Proceedings WARMICE-2012
Sudhir Kumar, et al. (eds)
I.A.H., 2012

IV-8

Application of Soft Computing Techniques in Flood Forecasting

Anil Kumar Lohani
National Institute of Hydrology, Roorkee-247667

Abstract : Flood is one of the most common hydrologic experienced by our country. The flood problem faced by India is unique in several respects due to varied climate and rainfall patterns in different parts of the country. Generally it is found that when part of the country is experiencing floods while another is in the grip of a severe drought. Heavy and intense rainfall is one of the important factors contributing the floods. The floods may also be caused due to many other factors which include failure of the flood control structures, drainage congestions, sudden release of water due to removal of ice jams or land slides in the mountainous streams and coastal flooding due to high tides etc. In spite of various short term and long term measures adopted to prevent and mitigate the consequences of floods, there have been considerable damages and losses due to greater interference by man in natural processes and encroachment of flood plain zones and even riverbeds by human beings. Flood forecasting is used to provide warning to people residing in flood plains and can alleviate a lot of distress and damage. Recently, there has been a growing interest in soft computing techniques viz. Artificial Neural Networks (ANN) and fuzzy logic in hydrologic modelling and forecasting. These models are capable of adopting the non-linear relationship between rainfall and runoff. This paper highlights the applications of soft computing based techniques in flood forecasting. Further, flood forecasting using ANN and Fuzzy inference system based techniques have been demonstrated in Mahanadi river system.

INTRODUCTION

Flood is one of the most common hydrologic extremes which are frequently experienced by our country. The flood problem faced by India is unique in several respects due to varied climate and rainfall patterns in different parts of the country. Generally it is found that when part of the country is experiencing floods while another is in the grip of a severe drought. Excessive runoff resulting due to heavy rain of high intensity results in the flooding of the river flood plains. However, the heavy and intense rainfall is not only factor contributing the floods. The floods may be caused due to many other factors which include failure of the flood control structures, drainage congestions, sudden release of water due to removal of ice jams or land slides in the mountainous streams and coastal flooding due to high tides etc. Inspite of various short term and long term measures adopted to prevent and mitigate the consequences of floods, there has been considerable damages and losses due to greater interference by man in natural processes and encroachment of flood plain zones and even riverbeds by human beings.

During the last decade the artificial neural networks and fuzzy logic techniques have become popular in hydrological modeling, particularly in those applications in which the deterministic approach presents serious drawbacks, due to the noisy or random nature of the data. The research in Artificial Neural Networks (ANNs) started with attempts to model the bio-physiology of the brain, creating models which would be capable of mimicking human thought processes on a computational or even hardware level. Humans are able to do complex tasks like perception, pattern recognition, or reasoning much more efficiently than state-of-the-art computers. They are also able to learn from examples and human neural systems are to some extent fault tolerant.

Recently use of fuzzy set theory has been introduced to inter-relate variables in hydrologic process calculations and modeling the aggregate behavior. Further, the concept of fuzzy decision making and fuzzy mathematical programming have great potential of application in water resources management models to provide meaningful decisions in the face of conflicting objectives. Fuzzy Logic based procedures may be used, when conventional procedures are getting rather complex and expensive or vague and imprecise information flows directly into the modeling process. With Fuzzy Logic it is possible to describe available knowledge directly in linguistic terms and according rules. Quantitative and qualitative features can be combined directly in a fuzzy model. This leads to a modeling process which is often simpler, more easily manageable and closer to the human way of thinking compared with conventional approaches.

The present paper describes the concept of ANN and fuzzy logic. Furthermore, this paper also presents a general review of the applications of ANN and fuzzy logic in hydrological modelling and its popular applications in flood forecasting.

BIOLOGICAL NEURON

It is claimed that the human central nervous system is comprised of about 1,3x1010 neurons and that about 1x1010 of them takes place in the brain. At any time, some of these neurons are firing and the power dissipation due this electrical activity is estimated to be in the order of 10 watts. A neuron has a roughly spherical cell body called soma (Figure 1). The signals generated in soma are transmitted to other neurons through an extension on the cell body called *axon* or *nerve fibres*. Another kind of extensions around the cell body like bushy tree is the *dendrites*, which are responsible from receiving the incoming signals generated by other neurons.

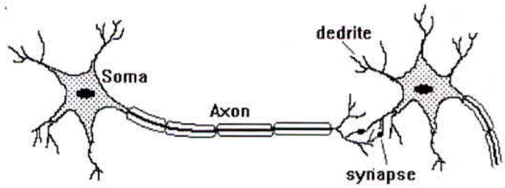

Fig. 1. Typical Neuron

As it is mentioned in the previous section, the transmission of a signal from one neuron to another through synapses is a complex chemical process in which specific transmitter substances are released from the sending side of the junction. The effect is to raise or lower the electrical potential inside the body of the receiving cell. If this graded potential reaches a threshold, the neuron fires. It is this characteristic that the artificial neuron model proposed by McCulloch and Pitts, (McCulloch and Pitts 1943) attempt to reproduce.

Research into *models* of the human brain already started in the 19th century (James, 1890). It took until 1943 before McCulloch and Pitts (1943) formulated the first ideas in a mathematical model called the McCulloch-Pitts neuron. In 1957, a first multilayer neural network model called the perceptron was proposed. However, significant progress in neural network research was only possible after the introduction of the back propagation method (Rumelhart, et al., 1986), which can train multi-layered networks.

ARTIFICIAL NEURON

Mathematical models of biological neurons (called artificial neurons) mimic the functionality of biological neurons at various levels of detail. A typical model is basically a static function with several inputs (representing the dendrites) and one output (the axon). Each input is associated with a weight factor (synaptic strength). The weighted inputs are added up and passed through a nonlinear function, which is called the *activation function* (ASCE, 2000a). The value of this function is the output of the neuron (Figure 2).

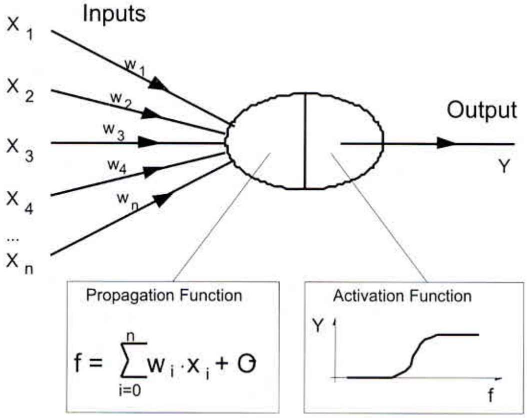

Fig. 2. Processing Element of ANN

NEURAL NETWORK ARCHITECTURE

A typical ANN model consists of number of layers and nodes that are organised to a particular structure. There are various ways to classify a neural network. Neurons are usually arranged in several *layers* and this arrangement is referred to as the *architecture* of a neural net. Networks with several layers are called *multi-layer* networks, as opposed to *single-layer* networks that only have one layer. The classification of neural networks is done by the number of layers, connection between the nodes of the layers, the direction of information flow, the non linear equation used to get the output from the nodes, and the method of determining the weights between the nodes of different layers. Within and among the layers, neurons can be interconnected in two basic ways: (1) *Feedforward networks* in which neurons are arranged in several layers. Information flows only in one direction, from the input layer to the output layer, and (2) *Recurrent networks* in which neurons are arranged in one or more layers and feedback is introduced either internally in the neurons, to other neurons in the same layer or to neurons in preceding layers. The commonly used neural network is three-layered feed forward network due to its general applicability to a variety of different problems and is presented in Figure 3

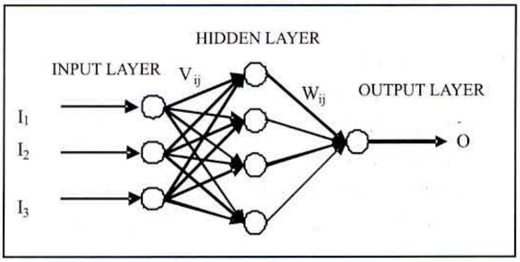

Fig. 3. A Typical Three-Layer Feed Forward ANN (ASCE, 2000a)

LEARING

The learning process in biological neural networks is based on the change of the interconnection strength among neurons. Synaptic connections among neurons that simultaneously exhibit high activity are strengthened. In artificial neural networks, various concepts are used. A mathematical approximation of biological learning, called Hebbian learning is used, for instance, in the Hopfield network. Multi-layer nets, however, typically use some kind of optimization strategy whose aim is to minimize the difference between the desired and actual behavior (output) of the net. Two different learning methods can be recognized: supervised and unsupervised learning:

Supervised learning: the network is supplied with both the input values and the correct output values, and the weight adjustments performed by the network are based upon the error of the computed output.

Unsupervised learning: the network is only provided with the input values, and the weight adjustments are based only on the input values and the current network output. Unsupervised learning methods are quite similar to clustering approaches.

MULTI-LAYER NEURAL NETWORK

A multi-layer neural network (MNN) has one input layer, one output layer and a number of hidden layers between them. In a MNN, two computational phases are distinguished:

1. *Feedforward computation.* From the network inputs $(xi, i = 1, \ldots, n)$, the outputs of the first hidden layer are first computed. Then using these values as inputs to the second hidden layer, the outputs of this layer are computed, etc. Finally, the output of the network is obtained.

2. *Weight adaptation.* The output of the network is compared to the desired output. The difference of these two values called the error, is then used to adjust the weights first in the output layer, then in the layer before, etc., in order to decrease the error. This backward computation is called error backpropagation. The error back propagation algorithm was proposed by and Rumelhart, et al. (1986) and it is briefly presented in the following section.

Feedforward Computation

In a multi layer neural network with one hidden layer, step wise the feed forward computation proceeds as:

I. Forward Pass

Computations at Input Layer

Considering linear activation function, the output of the input layer is input of input layer:

$$O_l = I_l \tag{1}$$

where, is the l^{th} output of the input layer and is the l^{th} input of the input layer.

Computations at Hidden Layer

The input to the hidden neuron is the weighted sum of the outputs of the input neurons:

$$I_{hp} = u_{1p}O_1 + u_{2p}O_2 + \ldots.. + u_{lp}O_l \tag{2}$$

or $p = 1,2,3,\ldots..m$

where, is the input to the p^{th} hidden neuron, is the weight of the arc between l^{th} input neuron to p^{th} hidden neuron and m is the number of nodes in the hidden layer.

Now considering the sigmoidal function the output of the p^{th} hidden neuron is given by:

$$O_{hp} = \frac{1}{(1+e^{-\lambda(I_{hp}-\theta_{hp})})} \tag{3}$$

where O_{hp} is the output of the p^{th} hidden neuron, I_{hp} is the input of the p^{th} hidden neuron, θ_{hp} is the threshold of the p^{th} neuron and λ is known as sigmoidal gain. A non-zero threshold neuron is computationally equivalent to an input that is always held at -1 and the non-zero threshold becomes the connecting weight values.

Computations at Output Layer

The input to the output neurons is the weighted sum of the outputs of the hidden neurons:

$$I_{Oq} = w_{1q}O_{h1} + w_{2q}O_{h2} + \ldots.. + w_{mq}O_{hm} \tag{4}$$

for $q = 1,2,3,\ldots.n$

where, I_{Oq} is the input to the q^{th} output neuron, w_{mq} is the weight of the arc between m^{th} hidden neuron to q^{th} output neuron.

Considering sigmoidal function, the output of the q^{th} output neuron is given by:

$$O_{Oq} = \frac{1}{(1+e^{-\lambda(I_{Oq}-\theta_{Oq})})} \tag{5}$$

where, O_{Oq} is the output of the q^{th} output neuron, λ is known as sigmoidal gain, θ_{Oq} is the threshold of the q^{th} neuron. This threshold may also be tackled again considering extra 0^{th} neuron in the hidden layer with output of -1 and the threshold value becomes the connecting weight value.

Computation of Error

The error in output for the r^{th} output neuron is given by:

$$\xi^{l} = \frac{1}{2}\sum_{r=1}^{n}(T_{Or} - O_{or})^2 \tag{6}$$

where O_{Or} is the computed output from the r^{th} neuron and T_{Or} is the target output.

Equation (4.19) gives the error function in one training pattern. Using the same technique for all the training patterns the error function become

$$\xi = \sum_{j=1}^{N}\xi^{j} \tag{7}$$

where, N is the number of input-output data sets.

Training of Neural Network

Training is the adaptation of weights in a multi-layer network such that the error between the desired output and the network output is minimized.

II. Backword Pass

For k^{th} output neuron, E_k is given by

$$\xi_k = \frac{1}{2}(T_k - O_{ok})^2 \tag{8}$$

where, T_k is the target output of the k^{th} output neuron and O_{ok} is the computed output of the k^{th} output neuron. The output of the k^{th} output neuron is given by

$$O_{Ok} = \frac{1}{(1+e^{-\lambda(I_{Ok}-\theta_{Ok})})} \tag{9}$$

The change of weight for weight adjustment of synapses connecting hidden neurons and output neurons is expressed as:

$$\Delta w_{ik} = -\eta\frac{\partial \xi_k}{\partial w_{ik}} = -\eta \cdot O_{hi} \cdot d_k \tag{10}$$

where, $d_k = \lambda \cdot (T_K - O_{Ok}) \cdot O_k \cdot (1 - O_{Ok})$ and η is learning rate constant

Learning rate coefficient determines the size of the weight adjustment made at each iteration and hence influences the rate of convergence. Poor choice of the learning coefficient can result in a failure in convergence. For a too large learning rate coefficient the search path will oscillate and jump past the minimum. For a very small learning rate coefficient the descent will progress in a small steps and thus significantly increase the time of convergence.

Therefore, change of weight for weight adjustment of synapses connecting input neurons and hidden neurons is expressed as:

$$\Delta u_{ij} = -\eta\frac{\partial \xi_k}{\partial u_{ij}} = -\eta\,[\{-w_{ik}d_k\}\cdot\{\lambda(O_{hi})(1 - O_{hi})\}\cdot\{I_{ij}\}] \tag{11}$$

The performance of the back propagation algorithm depends on the initial setting of the weights, learning rate, output function of the units (sigmoidal, hyperbolic tangent etc.) and the presentation of training data. The initial weights should be randomized and uniformly distributed in a small range of values. Learning rate is important for the speed of convergence. Small values of learning parameter may result in smooth trajectory in the weight space but takes long time to converge. On the other hand large values may increase the learning speed but result in large random fluctuations in the weight space. It is desirable to adjust the weights in such a way that all the units learn nearly at the same rate. The training data should be selected so that it represents all data and the process adequately. The major limitation of the back propagation algorithm is its slow convergence. Moreover, there is no proof of convergence, although it seems to perform well in practice. Sometimes it is possible that result may converge to local minimum and there is no way to reduce its possibility. Another problem is that of scaling, which may be handled using modular architectures and prior information about the problem.

ANN: MODEL DESIGN & TRAINING

Before applying ANN, the input data need to be standardized so as to fall in the range [0,1]. A typical hydrological variable, say river discharge (Q), which can vary between Qmin to some maximum value Q_{max} can be standardized by the following formula:

$$Q_s = \frac{Q - Q_{\min}}{Q_{\max} - Q_{\min}} \qquad (12)$$

where Q_s is the standardized discharge.

For a specific modeling problem, an ANN is designed in such a way to obtain a simple architecture which yields the desired performance. As there is no analytical solution to determine an optimal ANN architecture and therefore, a unique solution cannot be guaranteed. The numbers of input and output nodes are decided from the modeling problem. Further, the number of hidden layers and the number of nodes in each hidden layer are determined to produce most suitable ANN model architecture. Generally, a trial-and-error approach is used to find out the number of hidden layers and the number of nodes in each hidden layer. The number of nodes should be chosen carefully since the performance of a network critically depends on it. A network with too few nodes gives poor results, while it overfits the training data if too many nodes are present. The primary goal of training is to minimize the error function by searching for a set of connection strengths and threshold values that cause the ANN to produce outputs that are equal or close to targets. There are different types of learning algorithms that are quite suitable for specific problems. The supervised training algorithm uses a large number of inputs and outputs patterns. The inputs are cause variables of a system and the outputs are the effect variables. This training procedure involves the iterative adjustment and optimization of connection weights and threshold values for each of nodes. In contrast, an unsupervised training algorithm uses only an input data set. The ANN adapts its connection weights to cluster input patterns into classes with similar properties. Supervised training is most common in water resources applications.

WHAT IS FUZZY LOGIC?

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, Fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy Logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long time which are mapped into exact numeric ranges.

$$\frac{x}{b_{ij}} = \frac{1-C}{1} \Rightarrow b_{ij} = \frac{x}{1-C}$$

Fuzzy Logic has been gaining increasing acceptance during the past few years. There are over two thousand commercially available products using Fuzzy Logic, ranging from washing machines to high speed trains. Nearly every application can potentially realize some of the benefits of Fuzzy Logic, such as performance, simplicity, lower cost, and productivity.

Fuzzy Logic has been found to be very suitable for embedded control applications. Several manufacturers in the automotive industry are using fuzzy technology to improve quality and reduce development time. In aerospace, fuzzy enables very complex real time problems to be tackled using a simple approach. In consumer electronics, fuzzy improves time to market and helps reduce costs. In manufacturing, fuzzy is proven to be invaluable in increasing equipment efficiency and diagnosing malfunctions. Usefulness of fuzzy rule based modeling in hydrological modeling and forecasting is also demonstrated by various researchers.

FUZZY SETS

In ordinary (non fuzzy) set theory, elements either fully belong to a set or are fully excluded from it. Recall, that the membership $\mu_?(x)$of ??of a classical set A, as a subset of the universe x, is defined by:

$$\mu_A(x) = \begin{Bmatrix} 1, & \text{iff } x \in A \\ 0, & \text{iff } x \notin A \end{Bmatrix}$$

This means that an element ??is either a member of set A($\mu_?(x)=1$) or not ($\mu_?(x)=0$). This strict classification is useful in the mathematics and other sciences. Figure 4 presents difference between boolean logic and fuzzy logic.

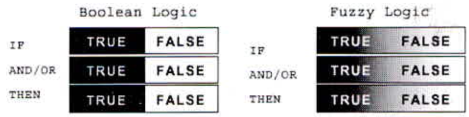

Fig. 4. Difference beetween boolean logic and fuzzy logic

MEMBERSHIP FUNCTION ASSIGNMENT AND RULE GENERATION

First, partition the input and output space as small, medium, large etc. After partition, the next step is to assign a membership function. First the data points whose membership grades are among the highest are chosen. The mid-point of these data points is assigned grade of one, which is the index of membership function. Then a membership grade C (0<C<1) is assigned.

The membership function is shown in the Figure 5, where c_{li} and b_{li} are the center and the half-width of the membership function respectively. And x is the average distance of the vertex to the left and the right edges. Thus, we have:

$$\frac{x}{b_{ij}} = \frac{1-C}{1} \Rightarrow b_{ij} = \frac{x}{1-C} \tag{13}$$

C is a parameter to be assigned. This C is usually determined by experience, although some optimization techniques may be used. Typical values of CM vary from 0.5 to 0.8. After partitioning the input and output spaces and assigning the membership functions, the next step is to construct the rules.

STEPS FOR DEVELOPING FUZZY LOGIC MODEL

Step by step procedure for developing a fuzzy model is given below:

- Define the model objectives and criteria: What am I trying to model? What do I have to do to model the system? What kind of

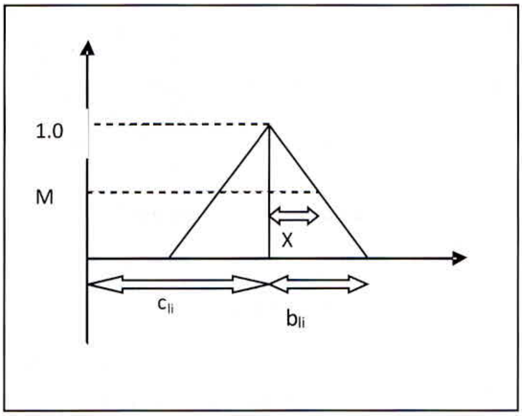

Fig. 5. The Triangular Membership Function

response do I need? What are the possible (probable) system failure modes?

- Determine the input and output relationships and choose a minimum number of variables for input to the Fuzzy Logic (FL) system.
- Using the rule-based structure of FL, break the modelling problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number of fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.
- Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.
- Create the necessary pre- and post-processing FL
- Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained. Figure 6 presents steps involved for developing of fuzzy model.

RULE-BASED FUZZY SYSTEMS

In rule-based fuzzy systems, the relationships between variables are represented by means of fuzzy if–then rules in the following general form:

If antecedent proposition then consequent proposition.

Fuzzy propositions are statements like "x is big", where "big" is a *linguistic label*, defined by a fuzzy set on the universe of discourse of variable x. Linguistic labels are also referred to as fuzzy

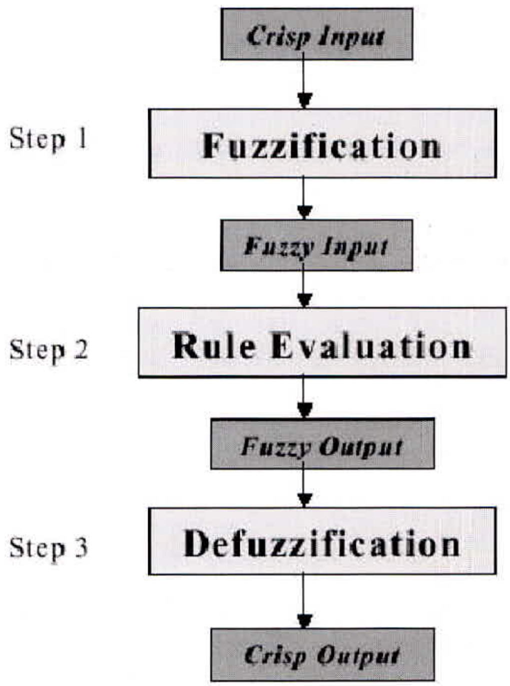

Fig. 6. Steps for developing fuzzy model

constants, fuzzy terms or fuzzy notions. Linguistic modifiers (hedges) can be used to modify the meaning of linguistic labels. For example, the linguistic modifier *very* can be used to change "*x* is big" to "*x* is *very* big".

The antecedent proposition is always a fuzzy proposition of the type "x is *A*" where x is a linguistic variable and *A* is a linguistic constant (term). On the basis of structure of the consequent proposition, different fuzzy rule based models are defined. In a *Linguistic fuzzy model* (Zadeh, 1973; Mamdani, 1977) both the antecedent and consequent are fuzzy propositions. *Singleton* fuzzy model is a special case where the consequents are singleton sets (real constants).

GENERAL LINGUISTIC FUZZY MODEL

The general Linguistic Fuzzy Model of a Multi-Input Single-Output system is interpreted by rules with multi-antecedent and single-consequent variables such as the following:

Rule l: IF I_1 is B_{l1} AND I_2 is B_{l2} AND I_r is B_{lr}

THEN O is D_l, $l = 1,2,..,n$ (14)

Where I_1, I_2,..., I_r are input variables and O is the output, B_{ij} (i=1, ...,n, j=1,...,r) and D_i (i=1, ...,n) are fuzzy sets of the universes of discourse X_1, X_2,., X_r, and Y of I_1, I_2 ,..., I_r and O respectively. The above rule can be interpreted as a fuzzy implication relation

$$B_l = B_{l1} \times B_{l2} x.x B_{lr} ® D_l \text{ in } (X = X_1 \times X_2 x.x X_r) x Y: \quad R_l(x,y) = T(B_l(x), D_l(y)), B_l(x) = T2 (B_{l1}(x), B_{l2}(x), \ldots B_{lr}(x)) \quad (15)$$

Where T and T' are the t-norm operators and may be different from each other. Let the fuzzy set A in the universe of discourse X be the input to the fuzzy system of (14). Then, each fuzzy IF-THEN rule determines a fuzzy set F_l in Y:

$$F_l(y) = T(R_l(x,y), A(x)) \quad (16)$$

For a crisp input $x^* = (x_1^*, x_2^*, \ldots . x_r^*)$

$$A_i(x) = \begin{cases} 1, & \text{if } x_i = x_i^* \\ 0, & \text{if } x_i \neq x_i^* \end{cases} \quad (17)$$

Then

$$\begin{aligned} F_l(y) &= T(R_l(x,y), A(x)) \\ &= T(B_{il}(x), A(x), D_i(y)) \quad (18) \\ &= T(B_{il}(x^*), D_i(y)) \end{aligned}$$

where $Bl(x)$ is called the Degree Of Firing (DOF) of rule l:

$$?_l(x^*) = T'(B_{li}(x^*), B_{l2}(x_2^*), \ldots . B_{lr}(x_r^*))? \quad (19)$$

The output fuzzy set F of the fuzzy system is the t-conorm of the n fuzzy sets F_l ($l=1,2,\ldots n$):

$$F(y) = S[F_1(y), F_2(y), \ldots F_n(y)]? \quad (20)$$

Where, S denotes the t-conorm operator. To obtain a crisp value of the output, the commonly used Center of Area (COA) method, may be used.

$$y^* = \frac{\int_{y_0}^{y_l} yF(y)dy}{\int_{y_0}^{y_l} F(y)dy} \quad (21)$$

Where, the real interval $Y = [y_0, y_l]$? is the universe of discourse for the output.

The fuzzy system is usually not analytical, but analytical formulation is essential for the use of training algorithms like Back Propagation (BP) and Least Mean Squared (LMS). We, therefore, use the following simplified fuzzy inference system: First, T-norm and T-conorm operators are chosen to be the multiplication and addition operators, respectively. Then equation (20) becomes,

$$F(y) = \sum_{l=1}^{n} F_i(y) = \sum_{i=1}^{n} B_l(x^*) \cdot D_l(y) \quad (22)$$

Obviously, the summation brings the output fuzzy set $F(y)$ out of the unit interval. However, it does not have an effect on the defuzzified value. By substituting for $F(y)$ in (21) we get the COA defuzzified value:

$$y^* = \frac{\int_{y_0}^{y_1} y \sum_{l=1}^{n} B_l(x^*) D_l(y) dy}{\int_{y_0}^{y_1} \sum_{l=1}^{n} B_l(x^*) D_l(y) dy}$$

$$= \frac{\sum_{l=1}^{n} B_l(x^*) \left\{ \frac{\int_{y_0}^{y_1} y D_l(y) dy}{\int_{y_0}^{y_1} D_l(y) dy} \right\}}{\sum_{l=1}^{n} B_l(x^*)}$$

$$= \frac{\sum_{l=1}^{n} B_l(x^*) y_l^*}{\sum_{l=1}^{n} B_l(x^*)} \qquad (23)$$

Where the y_l^*'s are the centroids of the fuzzy sets D_l.

$$y^* = f(x) = \frac{\sum_{l=1}^{n} y_l^* (\prod_{l=}^{r} 1 - \frac{|x_i - c_{li}|}{b_{li}})}{\sum_{l=1}^{n} (\prod_{l=}^{r} 1 - \frac{|x_i - c_{li}|}{b_{li}})}, c_{li} - b_{li} \le x_i \le c_{li} + b_{li}$$

The defuzzified value y^* is determined by the weighted average of the centroids of the individual consequent fuzzy sets. Using a symmetric triangular membership function, the fuzzy system becomes,

$$y^* = f(x) = \frac{\sum_{l=1}^{n} y_l^* (\prod_{l=}^{r} 1 - \frac{|x_i - c_{li}|}{b_{li}})}{\sum_{l=1}^{n} (\prod_{l=}^{r} 1 - \frac{|x_i - c_{li}|}{b_{li}})}, c_{li} - b_{li} \le x_i \le c_{li} + b_{li} \qquad (24)$$

Where, c_{li} and b_{li} are the center and the half-width of the triangular membership function respectively.

FLOOD FORECASTING

There are many ANN architectures and algorithms developed for different applications. Out of them most common are Multi layer feed forward, Hoppfield networks, Radial basis function network, Recurrent network, Self organization feature maps, Counter propagation networks. Selection of a particular network is application oriented. However, the multi layer feed forward networks are most commonly used for hydrological applications (Dawson and Wilby, 2001).Various researchers have investigated the use of multi-layer perceptron NN for rainfall- runoff modeling, flood forecasting successfully. Earlier the works of Bruen and Yang (2005), Campolo *et al.* (1999, 2003), Coulibaly *et al.* (2000), Lekkas *et al.* (2004), Minns and Hall (1996), Solomantine and Xue (2004), Solomantine and Price (2004), Zealand *et al.* (1999) emphasized the application of artificial neural networks over other methods.

In one study ANN is applied in forecasting the flood of downstream catchment of Mahanadi basin. Here peak values recorded over 12 years (1997-2007) are considered for development of the model. The Fig.7 shows Khairmal as base station (BS), Barmul as intermediate station (IS) and Mundali as forecast station (FS). The peak values are put into ANN architecture using MATLAB codes. The trial has been taken with a 3-layer feed forward network. Different combinations of feed forward network with changing transfer function, number of neurons and epochs varying at an increment of 50 are trailed. The combinations which are mostly as per performance criteria fixed are noted (Table1). The Cascade feed forward network has been most successful for Khairmal-Barmul and Khairmal-Barmul-Mundali and other two cases are with Feed forward network. The results are compared with statistical method and ANN shows development over conventional method (Fig.8).

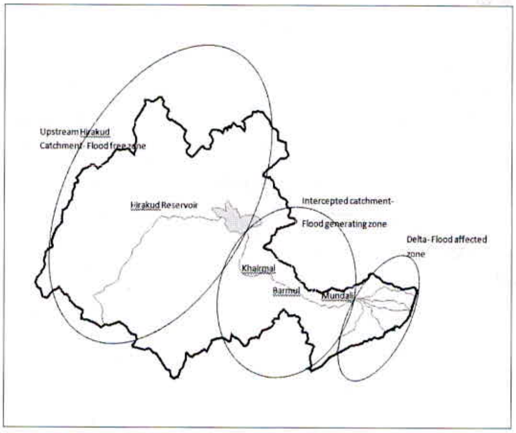

Fig. 7. Showing catchment details with different zones of Mahanadi basin.

Table 1. Relationship between discharges using ANN architecture

Between stations	ANN architecture	RMSE(m^3/s)		R^2		Efficiency	
		Training	Testing	Training	Testing	Training	Testing
(1)	(2)	(4)	(5)	(6)	(7)	(8)	(9)
Khairmal-Barmul	CFF 1,7,1	2249.1	1132.2	0.9720	0.9891	0.9446	0.9752
Barmul-Mundali	FF 1,7,1	1681.5	1251.9	0.9852	0.9962	0.9705	0.9745
Khairmal-Mundali	FF 1,5,1	2441.5	1741.7	0.9684	0.9789	0.9375	0.9506
Khairmal-Barmul-Mundali	CFF 2,15,1	1697	1324.1	0.9850	0.9922	0.9700	0.9714

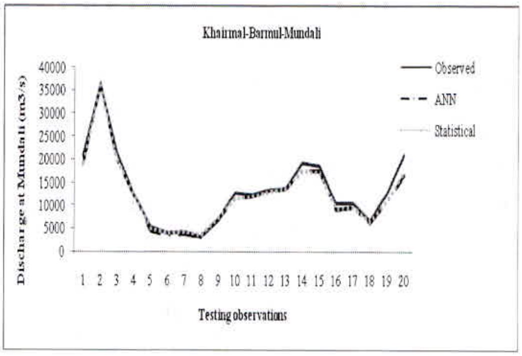

Fig. 8. Comparison between observed and computed discharge (Khairmal-Barmul-Mundali)

The classical fuzzy set theory introduced by Zadeh in 1965. Besides ANN in the field of hydrology fuzzy logic is being used invariably in the field of clustering, rainfall-runoff modeling, flood forecasting and other related fields. In the applications of the fuzzy system in control and forecasting, there are mainly two approaches, the first one being the Mamdani approach and the other the Takagi–Sugeno approach (Kruse et al., 1994). The T.S model has been applied successfully by Lohani et.al (2005a, 2005b, 2006, 2007, 2010), Nayak et.al (2005). For the Mamdani approach (Mamdani and Assilian, 1975), which has been used in some hydrological applications (Schulz and Huwe (1997) and Schulz et al. (1999). Tareghian and Kashefipour (2007) have applied ANN and fuzzy logic models for prediction of daily reservoir inflow in Dez reservoir of south-west Iran. They had found the superiority of Mamdani model over Takegi-Sugeno fuzzy model.

Discharge data for a period of 10 years from 2001-2010 in 3-hour basis at Khairmal, Barmul and Mundali have been used. A total 101 flood peaks of different magnitudes are being observed at 3 stations along with there corresponding flood peaks. Initial 50 peaks are considered for calibration of the fuzzy Mamdani model and rest 51 for validation.

Different membership functions are trialed and finally gauss2 membership function has been selected (Fig.9). To describe the relation between the magnitudes of the peaks there are 9 rules formed. Basing on the inputs, membership functions and rules associated a fuzzy output is computed. The input and fuzzy output is defuzzified to crisp output by using centroid method. The conceptual crisp outputs provide the flood forecast at forecasting site. The relation between the inputs and output is also represented by the 3-dimensional plot named as surface view (Fig.10). It shows the variation of output with respect to inputs. The defuzzified results are compared with observed values and seen that the

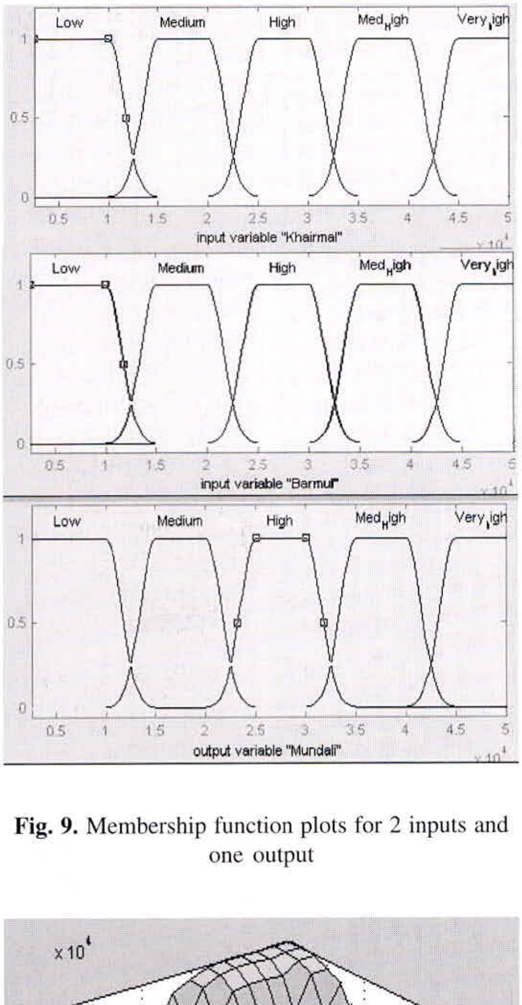

Fig. 9. Membership function plots for 2 inputs and one output

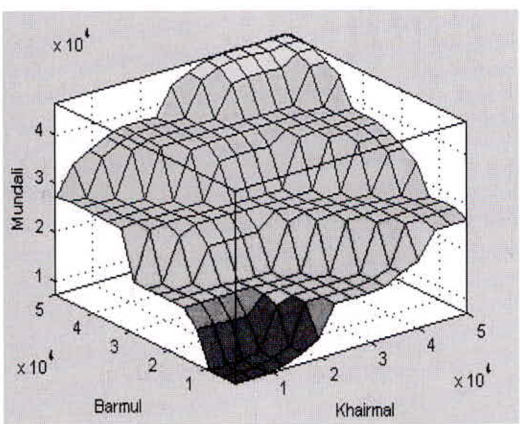

Fig. 10. Showing the surface view of inputs and output of fuzzy model

higher peaks are modeled more prominently than low or medium peaks (Fig.11).

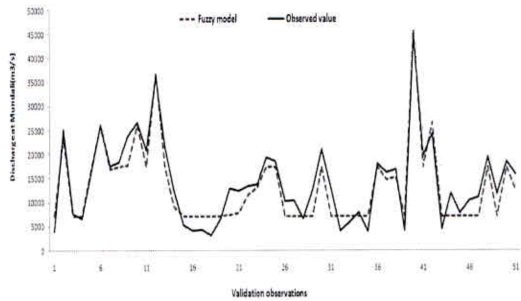

Fig.11. Comparison of results at Mundali

REMARKS

The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. Furthermore there is no need to devise an algorithm in order to perform a specific task; i.e. there is no need to understand the internal mechanisms of that task. They are also very well suited for real time systems because of their fast response and computational times which are due to their parallel architecture. Neural networks also contribute to hydrological modeling and forecasting. They are successfully used to model various hydrological processes. Even though neural networks have a huge potential one will only get the best of them when they are integrated with computing, AI, fuzzy logic and related subjects.

REFERENCES

ASCE Task Committee on Application of Artificial. Neural Networks in Hydrology (2000a). Artificial Neural Networks in hydrology, I: Preliminary Concepts. Journal of Hydrologic Engineering, ASCE, Vol. 5(2), 115-123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000b). Artificial Neurill Networks in hydrology, 11: Hydrological Applications. Journal of Hydrologic Engineering, ASCE, Vol. 5(2), 124-137.

Bruen,M and Yang, J. (2005) "Functional Networks in Real- Time Flood Forecasting - A Novel Application," Journal of Advances in Water Resources, vol. 28, no. 9, pp. 899-909.

Campolo, M., Andreeussi,P. and Soldati, A. (1999) "River Flood Forecasting with a Neural Network Model," Water Resources Research, Vol. 35(4), pp. 1191-1197.

Campolo, M., Soldati, A. and Andreeussi, P. (2003) "Artificial neural network approach to flood forecasting in the River Arno". Hydrological Sciences Journal, Vol.48(4), pp.381-398.

Coulibaly, P., Anctil,F. and Bobee, B. (2000) "Daily reservoir forecasting using artificial neural network with stopped training approach", Journal of Hydrology, Vol.230(3-4), pp.244-257.

Dawson, C.W. and Wilby, R.L. (2001)"Hydrological modeling using ANN", Progress in physical geography, Vol.25(1), pp.80-108.

James, W. (1890). Psychology (Briefer Course). New York: Holt.

Lekkas, D.F., Onof, C., Lee,M.J. and Baltas, E.A. (2004) "Application of neural networks for flood forecasting", Global Nest: the Int. Journal. Vol.6(3), pp. 205-211.

Lohani,A.K., Goel, N.K. and Bhatia, K.K.S. (2006) Takagi-Sugeno fuzzy inference system for modeling stage-discharge relationship, Journal of Hydrology, Vol.331, 146–160.

Lohani,A.K., Goel, N.K. and Bhatia, K.K.S. (2007a) Deriving stage—discharge—sediment concentration relationships using fuzzy logic, Journal of Hydrological Sciences, Vol.52(4), 793–807.

Lohani,A.K., Goel, N.K. and Bhatia, K.K.S. (2007) Deriving stage-discharge-sediment concentration relationships using fuzzy logic, Journal of Hydrological Sciences, Vol.52(4), 793–807.

Lohani,A.K., Goel, N.K. and Bhatia, K.K.S. (2010) Comparative study of neural network, fuzzy logic and linear transfer function techniques in daily rainfall-runoff modeling under different input domains, Hydrological processes, Doi:10.1002/hyp.7831.

Lohani, A.K., Goel, N.K. and Bhatia, K.K.S. (2005a) Real time flood forecasting using fuzzy logic, Hydrological Perspectives for Sustainable Development, vol. 1, Perumal M (ed). Allied Publishers Pvt. Ltd, New Delhi, 168–176.

Lohani, A.K., Goel, N.K. and Bhatia, K.K.S. (2005b) Development of fuzzy logic based real time flood forecasting system for river Narmada in Central India, In International Conference on Innovation Advances and Implementation of Flood Forecasting Technology, ACTIF/ Floodman / Flood Relief, October, 2005. Tromso, Norway, www. Actif. cc.net / conference 2005 / proceedings.

Mamdani, E.H., Assilian, S. (1975) An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man–Machine Studies, Vol.7(1), 1–13.

Mamdani, E.H. (1977) Application of fuzzy logic to approximate reasoning using linguistic systems. Fuzzy Sets and Systems, Vol.26, 1182-1191.

Nayak,P.C., Sudheer,K.P., Rangan,D.M. and Ramashastri, K.S. (2005) Short term flood forecasting with a neuro-fuzzy model, Water Resources Research, Vol.41, w04004, doi: 10, 1029/2004WR003562.

Panigrahi, D.P. and Mujumdar, P.P. (2000) Reservoir operation modelling with fuzzy logic, Journal of Water Resources Management, Vol.14(2), 89-109.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115 – 133.

Minns,A.W. and Hall, M.J. (1996) "Artificial neural networks as rainfall-runoff models", Hydrological Sciences Journal, Vol. 41 (3), pp. 399-417

Rumelhart,D.E., McLelland,J.L. and the PDP Research Group, (1986). Parallel Distributed Processing, Explorations in the Micro Structure of Cognition, Vo1. I: Foundations, The MIT Press, Cambridge.

Schulz, K. and Huwe, B. (1997) Water flow modeling in the unsaturated zone with imprecise parameters using a fuzzy approach, Journal of Hydrology, 201, 211–229.Schulz, K., Huwe, B. and Peiffer, S. (1999) Parameter uncertainty in chemical equilibrium calculations using fuzzy set theory. Journal of Hydrology, 217, 119–134.

Zadeh, L.A. (1965) Fuzzy Sets. Information and control, 8, 338-353.

Zadeh, L.A. (1973) Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man and Cybernetics, 1, 28-44.

Zealand, C.M., Burn, D.H. and Simonovic, S.P. (1999) "Short term streamflow forecasting using artificial neural network", Journal of Hydrology, vol.21, no.4, pp.32-48.