ANN Based Hydraulic Modeler for Flood Prediction

[1]Kulkarni Nilesh K., [2]V. V. Shete

*Dept. Of Electronics and Telecommunication
*MIT College of Engineering, University of Pune, Pune, India
Email: [1]nileshkk@gmail.com, [2]vvshete100@gmail.com

Abstract— Prediction of flood water level is a difficult task as nonlinearity is involved with river discharge. Working with Artificial Neural Networks (ANN) has proved to be most convenient and easy tool in order to model and analyze the non-linear events in recent years of research. ANN has ability to model non-linear events and thus turn out to be very important in hydrology to model various hydrological events that are dominantly non-linear in nature. ANN works with modeling Non-linear relationship between Rainfall and Runoff as compared to other techniques available in Mathematical modeling. In this paper, MATLAB is used for ANN modeling for prediction of water level at the reservoir. Back Propagation Network (BPN) model is used to evaluate error and back propagate it for more accurate training of ANN.

Index Terms—Artificial Neural Networks (ANN); Back Propagation Neural Network (BPN); Root Mean Square Error (RMSE).

I. INTRODUCTION

Flood prediction and warning system plays significant role in reducing the loss of life and movable property during floods. Working with flood prediction system is a complicated process in environmental modeling mostly because of the variability of river discharge, rainfall patterns, topographical characteristics, and the number of parameters considered for the calibration of the system. The predictions here depend on the long-term observation and runoff values of precipitation process. As there is nonlinearity involved in Hydrological cycle it makes hydrological modeling process very complicated. Design of flood control system involves Rainfall –Runoff modeling as it plays an important role in getting the river discharge values and water resources. An Artificial Neural Network (ANN) is a universal approximator since it can learn and generalize the data from sufficient data pairs. Thus ANN can solve large-scale complex problems such as hydrological time series predictions, nonlinear modeling, classification, association, control, pattern recognition etc. ANN models can approximate any measurable function to an arbitrary degree of accuracy thus are used for hydrological time series predictions. Therefore they find application in flood forecasting systems. ANN maps non-linear relationship between inputs and outputs and perform better than Process-based models thus find an application in the field of management of water resources. ANN is a useful tool in hydrological modeling such as modeling of rainfall - runoff processes, flow prediction, operation of reservoir systems etc. and thus are the effective part of the study. The objective of the present study is to develop flood prediction system using Artificial Neural Network (ANN) methods.

II. LITERATURE REVIEW

William James was the first to publish work on brain activity patterns in 1890. Since 1990s Artificial Intelligence (AI) has been popular and is being used widely. The model of the neuron that is still used today in artificial neural networking was produced by McCulloch and Pitts in 1943. Donald Hebb was the one to publish “The Organization of Behavior” which stated the law for synaptic neuron learning in 1949. It was later known as “Hebbian Learning” in his honor and turned out to be the simplest and most straight-forward learning rule for artificial neural networks. Marvin Minsky and Seymour Papert discovered two key issues with computational machines that processed neural networks, first issue being the exclusive-or circuit cannot be solved by single layer neural network. The second issue was the long run time required by large neural networks. The research stagnated after this publication until computers achieved greater processing power. The back propagation neural network was the key advancement which solved the exclusive-or problem. Disastrous flood events occur in many parts of the world and it is of utmost importance to work on hydraulic flood modeler and setting up forecasting centers on interstate rivers.

Back-propagation neural network (BPNN) is the most popular neuron network, which can be applied in flood water level prediction system successfully. BPNN technique is capable of modeling various characteristics of hydrologic resources system, including randomness and non-linearity. BPNN does the function approximation through training a network by input vector and corresponding output vector and consists of input layer, hidden layer and output layer, and it propagates backward the error at the output layer to the input layer through the hidden layer to decrease the global error.

III. PROPOSED SYSTEM ARCHITECTURE

This paper proposes a model that uses Artificial Neural

Networks (ANN) for flood water level prediction system. Here, the Artificial Neural Networks (ANN) along with the back propagation is used to build the Hydrological flood model. ANN as explained above is made up of highly interconnected neural computing elements and is a parallel distributed processing system.

Fig. 1 below shows the proposed system architecture.

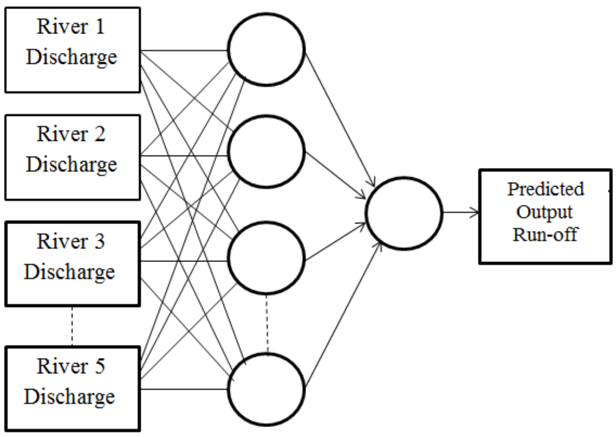

Fig. 1 Runoff Prediction ANN Model

Fig. 1 shows 5:5:1 ANN system i.e. 5 input layer neurons to which 5 river discharges are applied, 5 hidden layer neurons which reduce computational complexity and improves simulation speed and 1 output layer neuron which gives output runoff. Type of ANN used is feed-forward network as it most widely used in modeling non-linearity.

The Fig. 1 shows Runoff Prediction Model using ANN where input quantities fed to the input layer neurons that in turn, pass them on to the hidden layer neurons after multiplication by interconnection weights. A hidden layer neuron adds up the weighted input received from each input neuron and associates it with bias.

A three layered ANN is considered for this system: the input layers, where the river discharge data are introduced to the network; one or more hidden layers, where the processing of data takes place and the output layer produces the results for given inputs. The nodes that ANN consists are the neurons which possess the information and are the processing units. The signals are transmitted with help of connecting links these links possess an associated weights that are multiplied along with the incoming signal which is referred to as Net Input. By applying activation functions to the net input the outputs signal can be obtained. ANN can acquire knowledge through various learning mechanisms.

Each layer constitutes of several nodes. These layers are connected to each other by sets of correlation weights. Here, each of the input node unit (i=1,…,m) in input layer broadcasts the input signal to the hidden layer and each hidden node (j=1,…,n) sums its weighted input signals and sends this signal to all units in the hidden layer.

The result is produced by passing the net value of a neuron is passed through an activation or transfer function. Therefore, continuous-transfer functions are desirable for such a system. The transfer function defines the output of a neuron in terms of the activity level at its input and is denoted by f(k).

Several commonly used activation functions are stated below

- ➢ The Identity function
- ➢ The Binary Step function
- ➢ The Binary Sigmoid (Logistic) function
- ➢ The Binary Sigmoid (Hyperbolic Tangent) function

The transfer function used in the present paper is sigmoidal which is continuous and differentiable and is monotonically increasing function, and it is the most commonly used in the back propagation networks. 0 and 1 is the value to which the output is always bounded and the input data has to be normalized to a range between 0 to 1. The sigmoid activation function will process the signal that passes from each node then from second layer the signal is transmitted to third layer. The error information is transferred from the output layer back to early layers. This is known as the back propagation of the output error to the input nodes to correct the weights.

A. Learning Process

Learning rule implies to a procedure for modifying the weights and biases of a network. In order to perform a task the network must be trained and the learning rule helps to train the network.

They fall into three broad categories:

- ➢ Supervised learning
- ➢ Reinforcement learning
- ➢ Unsupervised learning

In this proposed system supervised learning process is being used. For proper network behavior this learning rule provides a set of training data. The inputs applied to the network and the network outputs are compared to the targets. The learning rule is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets

B. Training of ANNs

Number Configuration of a neural network is done in such way that the desired set of outputs is produced by applying of a set of inputs. Different methods are used to set the strengths of the connections. it can be achieved either by setting the weights explicitly, using a priori knowledge or another way is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule.

Once the network is structured for a particular application, it is ready to be trained. Initial weights are chosen randomly and then the training or learning begins. Thus, ANN offers different way to analyze data and recognize patterns within that data.

C. Back Propagation

Define Back propagation is generalization of Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vector in an appropriate way as defined. Multi-layer feed-forward network is advantageous as it can overcome many restrictions compared to single layer network. The problem of how to adjust the weights from input to hidden units still exists. The answer is to back-propagate the errors of units of output layer to determine the errors for the units of the hidden layer and thus this method is called the back-propagation learning rule. For non-linear activation functions and multilayer networks Back-propagation is considered as a generalization of the delta rule. Back-propagation is the supervised training algorithm which is commonly used in the multilayer feed-forward networks. The network weights are modified by minimizing the error between a target and computed outputs. The weights are updated continuously until minimum error is achieved. A training pair is selected from the training set and applied to the network. The outputs are calculated by the based on the inputs provided in this training pair. The resultant outputs from the network are then compared with the expected outputs identified by the training pair. The weights and biases of each neuron are then adjusted by a factor based on the derivative of the sigmoid function, the differences between the expected network outputs and the actual outputs (the error), and the actual neuron outputs. Through these adjustments it is possible to improve the results that the network generates, and thus the network is seen to learn. How much each neuron's weights and bias are adjusted in the back propagation algorithm also depends on a learning parameter—which is nothing but the learning Rate (α) and it is a single factor by which all adjustments are multiplied. A large learning rate can result in training oscillation from one poor extreme result to another and small learning rate can lead to a situation where the network does not learn anything and is caught in a local minimum, unable to reach a more accurate set of weights. So while training the ANN proper selection of learning rate is very important.

D. Procedure for Back propagation

Back propagation error can be calculated as,

$$ek = (tk - yk) f(y_{-ink}) \quad (1)$$

Where,

ek- error information,

tk- output target unit k,

yk- output unit k.

E. Weight and Biases Updation

Use Each output unit (yk) updates its bias and weight to minimize error between output and target.

1. The Weight Correction is given by,

$$\Delta W_{jk} = \alpha e_k Z_j \quad (2)$$

Where α is learning rate

Thus,

$$W(new) = W_{jk}(old) + \Delta W_{jk} \quad (3)$$

2. Bias Correction is Given by,

$$\Delta b_{ok} = \alpha e_k \quad (4)$$

Thus,

$$b_{ok}(new) = b_{ok}(old) + \Delta b_{ok} \quad (5)$$

Back propagation algorithm is thus divided in two phases i.e. propagation and weight update.

The ratio influences the speed and quality of learning; known as learning rate. If the ratio is greater network trains faster and if the ratio is low the training is more accurate.

IV. RESULTS AND DISCUSSION

For predicting peak discharge, the water discharges are the inputs and runoff rate is the target output for ANN. Since ANN effectively works on data values between 0 and 1, all the external input and output data are to be standardized.

Tables I below shows the discharge data of 5 rivers for a particular catchment area. Catchment area may vary from several thousand square kilometers to only few hectares.

Table I shows river discharge data of 5 rivers.

Discharge in m^3/s				
River 1	River2	River3	River4	River5
23.02	267.24	266.47	130.51	85.02
71.59	441.34	508.31	508.37	331.17
125.02	570.47	690.17	947.78	617.41
239.73	668.16	613.14	1179.38	768.28
374.90	731.90	536.11	1182.61	770.90
584.13	663.58	485.72	1047.29	682.24

860.18	588.62	435.54	938.02	611.05
1098.72	511.17	382.04	838.80	546.42
1388.97	485.78	358.31	749.82	488.46
1407.20	445.34	334.97	693.42	451.71
1254.57	412.21	331.66	656.41	427.61
1124.79	342.94	254.83	600.66	391.29
1006.32	354.45	254.43	529.08	344.65
924.59	291.54	207.76	469.06	305.56
838.04	280.87	227.78	435.66	283.80
766.38	269.31	207.56	415.46	270.64
717.59	256.64	204.45	412.55	268.75
653.52	250.62	204.24	392.23	255.51

Since ANN effectively works on data values between 0 and 1, all external input and output data are to be standardized by,

$$Zi = \frac{Xi}{Xmax + 1}$$

Where,

Z_i= Standardized value calculated for neuron i;

X_i= real input applied to neuron i; and

X_{max}= maximum input applied to neuron i.

Table II shows normalized Discharge Data

Normalized Discharges Data				
River1	River2	River3	River4	River5
0.16	0.36	0.39	0.11	0.11
0.05	0.60	0.74	0.43	0.43
0.09	0.78	1.00	0.80	0.80
0.17	0.91	0.89	1.00	1.00
0.27	1.00	0.78	1.00	1.00
0.41	0.91	0.70	0.88	0.88
0.61	0.80	0.63	0.79	0.79
0.78	0.70	0.55	0.71	0.71
0.99	0.66	0.52	0.63	0.63
1.00	0.61	0.48	0.59	0.59
0.89	0.56	0.48	0.55	0.55
0.80	0.47	0.37	0.51	0.51
0.71	0.48	0.37	0.45	0.45
0.66	0.53	0.30	0.40	0.40
0.60	0.38	0.33	0.37	0.37
0.54	0.37	0.30	0.35	0.35
0.51	0.35	0.30	0.35	0.35
0.46	0.34	0.30	0.33	0.33

Following results are evaluated using Matlab software package.

```
Shortcuts  How to Add  What's New

TRAINLM, Epoch 0/150, MSE 1.25731/0, Gradient 35.8605/1e-010
TRAINLM, Epoch 25/150, MSE 3.88712e-005/0, Gradient 0.00477806/1e-010
TRAINLM, Epoch 50/150, MSE 1.42445e-005/0, Gradient 0.0148252/1e-010
TRAINLM, Epoch 75/150, MSE 5.91897e-007/0, Gradient 0.000216824/1e-010
TRAINLM, Epoch 100/150, MSE 2.3128e-009/0, Gradient 0.00100926/1e-010
TRAINLM, Epoch 103/150, MSE 2.20538e-029/0, Gradient 5.85225e-015/1e-010
TRAINLM, Minimum gradient reached, performance goal was not met.
```

```
Std_Actual_Desired =                 Actual_Desired =

    0.1460    0.1460                   1.0e+003 *
    0.4325    0.4325
    0.6970    0.6970                     0.5929    0.5929
    0.8083    0.8083                     1.7563    1.7563
    0.8166    0.8166                     2.8306    2.8306
    0.8181    0.8181                     3.2824    3.2824
    0.8233    0.8233                     3.3162    3.3162
    0.8151    0.8151                     3.3224    3.3224
    0.8262    0.8262                     3.3435    3.3435
    0.9997    0.9998                     3.3102    3.3102
    0.7268    0.7268                     3.3553    3.3553
    0.6110    0.6110                     4.0601    4.0601
    0.5660    0.5660                     2.9518    2.9518
    0.5409    0.5409                     2.4813    2.4813
    0.4405    0.4405                     2.2986    2.2986
    0.4250    0.4250                     2.1967    2.1967
    0.4211    0.4211                     1.7889    1.7889
    0.3989    0.3989                     1.7260    1.7260
    0.3710    0.3710                     1.7102    1.7102
    0.3684    0.3684                     1.6201    1.6201
                                         1.5067    1.5067
                                         1.4961    1.4961
```

Fig. 2 ANN Training Outputs

Fig. 2 shows the ANN training outputs in the form of standard and desired values.

```
MATLAB
File  Edit  Debug  Desktop  Window  Help
Current Directory: C:\Documer
Shortcuts  How to Add  What's New
Layer1_Weights =

   10.6491    5.2699   -1.8842   -3.0699   -7.3065
   -2.8198    5.6214    8.2634    3.1506   -5.0480
   -4.3829   -2.2942    3.1637    6.0034    5.1439
    2.2288    5.5987    6.6750   -1.8388   -7.2976
    7.4304    0.6321   -1.4534    5.3488   -3.5152


Layer1_Biases =

   -8.7108
   -2.9252
   -5.5629
   -5.0645
    0.0075


Layer2_Weights =

   -3.7467    2.1655   -0.2621   -3.4382   -0.7600


Layer2_Bias =

    6.0883
    1.7073
   -3.1593
    0.2852
   -2.5090
```

Fig. 3 Final Weights and Biases

Fig. 3 shows final weights and biases obtained after successful training of ANN.

Figure below shows system performance in the form of error between Desired Training Goal and achieved Goal against user defined epochs. This error is then back propagated to achieve desired Goal. Here error is decreasing after number of training iterations called as Epochs.

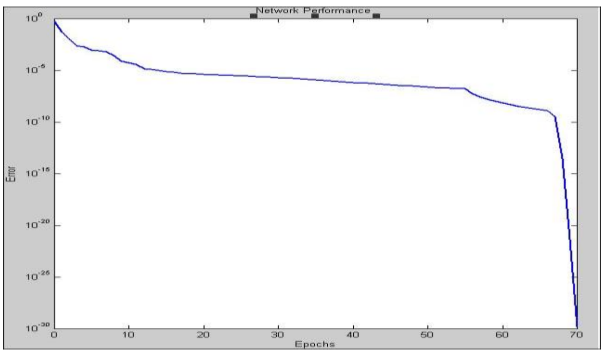

Fig. 4 ANN Performance Plot

Fig. 4 shows that how error drops down with increasing number of training iterations or in the other words EPOCHS. At the end of training we still get an error 10-23 in 87 iterations but it is very less almost equal to zero.

V. ROC CURVE ANALYSIS

The Receiver operating characteristic (ROC) is a metric used to check the quality of the classifiers. ROC applies threshold values across the interval [0, 1] to outputs for each class of classifier.

Two values are calculated for each threshold
- True Positive Ratio (TPR)
- False Positive Ratio (FPR)

TPR is nothing but no. of outputs greater or equal to threshold, divided by the no. of one targets and FPR is the no. of outputs less than the thresholds, divided by no. of zero targets.

Figure below shows the performance curve for classifier output that computes ROC.

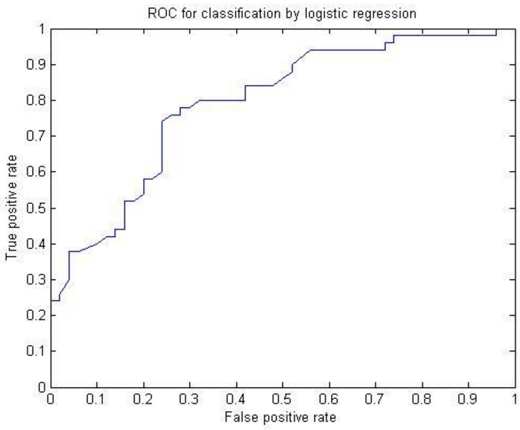

Fig. 5 ROC Curve

Here, in Fig. 5 we see the best point gives 0.8 TPR i.e. sensitivity (80%) and 0.3 FPR i.e. specificity (77%).

Table III displays optimized parameters for the BPN model

Network Configuration	[5,11,1] 'logsig','logsig','purelin'
Learning rate	0.01
Momentum constant	default
Training algorithm	Gradient Decent Back Propagation
Training goal	0.00
No. of epochs	150

VI. CONCLUSION

The In this paper, research on ANN is being carried out for a Hydrological Flood Modeler to predict the flood water level. Back Propagation learning rule is being used to optimize error by evaluating and back propagating error for more accuracy in training. After successful training of this ANN model, the results can be used to control Reservoir operations such as flood control structures and water usage management.

ACKNOWLEDGMENT

The authors would like to thank and acknowledge the support form Department of Electronics and Telecommunication, MIT College of Engineering, University Of Pune, India.

REFERENCES

[1] Ramli Adnan, Fazlina Ahmat Ruslan, Abd Manan Samad & Zainazlan Md Zain, "Extended Kalman Filter (EKF) prediction of flood waterlevel," IEEE Control and System Graduate Research Colloquium(ICSGRC), pp. 171-174, 2012

[2] Xian Luo, You-peng Xu, Jin-tao Xu, "Regularized Back-Propagation Neural Network for Rainfall-Runoff Modeling", International Conference on Network Computing and Information Security, IEEE 2011

[3] Luciano Carli Moreira de Andrade, Ivan Nunes da Silva "Very Short-Term Load Forecasting Using a Hybrid Neuro-Fuzzy Approach", 2010 IEEE, Eleventh Brazilian Symposium on Neural Networks.

[4] Singh, A., Panda, R.K., Pramanik, N., "Appropriate data normalization range for daily river flow forecasting using an artificial neural network," Hydro informatics in Hydrology, Hydrogeology and Water Resources. International Association of Hydrological Sciences Publication, UK, pp. 51-57, 2009

[5] Jingwen XU, Junjang Zha, Wanchang Zhang, Zhongda HU, Ziyan Zheng,"Mid-short- term daily runoff forecasting by ANNs and multiple process-based hydrological models", IEEE 2009

[6] R.Remesan, M. A. Shamim, D. Han, J. Mathew"ANFIS and NARX based Rainfall-Runoff Modelin g" 2008 IEEE, International Conference on Systems,Man and Cybernetics (SMC 2008)

[7] Gokmen Tayfur and Vijay P. Singh "ANN and Fuzzy Logic Models for Simulating Event-Based Rainfall-Runoff "Journal of Hydraulic Engineering © ASCE / December 2006

◈◈◈