International Journal of Computer Applications (0975 – 8887)
Volume 112 – No 4, February 2015

Identification of Global Minima of Back-Propagation Neural Network in the Prediction of Chaotic Motion

Abhishek Shukla
Scholar M. Tech (Project) Computer Science
C. V Raman University, Bilaspur, Chhattisgarh, India.

Sanjeev Karmakar
Bhilai Institute of Technology,
Bhilai House, Durg-491001, Chhattisgarh, India.

ABSTRACT
Modeling through back-propagation neural network to identify internal dynamics of chaotic motion during the prediction is a challenging task still today. While huge number of contributions is found in the literature. However, real applications of it are rarely visible. Two basic shortcomings have been observed. First optimization of its parameters is an effort and second reaching global minima during training period is a temporal timidity. Often these are impractical to achieve. In this study modeling of rainfall data time series (chaos) through back-propagation network is prepared. The parameters are optimized in this application and also obtained global minima. It is found the model reached in its global minima at 900000 epochs. At this point model was finally trained afterward model has shown negative influence. These experimental results are presented in this paper.

Keywords
Back-propagation; neural network; global minima; prediction; chaos; internal dynamics.

1. INTRODUCTION
Chaos is the science of nonlinear and the unpredictable. Chaos Theory deals with nonlinear things that are effectively impossible to predict or control, Its behavior can be determine in many natural system like turbulence, weather, the stock market, our brain states etc. Chaos theory identifies the behavior of dynamic system which is extremely responsive to their initial condition. Karmakar et al., have found that monsoon data series shows chaotic behavior which is very difficult to predict and forecaste due to its chaotic motion. Guhathakurta et al (1999), Rajeevan et al. (2004), Thapliyal & Rajeevan (2003) have found that the identification of internal dynamics of rainfall for long period are approximately difficult [8].

The artificial neural network (ANN) architecture such as back propagation network (BPN) is widely used by the researcher in their application around the world. Since climate and rainfall are highly nonlinear by its nature so that it is very difficult to predict exactly for chaos forecasting.

The training algorithm of BPN model involves four stages, Initialization the weights, feed forward, Back-propagation error, and updating and biases. During the first stage which is the initialization of weights, some small random values are assigned. During feed forward stage each input unit (x_i) receives an input signal and transmits this signal to each of hidden unit $z_i \ldots z_p$. Each hidden unit than calculates the activation function and send its z_j output unit. The output unit calculates the activation function to form the response of the net for the given input pattern. During the back-propagation of errors, each output unit compares its computed activation y_k with its target t_k to determine the associated error for that pattern with the unit. Based on the error, the factor $\delta_k \ (k = 1, .., m)$ is compared and is used to distribute the error at output unit y_k based to all units in the previous layer. Similarly the factor $\delta j \ (j = 1, .., p)$ is computed for each hidden unit z_j. During the final stage, the weight and biases are updates using the δ factor and the activation.

Sivanandam et al., have stated that, In BPN, the weight change is in a direction that is a current gradient and the previous gradient. This approach is beneficial when some training data are very different from a majority of the data. A small learning rate is used to avoid major disruption of the direction of learning when vary unusual pair of training pattern is presented.

The weight update formula for BPN with momentum is,

$$w_{jk}(t+1) = w_{jk}(t) + \alpha\delta_k z_j + \pi[[w_{jk}(t) - w_{jk}(t-1)]$$

$$v_{jk}(t+1) = v_{jk}(t) + \alpha\delta_j x_j + \pi[v_{ij}(t) - v_{ij}(t-1)]$$

μ is called the momentum factor. It ranges from $0<\pi<1$

The application procedure for BPN is shown below:

Step 1: Initialize weights (from training algorithm).
Step 2: For each input vector do step 3-5.
Step 3: For i=1 . . . .n : set activation of input unit x_i.
Step 4: For j=1 . . . .p :

$$z_{-inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$

Step 5: For k=1 . . . .m

$$y_{-inj} = w_{oj} + \sum_{j=1}^{p} z_j w_{jk}$$

$$y_k = f(y_{-ink})$$

1.1 Global Minima and Local Minima
The Back Propagation algorithm or its variation on multilayered feed forward networks is widely used in many applications. However, network starts to train the neurons the rate of MSE decreases and the lowest MSE point is known as local minima, after the point of local minima MSE increases up to some extent after it again decreases rate of MSE goes lowest as compare to local minima, this point where MSE are minimum is known as global minima, after that rate of MSE increases rapidly. Several researchers have introduced several models to find the point of global minima.

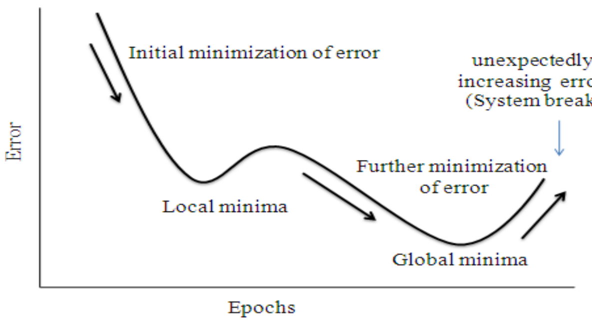

Fig 2 Global Minima and Local Minima during training process.

BPN is one of the architecture in ANN. Experimentally available data is feed as the training pairs to the BPN for learning. The learning rate of BPN may be modified if desired. Accuracy of the system is dependent on the learning rate and Epochs. Epoch is the number of times the loop of code executed to obtain minimum error. Here, we have designed the BPN in such a way that it has a higher learning rate and a greater accuracy. The design incorporates the Sigmoid normalization to convert the application data adaptable to the processing of BPN.

The main objective of this research is to find the point where the rate of MSE is too minimum so called global minima, so with the help of BPN model we can variate the value of learning rate and momentum factor, and will find those value of learning rate and momentum factor where the rate of MSE is too minimum. The value of learning rate and momentum factor may variate from 0-1. To identify the optimum value of 'α' and 'μ', firstly the network is trained with 10^5epochs under different value of 'α' in the close interval 0< α<1 and μ=1. At α=0.7 the convergence of initial weights and minimization of error (i.e., ,mean square error) process is found appropriate. Afterwards to find optimum value of μ, the network was trained again with α = 0.7(fixed) and with different value of μ in the close interval 0 < μ < 1 for 10^5 epochs. It was observed that the convergence of initial weights and minimization of error was appropriate with α=0.7 and μ=0.9. on this optimum value of α and μ the network was trained successfully from local minima of error= 0.00124021664666949 at 10^5 epochs to global minima of error=0.00122414250493303 at 15× 10^5 epochs.

2. THE ANN ARCHITECTURE

The artificial neural network is a computational model based on the structure and functions of biological neural network. An ANN can be defined as a highly connected array of elementary processors called Neurons. A widely used model called the BPN. The BPN type ANN consists of one input layer, one or more hidden layers and one output layer. Each layer employs several neurons and each neuron in a layer is connected to the neurons in the adjacent layer with different weights. Signals flow into the input layer, pass through the hidden layers, and arrive at the output layer. With the exception of the input layer, each neuron receives signals from the neurons of the previous layer linearly weighted by the interconnect values between neurons. The neuron produces its own output signal by passing the signal through a sigmoid activation function.

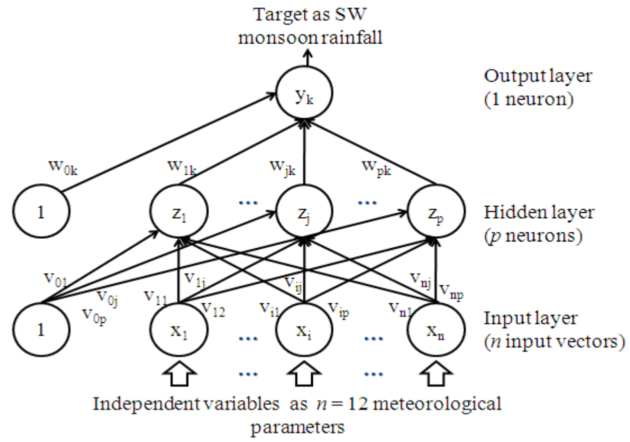

Fig-1. Generalization of BPN for this study.

The BPN uses an algorithm to perform parallel training to improve the efficiency. The algorithm is the most popular, effective, and easiest algorithm to produce a model for MLP's complex network. This algorithm has produced a large class of network types with many diverse topologies and training methods. The algorithm is a supervised learning method that involves backward error correction of the network weights. This algorithm uses a gradient descent method that attempts to minimize the error of the network by moving down the gradient of the error curve. The weights of the network are adjusted by the algorithm. Consequently, the error is reduced along a descent direction.

3. DATA DESCRIPTION AND PROCESSING

It is very difficult to predict the chaotic data time series such as monsoon rainfall. Basu & Andharia (1992) found that the rainfall data time series shows a chaotic behavior with its predictors not only to be chaotic in nature but also suffer from epochal changes [1]. Sixty two years (1951 - 2012) total monsoon rainfall data time series of Ambikapur region in India, which represent chaotic motion is considered for the study. Since BPN system with its transfer function sigmoid is limited to the close intervals 0.and 1 therefore data time series is normalized by using following Equation 1 and used as input to BPN system. Equation 2 is used to de-normalize authentic representation of output (results) in this paper. Data for first 57 years (1951 - 2007) are used for training the BPN and tested for the years 2008 to 2012.

$$r_i = \frac{(x_i + \min(x_i))}{(x_i + \max(x_i))} \quad (1)$$

$$x_i = \frac{\{(\min(x_i) - r_i . \max(x_i))\}}{r_i - 1} \quad (2)$$

4. BPN SYSTEM

In this study BPN in parametric forecast is illustrated in Figure 1, where in 12 input vectors ($x_1, x_2, \ldots, x_{12}$) in input layer are used to observed SW monsoon rainfall data time series (chaos), 3 neurons in hidden layer ($z_1 \ldots z_3$) and one neuron (y_k) in output unit are used to observe 12^{th}year prediction value. Karmakar et al. (2009, 2012) and Kowar et al. (2013) have found that the mean absolute deviation (MAD) is inversely proportional to number of input vector `n' and 11 < n < 15 is found appropriate. Therefore n =12 has been chosen. 11×23=33 hidden layer weights, 03 output layer weights, 03 hidden layer biases, and 01 output layer bias is used in the system to be trained. And these weights v_{ij} 's, w_{ij} 's; v_0, and w_0 (total 40) are trained during the training period. It is observed that one hidden layer is sufficient for all types of chaos, while use of two hidden layers rarely improves the model and it may introduce a greater risk of converging to local minima. One of the key causes is that it increases unknown variables (weights and biases) in the network to be trained. Karmakar et al. (2012) and Kowar et al. (2013) identified that the 3 neurons in hidden layer and 11 input vectors provided satisfactory performance of BPN in deterministic forecast. And further increment of neurons in hidden layer is increases MAD between actual and predicted values. The neurons output is obtained as $f(x_j)$ known as transfer function is typically the sigmoid axon given in the following Equation 3.

$$f(x) = \frac{1}{1 + e^{-\delta x + ή}} \quad (3)$$

Where δ determines the slope and ή is the threshold. In the proposed model δ= 1 and ή =0 have been considered with the output of the neuron in close interval [0, 1] as shown in Figure 2. The BPN in deterministic forecast is trained with 57 years (1951-2007) training dataset. In every epoch (i.e., parallel iteration process) during the training process (Rumelhart et al., 1986) algorithm is used to minimize the error i.e., mean square error (MSE). The training started with initial set of weights and biases between 0 and 1. During the experiments, it was carefully observed that how MSE got optimized regularly after each epoch.

The model acceptance criteria are measured by two statistical identifiers namely: standard deviation (SD) and mean absolute deviation (MAD) given in Equation 6 and 7 respectively with a hypothesis (H). Where 'H' is defined as MAD must incredibly less than or at least half of the SD. If H is true, then model can be accepted otherwise not. The performance criterion is measured by correlation coefficient (CC) between actual and model predicted values. To accept and check performance, the model is analyzed during the training period (1951-2007), and testing period (2008-2012).

5. IDENTIFICATION OF LEARNING RATE AND MOMENTUM FACTOR:

To observe the impacts of changes in the value of `α' and `μ' in the BPN model to identify the internal dynamics of chaotic motion, four experiments were performed with different values `α' and `μ' as follows:

1. Experiment 1 ($0 < \alpha < 1$, μ=1 and 10^5epochs).

2. Experiment 2 (α = 0.8, $0 < \mu < 1$ and 10^5epochs).

3. Experiment 3 (α = 0.8, μ= 0.9, and 5 x 10^5epochs).

5.1 Experiment- 1 ($0 < \alpha < 1$, and 10^5epochs)

Trainable weights of the model are initialized by the random values between 0 and 1.Emphasize is given on the impact of `α' by considering different values of `α', ranging from 0.1 to 0.9 in the model during the training period. For each value of `α' the model is trained with $10^5$epochs repeatedly for 10 times. Finally, their average MSE is analyzed as depicted in Table 1. From the data obtained from such experiment, convergence of the network has been analyzed. It is found to be lowest for α= 0.1 but it is already proved that the lower `α' leads to slower learning process, thus 0.1 cannot be considered as an appropriate value of `α', because the theory of Rumelhart et al. (1986) does not support this value practically and which also may cause slower learning as well as adverse effect to the results discussed by Sivanandam et al. (2006). Figure 3 demonstrated the graphical representation of the same result given in Table 1.Although in the experiment convergence was found slower (i.e., MSE = 0.0050710 9797 967113) at α= 0.8 as compare to at = 0.1 and 0.2.

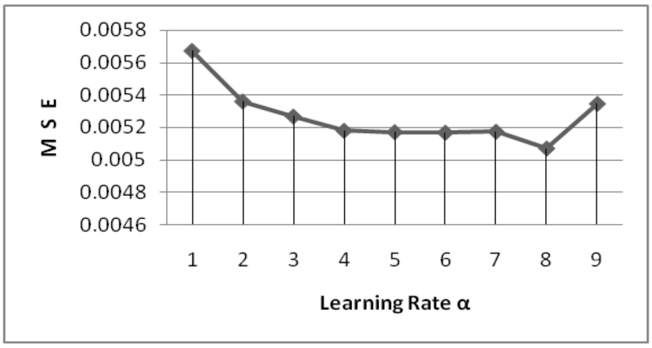

Fig. 2. Minimizing MSE at $0 < \alpha < 1$ and through 10^5epochs

5.2 Experiment-2(α = 0.8, $0 < \mu < 1$ and 10^5epochs)

To identify the impact of `μ' on BPN model, the model is trained with optimum value of `α' i.e., 0.8 and different values of between 0.1 to 0.9 was considered. For each value of `μ' the BPN is trained 10 times for 10^3epochs. Finally their average MSEs are found as given in Figure 3

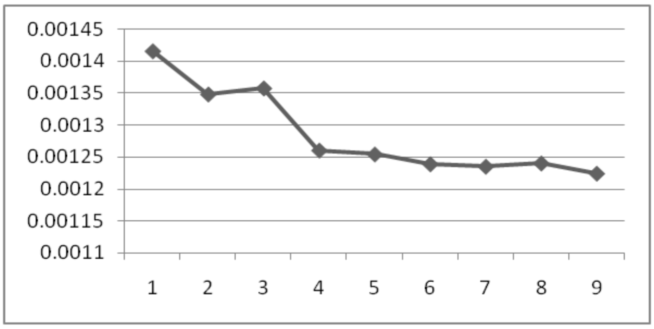

Fig. 3. Minimizing MSE at $0 < \mu < 1$ and through 10^5epochs

Results of above experiment illustrated that the minimum MSE is found at μ=0.9. And increased at= 0.2 to 0.3. At = 0.3 it is decreased. And after that however, the MSE continuously decreasing for μ= 0.4 to 0.9 with slight variation. As in the theory it is clearly mentioned that, in case of higher value of `μ' the weights may oscillate. Therefore, value of μ=0.9 is considered as optimum that the values of α and μ will be 0.8 and 0.9 respectively for further experiments.

5.3 Experiment- 3 (α = 0.8, μ=0.9 and 10^5epochs)

To evaluate and review the impact of variations in α and against the results of last experiment, the same experimental setup with same data set but with different values of α and μ has been repeated. Here, may cause weight changes to be in a direction that would increase the error. Thus the value of μ= 0.9 is considered as appropriate value for training the model which will accelerate the convergence but avoid the increase in error. The training started with initial set of weights between 0 and 1 as shown in Table 6, i.e., after 15x10^5epochs the MSE is minimized up to 4.99180426869658E-04 marked as MG (Global minima) and the optimized weights. The training started with initial set of weights between 0 and 1 at point 'P' where MSE = 0.00154276535844277. After 15x10^5epochs the MSE reached its lowest point at 0.0009003214792487 marked as M_G, the global minima or maximum trained network point as shown in Table 8 and Figure 6. In the previous literatures various authors have clearly mentioned that attaining such point is almost difficult or temporal nervousness. Interestingly, such point has been achieved in the present study. In this experiment MSE is more minimized than that obtained during experiment 3.

Table 1. Optimized MSE

No.	Epoch	MSE
1	10	0.0015427653584427700
2	100	0.0015426806384285000
3	1000	0.0015416193834172800
4	10000	0.0014330470828314100
5	100000	0.0012390158682683100
6	200000	0.0011858370106757200
7	300000	0.0010880326141770200
8	400000	0.0010324735045411800
9	500000	0.0010364698733853100

10	600000	0.0010472180381195800
11	700000	0.0010583641197651100
12	800000	0.0010593281466191800
13	900000	0.0009003214792487000
14	1000000	0.0011791892270713700
15	1100000	0.0012602301371275000
16	1200000	0.0014841074974976200
17	1300000	0.0016211183564409200
18	1400000	0.0018014347130957800
19	1500000	0.0019023238240947800

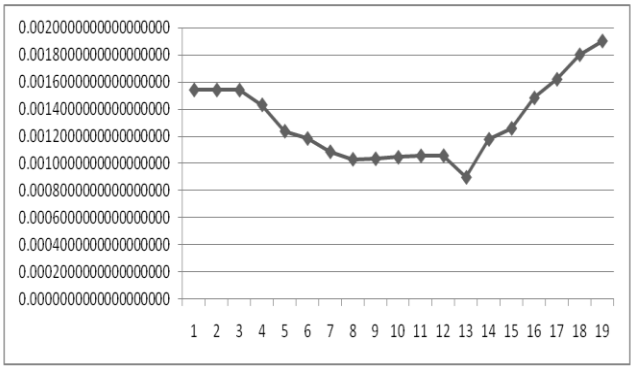

Fig. 3. Optimized MSE by Pictorial Representation

6. RESULTS AND DISCUSSIONS

From the above experiments it is found that the training started in experiment 4 with initial set of weights between 0 and 1 at point 'P' where MSE = 0.00154276535844277. After $15x10^5$ epochs the MSE reached its lowest point at 0.0009003214792487 marked as M_G, the global minima or maximum trained network point. In the previous literatures various authors have clearly mentioned that attaining such point is almost difficult or temporal nervousness. Interestingly, such point has been achieved in the present study. In this experiment MSE is more minimized than that obtained during experiment 3.

So it is found that in overall experiment, BPN model is sufficient to train the network efficiently.

7. CONCLUSIONS

The identification of internal dynamics of chaotic motion and its prediction for future is very difficult. While BPN model is sufficient to overcome such shortcomings, with a proper selection of appropriate parameters is all most importance and a challenging task. These parameters can be optimized by the theory except `α' and `μ'. These two parameters have unusual effects on the performance of BPN model. At the global minima the network was exhibited excellent performance in identification of internal dynamics of chaotic motion and in prediction of future values by past recorded data series.

8. ACKNOWLEDGMENTS

I would like to thank Dr. M.K. Kowar, Director, Bhilai Institute of Technology for providing necessary research lab facility, software and hardware.

9. REFERENCES

[1] Basu, S., Andharia, H. I., 1992. The chaotic time series of Indian monsoon rainfall and its prediction. Proc. Ind. Acad. Sci., 101, 27-34.

[2] Gowariker, V., Thapliyal, V., Sarkar, R. P., Mandal, G. S., Sikka, D. R., 1989: Parametric and power regression models: new approach to long range forecasting of monsoon rainfall in India. Mausam, 40, 115-122.

[3] Gowariker, V., Thapliyal, V., Kulshrestha, S. M., Mandal, G. S., Sen Roy, N. Sikka, D. R., 1991: a power regression model for long range forecast of southwest monsoon rainfall over India. Mausam, 42, 125-130.

[4] Guhathakurta, P., 1998: A neural network model for short term prediction of surface ozone at Pune. Mausam.

[5] Goswami, P., Srividya, 1996: A novel neural network design for long range prediction of rainfall pattern. Curr. Sci., 70, 447-457.

[6] Guhathakurta, P., Thapliyal, V., 1997: A neural network model for long range prediction of monsoon rainfall over India. Mausam.

[7] Shrivastava G., Karmakar S., Kowar M K., 2012:Application of Artificial Neural Networks in Weather Forecasting: A comprehensive literature review, IJCA, Vol-51, No- 18, 17-29.

[8] Rajeevan, M., Pai, D. S., Thapliyal, V., 1998: Spatial and temporal relationships between global land surface air temperature anomalies and Indian summer monsoon rainfall.

[9] Krishnakumar, K., Soman, M. K., Rupakumar, K., 1995: Seasonal forecasting of Indian summer monsoon rainfall: A review. Weather, 150, 449-467.

[10] Sivanandam, S N., Sumathi, S.,Deepa, S N., Introduction to neural Network using MATLAB 6.0, (CHAPTER 8) , TATA McGROW HILL., India.