American-Eurasian J. Agric. & Environ. Sci., 5 (6): 856-865, 2009
ISSN 1818-6769

Rainfall-runoff Prediction Based on Artificial Neural Network (A Case Study: Jarahi Watershed)

Karim Solaimani

GIS and RS Centre, University of Agriculture and Natural Resources, Sari, Iran

Abstract: The present study aims to utilize an Artificial Neural Network (ANN) to modeling the rainfall-runoff relationship in a catchment area located in a semiarid region of Iran. The paper illustrates the applications of the feed forward back propagation for the rainfall forecasting with various algorithms with performance of multi-layer perceptions. The monthly stream of Jarahi Watershed was analyzed in order to calibrate of the given models. The research explored the capabilities of ANNs and the performance of this tool would be compared to the conventional approaches used for stream flow forecast. Efficiencies of the gradient descent (GDX), conjugate gradient and Levenberg-Marquardt (L-M) training algorithms are compared to improving the computed performances. The monthly hydrometric and climatic data in ANN were ranged from 1969 to 2000. The results extracted from the comparative study indicated that the Artificial Neural Network method is more appropriate and efficient to predict the river runoff than classical regression model.

Key words: Rainfall-runoff . ANN . prediction . Jarahi watershed . Iran

INTRODUCTION

Rainfall-runoff models play an important role in water resource management planning and therefore, different types of models with various degrees of complexity have been developed for this purpose. These models, regardless to their structural diversity generally fall into three broad categories; namely, black box or system theoretical models, conceptual models and physically-based models [1]. Black box models normally contain no physically-based input and output transfer functions and therefore are considered to be purely empirical models. Conceptual rainfall-runoff models usually incorporate interconnected physical elements with simplified forms and each element is used to represent a significant or dominant constituent hydrologic process of the rainfall-runoff transformation [2]. Conceptual rainfall-runoff models have been widely employed in hydrological modeling. Some of the well-known conceptual models include the Stanford Watershed Model (SWM) [3], the Sacramento Soil Moisture Accounting (SAC-SMA) model [4], the Xinanjiang Model [5-7], the Soil Moisture Accounting and Routing (SMAR) Model [8, 9] and the Tank Model [10, 11]. Conceptual models are reliable in forecasting the most important features of the hydrograph [12, 13]. In comparison with the black box models, conceptual models have potential for evaluating land-use impact on hydrological processes based on relationships of the model parameters to measurable physical characteristics and development [2]. It seems reasonable to expect that conceptual models would prove to be more faithful in simulating rainfall-runoff process due to its physical basis.

There has been a tremendous growth in the interest of application the ANNs in rainfall-runoff modeling in the 1990s [14-20]. ANNs were usually assumed to be powerful tools for functional relationship establishment or nonlinear mapping in various applications. Cannon and Whitfield [21], found ANNs to be superior to linear regression procedures. Shamseldin [22], examined the effectiveness of rainfall-runoff modeling with ANNs by comparing their results with the Simple Linear Model (SLM), the seasonally based Linear Perturbation Model (LPM) and the Nearest Neighbor Linear Perturbation Model (NNLPM) and concluded that ANNs could provide more accurate discharge forecasts than some of the traditional models. The ability of ANNs as a universal approximator has been demonstrated when applied to complex systems that may be poorly described or understood using mathematical equations; problems that deal with noise or involve pattern recognition, diagnosis and generation; and situations where input is incomplete or ambiguous by nature [17]. An excellent overview of the preliminary concepts and hydrologic applications of ANNs was provided by the ASCE Task Committee on Artificial Neural Networks in Hydrology [23, 24].

Corresponding Author: Dr. Karim Solaimani, GIS and RS Centre, University of Agriculture and Natural Resources, Sari, Iran

While the capability of ANNs to capture nonlinearity in the rainfall-runoff process remains attractive features comparing with other modeling approaches [14], ANN models as illustrated in numerous previous studies, essentially belong to system theoretical (black box) model category and bear the weaknesses of this category [25]. The formation of ANN model inputs usually consists of meteorological variables, such as rainfall, evaporation, temperatures and snowmelt; and geomorphological properties of the catchment, such as topography, vegetation cover and antecedent soil moisture conditions. The frequently used inputs to ANNs also include observed runoff at nearby sites or neighboring catchments. In many cases, network inputs with or without time lags may also be considered in scenario analysis. Nevertheless, as concluded in previous studies, the lack of physical concepts and relations has been one of the major limitations of ANNs and reasons for the skeptical attitude towards this methodology, (ASCE Task Committee on Artificial Neural Networks in Hydrology [23, 24]. Despite that, the nonlinear ANN model approach is capable of providing a better representation of the rainfall-runoff relationship than the conceptual Sacramento soil moisture accounting model, also the ANN approach is by no means a substitute for conceptual watershed modeling since it does not employ physically realistic components and parameters [14]. Therefore, instead of using ANNs as simple black box models, the development of hybrid neural networks has received considerable attention [26-28]. The hybrid neural networks has shown the potential of obtaining more accurate predictions of process dynamics by combining mechanistic and neural network models in such a way that the neural network model properly accounts for unknown and nonlinear parts of the mechanistic model [26].

MATERIALS AND METHODS

Overview of artificial neural networks: An ANN is a highly interconnected network of many simple processing units called neurons, which are analogous to the biological neurons in the human brain. Neurons having similar characteristics in an ANN are arranged in groups called layers. The neurons in one layer are connected to those in the adjacent layers, but not to those in the same layer. The strength of connection between the two neurons in adjacent layers is represented by what is known as a 'connection strength' or 'weight'. An ANN normally consists of three layers, an input layer, a hidden layer and an output layer. In a feed-forward network, the weighted connections feed activations only in the forward direction from an input

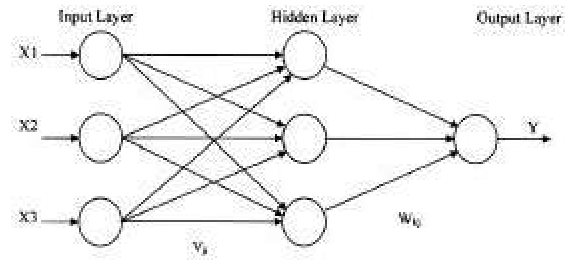

Fig. 1: Structure of a feed-forward ANN

layer to the output layer. On the other hand, in a recurrent network additional weighted connections are used to feed previous activations back to the network. The structure of a feed-forward ANN is shown in Fig. 1.

An important step in developing an ANN model is the determination of its weight matrix through training. There are primarily two types of training mechanisms, supervised and unsupervised. A supervised training algorithm requires an external teacher to guide the training process. The primary goal in supervised training is to minimize the error at the output layer by searching for a set of connection strengths that cause the ANN to produce outputs that are equal to or closer to the targets. A supervised training mechanism called back-propagation training algorithm [29, 30] is normally adopted in most of the engineering applications. Another class of ANN models that employ an 'unsupervised training method' is called a self-organizing neural network. The most famous self-organizing neural network is the Kohonen's Self-organizing Map (SOM) classifier, which divides the input-output space into a desired number of classes. Once the classification of the data has been achieved by using a SOM classifier, the separate feed forward MLP models can be developed through considering the data for each class using the supervised training methods. Since the ANNs do not consider the physics of the problem, they are treated as black-box models; however, some researchers have recently reported that it is possible detect physical processes in trained ANN hydrologic models [31-33].

The artificial neural network structure: Network structure includes input and output dimensions, the number of hidden neurons and model efficiency calculations. In this study, input dimension includes monthly stream flow, rainfall and average air temperature data for time step t. Output dimension is the predicted stream flow at time t+1. Only one hidden layer was used. This has been shown to be sufficient in a number of studies [34-36]. The appropriate number of neurons in the hidden layer is determined by using the constructive algorithm [36], by increasing the number of neurons from 2 to 20. There is a use of log-sigmoid, tangent-hyperbolic and linear activation functions. The ANN model for stream flow evaluation was written in the MATLAB environment, version 7. The L-M algorithms were evaluated for network training so that the algorithm with better achieved accuracy and convergence speed could be selected. In order to provide adequate training, network efficiency was evaluated during the training and validation stages, as suggested by Rajurkar *et al*. [34]. In this case, if the calculated errors of both stages continue to decrease, the training period is increased. This is continued to the point of the training stage error starting to decrease, but the validation stage error starting to increase. At this point training is stopped to avoid overtraining and optimal weights and biases are determined. Capability of the stream flow generation model during either training or validation stage can be evaluated by one of the commonly used error computation functions [34, 35].

Network training algorithms: The Back-propagation (BP) algorithm [37], has been the most commonly used training algorithm. A temporal BP neural network (TBP-NN) was used by Sajikumar and Thandaveswara [18] for rainfall-runoff modelling with limited data. Hsu *et al.* [14] proposed the Linear Least-squared Simplex (LLSSIM) for the training of ANNs. The CG algorithm has also been used to train ANNs by several researchers including Shamsedin [22]. In a study by Chiang *et al.* [35], the CG algorithm was found to be superior when compared with the BP algorithm in terms of the efficiency and effectiveness of the constructed network. In more recent studies the L-M algorithm is also being used due to its superior efficiency and high convergence speed [38, 39]. All commonly used algorithms for network training in hydrology, i.e. BP, CG and L-M algorithms apply a function minimization routine, which can back propagate error into the network layers as a means of improving the calculated output. Here is the corresponding equation [40].

$$\Delta\chi_k = \chi_{k+1} - \chi_k = \alpha_k p_k \quad (1)$$

where: xk is the current estimation point for a function G(x) to be minimized at the kth stage, p_k is the search vector and α_k is the learning rate, a scalar quantity greater than zero. The learning quantity identifies the step size for each repetition along p_k. Computation of p_k will depend on the selected learning algorithm. In the present research, the L-M algorithm is compared with the CG and GDX algorithm. When compared with the steepest gradient and the Newton's methods, the CG is viewed as being faster than the steepest gradient, while not requiring the complexities associated with calculation of Hessian matrix in the Newton's method. The CG is something of a compromise; it does not require the calculation of second derivatives, yet it still has the quadratic convergence property. It converges to the minimum of a quadratic function in a finite number of iterations [40]. On the other hand, the L-M algorithm is viewed as a very efficient algorithm with a high convergence speed. In correspondence with Eqn (1) the following equation is used as the function minimization routine in the L-M procedures [40].

$$\chi_{k+1} = \chi_k - A_k^{-1} g_k \quad (2)$$

in which g_k and A_k are the first and the second derivative of G(x) with respect to x. The function minimization routine is further described [40].

$$\chi_{k+1} = \chi_k - \frac{1}{2\mu_k}\nabla G(\chi) \quad (3)$$

In this equation, if the value of coefficient m_k is decreased to zero the algorithm becomes Gauss-Newton. The algorithm begins with a small value for μ_k (e.g. μ_k = 0.001). If a step does not yield a smaller value for G(x), the step is repeated with μ_k multiplied by a factor greater than one (e.g. 10). Eventually G(x) decreases, since we would be taking a small step in the direction of the steepest descent. If a step does produce a smaller value for G(x), then it is divided by the specified factor (e.g. 10) for the next step, so that the algorithm will approach Gauss-Newton, which should provide faster convergence. The algorithm provides a neat compromise between the speed of Newton's method and the guaranteed convergence of steepest descent.

Study site description: The purpose of this section is to briefly describe the study area and the structure of the utilized ANN. The proposed methodologies were applied to the Jarahi watershed system (Fig. 2) for the evaluation of the predication rainfall-runoff. Modeling capabilities of the GDX, CG and the L-M algorithm were compared in terms of their abilities in network training. Also, there has been an evaluation of the influences of monthly rainfall, stream flow and air temperature data as different input dimensions. The Jarahi watershed with a drainage area of 24/310 squire km is located in Ahvaz the southern region of Khuzestan province in Iran, which supplies water for agricultural, drinking and industrial purposes. The Jarahi watershed system discharges into the Alah River and Maroon River. The sources of water include spring

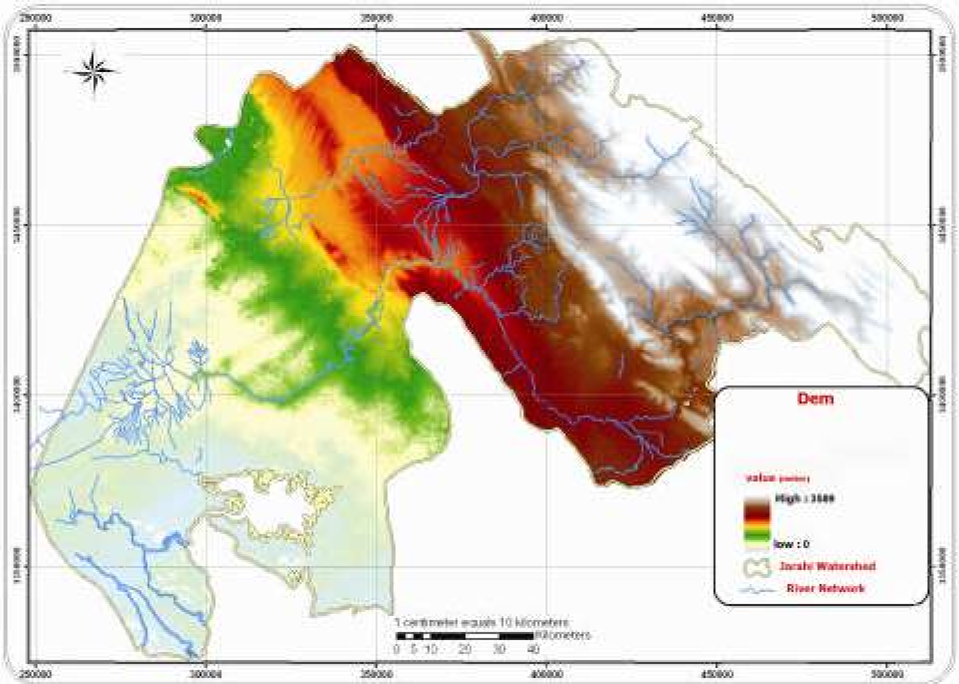

Fig. 2: DEM and river network map of the study area

discharge and individual rain events. The rainy season is limited to a 9 month period despite the continuous stream flow system (October-April). The mean annual precipitation is about 400 mm. Rain events usually last several days, although rainfall durations of 1 day or less do occur.

The used data included monthly recorded rainfall from the rain-gauges of Mashregh, Shohada, Gorg, Shadegan and Shoe; Shadegan, Shoe temperature data and Mashregh, Behbahan and Gorg stream flow as well. Duration of these recorded data was 17 years from 1983 to 2000. A number of ANN models were designed and evaluated for their capability on stream flow prediction. Computational efficiencies of the GDX, CG and L-M algorithms and the effect of enabling/disabling of input parameters (various combinations of stream flow, rainfall and average air temperature) were also evaluated.

Data preparation: Data on steam flow was limited to a 17-yr period, then this period of record was used for model development by considering different combinations of inputs variables, e.g. rainfall, stream flow and average air temperature. For each one of the developed models available data were separated as 80% for training and 20% for validation. Additional analysis for the detection of network overtraining was not necessary in this research as the number of data points was more than the number of parameters used by the network (weights and biases). So data could be divided into two parts for use in the training and validation stages [41]. Data usage by an ANN model typically requires data scaling. This may be due to the particular data behavior or limitations of the transfer functions. For example, as the outputs of the logistic transfer function are between 0 and 1, the data are generally scaled in the range 0.1-0.9 or 0.2-0.8, to avoid problems caused by the limits of the transfer function [36] In the present paper, the data were scaled in the range of _1 to +1, based on the following equation:

$$p_n = 2\left(\frac{p_0 - p_{min}}{p_{max} - p_{min}}\right) - 1 \quad (4)$$

in which, p_0 is the point observed data, p_n is scaled data and p_{max}, p_{min} are the maximum and minimum observed data points. The above equation was used to scale average of air temperature, as well stream flow and rainfall data in order to provide consistency for the analysis. Then the unit of the scaled p_n would correspond to individual data set. Burden *et al*. [42] suggested that before any data preprocessing is carried out, the whole data set should be divided into their respective subsets (e.g. training and validation). In this study, the sub-routines 2 section available in the Neural Network Toolbox of MATLAB were utilized for normalization of the training data set, the transformation of the validation data set and the un-normalization of the network output. The routine normalizes the inputs and targets between-1 and 1, so that they will have zero mean and unit standard deviation. The validation data set was normalized with the mean and standard deviation, which were computed for the training data set. Finally, the network output was un-normalized and a regression analysis was carried out between the measured data and their corresponding un-normalized predicted data.

Evaluation criteria for ANN prediction: The performances of the ANN are measured with four efficiency terms. Each term is estimated from the predicted values of the ANN and the measured discharges (targets) as follows:

- The correlation coefficient (R-value) has been widely used to evaluate the goodness-of-fit of hydrologic and hydrodynamic models [43]. This is obtained by performing a linear regression between the ANN-predicted values and the targets and is computed by

$$R = \frac{\sum_{i=1}^{N} t_i p_i}{\sqrt{\sum_{i=1}^{n} t_i^2}\sqrt{\sum_{i=1}^{n} p_i^2}}$$

  where R is correlation coefficient; N is the number of samples; $t_i = T_i - \overline{T}$; $P_i = P_i - \overline{P}$ and Ti and Pi are the target and predicted values for i=1,….,N and $\overline{T}$ and $\overline{P}$ are the mean values of the target and predicted data set, respectively. [Note that a case with R is equal to 1 refers to a perfect correlation and the predicted values are either equal or very close to the target values, whereas there exists a case with no correlation between the predicted and the target values when R is equal to zero. Intermediate values closer to 1 indicate better agreement between target and predicted values [43].
- The ability of the ANN-predicted values to match measured data is evaluated by the Root Mean Square Error (RMSE). It is defined [44].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i - P_i)^2}$$

Overall, the ANN responses are more precise if R, MSE and RMSE are found to be close to 1, 0 and 0, respectively. In the present study, MSE is used for network training, whereas R and RMSE are used in the network-validation phase.

Sensitivity of analysis

BPNN structure optimization: The crucial tasks in BP neural network modeling are to design a network with a specific number of layers, each having a certain number of neurons and to train the network optimally, so that it can map the system's nonlinearity reasonably well. Some researchers stressed that optimal neural network design is problem-dependent and is usually determined by a trial-and-error procedure, i.e. sensitivity of analysis [24, 36, 45].

Calibrations and validations: In neural network methodology, learning, which extracts information from the input data, is a crucial step that is badly affected through (1) the selection of initial weights and (2) the stopping criteria of learning [46, 47]. If a well-designed neural network is poorly trained, the weight values will not be close to their optimum and the performance of the neural network will suffer. Little research has been conducted to find good initial weights. In general, initial weight is implemented with a random number generator that provides a random value [46, 48]. In this study, the initial weights were randomly generated between-1 and 1 [49]. To stop the training process, we could either limit the number of iterations or set an acceptable error level for the training phase.

There is no guarantee that coefficients which are close to optimal values will be found during the learning phase even though the number of iterations is capped at a predefined value. Therefore, to ensure that overtraining does not occur, we used three criteria to stop the training process: (a) RMSE is predefined and the training is conducted until the RMSE decreases to the threshold value. The idea is to let the system train until the point of diminishing returns, that is, until it basically cannot extract any more information from the training data [40, 46]. (b) Based on preliminary examinations, it was observed that the neural network error decreases as low as threshold RMSE within 100 epochs for good initial weights without overtraining; however, the threshold values may never be achieved for poor initial weights, even after a large number of epochs. (c) The minimum performance gradient (the derivatives of network error with respect to those weights and bias) is set to 10^{10}. The termination of the training process of the network is justified because the BPNN performance does not improve even if training continues [40].

The training and validation procedures for specific network architectures were repeated in order to handle uncertainties of the initial weights and stopping criteria. In the preliminary investigation it was found that 10 trials were enough to find the best result. The performance efficiencies of each trial were recorded and compared. The result with the highest R-value of the training data set is considered the optimal ANN prediction for the network.

Another important task is the division of data for the network training and validation phase. The ASCE Task Committee [24], reported that ANNs are not very capable at extrapolation. Thus, in the present study, care was taken to have the training data include the highest as well as the lowest values, i.e. the two extreme input patterns. To ensure that the ANN is applicable to whole data set, about 40% of the total samples were chosen randomly from the rest of the data set for the validation phase. The stage and discharge records of the validation data are used for the rating curve prediction and comparison with ANN prediction.

Effect of including stream flow descriptive data: As mentioned in section 2.2, from descriptive parameters, the USGS is able to record the stream stage continuously. The stream width and cross-sectional area at the measuring station do not change often, thus these can be easily estimated from the stream stage and the topography of the measuring station. However, it often poses a difficult task to continuously record the mean velocity of the stream [50]. Therefore, the present study carried out the sensitivity of analysis for three input data sets: average rain fall, average temperature and average steam flow to forecast monthly stream flows at Jarahi watershed.

RESULTS AND DISCUSSION

Model structures: Three model structures were developed to investigate the impact of variable enabling/disabling of input dimension on model performance. Model 1 is enabled for average temperature data as input dimension of two stations,

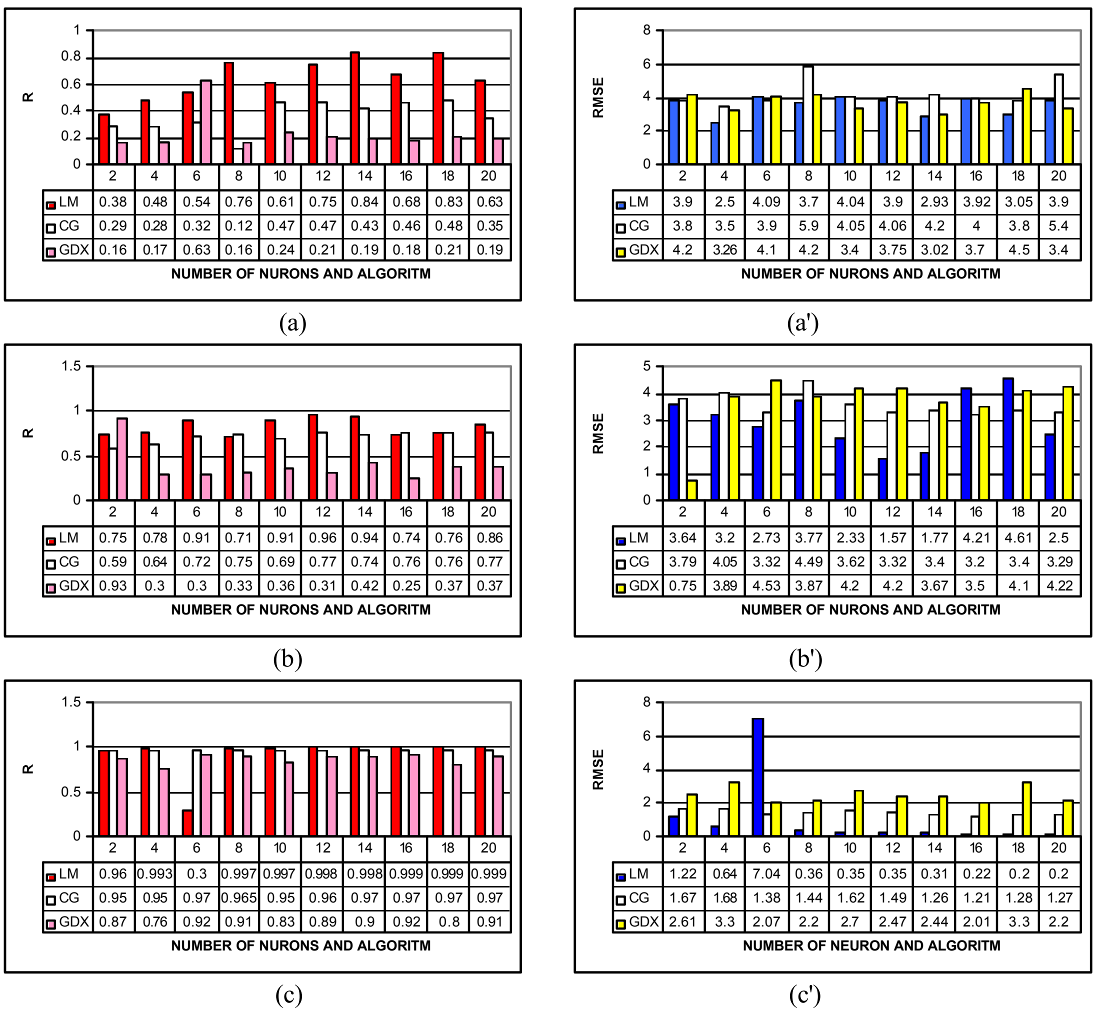

Fig. 3: Comparison of convergence speeds for the gradient descent (GDX), Conjugate Gradient (CG) and Levenberg-Marqurdt (LM) algorithms, as measured by number of neurons during validation stage for Model 1(a,a')-Model 2(b,b')-Model 3(c,c'); by(a,b,c) Correlation Coefficient (R), (a',b',c') root mean squared error (E_{RMS})

Table 1: Result of model performance level during training and validation stage

Model	Architecture		RMSE LM	RMSE CG	RMSE GDX	R LM	R CG	R GDX
Model1	2-4-1	A	4.50	4.03	3.52	0.370	0.28	0.15
		T	2.50	3.50	3.26	0.470	0.28	0.17
Model2	5-12-1	A	3.10	4.30	4.10	0.850	0.60	0.26
		T	1.57	3.32	4.20	0.960	0.77	0.31
Model3	10-20-1	A	0.26	3.19	4.52	0.997	0.84	0.61
		T	0.20	1.27	2.20	0.998	0.97	0.92

Where; RMSE is root means squared error and R is correlation coefficient. In Fig. 4a-c are indicated a comparison of predicated rainfall-runoff by different models

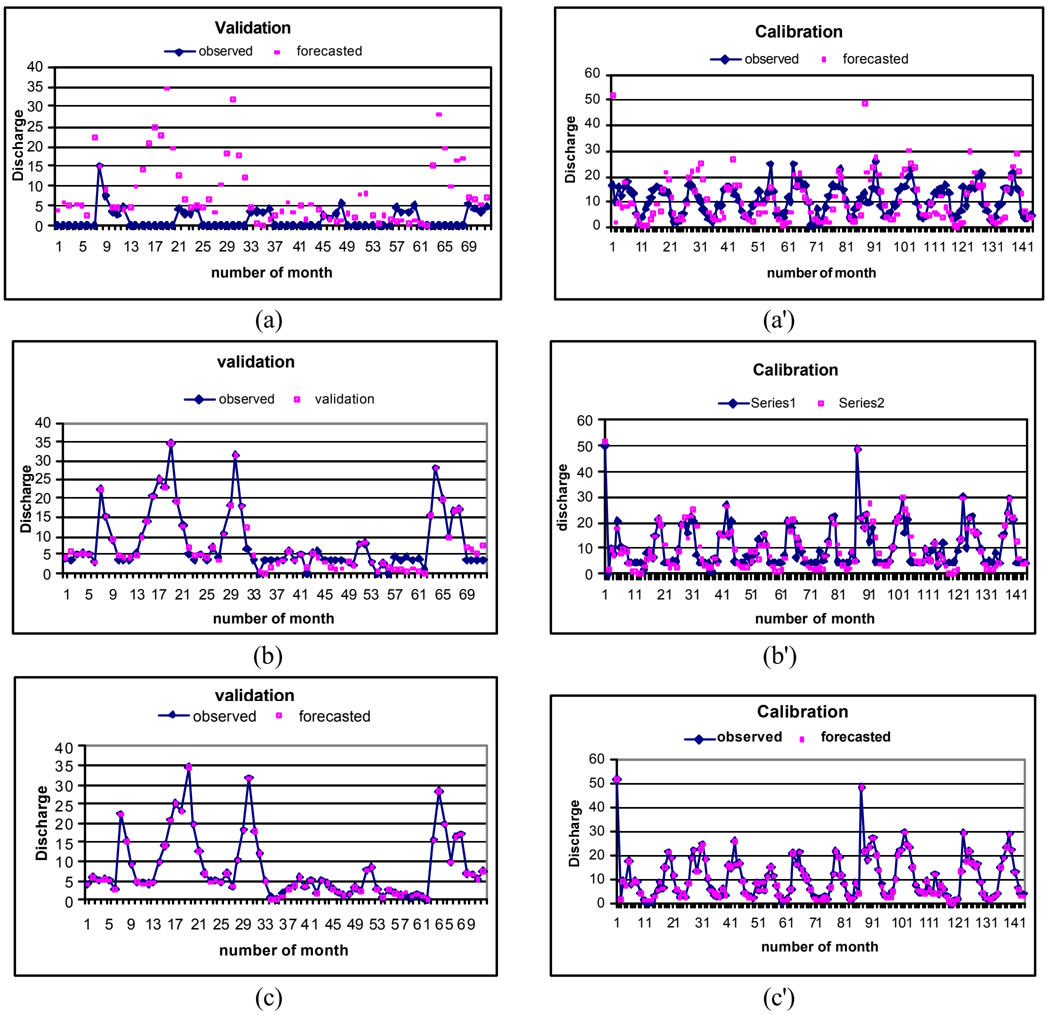

Fig. 4: Comparison of predicated rainfall-runoff for the Calibrations and validations by different models; Model 1(a,a')-Model 2(b,b')-Model 3(c,c') (a) Model 1; (b) model 2; (c) Model 3

model 2 is enabled for rain fall data as input dimension of five stations, model 3 is enabled for rainfall, average temperature and stream flow, Equations 8 to 10 represent model 1 to Model 3, respectively. Figure 3 is showing a competitive of convergence speeds

$$Q(t+1)_{sha} = f\{T_{Sho}, T_{Sha}\} \quad (8)$$

$$Q(t+1)_{sha} = f\{P_M, P_{She}, P_{Sho}, P_{Sha}, P_{Go}\} \quad (9)$$

$$Q(t+1)_{sha} = f\{P_M, P_{She}, P_{Sho}, P_{Sha}, P_{Go}, T_{Sho}, T_{Sha}, Q_m, Q_b, Q_{Go}\} \quad (10)$$

where; $Q(t+1)_{shadegan}$ is predicted rain fall-run off, for the time step of t+1; $\{Q_m, Q_b, Q_{Go}\}$ is monthly rainfall-runoff data of Mashregh, Behbahan and Gorg hydrometric station; and $\{P_M, P_{She}, P_{Sho}, P_{Sha}, P_{Go}\}$, are monthly rainfall data of Mashregh, She, Shohada, Shadegan and Gorg rain gauge stations for the time step of t and $T(t)_{sho}$, $T(t)_{sha}$, is average monthly air temperature data at Shohada and Shadegan station for the time step of t. (Table 1).

Model performance levels: Table 1 shows individual model performance levels as measured by ERMS and R and individual model architecture as represented by the number of neurons in the input, output and hidden layers. Furthermore, computed rainfall-runoff by individual models are compared with the corresponding observed values and illustrated by their graph (Fig. 4) which is indicated by the results, it can be concluded that model 1 resulted with the lowest achieved performance levels. Disabling of Shohada, Shadegan stations rain gauge data, (model 2) resulted in a considerable improvement of the performance levels. Sheilan, Shohada, Shadegan, Gorge stations rain gauge (model 3) it is possible for the rainfall, average temperature and stream flow at the time of step t.

CONCLUSION

The Artificial Neural Network (ANN) models show an appropriate capability to model hydrological process. They are useful and powerful tools to handle complex problems compared with the other traditional models. In this study, the results show clearly that the artificial neural networks are capable of model rainfall-runoff relationship in the arid and semiarid regions in which the rainfall and runoff are very irregular, thus, confirming the general enhancement achieved by using neural networks in many other hydrological fields. In this research, the influences of training algorithm efficiencies and enabling/disabling of input dimension on rainfall-runoff prediction capability of the artificial neural networks was applied. A watershed system in Ahvaz area in the south region of Iran was selected as case study. The used data in ANN were monthly hydrometric and climatic data with 17 years duration from 1983 to 2000. For the mentioned model 14 year's data were used for its development but for the validation/testing of the model 3 years data was applied. Three model structures were developed to investigate the probability impacts of enabling/disabling rainfall-runoff, rainfall, precipitation and the average air temperature input data. Efficiency of model 1 is enabled for average temperature data as input dimension with using two stations, model 2 for rain fall with using five stations and model 3 for rainfall, average temperature and stream flow data as input dimension with using six stations. Computational efficiencies, i.e. better achieved accuracy and convergence speed, were evaluated for the gradient descent (GDX), Conjugate Gradient (CG) and Levenberg-Marquardt (L-M) training algorithms. Since the L-M algorithm was shown to be more efficient than the CG and GDX algorithm, therefore it was used to train the proposed tree models. Based on the results validation stage of Root Mean Square Error (RMSE) and coefficient of determination (r) measures were: 2.5, 0.47 (model 1); 1.57, 0.96 (Model 2); 0.2, 0.998 (Model3). As indicated by the results, model 3 provided the highest performance. This was due to enabling of the rainfall, average temperature and stream flow data, resulting in improved training and thus improved prediction. The results of this study has shown that, with combination of computational efficiency measures and ability of input parameters which describe physical behavior of hydro-climatologic variables, improvement of the model predictability is possible in artificial neural network environment.

ACKNOWLEDGEMENT

The author would like to thank the Centre of Remote Sensing of the Faculty of Natural Resources, University of Agric. & Natural Res. of Sari for financial and technical support.

REFERENCES

1. Dooge, J.C.I., 1977. Problems and methods of rainfall-runoff modeling. In: Ciriani, T.A., U. Maione and J.R. Wallis (Eds.). Mathematical Models for Surface Water Hydrology: The Workshop Held at the IBM Scientific Center, Pisa. Wiley, London, pp: 71-108.
2. O'Connor, K.M., 1997. Applied hydrology I-deterministic. Unpublished Lecture Notes. Department of Engineering Hydrology, National University of Ireland, Galway.
3. Crawford, N.H. and R.K. Linsley, 1966. Digital Simulation in Hydrology: Stanford Watershed Model IV, Technical Report 10-Department of Civil Engineering, Stanford University, Stanford, CA.
4. Burnash, R.J.E., R.L. Ferral and R.A. McGuire, 1973. A generalized stream flow simulation system, report 220. Jt. Fed.-State River Forecasting Center, Sacramento, CA.
5. Zhao, R.J., Y.L., Zhang, L.R., Fang, X.R., Liu and Q.S., Zhang, 1980. The Xinanjiang model, in Hydrological forecasting. Proceedings of the Oxford Symposium, IAHS, 129: 351-356.

5. Jain, S.K. and D. Chalisgaonkar, 2000. Setting up stage-discharge relations using ANN. Journal of Hydrologic Engineering, 5 (4): 428-433.
6. Zhao, R.J., 1992. The Xinanjiang model applied in China. J. Hydrol., 135: 371-381.
7. Zhao, R.J. and X.R., Liu, 1995. The Xinanjiang model. In: Singh, V.P. (Ed.), Computer Models of Watershed Hydrology. Water Resources Publications, Littleton, CO.
8. O'Connell, P.E., J.E. Nash and J.P. Farrell, 1970. River flow forecasting through conceptual models. Part 2. The Brosna catchment at Ferbane. J. Hydro., 10: 317-329.
9. Tan, B.Q. and K.M. O'Connor, 1996. Application of an empirical infiltration quation in the SMAR conceptual model. J. Hydrol., 185: 275-295.
10. Sugawara, M., 1961. On the analysis of runoff structure about several Japanese rivers. Jap. J. Geophys., 2 (4): 1-76.
11. Sugawara, M., 1995. In: Singh, V.P. (Ed.). Tank model, inComputer Models of Watershed Hydrology. Water Resources Publications, Littleton, CO.
12. Kitanidis, P.K. and R.L. Bras, 1980a. Adaptive filtering through detection of isolated transient errors in rainfall-runoff models. Water Resource. Res., 16 (4): 740-748.
13. Kitanidis, P.K. and R.L. Bras, 1980b. Real-time forecasting with a conceptual hydrological model. Water Resource. Res., 16 (4): 740-748.
14. Hsu, Kuo-lin, H.V. and Gupta, S. Sorooshian, 1995. Artificial neural network modeling of the rainfall-runoff process. Water Resour. Res., 31 (10): 2517-2530.
15. Smith, J. and R.B. Eli, 1995. Neural-network models of rainfall-runoff process. J. Water Resour. Plng. Mgmt., ASCE, 4 (3): 232-239.
16. Dawson, C.W. and R. Wilby, 1998. An artificial neural network approach to rainfall-runoff modeling. Hydro. Sci., 43 (1): 47-66.
17. Tokar, A.S. and P.A. Johnson, 1999. Rainfall-runoff modeling using artificial neural network. J. Hydr. Eng., ASCE, 4 (3): 232-239.
18. Sajikumar, N. and B. Thandaveswara, 1999. A nonlinear rainfall runoff model using an arti.cial neural network. Journal of Hydrology, 216: 32-35
19. Tokar, A.S. and M. Markus, 2000. Precipitation-runoff modeling using artificial neural networks and conceptual models. J. Hydro. Eng., ASCE, 5 (2): 156-161.
20. Zhang, B. and R.S. Govindaraju, 2003. Geomorphology-based artificial neural networks (GANNs) for estimation of direct runoff over watersheds. J. Hydrol., 273 (1-4): 18-34.
21. Cannon, A.J. and P.H. Whitfield, 2002. Downscaling recent stream-flow conditions in British Columbia. Canada Using Ensemb le Neural Networks. J. Hydro., 259: 136-151.
22. Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modeling. J. Hydrol., 199: 272-294.
23. ASCE, Committee on application of Artificial Neural Networks in Hydrology, 2000b. Artificial neural networks in hydrology. II: hydrologic applications. J. Hydrologic Eng., 5 (2): 115-123.
24. ASCE, Task Committee on application of Artificial Neural Networks in Hydrology, 2000a. Artificial neural networks in hydrology. I: preliminary concepts. J. Hydro. Eng., 5 (2): 124-137.
25. Gautam, M.R., K. Watanabe and H. Saegusa, 2000. Runoff analysis in humid forest catchment with artificial neural network. J. Hydrol., 235: 117-136.
26. Lee, D.S., C.O. Jeon, J.M. Park and K.S. Chang, 2002. Hybrid neural network modeling of a full-scale industrial wastewater treatment process. Biotechnology. Bioeng., 78 (6): 670-682.
27. Van Can, H.J.L., H.A.B., Braake, C. Hellinga, K.C.A.M, Luyben and J.J. Heijnen, 1997. An efficient model development strategy for bioprocesses based on neural networks in macroscopic balance. Biotechnol. Bioeng., 54: 549-566.
28. Zhao, H., O.J. Hao, T.J. McAvoy and C.-H. Chang, 1997. Modeling nutrient dynamics in sequencing batch reactor. J. Environ. Eng., 123: 311-319.
29. Werbos, P., 1974. Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Harvard University, Harvard.
30. Rumelhart, E., G. Hinton and R. Williams, 1986. Learning internal representations by error propagation. Parallel Distributive Process, 1: 218-362.
31. 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 (2): 163-181.
32. Jain, A., K.P. Sudheer and S. Srinivasulu, 2004. Identification of physical processes inherent in artificial neural network rainfall-runoff models. Hydro. Process, 118 (3): 571-581.
33. Sudheer, K.P. and A. Jain, 2004. Explaining the internal behavior of artificial neural network river flow models. Hydrol. Process, 118 (4): 833-844.
34. Rajurkar, M.P., C. Kothyari and U. Chaube, 2004. Modeling of daily rainfall-runoff relationship with artificial neural network. Journal of Hydrology, 285: 96-113.

35. Chiang, Y.-M., L.C., Chang and F.J. Chang, 2004. Comparison of static-feedforward and dynamic-fedback neural networks for rainfall-runoff modeling. Journal of Hydrology, 290: 297-311.
36. Maier, H.R. and G.C. Dandy, 2000. Neural networks for prediction and forecasting of water resources variables: A review of modeling issues and applications. Environmental Modeling and Software, 15: 101-123.
37. Rumelhart, D.E., G. Hinton and R. Williams, 1986. Learning representations by back-propagating errors. Nature, 323: 533-536.
38. Antcil, F. and N. Lauzon, 2004. Generalisation for neural networks through data sampling and training procedures, with applications to stream of predictions. Hydrology and Earth Systems Science, 8 (5): 940-958.
39. De Vos, N.J. and T.M. Rientjes, 2005. Constraints of artificial neural networks for rainfall-runoff modeling: Trade-offs in hydrological state representation and model evaluation. Hydrology and Earth Systems Science, 9: 111-126.
40. Hagan, M.T., H.B. Demuth and M. Beale, 1996. Neural Network Design. 2nd Edn. PWS Publishing Company, Boston, USA runoff model using an artifcial neural network. Journal of Hydrology, 216: 32-35.
41. Salehi, F; S. Prashar; S. Amin; A. Madani; S. Jebelli; H. Ramaswamy; C. Drury 2000. Prediction of annual nitrate-N losses in drainage with artificial neural networks. Transaction of the ASAE, 43(5), 1137-1143
42. Burden, F.R., R.G. Brereton, P.T. Walsh, 1997. Cross-validatory selection of test and validation sets in multivariate calibration and neural networks as applied to spectroscopy. Analyst 122 (10), 1015-1022
43. Legates, D.R. and G.J. McCabe, 1999. Evaluating the use of goodness-of-fit measures in hydrologic and hydro-climatic model validation. Water Resources Research, 35 (1): 233-241.
44. Schaap, M.G. and F.J. Leij, 1998. Database related accuracy and uncertainty of pedotransfer functions. Soil Science, 163: 765-779.
45. Birikundavyi, S., R. Labib, H.T. Trung and J. Rousselle, 2002. Performance of neural networks in daily streamflow forecasting. Journal of Hydrologic Engineering, 7 (5): 392-398.
46. Principe, J.C., N.R. Euliano and W.C. Lefebvre, 1999. Neural and Adaptive Systems: Fundamentals through Simulations. Wiley, New York.
47. Ray, C. and K.K. Klindworth, 2000. Neural networks for agrichemical vulnerability assessment of rural private wells. Journal of Hydrologic Engineering, 5 (2): 162-171.
48. Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Macmillan, New York.
49. MathWorks, Inc., 2001. MATLAB. 3 Apple Hill Drive, Natick, MA, USA.