Scientific Journal of Impact Factor (SJIF): 4.72
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 11, November -2017

OPTIMISATION OF NEURAL NETWORKS FOR RAINFALL-RUNOFF MODELING

Lateef Ahmad Dar

Deptt.of Civil Engineering, National Institute of Technology Srinagar, J&K, India.

Abstract: *The relationship between rainfall and runoff is one of the most complex hydrologic phenomena to comprehend due to the tremendous spatial and temporal variability of watershed characteristics and precipitation patterns, and the number of variables involved in the modeling of the physical processes. As a result of these difficulties, and of a poor understanding of the real-world processes, empiricism can play an important role in modeling of R-R relationships. Artificial Neural Networks (ANNs) are typical examples of empirical models. Their ability to extract relations between inputs and outputs of a process, without the physics being explicitly provided to them, theoretically suits the problem of relating rainfall to runoff well, since it is a highly nonlinear and complex problem. The goal of this investigation was to develop rainfall-runoff models for the river Jhelum catchment that are capable of accurately modelling the relationships between rainfall and runoff in a catchment. Two types of ANN models viz. Back Propagation networks (BPN) and Radial Basis function (RBF) were developed. The network architecture in the back propagation network was changed by changing the number of neurons in the hidden layer. The analysis of performance of the various models was carried out by statistical analysis technique .The comparison was based on various statistical parameters like root mean square error (RMSE) and* R^2.

Keywords: *Rainfall, Neural Network, BPN, RBF*

1. Introduction

A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use [1].The fundamental processing element of an ANN is an artificial neuron. Just like the natural neuron in human brain, it can receive inputs, process them and produce the relevant output. To develop an ANN model, the primary objective is to optimize the architecture of the ANN that captures the relationship between the input and output variables. The process of determining the number of neurons in the input and output layers is based on the input and output variables considered to model the physical process. The number of neurons in the hidden layer can be optimized using the available data through the use of a trial and error procedure. Other methods such as the one proposed by Sudheer et al.[2003] can also be used to determine the optimum number of neurons in the hidden layer. [7] The regular ANN models (i.e., those not using wavelet decomposed input data) consisted of an input layer, one single hidden layer, and one output layer consisting of one node denoting the targeted daily water demand. Sigmoid and linear activation functions were used for the hidden and output nodes, respectively. The ANN models were trained and tested based on different combinations of input variables and the number of neurons in the model’s hidden layer. The input nodes consisted of various combinations of the following variables: the maximum temperature, the total precipitation, and the urban water demand. For each variable, data from the current day, from the previous day, from two days before, from three days before, and from four days before was explored. Each ANN model was tested on a trial and error basis for the optimum number of neurons in the hidden layer (found to be between 2 and 5). [8] All of the ANN models were first trained using the data in the training set (2001 to 2009) to obtain the optimized set of connection strengths and then tested using the testing data set (2010 to 2013) and compared using the statistical measures of goodness of fit.

2. Study Area

The Jhelum and its associated streams that drain the bordering mountain slopes together constitute the drainage network of the study area. The river network in the catchment is shown in figure 2. They include the fairly developed systems of the Sind, Rambiara, Vishaw and Lidder rivers as well as tiny rivulets such as the Sandran, Bringi and Arapat Kol . Adjusted to the varying nature of geomorphic and geological setting, the fluvial systems in the study area have peculiar characteristics of their own. Drainage system of the Upper Jhelum catchment has an evolutionary history marked by stupendous changes in level, rejuvenating at one time, and at others becoming sluggish, or even choking their channels with their own debris with consequent diversions and the ever-threatening process of mutual piracy.

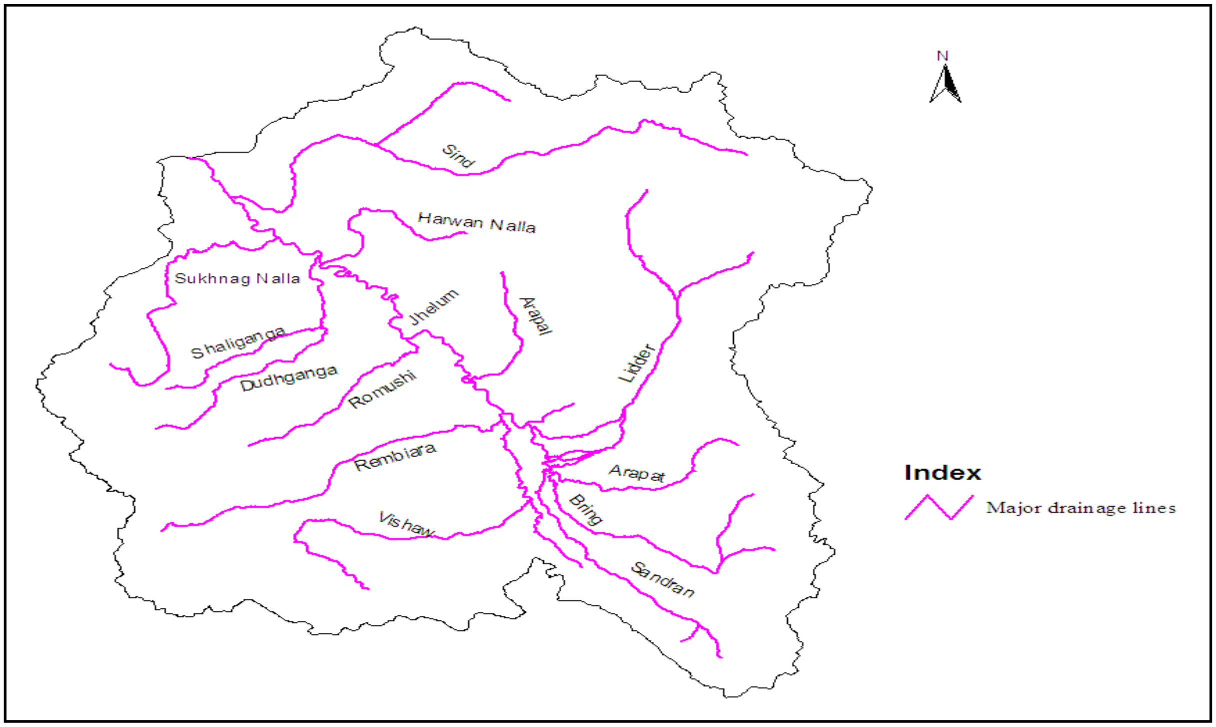

Fig.2 River Network in the Catchment.

3. Data:

The discharge data at padshahibagh from 2001-2013 was procured from the Irrigation and Flood Control Department, Srinagar. The precipitation data at Srinagar, Pahalgam and Qazigund stations for years 2001-2013 was procured from Indian meteorological department (IMD) and national climate data centre (NCDC,US).The collected data was then be digitized.

4. Artificial neural networks

An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain threshold), the neuron is activated and emits a signal though the axon. This signal might be sent to another synapse, and might activate other neurons. The complexity of real neurons is highly abstracted when modelling artificial neurons. These basically consist of inputs (like synapses), which are multiplied by weights (strength of the respective signals), and then computed by a mathematical function which determines the activation of the neuron. Another function (which may be the identity) computes the output of the artificial neuron (sometimes in dependance of a certain threshold). ANNs combine artificial neurons in order to process information. The higher a weight of an artificial neuron is, the stronger the input which is multiplied by it will be. Weights can also be negative, so we can say that the signal is inhibited by the negative weight. Depending on the weights, the computation of the neuron will be different. By adjusting the weights of an artificial neuron we can obtain the output we want for specific inputs. But when we have an ANN of hundreds or thousands of neurons, it would be quite complicated to find by hand all the necessary weights. But we can find algorithms which can adjust the weights of the ANN in order to obtain the desired output from the network. This process of adjusting the weights is called learning or training. The number of types of ANNs and their uses is very high. Since the first neural model by McCulloch and Pitts (1943) there have been developed hundreds of different models considered as ANNs. The differences in them might be the functions, the accepted values, the topology, the learning algorithms, etc. Also there are many hybrid models where each neuron has more properties than the ones we are reviewing here. Because of matters of space, we will present only an ANN which learns using the backpropagation algorithm (Rumelhart and McClelland, 1986) for learning the appropriate weights, since it is one of the most common models used in ANNs, and many others are based on it. Since the function of ANNs is to process information, they are used mainly in fields related with it. There are a wide variety of ANNs that are used to model real neural networks, and study behavior and control in animals and machines, but also there are ANNs which are used for engineering purposes, such as pattern recognition, forecasting, and data compression.

4.1 The Backpropagation Network(BPN):

The backpropagation algorithm (Rumelhart and McClelland, 1986) is used in layered feed forward ANNs. This means that the artificial neurons are organized in layers, and send their signals "forward", and then the errors are propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. There may be one or more intermediate hidden layers. The backpropagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea of the backpropagation algorithm is to reduce this error, until the ANN learns the training data. The training begins with random weights, and the goal is to adjust them so that the error will be minimal.

4.2 Radial Basis Function (RBF) Network

The Radial Basis Function (RBF) network is a variant of the standard feed-forward network. It can be considered as a two-layer feed-forward network in which the hidden layer performs a fixed non-linear transformation with no adjustable internal parameters. The output layer, which contains the only adjustable weights in the network, then linearly combines the outputs of the hidden neurons [after Chen et al., 1991]. The RBF network is trained by determining the connection weights between the hidden and output layer through a performance training algorithm. The hidden layer consists of a number of neurons and internal parameter vectors called 'centres', which can be considered the weight vectors of the hidden neurons. A neuron (and thus a centre) is added to the network for each training sample presented to the network.
The input for each neuron in this layer is equal to the Euclidean distance between an input vector and its weight vector (centre), multiplied by the neuron bias. The transfer function of the radial basis neurons typically has a Gaussian shape. This means that if the vector distance between input and centre decreases, the neuron's output increases (with a maximum of 1). In contrast, radial basis neurons with weight vectors that are quite different from the input vector have outputs near zero. These small outputs only have a negligible effect on the linear output neurons.

5. Results and Discussions

The networks developed were used for simulation modeling separately. The hidden layers in the network was fixed to one. Hence we start with one neuron in the hidden layer and increase it every time after training the network. The observed Vs predicted runoffs for various number of neurons in the hidden layer is shown in figure 2. Radial Basis Function network was also trained and validated for the same data set as used in BPN network and the results are shown in figure 3 and table 2.

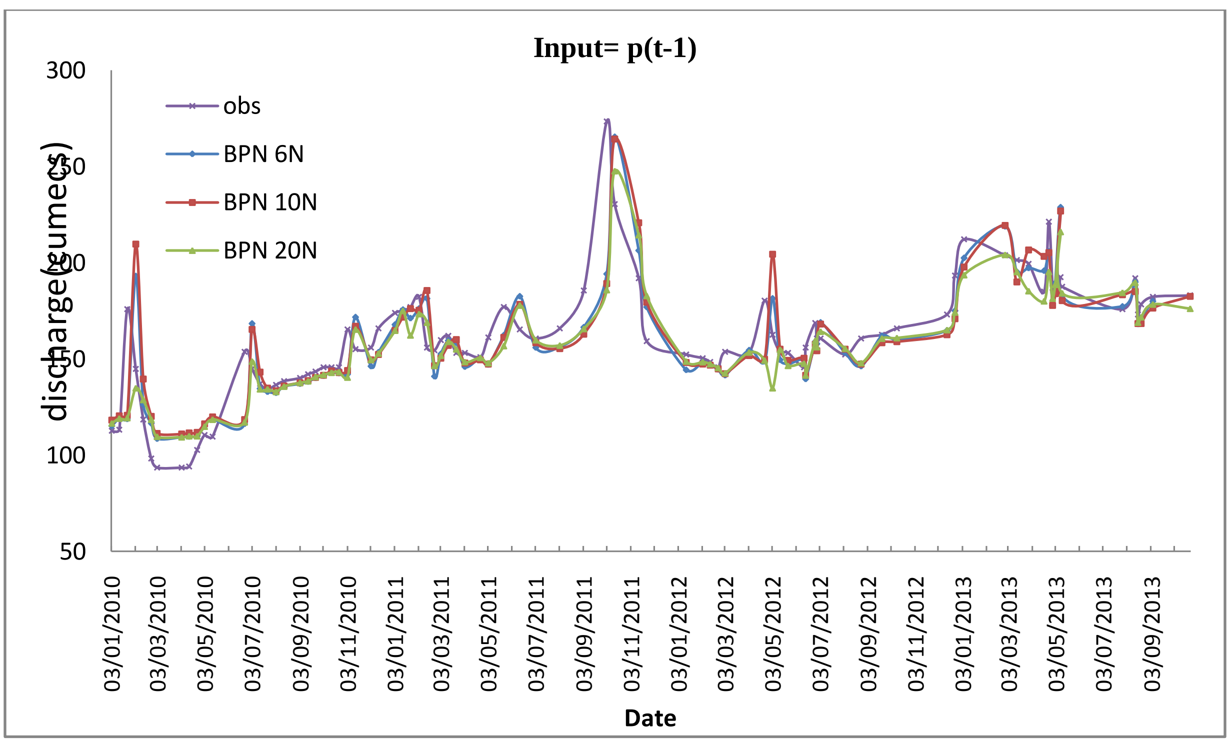

Fig. 2: Observed vs. predicted discharge, P(t-1) for BPN network.

Table 1: Goodness of fit for the effect of no. of neurons in the hidden layer of BPN network

NO. OF NEURONS	RMSE	R-SQUARE
6	0.736	0.77
10	0.612	0.81
20	0.775	0.758

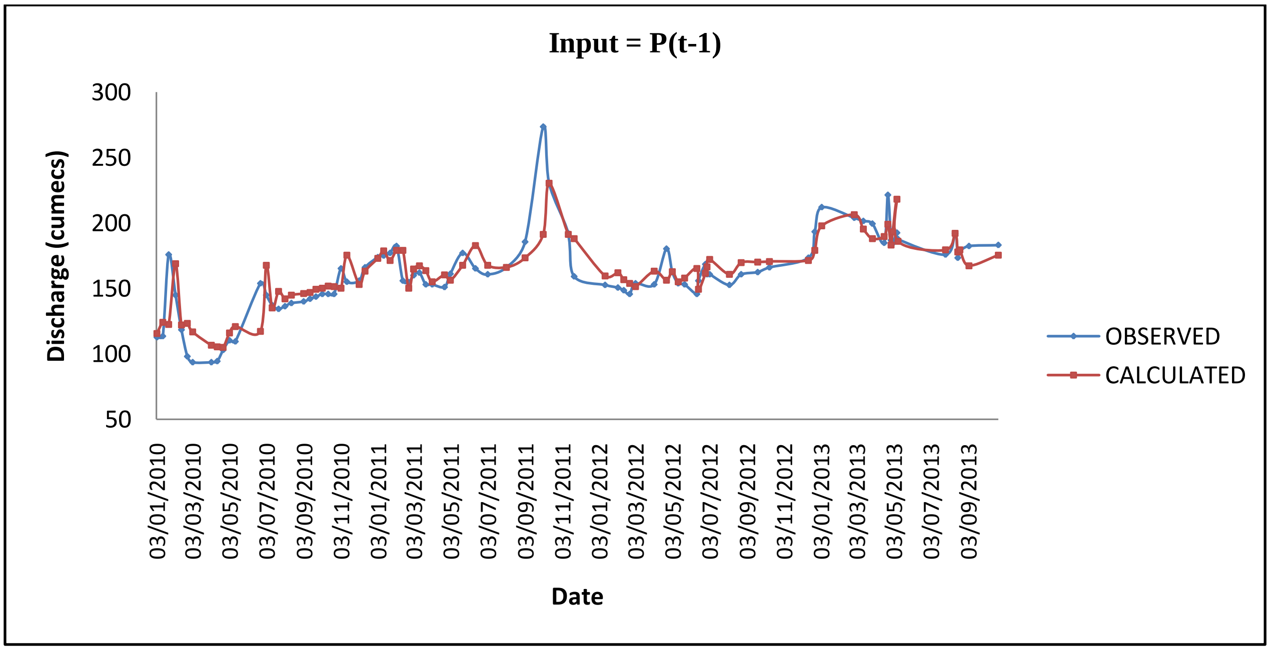

Fig. 3: Observed vs. predicted discharge, P(t-1) .

Table 2: Statistical indices of RBF model for various inputs.

NETWORK	INPUTS	RMSE	R-SQUARE
RBF	P(t-1)	0.097	0.912
	P(t-1)…P(t-5)	0.046	0.937

6. Conclusion

The developed ANN model was found performing to a good degree of accuracy compared to other models. Among the various ANN models developed Radial Based Function networks (RBF) with RMSE=0.046 and R^2=0.937 showed better overall performance than the Back Propagation Networks (BPN) Viz. BPN 6N (RMSE=0.245 and R^2=0.856) and ,10N(RMSE=0.132 and R^2=0.891) and MLR(RMSE=0.234 and R^2=0.827). The various models were then used for modelling individual hydrographs and it was seen that artificial neural networks (ANN's) proved to be superior. The Radial basis function (RBF) model with RMSE=0.016 and R^2=0.976 again proved to be better than the Back Propagation Networks (BPN) Viz. BPN 6N (RMSE=0.107 and R^2=0.905)and ,10N(RMSE=0.053 and R^2=0.933) and MLR(RMSE=0.212 and R^2=0.857).

References

[1] Adamowski, J. (2007), Development of a short-term river flood forecasting method based on wavelet analysis, J. Hydrol., 353(3–4), 247–266, doi:10.1016/j.jhydrol.2008.02.013.

[2] Banerjee, P., R. K. Prasad, and V. S. Singh (2009), Forecasting of groundwater level in hard rock region using artificial neural network, Environ. Geol., 58(6), 1239–1246, doi:10.1007/s00254-008-1619-z.

[3] Hagan, M. T., and M. Menhaj (1994), Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Networks, 5(6), 989–993, doi:10.1109/72.329697.

[4] Kisi, O. (2004), River flow modeling using artificial neural networks, J. Hydrol. Eng., 9(1), 60–63, doi:10.1061/(ASCE)1084-0699(2004)9:1(60).

[5]Govindaraju, Rao S. **2000** “Artificial neural networks in hydrology I: preliminary concepts” Journal of Hydrologic Engineering, Vol. **5**, No. **2**, pp. **115-123**, ASCE.

[6] Maas, T. (2003), What the experts think: Understanding urban water demand management in Canada. Retrieved from http://www.polisproject.org/publications/byauthor/Maas,%20Tony.

[7] Maidment, D., and S. Miaou (1986), Daily water use in nine cities, Water Resour. Res., 22(6), 845–851, doi:10.1029/WR022i006p00845.

[8] Maidment, D., and E. Parzen (1984), Monthly water use and its relationship to climatic variables in Texas, Water Resour. Bull., 19(8), 409–418.

[9]Meher-Homji, V. M**. (1971):** The Climate of Srinagar and its Variability. Geog. Rev. Ind., **33, (1): 1-14.**

[10] Mehnaza Akhter, Ahanger Manzoor Ahmad (2017) Climate Modeling of Jhelum River Basin - A Comparative Study. Environ Pollut Climate Change 1: 110.

[11] Mehnaza Akhter (2017) Application of ANN for the Hydrological Modeling. International Journal for Research in Applied Science & Engineering Technology (IJRASET**),** Volume 5 Issue VII, July 2017

[12] Partal, T. (2009), River flow forecasting using different artificial neural network algorithms and wavelet transform, Can. J. Civil Eng., 36(1), 26–38, doi:10.1139/L08-090.

[13] Sethi, R. R., A. Kumar, S. P. Sharma, and H. C. Verma (2010), Prediction of water table depth in a hard rock basin by using artificial neural network, Int. J. Water Resour. Environ. Eng., 2(4), 95–102.

