Training Course

On

Predictions in Ungauged Basins

[July 26 – 30, 2010]

CHAPTER-13

APPLICATION OF SOFT-COMPUTING IN RAINFALL-RUNOFF MODELING

By

A.K. Lohani
Scientist E1

Organized under Hydrology Project - II

National Institute of Hydrology
Roorkee – 247667 (Uttrakhand)

CHAPTER-13

APPLICATION OF SOFT-COMPUTING IN RAINFALL-RUNOFF MODELING

A.K. Lohani
Scientist E1
National Institute of Hydrology, Roorkee-276341

13.0 INTRODUCTION

During the last decade soft computing techniques mainly artificial neural networks and fuzzy logic techniques have become popular in hydrological modeling, particularly in 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 biophysiology 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 interrelate variables in hydrologic process calculations and modellling 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 lecture describes the concept of ANN and Fuzzy logic. Furthermore, this lecture also presents a general review of the applications of ANN and fuzzy logic in rainfall runoff modelling.

13.1 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.

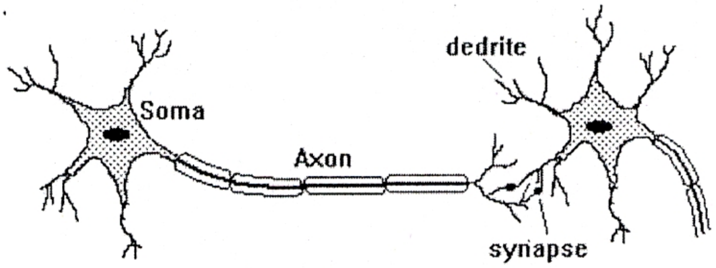

Figure 13.1 Typical Neuron

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 backpropagation method (Rumelhart, et al., 1986), which can train multi-layered networks.

13.2 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; APPENDIX-I). The value of this function is the output of the neuron (Figure 2).

Figure 13.2 Processing Element of ANN

13.2.1 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

Figure 13.3 A Typical Three-Layer Feed Forward ANN (ASCE, 2000a)

13.2.2 Learning

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.

13.2.3 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, . . . , 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 backpropagation

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, O_l is the l^{th} output of the input layer and I_l 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:

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

for $p = 1,2,3,.....m$

where, I_{hp} is the input to the p^{th} hidden neuron, u_{lp} 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:

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

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

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 θ_{Oq} 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 backpropagation 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 backpropagation 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.

13.3 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.

13.4 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.

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.

13.5 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.**

Figure 13.4 Boolean Logic Vs Fuzzy Logic.

13.5.1 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:

$$\frac{x}{b_{ij}} = \frac{1-C}{1} \Rightarrow b_{ij} = \frac{x}{1-C} \qquad \ldots(13)$$

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.

13.5.2 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 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.

Figure 13.6 Steps for developing fuzzy model

13.6 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 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).

13.6.1 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

$$B_l = B_{l1} \text{ x } B_{l2} \text{ x.x } B_{lr} \rightarrow D_l \text{ in } (X = X_1 \text{ x } X_2 \text{ x.x } X_r)\text{x}Y:$$
$$R_l(x,y) = T(B_l(x),D_l(y)),B_l(x)=T'(B_{l1}(x),B_{l2}(x),\ldots.B_{lr}(x)) \quad \ldots(15)$$

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:

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

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

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

Then

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

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

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

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

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

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.

$$y^* = \frac{\int_{y_0}^{y_i} yF(y)dy}{\int_{y_0}^{y_i} F(y)dy} \qquad \text{...(21)}$$

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,

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

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:

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

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

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

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

$$y^{*}=f(x)=\frac{\sum_{l=1}^{n} y_{l}^{*}(\prod_{l=}^{r} 1-\frac{|x_{l}-c_{li}|}{b_{li}})}{\sum_{l=1}^{n}(\prod_{l=}^{r} 1-\frac{|x_{l}-c_{li}|}{b_{li}})}, c_{li}-b_{li}\leq x_{i}\leq c_{li}+b_{li} \qquad \ldots(24)$$

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

13.7 RAINFALL-RUNOFF MODELLING

The problem of transformation of rainfall into runoff has been subject of scientific investigations throughout the evolution of the subject of hydrology. Hydrologists are mainly concerned with evaluation of catchment response for planning, development and operation of various water resources schemes. A number of investigators have tried to relate runoff with the different characteristics which affect it. For the purpose of rainfall-runoff process simulation and design flood evaluation, conceptual and physical based models are widely used. The model calibration and validation are the important aspects of the hydrological modelling proper calibration and validation of the hydrological model is necessary before using the model for simulation. In order to ascertain the uncertainity in the parameters as well as parameter stability the sensitivity analysis must be carried out.

13.8 GENERAL DATA REQUIREMENT FOR RAINFALL-RUNOFF MODELLING

Before undertaking rainfall-runoff modelling for a particular storm, it is advisable to assess the quantity and quality of available data. Quite often, the available data dictate the type of model to be used more than the problem itself. A general inventory of data frequently available or needed is given in what follows.

Watershed Characteristics

The most commonly available is the topographic map from which many useful geomorphic parameters can be extracted, that is, watershed area, subbasin areas, elevations, slopes, channel lengths, channel profiles, centroid, etc. Many other geomorphic parameters can then be computed. Another useful map is the landuse map, which provides data on areas of land-use practice, soil types, vegetation, forest areas, “lakes, urban development, etc.

Rainfall Characteristics

Determination of the average amount of rain that falls on a basin/subbasins during a given storm is a fundamental requirement for many rainfall-runoff models. A number of techniques for estimating mean areal rainfall have been developed. Rainfall hyetographs are needed for each subbasin. Some of the subbasins may not have a recording raingauge and may involve extrapolation of rainfall data from neighbouring subbasins. If a subbasin has more than one raingauge, then the mean areal rainfall hyetograph is to be determined. Sometimes, only standard/storage-type raingauges are available in some watersheds. The rainfall amounts then need to be properly distributed in time so that rainfall hyetographs can be prescribed.

Infiltration and other Loss Characteristics

In a majority of cases, no data are available on soil infiltration, interception, depression storage, and antecedent soil moisture. If data do exist in part or full, maximum advantage must be taken to estimate infiltration and other loss functions. If no information is available on antecedent soil moisture, then an antecedent precipitation index can be used to get an estimate of the antecedent soil moisture. Soil type and landuse vegetation complex can be used to estimate infiltration parameters.

Stream flow Characteristics

Streamflow may be available in terms of the stage at the watershed outlet and at some other gauges within the watershed. Appropriate rating curves can be used to convert stages into discharges. Part of the streamflow data may be used for model calibration and the remaining data for model verification.

13.9 HYDROLOGICAL PROCESSES IN RAINFALL-RUNOFF MODELS

Various stream flow simulation models generally consider the following hydrological processes to simulate the time series of stream flow.

(a) Land Surface Processes
(i) Interception
(ii) Infiltration
(iii) Overland flow
(iv) Evapotranspiration
(v) Snow accumulation and Melt

(b) Sub-surface Processes
(i) Interflow
(ii) Soil moisture storage and Movement
(iii) Ground water storage and flow

(c) Channel Processes
(i) Channel flow
(ii) Flood plain storage
(iii) Lakes, Reservoirs and Diversions

Modelling Procedures

The following procedures are usually followed for Rainfall-Runoff Modelling:

- Develop a suitable model structure to simulate various component processes keeping in mind the quantity and quality of the data available and nature of the problems for which the modelling is required.
- Calibrate the developed model using the historical records.
- Validate the model using the historical records which have not been considered for calibration.
- Perform sensitivity analysis study to identify the most sensitive parameters of the model which require proper investigation before arriving at the final parameter values.
- Use the calibrated and validated model for solving the specific hydrological problem for which the development of the model is intended for.

13.10 SOFT COMPUTING FOR RAINFALL-RUNOFF DYNAMICS

Various researchers have investigated the use of multi-layer perceptron NN for rainfall- runoff modelling successfully. ANNs are also studied for the rainfall-runoff modelling in snow covered catchment on the basis of measurement data. Regressive neural networks is also investigated for modeling and forecasting the rainfall runoff relations. Intensive research efforts have been made to identify optimal network structures, find appropriate training algorithms, and select proper training patterns for improving runoff prediction accuracy (Solomatine *et al.*, 2003). The application of fuzzy logic as a modelling tool in the field of water resources is a relatively new concept, although some studies have been carried out to some extent in the last decade and these studies have generated lots of enthusiasm. Bardossy and Duckstein (1992) applied fuzzy rule based modeling approach to a Karstic aquifer management problem. Bardossy and Disse (1993) used fuzzy rules for simulating infiltration. Panigrahi and Mujumdar (2000) applied fuzzy logic for reservoir operation and management problems. The fuzzy modeling approach has also been successfully applied for water quality management. Use of multiobjective fuzzy linear programming for sustainable irrigation planning and optimal land-water-crop planning has been demonstrated by Sahoo et al. (2006) respectively. Few attempts have been made to demonstrate the applicability of fuzzy rule based approach in river flow forecasting (Chang and Chen, 2001; Lohani et al., 2005a; Lohani et al., 2005b), modeling stage discharge relationship (Lohani et al., 2006, Lohani et al., 2007a), modeling stage discharge sediment relationship (Lohani et al., 2007b) and estiming monthly ground water recharge.

Selection of the input and output variables is the first step in development of a ANN or fuzzy rule based rainfall-runoff model. Runoff at the outlet of a catchment is a function of previous rainfall and runoff values, as well as of the meteorological, topological, and soil and vegetative conditions of the catchment. Theoretically, a non linear and time varying storage function may be useful to express the rainfall-runoff process. There are inherent difficulties in defining such functions particularly when sufficient data are not available and estimation of catchment response is only relying on available rainfall data. Therefore, in the case of a rainfall-runoff model with minimum available data, the output variable describes the runoff that is to be predicted and possible input variables are measured rainfall and runoff data.

13.11 REMARKS

The computing world has a lot to gain from soft computing techniques. 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. Soft computing techniques are also very well suited for real time systems because of their fast response and computational times. These techniques also contribute to hydrological modeling and forecasting. They are successfully used to model various hydrological processes. Even though each soft computing technique has a huge potential, however, one may get the best of them when they are integrated together.

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, 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, 5(2), 124-137.

Bardossy A. and Duckstein, L. (1992) Analysis of a Karstic aquifer management problem by fuzzy composite programming. Water Resources Bulletin 28(1), 63-73.

Bardossy, A. and Disse, M. (1993) Fuzzy rule based models for infiltration. Water Resources Research 29(2), 373-382.

Chang, F.-J. and Chen, Y.-C. (2001) A counterpropagation fuzzy-neural network modelling approach to real time streamflow prediction. Journal of Hydrology, 245, 153-164.

Garcia, L. A.and Shigidi, A. (2006) Using neural networks for parameter estimation in ground water. Journal of Hydrology, 318(1-4), 215-231.

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

Lohani, A.K., Goel, N.K. and Bhatia, K.K.S. (2005a) Real time flood forecasting using fuzzy logic. Hydrological Perspectives for Sustainable Development, Volume I, Eds. M. Perumal, 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/conference2005/proceedings.

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, 331, 146-160.

Lohani, A.K., Goel, N.K. and Bhatia K.K.S. (2007a) Reply to comments provided by Z. Sen on ''Takagi–Sugeno fuzzy system for modeling stage-discharge relationship'' by A.K. Lohani, N.K. Goel and K.K.S. Bhatia. Journal of Hydrology, *337*(1-2), 244-247.

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

Maier, H. R. & Dandy, G. C. (2000) Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. Environ. Modelling & Software 15, 101.124.

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

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

Nayak, P.C., Satyaji Rao, Y. R. and Sudheer, K. P. (2006) Groundwater level forecasting in a shallow aquifer using artificial neural network approach. Water resources management, 20(1), 77-90.

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

Raman, H., and Chandramauli, V. (1996). Deriving a general operating policy for reservoir using Neural Network. Journal of Water Resources Planning and Management, 122(5), 342-347.

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.

Sahoo, B. Lohani, A. K. and Sahu, R.K. (2006) Fuzzy multiobjective and linear programming based management models for optimal land-water-crop system planning. Journal of Water Resources Management, Springer, 20, 931-948.

Solomatine, D. P. & Dulal, K. N. (2003) Model trees as an alternative to neural networks in rainfall–runoff modelling. *Hydrol. Sci.* J. 48(3), 399–411.

Sudheer, K.P. & Jain, S.K. (2003). Radial Basis Function Neural Networks for Modeling Rating Curves, *J. Hydrologic Eng.,*ASCE, 8(3), 161–164.

Sugawara, M. (1979). Automatic calibration of tank model. Hydrological Science Bulletin, 24(3).

Supharatid, S. (2003). Application of a neural network model in establishing a stage-discharge relationship for a tidal river, *Hydrological Processes*, 17, 3085-3099.

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.