Journal of Civil Engineering (IEB), 40 (1) (2012) 47-59

Journal of Civil Engineering IEB

Black box model for flood forecasting

Dilip Kumar , Saroj Kumar Bhishm and Singh Khati

Govind Ballabh Pant Engineering College, Pauri (Garhwal), Uttarakhand 246001, India

Received 8 June 2011

Abstract

Due to heavy rainfall in the Himalayan and other watersheds of the North India, Haryana is endowed with extensive river system consisting of the Satluj, and its tributary. All the rivers in Haryana are flood prone, mainly because they receive heavy rainfall within a short duration. These rivers are in their earlier stage of maturity and are active agents of erosion. This region of India, which has about one third of the country's total water resources potential, is however not enjoying the enormous water resources potential; instead suffering a lot due to the regular flood hazard. Need of improving flow forecasting capability for better management of Water Resources has also been emphasized by Sarma et.al (2007). Therefore, an attempt has been made to develop a flood forecasting model using ANN in this study. Flood forecasting undoubtedly is the most challenging and important task of operational hydrology. Conventional methods for establishing the relationship between rainfall and runoff need to understand the behavior of hydrological cycles. The temporal and spatial variability that characterizes a river system makes flow forecasting a demanding task. Flow forecasting is a crucial part of flow regulation and water resource management. It is well known that floods kill more people and cause more damage than any other natural disaster. Consequently there is a need for systems capable of efficiently forecasting discharge rates in rivers. Artificial Neural Networks provide a fast and flexible means for developing non-linear flow routine models. Flood Forecasting (FF) forms an important tool in reducing vulnerabilities and flood risk. Therefore, the attempt of this paper is to explore the possibility of forecasting stream flow using scanty rainfall data using Artificial Neural Network technique in Mat lab.

Keywords: ANN, Flood, rainfall, forecasting

1. Introduction

Artificial Neural Network (ANN) is a mathematical structure that tries to operate in the same manner as the human brain**.** Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from

the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes. Figure 1. Shows various components of a neuron and in McCulloh and Pitts model (MCP, 1943).

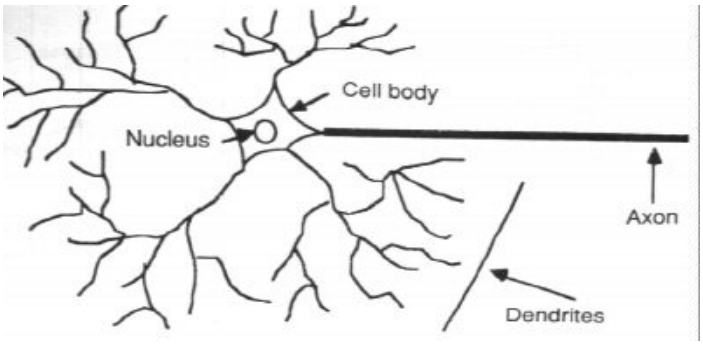

Fig. 1. Various Components of a Neuron

The inputs were 'weighted'; the effect that each input has at decision making is dependent on the weight of a particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case neuron does not fire. Fig shows a typical MCP neuron with different weights assigned to different input parameters. In mathematical terms, the neuron fires if and only if;

$$X1W1 + X2W2 + X3W3 + ... > T \quad (1)$$

Information Processing

The transfer function and learning algorithm are two important component of information processing.

Transfer Function

It is the mechanism of translating input signals to output signals for each processing element. Three main types of transfer functions are:

1. Hard Limit transfer function

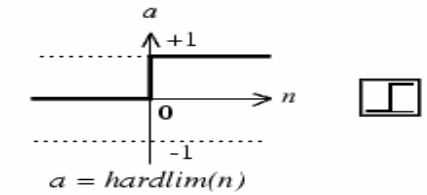

Fig. 2. Hard Limit Transfer Function

The hard-limit transfer function shown above limits the output of the neuron to either 0, if the net input argument n is less than 0; or 1, if n is greater than or equal to 0. The mathematical notation of hard limit function is shown in the figure.

2. Linear transfer function

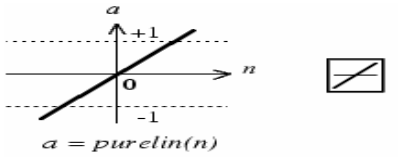

Fig. 3. Linear Transfer function

3. Log-sigmoidal transfer function (sigmoid function) commonly known as logistic function

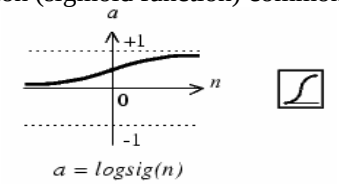

Fig. 4. Sigmoid Non-Linear Transfer Function

This transfer function is commonly used in back propagation networks, in part because it is differentiable. The symbol in the square to the right of each transfer function graph shown above represents the associated transfer function.

2. Learning Methods

A learning process, or training forms the inter connections (correlations) between neurons. It is accomplished by providing known input and output data in an orderly manner to the Neural Network. The learning corresponds to the parameter changes. A network can be subjected to supervised or unsupervised learning.

1. Supervised learning: External prototypes are used as target outputs for specific inputs, and the network is given a learning algorithm to follow and calculate new connection weights that bring output closer to the target output.
2. Unsupervised learning: It is a sort of learning takes place without a teacher. In learning without supervision, the desired responses is not known, thus explicit error information cannot be used to improve behavior.

Most of the neural Networks are trained using a supervised learning algorithm. There are many supervised learning algorithms, but one of the most widely used is back propagation algorithm; because of its mathematical simplicity.

3. Back Propagation Algorithm

In the back propagation algorithm there are two phases in the learning cycle-one, to propagate the input pattern through the network and the second to adopt the output by changing the weights in the network. It is the error signals that are back propagated in the network operation to the hidden layers. The error in the output layer is used as a basis for adjustment of connection weights between the input and hidden layers and subsequent recalculation of the output values becomes an iterative process which is carried out until the error falls below a tolerance level. The back propagation algorithm is summarized below. Implementation details can be found in most neural networks books (eg.Bishop, 1995).

1. Initialize network weights.
2. Present first input vector, from training data to the network.
3. Propagate the input vector through the network to obtain an output.

4. Calculate error signal by comparing actual output to the desired output (target).
5. Propagate error signal back through the network.
6. Adjust weights to minimize overall error.
7. Above steps is repeated with the new input vector, until overall error is satisfactorily small.

3.1 Faster training techniques

Various Back propagation Techniques are employed to train the model. All the techniques are discussed below. The aim is to create a network which gives an optimum result.

3.2 Variable learning technique

With standard steepest descent, the learning rate is held constant throughout training. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. It is not practical to determine the optimal setting for the learning rate before training. The type of ANN is feed forward network. The transfer functions used are “tansig” and “purelin”.The training technique is “traingda”. The various training parameters used in this technique are show, epoch, goal and learning rate. We can alter these training parameters.

4. Study Area

The present study has been carried out for the catchment of a tributary channel of the Satluj River, upstream of Bhakra reservoir (Fig.5). It is one of the main tributaries of the Indus River and flows through different areas having different climatic and topographic features. The channel receives runoff from snow and glaciers as well as rain. The lower part of the basin experiences a considerable amount of rainfall. This basin is elongated in shape and covers an area of about 450 km^2 up to Bhakra Dam. The elevation of the study basin varies from about 500 to 7000 m. The study area enjoys semi-arid conditions. The main occupation of the people in this area is agriculture. They depend on groundwater, because surface water resources are scarce. Due to erratic rainfall pattern and uncontrolled abstraction, groundwater levels have climate.

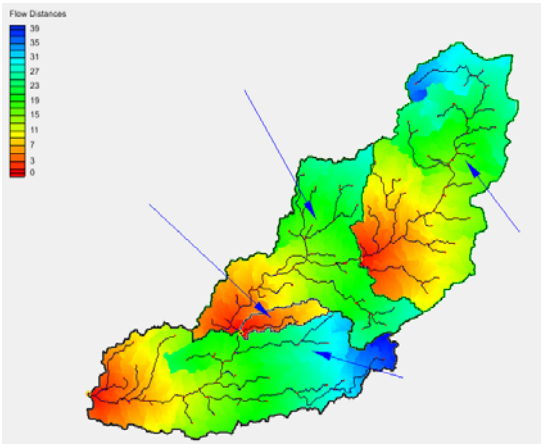

Fig. 5. Showing satluj watershed with flow distance

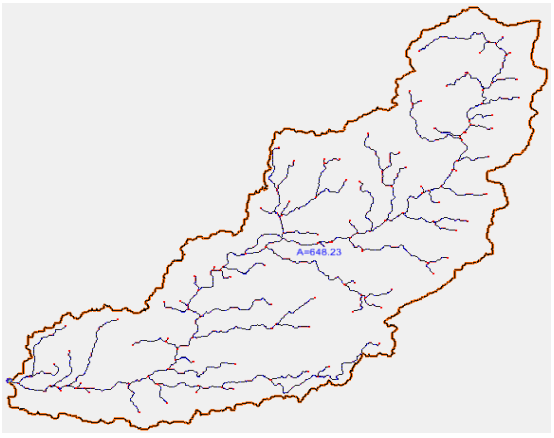

Fig. 6. Showing satluj watershed

The average annual rainfall for the whole study basin varied from 313 to 423 mm, most of which falls during southwest monsoon from June to September. The temperature reaches its high in April and May, and low in January, with the mean maximum of 42°C in May, a mean minimum of 11°C in January.

5. Methodology

5.1 Model development

The forecasting problem includes the development of model for predicting the runoff values based on rainfall values of different gauge stations as input. Rainfall data has been collected from two rainguage stations located at Satluj River Basin. The problem was handled by neural network approach. A four month data set of two rain guage stations was given as input and verified whether it is matching with the output. Normalized value of input and output values has been used to develop the ANN model. Following algorithms has been used in Mat lab.
Step 1.Normalization of the rainfall value of both the gauge stations, discharge value and water level between 0 to 1.

Step 2. Creation of a new feed forward network by providing minimum, maximum function, no of hidden neurons in the hidden layer, transfer functions and training algorithm to be used.
Step 3. Altering the training parameters goal, epoch and show by initially assuming arbitrary values.
Step 4. Simulating the model by using the command sim.
Step 5. Converting the normalized value into original value after simulation.
Step 6. Plotting the graph between actual and simulated value to observe the performance of different ANN models.

In the initial stage of this study neither daily discharge nor daily water level data were available. However, daily rainfall data of two rain gauge stations and monthly stream flow data were available. Therefore to initiate experimentation on forecasting capability of ANN, an attempt was made to generate daily discharge data from the available monthly stream flow data and rainfall data. An assumption that surface runoff varies proportionally was made to derive the daily discharge. Several exercise thus carried out has helped us to get insight into the ANN model development process.

Therefore the ANN Model development is presented here in two phases; first phase with daily discharge as output and in the second phase with daily water level as output. The first phase of model development can be considered purely as an academic exercise while the second phase of model development can be considered as an attempt to develop a model applicable in the real field.

5.2 Forecasting discharge with rainfall as input

120 data set of rainfall value from both the rain gauge stations has been taken and normalized between 0 to 1. Discharge data used are also normalized. By altering the training parameters like goal, epoch, shows etc and increasing the no of neurons in the hidden layer we have verified the performance of the model. In this trial daily stream flows values has been formulated from mean monthly flow values as target discharge values .Various algorithms has been tried with varying hidden neurons in the hidden layer.

5.3 One day lead forecast with rainfall and minimum discharge

In this trial daily stream flows values has been formulated from mean monthly flow values as target discharge values .Various algorithms has been tried with varying hidden neurons in the hidden layer. In the second attempt Forecasting discharge with Rainfall data of two gauge stations located at Nayabasti and Beruma College Road and Minimum Discharge of Pagladia River Basin has been done. The Results were presented as follows.

Network 1

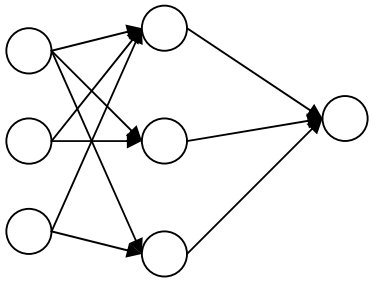

Fig. 7. Three no of neurons in the hidden layer

Network Parameters
Number of input neurons: 3
Number of hidden layers: 1
Number of Output Neurons: 1
Network Type: Feed forward Back propagation
Transfer Function: Tansig, Purelin

Table 1
Trainrp showing various training parameters

Training algorithm	Shows	Epoch	Goal	MSE	Neurons
Trainrp	20	73/400	0.001	0.0009713/0.001	6
Trainrp	30	59/500	0.001	0.0009890/0.001	8
Trainrp	30	103/500	0.0001	9.913e-005/0.0001	10
Trainrp	40	600/600	0.0001	0.0001905/0.0001	12
Trainrp	50	792/800	0.0001	9.992e-005/0.0001	20

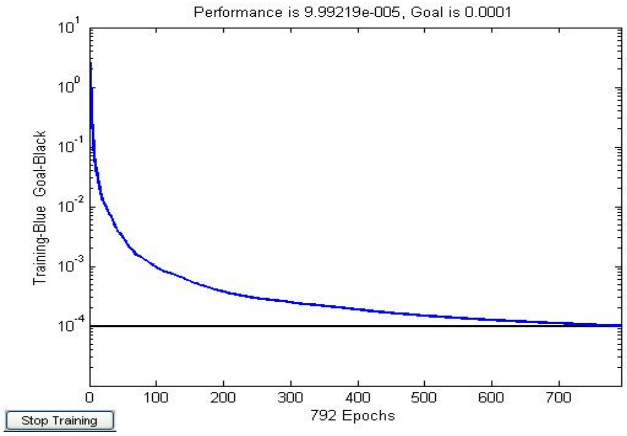

Fig. 8: Trainrp algorithm showing performance, goal and epoch

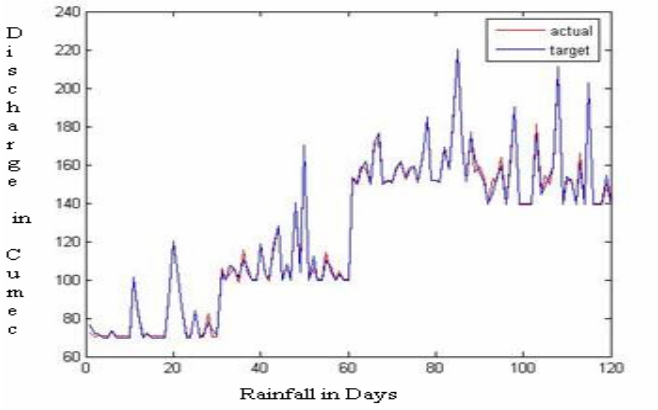

Fig. 9. Actual and Target discharge values

In the above attempt we have tried with Rainfall of both the Guage stations and minimum discharge as the input and discharge as the output. We have simulated the model using Trainrp algorithm with different training parameters. The Performance function with different training parameters like Goal, Epoch etc are shown above. Mean Square Error has been reduced up to 0.0067.In the next attempt Traincgf algorithm is attempted.
The Network is similar to above network

Network Parameters
Number of input neurons: 3
Number of hidden layers: 1
Number of Output Neurons: 1
Network Type: Feed forward Back propagation
Transfer Function: Tansig, Purelin

Table 2
Traincgf showing various training parameters

Training algorithm	Shows	Epoch	Goal	MSE	Neurons
Traincgf	20	42/400	0.001	0.000939/0.001	6
Traincgf	30	75/500	0.001	0.000984/0.001	8
Traincgf	30	77/500	0.001	0.000928/0.001	10
Traincgf	40	542/600	0.0001	9.97e-005/0.0001	12
Traincgf	50	436/800	0.0001	9.90e-005/0.0001	15

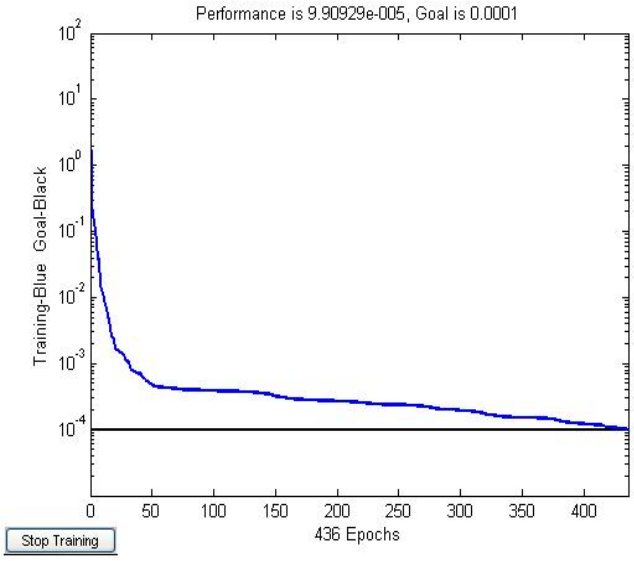

Fig. 10. Traincgf algorithm showing performance, epoch and goal

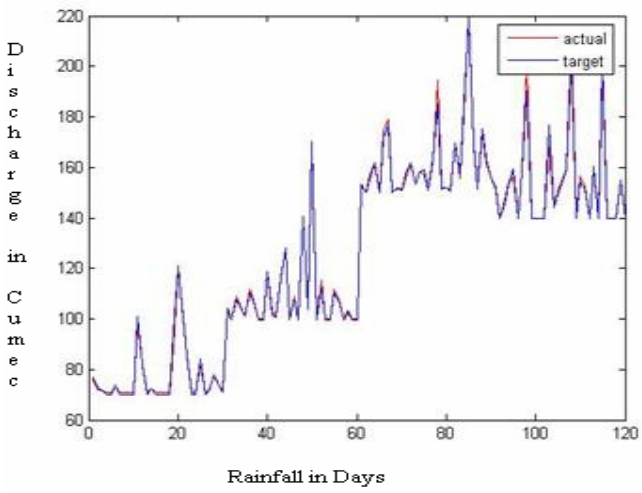

Fig. 11. Actual and Target discharge values

In the above attempt we have tried with Rainfall of both the Gauge stations and minimum discharge as the input and discharge as the output. We have simulated the model using Traincgf algorithm with different training parameters. The Performance function with different training parameters like Goal, Epoch etc are shown above. Mean Square Error has been reduced up to 0.0066**.**In the next attempt Trainlm algorithm is attempted.

Network 2

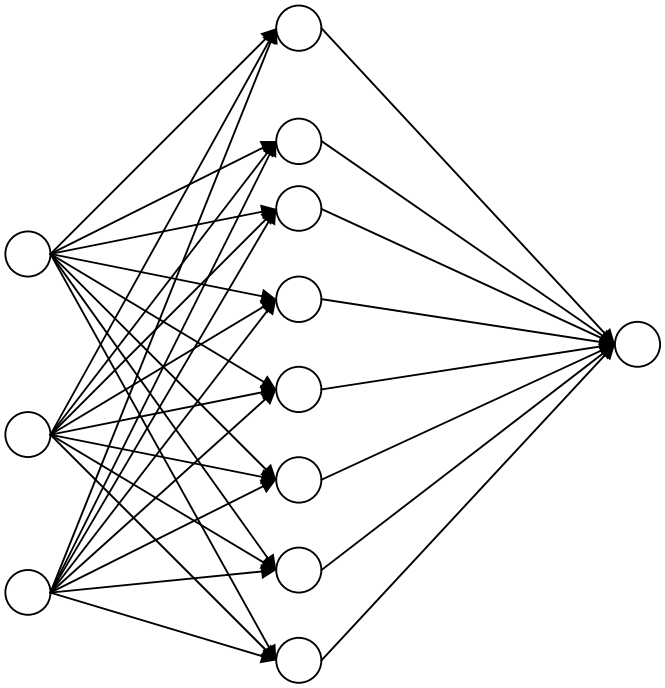

Fig.12. Eight no of neurons in the hidden layers

Network Parameters
Number of input neurons: 3
Number of hidden layers: 1
Number of neurons in hidden layer: 8, 10, 15, 18
Number of Output Neurons: 1
Network Type: Feed forward Back propagation
Transfer Function: Tansig, Purelin

Table 3
Traincgf showing various training parameter

Training algorithm	Shows	Epoch	Goal	MSE
Traincgf	30	29/300	0.0001	9.9563e-005/0.0001
Traincgf	40	55/400	0.0001	9.9563e-005/0.0001
Traincgf	40	93/500	0.0001	9.8562e-005/0.0001
Traincgf	50	115/600	0.0001	9.6705e-005/0.0001

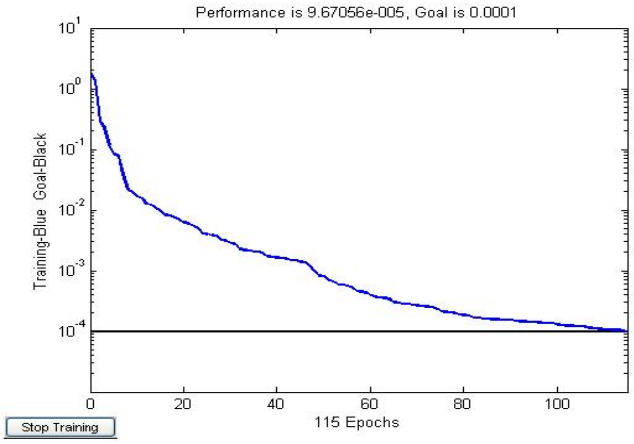

Figure13. Traincgf algorithm showing performance, goal and Epoch

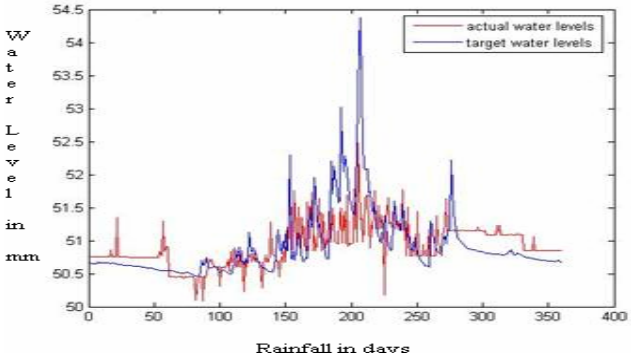

Fig. 14. Actual and Target Water Levels with one day lead

In the above attempt we have tried with Rainfall of both the gauge stations and minimum water level as the input and water level as the output. We have simulated the model using Traincgf algorithm with different training parameters. The Performance function with different training parameters like Goal, Epoch etc are shown above. Mean Square Error has been reduced up to 0.065**.**Actual and Target predicted water levels are shown. The table below shows various training parameters of Trainoss algorithm after simulation.

Table 4
Trainoss showing various training parameters

Training algorithm	Shows	Epoch	Goal	MSE
Trainoss	30	72/300	0.0001	8.8566e-005/0.0001
Trainoss	40	127/400	0.0001	9.9274e-005/0.0001
Trainoss	50	211/500	0.0001	9.9980e-005/0.0001
Trainoss	50	169/600	0.0001	9.8325e-005/0.0001

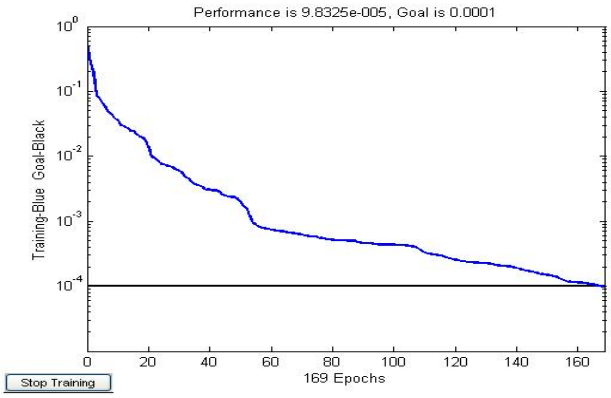

Fig. 15. Trainoss Algorithm showing performance, goal and Epoch

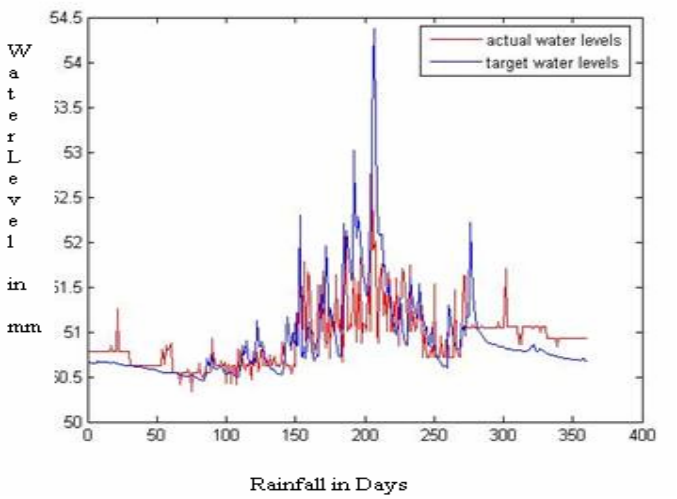

Fig. 16. Actual and Target Water Levels with one day lead

In the above attempt we have tried with Rainfall of both the gauge stations and minimum water level as the input and water level as the output. We have simulated the model using Trainoss algorithm with different training parameters. The Performance function with different training parameters like Goal, Epoch etc are shown above. Mean Square Error has been reduced up to 0.066**.**Actual and Target predicted water levels are shown. The table below shows various training parameters of Trainlm algorithm after simulation.

Table 5
Trainlm showing various training parameters

Training algorithm	Shows	Epoch	Goal	MSE
Trainlm	20	3/100	0.0001	6.4812e-005/0.0001
Trainlm	20	6/100	0.0001	7.5832e-005/0.0001
Trainlm	10	2/200	0.0001	9.4748e-005/0.0001
Trainlm	10	3/100	0.0001	6.9162e-005/0.0001

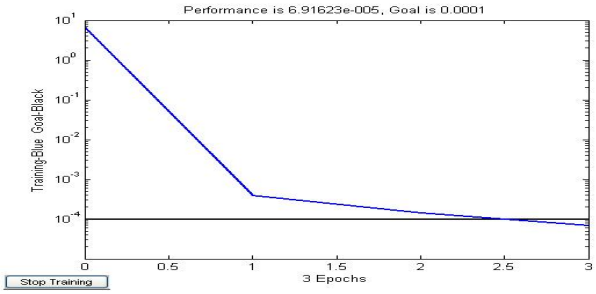

Fig. 17.Trainlm Algorithm showing performance, goal and Epoch

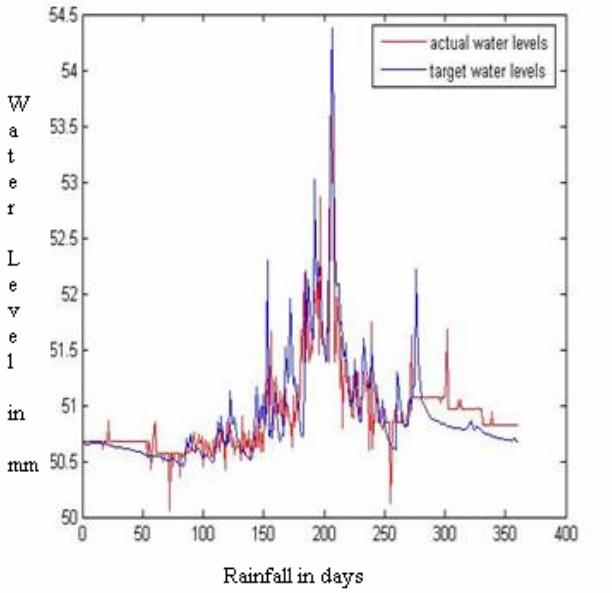

Fig. 18. Actual and Target water Levels with one day lead

In the above attempt we have tried with Rainfall of both the gauge stations and minimum water level as the input and water level as the output. We have simulated the model using Trainlm algorithm with different training parameters. The Performance function with different training parameters like Goal, Epoch etc are shown above. Mean Square Error has been reduced up to 0.056**.**Actual and Target predicted water levels are shown.

6. Conclusions

From all the above algorithms Trainlm converges very quickly in training data set. Mean Square Error has been significantly reduced in Trainlm algorithm with one day lead forecasting. It could reduce the error significantly if we increase the hidden neurons in the hidden layer upto 18. Since the lag gap between rainfall and water level is observed to be two days we are performing two day lead forecasting also. In testing case the mean square error could be reduced to 0.045 by Trainlm algorithm, but the other two algorithms Traincgf and Trainoss could reduce up to 0.06. The best model in testing case is 3-20-1. In both the cases Training, as well as Testing, peak water level is reached in the Trainlm algorithm after simulation of the model.

References

Ahmed, J.A. and A.K. Sarma, 2007. Artificial neural network for synthetic stream flow generation. *Water Resour. Manage.*,21(6),1015-1029.

ASCE Task Committee on Application of Artificial neural networks in Hydrology, "Artificial neural networks in hydrology (I & II)", Journal of Hydrologic Engineering, 5(2), April 2000, pp. 115-136.

Bevan, K.J., M.J. Kirby, N. Schofield and A.F. Tagg, 1984. Testing a physicically-based flood forecasting model (TOP MODEL) for three UK catchments. *J. Hydrol.,* 69: 119-143.
http://adsabs.harvard.edu/abs/1984JHyd...69..119B

Brath, A., A. Montanari and E. Toth, 2002. Neural networks and nonparametric methods for improving real-time flood forecasting through conceptual hydrological models. Hydrol. Earth Syst. Sci., 6(4), 627-639, 2002. doi:10.5194/hess-6-627-2002

Burnash, J. (1973) A generalized stream flow simulation system. Joint federal-state River forecast center.

Crawford, N.H and Linsley, R.K (1966).Digital Simulation in Hydrology. Stanford Watershed Model 4, TR 39.Department of Civil Engineering, Stanford.

Dawson, C.W. and R. Wilby, 1998. An artificial neural networks approach to rainfall-runoff modeling, *Hydrol. Sci. J.*, 43, 47-66.

F.J. ChangandY.-C.Chen, 2001 A Counter propagation fuzzy-neural network modeling approach to real time stream flow prediction,J.Hydro" vol.245,pp.153-164,2001.

Hoque, M.M., 1994. "Evaluating design characteristics for floods in Bangladesh."*Proceedings of the 2nd International Conference on River Flood Hydrology, (ICRFH'94),* Wiley, Chichester, England and New York, pp: 15-26

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

K.Horn & M. Stinchcombe and H. White, 1989 Multilayer feed forward networks are universal approximators, Neural Networks 2, 359-366, (1989).