3

ANN Applications in Hydrology - Merits and Demerits

D.Nagesh Kumar

ABSTRACT

Artificial Neural Networks (ANNs) have gained popularity among hydrologists, as is evidenced by the increasing number of papers on this topic appearing in hydrology journals, especially in the recent past. The practicing hydrologic community is becoming aware of the potential of ANNs as an alternative modelling tool. In terms of hydrologic applications, this modelling tool is still in its nascent stages. In this paper applications of ANN in various fields of hydrology are discussed. In addition merits and demerits of ANN are discussed.

3.1 INTRODUCTION

Artificial neural network (ANN) is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experimental knowledge and making it available for use (Haykin, 1999). It resembles the brain in two respects: (*i*).Knowledge is acquired by the network from its environment through a learning process and (*ii*). Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge. ANN development is based on the following rules (ASCE,2000 a):

- Information processing occurs at many single elements called nodes, also referred to as units, cells, or neurons.
- Signals are passed between nodes through connection links.
- Each connection link has an associated weight that represents its connection strength.
- Each node typically applies a nonlinear transformation called an activation function to its net input to determine its output signal.

A typical ANN consists of a number of nodes that are organized according to a particular arrangement. One way of classifying neural networks is by the number of layers: single (Hopfield nets); bilayer (Carpenter/Grossberg adaptive resônance networks); and multilayer (most backpropagation networks). ANNs can also be categorized based on the direction of information flow and processing. In a feed forward network, the nodes are generally arranged in layers, starting from a first input layer and ending at the final output layer. There can be several hidden layers, with each layer having one or more nodes. Information passes from the input to the output side. The nodes in one layer are connected to those in the next, but not to those in the same layer. Thus, the output of a node in a layer is only dependent on the inputs it receives from previous layers and the corresponding weights. On the other hand, in a recurrent ANN, information flows through the nodes in both directions, from the input to the output side and vice versa. A typical three-layer feed forward network with one hidden layer is shown in Figure 1. It has two input nodes, three output nodes and three nodes in the hidden layer.

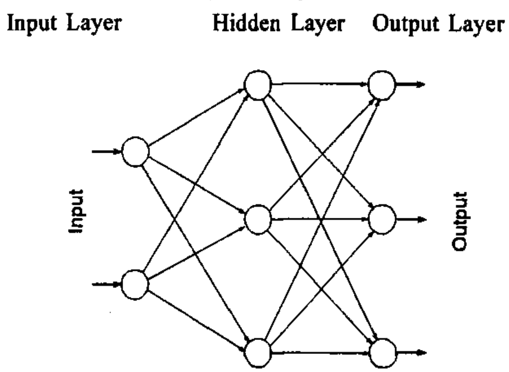

Figure. 1 Typical feed forward neural network

3.1.1 ANN Modelling

There are no fixed rules for developing an ANN, even though a general framework can be followed based on heuristics and experience. The goal of an ANN is to generalize a relationship of the form

$$Y = f(X)$$

where X is an n-dimensional input vector consisting of variables $x_1, \ldots, x_i, \ldots, x_n$; and Y is an m-dimensional output vector consisting of resulting variables of interest $y_1, \ldots, y_i, \ldots, y_m$. In hydrology, the values of x_i can be causal variables (ASCE, 2000a) such as rainfall, temperature, previous flows, water levels, spatial locations, evaporation, basin area, elevation, slopes, pump operating status, contaminant loads, meteorological data, and so on. The values of y_i can be hydrological responses such as runoff, stream flow, ordinates of a hydrograph, optimal pumping patterns, rain fields, hydraulic conductivities, contaminant concentrations, and others. A firm understanding of the hydrologic system under consideration is an important prerequisite for successful application of ANNs. For instance, physical insight into the problem being studied can lead to better choice of input variables for proper mapping. This will help in avoiding loss of information that may result if key input variables are omitted, and also prevent inclusion of spurious inputs that tend to confuse the training process. A sensitivity analysis can be used to determine the relative importance of a variable (Maier and Dandy, 1996) when sufficient data is available.

Determination of the optimal ANN architecture is an important step in ANN modelling. There is no specific approach for determination of such an optimal ANN architecture. Often, more than one ANN can generate similar results. The numbers of input and output nodes are problem dependent and are decided based on the number of input and output variables decided for the given problem. The flexibility lies in selecting the number of hidden layers and in assigning the number of nodes to each of these layers. A trial-and-error procedure is generally applied to decide on the optimal architecture. An excellent review of different approaches that allow the determination of network architecture with acceptable performance on the training is provided by Bishop (1995).

Training is a process by which the connection weights of an ANN are adapted through a continuous process of stimulation by the environment in which the network is embedded. 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. A score or grade is used to rate the network performance over a series of training patterns. The manner in which the nodes of an ANN are structured is closely related to the algorithm used to train it. Some of the algorithms used for ANN training include back-propagation algorithm, conjugate gradient algorithms, radial basis function, cascade correlation algorithm. All these algorithms are briefly described in ASCE (2000 a) with an exhaustive list of references.

3.2 APPLICATIONS IN HYDROLOGY

ANNs have found increasing use in diverse disciplines ranging over perhaps all branches of engineering and science. The ability to learn and generalize "knowledge" from sufficient data pairs makes it possible for ANNs to solve large-scale complex problems such as pattern recognition, nonlinear modeling, classification, association, control, and others -all of which find application in hydrology today. One attractive feature of ANNs is their ability to extract the relation between the inputs and outputs of a process, without the physics being explicitly provided to them. They are able to provide a mapping from one multivariate space to another, given a set of data representing that mapping. Even if the data is noisy and contaminated with errors, ANNs have been known to identify the underlying rule. These properties suggest that ANNs may be well-suited to the problems of estimation and prediction in hydrology.

Flood and Kartam (1994, 1997) reviewed the application of ANNs to various branches of civil engineering. However, this study did not focus on applications in hydrology. More recently a review of ANN applications in hydrology is presented in ASCE (2000b). ANNs have been used by researchers for rainfall-runoff modeling, stream flow prediction, ground-water modeling, water quality, water management, precipitation forecasting, time series, reservoir operations, and other hydrologic applications (ASCE, 2000b). Although there are a number of works reported in the literature, one or two articles on each topic are cited in the following paragraphs for illustration. It may be noted that they need not necessarily represent all the works carried out on the topic.

The problem of rainfall-runoff modeling has received the maximum attention by ANN modelers in hydrology. This problem lends itself admirably to ANN applications (Hsu *et al* 1995). The nonlinear nature of the relationship, availability of long historical records, and the complexity of physically-based models in this regard, are some of the factors that have caused researchers to look at alternative models-and ANNs have been a logical choice. Tokar and Johnson (1999) reported that ANN models provided higher training and testing accuracy when compared with regression and simple conceptual models. Their goal was to forecast daily runoff for the Little Patuxent River, Maryland, with daily precipitation, temperature, and snowmelt equivalent serving as inputs. It was found that the selection of training data has a large impact on accuracy of prediction. The authors trained and tested the ANN with wet, dry, and average-year data, respectively, as well as combinations of these, in order to illustrate the impact of the training series on network performance. The ANN that was trained on wet and dry data had the highest prediction accuracy. The length of training record had a much smaller impact on network performance than the types of training data.

ANNs have achieved some success in stream flow prediction (time series), particularly when these are desired over a certain range of stream flow values. A major limitation appears to be in trying to design robust prediction techniques over a wide range of streamflows. The studies of Karunanithi *et al* (1994) and Thirumalaiah and Deo (1998) directed network training to better replicate low stream flow events, while Poff *et al* (1996) concentrated on high flow events to generate improved statistics for floods.

Aziz and Wong (1992) illustrated the use of ANNs for determining aquifer parameter values from normalized drawdown data obtained from pumping tests-commonly referred to as the inverse problem in ground-water hydrology. This study drew on the pattern recognition ability of an ANN based on aquifer test data. Yang *et al* (1997) utilized an ANN to predict water table elevations in subsurface-drained farmlands. Daily rainfall, potential evapotranspiration, and previous water table locations were selected as inputs to the ANN.

ANNs were used for estimating precipitation with limited success. Zhang *et al* (1997) proposed that ANNs need to be employed in groups when the transformation from the input to the output space is complex. This group theory treats the input output mapping as being piecewise continuous. The idea is that each network predicts only in the range where the transformation is continuous, while a "reasoning" network determines the appropriate summation of responses. The authors were successful in making half-hourly rainfall estimates.

Recently, prediction of watershed runoff using Bayesian concepts and modular neural networks is made by Anmala *et al* (2000). In this article networks were organized in a modular architecture to handle complex sets of rainfall-runoff data. Such data often contain examples corresponding to different rules that may be associated with high, low, and medium streamflows. Different modules within the network were trained to learn subsets of the input space in an expert fashion. A gating network was used to mediate the response of all the experts. The problem was posed as one of Bayesian statistics combined with maximum likelihood estimation of network parameters. The performance of modular networks in predicting runoff over three medium-sized watersheds was examined. On the basis of the results, modular based networks appear to be a good alternative for predicting runoff and in other situations in which a hydrologic variable can be descretised into number of states.

While the articles reviewed in this paper on ANN applications in hydrology are not exhaustive, it is obvious that ANNs have made a significant impact in this area. For the sake of illustration, a couple of studies carried out by author and his team is mentioned briefly next.

3.2.1 River Flow Forecasting using Recurrent Neural Networks

Nagesh Kumar *et al* (2004) used two different networks, namely the feed forward network (FFN) and the recurrent neural network (RNN), to

Table 1. Network configuration for the FFN and RNN models

Variable	*FFN*	*RNN*
Number of input units	5	5
Number of hidden layers	2	2
Number of units in first hidden layer	14	10
Number of units in second hidden layer	14	10
Number of output units	1	1
Number of previous output values passed to first hidden layer		3
Value of learning rate	0.01	0.90
Value of the momentum rate	0.2	0.00
Number of iterations performed for training	50,000	50,000
Root mean squared value of pattern error	0.161	0.154

forecast monthly river flows. The feed forward network is trained using the modified back propagation algorithm and the recurrent neural network is trained using the method of ordered partial derivatives. Details of the FFN and RNN configurations are given in Table 1. The selected networks were used to train and forecast the monthly flows of Hemavathi river, Karnataka, India, with a catchment area of 5,189 sq. km. up to the gauging site at Akkihebbal. Monthly flow data is available at the site from 1916-17 to 1972-73, *i.e.* for 57 years. The monthly flow ranges from 1.84 million cubic metres (M.cu.m.) in summer months to 2,894.81 M.cu.m. in monsoon months.

The trained networks are used for both single step ahead and multiple step ahead forecasting. A comparative study of both networks indicated that the recurrent neural networks performed better than the feed forward networks. In addition, the size of the architecture and the training time required were found to be less for the recurrent neural networks.

3.2.2 Temporal Disaggregation of Rainfall Data using ANN

Seasonal rainfall in many regions is influenced by global climatic parameters such as El Nino Southern Oscillation (ENSO), La Nina. Seasonal rainfall predictions are based on such global climatic parameters. However, for operational purposes, such as reservoir operation or river basin management in a larger context, rainfall data at a finer time interval are required. Nagesh Kumar and Raju (2000) developed a temporal disaggregation model based on Artificial Neural Networks for obtaining rainfall of monthly or shorter time periods from the given seasonal rainfall prediction. A feed forward neural network with back-propagation algorithm for learning was used. The methodology is applied to obtain monthly rainfall data from monsoon (June-September season) rainfall data for a sub-division in Orissa state. Monthly rainfall data for 124 years (1871-1994) for Orissa state, India was used for application. Rainfall data for the first 100 years was used for ANN training and remaining 24 years data was used for testing. Seasonal rainfall is used as the input neuron and the corresponding four monthly rainfall values as

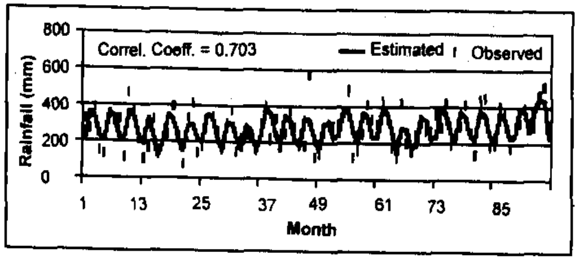

Figure 2. Comparison of observed and disaggregated rainfall using ANN (1,5,5,4) for the 4 monsoon months for 24 years (1971-1994)

4 output neurons. Two ANN architectures viz., (*i*) ANN with one hidden layer comprising of 10 neurons denoted as ANN(1,10,4) and (*ii*) ANN with two hidden layers each comprising of 5 neurons denoted as ANN(1,5,5,4) were used. A comparison of disaggregated rainfall data using ANN(1,5,5,4) with the corresponding observed rainfall data is shown in Figure 2. The disaggregated monthly rainfall data compared well with observed monthly data. In addition they preserved all the basic statistics such as summing to the seasonal value, cross correlation structure among monthly flows.

3.3 MERITS OF ANN

ANN derives its computing power through, (*i*) its massively parallel distributed structure and (*ii*) its ability to learn and therefore generalize. *Generalization* refers to the neural network producing reasonable outputs for inputs not encountered during training (learning). These two information-processing capabilities make it possible for neural networks to solve complex (large-scale) problems that are currently intractable. The use of neural networks offers the following useful properties and capabilities (Haykin, 1999):

1. *Nonlinearity:* An artificial neuron can be linear or nonlinear. A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear. Moreover, the nonlinearity is of a special kind in the sense that it is *distributed* throughout the network. Nonlinearity is a highly important property, particularly if the underlying physical mechanism responsible for generation of the input signal (*e.g.*, runoff from a watershed) is inherently nonlinear.

2. *Input-Output Mapping:* A popular paradigm of learning called *supervised learning* involves modification of the synaptic weights of a neural network by applying a set of labeled *training samples* or *task examples.* Thus the network learns from the examples by constructing an *input-output mapping* for the problem at hand. Such an approach brings to mind the study of *nonparametric statistical inference,* which is a branch of statistics dealing with model-free estimation (the term "nonparametric" is used here to signify the fact that no prior assumptions are made on a statistical model for the input data).

3. *Adaptivity:* A neural network trained to operate in a specific environment can be easily *retrained* to deal with minor changes in the operating environmental conditions. Moreover, when it is operating in a *nonstationary* environment (*i.e.*, one where statistics change with time), a neural network can be designed to change its synaptic weights in real time (*e.g.*, modeling nonstationary hydrologic time series). As a general rule, it may be said that the more adaptive we make a system, all the time ensuring that the system remains stable, the more robust its performance will likely to be when the system is required to operate in a nonstationary environment. It should be emphasized, however, that adaptivity does not always lead to robustness; indeed, it may

do the very opposite. To realize the full benefits of adaptivity, the principal time constants of the system should be long enough for the system to ignore spurious disturbances and yet short enough to respond to meaningful changes in the environment; the problem described here is referred to as the *stability-plasticity dilemma* (Grossberg, 1988).

4. *Evidential Response:* In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to *select*, .but also about the *confidence* in the decision made. This latter information may be used to reject ambiguous patterns, should they arise, and thereby improve the classification performance of the network.

5. *Contextual Information:* Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, a neural network deals with contextual information naturally.

6. *Fault Tolerance:* A neural network, implemented in hardware form, has the potential to be inherently *fault tolerant*, or capable of robust computation, in the sense that its performance degrades gracefully under adverse operating conditions. For example, if a neuron or its connecting links are damaged, recall of a stored pattern is impaired in quality. However, due to the distributed nature of information stored in the network, the damage has to be extensive before the overall response of the network degraded seriously.

7. *VLSI Implementability:* The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks. This same feature makes a neural network well suited for implementation using *very-large-scale-integrated* (VLSI) technology. One particular beneficial virtue of VLSI is that it provides a means of capturing truly complex behavior in a highly hierarchical fashion (Mead, 1989).

8. *Uniformity of Analysis and Design:* Basically, neural networks enjoy universality as information processors in the sense that the same notation is used in all domains involving the application of neural networks. This feature manifests itself in different ways:

- Neurons, in one form or another, represent an ingredient *common* to all neural networks.
- This commonality makes it possible to *share* theories and learning algorithms in different applications of neural networks.
- Modular networks can be built through a *seamless integration of modules.*

9. *Neurobiological Analogy:* The design of a neural network is motivated by analogy with the brain, which is a living proof that fault tolerant parallel processing is not only physically possible but also fast and powerful. Neurobiologists look to (artificial) neural networks as a research tool for the

interpretation of neurobiological phenomena. On the other hand, engineers look to neurobiology for new ideas to solve problems more complex than those based on conventional hard-wired design techniques.

3.4 DEMERITS OF ANN

Although several studies indicate that ANNs have proven to be potentially useful tools in hydrology, their disadvantages should not be ignored. A major limitation of ANNs is their lack of physical concepts and relations. Most ANN applications have been unable to explain in a comprehensibly meaningful way the basic process by which ANNs arrive at a decision. This has been one of the primary reasons for the skeptical attitude towards this methodology. ANNs can be trained on input-output data pairs with the hope that they are able to mimic the underlying hydrologic process. In essence, the physics is locked up in the set of optimal weights and threshold values and is not revealed back to the user after training. Therefore, artificial neural networks cannot be considered as a panacea for hydrologic problems, nor can they be viewed as replacements for other modeling techniques. Many studies have also warned about the pitfalls of ANNs and caution against their use indiscriminately (Chatfield, 1993; Carpenter and Barthelemy, 1994). It is highly desirable that ANNs be capable of imparting an explanation, even if only a partial one, as an integral part of its function. Some effort along these lines has gone into the formation of knowledge-based ANNs and rule extraction techniques. Hsu *et al* (1997) provided some heuristic functional relationships between input and output variables of an ANN by using self-organizing feature maps of the input variables. However, such ideas have not been utilized in hydrologic applications.

Another issue is that there is no standardized way of selecting network architecture. The choice of network architecture, training algorithm, and definition of error are usually determined by the user's past experience and preference, rather than the physical aspects of the problem. Very often, ANN training is performed with limited length of hydrologic data. Under these circumstances, it will be difficult to say 'when generalization will fail', to decide the range of applicability of the ANN. Unfortunately, this question cannot be answered easily because cross validation does not always provide a complete answer. In contrast, physically- based methods have an advantage, because the physics can be used to fill the gaps where data is not available.

For ANN training, back propagation algorithm (Rumelhart *et al* 1986) is the most frequently used and popular algorithm. In spite of the potential capabilities of back propagation, it suffers from several drawbacks that can lead to problems during training. The most commonly faced problems (ASCE, 2000a) are long, ambiguous training process, local minima, moving target and network paralysis. These problems are briefly explained below.

- *Long Training Process* : For complex problems, the direction of error reduction with a number of epochs is not very clear and may lead to some confusion regarding the continuation of the training process. The training process can be very long and sometimes futile for a bad initial weight space. The two main reasons behind slow learning are problems of step size and moving target.
- *Local Minima* : During the search for an optimal weight space, the error is continuously reduced by reducing the step size until an optimal minimum is reached. For most practical scenarios, this local minimum may yield good results or may be the global minimum itself. However, no fixed guidelines are available for the rate at which this step size should be reduced during network training. Reducing the step size by infinitesimal amounts may not be practically feasible. A large step size may propel the search in a very different region of weight space and may yield a poor solution. The problem of local minima is faced by many traditional optimization methods, and such is the case for neural networks. This problem arises due to downward search in a complex, high-dimensional space full of hills, valleys, saddle points, etc. By changing the learning rate or step size, this problem can be avoided to some degree.
- *Moving Target*: The problem of moving target arises as the weights have to continuously adjust their values from one output to another for successive patterns. The change in a weight during one training pass may be nullified in the next pass because of a different training pattern. Thus, the attenuation and dilution of weight changes also slows down the process of training.
- *Network Paralysis*: The problem of network paralysis arises due to the large adjustment of weights in the initial epochs. When all the nodes produce large outputs, the derivative of the activation function can become very small. Since the error sent back from the output layer in the backward pass (or the weight adjustment) is proportional to the derivative of the activation function, the learning process slows down, and weight adjustment might be insignificant. Reducing the step size or learning rate usually avoids this problem.

Many improvements are suggested in the literature to overcome some of these limitations in the conventional back propagation algorithm and a few of them are briefly discussed in Nagesh Kumar *et al* (2004).

3.5 CONCLUSIONS

Following conclusions may be drawn from the present study.

- A very interesting feature of ANNs is the data exploration with capabilities to discover unknown dependencies and relationships

among the variables. ANNs seem to learn interesting and new nonlinear relationships and bring out features of the input data that are not revealed by other techniques. There is need to find innovative ways of extracting this information, which is buried in the weight and threshold vectors of a trained network in an incomprehensible fashion. This feature has received very little attention among hydrologists.
- Proper use of an ANN requires not only a physical understanding of the hydrological process under consideration but also knowledge of ANNs and their operation. Trying to extract rules from a network or impart them with some explanation capability will entail extra computer effort. These fundamental aspects will lead to the construction of good training and validation data sets, selection and inclusion of relevant input variables, and development of proper ANN architectures and selection of training algorithms.
- Problems that involve decisions at several levels, or those dealing with hierarchical decision making may be better represented through modular neural networks, which consist of a group of regular neural networks. Many other applications in hydrology using ANN should be forthcoming but at the same time proper care should be taken in adopting ANN to different problems in hydrology.

3.6 ACKNOWLEDGEMENTS

For the preparation of this article and discussion on various critical issues, work presented by ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000 a, b) is extensively referred. The author wishes to acknowledge the efforts of the task committee for highlighting many relevant issues of this recent topic.

REFERENCES

1. 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), pp. 115-123.
2. ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. 2000b. Artificial neural networks in hydrology. II: hydrologic applications, Journal of Hydrologic Engineering, ASCE, 5(2), pp. 124-137.
3. J Anmala, B. Zhang, RS Govindaraju. 2000. Comparison of ANNs and empirical approaches for predicting watershed runoff, Journal of Water Resources Planning and Management, ASCE, 126(3), pp. 156-166.
4. ARA Aziz, KFV Wong. 1992. Neural network approach to the determination of aquifer parameters, Ground Water, 30(2), pp. 164- 166.
5. CM Bishop. 1995. Neural Networks for Pattern Recognition, O.U.P New York.

6. WC Carpenter, JF Barthelemy. 1994. Common misconceptions about neural networks as approximators, Journal of Computing in Civil Engineering, ASCE, 8(3), pp.345-358.
7. C Chatfield. 1993. Neural networks: forecasting breakthrough or passing fad, International Journal of Forecasting, 9, pp.1-3.
8. I Flood, N Kartam. 1994. Neural networks in civil engineering. I: principles and understandings, Journal of Computing in Civil Engineering, ASCE, 8(2), pp. 131-148.
9. I Flood, N Kartam. 1997. Systems, Artificial Neural Networks for Civil Engineers, in Fundamentals and Applications, N Kartam, I Flood, JH Garrett Jr., (Eds.) ASCE, New York, pp. 19-43.
10. S Grossberg. 1988. Neural Networks and Natural Intelligence, MIT Press, Cambridge, MA.
11. S Haykin.1999. Neural Networks: A Comprehensive Foundation, Prentice Hall, New Jersey.
12. K Hsu, HV Gupta, S Sorooshian. 1995. Artificial neural network modeling of the rainfall-runoff process, Water Resources Research, 31(10), pp. 2517-2530.
13. N Karunanithi, WJ Grenney, D Whitley, K Bovee. 1994. Neural networks for river flow prediction. Journal of Computing in Civil Engineering , ASCE, 8(2), pp. 201-220.
14. HR Maier, GC Dandy. 1996. The use of artificial neural networks for the prediction of water quality parameters, Water Resources Research, 32(4), pp. 1013-1022.
15. CA Mead. 1989. Analog VLSI and Neural Systems, Addison-Wesley Pub. Co.
16. D Nagesh Kumar, K Srinivasa Raju, T Sathish. 2004. River flow forecasting using recurrent neural networks, Water Resources Management, 18(2), pp.143-161.
17. D Nagesh Kumar, K Srinivasa Raju. 2000. Temporal disaggregation of rainfall data using artificial neural networks, Proceedings of International Conference on Integrated Water Resources Management for Sustainable Development, New Delhi, India, December 19-21, pp. 788-795.
18. NL Poff, S Tokar, P Johnson. 1996. Stream hydrological and ecological responses to climate change assessed with an artificial neural network, Limnology and Oceanography, 41(5), pp. 857-863.
19. DE Rumelhart, GE Hinton, RJ Williams. 1986. Learning internal representation by back-propagating errors, Nature, 32(3), pp. 533-536.
20. K Thirumalaiah, MC Deo. 1998. River stage forecasting using artificial neural networks, Journal of Hydrologic Engineering, ASCE, 3(1), pp. 26-32.
21. AS Tokar, PA Johnson. 1999. Rainfall-runoff modeling using artificial neural networks, Journal of Hydrologic Engineering, ASCE, 4(3), pp. 232- 239.
22. CC Yang, SO Prasher, R Lacroix, S Sreekanth, NK Patni, L Masse. 1997. Artificial neural network model for sub surface drained farmland, Journal of Irrigation and Drainage Engineering Division, ASCE, 123(4), pp. 285-292.
23. M Zhang, J Fulcher, RA Scofield. 1997. Rainfall estimation using artificial neural network group, Neurocomputing, 16, pp. 97-115.