Rainfall-Runoff Analysis using Artificial Neural Network

Ms. S.V.Shende[1] and Dr. D.C.Joshi[2]
[1]Research student, Department of Mathematics, V.N.S.G.U, Surat
[2]Asso. Prof, Department of Mathematics, V.N.S.G.U, Surat

Abstract

Artificial Neural Networks(ANN), which are simplified models of the biological neurons system, is a massively parallel distributed processing system made up of highly interconnected neural computing elements that have the ability to learn and thereby acquire knowledge and make it available for use. The application of ANN methodology for modeling amount of runoff for catchment of the Machhan river located in Dahod district of Gujarat(India) is presented. Dahod district is a semi –arid region with around 800m.m of average annual rainfall which is erratic in nature. The model uses rainfall as input and gives runoff as output. On calibration of the model is found to be giving good comparison of the observed and simulated flows.

Introduction

The relationship of rainfall-runoff is a complex hydrological phenomenon, as this process is highly nonlinear, time-vary and spastically distributed. Rainfall-runoff models play an important role in water resource management planning and therefore, different types of models with various degrees of complexity have been developed for this purpose. These models, regardless to their structural diversity generally fall into three broad categories; namely, black box or system theoretical models, conceptual models and physically-based models.

Artificial neural networks are flexible mathematical structures that are capable of identifying complex nonlinear relationships between input and output data sets. ANN is widely accepted as a potentially useful way of modeling complex non-linear systems with a large amount of data. ANN is particularly useful in situation where the underlying physical process is not yet fully understood, and can be used as a substitute for the conventional and statistical models. A neural network consists of a large number of simple processing elements that are variously called neurons, units, cells, or nodes. Each neuron is connected to other neurons by means of direct communication links, each with an associated weight that represents information being used by the net to solve a problem. The network usually has two or more layers of processing units where each processing unit in each layer is connected to all processing units in the adjacent layers.

Brief Review of Artificial Neural Network

There has been a tremendous growth in the interest of application the ANNs in rainfall-runoff modeling in the 1990s. ANNs were usually assumed to be powerful tools for functional relationship establishment or nonlinear mapping in various applications. Cannon and Whitfield, found ANNs to be superior to linear regression procedures. Shamseldin, examined the effectiveness of rainfall-runoff modeling with ANNs by comparing their results with the

Simple Linear Model (SLM), the seasonally based Linear Perturbation Model (LPM) and the Nearest Neighbor Linear Perturbation Model (NNLPM) and concluded that ANNs could provide more accurate discharge forecasts than some of the traditional models. The ability of ANNs as a universal approximation has been demonstrated when applied to complex systems that may be poorly described or understood using mathematical equations.

An ANN is a highly interconnected network of many simple processing units called neurons, which are analogous to the biological neurons in the human brain. Neurons having similar characteristics in an ANN are arranged in groups called layers. The neurons in one layer are connected to those in the adjacent layers, but not to those in the same layer. The strength of connection between the two neurons in adjacent layers is represented by what is known as a 'connection strength' or 'weight'. An ANN normally consists of three layers, an input layer, a hidden layer and an output layer. In a feed-forward network, the weighted connections feed activations only in the forward direction from an input layer to the output layer. On the other hand, in a recurrent network additional weighted connections are used to feed previous activations back to the network. The structure of a feed-forward ANN is shown in Fig. 1.

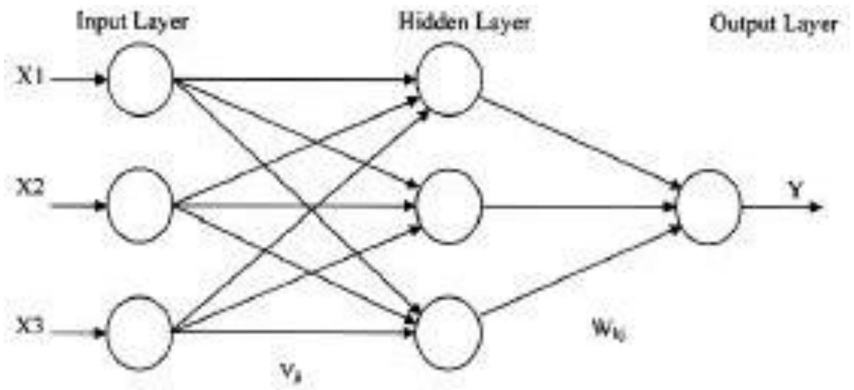

Fig. 1: Structure of a feed-forward ANN

The activation function of the artificial neurons in ANNs implementing the backpropagation agorithm is a weighted sum (the sum of the inputs x multiplied by their respective weights w)

The output of the neuron is computed from a nonlinear activation function. The most commonly used in this type of network is the logistic sigmoid function. This activation function is continuously differentiable, symmetric and bounded between -1 and 1.

An important step in developing an ANN model is the determination of its weight matrix through training. There are primarily two types of training mechanisms, supervised and unsupervised. A supervised training algorithm requires an external teacher to guide the training process. The primary goal in supervised training is to minimize the error at the output layer by searching for a set of connection strengths that cause the ANN to produce outputs that are equal to or closer to the targets. A supervised training mechanism called back-propagation training algorithm is normally adopted in most of the engineering applications. Another class of ANN models that employ an 'unsupervised training method' is called a self-organizing neural network. The most famous self-organizing neural network is the Kohonen's Selforganizing Map (SOM) classifier, which divides the input-output space into a desired number of classes. Once the classification of the data has been achieved by using a SOM classifier, the separate feed forward MLP models can be developed through considering the data for each class using the supervised training methods. Since the ANNs do not consider the physics of the problem, they are treated as black-box models; however, some researchers have recently reported that it is possible detect physical processes in trained ANN hydrologic models.

Introduction of the Area

The state of Gujarat is situated between 20° 6’ to 24°42’ north latitude and 68°10’ to 74°28’ east longitude, it covers a total geographical area of 1,96,024 Km^2 in western part of india. It has common borders with the state of Rajasthan, Madhya Pradesh and Maharashtra along North, East and South and with Pakistan in North-West.

Dahod district is located in the North-East direction of Gujarat. Jhalod is one of the taluka places of Dahod. The Machhan river is flowing through Jhalod taluka. The number of rivers meets the Machhan River.

The overall topography of the region from which Machhan river flows is highly swelling and of varying slopes. The catchment of the river is 431 Km2 up to its meeting place with river Anas wchich is a tributary of Mahi River.

South-west part of basin is occupied by Basalt. It occurs as small flat-topped to conical hills rising above the flatness of Precambrain terrain which is covered with a residual black cotton soil. The area is covered by compact mermorphic rock of Proterozoic succession. Quartzite and phyllite belonging to Lunawada group are encountered in the study area. The most common rock types granites, gneisses, quartzite, schist and associated phyllites, slate etc are found in the catchment.

Dahod district is moderately humid area. The basin receives much of its rainfall from the southwest monsoon during the period between June and September, its maximum intensity being in the months of July and August. The annual rainfall in the area varies between 800 mm to 1000mm.

Mathematical Formulation

Model was calibrated for the year 2003 on daily basis for the monsoon period of 153 days. The output function, Ai , of linear system is related to the input function x_i received by it

$$A_i = \sum_{i=0}^{n} x_i \, w_i \qquad (1)$$

Here we create a feed forward backpropagation network. This means that the artificial neurons are organized in layers, and send their signals “forward”, and then the errors are propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. There may be one or more intermediate hidden layers. The backpropagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea of the backpropagation algorithm is to reduce this error, until the ANN learns the training data. The training begins with random weights, and the goal is to adjust them so that the error will be minimal. Hidden layers use Hyperbolic symmetric sigmoidal function as transfer function, which converts a neural network layer’s input into its net output. The mathematical expression of the same is given by

$$y_i = \frac{2}{1+e^{-2A_i}} - 1 \quad (2)$$

Output layer use Linear transfer function.

Now, the goal of the training process is to obtain a desired output when certain inputs are given. Since the error is the difference between the actual and the desired output, the error depends on the weights, and we need to adjust the weights in order to minimize the error.

The backpropagation algorithm now calculates how the error depends on the output, inputs, and weights. After we find this, we can adjust the weights using the method of gradient descendent:

$$\Delta w_i = -\gamma \frac{\partial E}{\partial w_i} \quad (3)$$

We use Gradient decent with adaptive learning backpropagation. An adaptive learning rate requires some changes in the training procedure. First, the initial network output and error are calculated. At each epoch new weights and biases are calculated using the current learning rate. New outputs and errors are then calculated.

Performance Evaluation Criteria

The performance evaluation measure included in this study are mean square error(MSE) and mean absolute error(MAE).

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=0}^{n} (Y_i - y_i)^2 \quad (4)$$

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=0}^{n} |Y_i - y_i| \quad (5)$$

Where Yi, yi are observed and predicted runoff values.

Description	**Mean Square Error**	**Mean Absolute Error**	**Training : R**	**Testing : R**
Three Hidden Layers	43.1399	2.8234	0.6918	0.9468
Five Hidden Layers	37.8563	2.8214	0.6892	0.9691

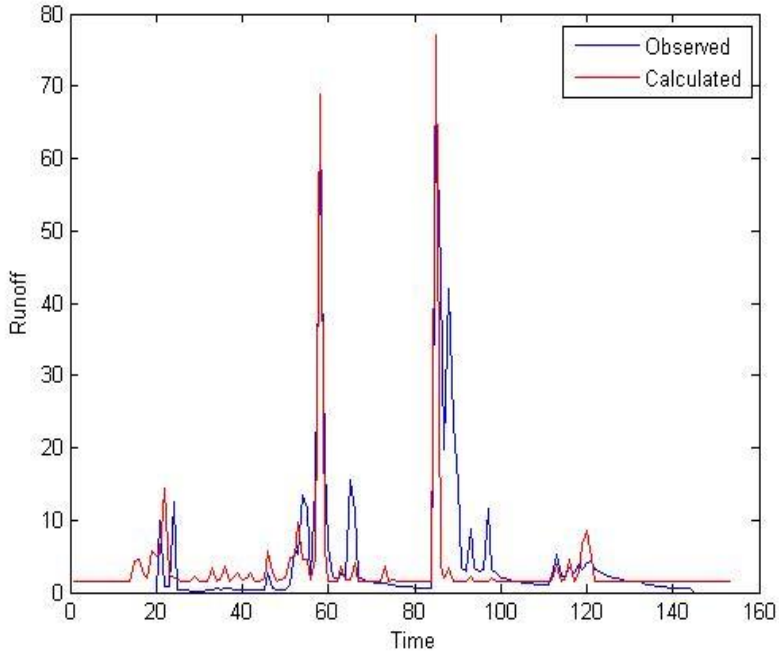

Fig 2 : Three Hidden Layers

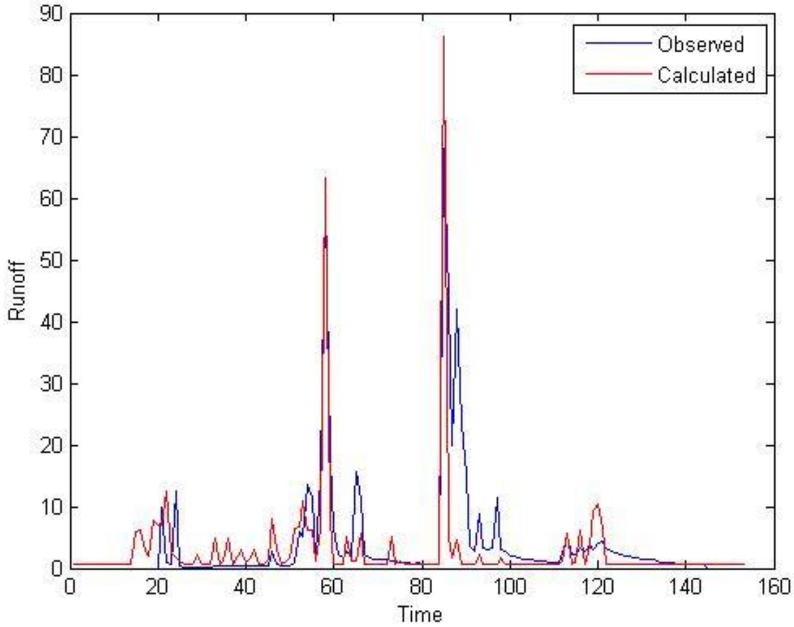

Fig 3 : Five Hidden Layers

Conclusion

As runoff generation involves lots of physical parameter which is not feasible to take into account at the time of calculation hence to overcome such problem ANN used because in ANN modeling neither physical process nor information about the catchment area necessary to incorporate hence it entice as a substitute for watershed modeling. In above table and graph comparison of initial and final stage shown. But final stage can be improved by increasing number of neurons and number of layers. Moreover input function, number of neurons and number of layers can be decided as per the complexity of the problem.

References

1. Luk, K., Ball, J.E., and Sharma, A. 2001. An application of artificial neural networks for rainfall forecasting. Mathematical and Computer Modeling 33: 683-693.

2. Rajurkar, M.P., Kothyari, U.C., and Chaube, U.C. 2004. Modeling of the daily rainfall-runoff relationship with artificial neural network. Journal of Hydrology 285: 96-113.

3. Govindaraju R. S. (2000). "Artificial Neural Networks In Hydrology, Part II: Hydrologic Applications". J. of Hydr. Engrg., 5(2): 124-135.

4. Karunithi N., Grenney W.J., Whitley D., and Bovee K. (1994). "Neural Networks for River Flow Prediction". J. of Comp. in Civil Eng. 8(2): 201-220.

5. MathWorks, Inc., 2001. MATLAB. 3 Apple Hill Drive, Natick, MA, USA.

6. Sudheer, K.P. and A. Jain, 2004. Explaining the internal behavior of artificial neural network river flow models. Hydrol. Process, 118 (4): 833-844.

7. Wilby, R.L., R.J. Abrahart and C.W. Dawson, 2003. Detection of conceptual model rainfallrunoff processes inside an artificial neural network, Hydrol. Sci. J., 48 (2): 163-181.

8. Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modeling. J.Hydrol., 199: 272-294.