Research Journal of

Environmental Sciences

ISSN 1819-3412

Research Journal of Environmental Sciences 2 (5): 353-365, 2008
ISSN 1819-3412
© 2008 Academic Journals Inc.

Comparative Analysis of Training Methods and Different Data for the Rainfall-Runoff Predication Using Artificial Neural Networks

[1]Karim Solaimani and [2]Zahra Darvari
[1]GIS and RS Centre, University of Mazandaran, Iran
[2]Department of Watershed Management, University of Mazandaran, Iran

Abstract: Hydrology and climatic monthly data's influence on training of Artificial Neural Networks (ANNs) for monthly rain fall prediction is investigated. For improved computed performances, efficiencies of the Conjugate Gradient (CG) and Levenberg-Marquardt (L-M) training algorithms are compared. The rain fall-run off influence is studied for a watershed in Northern Iran, representing a continuous rain fall-run off with stream flow regime occurring. The used data in ANN was hydrometric and climatic monthly data with 31 years duration from 1969 to 2000. For the mentioned model 27 year's data were used for its development but for the validation/testing of the model 4 years data was applied. Based on the results, the L-M algorithm is more efficient than the CG algorithm, so it is used to train six ANNs models for rain fall-runoff prediction at time step t+1 from time step t input. The used network in this study was MLP with B.P. (back propagation) algorithm. Model 1 uses enabled rain fall data as input dimension with use tree station, Model 2 uses enabled rain fall and average temperature, Model 3 uses enabled rain fall, average temperature and stream flow at time step t-1 and Model 4 uses enabled rain fall and stream flow at time step (t, t-1, t-2), Model 5 uses enabled rain fall and stream flow at time step (t, t-1, t-2.t-3) Model 6 uses enabled rain fall, average temperature and stream flow at time step (t-1, t-2). Validation stage Root Mean Square Error (RMSE), Root Mean Absolute Error (RMAE) and Correlation Coefficient (R) measures are: 0.07, 6×10^{-4}, 0.99 (Model 1); 0.1, 9×10^{-4}, 0.99 (Model 2); 0.01, 9×10^{-5}, 1(Model 3); 0.005, 6×10^{-5}, 1(Model 4); 0.001, 0.7×10^{5}, 1 (Model 5); 0.001, 6×10^{-5}, 1 (Model 6) and, respectively. The influence of rain fall and stream flow at time step (t, t-1, t-2) on improved Model 4 performance is discussed.

Key words: Rainfall-runoff, ANN, training algorithms, kasilian watershed, Iran

INTRODUCTION

Runoff data is the important information for the design and planning of many water resources engineering projects. Many hydrologists devote themselves to develop rainfall-runoff models to estimate runoff. The rainfall-runoff process, which involves many mechanisms, is known as a highly complicated and nonlinear phenomenon. Difficulties exist in the modeling of the rainfall-runoff process. Thus, an accurate and easily-used rainfall-runoff model that can appropriately model the rainfall-runoff process is of strong demand. In the rainfall-runoff process, there is a temporal dependency between the rainfall and the runoff. The temporal dependency can be described using the linear system model (Chow *et al*., 1988):

$$Q(t) = \int_0^t I(\tau)u(t-\tau)d\tau \qquad (1)$$

Corresponding Author: Karim Solaimani, GIS and RS Centre, University of Mazandaran, Iran
Tel: (+98) 9111521858

Where:

Q(t)	=	Output at time
t, I(τ)	=	Input at time
s and u(t _ τ)	=	Impulse response function of the system

In the modeling of the rainfall-runoff process, the excess rainfall is the input of the system, the direct runoff is the output of the system and the Instantaneous Unit Hydrograph (IUH) is the impulse response function. From Eq. 1, the output at time t is produced by the convolution integral of the impulse response function and the input up to time t. In actual applications, the discrete form of Eq. 1 is more practical. The discrete form of Eq. 1 is:

$$Qt = \sum_{i=1}^{L} P_{t-i} + {}_{1}u_{i} \qquad (2)$$

Notation

A_k	:	Kth stage second derivative of G(x)
a_i	:	Network output (i, 1/4 1, 2, y, S)
b_i	:	Bias term (i, 1/4 1, 2, y, S)
E_{MA}	:	Computed mean absolute error
E_{RMA}	:	Computed root mean absolute error (RMAE)
E_{RMS}	:	Computed root mean squared error (RMSE)
f	:	Transfer (activation) function
f_l	:	Function predicting stream flow at time t+1 based on information of time t
G(x)	:	Function to be minimised
g_k	:	kth stage first derivative of G(x)
N	:	Sample size
N_i	:	Sum of individual weights (input to f) (i, 1/4 1, 2, y, S)
$P(t)_{darzi}$	:	Darzikola rainfall at time step t (mm)
$P(t)_{sang}$	:	Sangedeh rainfall at time step t (mm)
$P(t)_{kaleh}$	:	Kaleh rainfall at time step t (mm)
p_i	:	Network input (i, 1/4 1, 2, y, S)
p_o	:	Observed air temperature (1C)
p_k	:	Search vector
p_{max}	:	Maximum air temperature (1C)
p_{min}	:	Minimum air temperature (1C)
p_n	:	Scaled temperature
$Q(t)_{Valik}$	:	Valikbon predicted stream flow at time step t (m_3/s)
$Q(t+1)_{Valik}$	:	Valikbon observed stream flow at time step t (m_3/s)
$Q_{obs(m)}$	:	m_{th} observed stream flow value (m_3/s)
$Q_{sim(m)}$	:	m_{th} simulated stream flow value (m_3/s)
~Q	:	Mean observed stream flow value (m_3/s)
R	:	Number of inputs
S	:	Number of neurons
$T(t)_{Darzi}$	:	Darzikola average air temperature (1C)
$T(t)_{sang}$	:	Sangedeh average air temperature (1C)
W	:	Matrix of weights of S neurons (rows) and R inputs (columns)

$W_{i,j}$	:	An element of matrix w (i 1/4 1, 2, y, S; j1/4 1, 2, y, R)
X_k	:	Current estimation point to be minimized at k_{th} stage
x_{k+1}	:	k+1th stage estimated point
ak	:	kth stage learning rate
ΔDG(x)	:	Rate of change of G(x)
ΔDxk	:	kth stage rate of change of xk
μ_k	:	kth stage trial value to minimise G(x)

where, Qt is the direct runoff, Pt_i+1 is the excess rainfall at time t- i + 1 and Ui is the IUH of discrete form. The L in Eq. 2 is a crucial term. The value of L indicates the duration of the influence of an excess rainfall lasting. In other words, L indicates the extent of the temporal dependency between the rainfall and the runoff. The temporal dependency between the rainfall and the runoff plays a significant role for modeling the rainfall-runoff process. Therefore, the value of L should be determined carefully. In recent years, Artificial Neural Networks (ANN) are widely used for modeling the rainfall-runoff process. ANN can model the highly nonlinear phenomenon in a simple way. Two different types of ANN are frequently used to model the rainfall-runoff process. One is the static networks including the multilayer perceptron (MLP) and the Radial Basis Function network (RBF) (Lin and Chen, 2004; Rajurkara *et al*., 2004). The other one is the dynamic network such as the real-time recurrent network (Chang *et al*., 2002, 2004; Chiang *et al*., 2004). One reason of the popularity of ANN is that it is embedded with an algorithm for estimating the synaptic weights of the neurons (i.e., the parameters of ANN). That is, ANN based rainfall-runoff models can be calibrated using the embedded algorithm easily. Another reason of the popularity of ANN based rainfall-runoff models is that the model's configuration can be adjusted according to the characteristics of watersheds. These are two advantages of ANN based models over the traditional models. In the theory of ANN, once the temporal dependency between the input and the output of a system is presented, then it is said that the system has memory (Haykin, 1999). In this study, the term memory is used to indicate the temporal dependency between the rainfall and the runoff for prediction. The rainfall-runoff process is a typical system that has memory. The magnitude of the temporal dependency between the rainfall and the runoff is referred to as the memory length of the rainfall-runoff process. In the ANN based rainfall-runoff models, the role of the memory length is analogous to that of the l in Eq. 2.When one uses ANN to model the rainfall-runoff process, the construction of the memory and the evaluation of the memory length of the rainfall-runoff process are important concerns. While many studies have investigated different approaches including artificial neural networks for infilling data gaps of stream flows. Serial correlations of daily or monthly stream flows provide a basis for successful application of simple statistical methods. Conversely, the non stationary behavior of precipitation patterns does not allow a proper application of simple linear regression or interpolation methods (Creutin *et al*., 1997). Consequently, various statistical methods have been proposed for infilling missing stream flow values including ANN approach (Khalil *et al*., 2001). In the rainfall modeling context used synthetically generated rainfall storms to calibrate an ANN model and generated plausible rainfall events. Luck *et al*. (2000) identified an optimal set of spatio-temporal inputs for an ANN rainfall forecasting model using 15-min-intervals rainfall records. Recently, Abebe *et al*. (2000) compared fuzzy rule-based and ANN models for reconstructing missing precipitation events. In the last 15 years, ANNs have shown promising potential in various hydrologic applications (ASCE Task Committee, 2000a, b; Maier and Dandy, 2000). More recent hydrologic applications of ANNs are summarized in Cigizoglu (2005), Cigizoglu and Kisi (2006), Coulibaly *et al*. (2005) and Dibike and Coulibaly (2006). By considering complexity of the phenomena involved there is a strong need to explore alternative solutions through modeling direct relationship between the input and output data without having the complete physical understanding of the system. While data-driven models do not provide any information on the physics of the hydrologic processes, they are very useful for river flow forecasting where the main concern is accurate predictions of runoff (Nayak *et al*., 2005). Due to the ability of ANNs in modeling complex nonlinear systems, successful applications of these method in water resources modeling have been widely reported, such as for rainfall-runoff simulation (Campolo *et al*., 1999a, b; Sudheer *et al*., 2002; Chang *et al*., 2002; Jain and Srinivasulu, 2004), river flow forecasting (Imrie *et al*., 2000; Hu *et al*., 2001; Kumar *et al*., 2004) and rainfall forecasting (Kuligowski and Barros, 1998; Luk *et al*., 2001; Rami'rez *et al*., 2005). And Comparison of neural nor infilling missing daily weather records (Coulibaly and Evora, 2007) The study focused on Comparative analysis of training methods and different data's for rainfall-runoff predication with use ANNs. In this respect, computational efficiencies measured in terms of better achieved accuracy and convergence speed of the Lavenberg-Marquardt (L-M) and Conjugate Gradient (CG) algorithms in network training are evaluated and compared. Furthermore, the effects of enabling/disabling of stream flow, temperature, monthly rainfall and (as parts of the input dimension) on improved rainfall-runoff predication are also evaluated. A watershed system with continuous regime of stream flow in Northern Iran is used as a case study.

MATERIALS AND METHODS

Overview of Artificial Neural Networks

An Artificial Neural Network (ANNs) is a mathematical model which is able to emulate the behavior of the human brain. An ANN model is appealing to use since it has the capacity to extract patterns from the observed data without assumptions about the underlying relationships. Such non-linear models can approximate unknown mappings and their derivatives by a three-layer structure; input, hidden and output layers (Hornik *et al*., 1989). The feed forward architecture, also know as the multi layer preceptron, MLP is the most popular net-work architecture used by researchers in hydrology for stream flow modeling (Jain *et al*., 1999; Maier and Dandy, 2000; Gaume and Gosset, 2003). The topology of feed forward ANNs consists of several nodes as the basic unit of information processing, which resemble the structure of neurons. The nodes can be linked together through a number of layers, identified as input, hidden and output layers. The hidden layer itself can be a single layer or a multilayer design. Figure 1 shows a single layer network with S neurons (nodes). Input into the network is represented by p1, p2, y, pR. The network outputs are represented by a1, a2, ..., aS.

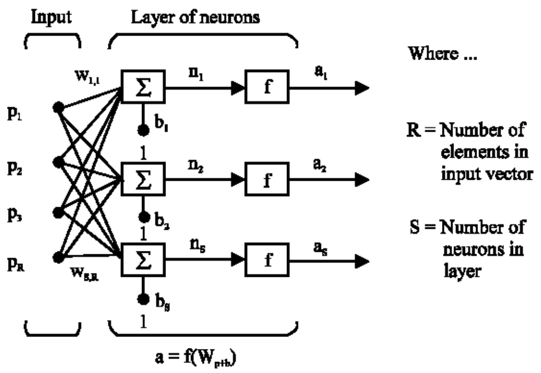

Fig. 1: A feed forward artificial neural network (Hagan *et al*., 1996)

Every input is connected to all of the neurons. As illustrated by Fig. 1, the single layer includes the weight matrix, summers and the bias terms where, p1, p2, y, pR network inputs; wi, j., (i = 1, 2, …, S; j = 1, 2, …, R) element of matrix of weights of S neurons and R inputs; b1, b2, …, bS bias terms; n1, n2, …, nS sum of individual weights and bias terms (input to f); f transfer function; a1, a2, …, aS network outputs b1, b2, …, bS and the activation function also called transfer function, f. The level of information processing between nodes is evaluated by a weight, wi, j, which is an element of the weight matrix w, consisting of S rows (neurons) and R columns (inputs). In a preceptron the nodes use these weights to calculate a linear combination of the information that comes to them to which a constant bias bi (i = 1, …, S) is added and is identified as ni. Each calculated values of ni is an input into the activation function. The activation function is used to take any real input into an enclosed subset. The result of this operation is the output of the node, which will become an input to another node situated in the next layer. The weights associated with the connections that arrive at a node, including the bias, are the coefficients of the linear combination. Generally, the same activation function is used for all nodes of each layer, but it is possible to use a different activation function for each node. For a feed forward ANN, the available data are usually divided into sub-sets for use in the training/calibration and validation/testing stages. In the training stage the network learns the process of developing output values from the input set. Network learning is achieved by using a learning algorithm and adjustment of the designated weights, used in conjunction with the training data set. The testing stage constitutes the final performance evaluation of the developed ANN model by using the testing data set. This phase evaluates the capability of the model for producing acceptable output values through previously established input-output relationships. Network performance is checked to avoid over-fitting or over-training. A separate set of data maybe needed for evaluation of over-training situation.

Network Training Algorithms

The Back-Propagation (BP) algorithm has been the most commonly used training algorithm. Rainfall-runoff modeling with limited data. In a study by Chiang *et al*. (2004), 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 because of its superior efficiency and high convergence speed (Antcil and Lauzon, 2004; de Vos and Rientjes, 2005). 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. The follow-ing shows the corresponding equation (Hagan *et al*., 1996):

$$\Delta\chi_k = \chi_{k+1} - \chi_k = \alpha_k p_k \tag{3}$$

where, x_k is the current estimation point for a function G(x) to be minimized at the kth stage, p_k is the search vector and a_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 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 (Hagan *et al*., 1996). On the other hand, the L-M algorithm is viewed as a very efficient algorithm with a high convergence speed. In correspondence with Eq. 1 the following equation is used as the function minimization routine in the L-M procedures (Hagan *et al.*, 1996):

$$\chi_{k+1} = \chi_k - A_k^{-1} g_k \tag{4}$$

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 as Hagan *et al.* (1996)

$$\chi_{k+1} = \chi_k - \frac{1}{2\mu_k} \nabla G(\chi) \tag{5}$$

In the above 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., $\mu_k = 0.0001$). If a step does not yield a smaller value for G(x), the step is repeated with μ_k multiplied by some 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 mkis 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 upper sub-watershed of the Kasilian watershed system (Fig. 2) for the evaluation of the predication rain fall -run off. Modeling capabilities of the CG and the L-M algorithm were compared in terms of their abilities in network training. The influences of monthly rainfall, stream flow and air temperature data as different input dimensions were also evaluated. The study area of Kasilian watershed with a drainage area of 3360 km^2 is located in Mazandaran province, Northern part of Iran, with an important role for the agricultural and industrial activities.

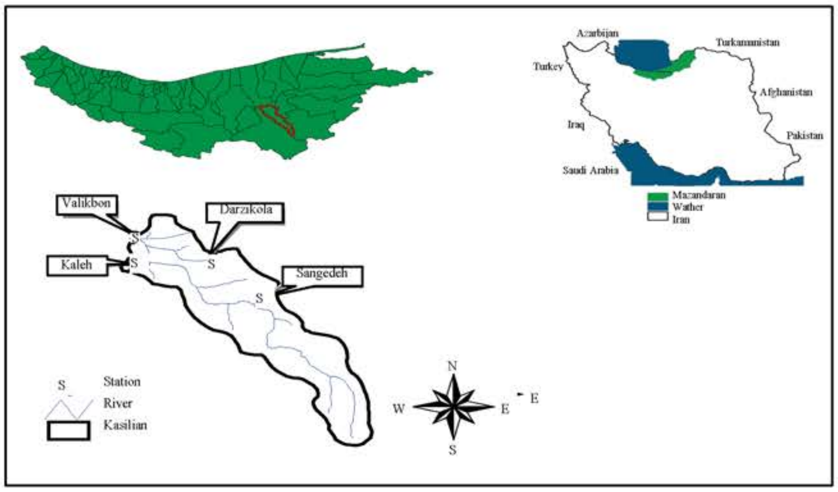

Fig. 2: Geographical location of the study area in Iran

The sources of water are mostly originated from snowmelt, springs and also individual rain events. In spite of a permanent flow system, the rainy season is limited to 9 months duration (October-April) with a mean annual precipitation of near to 800 mm. Rainfall events are usually occurred during the last days but rainfall durations of 1 day or less is also recorded. Monthly optimized precipitation occurs in August to September and monthly minimum precipitation recorded in February to March. This watershed is included two sub-basin of Talarsarband and Tilehsiah which are following to the north direction. Figure 2 shows the location of the hydrometric stations of the whole basin. The used data with 31 years duration (1969-2000) for this study included monthly rainfall and discharge of Sangedeh, Darzikola and Keleh stations.

Data Preparation

The used discharges data of Kasilian basin were limited with a 31 year duration, which was used for a purposed model development to considering of different combinations of inputs variables, e.g., rainfall, stream flow and average air temperature. For all of the mentioned models available data were separated in 80% for training and 20% for validation. Since in this research the number of data points was more than the number of used parameters by the network (weights and biases), therefore additional analysis for the detection of network overtraining was seemed dispensability. So data could be divided in two parts for use in the training and validation stages, i.e., cross-validation analysis to stop overtraining was not required (Salehi *et al*., 2000). 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 (Maier and Dandy, 2000). In this study, the data were scaled in the range of -1 to +1, based on the following equation:

$$p_n = 2\left(\frac{p_0 - p_{min}}{p_{max} - p_{min}}\right) - 1 \qquad (6)$$

where, p_0 is the observed data, p_n is the scaled data and p_{max}, p_{min} are maximum and minimum observed data points. In order to provide the consistency of analysis, the above equation was used to scale the average air temperature, stream flow and rainfall data, then the unit of the scaled p_n would correspond to individual data series.

The Artificial Neural Network Structure

Network structure includes input and output dimensions, the number of hidden layers, number of hidden neurons and model efficiency calculations. In the present study, input dimension includes monthly stream flow, rainfall and average air temperature data for the time step of t. Output dimension is the predicted stream flow at the time of t+1. Only one hidden layer was used which has been shown to be sufficient in a number of studies (Rajurkara *et al*., 2004; Chiang *et al*., 2004; Maier and Dandy, 2000). The appropriate number of neurons in the hidden layer is determined by using the constructive algorithm (Maier and Dandy, 2000), by increasing the number of neurons from 1 to 30. The log-sigmoid, tangent-hyperbolic and linear activation functions are also used. The ANN model for stream flow evaluation was written in the MATLAB environment, version 7. The L-M and the CG algorithms were evaluated for the network training so that the algorithm with better achieved accuracy and convergence speed could be selected. To provide an adequate training, network efficiency was evaluated during the training and validation stages, as suggested by Rajurkara *et al*. (2004). In this case, if the calculated errors of both stages are trended to decreasing, then the training period will increase. This is continued to the point of the training stage error starting to decrease, but the validation stage of 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 (Rajurkara *et al*., 2004; Chiang *et al*., 2004).

The Used Error Functions

Root means squared error (RMSE):

$$E_{RMS} = \sqrt{\frac{1}{N}\sum_{m=1}^{N}(Q_{sim(m)} - Q_{obs(m)})^2} \tag{7}$$

where, ERMS is the root mean squared error; $W_{sim(m)}$ and $Q_{obs(m)}$ are the simulated and observed stream flow values, respectively and N is the sample size.

Root mean absolute error (RMAE) is given by

$$E_{RMA} = \frac{E_{MA}}{\tilde{Q}} \tag{8}$$

$$E_{MA} = \frac{1}{N}\sum_{m=1}^{N}\left|Q_{sim(m)} - Q_{obs(m)}\right| \tag{9}$$

where, ERMA and EMA are the root mean absolute error and mean absolute error, respectively; Correlation Coefficient (R)

$$R = \frac{\sum_{i=1}^{n}(Q_{obs(m)} - \bar{Q}_{obs(m)})(Q_{sim(m)} - \bar{Q}_{obs(m)})}{\sqrt{\sum_{i=1}^{n}(Q_{obs(m)} - \bar{Q}_{obs(m)})^2\sum_{i=1}^{n}(Q_{sim(m)} - \bar{Q}_{aim(m)})^2}} \tag{10}$$

where, $\bar{Q}_{obs(m)}$ is the mean of the observed flow and $\bar{Q}_{sim(m)}$ the mean of the modeled flow, $Q_{oms(m)}$ is the observed flow and $Q_{sim(m)}$ the modeled flow.

RESULTS AND DISCUSSION

Model Structures

Six model structures were developed to investigate the impact of variable enabling/disabling of input dimension on model performance. Model 1 is enabled for rain fall data as input dimension of tree stations, model 2 is enabled for rain fall and average temperature, model 3 is enabled for rainfall, average temperature and stream flow at the time step of t-1 and model 4 is enabled for rain fall and stream flow at the time step of (t, t-1, t-2), model 5 is enabled for rain fall and stream flow at the time step of (t, t-1, t-2.t-3) and finally, model 6 is enabled for rainfall, average temperature and stream flow at the time step of (t-1, t-2). Equations 11 to 16 represent model 1 to Model 4, respectively. Table 1 is showing a competitive of convergence speeds.

$$Q(t+1)_{valik} = f\{p(t)_{darzi}, p(t)_{sang}, p(t)_{kale}\} \tag{11}$$

$$Q(t+1)_{valik} = f\{P(t)_{darzi}, P(t)_{sang}, P(t)_{kale}, T(t)_{darzi}, T(t)_{sang}\} \quad (12)$$

$$Q(t+1)_{valik} = f\{P(t)_{darzi}, P(t)_{sang}, P(t)_{kale}, T(t)_{darzi}, T(t)_{sang}, Q(t-1)_{valik}\} \quad (13)$$

$$Q(t+1)_{valik} = f\{P(t)_{darzi}, P(t)_{sang}, P(t)_{kale}, Q(t)_{valik}, Q(t-1)_{valik}, Q(t-2)_{valik}\} \quad (14)$$

$$Q(t+1)_{valik} = F\{P(t)_{darzi}, P(t)_{sang}, P(t)_{kale}, Q(t)_{valik}, Q(t-1)_{valik}, Q(t-2)_{valik}, Q(t-3)_{valik}\} \quad (15)$$

$$Q(t+1)_{valik} = F\{P(t)_{darzi}, P(t)_{sang}, P(t)_{kale}, T(t)_{darzi}, T(t)_{sang}, Q(t-1)_{valik}, Q(t-2)_{valik}\} \quad (16)$$

where, $Q(t+1)_{valik}$ is predicted rain fall -run off, for the time step of t+1; $Q(t)_{valik}$ is monthly rainfall- runoff data of Valikbon hydrometric station and $P(t)_{darzi}$, $P(t)_{sang}$, $P(t)_{kale}$ are monthly rainfall data of Darzikola, Sangedeh and Keleh rain gauge stations for the time step of t and $T(t)_{darzi}$, $T(t)_{sang}$ is average monthly air temperature data at Darzikola and Sangedeh station for the time step of t (Table 2).

where, RMSE is root means squared error; ERMA is root mean absolute error and R is correlation coefficient. In Fig. 3a-f are indicated a comparison of predicated rainfall- runoff by different models.

Table 3 is showing a comparison of convergence speeds for Levenberg- Maqurdt and the gradient descent algorithms, as measured by number of neurons during a validation stage for model 5.

Table 1: Result of convergence speeds for model 5 performance with different number of epochs

		No. of epochs				
		20	40	60	80	100
LM	RMSE	0.212	0.071	0.006	0.003	0.001
	RMAE	0.0014	0.0006	0.0006	0.00003	0.000007
CG	RMSE	2.1	1.41	2.06	0.91	0.74
	RMAE	0.018	0.019	0.025	0.009	0.0054

Table 2: Result of model performance level during training and validation stage

		RMSE		ERMA		R	
Model	Rrchitechture	Training	Validation	Training	Validation	Training	Validation
Model 1	3181	3.21	0.070	0.02	0.0006	0.47	0.99
Model 2	5111	0.47	0.100	0.003	0.0009	0.997	0.9998
Model 3	6201	0.03	0.010	0.0003	0.000094	0.99	1
Model 4	7201	0.01	0.005	0.0001	0.00006	0.99	1
Model 5	7121	0.02	0.001	0.0001	0.000007	0.99	1
Model 6	7141	0.03	0.001	0.07	0.0006	0.99	1

Table 3: Result of model 5 performance with different number of neurons

		No. of neurons									
		2	4	6	8	10	12	14	16	18	20
LM	RMSE	0.435	0.135	0.375	8.870	0.029	0.013	0.023	0.010	0.005	8.580
	R	0.990	0.990	0.990	0.990	0.990	0.990	0.990	0.990	0.990	0.810
	RMAE	0.270	0.090	0.210	0.015	0.017	0.008	0.018	0.007	0.023	0.075
CG	RMSE	1.880	2.500	0.920	0.770	0.780	0.740	1.250	1.880	0.750	0.750
	R	0.820	0.970	0.980	0.980	0.980	0.970	0.830	0.980	0.970	0.980
	RMAE	1.166	2.100	0.590	0.001	0.530	0.740	1.010	1.450	3.190	0.007

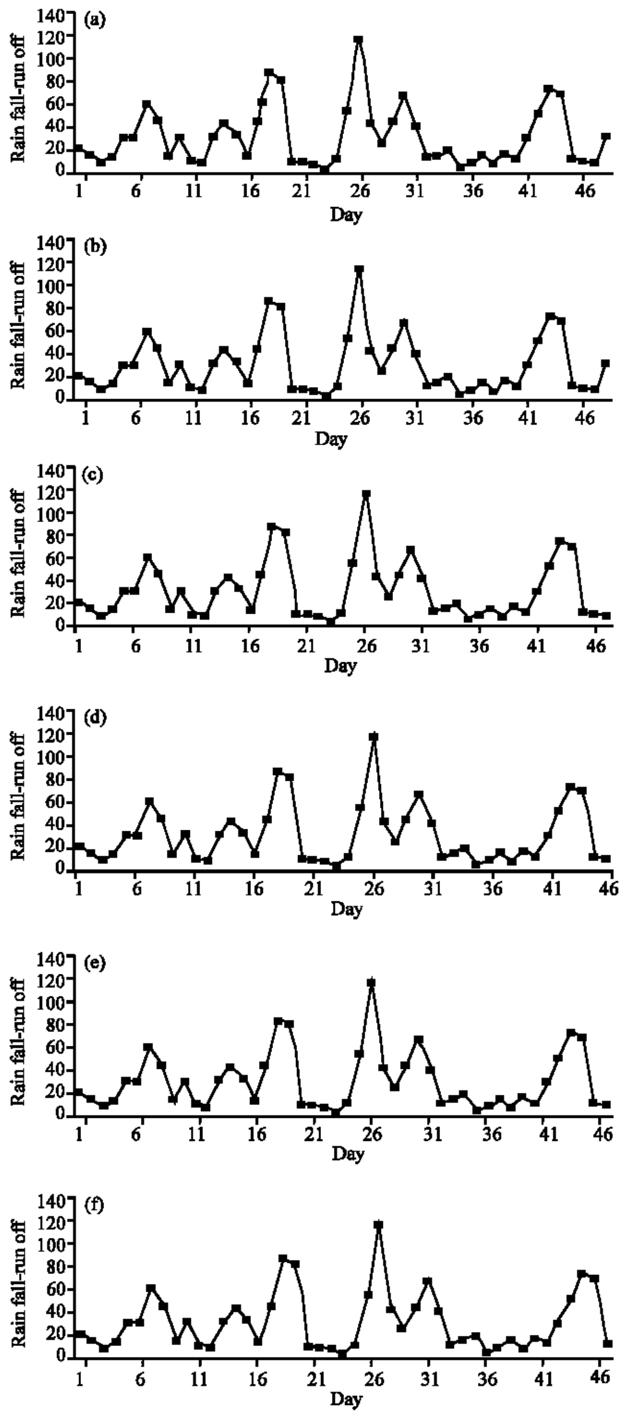

Fig. 3: Comparison of predicated rainfall- runoff by different models, (a) Model 1, (b) model 2, (c) Model 3, (d) Model 4, (e) Model 5 and (f) Model 6

Model Performance Levels

Table 1 shows individual model performance levels as measured by ERMS, EMSE and R and individual model architecture as represented by the number of neurons in the input, output and hidden layers. Furthermore, computed rainfall-ranoff by individual models are compared with the corresponding observed values and illustrated by their graph (Fig. 3) which is indicated by the results, it can be concluded that model 1 resulted with the lowest achieved performance levels. Disabling of Darzikola, Sangede, Kaleh stations rain gauge data (model 2) resulted in a considerable improvement of the performance levels. Darzikola, Sangede, Keleh stations rain gauge and using average temperature (model 3) it is possible for the rainfall, average temperature and stream flow at the time of step t-1 and model 4 representing for the rain fall and stream flow at the time step of (t, t-1, t-2). Model 5 is further improved the achieved performances representing for rain fall and stream flow at the time step of (t, t-1, t-2.t-3). Finally using model 6 is possible for rainfall, average temperature and stream flow at the time step of (t-1, t-2).

Levenberg-Marquardt and Conjugate Gradient Algorithms

Based on the gained information from model 5, computational efficiencies of the L-M and the CG algorithms were compared. Figure 3 shows achieved performances for determination of the optimal number of neurons and epochs during the training phase, using ERMS, EMAE and R as measurement criteria. According to these figures superiority of the L-M algorithm over the CG algorithm for better performance (lower estimated error and increased efficiency in determining the number of optimal neurons) and higher convergence speed (when determining epoch size) is clearly established. Therefore the L-M was used for training of the model structures.

CONCLUSION

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 Mazandaran province in the north region of Iran was used for the case study. The used data in ANN was monthly hydrometric and climatic data with 31 years duration from 1969 to 2000. For the mentioned model 27 year's data were used for its development but for the validation/testing of the model 4 years data was applied. Six 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 rain fall data as input dimension with using tree stations, model 2 for rain fall and average temperature, model 3 for rain fall, average temperature and stream flow at the time step of t-1 and model 4 is enabled for rain fall and stream flow at time step of (t, t-1, t-2), model 5 for rain fall and stream flow at the time step of (t, t-1, t-2.t-3), model 6 for rain fall, average temperature and stream flow at the time step of (t-1, t-2). Computational efficiencies, i.e., better achieved accuracy and convergence speed, were evaluated for the conjugate gradient (CG) and Levenberg-Marquardt (L-M) training algorithms. Since the L-M algorithm was shown to be more efficient than the CG algorithm, therefore it was used to train the proposed six models. Based on the results validation stage of root mean square error (RMSE), Mean Squared Error (MSE) and coefficient of determination (r) measures were: 0.07, 6×10^{-4}, 0.99 (model 1); 0.1, 9×10^{-4}, 0.99 (Model 2); 0.01, 9×10^{-5}, 1 (Model3); 0.005, 6×10^{-5}, 1 (Model 4); 0.001, 0.7×10^{5}, 1 (Model 5); 0.001, 6×10^{-5}, 1 (Model 6). As indicated by the results, model 5 provided the highest performance. This was due to enabling of the precipitation and rainfall, resulting in improved training and thus improved prediction. This study has shown that when improved computational efficiency measures are combined with enabling of input parameters describing physical behavior of hydro-climatologic variables, improved model predictability becomes possible by the artificial neural networks.

ACKNOWLEDGMENT

This research project performed during 2007-2008 in Watershed Management Department with financial supported of the University of Mazandaran, Iran and its GIS and remote sensing technical facilities.

REFERENCES

Abebe, A.J., D.P. Solomatine and R.G.W. Venneker, 2000. Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events. Hydrol. Sci. J., 45 (3): 425-436.

Antcil, F. and N. Lauzon, 2004. Generalisation for neural networks through data sampling and training procedures, with applications to stream flow predictions. Hydrol. Earth Syst. Sci., 8 (5): 940-958.

ASCE Task Committee, 2000a. Artificial neural networks in hydrology. Int. J. Hydrol. Eng. ASCE., 5 (2): 115-123.

ASCE Task Committee, 2000b. Artificial neural networks in hydrology. Int. J. Hydrol. Eng. ASCE., 5 (2): 124-132.

Campolo, M., P. Andreussi and A. Soldati, 1999a. River flood forecasting with a neural network model. Water Resour. Res., 35 (4): 1191-1197.

Campolo, M., A. Soldati and P. Andreussi, 1999b. Forecasting river flow rate during low-flow periods using neural networks. Water Resour. Res., 35 (12): 3547-3552.

Chang, F.J., L.C. Chang and H.L. Huang, 2002. Real-time recurrent learning neural network for stream-flow forecasting. Hydrol. Process, 16 (13): 2577-2588.

Chang, L.C., F.J. Chang and Y.M. Chiang, 2004. A two-step-ahead recurrent neural network for stream-flow forecasting. Hydrol. Process, 18 (1): 81-92.

Chiang, Y.M., L.C. Chang and F.J. Chang, 2004. Comparison of static feed forward and dynamic-feedback neural networks for rainfall-runoff modeling. J. Hydrol., 290 (3-4): 297-311.

Chow, V.T., D.R. Maidment and L.W. Mays, 1988. Applied Hydrology. McGraw-Hill, New York, pp: 570.

Cigizoglu, H.K., 2005. Application of the generalized regression neural networks to intermittent flow forecasting and estimation. J. Hydrol. Eng., 10 (4): 336-341.

Cigizoglu, H.K. and O. Kisi, 2006. Methods to improve the neural network performance in suspended sediment estimation. J. Hydrol., 317 (3-4): 221-238.

Coulibaly, P., Y.B. Dibike and F. Anctil, 2005. Downscaling precipitation and temperature with temporal neural networks. J. Hydrol., 6 (4): 483-496.

Coulibaly, A.P. and N.D. Evora, 2007. Comparison of neural network methods for infilling missing daily weather records. J. Hydrol., 341 (1-2): 27-41.

Creutin, J.D., E. Andrieu and D. Faure, 1997. Use of weather radar for the hydrology of a mountainous area. Part II: Radar measurement validation. J. Hydrol., 193 (1-2): 26-44.

de Vos, N.J. and T.H.M. Rientjes, 2005. Constraints of artificial neural networks for rainfall-runoff modeling: Trade-offs in hydrological state representation and model evaluation. Hydrol. Earth Syst. Sci., 9 (2): 365-415.

Dibike, Y.B. and P. Coulibaly, 2006. Temporal neural networks for downscaling climate variability and extremes. Neural Networks, 19 (2): 135-144.

Gaume, E. and R. Gosset, 2003. Over-parameterisation, a majorobstacle to the use of artificial neural networks in hydrology. Hydrol. Earth Syst. Sci., 7 (5): 693-706.

Hagan, M.T., H.B. Demuth and M. Beale, 1996. Neural Network Design. 2nd Edn. PWS Publishing Company, Boston, USA., pp: 736.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. 2nd Edn. Prentice Hall. New Jersey, USA., pp: 696.

Hornik, K., M. Stinchcombe and H. White, 1989. Multilayer feedforward networks are universal approximators. Neural Networks, 2 (5): 359-366.

Hu, T.S., K.C. Lam and S.T. Ng, 2001. River flow time series prediction with a range-dependent neural network. Hydrol. Sci. J., 46 (5): 729-745.

Imrie, C.E., S. Durucan and A. Korre, 2000. River flow prediction using artificial neural networks: Generalization beyond the calibration range. J. Hydrol., 233 (1-4): 138-153.

Jain, A. and S. Srinivasulu, 2004. Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques. Water Resour. Res., 40 (4): 199-214.

Jain, S.K., A. Das and D.K. Srivastava, 1999. Application of ANN for reservoir in.ow prediction and operation. J. Water Res. Plan. Manage. ASCE., 125 (5): 263-271.

Khalil, M., U.S. Panu and W.C. Lennox, 2001. Groups and neural networks based streamflow data infilling procedures. J. Hydrol., 241 (3-4): 153-176.

Kuligowski, R.J. and A.P. Barros, 1998.Experiments in short-term precipitation forecasting using artificial neural networks. Monthly Weather Review, 126 (2): 470-482.

Kumar, D.N., K.S. Raju and T. Sathish, 2004. River flow forecastingusing recurrent neural networks. Water Res. Mange., 18 (2): 143-161.

Lin, G.F. and L.H. Chen, 2004. A non-linear rainfall-runoff model using radial basis function network. J. Hydrol., 289 (1-4): 1-8.

Luck, K.C., J.E. Ball and A. Sharma, 2000. A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. J. Hydrol., 227 (1-4): 56-65.

Luk, K.C., J.E. Ball and A. Sharma, 2001. An application of artificial neural networks for rainfall forecasting. Math. Comput. Modelling, 33 (6-7): 683-693.

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. Environ. Modelling and Soft., 15, 101-123 ASCE J. Hydrol. Eng., 10 (4): 336-341.

Nayak, P.C., K.P. Sudheer, D.M. Rangan and K.S. Ramasastri, 2005. Short-term flood forecasting with a neurofuzzy model. Water Resour. Res., 41 (8-9): 2517-2530.

Rajurkara, M.P., U.C. Kothyarib and U.C. Chaubec, 2004. Modeling of the daily rainfall-runoff relationship with artificial neural network. J. Hydrol., 285 (1-4): 96-113.

Rami'rez, M.C.P., H.F.C. Velho and N.J. Ferreira, 2005. Artificial neural network technique for rainfall forecasting applied to the Sao˜Paulo region. J. Hydrol., 301 (1-4): 146-162.

Salehi, F., S.O. Prashar, S. Amin, A. Madani, S.J. Jebelli, H.S. Ramaswamy and C.F. Drury, 2000. Prediction of annual nitrate-N losses in drain outflows with artificial neural networks. Trans. ASAE., 43 (5): 1137-1143.

Sudheer, K.P., A.K. Gosain and K.S. Ramasastri, 2002. A data-driven algorithm for constructing artificial neural network rainfall-runoff models. Hydrol. Process, 16 (6): 1325-1330.