ARPN Journal of Science and Technology
©2011-2012. All rights reserved.

http://www.ejournalofscience.org

Fuzzy Based Flood Control System with Prediction Using ANN for a Physical Hydraulic Model

[1] Yogesh Shirke, [2] Prof.Dr.Rameshwar Kawitkar, [3] Selva Balan
[1,2] Department of Electronics and Telecommunication, Sinhgad College of Engg., UOP, India.
[3] CWPRS, Pune, India.
[1] yogesh.shirke@gmail.com
[2] rskawitkar@rediffmail.com
[3] msels72@gmail.com

ABSTRACT

Hydrological cycle is a highly nonlinear system which makes hydrological modeling very complicated. In recent years Artificial Neural Networks (ANN) proved to be most convenient and easy tool for modeling and analysis of such non-linear events. This paper describes an application of ANN to model these predominent non-linear Hydroligical events. Back Propagation (BP) is used for more accurate training of ANN. Here, Fuzzy technique and its interfacing with ANN is also overviewed. This research study is being performed at Central Water and Power Research Station (CWPRS), Khadakwasla (Pune) on Bombay Model.

Keywords: *Artificial Neural Network (ANN), Back Propagation (BP), Fuzzy Technique.*

1. INTRODUCTION

Modeling of non-linearity and uncertainty associated with Hydrological process has lot of importance.Hydrological Event modeling has a significant role in operational flood management procedures likeflood forecasting, flood warning and design of hydraulic systems. This modeling also plays an important role in flood control, water resources and water environment management [1]. An ANN (Artificial Neural Network) is capable to learn and generalize ‘‘data’’ from sufficient data pairs. This makes ANN a powerful tool to solve large-scale complex problems such as pattern recognition, nonlinear modeling, classification, association, control, hydrology and many other. ANN models are well suited for hydrological modeling since they can approximate virtually any measurable functions almost accurately. Thus daily runoff forecasting based on artificial Neural Network (ANN) models has become quite important for effective planning and management of water resources. ANN models perform better than Process-based models .[2] Several studies indicate that ANN have proven to be potentially useful tools in hydrological modeling such as for modeling of rainfall-runoff processes, flow prediction, water quality predictions, operation of reservoir system etc. The objective of the present study is to develop rainfall-runoff models using ANN methods.

This paper is organised as Literature Review, Proposed System Architecture, Conclusion and Discussion, Acknowledgement and References.

2. LITERATURE REVIEW

There are various **rainfall-runoff models**. Many of them are conceptual models or physically-based models which simulate rainfall-runoff process based on meteorological and physiographical characteristics. It is usually difficult with lack of related data or enough hydrology knowledge to apply these models. To resolve these problems, hydrologists have adopted artificial neural networks (ANNs).ANNs plays an important role in hydrology nowadays and they can solve complex nonlinear problems due to their powerful approximation ability. With the help of ANN, flow can be predicted without any physical process. [3]

Artificial Intelligence (AI) has been popular since 1990s and has been widely used in many areas. In 1890, William James published the first work about brain activity patterns. In 1943, McCulloch and Pitts produced a model of the neuron that is still used today in artificial neural networking. In 1949, Donald Hebb published —’The Organization of Behavior’, which outlined a law for synaptic neuron learning. This law, later known as “Hebbian Learning” in honor of Donald Hebb, which is one of the simplest and most straight-forward learning rule for artificial neural networks. In 1951, Marvin Minsky created the first ANN while working at Princeton. In 1958 —’The Computer and the Brain’ was published, a year after John von Neumann‘s death. In that book, von Neumann proposed many radical changes to the way in which researchers had been modeling the brain.

Back-propagation neural network (BPNN) is the most popular neuron network, which can apply in rainfall-runoff modeling successfully. BPNN technique has the capability to model various characteristics of hydrologic resources system, including randomness, fuzziness, non-linearity, etc. BPNN is usually used for function approximation through training a network by input vector and corresponding output vector. A BPNN 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 global error. [3]

Fuzzy sets were developed in the mid-1960s. Development of fuzzy sets made many-valued logic more useful in logic reasoning of an intelligent system. Fuzzy logic can be considered as an extended set many-valued logic. Fuzzy sets and fuzzy logic have been developed to manage non-linearity and uncertainty in a reasoning process of an intelligent system such as a knowledge based system, an expert system, or a logic control system. Uncertainty occurs when the truth of a predicate is neither true nor false, but a value between true and false.

3. PROPOSED SYSTEM ARCHITECTURE

This paper includes a Hybrid system of Artificial Neural Networks (ANN) with Back propagation Algorithm-for Modeling of Hydrological event and Fuzzy technique-for Dam Gate operation.

A. Artificial Neural Networks (ANN):

To simplify the problems of prediction, neural networks are being introduced. ANN is expert at mapping non-linear relationship between inputs and outputs. It is a parallel distributed processing system made up of highly interconnected neural computing elements. ANN has the ability to learn and acquire knowledge. Fig.1 shows .Proposed System Architecture.

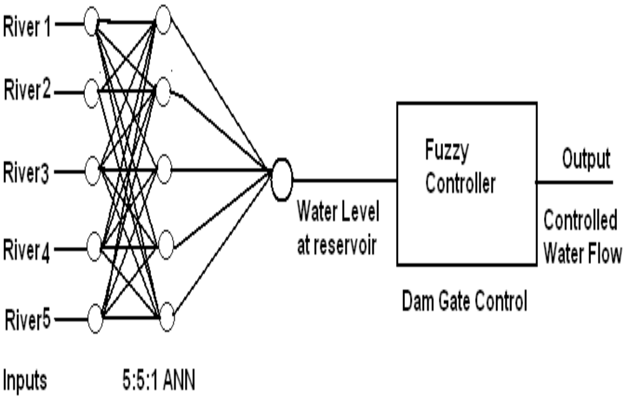

Fig.1: Proposed System Architecture

As shown in fig.1, Proposed System mainly consists of hybrid Neuro-Fuzzy model of 5:5:1 ANN and Fuzzy Controller. ANN consists of three layers: the input layers, where the data are introduced to the network; one or more hidden layers, where the data are processed and the output layer, where the results for given inputs are produced. ANN consists of many nodes, which are processing units called neurons. These Neurons, processes the information. The signals are transmitted by means of connecting links. The links posses associated weights, which are multiplied along with the incoming signal (Net Input).Output Signal is obtained by applying activation functions to the net input. Various learning mechanisms exist to enable the ANN to acquire knowledge.

Each layer is made up of several nodes, and layers are interconnected by sets of correlation weights. Each input node unit (i=1,…,m) in input layer broadcasts the input signal to the hidden layer. Each hidden node (j=1,…,n) sums its weighted input signals according to

$$Z_{\_inj} = b_j + x_i W_{ij}$$

applies its activation function to compute its output signal from the input data as

$$Z_j = f\,(Z_{\_inj})$$

and sends this signal to all units in the hidden layer. Where wij is the weight between input layer and hidden layer, b_j is the weight for the bias and x_i is the input signal.

The net of a neuron is passed through an activation or transfer function to produce its result. Therefore continuous-transfer functions are desirable. The transfer function, denoted by f (k), defines the output of a neuron in terms of the activity level at its input. There are several commonly used activation functions defined as

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 report is sigmoid which continuous, differentiable, monotonically increasing function is, and it is the most commonly used in the back propagation networks. The output is always bounded between 0 and 1, and the input data have been normalized to a range between 0 to 1. The Slope a is taken assumed to be 1.The sigmoid activation function will process the signal that passes from each node by

$$f\,(Z_{inj}) = 1 / (1+ e^{-Zinj})$$

Then from second layer the signal is transmitted to third layer. The error information is transfer 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 Processes:

By learning rule we mean a procedure for modifying the weights and biases of a network. The purpose of learning rule is to train the network to perform some task. They fall into three broad categories:

- Supervised learning
- Reinforcement learning
- Unsupervised learning

In this proposed system supervised learning process is being used. This learning rule is provided with a set of training data of proper network behavior. As the inputs are applied to the network, 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 an ANNs

A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule.

c) Back Propagation

A single-layer network has certain limitations, whereas as a two or multi-layer feed-forward network can overcome many restrictions. But still it did not present a solution to the problem of how to adjust the weights from input to hidden units. An answer to this question was presented by Rumelhart, Hinton and Williams in 1986 and similar solutions appeared to have been published earlier. The central idea behind this solution is that the errors for the units of the hidden layer are determined by back-propagating the errors of the units of the output layer. And hence this method is often called the back-propagation learning rule. Back-propagation can also be considered as a generalization of the delta rule for non-linear activation functions and multilayer networks.

Back propagation is the most commonly used supervised training algorithm in the multilayer feed-forward networks. The network weights are modified by minimizing the error between a target and computed outputs. The error between the output of the network and the target outputs are computed at the end of each forward pass. If an error is higher than a selected value, the procedure continuous with a reverse pass, otherwise, training is stopped. The weights are updated continuously until minimum error is achieved. The basis of the back-propagation algorithm is that, a training pair is selected from the training set and applied to the network. The network calculates the output 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 oscillatation from one poor extreme result to another, whereas a 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 proper selection of learning rate is most important before training the ANN.

B. FUZZY SYSTEM

Fuzzy logic (FL) is a super set of conventional (or Boolean) logic and contains similarities and differences with Boolean logic. Boolean logic is similar to Boolean logic, in that Boolean logic results are returned by fuzzy logic operations when all fuzzy memberships are restricted to 0 to 1.FL is problem-solving control system methodology. FL leads itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or Workstation based data acquisition and control system. It can be implemented in hardware, software, or combination of both. FL provides simple way to arrive at a definite conclusion based on noisy or missing input information. Fuzzy systems are made up of fuzzy sets, defined by their membership functions and fuzzy rules that determine the action of fuzzy systems. Fuzzy systems can model general nonlinear mappings in a manner similar to feed forward neural networks.

The principle of designing a fuzzy logic controller is to integrate knowledge and operator experience into controllers by using fuzzy sets and fuzzy rules. Neuro-Fuzzy system is an architecture that will build a fuzzy set of rules “if-then” with appropriate membership functions to generate certain input-output pairs. This system can be used in the modeling of nonlinear functions, in the identification of on-line components of a control system and time series forecasting. There are variety of applications of fuzzy logic ranging from consumer products, industrial process control and vision systems to medical instrumentation, information systems, signal processing and decision analysis.[4]

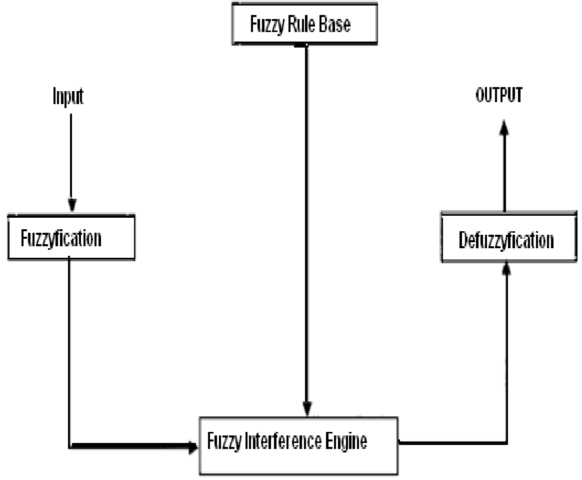

Fig. 2: Schematic representation of fuzzy system

A general fuzzy system, as shown in fig. 2 has basically four components—

- Fuzzy frication,
- fuzzy rule base,
- fuzzy output engine, and
- defuzzification

- **Fuzzification** converts each input data to degrees of membership by a look-up in one or more several membership functions. The key idea in fuzzy logic is theto allow partial belongings of any object to different subsets of the universal set instead of belonging to a single set completely. Partial belonging to a set can be described numerically by a membership function, which takes on values between 0 and 1 inclusive.

In this proposed system simple linear functions, such as triangular ones, are preferable.

- **The fuzzy rule base** contains rules that include all possible fuzzy relations between inputs and outputs. These rules are expressed in the IF–THEN format. In the fuzzy approach, there are no mathematical equations and model parameters. All the uncertainties, nonlinear relationships, or model complications are included in the descriptive fuzzy inference procedure in the form of IF–THEN statements. There are basically two types of rule systems, namely, Mamdani and Sugeno. Depending upon a problem under consideration, we can choose the appropriate rule system. According to the Sugeno rule system, the fuzzy rule is expressed as a mathematical function of the input variable and such a system is more appropriate for neurofuzzy systems. In the Mamdani rule system, the fuzzy rule is also expressed verbally.

In this proposed system IF-THEN format will be used such as,

"IF Water level of reservoir is < 90% of Full capacity THEN Dam gate should remain closed."

Or

"IF Water level of reservoir is > 90% of Full capacity THEN Dam gate should be open gradually."

- **The fuzzy inference engine** takes into account all the fuzzy rules in the fuzzy rule base and learns how to transform a set of inputs to corresponding outputs. To do so, it uses either 'min' or 'prod' activation operators.. The 'prod' activation multiplies or scales the membership curves, thus preserving the initial shape, rather than clipping them as the 'min' activation does.

- **Defuzzification** converts the resulting fuzzy outputs from the fuzzy inference engine to a number. There are many defuzzification methods, such as COG, bisector of area (BOA), mean of maxima (MOM), left-most maximum (LM), and right-most maximum(RM).

4. CONCLUSION AND DISCUSSIONS

In this paper, research on ANN is being done for Hydrological Model to Predict water level at Dam. After successful training of this ANN model, the results have to be applied to the Fuzzy system for further analysis. Fuzzy system will help to open and close Gate of a Dam. Thus Hybrid Hydrological Neuro-Fuzzy model is to be prepared for Dam-Gate operation for flood monitoring and controlling.

ACKNOWLEDGMENT

The authors are grateful to Mr. J. A. Shimpi (Research Officer, Bombay Model) for their kind encouragement and help during the progress of this work.

REFERENCES

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

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

[3] 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

[4] 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

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

[6] Tata McGraw-Hill Publication's "Introduction To Neural Networks Using MATLAB 6.0" by S. N. Sivanandam,S. Sumathi & S.N.Deepa.