Adaptive Neural Networks for Flood Routing in River Systems

Saman Razavi a & Mohammad Karamouz b

a School of Civil and Environmental Engineering, Amirkabir University (Tehran Polytechnic), Tehran, Iran
b School of Civil Engineering, Center of Excellence for Infrastructure Engineering & Management, University of Tehran, Tehran, Iran

Available online: 22 Jan 2009
Adaptive Neural Networks for Flood Routing in River Systems

Saman Razavi, School of Civil and Environmental Engineering, Amirkabir University (Tehran Polytechnic), Tehran, Iran and Mohammad Karamouz, School of Civil Engineering, Center of Excellence for Infrastructure Engineering & Management, University of Tehran, Tehran, Iran

Abstract: A methodology based on adaptive ANN models is proposed for flood routing in river systems. The proposed methodology is capable of modeling both converging and diverging river networks. A Multilayer Perceptron Network (MLP), a Recurrent Neural Network (RNN), a Time Delay Neural Network (TDNN) and a Time Delay Recurrent Neural Network (TDRNN) are applied in this study. An Adaptive training procedure based on the Forgetting Factor (FF) approach is used to train ANNs models. The methodology provides a lead time equal to travel time for the flood estimation downstream of the river. The performances of the models are tested within the two distinctive parts of the Karoon River in south-west Iran. The first case study uses synthetic floods generated by the HEC-RAS hydraulic model; the second one uses observed floods. Besides, the Muskingum routing method is used in the second case study to be compared with the results of ANN models. Overall, the results demonstrated that the proposed methodology performs well considering goodness-of-fit criteria. Moreover, the dynamic neural networks outperform the static MLP and the Muskingum model.

Keywords: Flood Routing, Neural Network, Adaptive training, Forgetting Factor

Introduction

Flood disasters occur in many regions around the world and cause many casualties and much property damage. Mitigating the effects of floods can be accomplished using either structural or non-structural systems, or by employing a combination of both. World Bank Policy Paper on Water Resources Management (World Bank, 1993) proposed that: “Governments should intensify efforts to achieve flood control with non-structural measures that are less costly, yet no less effective in preventing disasters, than more expensive structural measures”. Non-structural measures include the flood proofing of properties, land-use regulation within flood plains and flood warning systems.

In order to perform the prior estimation of hydrological phenomena in real time, and to warn general public in the emergency situations, Hydrological Forecasting Systems (HFSs) are developed. Nemec (1986) has provided a classification of models for the purpose of hydrological forecasting in which two main categories are presented as the subsets of purely deterministic forecasting models:

1. Rainfall-runoff models
2. Hydrometric data-based models (involving only streamflow data)
Models in each category vary according to their data requirements, their computational resources and, subsequently, to the type of problem at hand. When the basin response time is short, which is the case in small and medium-sized basins, rainfall-runoff models are probably the preferred techniques. For large river basins, which have a long travel time from an upstream to a downstream gauging station, routing techniques (hydrometric data-based) are appropriate alternatives. Due to longer travel time, they can provide sufficient time to forecast. In the present study, the emphasis is placed on the second category, which is referred to as flood routing.

Two classic approaches, “hydrologic routing” and “hydraulic routing,” are generally employed to route the flood wave in the natural channels. Hydrologic routing is based on the storage-continuity equation, while, in hydraulic routing, the Saint-Venant equations governing the phenomenon are solved numerically. In addition to these process-based models, there are several data-driven models, such as artificial neural networks, which could be accurate predictive tools among the nonlinear flood forecasting methods.

Overall, ANNs have proven to be an efficient alternative to traditional methods for modeling qualitative and quantitative water resource variables (Hsu et al. 1995; Maier and Dandy 1996; Shamseldin 1997; Karamouz et al. 2004; Karamouz et al. 2005). Most of ANN’s applications in hydrology have used the static feedforward neural network (Coulibaly et al. 1999); while, in most of hydrological problems, the governing equations are temporal and the relations between input and output sets are changed dynamically. As a result, the present study places an emphasis on the use of dynamic neural networks with an adaptive training procedure in order to involve the dynamic behavior of the problem.

In this paper, an investigation on applying three types of dynamic neural networks for flood routing is presented. Tapped delay lines (TDLs), recurrent (feedback) connections and the hybrid method are different approaches used to design the dynamic neural networks, including an Elman recurrent neural network (Elman, 1990), a time delay neural network (TDNN) and a time delay recurrent neural network (TDRNN). These networks are applied to present predictive models, which are able to estimate the downstream flow with a lead-time using the past values of upstream and downstream flows in river systems. A static MLP network is also used for comparison purposes in the same problem. The adaptive training based on Forgetting Factor (FF) approach (Ljung 1999; Karamouz et al. 2007) is used to train the ANNs-based models. Two distinctive parts of the Karoon river network are considered as the case studies. The first case study uses synthetic data generated by the HEC-RAS hydraulic model, while the second uses observed data. Besides, the Muskingum hydrologic routing model is used in the second case study in order to be compared with the ANN-based models.

**Artificial Neural Networks**

The architectures of ANNs are adapted from biological neural networks, which can recognize patterns and learn from their interaction with the environment. Most of ANN’s applications in hydrology have used feedforward neural network, particularly the standard multilayer perceptron (MLP) trained with the backpropagation algorithm (Coulibaly et al. 1999). MLP...
is a static network in which information is transmitted through the connections between its neurons forward only and network has no feedback or memory, which covers its initial and past states. Figure 1 (without recurrent connections) shows a typical three layer MLP, which, mathematically, can be expressed as equations 1 and 2.

\[ a_j^1(t) = f \left( \sum_{i=1}^{R} w_{j,i}^1 p_i(t) + b_j^1 \right) \]

\[ 1 \leq j \leq S_1 \]

\[ a_k^2(t) = g \left( \sum_{j=1}^{S_1} w_{k,j}^2 a_j^1(t) + b_k^2 \right) \]

\[ 1 \leq k \leq S_2 \]

where \( t \) denotes a discrete time, \( R \), \( S_1 \) and \( S_2 \) are the numbers of input, hidden and output neurons, respectively. \( w^1 \) and \( w^2 \) are the weight matrices of hidden and output layers, respectively. \( p \) is the input vector of the network and \( a^1 \) and \( a^2 \) are the output vectors of the hidden and output layers, respectively. \( f \) and \( g \) are the activation functions of the hidden and output layers, as well.

The MLP model does not perform temporal processing and the input vector space does not consider the temporal relationship of the inputs. As a result, it often yields sub-optimal solutions. In many tasks, the input pattern comprises one or more temporal signals, as in speech recognition, time series prediction, and signal filtering (Bose and Liang 1996). When trying to involve dynamic behavior, recurrent (feedback) connections, tapped delay lines (TDLs) and the hybrid method are different approaches, all of which are used to design temporal neural networks.

The recurrent scheme differs from the MLP, in the way that the outputs of hidden or output layers recur to a context unit and, after one time step, return to the input layer. This network is called recurrent neural network (RNN), which represents time implicitly by its effects on processing. Depending on the architecture of the recurrent (feedback) connections, there are some general models of RNN (Jordan 1986; Elman 1990; Frasconi et al. 1992). In this study, the Elman RNN, shown in Figure 1 (with recurrent connections), is selected and formulated by equations 3 and 4.

\[ a_j^1(t) = f \left( \sum_{i=1}^{R} w_{j,i}^1 p_i(t) + \sum_{c=1}^{S_1} w_{c,j}^1 a_c^1(t-1) + b_j^1 \right) \]

\[ 1 \leq j \leq S_1 \]

\[ a_k^2(t) = g \left( \sum_{j=1}^{S_1} w_{k,j}^2 a_j^1(t) + b_k^2 \right) \]

\[ 1 \leq k \leq S_2 \]

where \( w^c \) is the weight matrix of the context unit. The other parameters and variables are the same as the MLP network as stated in equations 1 and 2.

Another efficient alternative for involving dynamic behavior is the conversion of the temporal signals at the input into spatial patterns. This is accomplished by using tapped delay lines. This network, which is called IDNN (Input-Delayed Neural Network), consists of a static network (in this case a MLP), which some tapped delay lines are attached to its input layer (Coulibaly et al. 2001). A further generalization over the IDNN is obtained by also replacing the internal connection weights in the network by tapped delay lines (Waibel 1989; Atiya and Parlos 1992; Karamouz 2007). The resulting network, which can represent time explicitly, is sometimes called Time Delay Neural Network (TDNN) or spatio-temporal network. A three layer TDNN is shown in Figure 2 (without recurrent connections) and is described by equations 5 and 6.

\[ a_{j,0}^1(t) = f \left( \sum_{d_i=0}^{D_1} \sum_{i=1}^{R} w_{j,i,d_i}^1 p_i, d_{i-1} (t) + b_j^1 \right) \]

\[ 1 \leq j \leq S_1 \]

\[ a_{k}^2(t) = g \left( \sum_{d_i=0}^{D_1} \sum_{j=1}^{S_1} w_{k,j,d_i}^2 a_{j,d_i}^1 (t) + b_k^2 \right) \]
where $D_1$ and $D_2$ are the memory lengths (TDL orders) of the input and hidden layers, respectively. The other parameters and variables are the same as the MLP network, as stated in equations 1 and 2.

If Elman recurrent connections are added to the structure of a time delay neural network (TDNN), the resulting network is a Time Delay Recurrent Neural Network (TDRNN). The main feature of this network is that the hidden layer receives the contents of both the input time delays and the context unit. This makes it appropriate for complex sequential input learning. Coulibaly et al. (2001) used this approach to forecast multivariate reservoir inflow and compared it with the Elman RNN and the IDNN networks’ results. Their TDRNN only uses tapped delay lines in its input layer, which also referred to as the input-delayed recurrent neural network (IDRNN). Coulibaly et al. (2001) showed that their TDRNN performs better in accounting with long-term memory of time series. Karamouz et al. (2007) developed a TDRNN for long-lead rainfall forecasting using climate signals as the predictors and compared its performance with the statistical ARMAX model. They reported that the TDRNN showed significant improvement in forecast resolution over the ARMAX model. The architecture of the TDRNN is shown in Figure 2 (with recurrent connections) and it is described by equations 7 and 8.

$$a_{j,0}^{1}(t) = f \left( \sum_{d_i=0}^{D_1} \sum_{i=1}^{S_1} w^1_{j,i,d_i} p_{i,d_i} + \sum_{c=1}^{S_1} w^1_{j,c} a_{c}^{1}(t-1) + b_{j}^{1} \right)$$

$$1 \leq j \leq S_1$$

$$a_{k}^{2}(t) = g \left( \sum_{d_2=0}^{D_2} \sum_{j=1}^{S_1} w^2_{k,j,d_2} a_{j}^{1}(t) + b_{k}^{2} \right)$$

$$1 \leq k \leq S_2$$

where $w^c$ is the weight matrix of the context unit, and $D_1$ and $D_2$ are the memory lengths (TDL orders) of the input and hidden layers, respectively. The other parameters and variables are the same as the MLP network as stated in equations 1 and 2.

**Methodology of Flood Routing**

The objective of this study is to present a methodology for flood routing in river networks based on the past values of upstream and downstream flows. The proposed models will be able to estimate the downstream flow with the lead time equal to the travel time between the upstream and downstream gauging stations. This methodology is suitable for both single inflow and multiple inflow networks. It can also be used for multiple outflow networks.

**Networks’ Architectures**

Selecting the suitable architecture of neural networks for a specific problem is important, because...
it directly affects its complexity and its generalization capability. Hornik et al. (1989) proved that a three-layer network, with the sigmoid activation function in the hidden layer and the linear function in the output layer, is capable of approximating the complex relationship between any input-output sets, subject to appropriate selection of the number of hidden neurons. Thus, three layers with the hyperbolic tangent sigmoid function in the hidden layer and the linear function in the output layer are considered for all networks. But, there is no established methodology for selecting the appropriate number of hidden neurons prior to training (Hsu et al. 1995, Coulibaly et al. 2001). To find the best number of hidden neurons of the network, several networks, differing only by their number of hidden neurons, are tested during the calibration process. The preference is to work with not only the most efficient, but also the least complex networks.

Length of Networks’ Memories

In the networks with tapped delay lines, the TDNN and the TDRNN, the length of TDLs, as well as their location, should be determined. Their length should be adapted considering the characteristic and nature of the problem and the input-output data. Thus, a sensitivity analysis should be performed in order to determine the number of time delay operators (the length of TDLs) in each layer of the networks with TDLs during the calibration process.

Adaptive Training

In many cases, it may be necessary to estimate a model online, such as adaptive prediction, at the same time as the new input-output data is received. The adaptive (online) training updates the weights and biases of the network at each time step after passing through the entire input-output set from the beginning of modeling time up to the current time. The major feature of the adaptive training is the fact that there is no need for a considerable amount of data and the only essential data is the most recently observed data before the time of forecast. The adaptive training is able to re-calibrate the model online, at each time step, as soon as new observations become available. This adaptability enables the model to capture the characteristics of the current situation of an ongoing event (Toth et al. 2000).

The approach for the adaptive training in this study is to discount old measurements exponentially, so that an observation that is \( \tau \) samples old carries a weight \( \lambda^\tau \) that is of the weight of the most recent observation. Thus, the objective of training in each time step is to minimize the following function:

\[
F(t) = \sum_{n=1}^{t} \lambda^{t-n}e(n)
\]

where, \( F \) is the performance function, \( t \) denotes the current time, \( e \) is the error function and \( \lambda \) is a positive number (slightly) less than 1. \( \tau = 1/(1-\lambda) \) can be considered as the memory horizon of the approach. Data that is older than \( \tau = 1/(1-\lambda) \) carry a weight that is less than about 0.3. The optimal value of \( \lambda \) depends on the memory length of the relationships between input and output sets. It can be determined by sensitivity analysis during calibration procedure. Typical values of \( \lambda \) are in the range 0.97–0.995. This algorithm is called the Forgetting Factor (FF) approach to adaptation (Ljung 1999). Furthermore, the Sum Squared Error (SSE) is considered as the error function, \( e \), in this study. This approach has been used by Karamouz et al. (2007) in training of a time delay recurrent neural network (TDRNN) for long-lead rainfall forecast.

The illustrated adaptive training procedure can be applied to both static and dynamic networks, although it is more appropriate for the latter. For modifying the weights in order to minimize the performance function (equation 9) at each time step, the well-known Backpropagation algorithm (BPA) is used due to its high efficiency in supervised training. Detailed description of BPA is available in many texts, such as Bose and Liang (1996).

In order to improve generalization capability, and to avoid overtraining in the adaptive training, a parameter named maximum performance function is defined, which stands for the maximum acceptable limit of the performance function. In this way, at the beginning
of each time step, the network’s output is calculated and saved as this time step estimation, and then the weights are adjusted frequently using BP algorithm. This continues until the performance function becomes smaller than the maximum performance function. If this parameter is selected as a relatively large number, networks cannot properly learn the current state of the system. In fact, before networks can learn enough, the condition of maximum performance function is satisfied and algorithm proceeds to estimate the next time step variables. On the contrary, the smaller this parameter is, the more computational time has to be spent. Over training may also occur. The optimal value of the maximum performance function depends on the case and can be determined by sensitivity analysis during calibration.

Furthermore, all the networks are initialized with random weights and, in order to obtain the results, which are less sensitive to the initial conditions, several distinct runs are performed for each model. The final results are obtained by averaging the 3 best results.

**Flood Routing**

The minimum lead time of the flood estimation is limited to the simulation time interval ($\Delta t$). The lead time can be selected equal to a multiple of the time interval. The maximum of the lead time is limited to the travel time in the routing reaches, $k$, as well. As indicated before, this methodology is based on the correlations between flow values at the upstream and downstream of the river networks. The time interval and the lead time should thus be selected as they can provide reasonable correlations between upstream and downstream data. Needless to say, the maximum correlation nearly exists between upstream data at time $t$ and downstream data at time $t+k$. Since there may be several inflows (tributaries) to the river network with different travel times, the lead time should be selected equal to the least travel time.

If it is assumed that $R$ independent variables (inflows) are affecting the downstream flow, then:

$$f\left(\text{Upstream Variable}_1(t-lag_1), \text{Upstream Variable}_2(t-lag_2), \ldots, \text{Upstream Variable}_R(t-lag_R)\right)$$

$$\begin{align*}
lag_1 &= k_1 - L \\
lag_2 &= k_2 - L \\
\vdots \\
lag_R &= k_R - L \\
\end{align*}$$

where,

$k_1$, $k_2$, $\ldots$ and $k_R$ are the travel times of flows 1, 2 and $R$, respectively.

$L$: lead time

$lag_1$, $lag_2$, $\ldots$ and $lag_R$ are the lag times of upstream flow variables 1, 2, $\ldots$ and $R$, respectively.

Since the least travel time is considered as the lead time, $L$, the lag time of the inflow with the least travel time is equal to zero. Figure 3 demonstrates the moving window (timing) of input-output set during the routing procedure.

If the current time is $t$, for estimating the downstream flow at time $t+L$:

1) The network should be trained with the observations up to the time $t$ (including the time $t$) according to the illustrated adaptive training algorithm. Thus, the last weights update is done after simulation of

![Figure 3- Moving window of input-output data](image-url)
Figure 4- (a) The location Map of Karoon river basin (b) Case study 1 (c) Case study 2
river system at time \( t \). After this step the network’s weights will not change until new observations at time \( t + \Delta t \) (\( \Delta t \) is the time interval) become available.

2) In this step the network begins to estimate the downstream flows at times \( t + \Delta t \), \( t + 2\Delta t \), \( \ldots \) and \( t + n\Delta t \) where \( n\Delta t = L \).

3) Wait till time proceeds to the next time step \( (t + \Delta t) \). Now, the observations of the time \( t + \Delta t \) become available. Thus, the time \( t + \Delta t \) is considered as the current time and the procedure returns to step 1 in order to estimate the downstream flows at time \( t + \Delta t + L \).

In this study, four types of criteria are used to describe various goodness-of-fit of the models. The coefficient of efficiency (CE), the peak-weighted root mean square error (PWRMSE), the mean error of time to peak and the volume error (the average ratio of estimated flood peaks minus observed to the observed flood peak) are used to evaluate the models’ performance. CE is a correlation-based measure that ranges from minus infinity to plus one, with higher values indicating better agreement. In the PWRMSE, the weight assigned to each ordinate is proportional to the magnitude of the ordinate (USACE, 2000). The CE and the PWRMSE can be formulated as:

\[
CE = 1 - \frac{1}{T} \sum_{t=1}^{T} \frac{(obs_t - est_t)^2}{\left( \frac{1}{T} \sum_{t=1}^{T} (obs_t - obs_{mean})^2 \right)^{1/2}}
\]

(11)

\[
P \ W \ R \ M \ S \ E = \frac{1}{T} \left[ \sum_{t=1}^{T} (obs_t - est_t)^2 \left( \frac{obs_t + obs_{mean}}{2obs_{mean}} \right) \right]^{1/2}
\]

(12)

where, \( obs_t \) and \( est_t \) represent the observed and estimated values, respectively, \( obs_{mean} \) represents the mean of observed values, and \( T \) is the number of input-output sets.

**HEC-RAS hydraulic model**

The HEC-RAS software supports steady and unsteady flow water surface profile calculations and, in terms of unsteady flow, provides an implicit solution for saint-venant equations based on finite difference numerical method (USACE, 2002). Since there is a lack of observed inflow-outflow data in the first case study, the synthetic data generated by the HEC-RAS hydraulic model is used. As a matter of fact, the output of HEC-RAS is considered as the actual flow in the absence of enough measurements for the first case study. The objective is to validate the flexible ANNs-based models in different and critical conditions as an alternative tool for the data-intensive HEC-RAS model. This procedure is further explained in case study-Part one section.

**Case Studies**

The Karoon drainage basin is located in the southwestern part of Iran, carrying more than one fifth of surface water supply of the country. Figure 4(a) shows the location map of the Karoon river basin. The total length of the Karoon River is about 890 Kilometers, with a catchment area of about 42800 square Kilometers. As shown in Figure 4(a), two distinctive parts of the Karoon River are selected as the case studies. A full description of each case study and results are presented in the two following sections.

**Case Study-Part one**

The first case study is a reach of the Karoon river, 51 km long, located between Ahvaz and Mollasani gauging stations, at the downstream and upstream bounds, respectively (Figure 4(b)). Since there is a lack of inflow-outflow data in this River system in a torrential situation, this case study is based on the synthetic data generated by the HEC-RAS hydraulic model. In order to generate data in this case study, first the HEC-RAS model is calibrated using existing inflow-outflow data to perform unsteady flow simulation in this river reach. Then thirty synthetic flood hydrographs with the different values of peak and base time for Mollasani station are generated according to the SCS dimensionless unit hydrograph, considering the watershed characteristics. Some hydrographs are
produced deliberately with steeper slopes (in comparison with the natural floods) in rising and falling limbs, in order to place a burden on the system to react. These thirty series of data are arranged back to back in order to obtain 2870 hours of flood data at Mollasani station. Notwithstanding the fact that the sequential occurrence of floods rarely happens naturally, the major feature of applying this inflow-outflow data is to put the models in a more intense mode. The resulting hydrographs are routed through the river by the HEC-RAS in order to obtain the corresponding downstream hydrograph at the Ahvaz Station. The obtained upstream and downstream boundary conditions, shown in Figure 5, are considered as the actual data and are given to the ANN routing models during online training process.

For calibration of the models, the first 686 hours of generated flood data (ten first floods) are used, and the remaining 2184 hours of flood data (twenty remaining floods) are used to test the performance of the models.

Since the streamflow at the hydrometric stations in this region is gauged every two hours, the time interval for the modeling is set as two hours. Besides, the estimated travel time between Mollasani and Ahvaz stations is about 8 hours which is the quadruple of the time interval. This time could be of a significant value in flood warning and it would be the lead-time of the models’ forecasts.

Results and Discussion (part one)

Table 1 shows the most appropriate architecture of the networks in which \( R, S_1 \) and \( S_2 \) represent the numbers of input, hidden and output neurons, respectively. To determine the appropriate values of the maximum performance function and the forgetting factor, various values are tested during calibration. For instance, Figures 6(a) and 6(b) show the variation of the PWRMSE versus the maximum performance function and the forgetting factor, respectively, for the TDNN during the calibration process. As shown in Figure 6(a), for the high values of the parameter max. perf. Func. (the range of \( 1 \times 10^{-4} \)), the accuracy is low and as the parameter gets smaller, the accuracy increases. With the parameters smaller than \( 1 \times 10^{-8} \) up to \( 1 \times 10^{-12} \), the accuracy remains the same and, after that, the accuracy decreases. In terms of the forgetting factor, it is worth noting that if this parameter is one, the training procedure with forgetting factor is changed to conventional training procedure. So, the corresponding accuracy of \( \lambda = 1 \) (in Figure 6(b)) is the same as the results of the conventional training. As it is demonstrated in Figure 6(b), a decrease in \( \lambda \) to about 0.995 leads to a considerable increase in the modeling accuracy. Then, the accuracy increases marginally with some fluctuations, and after the value of about 0.974, the accuracy begins to decrease. It means that, in this situation, the model is not able to consider all contributing information of the time series. As a result, \( 1 \times 10^{-8} \) and 0.974 seem to be proper values for the maximum performance function and the forgetting factor, respectively. The same values of \( 1 \times 10^{-8} \) and 0.974 are applied for all other networks, as well.
All four types of mentioned goodness-of-fit criteria are used for calibration and testing results, as summarized in Table 2. As demonstrated there, all dynamic networks result in the better estimation in comparison with the static MLP network. According to the PWRMSE statistic, the TDRNN, the RNN and the TDNN could improve the forecast accuracy by 18%, 23% and 35%, respectively, relative to the MLP in the testing period. The TDNN performs more accurately than the other models. The PWRMSE, the CE, the mean error of time to peak and the volume error of highest peaks resulted by the TDNN in testing period are 182.3, 0.962, 1.4 and 10.7, respectively. Moreover, the improvement of the TDNN over the MLP is about 54% according to the volume error of the flood peaks statistic. The RNN and the TDRNN result equally well in this case study. In terms of computational time and number of weights, the TDNN with 55 weights, as stated in Table 1, is the most complex network computationally, whereas it is the most reliable alternative compared to the other models.

Furthermore, Figure 7 shows the best streamflow estimation with the lead-time of eight hours resulted by the TDNN during the calibration and testing periods. Figure 8 presents the scatter plot of the eight-hour-ahead estimation of the TDNN during the testing period, as well. As shown in Figure 7, at the first time steps the network outputs is rather far from the outflow values obtained from the HEC-RAS model and fluctuate around them. But, after some time steps, the network outputs approach the HEC-RAS outflows and the model accuracy increases. Using the adaptive training and choosing the initial weights randomly are the main reasons for this problem and it may take some time steps before the networks’ weights converge to the optimal values. Moreover, as shown in Figure 8, the flow estimations of the TDNN model are close to the 45° line indicating a good match.

**Case study- Part two**

In the second case study, the application of the proposed models for multiple inflows routing (estimation with a lead time) was carried out using the inflow-outflow data of a channel network of the Karoon drainage basin shown in Figure 4(c). Armand and Poleshaloo gauging stations both measure the inflow

<table>
<thead>
<tr>
<th>Model</th>
<th>Architecture (S_1-S_2)</th>
<th>(R)-Memory order of Input Layer</th>
<th>Memory order of Hidden Layer</th>
<th>Number of Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>1-6-1</td>
<td>------</td>
<td>------</td>
<td>19</td>
</tr>
<tr>
<td>RNN</td>
<td>1-4-1</td>
<td>------</td>
<td>------</td>
<td>29</td>
</tr>
<tr>
<td>TDNN</td>
<td>1-9-1</td>
<td>2</td>
<td>1</td>
<td>55</td>
</tr>
<tr>
<td>TDRNN</td>
<td>1-3-1</td>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 1- Architecture Description of Selected ANN Models-Case 1
and outflow of the main river reach, respectively. As shown in Figure 4(c), two major tributaries, Bazoft and Khersan, join the Karoon River in this reach. The tributary Bozoft is gauged at Marghak gauging station, and the tributary Khersan is gauged at Barz. Streamflow at all these gauging stations is recorded every two hours during floods. It is noteworthy that the Karoon-3 reservoir dam is constructed downstream of Poleshaloo station at a distance of 15 km. Thus, the forecast of floods at Polesaloo station is an important issue of the Karoon-3 reservoir management. Distances between Poleshaloo station and Armand, Marghak and Barz stations are 74, 54 and 41 km, respectively; also, the estimated travel time in these reaches are 7.1, 4.6

![Figure 7- Eight-hour-ahead estimation of downstream flow from TDNN versus HEC-RAS outputs](image)

<table>
<thead>
<tr>
<th>Model</th>
<th>PWRMSE</th>
<th>CE</th>
<th>Mean Error of Time to Peak (hr)</th>
<th>Volume Error of Highest Peaks (%)</th>
<th>PWRMSE</th>
<th>CE</th>
<th>Mean Error of Time to Peak (hr)</th>
<th>Volume Error of Highest Peaks (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>271.9</td>
<td>0.852</td>
<td>2.4</td>
<td>26.2</td>
<td>282.0</td>
<td>0.889</td>
<td>2.6</td>
<td>23.2</td>
</tr>
<tr>
<td>RNN</td>
<td>227.0</td>
<td>0.909</td>
<td>1.6</td>
<td>14.7</td>
<td>216.0</td>
<td>0.932</td>
<td>1.8</td>
<td>11.8</td>
</tr>
<tr>
<td>TDNN</td>
<td>210.0</td>
<td>0.930</td>
<td>1.2</td>
<td>14.6</td>
<td>182.3</td>
<td>0.962</td>
<td>1.4</td>
<td>10.7</td>
</tr>
<tr>
<td>TDRNN</td>
<td>225.7</td>
<td>0.918</td>
<td>1.6</td>
<td>13.5</td>
<td>230.9</td>
<td>0.937</td>
<td>1.8</td>
<td>13.3</td>
</tr>
</tbody>
</table>

Table 2- Comparative Performance of ANN Models for Eight-Hour-Ahead Estimation-Case 1
and 4 hrs, respectively. So, the least estimated travel time (between Poleshaloo and Barz stations) is about 4 hours, which is double a time interval and would be considered as the lead-time of ANN-based models.

Figure 9 shows three major floods occurring in this river, which are measured in the four aforementioned gauging stations. The flood event of January 1989 is of approximately 150 hour duration, the flood event of December 1991 is of approximately 210 hour duration and the flood event of March 1998 is of approximately 90 hour duration. The models are calibrated using the flood event of January 1989. Then, performances of the models are tested using the flood event of December 1991 and the flood event of March 1998.

Results and Discussion (part two)

Table 3 shows the most appropriate architecture of the networks in which $R$, $S_1$ and $S_2$ represent the numbers of input, hyperbolic tangent sigmoid hidden and linear output neurons, respectively. The appropriate values of the maximum performance function and the forgetting factor in the second case study are obtained by the same procedure explained in the first case study. As a result, the maximum performance function and the forgetting factor for all networks are considered as $10^{-10}$ and 0.980, respectively.

To quantify the ANNs performance in the second case study, the Peak-Weighted Root Mean Square Error (PWRMSE) and the Coefficient of Efficiency (CE) are chosen. These criteria are used for each flood event separately and are reported in Table 4. The Muskingum hydrologic routing model is used in the second case study in order to be compared with the ANN-based models. The performance of the Muskingum routing

Figure 8- Scatter plot of eight-hour-ahead estimation using TDNN during testing versus streamflow resulted by HEC-RAS

Figure 9- Flood events of January 1989, December 1991, and March 1998 at three upstream and one downstream (DS) gauging stations

Adaptive Neural Networks for Flood Routing in River Systems
model for the second case study is reported in Table 4, as well.

As demonstrated in Table 4, all dynamic networks perform better in comparison with the static MLP network and the Muskingum model. The TDRNN estimates more accurately than the other models. Figure 10(a) demonstrates the estimation results with the lead-time of four hours obtained by the TDRNN for the three mentioned floods. CE values resulted by the TDRNN for flood events of January 1989 (calibration), December 1991 (testing) and March 1998 (testing) are 0.990, 0.972 and 0.980, respectively. The TDNN also seems to perform better than the RNN. Furthermore, in terms of the number of weights, the TDRNN with 53 weights, as stated in Table 3, is the most computationally complex network. The Muskingum model, although it has no lead time, is less accurate than ANN-based models. Figure 10(b) demonstrates the routing results obtained by the Muskingum method for the three observed floods. Furthermore, Figure 11 presents the scatter plots of the four-hour-ahead estimation resulted by the TDRNN and the results of the Muskingum method versus observed streamflow for the two testing cases.

The scatter plot of the TDRNN (Figure 11(a)) fits well to the 45° line in both low and high flows, but the scatter plot of the Muskingum model (Figure 11(b)) falls closer to that line in low flows rather than high flows.

Summary and Conclusion

A methodology based on adaptive artificial neural networks for flood routing has been proposed in this study. The models are able to estimate the

<table>
<thead>
<tr>
<th>Model</th>
<th>Architecture</th>
<th>Memory order of</th>
<th>Memory order of</th>
<th>Number of Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>3-4-1</td>
<td>-----</td>
<td>-----</td>
<td>21</td>
</tr>
<tr>
<td>RNN</td>
<td>3-3-1</td>
<td>-----</td>
<td>-----</td>
<td>25</td>
</tr>
<tr>
<td>TDNN</td>
<td>3-5-1</td>
<td>1</td>
<td>1</td>
<td>46</td>
</tr>
<tr>
<td>TDRNN</td>
<td>3-4-1</td>
<td>1</td>
<td>1</td>
<td>53</td>
</tr>
</tbody>
</table>

Table 3- Architecture Description of Selected ANN Models-Case 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Calibration)</td>
<td>(Testing)</td>
<td>(Testing)</td>
</tr>
<tr>
<td></td>
<td>PWRMSE</td>
<td>CE</td>
<td>PWRMSE</td>
</tr>
<tr>
<td>MLP</td>
<td>272.1</td>
<td>0.947</td>
<td>264.1</td>
</tr>
<tr>
<td>RNN</td>
<td>224.5</td>
<td>0.976</td>
<td>176.1</td>
</tr>
<tr>
<td>TDNN</td>
<td>204.6</td>
<td>0.983</td>
<td>152.3</td>
</tr>
<tr>
<td>TDRNN</td>
<td>138.8</td>
<td>0.990</td>
<td>147.0</td>
</tr>
</tbody>
</table>

Table 4- Comparative Performance of ANNs and Muskingum Models-Case 2
Figure 10- (a) Four-hour-ahead estimation using TDRNN (b) flood routing using Muskingum method

Figure 11- Scatter plots of (a) four-hour-ahead estimation using TDRNN and (b) estimation of Muskingum method versus observed streamflow for two testing cases
downstream flow with a lead time using the past values of upstream and downstream flows in both converging and diverging river networks. The proposed models can also be used as part of a continuous flood forecasting system. Four types of neural networks, including the Multilayer Perceptron Network (MLP), Recurrent Neural Network (RNN), Time Delay Neural Network (TDNN) and Time Delay Recurrent Neural Network (TDRNN), have been applied in this study. The adaptive training based on the Forgetting Factor (FF) approach has been used to train ANNs models.

The performances of the models have been tested in two case studies. The case studies are two distinctive parts of the Karoon River located in the southwestern part of Iran. The first case study used synthetic data obtained by the HEC-RAS hydraulic model, while the second case study used the gauged data in a river network with three tributaries. As a result, both case studies demonstrate that the proposed methodologies have high efficiency and can estimate the outflow hydrograph accurately with a reasonable lead time, considering four well-known evaluation criteria. The results also show that all applied dynamic networks perform more accurately than the static MLP network. The TDNN and the TDRNN have resulted in better estimation in the first and second case studies, respectively. Furthermore, the Muskingum hydrologic routing model has been used in the second case study to be compared with the ANN-based models. The results show that the proposed models significantly perform better than the Muskingum model, despite having no lead time. Finally, it can be concluded that the use of dynamic neural networks with the adaptive training procedure is a suitable way to realize and detect ongoing problems, such as flood routing.

References


