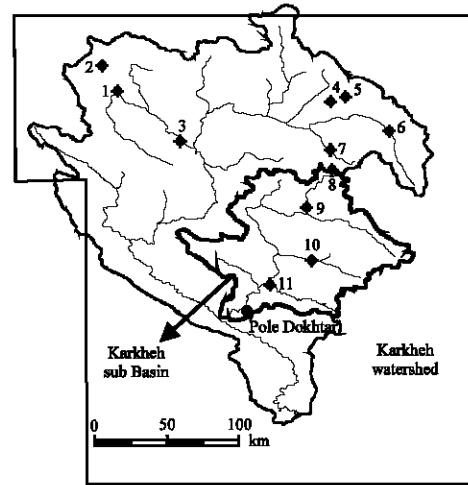


Fig. 1: Kashkan River sub-basin

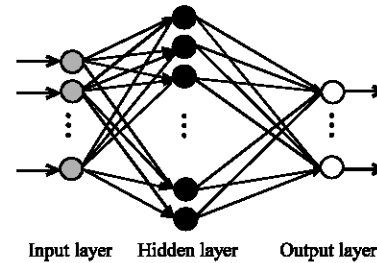


Fig. 2: Multi layer perceptron neural network

Complex Evolution (SCE) algorithm for model automatic calibration. In this regard, discharges at Poledokhtar station, precipitation, evaporation and temperature of 11 different stations during the period of 1994-1999 have been collected, in which the greatest floods occurred and divided into two sets for model training/calibration and verification. The training/calibration set is the data from the period of 1994-1998 and the rest of data is used to verification.

Multi layer perceptron: The MLP network, sometimes called Back Propagation (BP) network, is shown in Fig. 2, is probably the most popular ANN in engineering problems in the case of non-linear mapping and is called Universal Approximator (Hecht-Nielsen, 1987).

The learning process is performed using the well-known BP algorithm (Werbos, 1974; Rumelhart *et al.*, 1986). In this study, the standard BP algorithm based on the delta learning rule is used.

Two main processes are performed in a BP algorithm. A forward pass and a backward pass. In the forward pass, an output pattern is presented to the network and its effect is propagated through the network, layer by layer. For each neuron, the input value is calculated as follows:

$$\text{net}_i^n = \sum_{j=1}^m w_{ji}^n \cdot O_j^{n-1} \quad (1) \quad \alpha = \text{The momentum term, both in the range of } (0, 1) \text{ (Rumelhart } et al., 1986)$$

Where:

- net_i^n = The input value of i^{th} neuron in n^{th} layer
- w_{ji}^n = The connection weight between i^{th} neuron in n^{th} layer and j^{th} neuron in the $(n-1)^{\text{th}}$ layer
- O_j^{n-1} = The output of j^{th} neuron in the $(n-1)^{\text{th}}$ layer
- m = The No. of neurons in the $(n-1)^{\text{th}}$ layer

In each neuron, the value calculated from Eq. 1 is transferred by an activation function. The common function for this purpose is the sigmoid function, which is given by:

$$\text{Sig}(\text{net}_i^n) = 1 / (1 + \exp(-\text{net}_i^n)) \quad (2)$$

Hence, the output of each neuron is computed and propagated through the next layer until the last layer. Finally, the computed output of the network is prepared to be compared with the target output. In this regard, an appropriate objective function such as the Mean Relative Error (MRE) or Root Mean Square Error (RMSE) is calculated. The former which is used in this study is computed as follows:

$$\text{MRE} = \frac{1}{n_p \cdot n_o} \sum_{i=1}^{n_p} \sum_{j=1}^{n_o} \frac{|T_{ij} - O_{ij}|}{T_{ij}} \times 100 \quad (3)$$

Where:

- T_{jp} = The j^{th} element of the target output vector related to the p^{th} pattern
- O_{jp} = The computed output of j^{th} neuron related to the p^{th} pattern
- n_p = The number of patterns
- n_o = The number of neurons in the output layer

After calculating the objective function, the second step of the BP algorithm, i.e., the backward process is started by back propagation of the network error to the previous layers. Using the gradient descent technique, the weights are adjusted to reduce the network error by performing the following equation (Rumelhart *et al.*, 1986):

$$\Delta w_{ji}^n(m+1) = \eta \cdot \frac{\partial(E)}{\partial w_{ji}^n} + \alpha \cdot \Delta w_{ji}^n(m) \quad (4)$$

Where:

- $\Delta w_{ji}^n(m+1)$ = The weight increment at the $(m+1)^{\text{th}}$ iteration (Epoch)
- η = The learning rate

This process is continued until the allowable network error is occurred.

Radial basis function neural network: The RBF network is a feed forward network consists of one hidden layer. The activation function of the hidden layer is often a Gaussian function and the output layer has often linear functions (Dibike *et al.*, 1999; Mason *et al.*, 1996). Learning of a RBF network is generally divided into two phases. The first one is an unsupervised (self-organizing) in which the hidden unit parameters depend on the input distribution are adjusted. The second one is a supervised learning in which the weights between hidden and linear output layer are adjusted using gradient descent or linear regression techniques.

A RBF hidden neuron has one parameter associate with each input neuron. These parameters, w_{ij} , are the centers of units. The output of each hidden neuron is a function of the distance between the input vector $X = (x_1, x_2, \dots, x_n)$ and radial center vector $W_j = (w_{1j}, w_{2j}, \dots, w_{nj})$ and can be written as follows:

$$\delta_j = \sqrt{\sum_{i=1}^n (x_i - w_{ij})^2} \quad (5)$$

The hidden neuron output can be calculated in many ways. The most popular activation function for this purpose is the Gaussian one described as follows (Mason *et al.*, 1996):

$$f(\delta_j) = \exp(-\lambda \cdot \delta_j^2) \quad (6)$$

where, λ is a constant. At last the output layer computes the output as follows:

$$Z_k = (\sum_{j=1}^J b_{jk} \cdot y_j) / (\sum_{j=1}^J y_j) \quad (7)$$

Where:

- b_{jk} = The weight coefficient between j^{th} neuron between hidden layer and k^{th} neuron of output layer
- y_j = The output of the j^{th} hidden neuron

ANN modeling assumptions: The performance of rainfall-runoff black box modeling is highly dependent on the type and number of input elements. In this regard, several models indicating different number of input elements are evaluated as follows:

M1: $Q(t) = f [P_i(t)], \quad i = 1, 2, \dots, 11$

M2: $Q(t) = f [Q(t-1), Q(t-2)],$

M3: $Q(t) = f [Q(t-1), P_i(t)] \quad i = 1, 2, \dots, 11$

Where:

$Q(t)$ = The discharge

$P_i(t)$ = The precipitation at i^{th} station at the t^{th} time step

Many researchers have reported the importance of careful data preparation and the choice of effective input variables (Dawson and Wilby, 1998). So, the aforementioned three models (M1, ..., M3) are evaluated using the MLP and RBF neural networks.

Lumped conceptual rainfall-runoff model: The basis and general concepts that govern this model is based on the NAM (Nedbør Affstrømnings Model) model (Nielsen and Hansen, 1973). In this model, the quantitative simulation of watershed is based on hydrologic cycle and its parameters represent an average value for the whole watershed. Figure 3 shows the general structure of the model with its four different and mutually interrelated storages and their corresponding flows. The four storage layers are snow storage, surface storage, lower zone storage and underground storage and the three flows are surface flow (QOF), interflow (QIF) and underground flow (QBF). Generally, the input data needed for this model are rainfall, potential evaporation and temperature and the output of the model is the watershed outflow with respect to time.

In this study, Shuffled Complex Evolution (SCE) methodology is implemented as the NAM automatic calibration tool. To obtain the global optimization point, this method uses four basic concepts. These concepts are combination of random and deterministic approaches, clustering, the concept of a systematic evolution of a complex of point spanning the space, the concept of competitive evolution (Holland, 1975). The success of each of the above concepts in separate applications has been proved (Sorooshian *et al.*, 1993; Duan *et al.*, 1992; Holland, 1975; Wang, 1991). The combination of these concepts made the SCE known as a powerful, effective and flexible method. A detailed application of SCE in NAM model auto calibration for Karkheh River Basin in west of Iran is reported by Eslami *et al.* (2005). A general description of the steps of the SCE-UA method is given below (Duan *et al.*, 1994):

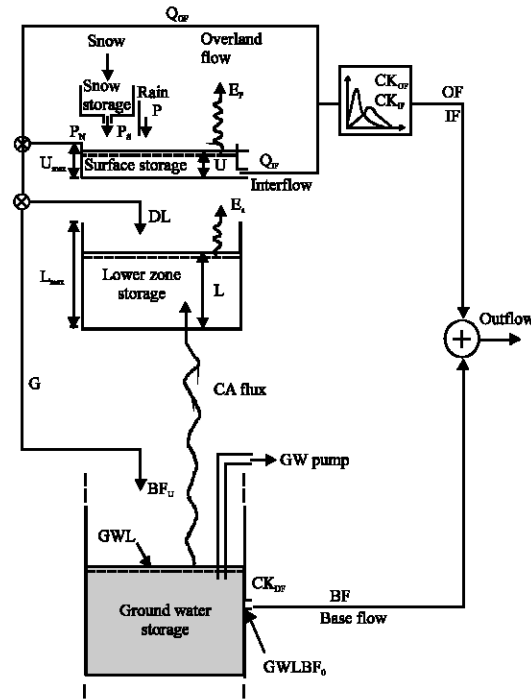


Fig. 3: Schematic structure of the NAM model

Step 1: Generate sample: Sample s points randomly in the feasible parameter space and compute the criterion value at each point. In the absence of prior information on the approximate location of the global optimum, use a uniform probability distribution to generate a sample.

Step 2: Rank points: Sort the s points in order of increasing criterion value so that the first point represents the smallest criterion value and the last point represents the largest criterion value (assuming that the goal is to minimize the criterion value).

Step 3: Partition into complexes: Partition the s points into p complexes, each containing m points. The complexes are partitioned such that the first complex contains every $p(k-1) + 1$ ranked point, the second complex contains every $p(k-1) + 2$ ranked point and so on, where, $k = 1, 2, \dots, m$.

Step 4: Evolve each complex: Evolve each complex according to the Competitive Complex Evolution (CCE) algorithm (which is elaborated below).

Step 5: Shuffle complexes: Combine the points in the evolved complexes into a single sample population; sort the sample population in order of increasing criterion value; shuffle (i.e., re-partition) the sample population into p complexes according to the procedure specified in Step 3.

Step 6: Check convergence: If any of the pre-specified convergence criteria are satisfied, stop; otherwise, continue.

Step 7: Check the reduction in the number of complexes: If the minimum number of complexes required in the population, p_{min} , is less than p , remove the complex with the lowest ranked points; set $p = p-1$ and $s = pm$; return to Step 4. If $p_{min} = p$, return to Step 4.

The initial random sampling of the parameter space provides the potential for locating the global optimum without being biased by pre-specified starting points. The partition of the population into several communities facilitates a freer and more extensive exploration of the feasible space in different directions, thereby allowing for the possibility that the problem has more than one region of attraction. The shuffling of communities enhances the survivability by a sharing of the information (about the search space) gained independently by each community. One key component of the SCE-UA method is the CCE algorithm, as mentioned in Step 4. This algorithm is based on the Nelder and Mead (1965) Simplex downhill search scheme.

RESULTS AND DISCUSSION

The training/calibration and verification datasets described in the previous sections are used to learn/calibrate and test the ANNs and NAM models. Many different architectures, including number of hidden layers for the MLP, number of hidden neurons for both MLP and RBF, activation functions for the MLP and number of optimal epochs For the MLP are evaluated during the training and verification phases.

The model performance is evaluated using Mean Relative Error (MRE) and Nash-Sutcliffe between the observed and forecasted discharges. The former function is recommended in case of flood forecasting (Nash and Sutcliffe, 1970) which is computed as follows:

$$NASH = 1 - \left[\frac{\sum_{t=1}^T (Q_o^t - Q_p^t)^2}{\sum_{t=1}^T (Q_o^t - Q_m)^2} \right] \quad (8)$$

Where:

- Q_o and Q_p = Observed and predicted discharge at time t , respectively
- Q_m = Denotes the mean observed discharge

The results are shown in Table 1. As shown in Table 1, the performance of the MLP neural network is generally better than other models for Kashkan River Basin with available data. On the other hand, when antecedent discharges were added to the input pattern, the performance of the ANNs is improved.

Comparing both applied neural networks indicate that the RBF performance is superior to the MLP, when precipitation is the only input parameter (M1 model), whereas river flow forecasting without exogenous inputs (M2 model) is better performed using the MLP neural network. On the other hand, both MLP and RBF neural network forecasts, improved when M3 modeling assumption is mentioned. Comparing applied neural networks to the NAM model shows that the MLP network trained with M3 modeling assumption is the most superior model in both training/calibration and verification periods.

Forecasted discharges by proposed MLP (trained through M3 modeling assumption) and the NAM model, versus observed discharge is plotted for the training/calibration and verification datasets in Fig. 4 and 5, respectively.

Figure 6 and 7 show the observed and forecasted hydrographs by proposed MLP and the NAM models in the verification period.

Integration of plotted points near the line $y = x$ in Fig. 5 and 6 clearly indicates the superiority of the MLP neural network performance in these graphs. In detail, low and high flows are forecasted adequately by the MLP compared to the NAM model.

As Fig. 6 and 7 show, it can be obviously seen that both base flow and peak discharges are more precisely predicted by the MLP neural network. Returning to the Table 1 indicates the effective role of antecedent discharge in the improvement of the MLP forecasts.

Table 1: Performance evaluation of different models

	MLP				RBF				NAM			
	Verification		Training		Verification		Training		Verification		Training	
	MRE	Nash.	MRE	Nash.	MRE	Nash.	MRE	Nash.	MRE	Nash.	MRE	Nash.
M1	291.7	-2.30	110.1	0.44	202.2	0.21	75.0	0.35				
M2	30.2	0.90	12.3	0.75	770.2	-2.80	23.1	0.70	37.2	0.22	23.2	0.71
M3	12.3	0.93	9.0	0.94	49.0	0.41	23.7	0.78				

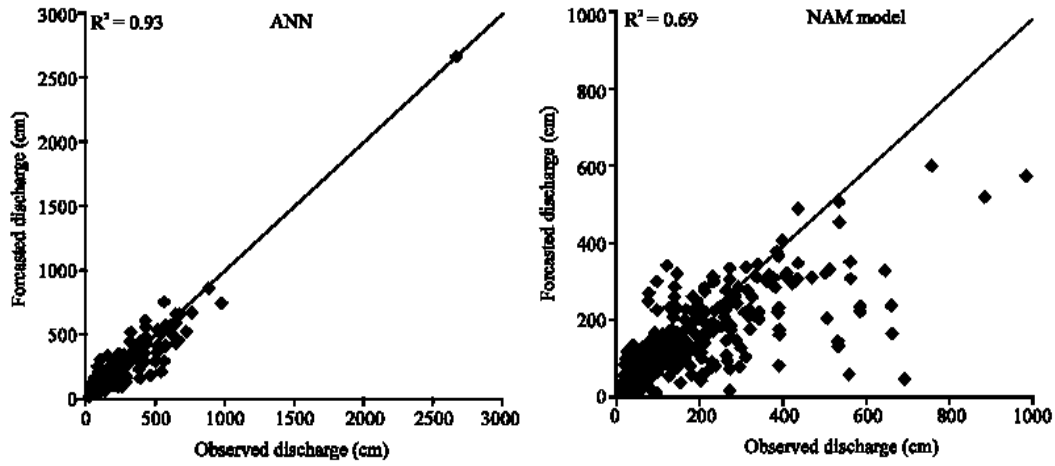


Fig. 4: Forecasted versus observed discharge in the training/calibration period

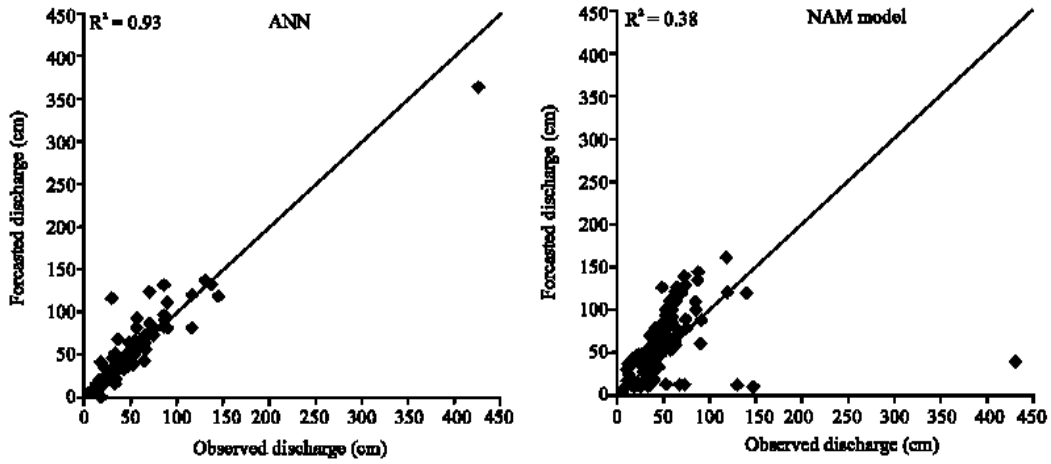


Fig. 5: Forecasted versus observed discharge in the verification period

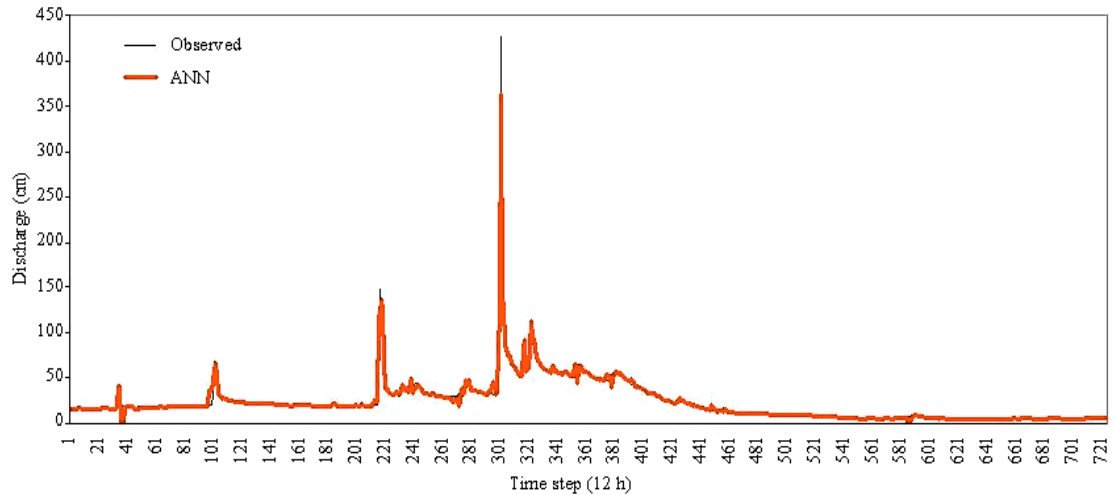


Fig. 6: Observed and forecasted hydrographs in the verification period (MLP)

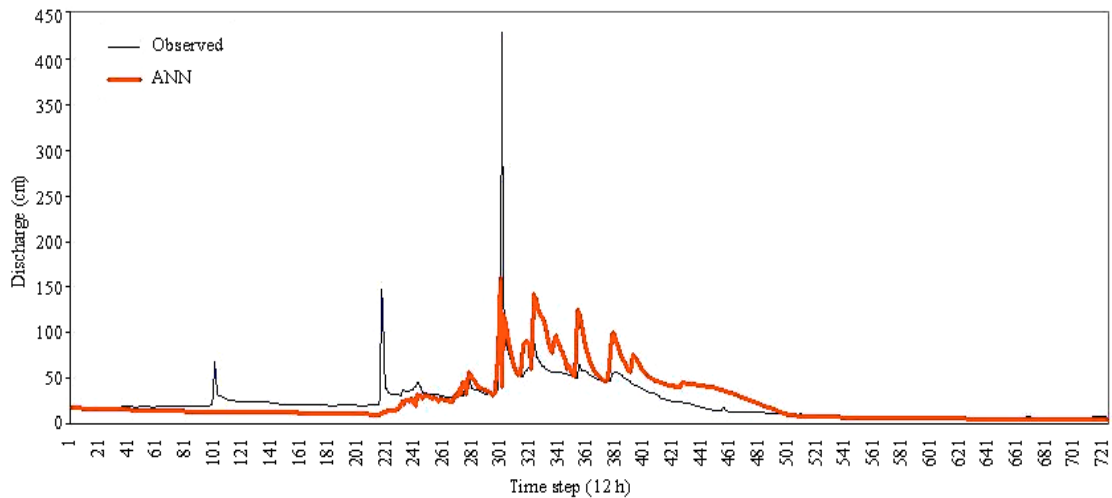


Fig. 7: Observed and forecasted hydrographs in the verification period (NAM)

CONCLUSIONS

Artificial Neural Networks can provide useful tools in analyzing and solving a wide range of problems of water engineering in a simpler structure and faster training rather than conceptual models. ANNs are robust and flexible tools, trained by example illustration and act as a distributed memory to save the inner knowledge of training data.

Multi Layer Perceptron neural network applied to Kashkan River flow forecasting shows encouraging performance compared to Radial Basis Function neural network and NAM hydrologic model with SCE algorithm for auto calibration in case of peak flood and base flow forecasting. Comparing both applied neural networks in Kashkan river flow forecasting indicate that the RBF performance is slightly superior to the MLP, when precipitation is the only input parameter (M1 model), whereas river flow forecasting without exogenous inputs (M2 model) is better performed using the MLP neural network.

Furthermore, it is concluded that the presence of the discharge in the previous time step as model input can greatly improve the ANN forecasts.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Water Research Institute for their support and consent.

REFERENCES

Babovic, V. and V. Bojcow, 2001. Runoff modeling with genetic programming and artificial neural networks. D2K Technical Report D2K TR 0401-1.

- Campolo, M., P. Andreussi and A. Soldati, 1999. River flow forecasting with a neural network model. *Water Resour. Res.*, 35: 1191-1197.
- Campolo, M., A. Soldati and P. Andreussi, 2003. Artificial neural network approach to flood forecasting in the River Arno. *Hydrol. Sci. J.*, 48: 381-398.
- Chang, F. and Y. Chen, 2001. A counter propagation fuzzy-neural network modeling approach to real time streamflow prediction. *J. Hydrol.*, 245: 153-164.
- Chau, K.W., C.L. Wu and Y.S. Li, 2005. Comparison of several flood forecasting models in Yangtze river. *J. Hydrol. Eng.*, 10: 485-491.
- Chau, K.W., 2006. Particle swarm optimization training algorithm for ANNs in stage prediction of shing Mun river. *J. Hydrol.*, 329: 363-367.
- Dawson, C.W. and R. Wilby, 1998. An artificial neural network approach to rainfall-runoff modeling. *Hydrol. Sci. J.*, 43: 47-66.
- Dawson, C.W., R.J. Abrahart, A.Y. Shamseldin and R.L. Wilby, 2006. Flood estimation at ungauged sites using artificial neural networks. *J. Hydrol.*, 319: 391-409.
- Dibike, Y.B., D.P. Solomatine and M.B. Abbot, 1999. On the encapsulation of numerical-hydraulic models in artificial neural networks. *J. Hydraul. Res.*, 37: 147-161.
- Duan, Q., S. Sorooshian and H.V. Gupta, 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour. Res.*, 28: 1015-1031.
- Duan, Q., S. Sorooshian and H.V. Gupta, 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. *J. Hydrol.*, 158: 265-284.

- Eslami, H.R., K. Qaderi and J.M.V. Samani, 2005. Auto-calibration of flood forecasting using shuffled complex evolution (SCE) method. In: Proceedings of the International Conference Innovation, Advances and Implementation of Flood Forecasting Technology, Tromso, Norway. Publisher: HR Wallingford, Oxford Shire OX10 8BA, UK.
- French, M.N., W.F. Krajewski and R.R. Cuykendall, 1992. Rainfall forecasting in space and time using a neural network. *J. Hydrol.*, 137: 1-31.
- Gautam, M.R., K. Watanabe and H. Saegusa, 2000. Runoff analysis in humid forest catchment with artificial neural network. *J. Hydrol.*, 235: 117-136.
- Gorzalczany, M.B., 2002. Computational Intelligence Systems and Applications. Physica-Verlag Company, Heidelberg, Germany.
- Hecht-Nielsen, R., 1987. Kolmogorov's Mapping Neural Network Existence Theorem. 1st IEEE ICNN, Vol. 3. San Diego, CA.
- Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann. Arbor.
- Hsu, K.L., H.V. Gupta and S. Sorooshian, 1995. Artificial neural network modeling of the rainfall-runoff process. *Water Resour. Res.*, 31: 2517-2530.
- Imrie, C.E., S. Durucan and A. Korre, 2000. River flow prediction using artificial neural networks: Generalisation beyond the calibration range. *J. Hydrol.*, 233: 138-153.
- Jain, A., K.P. Sudheer and S. Srinivasulu, 2004. Identification of physical processes inherent in artificial neural network rainfall runoff models. *Hydrol. Process*, 18: 571-581.
- Karayiannis, N.B. and A.N. Venetsanopoulos, 1993. Artificial Neural Networks: Learning Algorithms, Performance Evaluation and Applications. Kluwer Academic Publishers, Boston.
- Mason, J.C., R.K. Price and Tem'ne, 1996. A neural network model of rainfall-runoff using radial basis functions. *J. Hydraul. Res.*, 34: 537-548.
- Minns, A.W. and M.J. Hall, 1996. Artificial neural networks as rainfall-runoff models. *Hydrol. Sci. J.*, 41: 399-417.
- Nash, J.E. and J.V. Sutcliffe, 1970. River flow forecasting through conceptual models. Part I. A discussion of principles. *J. Hydrol.*, 10: 282-290.
- Nelder, J.A. and R. Mead, 1965. A simplex method for function minimization. *Comput. J.*, 7: 308-313.
- Nielsen, S.A. and E. Hansen, 1973. Numerical simulation of the rainfall runoff process on a daily basis. *Nordic. Hydrol.*, 4: 171-190.
- Pang, B., S. Guo, L. Xiong and C. Li, 2007. A nonlinear perturbation model based on artificial neural network. *J. Hydrol.*, 333: 504-516.
- Rajurka, M.P., U.C. Kothiyari and U.C. Chaube, 2004. Modelling of the daily rainfall-runoff relationship with artificial neural network. *J. Hydrol.*, 285: 96-113.
- Rumelhart, D.E., J.L. McClelland and the PDP Research Group, 1986. Parallel Recognition in Modern Computers. In Processing: Explorations in the Microstructure of Cognition. Vol 1. Foundations, MIT Press/Bradford Books, Cambridge Mass.
- Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modelling. *J. Hydrol.*, 199: 272-294.
- Shamseldin, A.Y. and K.M. O'Connor, 2001. A nonlinear neural network technique for updating river flow forecasts. *Hydrol. Earth Syst. Sci.*, 5: 577-597.
- Sivakumar, B., A.W. Jayawardena and T.M.K.G. Fernando, 2002. River flow forecasting: Use of phase-space reconstruction and artificial neural networks approaches. *J. Hydrol.*, 265: 225-245.
- Smith, J. and R.N. Eli, 1995. Neural network models of the rainfall-runoff process. *J. Water Res. Pl.-ASCE.*, 121: 499-508.
- Sorooshian, S., Q. Duan and H.V. Gupta, 1993. Calibration of rainfall-runoff models: Application of global optimization to the sacramento soil moisture accounting model. *Water Resour. Res.*, 29: 1185-1194.
- Wang, Q.J., 1991. The genetic algorithms and its application to calibrating conceptual rainfall-runoff models. *Water Resour. Res.*, 27: 2467-2471.
- Werbos, P.J., 1974. Beyond regression: New tools for prediction and analysis in behaviour sciences. Ph.D. Thesis, Harvard University, Cambridge, Mass.
- Wilby, R.L., R.J. Abrahart and C.W. Dawson, 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrol. Sci. J.*, 48: 163-181.
- Xiong, L. and K.M. O'Connor, 2002. Comparison of four updating models for real-time river flow forecasting. *Hydrol. Sci. J.*, 47: 621-639.
- Xiong, L., K.M. O'Connor and S.L. Guo, 2004. Comparison of three updating schemes using artificial neural network in flow forecasting. *Hydrol. Earth Syst. Sci.*, 8: 247-255.
- Zhu, M.L., M. Fujita and N. Hashimoto, 1994. Application of Neural Networks to Runoff Prediction. In: Stochastic and Statistical Methods in Hydrology and Environmental Engineering, Hipel, K.W. *et al.* (Eds.). Kluwer, Dordrecht, The Netherlands, pp: 205-216.