TIME SERIES FORECASTING USING ARTIFICIAL NEURAL NETWORK WITH EXTENDED ADAPTIVE LEARNING

Mr. Dhanaji Vilas Mirajkar  
Computer Engineering  
Rajarambapu Institute of Technology  
Islampur (Maharashtra), India

ABSTRACT - Artificial neural network (ANN) mainly consists of learning algorithms, which are require to optimize the convergence of neural networks. We need to optimize the convergence of neural networks in order to improve the speed and accuracy of decision making process. To enable the optimization process one of the widely used algorithm is back propagation learning algorithm.

Objective of study is to applied backpropagation algorithm for solving multivariate time series problem. To better the accuracy of neural network it is important to find optimized architecture for the problem under consideration. The learning rate is also an important factor which affects the performance of result. In this study, we proposed extended adaptive learning approach in which learning rate is adapted from number of previous iteration error trend in first half of training. In next half of training learning rate is adapted as per adaptive learning rate algorithm. Compare performance of three variation of backpropagation algorithm. All these variation experimented on two standard dataset. Experimental result shows that during validation and training ANN with extended adaptive learning rate outperforms other than two variations.

Keywords - Artificial Neural Network (ANN), Backpropagation Algorithm, Adaptive Learning Rate, Extended Adaptive Learning Rate, Transfer Function

I. INTRODUCTION

For planning and decision making process forecasting play important role. Time series forecasting means predicting future values using historical data. Time series forecasting has a wide variety of application in many different fields of operation management, marketing, economics, industrial process control and demography. In operation management forecasting approach is used for Control inventories, manage the supply chain, and determine staffing requirements and plan capacity. Prediction is useful for taking marketing decision by finding sales response, advertisement expenditure and product effectiveness. Forecasting is also play important and major role in economics such as prediction of major economic variable, interest rates, inflation (currency growth), job growth, production, and consumption. Forecasts are an integral part of the guidance behind monetary and fiscal policy and budgeting plans and decisions made by governments. Forecasts of population by country and regions are made routinely, often stratified by variables such as gender, age, and race. Demographers also forecast births, deaths, and migration patterns of populations. Governments use these forecasts for planning policy and social service actions, such as spending on health care, retirement programs, and antipoverty programs [1].

The rest of the paper is organized as follows: In Section 2, discussed backgrounds of backpropagation algorithm its drawback. Key points for ANN architecture are described. Methodology and proposed method are explained in Section 3. Section 4 is about discussion of results obtained from ANN implementation and its variations. Finally, the conclusions of our study are outlined in Section 5.

II. BACKGROUND & LITERATURE REVIEW

1.1 Backpropagation Algorithm (BP)

In forecasting backpropagation algorithm is widely used. Some of applications from literature is discussed below. Temperature forecasting is also important issue to protect life and to take agricultural decisions. Ch. Jyosthna Devi et al. done temperature forecasting using ANN by collecting
quantitative data about current state of atmosphere. For learning in ANN author uses backpropagation algorithm [2]. In “Application of Back-Propagation Artificial Neural Network to Predict Maintenance Costs and Budget for University Buildings” paper author predict the maintenance cost using independent variables like age of building, number of floors, and elevator facilities. For this purpose author uses basic backpropagation algorithm [3].

Somnath Mukhopadhyay has proposed that external factors like government involvement; security issues, etc. affected e-commerce growth (and may continue to be). Artificial Neural Networks (ANN or simply NN) is a logical choice for such modelling. ANN has two key advantages over the traditional methods. Because of neural-like connections in the network, this modelling technique is sometimes called a connectionist approach [4].

Working of Backpropagation

BP is a learning algorithm based on gradient descent in which weights are adjusted to reduce the system error. For training BP algorithm work in two stages forward pass and backward pass

In forward input pattern is given to input layer. The input layer passes the pattern activations to the neurons in hidden layer. Output of hidden neuron is input of output layer. The forecasted output neural network is acquired from the activations of the output layer. Error is calculated by actual minus forecasted value. In backward pass error is used to update the weights and it repeated to n number of epochs. As per error and learning rate used weight is updated at hidden layer and output layer. Forward pass and backward pass step is repeated for each training pattern till stopping criteria met. Epoch is an iteration through which the entire training set is trained [5].

Drawback of Backpropagation

Under-Fitting in ANN

If training done by ANN model is very poor means model is unable to find out relationship between the input values and the target value [6].

Over-Fitting in ANN

In this process of overfitting, the performance on the training examples still increases while the performance of validation set becomes worse [6]. Over learning or overfitting occurs when an the algorithm is run for too many epochs or unseen data is very different from training dataset [7].

Authors in [7] try to minimize under-fitting and overfitting in ANN using following strategy, If under-fitting occurs (ANN doesn't attain an adequate performance level) try adding more hidden nodes to the hidden layer(s). If over-fitting occurs (validation error rise) try to minimize hidden layer size

1.2 Key Points for ANN Architecture

ANN accuracy depends on architecture model of neural network. Various researchers’ works on architecture of forecasting model key points for ANN architecture are listed below

No of nodes in hidden layer [8][9][10]

Hidden layer size also affects the performance of ANN. Less number of hidden neuron cause poor training while too may many hidden neurons in hidden layer leads overfitting.

Transfer Function

Each hidden node and output node applies transfer function to input patterns [9]. The selection of transfer functions may strongly impact complexity and performance of neural networks and have been play key role in the convergence of the training algorithms [11].

The learning rate

Backpropagation algorithm performance can be improved by finding optimal learning rate. For a new user selection of optimal learning rate is very challenging. Learning rate is multiplied by a negative gradient to determine the change in the weights and biases. The learning rate is higher, the greater the step . If the learning rate is too large, the algorithm becomes unstable. If the learning rate is too small, the algorithm takes a long time to converge [10].

Learning rate is the basis of a two-layer neural network (NN) of the training process. Therefore, many studies have been done to find the best learning rate, so that the maximum error reduction can be achieved in all iterations. Choose a good learning rate decrease training time, but it may require a lot of trial and error [12].

III. METHODOLOGY & PROPOSED METHOD

a. Implementation Steps:

In order to develop a neural network, we use the backpropagation learning algorithm. We use the following steps to train and validate network.

1) Data Pre-processing:

Neural network training could be made more efficient by performing certain preprocessing steps on the network inputs and targets. Without this standardization, training the neural network will be very slow. There are many types of data normalization. It can be used to scale in the same range for each input feature value data, one feature to another in order to minimize bias neural network. Different techniques can apply different patterns such as max rule, min rule, sum rule, product regulation and hence along [13]. We used Z-score normalization technique which is discussed below

Statistical or Z-Score Normalization
In this technique mean ($\mu$) and standard deviation ($\sigma$) of each feature vector is calculated. The transformation is applied to input vector as per equation 3.1

2) Defining Topology

Input neurons

Input neurons only receive information from predictor or user and pass them to hidden layer. It is dispatcher of information from user to hidden layer. In ANN there is Number of hidden layers in Neural Network

For almost all problems, single hidden layer is satisfactory. Two hidden layers are required when data is like a saw tooth wave form. Using two hidden layers hardly improves the performance. On that point is no theoretical accepting for utilizing more than two hidden layers [9].

Selection of Hidden Layer Size

As per describe earlier there are two types of effect occurred due to the hidden layer size which are overfitting and under-fitting [7]. Selection of proper number of hidden neuron is very important. A researcher gives various methods to find optimal hidden layer size some of them are discussed below.

Forward Approach- This method first selects a small number of hidden neurons. We usually start with two hidden neurons. Later, training and validate the neural network and then increase the number of hidden layer neurons. Repeat the above steps until the training and validation improvement [14].

Backward Approach- This practice is a long-term approach. In this way, we were starting a large bit hidden neurons. Then prepare and validate the NN. Then gradually reduce the number of hidden neurons and retraining and testing NN. Repeat the process until the training and validation results are improved [14].

Rule of thumb method [9][14] - Rule of thumb is to adjust the hidden layer neurons initially, following are rules to set hidden layer size initially:

a) The number of hidden neurons should be in the range between the sizes of the input layer and the size of the output layer

b) The number of hidden neurons should be 2/3 of the input layer size, plus the size of the output layer

c) The number of hidden neurons should be less than twice the input layer size

Output Neurons

Initialize number of output neurons depend on the dataset which is applied.

Transfer function

In this implementation we used tansig transfer function at hidden layer and purlin at output layer.

Training/Learning

$$x' = \frac{x_i-\mu_i}{\sum_{i} \sigma_i} \quad (3.1)$$

Training means updating the synaptic weights of a neural network by considering error at each epoch and learning rate for that epoch .

Performance Measurement:

Performance measurement of ANN in forecasting application is done by RMSE and MAPE (Mean Absolute Percentage Error) [15]. Performance of ANN is calculated in this paper by Root Mean Square Error (RMSE) (also called the root mean square deviation, RMSD) which is calculated by using following formula,

$$RMSE = \sqrt{\frac{\sum (X_{obs,i} - X_{model,i})^2}{n}} \quad (3.2)$$

Another factor to find performance of ANN is Mean Absolute Percentage Accuracy (MAPA) which is calculated considering MAPE and is given by in equation 3.3,

$$MAPA= \frac{100 - \text{(Absolute (Actual– Forecast))}}{\text{Actual}} * 100 \quad (3.3)$$

b. ANN Model

In the paper we use the model shown in Figure 1. Here we change the number of inputs and hidden layer size.

![Figure 1.ANN Model (taken from [16])](image)

Where I: inputs,
WI: weight associated with input neuron and
WO: weight associated with hidden neuron

As an architecture point of view from figure 1 our forecasting ANN model consist of three layer input layer, hidden layer and output layer. For each input layer we assigned weight WI and weight associated with hidden neuron is called WO. Here we used tanh transfer function at hidden layer and purelin to output layer.

Mathematical model used for taking the output form neural network and hidden layer is discussed in following equation. Weight updating for generalization also discussed below which is referred in [16].

Output of Hidden Neuron is,

\[ OHL = \text{Transfer Function} (I^T.WI) \]  
(3.4)

Output of the network is calculated by;

\[ \text{Network Output} = OHL.WO \]  
(3.5)

\[ \text{ERROR} = (\text{Network Output} - \text{Required Output}) \]  
(3.6)

Weight update at Output neuron:

\[ WO = WO - (LR \times \text{ERROR} \times OHL) \]  
(3.7)

Weight update at Hidden neuron:

\[ WI = WI - (LR \times \text{ERROR} \times WO \times (1 - OHL^2)). I^T \]  
(3.8)

Where

LR = Learning Rate  
WI=Weight associated with input neuron  
WO=Weight associated with hidden neuron  
OHL= Output of Hidden Layer

c. ANN with Adaptive Learning Rate

The variable learning rate method is used for the self-adaptive adjustment of learning rate, according to the change of error. The formula below shows the adaptation of learning rate [17]:

\[ a(t+1) = \begin{cases} k_{\text{inc}} a(t) & \text{if } E(t+1) < E(t) \\ k_{\text{dec}} a(t) & \text{if } E(t+1) > E(t) \\ a(t) & \text{Otherwise} \end{cases} \]  
(3.9)

Here \( k_{\text{inc}} \) is learning rate incremental constant which is in between 1.06 to 1.08 and \( k_{\text{dec}} \) learning rate decrement constant which is normally 0.7 . Where \( a \) is learning rate, which incremented or decremented depends on current iteration error \( E(t+1) \) and previous iteration error \( E(t) \). \( t+1 \) = current iteration.

d. Proposed Method (ANN with Extended Adaptive Learning)

We proposed new method for learning rate adaptation is ANN with extended adaptive learning rate (ANN-EALR) in which we find error trend by considering more than 4 previous iteration error and as per equation given in 3.10 we update learning rate. In proposed approach we combine proposed method described in 3.10 with ANN-ALR method described in equation 3.9. We apply proposed method for first half of training phase. In second half of training phase applies the ANN-ALR algorithm.

Mathematical formulation of learning rate updating is given below,

For epochs 0 to epochs/2

\[ \begin{align*}  
  k_{\text{maxinc}} &= \frac{\alpha}{E(t+1) < E(t-cn) \text{ for all } cn = 0 \text{ to } N} \\
  k_{\text{dec}} &= \frac{\alpha}{E(t+1) > E(t-cn) \text{ for all } cn = 0 \text{ to } N} \\
  k_{\text{maxdec}} &= \frac{\alpha}{E(t+1) < E(t-cn) \text{ for all } cn = 0 \text{ to } N} \\
  a(t+1) &= \frac{\alpha}{E(t+1) < E(t-cn) \text{ for all } cn = 0 \text{ to } N} \\
  k_{l1\text{dec}} &= \frac{\alpha}{E(t+1) < E(t-cn) \text{ more than } N/2} \\
  k_{l2\text{dec}} &= \frac{\alpha}{E(t+1) < E(t-cn) \text{ less than } N/2} \\
  \end{align*} \]  
(3.10)

For epochs/2 to epochs

\[ \begin{align*}  
  k_{\text{inc}} &= \frac{\alpha}{E(t+1) < E(t)} \\
  k_{\text{dec}} &= \frac{\alpha}{E(t+1) > E(t)} \\
  a(t+1) &= \frac{\alpha}{E(t+1) > E(t)} \text{ Otherwise} \\
  \end{align*} \]  
(3.9)

Where

\( \alpha = \text{Learning Rate}, E = \text{Error}, t = \text{Iteration} \)

\( N = \text{Number of previous iteration error compared} \)
\( k_{\text{max inc}} \) = Maximum increasing constant (1.09 to 1.2)
\( k_{\text{inc}} \) = Increasing constant (In between 1.06 to 1.8)
\( k_{\text{max dec}} \) = Maximum decreasing constant (0.08 to 0.9)
\( k_{l1 \text{dec}} \) = Maximum decreasing constant (0.06 to 0.079)
\( k_{l2 \text{dec}} \) = Maximum decreasing constant (0.04 to 0.059)

IV. RESULT AND DISCUSSION

For experimentation purpose we use real time multivariate dataset which are discussed below.

Dataset 1

Most important character of a complete solution in a neural network is a data collection. It depends on the pattern of neural networks for data quality, accessibility, reliability and relevance. For the implementation, we utilized information from the literature. On the real data collected from Chandigarh (India), National Bank of India branch basis for three months. Author chose this branch, because there are different types of branches, especially -salary accounts in order to have more data patterns. Data collection time is the 2004 2nd April to 30th June 2004 [18].

Dataset 2

Crop production is a complex phenomenon that is influenced by agro-climatic input parameters. Agriculture input parameters change from field to field and farmer to farmer. Gathering such data on a larger area is a daunting undertaking. All the same, the climatic information collected in India at every 1 sq. m area in different portions of the district is tabulated by the Indian Meteorological Department [19]

a. Artificial Neural Network (ANN)

To test the performance of ANN we vary the number of hidden neurons by keeping constant learning rate 0.2, transfer function tansig-purlin and number of epoch 2000. The following Table 1 shows the performance of neural network by varying the number of hidden neurons by using forward selection approach n Dataset 1.

| Hidden Layer Size | RMSE | MAPA |
|-------------------|------|------|
|                   | Training | Validation | Training | Validation |
| 4                 | 0.8504     | 2.3333       | 88.7205   | 83.2946     |
| 9                 | 0.6788     | 1.3246       | 92.9124   | 86.7309     |
| 14                | 0.6761     | 1.7619       | 93.0060   | 85.4102     |
| 19                | 0.6826     | 2.0577       | 91.3091   | 83.4450     |
| 24                | 0.6601     | 2.4261       | 93.2381   | 82.8933     |
| 29                | 0.6356     | 4.1495       | 93.4902   | 82.2054     |
| 34                | 0.7404     | 3.3200       | 90.1188   | 80.4652     |
| 39                | 0.6493     | 4.7669       | 92.7456   | 69.2925     |

Same experimentation is done on data set 2 and results are tabulated in Table 2. In that we see that training accuracy is 100 percent but overtraining occurs in some cases.

| Hidden Layer Size | RMSE | MAPA |
|-------------------|------|------|
|                   | Training | Validation | Training | Validation |
| 4                 | 15.8791   | 96.5184     | 99.0534   | 90.5605     |
| 9                 | 0.0000    | 99.1723     | 100.0000  | 89.7757     |
| 14                | 0.0000    | 82.9639     | 100.0000  | 91.8808     |
| 19                | 0.0000    | 87.1491     | 100.0000  | 92.1268     |
| 24                | 0.0000    | 75.5362     | 100.0000  | 91.9908     |
| 29                | 0.0000    | 80.7726     | 100.0000  | 92.3562     |
| 34                | 0.0000    | 77.5118     | 100.0000  | 92.1446     |
| 39                | 0.0000    | 127.4094    | 100.0000  | 87.3315     |

From Table 2 for best case RMSE is 77.5118 and accuracy is 92.3562. But performance is greater than traditional ANN but not in such up to mark.

To test the effect of learning rate, we vary the learning rate from 0.05 to 5 by keeping constant transfer function tansig-purlin and number of epochs 2000. The following Table 2
shows the performance of neural network with varying learning rate by using the forward selection approach. From Table 3 we find that 85.9561% accuracy when learning rate is 0.1 for this constant hidden layer size is 13.

Table 3. Effect of Learning Rate on Dataset 1

| Learning Rates | RMSE  | MAPA  |
|----------------|-------|-------|
|                | Training | Validation | Training | Validation |
| 0.05           | 0.6607  | 1.2882  | 91.5092  | 85.4854    |
| 0.1            | **0.6656** | **1.2464** | **91.4464** | **85.9561** |
| 0.15           | 0.6539  | 1.2930  | 91.5961  | 85.4315    |
| 0.2            | 0.6816  | 1.3641  | 91.2404  | 84.6298    |
| 0.25           | 0.6635  | 1.4727  | 91.4727  | 83.4060    |
| 0.3            | 0.6740  | 1.5377  | 91.3378  | 82.6740    |
| 0.35           | 0.6807  | 1.6193  | 91.2514  | 81.7545    |
| 0.4            | 0.5477  | 1.8113  | 93.2025  | 80.0540    |
| 0.45           | 0.6300  | 1.9254  | 92.2789  | 79.1045    |
| 0.5            | 0.6537  | 2.0820  | 91.5768  | 78.3567    |

Same experiment done on dataset 2 and result illustrated on Table 4

Table 4. Effect of Learning Rate on Dataset 1

| Learning Rate | RMSE  | MAPA  |
|---------------|-------|-------|
|               | Training | Validation | Training | Validation |
| 0.05          | 0.0664  | 82.9507  | 99.9959  | 91.2378    |
| 0.1           | 0.0000  | 76.2282  | 100.0000 | 91.7982    |
| 0.15          | 0.0000  | 94.4784  | 100.0000 | 90.1842    |
| 0.2           | 0.0000  | 75.5362  | 100.0000 | 91.9908    |
| 0.25          | 0.0000  | 92.7412  | 100.0000 | 89.9627    |
| 0.3           | 0.0000  | 106.2367 | 100.0000 | 89.4386    |
| 0.35          | 0.0000  | 103.7465 | 100.0000 | 88.9191    |
| 0.4           | 0       | 83.2871  | 100.0000 | 91.2297    |
| 0.45          | 0.0000  | 75.2128  | 100.0000 | 92.6277    |
| 0.5           | 0.0000  | 77.3600  | 100.0000 | 92.8676    |

Figure 2 shows that accuracy percentage is constantly decreasing by increasing learning rate for dataset 1. But for dataset 2 finding appropriate learning rate is difficult.

Table 4. Effect of Learning Rate on Dataset 1

| Learning Rate | RMSE  | MAPA  |
|---------------|-------|-------|
|               | Training | Validation | Training | Validation |
| 0.05          | 1.4306  | 3.9300  | 89.7483  | 73.7637    |
| 0.1           | 0.6187  | 1.0296  | 92.9921  | 88.0601    |
| 0.15          | 0.6126  | 1.0548  | 93.2347  | 87.5138    |
| 0.2           | 0.6213  | 1.0534  | 92.6968  | 87.4215    |
| 0.25          | 0.6174  | **0.9590** | 93.5174  | **88.8470** |
| 0.3           | 0.6135  | 1.0974  | 93.3365  | 87.8620    |
| 0.35          | 0.6163  | 1.0428  | 92.8461  | 87.1492    |
| 0.4           | 0.6041  | 1.0454  | 93.3419  | 87.7128    |

Figure 2. Effect of Learning Rate

b. Artificial Neural Network with Adaptive Learning Rate (ANN-ALR)

We apply the ANN-ALR algorithm which mention in equation 3.9 to dataset 1 and dataset 2 results are illustrated below by varying hidden layer size and by keeping the number of epochs 2000. In this case initially we have given learning rate 0.5.

Table 4 gives performances of ANN –ALR is given for dataset 1. In a best case performance in terms of RMSE is 0.9590 which is less as compared to ANN has1.3246. Mean Absolute Percentage Accuracy for ANN-ALR is 88.8470 which is greater than ANN 86.7309.

Table 5. Performance of ANN-ALR on Dataset 1

| Hidden Layer Size | RMSE  | MAPA  |
|-------------------|-------|-------|
|                   | Training | Validation | Training | Validation |
| 4                 | 1.4306  | 3.9300  | 89.7483  | 73.7637    |
| 9                 | 0.6187  | 1.0296  | 92.9921  | 88.0601    |
| 14                | 0.6126  | 1.0548  | 93.2347  | 87.5138    |
| 19                | 0.6213  | 1.0534  | 92.6968  | 87.4215    |
| 24                | 0.6174  | **0.9590** | 93.5174  | **88.8470** |
| 29                | 0.6135  | 1.0974  | 93.3365  | 87.8620    |
| 34                | 0.6163  | 1.0428  | 92.8461  | 87.1492    |
| 39                | 0.6041  | 1.0454  | 93.3419  | 87.7128    |

In Table 5 gives performance of ANN –ALR is given for dataset 2. In best case performance in terms of RMSE is 74.4263 which is less as compared to ANN. Mean Absolute Percentage Accuracy for ANN-ALR is 93.2082.
Table 6. Performance of ANN-ALR on Dataset 2

| Hidden Layer Size | RMSE Training | Validatio n | MAPA Training | Validatio n |
|------------------|---------------|-------------|---------------|-------------|
| 4                | 47.0217       | 76.5068     | 97.0131       | 89.7828     |
| 9                | 44.2850       | 77.4256     | 97.5287       | 86.7506     |
| 14               | 48.0040       | 85.3845     | 96.9970       | 90.0723     |
| 19               | 41.0187       | 91.4824     | 97.7651       | 93.2085     |
| 24               | 47.0399       | 86.5338     | 97.4454       | 93.2082     |
| 29               | 44.6093       | **74.4263** | 97.5606       | 92.1611     |
| 34               | 46.3399       | 83.1975     | 97.1565       | 90.6447     |
| 39               | 45.1490       | 82.8089     | 97.4039       | 90.0234     |

As we discussed earlier in section 3.4 we combine the two methods described in equation 3.9 and 3.10 apply to dataset 1 and dataset 2. In ANN-EALR we keep same number of epochs taken in ANN-ALR and ANN i.e. 2000. Initially we keep learning rate 0.5 and hidden layer size 4. By forward selection method hidden layer size increases up to 39.

Result is illustrated in following Table 6. For best case RMSE validation is 0.8892 and accuracy is 90.6346%.

Table 7. Performance of ANN-EALR on Dataset 1

| Hidden Layer Size | RMSE Training | Validatio n | MAPA Training | Validatio n |
|------------------|---------------|-------------|---------------|-------------|
| 4                | 0.8107        | 2.4928      | 89.4257       | 78.8574     |
| 9                | 0.6463        | 0.9780      | 93.2789       | 89.5805     |
| 14               | 0.6404        | **0.8892**  | 93.6305       | **90.6346** |
| 19               | 0.6451        | 0.9068      | 93.5007       | 89.7008     |
| 24               | 0.7489        | 1.6900      | 91.8412       | 87.3894     |
| 29               | 0.6167        | 1.0564      | 93.0462       | 88.4241     |
| 34               | 0.6476        | 1.2177      | 92.3184       | 85.5055     |
| 39               | 0.6132        | 1.0547      | 93.2179       | 88.6451     |

We applied it to dataset 2 also and result is mentioned in Table 7. For best case accuracy is 95.2591% and RMSE is 55.8696. ANN, ANN-ALR and ANN-EALR are compared in performance analysis section.

Table 8. Performance of ANN-EALR on Dataset 2

| Hidden Layer Size | RMSE Training | Validatio n | MAPA Training | Validatio n |
|------------------|---------------|-------------|---------------|-------------|
| 4                | 42.6715       | 77.9943     | 97.0943       | 92.7469     |
| 9                | 4.5194        | 70.0866     | 99.6327       | 93.2079     |
| 14               | 4.5366        | 67.1637     | 99.7248       | 93.0404     |
| 19               | 0.2336        | 60.1889     | 99.9871       | 94.6249     |
| 24               | 6.7803        | **55.8696** | 99.5007       | **95.2591** |
| 29               | 6.0797        | 77.4789     | 99.5208       | 92.4711     |
| 34               | 4.2169        | 63.0233     | 99.6924       | 94.5567     |
| 39               | 5.9523        | 56.2099     | 99.5653       | 95.0748     |

c. Proposed Method - ANN with Extended Adaptive Learning Rate (ANN-EALR)

Table 7. Performance of ANN-EALR on Dataset 1

| Hidden Layer Size | RMSE | MAPA |
|------------------|------|------|
|                  |      |      |
| 4                | 0.8107 | 89.4257 |
| 9                | 0.6463 | 93.2789 |
| 14               | 0.6404 | 93.6305 |
| 19               | 0.6451 | 93.5007 |
| 24               | 0.7489 | 91.8412 |
| 29               | 0.6167 | 93.0462 |
| 34               | 0.6476 | 92.3184 |
| 39               | 0.6132 | 93.2179 |

As we discussed earlier in section 3.4 we combine the two methods described in equation 3.9 and 3.10 apply to dataset 1 and dataset 2. In ANN-EALR we keep same number of epochs taken in ANN-ALR and ANN i.e. 2000. Initially we keep learning rate 0.5 and hidden layer size 4. By forward selection method hidden layer size increases up to 39.

Result is illustrated in following Table 6. For best case RMSE validation is 0.8892 and accuracy is 90.6346%.

d. Comparison

We compare the performance of all variations of ANN. In which we find that for all cases Performance of ANN-EALR has better performance than the ANN-ALR and ANN. For comparison purpose we kept the number of epochs 2000 and hidden layer size 4 to 39.

Comparison with respect to RMSE is tabulated in Table 8. Comparison with respect to MAPA is tabulated in Table 9.

For best case ANN-EALR has RMSE is 0.8892 which is less than traditional ANN and ANN-ALR. Initially for less number of hidden layer size under-fitting occurs shown in first row of Table 8.

Table 9. RMSE Performance comparison on Dataset 1

| Hidden Layer Size | ANN Validation | ANN-ALR Validation | ANN-EALR Validation |
|------------------|----------------|---------------------|---------------------|
| 4                | 2.3333         | 3.9300              | 2.4928              |
| 9                | 1.3246         | 1.0296              | 0.9780              |
| 14               | 1.7619         | 1.0548              | **0.8892**          |
| 19               | 2.0577         | 1.0534              | 0.9068              |
| 24               | 2.4261         | **0.9590**          | 1.6900              |
| 29               | 4.1495         | 1.0974              | 1.0564              |
| 34               | 3.3200         | 1.0428              | 1.2177              |
| 39               | 4.7669         | 1.0454              | 1.0547              |
To performance measurement in MAPA for best case ANN-EALR has accuracy 90.6346 which is highest than other two approach.

**Table 10. MAPA Performance comparison on Dataset 1**

| Hidden Layer Size | MAPA Dataset 1 | |
|-------------------|----------------|---|
|                   | ANN Validation | ANN-ALR Validation | ANN-EALR Validation |
| 4                 | 83.2946        | 73.7637             | 78.8574              |
| 9                 | **86.7309**    | 88.0601             | 89.5805              |
| 14                | 85.4102        | 87.5248             | **90.6346**          |
| 19                | 83.445         | 87.4215             | 89.7008              |
| 24                | 82.8933        | **88.847**          | 87.3894              |
| 29                | 82.2054        | 87.862              | 88.4241              |
| 34                | 80.4652        | 87.1492             | 85.5055              |
| 39                | 69.2925        | 87.7128             | 88.6451              |

In Figure 3 is drawn MAPS versus hidden layer size for all ANN variations which shows that ANN-EALR is give best performance in all cases.

**Table 11. MAPA Performance comparison on Dataset 2**

| Hidden Layer Size | MAPA Dataset 2 | |
|-------------------|----------------|---|
|                   | ANN Validation | ANN-ALR Validation | ANN-EALR Validation |
| 4                 | 90.5065        | 89.7828             | 92.7469              |
| 9                 | 89.7757        | 86.7506             | 93.2079              |
| 14                | 91.8808        | 90.0723             | 93.0404              |
| 19                | 92.1268        | **93.2085**         | 95.0748              |
| 24                | 91.9908        | 93.2082             | **95.2591**          |
| 29                | **92.3562**    | 92.1611             | 92.4711              |
| 34                | 92.1446        | 90.6447             | 94.5567              |
| 39                | 87.3315        | 90.0234             | 94.0748              |

In Figure 4 is drawn MAPA versus hidden layer size for all ANN variations which shows that ANN-EALR is give best performance in all cases. Performance of ANN-EALR is indicated as continuous red line which is always above the other two approaches shown in Figure 4.

**Table 10 RMSE Performance comparison on Dataset 2**

| Hidden Layer Size | RMSE Dataset 2 | |
|-------------------|----------------|---|
|                   | ANN Validation | ANN-ALR Validation | ANN-EALR Validation |
| 4                 | 96.5184        | 97.4773             | 77.9943              |
| 9                 | 99.1723        | 124.7               | 70.0866              |
For best cases Actual vs Forecasted result for Dataset 1 considering all approaches presented in Figure 5.

![Figure 5. Actual Versus Forecasted on Dataset 1](image)

For best cases Actual vs Forecasted result for Dataset 2 considering all approaches presented in Figure 6.

![Figure 6. Actual Versus Forecasted on Dataset 2](image)

V. CONCLUSION

Performance of neural network is depending on learning rate, number of hidden layer size. We cannot kept same hidden layer size and learning rate for all the data set patterns. The issue of the amount of hidden node and learning rate is worked out in this study. To see out the optimum hidden layer size we used forward selection approach. In this case the error is decreased by increasing number of hidden layers up to a certain level then the error goes on increasing.

Adaptive learning is best choice to train all data patterns. One of the methods to enhance performance of ANN is adaptive learning rate by comparing previous iteration error. Adaptive learning is also merging with extended adaptive learning in which for adaptation of learning rate is done by comparing n number of previous iterations error. As per Error trend we update the learning rate given in equation 3.10. For dataset 1 ANN -EALR gives 90.6346 percent accuracy while ANN and ANN-ALR gives 86.7309 and 88.847 respectively. After applying dataset 2 ANN-EALR outperforms in all cases. For best case ANN-EALR has accuracy of 95.2591 which is better than other two variations.

In future ANN is also hybridized with GA to find automatic ANN architecture parameters to get optimum result.

VI. REFERENCES

[1] Montgomery, D. C., Jennings, C. L., & Kulaheci, M. (2015). Introduction to time series analysis and forecasting. John Wiley & Sons.

[2] Devi, C. J., Reddy, B. S. P., Kumar, K. V., Reddy, B. M., & Nayak, N. R. (2012). ANN approach for weather prediction using back propagation. International Journal of Engineering Trends and Technology, 3(1), 19-22.

[3] Zorn, D. M. (2011). Enactive education: Dynamic co-emergence, complexity, experience, and the embodied mind (Doctoral dissertation, University of Toronto).

[4] Mukhopadhyay, S., Samadder, S., & Nargundkar, S. (2008). Predicting Electronic Commerce Growth: An Integration of Diffusion and Neural Network Models. Journal of Electronic Commerce Research, 9(4), 280.

[5] Afzal, S., & Wani, M. A. (2015). Comparative Study of Adaptive Learning Rate with Momentum and Resilient Back Propagation Algorithms for Neural Net Classifier Optimization. International Journal of Distributed and Cloud Computing, 2(1).

[6] http://cs231n.github.io/neural-networks-3/

[7] Panchal, Gaurang, et al. "Determination of over-learning and over-fitting problem in back propagation neural network." International Journal on Soft Computing 2.2 (2011): 40-51.

[8] International journal of forecasting, 14(1), 35-62. Panchal, G., Ganatra, A., Kosta, Y. P., & Panchal, D. (2011). Behaviour analysis of multilayer perceptron’s with multiple hidden neurons and hidden layers. International Journal of Computer Theory and Engineering, 3(2), 332.

[9] Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art.

[10] Lahiri, S. K., & Ghanta, K. C. (2009). Artificial neural network model with the parameter tuning assisted by a differential evolution technique: The study of the holdup of the slurry flow in a pipeline. Chemical Industry and Chemical Engineering Quarterly/CICEQ. 15(2), 103-117.
[11] Saduf, M. A. W. (2013). Comparative study of back propagation learning algorithms for neural networks. International Journal of Advanced Research in Computer Science and Software Engineering, 3(2), 1151-6.

[12] Thota, L. S., & Changalasetty, S. B. (2013). Optimum learning rate for classification problem with mlp in data mining. Advances in Engineering and Technology, 6(1), 35-44.

[13] Pankaj Sharma, Naveen Malik, Naeem Akhtar, Rahul, and Hardeep Rohilla (2013). Feed Forward Network Review,” International Journal of Advanced Research in Engineering and Applied Sciences, Vol. 2 | No. 10.

[14] Panchal, F. S., & Panchal, M. (2014). Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network. International Journal of Computer Science and Mobile Computing, 3(11), 455-464.

[15] Hyndman, R. J., & Athanasopoulos, G. (2014). Forecasting: principles and practice. OTexts.

[16] http://www.philbrierley.com/main.html?code/matlab.html&code/codeleft.html

[17] Li, J., Cheng, J. H., Shi, J. Y., & Huang, F. (2012). Brief introduction of back propagation (BP) neural network algorithm and its improvement. In Advances in Computer Science and Information Engineering (pp. 553-558). Springer Berlin Heidelberg.

[18] Kumar, P., & Walia, E. (2006). Cash Forecasting: An Application of Artificial Neural Networks in Finance. IJCSA, 3(1), 61-77.

[19] Veenadhari, S., Misra, B., & Singh, C. (2014, January). Machine learning approach for forecasting crop yield based on climatic parameters. In Computer Communication and Informatics (ICCCI), 2014 International Conference on (pp. 1-5). IEEE.