ISSN:0975-9646

Neuro-Genetic Hybrid Approach for Rainfall Forecasting

Abhishek Saxena, Neeta Verma, K.C.Tripathi

Department of Computer Science & Engineering,
Inderprastha Engineering College, Ghaziabad,
Uttar Pradesh, INDIA

Abstract-Weather is certainly the most important factor over which man has no control, and hence it has created dominance on the success or the failure of agricultural enterprises. Most important efforts since long time have been on weather and rain forecasting. However, the unpredictable nature of rainfall has not changed. Meteorologists can neither solve nor evaluate the problem of effective rainfall merely from tables of frequency, amount and intensity of rainfall or from physical phenomena in the atmosphere. It is a task in which several disciplines and sub-disciplines overlap. The present study investigates the ability of hybrid approach of artificial neural network (ANN) and genetic algorithm (GA) in forecasting rainfall. A standard feed forward network (FFN) is utilized for performing the prediction task. Moreover GA is used to determine the optimal structure of ANN.

***Keywords*: Rainfall Forecasting, Artificial Neural Networks (ANN), Genetic Algorithm (GA), Neuro-Genetic Hybrid, Feed forward Network**

1. INTRODUCTION

Rain is one of the nature's greatest gifts. In developing countries, the primary source of water for agricultural production is rainfall. It is thus a major concern, to identify any trends for rainfall to deviate from its periodicity; otherwise it would disrupt the economy of the country. This fear has occurred due to threat caused by global warming and greenhouse effect. The parameters that are required to predict the rainfall are enormously wide and complex. Thus accurate prediction to forecast rainfall is one of the greatest challenges in operational hydrology; despite many advances being made in weather forecasting in recent decades.

There are two approaches to predict and forecast rainfall. The first approach involves the study of the rainfall processes in order to model the underlying physical laws. However, this physical based process modeling approach is not feasible since (i) rainfall is an end product of a number of complex atmospheric processes which vary both in space and time, (ii) the volume of calculations involved may be prohibitive even if the rainfall processes can be described concisely and completely, (iii)and the data that is available to assist, in definition of control variables for the process models, such as rainfall intensity, wind, speed, and evaporation,etc, are limited in both the spatial and temporal dimensions.

The importance of precise knowledge on the subject of effective rainfall needs little elaboration. Due to this, the development of Artificial Neural Networks (ANN); which performs nonlinear mapping between inputs and outputs, has become a second approach to forecast rainfall [1]. ANN is able to approximate any continuous function to arbitrary accuracy. It becomes an attractive inductive approach in rainfall forecasting owing to their high nonlinearity, flexibility and data driven, learning nature in building models [2], without having any prior knowledge about catchment behavior and flow processes. Researchers often overlook the effect of parameters [3] on the performance of ANN.

The performance of ANN is critically dependent on the learning algorithms, the network architecture and the choice of the control parameters. Even when a suitable setting of parameters (weight) can be done, the ability of the resulting network to generalize the data is not seen during learning and hence it may be far from optimal behavior. For these reason, it seems logical and attractive to apply genetic algorithms.

Genetic Algorithms (GA) may provide a useful tool for automating the design of neural network. Although GA and ANN have evolved along separate paths and each have its own strengths and weaknesses but recently [4-8] there have been attempts to combine the two technologies to form Neuro-Genetic Hybrid Systems. These models are used to forecast rainfall for short regions as well as for large regions and even for the whole country [9, 10].

In this paper, we have used hybrid Neuro-Genetic system, in which GA is used for the selection of neural network architecture, connection weights in the prediction for rainfall analysis. MATLAB ANN's toolbox is employed to optimize ANN for this purpose. The program is capable of plotting the various relations between the neural network output and the required target. The program stops calculation when the mean square error or sum square error reaches a desired minimum or met one of the default stopping criteria.

[A] Artificial Neural Network

ANN, often just called a neural network, is a mathematical model inspired by biological neural networks. The motivation for the development of neural network technology stemmed from the desire to implement an artificial system that could perform intelligent tasks similar to those performed by the human brain. It is a powerful data modeling tool that is able to capture and represent complex input output relationships. In most cases a neural network is an adaptive system, between inputs and outputs, to find patterns in data.

Neural network resemble the human brain in the following two ways:
1) A neural network acquires its knowledge through learning.
2) A neural network's knowledge is stored within interneuron connection strengths known as synaptic weights
A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. An ANN is composed [11] of basic units called *neurons* that are processing elements (PEs) in a network. Each neuron receives input data, processes it and delivers a single output. The input data can be raw data or output of other processing elements (PEs). The output can be the final product or it can be an input to another neuron.
An ANN is formed by nodes connected together. Nodes with similar characteristics are arranged into layer. A layer can be seen as a group of nodes which have connections to other layers, or to external environment, but which have no interconnections. There are basically three types of layers. The first layer connecting to the input variables is called input layer. The last layer connecting to the output variables is called the output layer. It is straightforward to extend to multiple nodes. Layers between the input and output layers are called hidden layers. Information is transmitted through the connections between nodes. This type of network is called feed forward network, or multilayer feed forward network.
Layered Feed forward networks have become very popular for few reasons. First they have been found in practice to generalize well. Secondly, a training algorithm called back propagation exists which can often find a good set of weights (and biases) in a reasonable amount of tune.
Back propagation is a variation on gradient search. It generally uses a least square optimality criterion. But there are some drawbacks to backpropagation[12]. For one there is the "scaling problem". It works well on simple training problems. However, as the problem complexity increase, the performance of Back propagation falls off rapidly, which makes it infeasible for real –world problems. The performance degradation appears to stem from the fact that complex spaces have nearly global minima which are sparse among the local minima. Gradient Search techniques tend to get trapped at local minima. With a high enough gain (or momentum), back propagation can escapes local minima. When the nearly global minima are well hidden among the local minima, back propagation can end up bouncing between local minima without much overall improvement, thus making for very slow training.

[B] Genetic Algorithm

Genetic Algorithm (GA) is a search algorithm based on survival of the fittest among string structures to form a search algorithm. [8] For solution of optimization problems, GA has been investigated recently and shown to be effective at exploring a complex space in an adaptive way, guide by the biological evolution mechanisms of reproduction, crossover and mutataion.GA may avoid falling in to minimum when local search method like gradient descent is used in ANN.

Genetic Algorithms perform the search process in four stages: Initialization, Selection, Crossover and Mutation. The following are the basic steps of GA as shown in fig 1:

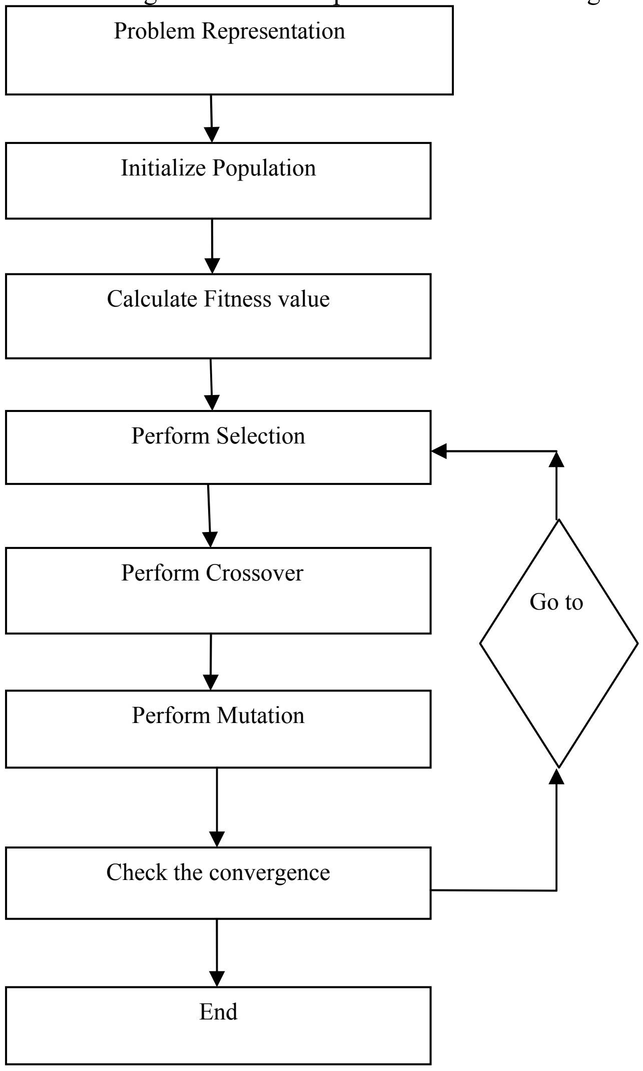

Fig 1: Steps of GA

GA serves as an, intelligent search and optimization technique, and, adaptation of network weights. It is used to improve the performance of ANN i.e. to decide the connection weight in the form of binary or real number. It is also used for topology selection, training of network [13], determining the number of nodes in each layer, evolution of connection weights, evolution of learning rule etc.
Genetic Algorithm does not have the problem with scaling as back propagation. One reason for this is that they generally improve the current best candidate monotonically. They do this by keeping the current best individual as part of their population while they search for better candidates. Secondly, genetic algorithms are generally not bothered by local minima. The mutation and crossover operators can step from a valley across a hill to an even lower valley with no more difficulty than descending directly into a valley.

2. METHODOLOGY

The 140 years monthly dataset of (1871-2010) of All India Rainfall of 30 metrological subdivisions encompassing 2,880,324 sq km with a resolution of up to 0.1mm per month is obtained from the Indian Institute of Tropical Metrology website with the original source as referred by the department being the Indian Metrological Dept (IMD) is used for analysis.

The method essentially involves the time series prediction. For determining the predictors for a series of a particular month (hereby called the predictand month), [14] the correlation of the series of the predictand month was also calculated, with those months, that precede the predictand month. Only those months have been taken as predictors, which gave a correlation coefficient better than 0.5.The following normalization scheme is used:

$$X_n = [(X-X_{min})/ (X_{max}-X_{min})]*\mathbf{0.6+0.2}$$

where X is the rainfall anomaly, X_{max} is the maximum anomaly for the month, X_{min} is the minimum anomaly for the month and X_n is the normalized value of the anomaly. The factors of 0.6 and 0.2 are included so that the normalized values are not 0.0 or 1.0.As X_n approaches these extremes values, the derivative of the sigmoid function becomes 0.For training the ANN, the Back propagation algorithm and sigmoid function are used in this model and weights are optimized by GA. However before GA is executed, a suitable coding for the problem is devised, the fitness function which assigns merit to each of the individuals in the population, is formulated, and parents are selected for reproduction and crossover to generate offspring.

For this we assume a BPN whose network configuration is l-m-n, i.e. 1-4-1. The number of weights that are to be determined are *(l+n)m* i.e. 8.With each weight (gene) being a real number, and assuming the number of digits *d* to be randomly generated, for representing a weight value as 5,the string S, represents the chromosome of weights as 8 x 5 = 40,in length. We first randomly generate the initial population [35] P_0 of size p=40.Suppose $C_1^0, C_2^0, \ldots\ldots C_{40}^0$ represent the 40 chromosomes of P_0 and, to determine the fitness values for each of the chromosomes, we extract weights from each of the chromosomes.

Suppose $x_1, x_2, \ldots\ldots x_d, \ldots\ldots, x_L$ represent a chromosome and $x_{kd+1}, x_{kd+2}, \ldots\ldots x_{(k+1)d}$ represents the kth gene ($k \geq 0$) in the chromosome. The actual weight w_k is obtained [15] by using the following Eq 1:

$$w_k = \begin{cases} +\dfrac{x_{kd+2}10^{d-2} + x_{kd+3}10^{d-3} + \ldots . x_{(k+1)d}}{10^{d-2}}, & if 5 \leq x_{kd+1} \leq 9 \\ -\dfrac{x_{kd+2}10^{d-2} + x_{kd+3}10^{d-3} + \ldots . x_{(k+1)d}}{10^{d-2}}, & if 0 \leq x_{kd+1} \leq 5 \end{cases} \qquad (1)$$

putting k = 0, d=5, in x_{kd+1}, gives x_1, such that the value of x_1 lies between 5 and 9 or between 0 and 5, gives the weight w_0 for gene 0.Similarly,putting k=1, d=5, in x_{kd+1}, gives x_6, such that the value of x_6 lies between 5 and 9 or between 0 and 5, gives the weight w_1 for gene 1.Progressing in this way, we obtain the $W_1^0, W_2^0, \ldots\ldots W_{40}^0$ be the weight sets extracted from each of C_i^0, i=1,2,……40 using the eq (1).

The fitness function is devised using algorithm FITGEN () [15] which illustrate the procedure as:

```
Algorithm  FITGEN ( )
{     Let $(I_i,T_i)$, i=1,2,….N where $I_i=(I_{1i},I_{2i},\ldots I_{1i})$ and $T_i=(T_{1i},T_{2i},\ldots .T_{ni})$ represent the input-output pairs of the problem to be solved by BPN with a configuration l-m-n.
For each chromosome $C_i$, i=1, 2… p belonging to the current population $P_i$ whose size is p.
{Extract weights $W_i$ from $C_i$ with the help of equation (1);
Keeping $W_i$ as a fixed weight setting train the BPN for the N input instances;

Calculate error $E_i$ for each of the input instances using the formula,
    $E_i = \sum_j (T_{ji} - O_{ji})^2$   ………(2)
Where $O_i$ is the output vector calculated by BPN;
Find the root mean square E of the errors $E_i$, i=1, 2,…… .N
i.e.    $E = \sqrt{\frac{\sum_i E_i}{N}}$   ………(3)
Calculate the fitness value $F_i$ for each of the individual string of the population as
    $F_i = \frac{1}{E}$   ……..(4)
}Output $F_i$ for each $C_i$, i=1, 2……..p;
}
END FITGEN
```

Using algo FITGEN(),a set of input-output pairs having 1 to 1560 values is used for this problem P and is solved by BPN with a configuration of 1-4-1.For a fixed weight set $\mathbf{W_i^0}$,the BPN is trained for all the input instances and error E_i is calculated, for each of the input instances using the equation (2).The root mean square E of the errors E_i, i = 1, 2 … N is calculated by equation (3) and the fitness value F_1 for the chromosome C_1^0 is calculated by using eq (4) of the algorithm.

Similarly the next weight set W_2^0 is extracted from the next chromosome C_2^0.The BPN as before is trained using the extracted weights, for all the given input instances. The RMSE is computed using eq (3)and the fitness value of the chromosome C_2^0 is obtained by using, $F_2=1/E$. Proceeding in this way, the fitness values for all other chromosomes in the initial population are obtained. Since the population size is p=40, for the initial population F_i, i=1, 2 ….40 are computed.

Before the parent chromosomes reproduce to deliver offspring with better fitness, the mating pool is first formed by excluding that chromosome C_l with the least fitness F^{min} and replacing it with duplicate copy of the chromosome C_k having highest fitness F^{max} i.e. the best fit individuals have multiple copies while worst fit individuals die off.

Having formed the mating pool, the parents are selected in pairs at random and recombined using the two point crossover. The offspring's which now form the current population again have their fitness calculated by algo FITGEN() which produces new population P_1.P_1 comprises 40 chromosomes which are the offsprings of previous population P_0.The P_1 now undergoes the process of selection, reproduction, and crossover. The fitness values for the chromosomes in P_1 are computed, the best individuals replicated and reproduction carried out using two-point crossover operator to form the next generation P_2 of chromosomes.

The process of generation proceeds until at one stage 95% of the chromosomes in the population P_i, converge to the same fitness value. At that stage, the weights extracted from the population P_i are the final weights used by the BPN.

The algorithm for the [15] GA based weight determination is:

Algorithm **GA-NN-WT**

```
{i ← 0;
Generate the initial population Pi of the real-coded chromosomes $c^i_j$ each representing a weight set for the BPN;
While the current population Pi has not converged
{Generate fitness values $F^i_j$ for each $\mathbf{C^i_j} \in P^i$ using the Algorithm FITGEN ( );
Get the mating pool ready by terminating worst fit individuals and duplicating high fit individuals;
Using the cross over mechanism, reproduce offspring from the parent chromosomes;
        i ← i +1;
Call the current population Pi;
Calculate fitness values $F^i_j$ for each $C^i_j \in P^i$;
}
Extract weights from $P_i$ to be used by the BPN ;}
End GA- NN-WT
```

The complete flowchart representing the Neuro-Genetic Hybrid Algorithm is as shown in fig 2

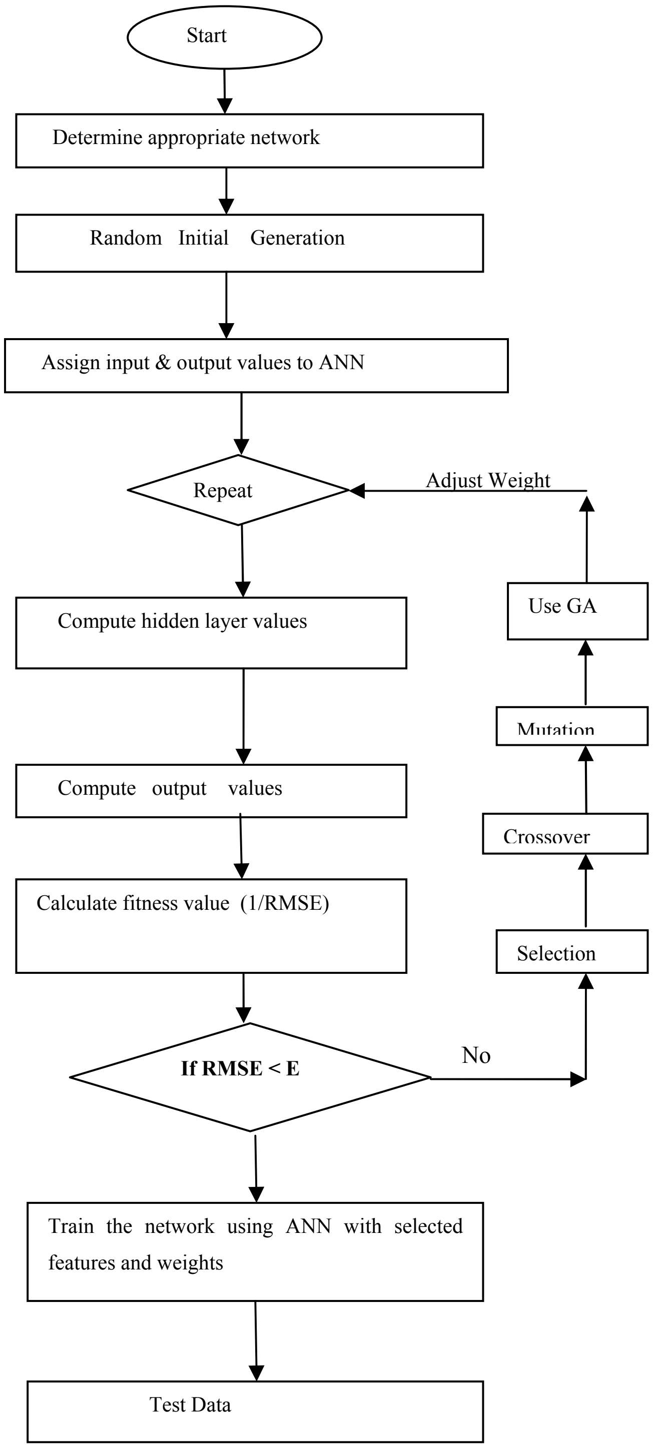
Start
Determine appropriate network
Random Initial Generation
Assign input & output values to ANN
Repeat
Adjust Weight
Compute hidden layer values
Use GA
Mutation
Compute output values
Crossover
Calculate fitness value (1/RMSE)
Selection
If RMSE < E
No
Train the network using ANN with selected features and weights
Test Data

Fig 2: Flow chart of Neuro-Hybrid Approach

3. RESULTS

In this research, we have used the following methods and parameters (table 1):

Table 1: ANN-GA parameters

Search Method	**Genetic Algorithm**
Population Size	40
Fitness Function	1/R.M.S.E
Number of Generations	5
Network Architecture Used	1-4-1
Transfer Function	Sigmoid

The GA-BPN has been implemented by taking different population size. For each value of population, the root mean square error (R.M.S.E) value has been calculated. Training results are shown in table 2

Generation	Population	R.M.S.E	Average R.M.S.E
I	P_0	11.706	0.2926
II	P_1	5.7671	0.1441
III	P_2	4.3106	0.1077
IV	P_3	5.1778	0.1294
V	**P_4**	**2.6498**	**0.0662**

Table 2: Training results with different Population size

It can be observed from the table 2 that the individual RMSE of the generation is calculated by taking the sum of all the obtained values of population while the average RMSE is taken by considering the average of the value of RMSE of respective population. Hence it shows that training gives better results with population P4.

The ANN is trained with the dataset using Neuro-Genetic Hybrid Algorithm. Figure 3 shows the error after each epoch during training process.

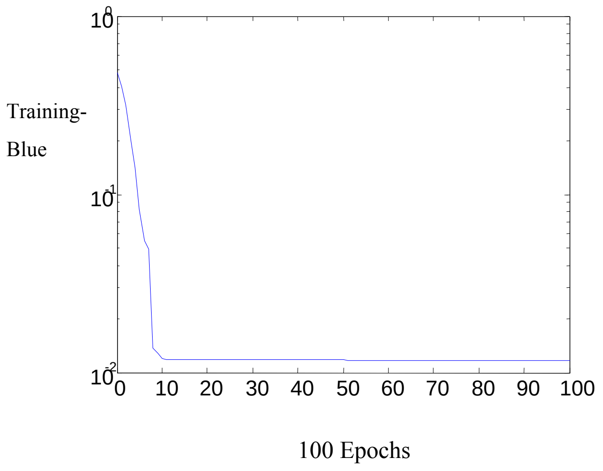

Fig 3: Error rate during Training

4. Conclusion

In this paper a method was proposed for constructing a Neural Network by employing Genetic Algorithm. A comparison is conducted between a GA-ANN Hybrid approach and traditional Back propagation ANN. It is concluded that applying Genetic Algorithm to initialize the weight of ANN can take its advantage of optimization and overcome the shortcomings of ANN's slow convergence and stuck in the local minima.

References

[1] Guoqiang Zhang, B.Eddy Patuwo,Miachael Y.Hu, "Forecasting with artificial neural network: The state of Art", International Journal of Forecasting, Vol. 14, pp. 35-62,1998

[2] Kumar Abhishek, M.P.Singh, Saswata Ghosh and Abhishek Anand, "Weather Forecasting model using Artificial Neural Network", Procedia Technology, Vol. 4, pp. 311-318, 2012

[3] Pedro A. Castillo-Valdivieso, Juan J. Merelo, Alberto Prieto, *Member, IEEE*, Ignacio Rojas, and Gustavo Romero, "Statistical Analysis of the Parameters of a Neuro-Genetic Algorithm", IEEE Transactions on Neural Networks, Vol. 13, No. 6, pp. 1374-1394,2002

[4] Yung-Keun Kwon and Byung-Ro Moon, *Member, IEEE,* "A Hybrid Neurogenetic Approach for Stock Forecasting", IEEE Transactions on Neural Networks, Vol. 18, No. 3, pp. 851-864,2007

[5] Georgios Dounias, "Hybrid Computational Intelligence in Medicine", University of the Aegean, Dept. of Business Administration,8 Michalon Street, 82100 Chios, Greece

[6] Mlungisi Dumaa, Tshilidzi Marwalaa, Bhekisipho Twalaa, Fulufhelo Nelwamondo, "Partial imputation of unseen records to improve classification using a hybrid multi-layered artificial immune system and genetic algorithm", Elsevier: Applied Soft Computing,Vol. 13 ,pp. 4461–4480,2013

[7] Giuliano Armano, Gianmaria Mancosu, Luciano Milanesi,Alessandro Orro, Massimiliano Saba1 and Eloisa Vargiu1, "A Hybrid Genetic-Neural System for Predicting Protein Secondary Structure", BMC Bioinformatics ,Vol. 6,No 4,pp. 1-7,2005

[8] Mohammad-Bagher Aghajanloo, Ali-Akbar Sabziparvar, P. Hosseinzadeh Talaee, "Artificial neural network–genetic algorithm for estimation of crop evapo transpiration in a semi-arid region of Iran", Neural Compution & Application, Vol 23, pp1387–1393, 2013.

[9] A.F.M. Khodadad Khan, Mohd Anwar etc, "Forecasting Bangladeshi monsoon rainfall using neural network and genetic algorithm approaches", International Technology Management Review", Vol. 2,no 1,2009

[10] M.Nasseri, K.Asghari, M.J.Abeni, "Optimized scenario for rainfall forecasting using genetic algorithm coupled with artificial neural network", Expert Systems with Applications, Vol. 35, pp. 1415-1421, 2008

[11] Auroop R. Gangly, "A Hybrid Approach to Improving Rainfall Forecasts", Computing in Science & Engineering, Vol. 4, No 6, pp. 83-88, 2002

[12] Jatinder N.D.Gupta, Randall S.Sexton, "Comparing backpropagation with a genetic algorithm for neural network training", Omega Vol. 27, pp. 679-684, 1999

[13] David J.Montana and Lawrence Davis, "Training Feedforward Neural Network Using Genetic Algorithms", Proceedings of Eleventh International Joint Conference on AI, pp. 762-767, 1989

[14] K.C.Tripathi, IML Das and A.K.Sahai, "Predictability of Sea surface temperature anomalies in the Indian Ocean using Artificial Neural Network", Indian Journal of Marine Sciences, Vol. 35,No 3,pp. 210-220,2006

[15] S.Rajasekaran and G.A Vijayalakshmi Pai, "Genetic Algorithm based Weight Determination for Back propagation Networks", Proceedings of the Fourth International Conference on Advanced Computing, pp. 73-79, 1996.