Vol. 43, No. 3, pp 250-254, 2015

Indian Journal of Soil Conservation

Online URL:http://indianjournals.com/ijor.aspx?target=ijor:ijsc&type=home

Optimization of neural network structure for radial basis function network for simulation of hydrological processes

Ajai Singh

Centre for Water Engineering and Management, Central University of Jharkhand, Ranchi, Jharkhand- 835205.

E-mail: ajai_jpo@yahoo.com

ARTICLE INFO

Article history :
Received : December, 2013
Revised : April, 2015
Accepted : June, 2015

Key words :
Artificial neural network,
Learning rate,
Momentum factor,
Network structures,
Number of nodes

ABSTRACT

The results of optimization of neural network structures of Radial Basis Function Network (RBFN) model to predict sediment yield as a function of daily rainfall and stream flow during rainy season for a watershed area in India, has been presented in this paper. Various combinations of shuffled data set of different time lags were developed and compared for their ability to make a reliable sediment yield prediction. Measured data with 736 patterns of input-output vector was divided into three sets: 391 patterns for training, 215 patterns for testing and the remaining 130 patterns for validation. The root mean square error (RMSE) and Correlation coefficient (R) were determined to ascertain the model performance. The momentum rate, number of nodes at hidden layer, linear coefficient, learning rule and transfer functions were optimized based on lowest RMSE and highest R values. The RBFN model yielded better performance at 8 numbers of hidden layer node, momentum rate of 0.40, linear rate of 0.60, Delta learning rule and TanH transfer function. The RBFN model performed satisfactorily with R^2 and NSE values of 0.98 and 0.97 during calibration, and 0.91 and 0.91 during validation periods, respectively.

1. INTRODUCTION

Artificial intelligence is an area of computer science which deals with designing intelligent computer systems. Some of the methods for intelligent problem solving are expert systems, neural networks, fuzzy logic, cellular automata and probabilistic reasoning. Of these technologies, neural networks, fuzzy logic, and probabilistic reasoning are known as soft computing (Rajasekaran and Pai, 2010). Neural networks are simplified models of the biological nervous system and are highly interconnected network of a large number of processing elements called neurons in an architecture. Neural networks learn by examples. They can be trained with known examples of problems to acquire knowledge about it. Once appropriately trained, the network can be applied to solve unknown or untrained instances of the problem. Neural networks have been successfully applied in the field of pattern recognition, image processing, data compression, forecasting and optimization. In the past few decades, modeling approaches have been applied to study and understand the hydrological processes of a watershed. Based on the governing processes, the models are classified as either physics based or system theoretic. Physics based models are governed by the various physical processes controlling the hydrologic behavior of a system. However, system theoretic models do not require the physical characteristics of watershed. They use the transfer functions to map the data from input to output. Artificial neural network (ANN) models are example of system theoretic models that have gained considerable popularity in recent years in describing the hydrological processes. ANN models are developed upon the input and output data without going into the details of the complex physical laws governing the process under investigation and are usually able to provide reasonably accurate model. They are particularly suited for modeling non-linear systems where traditional parameter estimation techniques are not convenient and cost effective. Preliminary concepts and hydrologic applications of ANNs have been elaborated by ASCE (2000a, b). Neural network architecture is responsible for the overall structure and flow of information in ANN models. The neural network architectures have been broadly classified as single layer feed-forward networks, multilayer feed-forward networks, and recurrent networks. In feed-forward networks, the information propagation is only in one direction, *i.e.* from input layer to the output layer. Multilayer perceptrons are the most common form of feed-forward model architecture. Other feed forward network architectures are Generalized Regression Neural Networks, Radial Basis Function Networks (RBFN), Neuro-fuzzy networks and Support Vector Machines. In recent years, artificial neural networks have been used for modeling complex hydrological processes (Rajurkar *et al.*, 2002; Wilby *et al.*, 2003;

Giustolisi and Laucelli, 2005; Jain and Srinivasulu, 2006 and Singh *et al.,* 2008) and shown to be one of the most promising tools in hydrology (Maier *et al*., 2010).

Mutlu *et al.* (2008) compared the multilayer perceptron and the radial basis neural network models for simulating the stream flow at four gauging stations. They concluded that both models performed satisfactorily for flow predictions at multiple gauging stations, however, the multilayer perceptron model was found better. Agarwal *et al*. (2009) compared the runoff and the sediment yield from an Indian watershed using the back propagation artificial neural network modeling technique, and single and multi-input linear transfer function models. They reported that the single input linear transfer function runoff and sediment yield forecasting models were more efficacious than the multi input linear transfer function and ANN models. Wang *et al*. (2009) compared the performances of the back-propagation and generalized regression neural network ANN models, and found that ANN models were superior to the classical regression model in the weir sediment modeling. Yenigun *et al*. (2010) estimated sediment values by using artificial neural networks and found high correlation between the values obtained by using a feed-forward and backpropagation, and the observed values. De Farias *et al*. (2010) applied ANN model for estimating the sediment yield based on runoff and climatological data. They reported that the ANN model appeared to be superior to those generated by the multiple linear regression models, and the ANN model was suitable for identifying and extracting nonlinear trends for significant variables.

The optimal network structure generally strikes a balance between generalization ability and network size, and the number of free parameters. If network complexity is too low or an inappropriate functional form is selected, the network may be unable to capture the desired relationship. The network may decrease the generalization ability and processing speed for high network complexity. This could be more difficult to calibrate and might be less transparent. The methods for determining the optimal ANN structure can be classified into three types: global, stepwise, and ad-hoc. Despite the important role of the network structure in determination of the relationship between model inputs and outputs, little effort has been made into this area of the ANN model development process. In the present study, a trial-and-error or ad-hoc method to determine the parameters of network structures of RBFN model has been attempted.

2. MATERIALS AND METHODS

Study Area

The Nagwa watershed, located in Damodar Barakar Catchment, Jharkhand, India is approximately 92.46 km^2 in area, of which about 30-40% is under shrubs and forest and the remaining under cultivation. The average elevation of the command is 540 m above msl. The topography of the watershed is undulating with flat land in major parts. The study area comes under the Sub-humid tract of Eastern Plateau in the 12^{th} Agro-Ecological Zone of India. The texture of red and yellow soils of the watershed ranges from sandy loam to clay loam. Soil structures vary from moderately fine sub angular blocky to coarse sub angular blocky. The overall soils of the watershed are neutral to slightly acidic with medium organic matter and low salt content. Bulk density of the soils varies around 1.4 to 1.5 gcc^{-1} with moderately low saturated hydraulic conductivity ranging from 40 to 170 mm day^{-1}. It is bounded by latitudes of $23^0$59'08" N to $24^0$05'41" N and longitudes of $85^0$16'35" E to $85^0$23' 45" E.

Description of RBFN Model

Radial basis function approach was first developed by Powell (1987) as a technique for performing interpolation of a set of data points in a multi-dimensional space. In neural networks, radial basis functions were first exploited by Broomhead and Lowe (1988). As the artificial neuron is not intended to be a copy of biological neuron, many forms of non linear functions have been suggested. The RBFNs are general purpose networks which can be used for a variety of problems including system modeling, prediction and classification. Radial Basis Functions are used as transfer functions in many neural network simulators. These types of functions have been in use in approximation theory and in pattern recognition under different names for many years. A very good introduction to RBF and more general regularization networks was given by Poggio and Girosi (1990). Several types of localized radial basis functions exist and they all treat the activation value as radial coordinate. The structure of the RBF networks is similar to that of the feed forward neural network and consists of three layers, an input layer, a hidden layer, and an output layer (Fig. 1). The standard Euclidean distance is used to measure how far an input vector from the center is. The Euclidean distance is determined from the point which is being evaluated to the center of each neuron, and a radial basis function is applied to the distance to compute the weight for each neuron. The radial basis function is so named because the radius distance is the argument to the function. Different types of radial basis functions are used, but the most common is the Gaussian function. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron. The output of the RBFN is determined

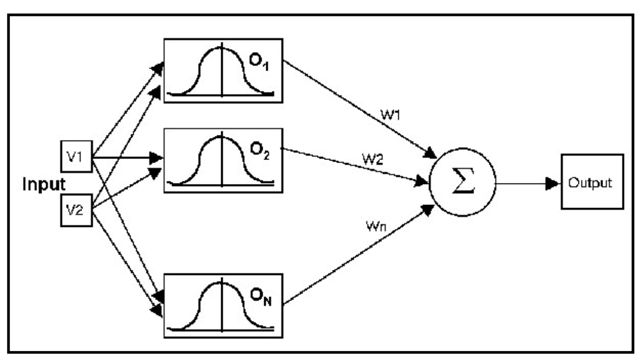

Fig. 1. Functioning of radial basis function network

as given hereunder (Mutlu *et al*., 2008).

$$Y = \sum W\theta(\| X\text{-}C \|) \quad ...(1)$$

Where, X is the input value, Y is the output value, $\theta(\ )$ is the radial basis function, W is the weights connecting the hidden and output nodes, C is the centre of hidden node, which depends on the observed input data and $\|X - C\|$ is the Euclidean distance between the input and hidden nodes. In the present study, the input and output data have been scaled to make it bounded in the intervals -1 and +1 and the standardization equation is given hereunder (Maier and Dandy, 2000).

$$y_t = 008 \times \left(\frac{x_t - x_{\min}}{x_{\max} - x_{\min}} \right) + 0.1 \quad ...(2)$$

Where, x_t is the original series of input data, y_t is the transformed series, x_{min} is the minimum value of the original series and x_{max} is the maximum value of the original series. After simulation, all the output values were de-standardized by multiplying with the respective standardization factor to get actual stream flow values.

Optimization of Neural Network

A proper selection of tuning parameters such as momentum factor, learning coefficient, sigmoidal gain and threshold values are required for efficient learning and designing of a stable network. Neural network structure defines the relationship between model input and output functions. Determination of an appropriate network structure involves the selection of a suitable number of hidden nodes, number of layers, number of nodes per layer, learning rule and type of transfer function. The methods for determining the optimal ANN structure can be classified into three types: global, stepwise and ad-hoc method. In the global method, the structure of an ANN model is arrived by using global method based on competitive evolution found in nature e.g. genetic algorithm, particle swarm optimization, simulated annealing, *etc.* In this method, it is possible to simultaneously optimize network parameters and structure. A stepwise trial and error procedure includes a basic ANN structure, which is modified with each trial with the objective of achieving a structure that is neither too complex nor too simple. The stepwise methods can further be divided into two categories, one based on pruning algorithms and the second based on constructive approaches. A pruning algorithm starts with a sufficiently complex ANN structure that is assumed to be capable of capturing the complexities involved in the physical system being modeled (Maire *et al*., 2010). In a constructive algorithm, the simplest ANN structure is taken first, which is successively made more complex by adding hidden neurons/layers one at a time and calibrating the resulting model. This process is repeated until there is significant improvement in model performance. Pruning and constructive algorithms can also be computationally intensive, as ANN models with many different structures generally need to be trained, and manually examined before arriving at the optimal structure. The approach to find out an appropriate network structure by using a trial-and-error method has been classified as ad-hoc.

The daily weather, stream flow and sediment data for the periods from 1995 to 2006 during monsoon months from June to October were collected from the Soil Conservation Department, Damodar Valley Corporation, Hazaribagh. The rainfall and stream flow of selected events were used as input and sediment yield as output of the RBFN model. A validation dataset was made before separating the data into training and testing data. Measured data with 736 patterns of input-output vector was divided into three sets: 391 patterns for training, 215 patterns for testing and the remaining 130 patterns for validation. The model performance during validation indicates the model output with new data set. The training of an RBFN model is usually a two stage process in which the basis functions are established at the hidden layer in the first stage, and the weights connecting hidden layer and output layer neurons are directly determined in the second stage. The numbers of nodes were changed in the hidden layer while keeping other parameters unchanged and correlation coefficient (R), and root mean square error (RMSE) values were noted. The momentum rate was kept constant with optimized number of nodes in the hidden layer. Similarly, learning rate was optimized with earlier values of momentum rate and number of nodes. A transfer function is a non-linear function that transforms the weighted sum of the effective inputs to a potential output value. When designing a network, the initial transfer function applies to each layer of the network. We applied the Linear, Sigmoid, Hyperbolic Tangent (TanH), Sine, Digital Neural Network Architecture (DNNA), and Exponential Function (Exp) transfer functions and the best transfer function were evaluated in terms of lowest RMSE and highest R values. Learning rules were optimized in similar fashion. Learning describes the way in which connecting weights are updated to generate a desired effect. The learning rule is the mathematical equation that determines the increment or decrement by which weights of a processing element change during the learning phase. The coefficients for learning rules are determined by the learning and recall schedule selected for each layer. A choice of learning rule depends on the network that we create. In the present study, the learning rules: Delta-Rule, Normalize Cumulative Delta (Norm-Cum-De), Extended Delta-Bar-Delta (Ext DBD), QuickProp, MaxProp, and Delta-Bar-Delta (DBD) were optimized for lowest RMSE and highest R values. RBFN model development and optimization were carried out by using NeuralWorks Pro II Plus.

3. RESULTS AND DISCUSSION

Number of Nodes

The RMSE and R values for input data set of daily rainfall and stream flow, and sediment yield as output data optimization of RBFN model were determined. The RBFN model structure was optimized for number of nodes at hidden layer, momentum rate, linear coefficient, learning rule and transfer function. There is no general criterion about deciding the number of hidden nodes. The guiding criterion is to select the minimum nodes which would not impair the network performance so that the memory demands for the storing the weights can be kept minimum. As far as generalization is concerned, the number of hidden nodes should be small compared to the number of training patterns. The number of hidden layer neurons significantly influences the performance of the network. If this number is small, the network may achieve a desired level of accuracy, while with too many nodes, it will take long time to get trained and sometimes over fit the data. The numbers of nodes were varied from 4 to 30. Increasing the number of nodes after 8 does not influence the results. Therefore, 8 neurons or number of nodes at hidden layer were accepted. There was a trend of decreasing R and increasing RMSE values with increase in number of nodes.

Momentum Factor

Weight adjustment is made based on momentum method. The momentum factor has a significant role in deciding the values of learning rate that will produce rapid learning. It determines type step size of the change in weight or biases. If momentum factor is zero, the smoothening is minimum and the entire weight adjustment comes from the newly calculated change. If the momentum factor is one, new adjustment is ignored and previous one is repeated. Between 0 and 1 is a region where the weight adjustment is smoothened by an amount proportional to the momentum factor. Momentum factor was varied from 0.1 to 0.8 and momentum factor of 0.4 gave the lowest RMSE and highest R values (Fig. 2). The role of momentum factor is to increase the speed of learning without leading to oscillations. The momentum terms effectively filters out high frequency variations of the error surface in the weight space, since it adds the effect of past weight changes on the current direction of movement in the weight space.

Learning Rate

The range of learning rate depends upon the number and type of input patterns. Learning rate coefficient determines the size of the weight adjustments made at each iteration and hence influences the rate of convergence. Poor choice of coefficient can result in a failure in convergence. If the learning rate coefficient is too large, the search path will oscillate and converges more slowly than a direct descent. The largest value of learning coefficient is obtained if each pattern considered is a separate type. The optimum value lies between the lowest and largest extremes. This coefficient must be smaller for many input patterns because the step length is controlled by the learning coefficient. If the learning is small, the weights are changed in small increments, thus, causing the systems to converge more slowly but with little oscillation. The learning rate has to be chosen as high as possible to allow fast learning without leading to oscillation.

The epoch level was set based upon the number of training data and transition point between clustering and output layer. No noticeable change on the performance of RBFN model was observed with the change in epoch level. Similarly linear coefficient of 0.60 yielded better performance (Fig. 3). Delta learning rule (Fig. 4) and TanH transfer function performed better than other learning rule and transfer functions.

Training and Validation of RBFN Model

Initially, only average discharge was used as the input. Later daily rainfall was added and improved results were obtained. It is observed that adding previous day average discharge did not have a significant effect on the output. Hence daily rainfall and average discharge as its input was taken as the optimum neural network model for the present study for RBFN model. The observed values of sediment yield and RBFN model simulated values were plotted on

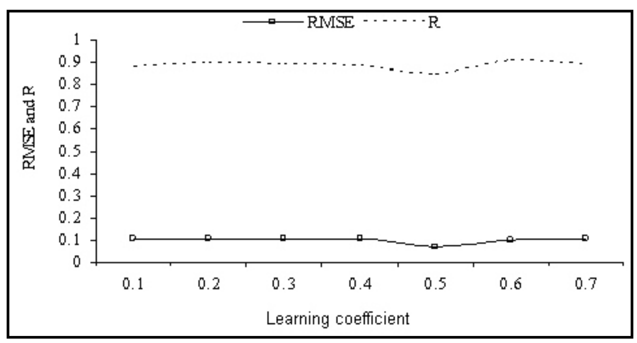

Fig. 3. Variation of RMSE and R with change in learning rate

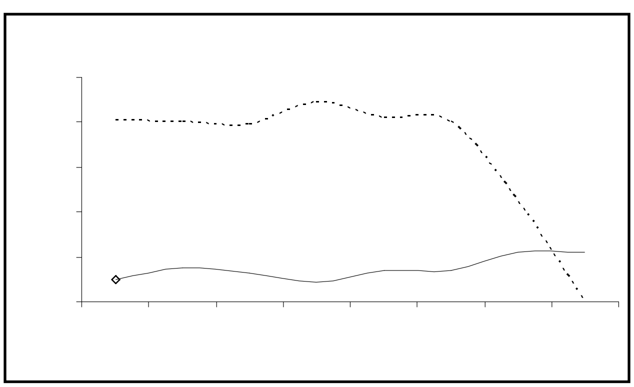

Fig. 2. Change of RMSE and R values with momentum rate under RBFN model

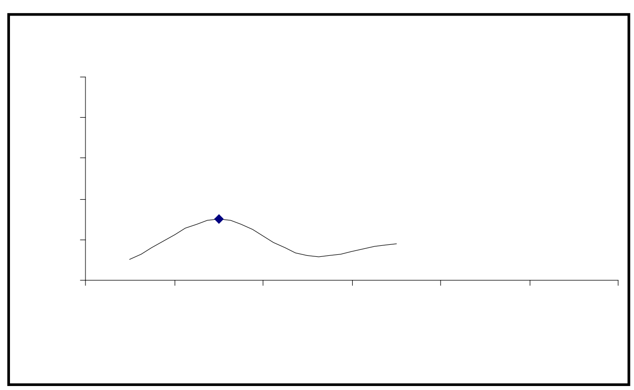

Fig. 4. Change of RMSE and R values with learning rules under RBFN model

scatter plot for the validation periods (Fig. 5). Scatter plots between observed and predicted sediment yield data serve a strong basis to assess a model's accuracy. The closer the scatter points are to the line of the best fit, the better the model. Lower values of sediment yield were well simulated by the model. The inability of the model in estimating large values of sediment load may be attributed to non linearity of governing processes of sediment detachment and final sediment yield generated from a watershed area. The training data should have a proper representation of lower, medium and larger values of sediment yield. The RBFN model performed satisfactorily with Coefficient of determination (R^2) and Nash-Sutcliffe efficiency (NSE) values of 0.98 and 0.97 during calibration, and 0.91 and 0.91 during validation periods, respectively. RBFN networks do not have any associated connections between input hidden nodes, which give a weighted input to each hidden node before the nonlinear transformation takes place. Thus, in an RBFN, for any point in the input space the response of the closest basis function plays a major role in the output of the network. The RBFN model simulated mean annual sediment yield as 3.6 and 3.5 t ha^{-1} during training and validation periods was very close to the mean annual observed sediment yield of 3.7 t ha^{-1}.

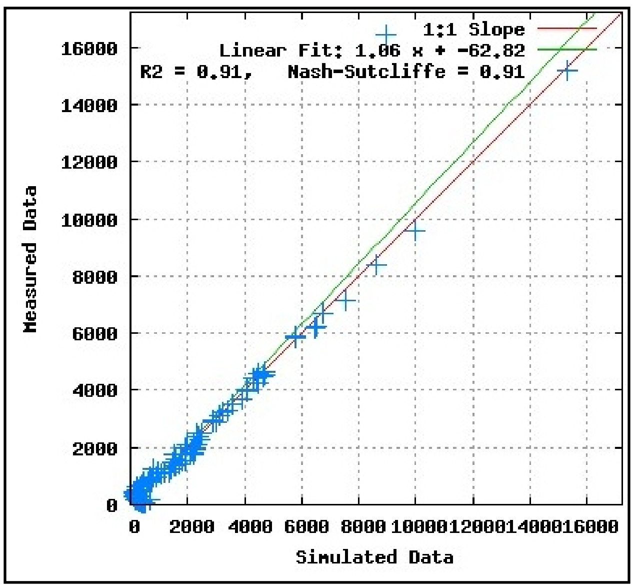

Fig. 5. Scatter plots of daily measured and RBFN simulated sediment yield (tonnes) for selected events during validation periods

4. CONCLUSIONS

The non-linear relationship of hydrological processes is appropriate for the application of RBFN model. The optimization of model structure parameter for RBFN model for simulating sediment yield was carried out in this study. The RBFN model was optimized with number of nodes at hidden layer as 8, momentum rate of 0.40, linear coefficient of 0.60, Delta learning rule and TanH transfer function. The performance of RBFN model, for predicting daily sediment yield at single point outlet in the watershed on the basis of statistical analysis in terms of the R^2 and NSE during training and validation, was found good. The findings of this study will help the neural network community to train the model with this standardized parameter to achieve the best performance.

REFERENCES

Agarwal, A., Rai, R.K. and Upadhyay, A. 2009. Forecasting of runoff and sediment yield using artificial neural networks. *J. Water Resour. Prot.*, 1: 368-375.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. 2000a. Artificial neural networks in hydrology. I: preliminary concepts, *J. Hydrol. Eng.,* ASCE, 5(2): 115-123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. .2000b. Artificial neural networks in hydrology. II: hydrologic applications, *J. Hydrol. Eng.,* ASCE, 5(2):124-137.

Broomhead, D.S. and Lowe, D. 1988. Multivariable functional interpolation and adaptive networks. *Complex System,* 2: 321-355.

De Farias, Camilo A.S., Alves, Francismário M., Santos, Celso A.G. and Koichi, S. 2010. An ANN-based approach to modelling sediment yield: a case study in a semi-arid area of Brazil. *Sediment Dynamics for a Changing Future* (Proceedings of the ICCE symposium held at Warsaw University of Life Sciences - SGGW, Poland, 1418 June, 337:316-321.

Giustolisi, O. and Laucelli, D. 2005. Improving generalization of artificial neural networks in rainfall-runoff modeling. *Hydrol. Sci. J.,* 50(3): 439-457.

Jain, A. and Srinivasulu, S. 2006. Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques. *J. Hydrol.,* 317: 291-306.

Maier, H.R. and Dandy, G.C., 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environ. Modelling Software,* 15: 101-124.

Maier, H.R., Jain, A., Dandy, G.C. and Sudheer, K.P. 2010. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environ. Modelling Software,* 25: 891-909.

Mutlu, E., Chaubey, I., Hexmoor, H. and Bajwa, S.G. 2008. Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. *Hydrol. Process.,* 22(26): 50975106.

Poggio, T. and Girosi, F. 1990. Network for approximation and learning. *Proc. IEEE*, 78(9): 14811497.

Powell, M.J.D. 1987. Radial basis functions for multivariable interpolation: a review. *Proceedings of IMA Conference on Algorithms for Approximation*, Mason JC, Cox MG (eds). Oxford University Press, New York, 143-167.

Rajasekaran, S. and Pai, G.A.V. 2010. Neural networks, Fuzzy logic, and Genetic algorithms. Synthesis and Application. PHI Learning Private Limited, New Delhi.

Rajurkar, M.P., Kothyari, U.C. and Chaube, U.C. 2002. Artificial neural networks for daily rainfall-runoff modelling. *Hydrol. Sci. J.,* 47(6): 865-877.

Singh, R., Sharma, H.C. and Kumar, Ambrish. 2008. Artificial neural network modelling of reference evaporation for Garhwal region of Uttarakhand. *Ind. J. Soil Cons.,* 36(3): 131-135.

Wang, Y.M., Kerh, T. and Traore, S. 2009. Neural Networks Approaches for modelling river suspended sediment concentration due to tropical storms. *Global Nest Journal*, 11(4): 457-466.

Wilby, R.L., Abrahart, R.J. and Dawson, C.W. 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrol. Sci. J.,* 48(2): 163-181.

Yenigun, K., Mahmut, B., Reşit, G. and Mehmet, M. 2010. A comparative study on prediction of sediment yield in the Euphrates Basin. *Internl. J. Phys. Sci.,* 5(5):518-534.