American-Eurasian J. Agric. & Environ. Sci., 2 (6): 784-791, 2007
ISSN 1818-6769

Flood Estimation at Ungauged Sites Using A New Nonlinear Regression Model and Artificial Neural Networks

[1]*Mahsa Hassanpour Kashani,* [1]*Majid Montaseri and* [2]*Mohammad Ali Lotfollahi Yaghin*

[1]Department of Agriculture, Urmia University, Urmia, Iran
[2]Department of Civil Engineering, Tabriz University, Tabriz, Iran

Abstract: Artificial neural networks (ANNs) have been applied within the field of hydrological modeling but relatively little attention has been paid to the use of these tools for regional flood modeling and flood estimation in ungauged catchments. In this paper, the ability of Multi-Layer Perceptron (MLP) and Elman networks for T-year flood estimation in western and southern catchments of Urmia lake have been evaluated and compared with the result of a new regression model. At first, these networks used physiographic and climatic data selected from the multiple regression model, to train. Finally, the best structure of these networks is chosen based on correlation coefficient between actual and estimated discharges. The obtained results have proved the ability of ANNs to predict T-year flood events and the effect of networks types on prediction precision.

Key words: MLP network . Elman network . new regression model . Regional flood modelling . ungauged catchments

INTRODUCTION

The temporal and spatial variability that characterizes a river system makes flow forecasting a very demanding task. Flow forecasting is a crucial part of flow regulation and water resources management, as it is related to issues such as drought prevention, flood forecasting for dam and human safety and ecosystem sustainability [1]. As it is reported, floods and droughts kill more people and cause more damage than any other natural disaster [2]. Consequently, there is a need for systems capable of efficiently forecasting water levels or discharge rates in rivers. It is difficult to estimate flood at ungauged catchments that there is no information about discharge rates or flood peak data. So, this is one of the biggest problems for hydrologist to estimate flood event magnitudes from catchment properties and/or regional climatology [3]. The UK Flood Estimation Handbook (FEH) recommends that, wherever possible, such estimates should be based on the transfer or analogous data from sites that are hydrologically similar in terms of catchment area, rainfall and soil type [4]. So, it is better to apply methods for flood forecasting that need few hydrological data [3, 5].

Artificial neural networks were first introduced in the 1940s (McCulloch and Pitts, 1943). Interest grew in these tools until the 1960s when Minsky and Papert (1969) showed that networks of any practical size could not be trained effectively. It was not until the mid-1980s that ANNs once again became popular with the research community when Rumelhart and McClelland (1986) rediscovered a calibration algorithm that could be used to train networks of sufficient sizes and complexities to be of practical benefit. Since that time research into ANNs has expanded and a number of different network types, training algorithms and tools have evolved [4]. The structure of ANNs is very similar to human brain's structure and has the ability of learning, generalizing and deciding. So, they can solve many problems that are difficult to understand, define and quantify such as economical, medical, engineering problems, etc [6].

It was the first time to apply ANNs in water sciences by Daniel in 1991. Then, it was applied for discharge prediction at different catchments [7]. In the context of this paper, ANNs are trained to represent the relationship between a range of catchment descriptors, rainfall and associated flood event magnitudes. There is no need for the modeler in this case to fully define the intermediate relationships (physical processes) between catchment descriptors or rainfall and flood event magnitudes-the ANN identifies these during the 'learning process'.

Corresponding Author: Dr. Mahsa Hassanpour Kashani, Department of Agriculture, Urmia University, Urmia, Iran

TOOLS AND METHODS

Nonlinear Regression Model (NRM): In the present study, a new regression model which does not have been applied within the field of regional flood modeling, is defined as follows:

$$Q_{T_r} = \alpha T_r^{\beta} \tag{1}$$

Where Q_{T_r} is T-year flood event magnitude (m^3/s), T_r is the return period (year), α and β are the parameters which are defined based on the physiographical and climatic data as follows:

$$\begin{aligned}\alpha = {} & a_0 + a_1 A + a_2 P + a_3 L_r + a_4 S_r + a_5 S_b \\ & + a_6 F_g + a_7 F_b + a_8 T_c + a_9 H + a_{10} R_2\end{aligned} \tag{2}$$

$$\begin{aligned}\beta = {} & b_0 + b_1 A + b_2 P + b_3 L_r + b_4 S_r + b_5 S_b \\ & + b_6 F_g + b_7 F_b + b_8 T_c + b_9 H + b_{10} R_2\end{aligned} \tag{3}$$

Where A is the catchment drainage area (km^2), P is the catchment perimeter (km), L_r is the Longest drainage path (km), S_b is the catchment mean slope (%), S_r is the river mean slope (%), F_g is the gravelius factor, F_b is the form factor, T_c is the time of concentration (hr), R_2 is the 2-year rainfall (mm), H is the Mean altitude of catchment above sea level (m) and a_i and b_i are the model constant coefficients.

Multi-Layer Perceptron Network (MLP): Although there are now a significant number of network types and training algorithms, this paper will focus on the Multi-Layer Perceptron (MLP) and Elman networks. Figure 1 and 2 provide an overview of the structure of these networks, respectively.

In this case, the ANN has three layers of neurons (nodes)-an input layer, a hidden layer and an output layer. Each neuron has a number of inputs (from outside the network or the previous layer) and a number of outputs (leading to the subsequent layer or out of the network). A neuron computes its output response based on the weighted sum of all its inputs according to an activation function (in this case the tangent sigmoid). Data flows in one direction through this kind of network-starting from external inputs into the first layer (the predictors), that are transmitted through the hidden layer and then passed to the output layer from which the external outputs (predictands) are obtained. The network is trained by adjusting the weights that connect the neurons using a procedure called error back propagation. In this procedure, the network is presented with a series of training examples (predictors and their associated predictands) and the internal weights are adjusted in an attempt to model the predictor/predictand relationship. This procedure must be repeated many times before the network begins to model the relationship [4, 6].

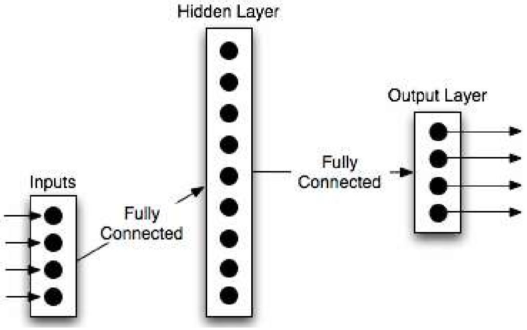

Fig. 1: Multi-Layer Perceptron (MLP) network

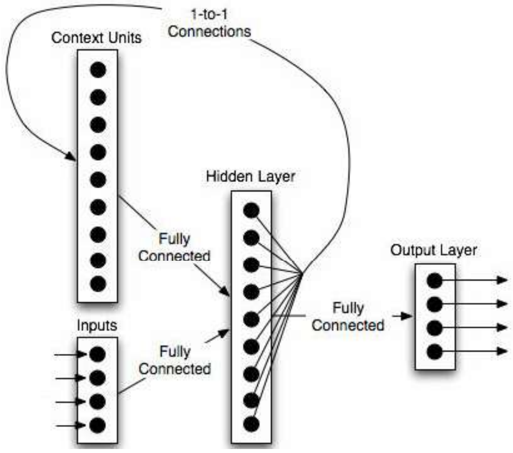

Fig. 2: Elman network

Elman network: Elman Networks are a form of recurrent Neural Networks which have connections from their hidden layer back to a special copy layer. This means that the function learnt by the network can be based on the current inputs plus a record of the previous state(s) and outputs of the network. In other words, the Elman net is a finite state machine that learns what state to remember (i.e., what is relevant). The special copy layer is treated as just another set of inputs and so standard back-propagation learning techniques can be used (something which isn't generally possible with recurrent networks). At each time step, a copy of the hidden layer units is made to a copy layer. Processing is done as follows:

- Copy inputs for time t to the input units
- Compute hidden unit activations using net input from input units and from copy layer
- Compute output unit activations as usual
- Copy new hidden unit activations to copy layer

RESULTS AND DISCUSSION

Nonlinear regression model: The parameters of the nonlinear model (α and β), are calculated using Excel software and the Linest function. In order to determine the best variable that has the most effect on the multiple regression models, T-year flood magnitudes are estimated using each variable in the both models. The first variable is chosen based on the maximum correlation coefficient between actual and estimated discharges. The second effective variable is also chosen by testing the effect of the both rest variables and the first variable on the models and etc. This will be continued until there is no difference between two successive maximum correlation coefficients or the last obtained correlation coefficient is enough high. Table 1 shows these selected variables. As can be seen from Table 1, the presence of F_b and F_g in south catchment and also the presence of S_b and S_r in west catchment models is not reasoned because of their similar influences on T-year flood magnitudes. So, the data of A and F_g (No.2) and F_b, A, S_r and H (No.4) are recommended as the best variables for flood estimation in south and west catchments, respectively.

MLP network: The selected data from the regression model are considered as the neural network's inputs. They were normalized in the range of [-1,1] because of using the tangent sigmoid transfer function in the hidden layer, as follow:

$$P_n = \frac{2(P_i - P_{min})}{(P_{max} - P_{min})} - 1 \qquad (4)$$

Where P_i is the inputs, P_{min} and P_{max} are minimum and maximum input respectively and P_n is the normalized inputs. In order to improve generalization, all data were divided into three sets: training set (%75 of data), validation set (%30 of training set) and test set (%25 of data). Twelve training algorithms such as: Basic gradient descent (gd), Gradient descent with momentum (gdm), Adaptive learning rate (gda, gdx), Resilient back propagation (rp), Fletcher-Reeves conjugate gradient algorithm (cgf), Polak-Ribiére conjugate gradient algorithm (cgp), Powell-Beale conjugate gradient algorithm (cgb), Scaled conjugate gradient algorithm (scg), BFGS quasi-Newton method (bfg), One step secant method (oss) and Levenberg-Marquardt algorithm (lm) were applied to train the network. After training, network was tested based on the test data and performed a linear regression between the network outputs and the corresponding targets by putting the entire data set through the network (training, validation and test) to measure performance of the trained network. It returns three parameters. The first two, m and b, correspond to the slope and the y-intercept of the best linear regression relating targets to network outputs. If we have a perfect fit (outputs exactly equal to targets), the slope would be 1 and the y-intercept would be 0. The third variable returned by regression analysis is the correlation coefficient (R-value) between the outputs and targets. It is a measure of how well the variation in the output is explained by the targets. If this number is equal to 1, then there is perfect correlation between targets and outputs. Finally, the optimum number of the hidden neurons for each algorithm and input numbers was determined based on the maximum correlation coefficient (Table 2 and 3). It can be seen that lm and bfg algorithms are the best algorithms in south and west catchments with average correlation coefficient of 0.973 and 0.916, respectively. Also, their results are better than the nonlinear regression model. Gd, gdm, gda, gdx algorithms were recognized as weak algorithms. Maximum average correlation coefficient of all algorithms at each input

Table 1: The best variables chosen by the nonlinear regression model

Catchments	South catchment		West catchment	
No.	Best variables	Correlation coefficient	Best variables	Correlation coefficient
1	A	0.874	F_b	0.712
2	A- F_g	0.901	F_b -A	0.838
3	A- F_g - F_b	0.925	F_b -A-S_r	0.886
4	A- F_g - F_b - R_2	0.930	F_b -A- S_r -H	0.936
5	A- F_g - F_b - R_2 - S_r	0.948	F_b -A- S_r -H-S_b	0.949
6	A- F_g - F_b - R_2 - S_r -L_r	0.974	F_b -A- S_r -H- S_b -L_r	0.953

Table 2: Results of MLP and Elman networks in south catchment

		gd			gdm			gda					
Algorithms	No.	R	neuron	epochs	R	neuron	epochs	R	neuron	epochs			
Back propagation	1	0.738	3	110	0.790	13	110	0.845	13	89			
	2	0.803	7	110	0.835	1	110	0.930	8	105			
	3	0.781	7	110	0.780	10	110	0.892	10	95			
	4	0.651	7	110	0.783	7	110	0.896	11	105			
	5	0.736	10	110	0.778	8	110	0.935	9	102			
	6	0.750	9	110	0.879	1	110	0.960	9	110			
		gdx			gdx*			rp					
		R	neuron	epochs	R	neuron	epochs	R	neuron	epochs			
Back propagation	1	0.864	19	110	0.855	18	110	0.891	19	110			
	2	0.906	10	110	0.899	15	110	0.965	16	108			
	3	0.915	9	110	0.911	12	110	0.960	11	84			
	4	0.896	8	80	0.918	9	110	0.951	11	102			
	5	0.926	8	110	0.931	6	110	0.971	8	60			
	6	0.917	4	110	0.936	7	110	0.975	9	47			
		cgf			cgb			cgp			scg		
		R	neuron	epochs	R	neuron	epochs	R	neuron	epochs	R	neuron	epochs
Conjugate gradient	1	0.883	19	77	0.890	15	81	0.888	20	110	0.880	17	88
	2	0.961	7	70	0.971	15	59	0.964	11	110	0.966	12	110
	3	0.967	13	52	0.974	5	97	0.957	9	110	0.966	13	65
	4	0.964	10	84	0.966	4	104	0.961	4	71	0.964	9	89
	5	0.969	6	88	0.969	7	49	0.969	5	55	0.973	9	70
	6	0.975	7	66	0.973	6	58	0.969	7	41	0.974	2	108
		oss			bfg								
		R	neuron	epochs	R	neuron	epochs						
Quasi-Newton	1	0.882	13	108	0.929	19	110						
	2	0.947	16	62	0.972	12	41						
	3	0.956	2	84	0.965	10	56						
	4	0.960	6	70	0.974	4	68						
	5	0.958	8	53	0.974	8	50						
	6	0.970	7	85	0.974	3	75						
		lm											
		R	neuron	Epochs									
Levenberg -Marquardt	1	0.937	8	55									
	2	0.979	9	7									
	3	0.981	7	10									
	4	0.982	7	6									
	5	0.979	9	6									
	6	0.980	8	7									

gdx*is related to Elman network

number was equal to 0.941 related to the network with six input variables (No. 6) and 0.905 related to the network with five input variables (No. 5) in both south and west catchments, respectively. But, there was not more difference between them and 0.933 (No. 2) and 0.891 (No. 4) in both catchments, respectively. So, it can be concluded that MLP network is able to estimate flood using just two variables in south catchment and four variables in west catchment, precisely and there is no need for more inputs. In other words, MLP network is able to recognize the effective variables needed for flood forecasting. More input variables do not always cause to a good performance of MLP network.

Elman network: Elman network was trained as similar as MLP, but only gdx algorithm was applied to its training. Table 2 and 3, also show the results of this network. It can be seen that, this network with six input

Table 3: The results of MLP and Elman networks in west catchment

		gd			gdm			gda					
Algorithms	No.	R	neuron	epochs	R	neuron	epochs	R	neuron	epochs			
Back propagation	1	0.691	13	120	0.606	5	48	0.711	13	101			
	2	0.664	11	120	0.687	9	18	0.841	8	110			
	3	0.669	3	120	0.719	5	120	0.879	5	112			
	4	0.554	1	120	0.847	9	48	0.888	5	120			
	5	0.772	6	120	0.673	3	31	0.928	11	101			
	6	0.657	9	120	0.658	6	120	0.910	4	105			
		gdx			gdx*			rp					
		R	neuron	epochs	R	neuron	epochs	R	neuron	epochs			
Back propagation	1	0.725	17	120	0.698	12	118	0.712	16	69			
	2	0.856	17	120	0.788	9	85	0.891	11	80			
	3	0.887	13	120	0.870	10	120	0.893	12	50			
	4	0.940	9	120	0.888	10	120	0.930	8	79			
	5	0.927	1	120	0.875	10	120	0.950	8	120			
	6	0.787	3	120	0.906	7	120	0.903	3	20			
		cgf			cgb			cgp			scg		
		R	neuron	epochs	R	neuron	epochs	R	neuron	epochs	R	neuron	epochs
Conjugate gradient	1	0.716	13	17	0.717	5	37	0.714	17	21	0.713	17	28
	2	0.935	9	63	0.913	4	43	0.937	14	40	0.907	10	48
	3	0.910	5	27	0.907	14	29	0901	12	42	0.893	12	32
	4	0.901	9	41	0.942	9	52	0.922	9	48	0.939	9	49
	5	0.924	1	20	0.946	2	53	0.961	6	30	0.922	1	17
	6	0.929	4	46	0.939	4	43	0.957	6	69	0.924	1	32
		oss			bfg								
		R	neuron	epochs	R	neuron	epochs						
Quasi-Newton	1	0.716	16	32	0.731	5	68						
	2	0.924	13	41	0.931	13	52						
	3	0.918	13	77	0.947	11	44						
	4	0.936	10	58	0.945	12	31						
	5	0.944	6	91	0.973	9	34						
	6	0.942	8	50	0.972	7	40						
		lm											
		R	neuron	Epochs									
Levenberg -Marquardt	1	0.723	7	18									
	2	0.947	4	36									
	3	0.937	8	12									
	4	0.951	5	9									
	5	0.937	4	40									
	6	0.966	3	8									

gdx* is related to Elman network

variables (No.6) has the best performance in west catchment because of the maximum correlation coefficient value (0.906) and there is a remarkable difference between it and the one of No.4 (0.888). Also in south catchment, the network shows the best results when it takes six input variables (No.6) because of the maximum correlation coefficient value (0.936) that is more than the correlation coefficient of No.2 (0.899). So, Elman network could not recognize the best variables for flood forecasting, well. In general, Elman network shows a weak performance compared with MLP and the nonlinear regression model in both catchments. According to the Tables, it can be seen that Elman network has a low convergence speed. Also, this network needs more hidden neurons than MLP. Figure 3 and 4 shows the actual and estimated flow from the best algorithms of MLP, Elman network and the nonlinear regression model using the suggested variables in west and south catchments, respectively.

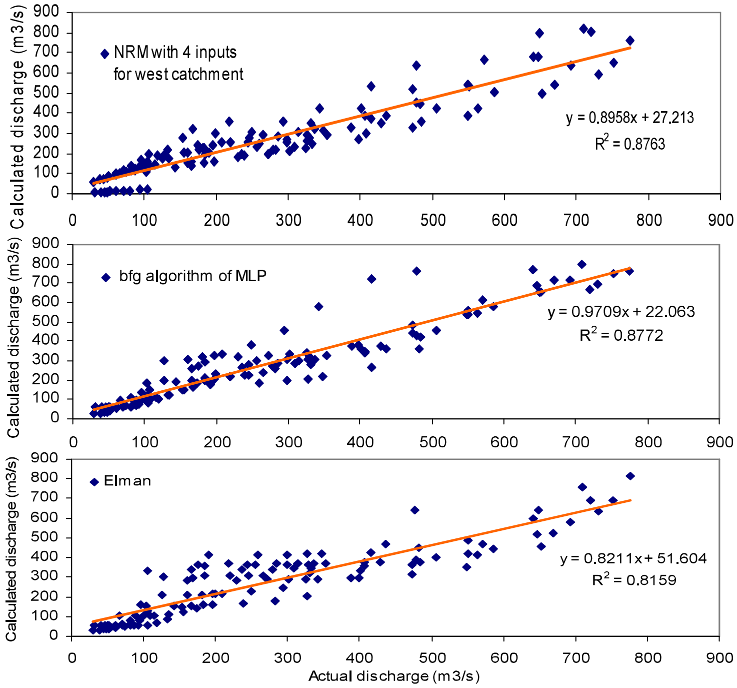

Fig. 3: Actual and estimated flow from NRM, MLP (bfg) and Elman networks for west catchment

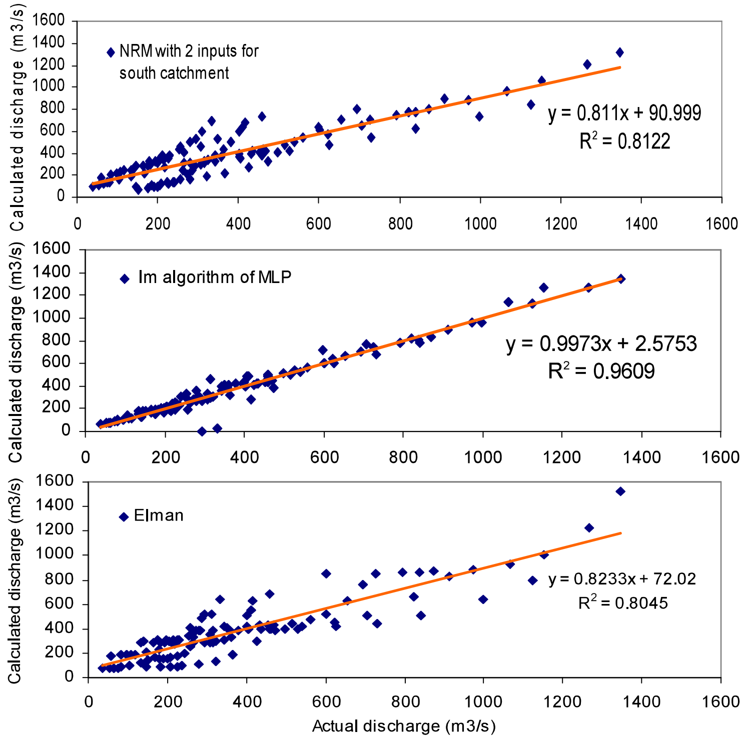

Fig. 4: Actual and estimated flow from NRM, MLP (lm) and Elman networks for south catchment

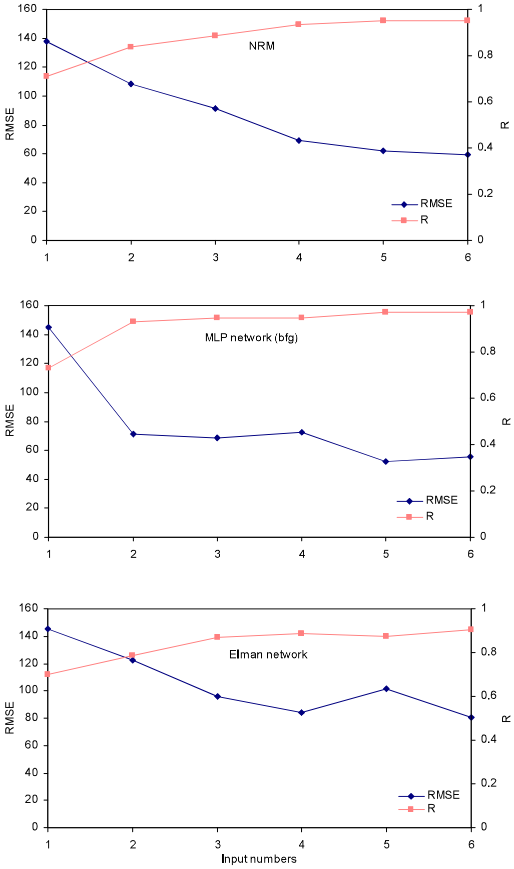

Fig. 5: RMSE and correlation coefficients (R) values versus input numbers of NRM, MLP (bfg) and Elman networks in west catchment

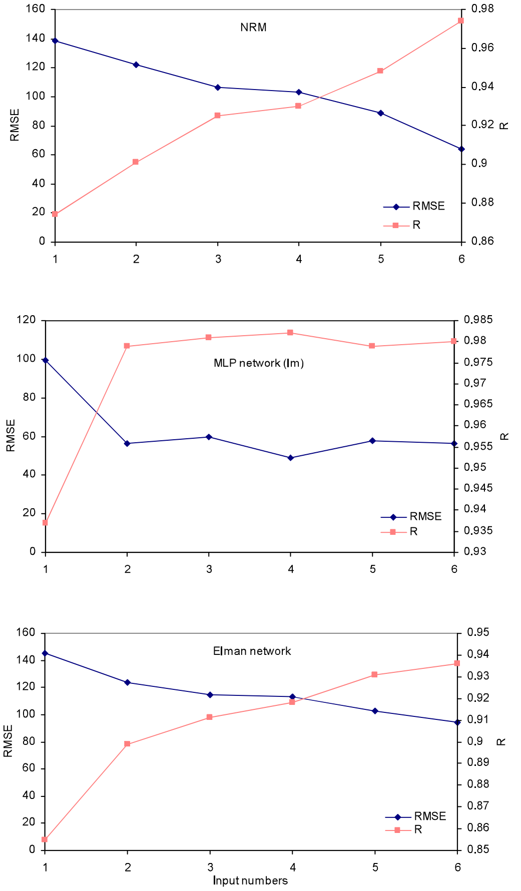

Fig. 6: RMSE and correlation coefficients (R) values versus input numbers of NRM, MLP (lm) and Elman networks in south catchment

The performances of networks with varying number of input (from 1 to 6 in addition to the return period variable) and hidden nodes were also investigated and classified according to the RMSE (Root Mean Squared Error) of the estimated T-year floods as compared to actual T-year floods. RMSE is stated as below:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(Q_{T_{ri}}^{est.} - Q_{T_{ri}}^{act.}\right)^{2}}{N}} \tag{5}$$

Where $Q_{T_{ri}}^{est.}$ = estimated T-year flood, $Q_{T_{ri}}^{est.}$ = actual T-year flood and N = total number of T-year flood values.

The obtained results confirm the previous obtained results based on the correlation coefficients.

Figure 5 and 6 shows RMSE and correlation coefficients (R) values versus input numbers of NRM, the best algorithms of MLP and Elman networks in west and south catchments, respectively.

CONCLUSION

In the nonlinear regression model, as the number of input variables increases, the values of correlation coefficients increase and the RMSE values reduce. In other words, the regression model needs more variables to estimate flood precisely and it is difficult to recognize the best variables using the regression model. In south catchment, form of the sub catchments and in west catchment, the altitude of sub catchments has more effect on T-year flood magnitudes. MLP network (except some weak algorithms) outperforms Elman and the regression model; also, lm and bfg algorithms of this network are suggested to estimate flood in south and west catchments, respectively. The regression model (in determining MLP's input variables) and MLP network (in determining the best variables for flood forecasting and estimating T-year floods) as a hybrid model showed satisfactory results. In south catchment, A and F_g variables and in west catchment, F_b, A, S_r and H variables are suggested to estimate T-year flood magnitudes. Elman network has a low convergence speed and needs more hidden neurons than MLP. Elman network did not outperform MLP and the regression model in both catchments, so, it is not suggested to estimate T-year flood events in both catchments of Urmia lake.

REFERENCES

1. Lekkas, D.F., C. Onof, M.J. Lee and E.A. Baltas. 2004. Application of artificial neural networks for flood forecasting. The Int J., 6 (3): 209-215.
2. WMO, 1992. Simulated real-time intercomparison of hydrological models. In: WMO operational hydrology report, No. 38, WMO No. 779, World Meteorological Organization, Geneva.
3. Rostami, S., 2003. Regional flood modeling for western basin of Urmia lake, M.S. Thesis, Urmia Univ., Iran.
4. Dawson, C.W., R.J. Abrahart, A.Y. Shamseldin and R.L. Wilby, 2006. Flood estimation at ungauged sites using artificial neural networks. Journal of Hydrology, 319: 391–409.
5. Salehi, M., 2003. Regional flood modeling for southern basin of Urmia lake. M.S. Thesis, Urmia Univ., Iran.
6. Ghorbani, M.A., 2004. Application of artificial neural networks for simulation of rainfall-runoff process. Journal of Daneshe Keshavarzi.
7. Govindaraju, R.S., 2000. Artificial neural networks in hydrology.II: Hydrologic applications. Journal of Hydrologic Engineering, Vol: 5 (2).