INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Stock Price Prediction using Artificial Neural Network

Omprakash Barapatre
omprakashbarapatre@bitraipur.ac.in
Bhilai Institute of Technology Raipur, Chhattisgarh

Eric Tete
erictete1995@gmail.com
Bhilai Institute of Technology Raipur, Chhattisgarh

Chandan Lal Sahu
chandansahu1303@gmail.com
Bhilai Institute of Technology, Raipur, Chhattisgarh

Domesh Kumar
Domeshnirmalkar8827@gmail.com
Bhilai Institute of Technology Raipur, Chhattisgarh

Hira Kshatriya
hira.cool007@gmail.com
Bhilai Institute of Technology Raipur, Chhattisgarh

ABSTRACT

Stock market prediction is the art of determining the future value of a company stocks. This paper proposes a machine learning (ANN) artificial neural network model to predict stock market price. The proposed algorithm integrates the back – propagation algorithm. The backpropagation algorithm is employed here to training the ANN network model. And in this paper, we have done a research on the TESLA dataset.

Keyword: *Back-Propagation, Artificial Neural Network, Stock-Market Prediction.*

1. INTRODUCTION

In the past few decades, prediction of the stock price is gaining more attention as the profitability of the investors in the stock market is mainly depends on the predictability. If the direction of the market is successfully predicted then the investor can yield enough profit. For solving the kind of financial problem the relationship between the input and output is very complex so that's why we have used ANN for solving or predicting the stock price.

An artificial neural network model is computer model whose architecture essentially mimics the learning capability of the human brain. The processing element of artificial neural network resembles the biological structure of the neuron and the internal operation of the human brain.[5]

In this paper, Multilayer feed forward back propagation neural network is used for the prediction purpose.[2] Feed forward neural network is a unidirectional connection between the neurons that means the information can flow only in forward direction. Here there is no connection between the neurons present in the same layer. Input has been fed into the first layer and with the help of hidden layers connected to the last layer that produces the output. And since all of the information is constantly feeding forward from one layer to the next hence it is called feed forward network.

One of the learning methods in multilayer Perceptron Neural Networks is the error back propagation in which the network learns the pattern in the dataset and justifies the weight of the connections in the inverse direction respect to the gradient vector of error function which is usually regularized sum of square error.[9] The backpropagation method picks a training vector from training data set and moves it from the input layer toward the output layer. In the output layer, the error is calculated and propagated backward so the weight of the connection will be corrected. This will usually go on until the error reaches a pre-defined value. It's proved that we can approximate any continuous function with a three-layer feedback network with any precision. It should be said that the learning speed will dramatically decrease according to the increase of the number of neurons and layer of the network.

1.1. Multilayer feed forward perceptron

In this paper, we have used multilayer feedforward perceptron below figure illustrates the how the multilayer feed forward perceptron looks like.

Multilayer:[3][4]In multilayer neural network what happened there is a number of hidden layers are available in between the input layer and output layer called as a hidden layer so in the multilayer perceptron neural network more than one hidden layer is available.

Feedforward: in feed forward neural network what happened there is no edges are available in between the neurons present in the same layer here the Synaptics are available in between only the neuron present in the different layer of the neural network.

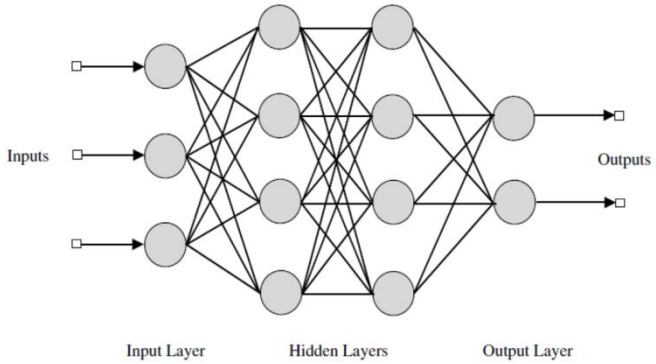

Fig 1: Feed forward Multilayer Perceptron

2. ALGORITHMS

Backpropagation algorithm

The backpropagation algorithm falls into the general category of gradient descent algorithm. Purpose of gradient descent algorithm is to find the minima and maxima of a function by iteratively moving in the direction of negative slope of the fun
ction that we want to minimize or maximize.[6]

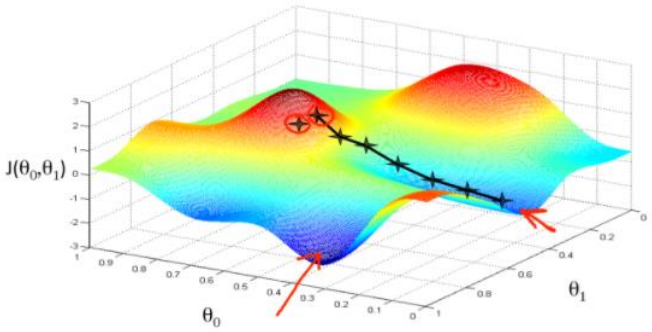

Fig 2: Gradient Descent

In backpropogation algorithm the network is trained by repeatedly processing the training data set and comparing the network output with the actual output if anything differs between the what output we are getting from our network and the desired output then this is called as error so after that what we are doing is we are again back propagate the whatever the output we got from our neural network to the back layer of the neural network till first hidden layer so that again our network will assign different weight randomly to the synaptic and try to generate the output with lesser error from the previous time and at every iteration our network output should be less from the output of previous time so that after some iteration we should get desire output and here we are keeping back propagate the output from our neural network till our network trained completely. This below flowchart shows how backpropagation algorithm will work.[10]

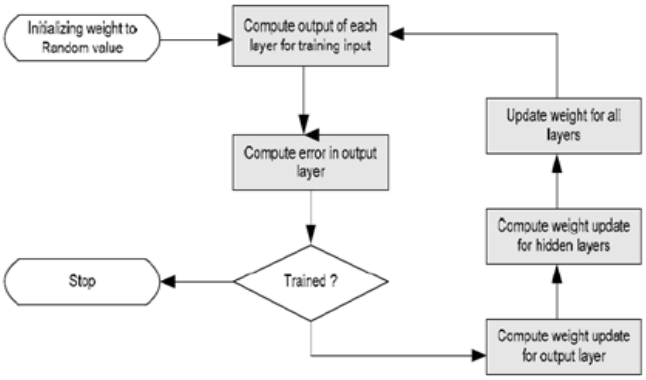

Fig 3 flowchart of backpropagation algorithm

Mathematical derivation of backpropagation algorithm

In backpropagation our main intension is to find out the changes is supposed to happen on the weight assigned to the synaptic when the output we are getting from our network model is again backpropagated to the again first hidden layer so in this computation our main intension is to calculate Δw_{ij}.

The first equation is about finding out the error generated from the neural network.

$$E_j(n) = d_j(n) - y_j(n) \quad ..equ(1)$$

The above equation shows the error generated from our network and it can be calculated by just taking the difference of desired output and output we are getting from out neural network. Here the term $d_j(n)$ will represent the desired output we are getting from out neural network and the term $y_j(n)$ is the output we got from our neural network. After that, we found out the square error by just doing this.

$$E(n) = \tfrac{1}{2} \sum_{j \epsilon C} e_j^2(n) \quad equ(2)$$

The above equation shows the error energy here we are actually intended to calculate the square energy by just doing a square of all of the error and summing up the squared error we are getting from all of the neurons present in the output layer. Now we will find the average error energy.

$$E_{avg}(n) = 1/N \sum_{n=1}^{N} E(n) \quad equ(3)$$

In above equation, N is the number of iteration and we are summing up all the error energy from 1st iteration to N iteration for calculating the average error energy. Now we will calculate the value for the induced field.

$$V_j(n) = \sum_{i=1}^{m} W_{ij}(n) * Y_{ij}(n) \quad equ(4)$$

Local induced field value can be calculated by just doing summation of multiplication of $W_{ij}(n)$ and $Y_{ij}(n)$ from number neurons in the previous layer where is represented the number of neuron in previous layer and the value of Y_{ij} can be calculated by applying the activation function over induced field of J^{th} layer neuron below equation will illustrate this.

$$Y_j(n) = \phi(V_j(n)) \quad equ(5)$$

Now we are intended to find out the is change is happened in error with respect to change in the weight by just applying the chain rule of differentiation.

$$\frac{dE(n)}{dWij(n)} = \frac{dE(n)}{dej(n)} \cdot \frac{dej(n)}{dyj(n)} \cdot \frac{dyj(n)}{dvj(n)} \cdot \frac{dvj(n)}{dWij(n)} \quad equ(6)$$

Here From equ(2) $\frac{dE(n)}{dej(n)} = e_j(n)$

From equ(1) $\frac{dej(n)}{dyj(n)} = -1$

From equ(5) $\frac{dyj(n)}{dvj(n)} = \phi'(V_j(n))$

From equ(4) $\frac{dvj(n)}{dWij(n)} = Y_j(n)$

By putting this all of the value in equation 6 we got the following result:

$\frac{dE(n)}{dWij(n)} = -e_j(n).\ \phi'(V_j(n)).\ Y_j(n)$ ..equ(7)

Δw_{ij} is applied to the W_{ij} and which is proportional to the $\frac{dE(n)}{dWij(n)}$ so according to the definition of proportionality we can write as.

$\Delta W_{ij} = -\eta \frac{dE(n)}{dWij(n)}$ equ(8)

In above equation, η is constant of proportionality and the value of η is 9.25.from the equation (7) we can write equation (8) as.

$\Delta W_{ij} = \eta\ e_j(n).\ \phi'(V_j(n)).\ Y_j(n)$

$\delta_j(n) = e_j(n).\ \phi'(V_j(n)).$ equ(9)

$\Delta W_{ij} = \eta\ \delta_j(n)\ Y_j(n)$ equ(10)

The below figure shows the signal flow of backpropagation algorithm.

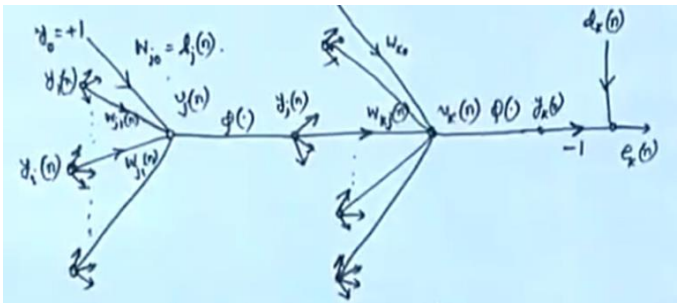

Fig4: Signal Flow of Back propagation Algorithm

Now we can find out the δ_j,

$$\delta_j = -\frac{dE(n)}{dvj(n)}$$

$$= -\frac{dE(n)}{dyj(n)} \cdot \frac{dyj(n)}{dvj(n)}$$

$$= -\frac{dE(n)}{dyj(n)} \cdot \phi'(Vj(n))$$

Now as above figure if we consider k as output layer and j as preceding layer then calculation might looks likes:

$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n)$ equ(11)

$\frac{dE(n)}{dyj(n)} = \sum_k e_k(n) \cdot \frac{dek(n)}{dyj(n)}$ equ(12)

$\frac{dE(n)}{dyj(n)} = \sum_k e_k(n) \cdot \frac{dek(n)}{dvk(n)} \cdot \frac{dvk(n)}{dyj(n)}$ equ(12)

$e_k(n) = d_k(n) - y_k(n)$

$= d_k(n) - \phi(V_k(n))$ equ(13)

$\frac{dek(n)}{dvk(n)} = -\phi(Vk(n))$ equ(14)

For neuron k,

$V_k(n) = \sum^{m}_{j=0} W_{kj}(n) * Y_j(n)$ equ(15)

$\frac{dvk(n)}{dyj(n)} = W_{kj}(n)$ equ(16)

So now equation 12 can be written as

$$\frac{dE(n)}{dyj(n)} = -\sum_k e_k(n).\, \phi(Vk(n)).W_{kj}(n)$$

$$= -\sum_k \delta_k(n) .W_{kj}(n)$$

$\delta_k(n) = e_k(n)\ \phi'(Vk(n))$

so if we talk about backpropogation algorithm in a nutshell then It can be represented like this:

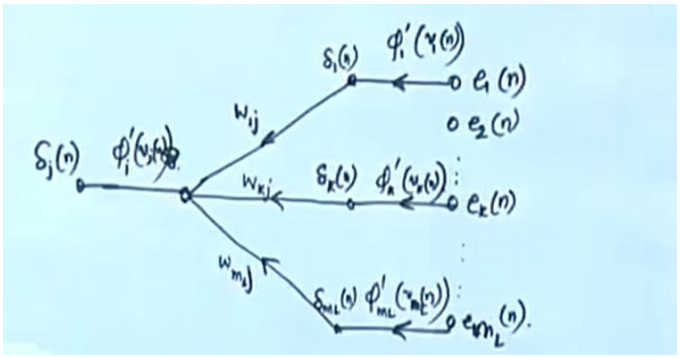

Fig5: Back propagation Algorithm in Nutshell

3. METHODOLOGY

We will be using stock market data to predict closing price the workflow for the general neural network design has five primary steps:[1]

a) Data collection and preparation
b) Network creation
c) Training the network
d) Validating the network
e) Using the network

a) Data collection and preparation

Data collection is the primary step and it is necessary in order to train, validate and test the neural network.

Date	Open	High	Low	Close	Adj Close	Volume
2010-06-29	19	25	17.540001	23.889999	23.889999	18766300
2010-06-30	25.790001	30.42	23.299999	23.83	23.83	17187100
2010-07-01	25	25.92	20.27	21.959999	21.959999	8218800
2010-07-02	23	23.1	18.709999	19.200001	19.200001	5139800
2010-07-06	20	20	15.83	16.110001	16.110001	6866900
2010-07-07	16.4	16.629999	14.98	15.8	15.8	6921700
2010-07-08	16.139999	17.52	15.57	17.459999	17.459999	7711400
2010-07-09	17.58	17.9	16.549999	17.4	17.4	4050600
2010-07-12	17.950001	18.07	17	17.049999	17.049999	2202500
2010-07-13	17.389999	18.639999	16.9	18.139999	18.139999	2680100
2010-07-14	17.940001	20.15	17.76	19.84	19.84	4195200
2010-07-15	19.940001	21.5	19	19.889999	19.889999	3739800
2010-07-16	20.700001	21.299999	20.049999	20.639999	20.639999	2621300
2010-07-19	21.370001	22.25	20.92	21.91	21.91	2486500
2010-07-20	21.85	21.85	20.049999	20.299999	20.299999	1825300
2010-07-21	20.66	20.9	19.5	20.219999	20.219999	1252500

Fig 6 : Tesla Data Set

For collection the data google finance has been used for collection the historical stock price details of any one of the company. This all of the data set are then feded into the network.

b) Network creation

After collecting the all of the data set the next hectic task is to create your neural network model here the selection of what type of

neural network is going to be a difficult task. Neural network model like supervised or unsupervised and single layer or multilayer you should choose any one of those which is appropriate to solve your problem in our case it is supervise and multilayer perceptron.

c) Training the network

as we are going to solve the problem with the help of artificial neural network then actually what we are trying to do is we are actually mimicking the functionality of biological neural to some extent since there is requirement of training your brain here also there is requirement of training your network for doing the task by itself once the your network trained this will perform every task correctly so for training the network there is lots of way but in our case we have used backpropogation algorithm because the efficiency of backpropogation algorithm is high very high as we are doing back propagate error.

d) Validating the network

Once the training has done the network is validated using the validated data to enhance the performance of the network.

e) Using the network

Once the network is optimized. It has been tested using the test data. In our case collected data of TESLA has been used to predict the adjusted closing price of the stock.

Some of the screenshot our result is shown below:

Date	Open	High	Low	Close	Volume	Ex-Dividend
2010-06-29	19.00	25.0000	17.54	23.89	18766300.0	0.0
2010-06-30	25.79	30.4192	23.30	23.83	17187100.0	0.0
2010-07-01	25.00	25.9200	20.27	21.96	8218800.0	0.0
2010-07-02	23.00	23.1000	18.71	19.20	5139800.0	0.0
2010-07-06	20.00	20.0000	15.83	16.11	6866900.0	0.0

Date	Split Ratio	Adj. Open	Adj. High	Adj. Low	Adj. Close
2010-06-29	1.0	19.00	25.0000	17.54	23.89
2010-06-30	1.0	25.79	30.4192	23.30	23.83
2010-07-01	1.0	25.00	25.9200	20.27	21.96
2010-07-02	1.0	23.00	23.1000	18.71	19.20
2010-07-06	1.0	20.00	20.0000	15.83	16.11

Date	Adj. Volume	100ma
2010-06-29	18766300.0	23.890000
2010-06-30	17187100.0	23.860000
2010-07-01	8218800.0	23.226667
2010-07-02	5139800.0	22.220000
2010-07-06	6866900.0	20.998000

Accuracy: 0.9395375335435104

Fig 7: Stock Data Set

Fig 8: Closing and Moving Average

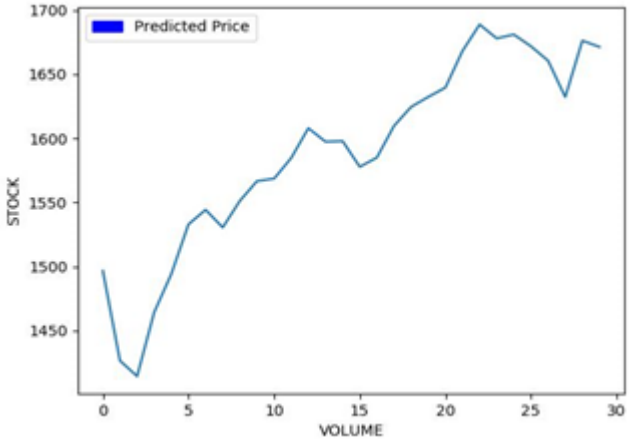

Fig 9: Predicted Price

4. CONCLUSION

In these paper, we used neural networks model to predict the value of stock share by making the use of historical details of stock market value. For this purpose multilayer feed forward network are applied to solve the problem. The obtained result shows that for predicting the direction of the changes of the value of stock none of the method are better than backpropogation algorithm as we got 94% accuracy.

5. REFERENCES

[1] Palavi Ranka "*Stock Market Prediction using Artificial Neural Networks* " IME611 - Financial Engineering Indian Institute of Technology, Kanpur (208016), India
[2] Rosenblatt, F. "The *Perceptron: A Perceiving and Recognizing Automaton."* Cornell Univ. Ithaca. NY. Project PARA Cornell Aeronaut Lab. Rep (1997) 85-460
[3] Mandic, D. P. and Chambers, J." *A. (2001) Recurrent Neural Networks for Prediction"*, John Wiley & Sons, LTD.
[4] Lippmann, R.P., 1987. *"An introduction to computing with neural nets"*. IEEE Accost. Speech Signal Process. Mag., April: 4-22.
[5] Lee Jacobson , *"Introduction to Artificial Neural Networks Part 2 – Learning"*, 26th March 2014
[6] "*Theory of Backpropagation in neural network"* by Robert Hecht Nielson
[7] Hakob GRIGORYAN Bucharest University of Economic Studies, Bucharest, Romania "*Stock Market Prediction using Artificial Neural Networks."* Case Study of TAL1T, Nasdaq OMX Baltic Stock.
[8] iJADE Stock Advisor: "*An Intelligent Agent Based Stock*
Prediction System Using Hybrid RBF Recurrent Network"
Raymond S. T. Lee
[9] "*Make your own neural network"* by Tariq Rashid
[10] "Neural network for pattern recongnistation "By Christoper M.Bishop

