STOCK MARKET PREDICTION
Using Back propagation Algorithm

SLOT: B2+TB2

FINAL REVIEW

GROUP MEMBERS

V SAI NIKHIL	16MIS0257
K V KARTHIK	16MIS0327
M ANITH	16MIS0126
P JOHN PAUL	16MIS0375

FACULTY: PROF.SENTHIL KUMAR P

ABSTRACT:

The aim of this project is implementation of neural networks with back propagation algorithm for stock market. Borrowing from biology, researchers are exploring neural networks, no algorithmic approach to information processing. A neural network is a powerful data-modelling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform “intelligent" tasks similar to those performed by the human brain. Artificial Neural Networks are being counted as the wave of the future in computing. They are indeed self-learning mechanisms which don’t require the traditional skills of a programmer. Back propagation is one of the approaches to implement concept of neural networks.

Back propagation is a form of supervised learning for multi-layer nets. Error data at the output layer is back propagated to earlier ones, allowing incoming weights to these layers to be updated. It is most often used as training algorithm in current neural network applications. In this paper, we apply data mining technology to stock market in order to research the trend of price; it aims to predict the future trend of the stock market and the fluctuation of price. This paper points out the shortage that exists in current traditional statistical analysis in the stock, then makes use of BP neural network algorithm to predict the stock market by establishing a three-tier structure of the neural network, namely input layer, hidden layer and output layer. Finally, we get a better predictive model to improve forecast accuracy.

In the last few years, machine learning has become a very popular tool for analysing financial text data, with many promising results in stock price forecasting from financial news, a development with implications for the Ecient Markets Hypothesis (EMH) that underpins much economic theory. In this work, we explore recurrent neural networks with character-level language model pre-training for both intraday and interlay stock market forecasting. In terms of predicting directional changes in the Standard & Poor’s 500 index, both for individual companies and the overall index, we show that this technique is competitive with other state-of-the-art approaches.

KEYWORDS:

Artificial Neural Network, Back propagation, Data mining, Deep learning, LSTM

INTRODUCTION:

The stock market is affected by a large number of highly interrelated economical, political and psychological factors which interact with each other in a complex fashion. Since most of these relationships seem to be probabilistic and therefore cannot be expressed as deterministic rules, financial analysis is one of the most well suited and promising applications of artificial neural networks. The proposals have been made to use neural network models for prediction and forecasting problems in the financial area, such as locating sources of forecast uncertainty in a recurrent gas market model, corporate bond rating , mortgage delinquency prediction, chaotic time series prediction , prediction of IBM daily stock prices , prediction of three selected German stock prices and prediction of the weekly Standard Poor 500 index . In some of these proposals, the neural networks performed better than regression techniques or as good as the Box-Jenkins technique , while in others the results were disappointing.

In this project, we apply several alternatives of the backpropagation neural network. The basic idea is to let the network learn an approximation of the mapping between the input and output data in order to discover the accurate output. The trained network is then used to predict the weekly closing prices for the future.

Our work differs from previous stock price predictions with neural techniques [5] in the data presented to the networks. While in other approaches the input data was exclusively based on stock prices, we also consider other important economical factors, namely a subset of those considered in the fundamental and technical analysis methods used by human analysts to make their investment decisions. Thus, although we regard a neural network for stock price prediction primarily as a technical analysis tool, elements of the fundamental and the technical analysis are combined in our approach. Similar to a human analyst who is probably more successful if he or she is aware of both methods, we expect the networks to produce high quality predictions in the combined approach. Several simulation results will be presented in order to see if our expectations will be fulfilled.

Another strong trend in deep learning for text is the use of a word embedding layer as the main representation of the text. While this approach has notable advantages, word-level language models do not capture sub-word information, may inaccurately estimate embedding for rare words, and can poorly represent domains with long-tailed frequency distributions. These were motivations for character level language models showed are capable of learning high level representations despite their simplicity. These motivations seem applicable in our domain: character-level representations can for example generalise across numerical data like percentages (e.g. the terms 5% and 9%) and currency, and can handle the large number of infrequently mentioned named entities. Character level models are also typically much more compact. In this work we propose an automated trading system that, given the release of news information about a company, predicts changes in stock prices. The system is trained to predict both changes in the stock price of the company mentioned in the news article and in the corresponding stock exchange index (S&P 500). We also test this system for both intraday changes, considering a window of one hour after the release of the news, and for changes between the closing price of the current trading session and the closing price of the next day session. This comparative analysis allow us to infer whether the incorporation of new information is instantaneous or if it occurs gradually over time. Our model consists of a recurrent neural network pre-trained by a character level language model. The remainder of the paper is structured as follows: We describe event-driven trading and review the relevant literature. We describe our model and the experimental setup used in this work. We presents and discuss the results. Finally, we summarize our work and suggest directions for future research.

LITERATURE SURVEY:

A share market could be a place of high interest to the investors because it presents them with a chance to learn financially by finance their resources on shares and derivatives of varied firms. A chaos system; means the activity traits of share costs area unit unpredictable and unsure. To create some style of sense of this chaotic behaviour, researchers were forced to search out a way, which may estimate the result of this uncertainty to the flow of share costs.

From the analyses of varied applied math models, Artificial Neural Networks area unit analogous to non-parametric, nonlinear, regression models. So, Artificial Neural Networks (ANN) actually has the potential to tell apart unknown and hidden patterns in information which may be terribly effective for share market prediction. If successful, will this will this could this may} be useful for investors and finances which can completely contribute to the economy. There are unit totally different strategies that are applied so as to predict Share Market returns. The securities market reflects the fluctuation of the economy, and receives 10 million investors' attention since its initial development. The securities market is characterized by bad, high-yield, thus investors are involved concerning the analysis of the securities market and making an attempt to forecast the trend of the securities market.

However, securities market is wedged by the politics, economy and plenty of different factors, let alone the quality of its internal law, like value (stock index) changes within the non-linear, and shares knowledge with high noise characteristics, so the normal mathematical applied mathematics techniques to forecast the securities market has not yielded satisfactory results. Neural networks will approximate any advanced non-linear relations and has hardiness and fault-tolerant options.

Therefore, it's terribly appropriate for the analysis of stock knowledge. In dozens of neural network models that were suggests, researchers usually use the hop garden network. hop garden network is that the commonest feedback network model, it's one among the models that almost typically studied currently. The hop garden network is that the mono layer recognized by an equivalent vegetative cell, and is additionally a symmetrically connected associative network while not learning operates.

PROPOSED SOLUTIONS:

BP network is that the back-propagation network. It’s a multi-layer forward network, learning by minimum mean sq. error. It may be employed in the sphere of language integration, identification and adaptation management, etc. BP network is semi supervised learning. Initial of all, artificial neural network has to learn an exact learning criteria, so it will work. Tips for e-learning (Electronic Learning) may be listed as below. If the result yielded by network is wrong, then the network ought to scale back the chance of creating identical mistake next time through learning. This project uses data processing technique to check historical information concerning share market in order that it will predict the desired values a lot of accurately.

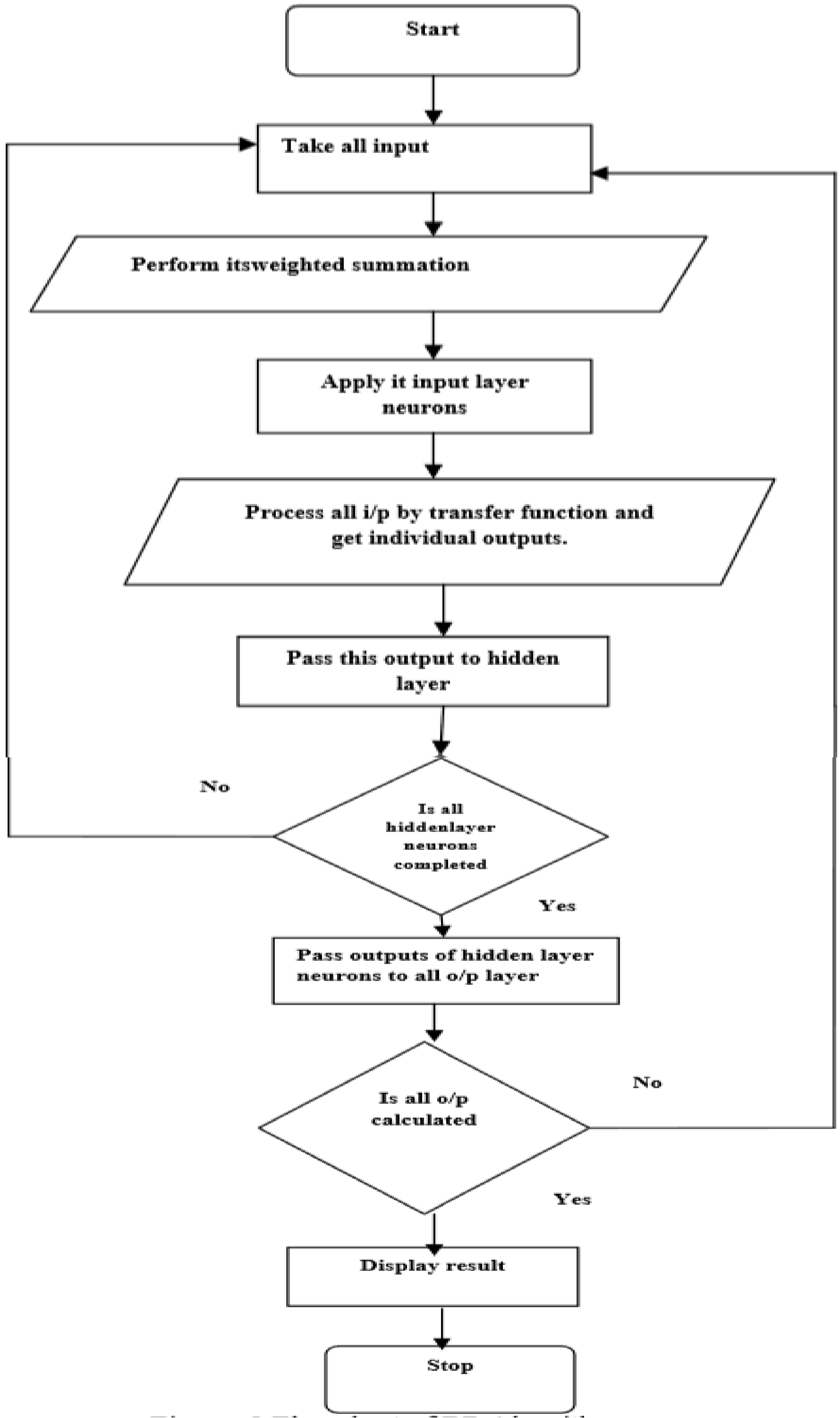
The structure of a unidirectional SLSTM depends on learning from data inputs on which its hidden states have passed through. It only sees information from the past. It has no mechanism to enable it to consider information in the future while predicting the present. The stacked architecture can model more sophisticated data patterns. It can perform deeper analysis on the training data. On the other hand, BLSTM has the ability to process and learn from data in both directions; from future to past and from past to future. BLSTM while going through data combines forward and backward contextual information and uses it to make prediction or classification.

BACK PROPOGATION ALGORITHM:

1) Accept input sample
2) Perform its weighted summation.
3) Apply it to input layer neurons.
4) Process all inputs at each neuron by transfer function to get individual.

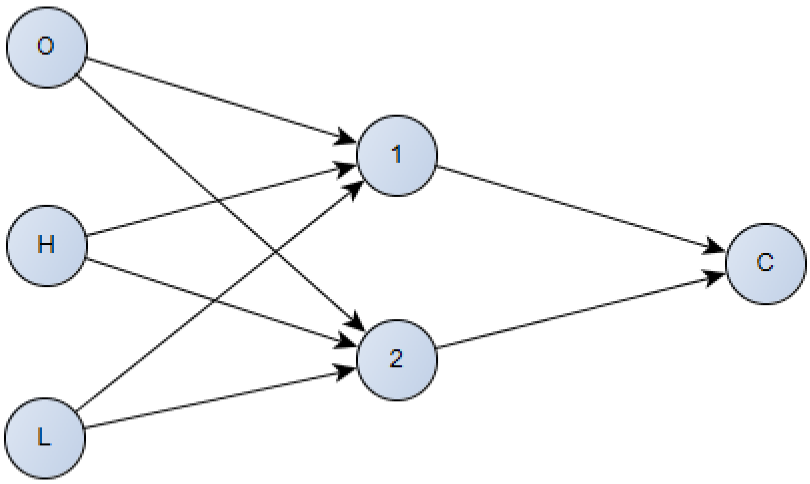
5) Hidden layer and repeat 1,2,3,4 steps pass it as an input to all neurons of for hidden layer neurons.
6) Pass output of hidden layer neurons to all output layers and repeat 1,2,3,4 steps to get final output.
7) Display the final output.

FLOW CHART FOR BACK PROPOGATION:

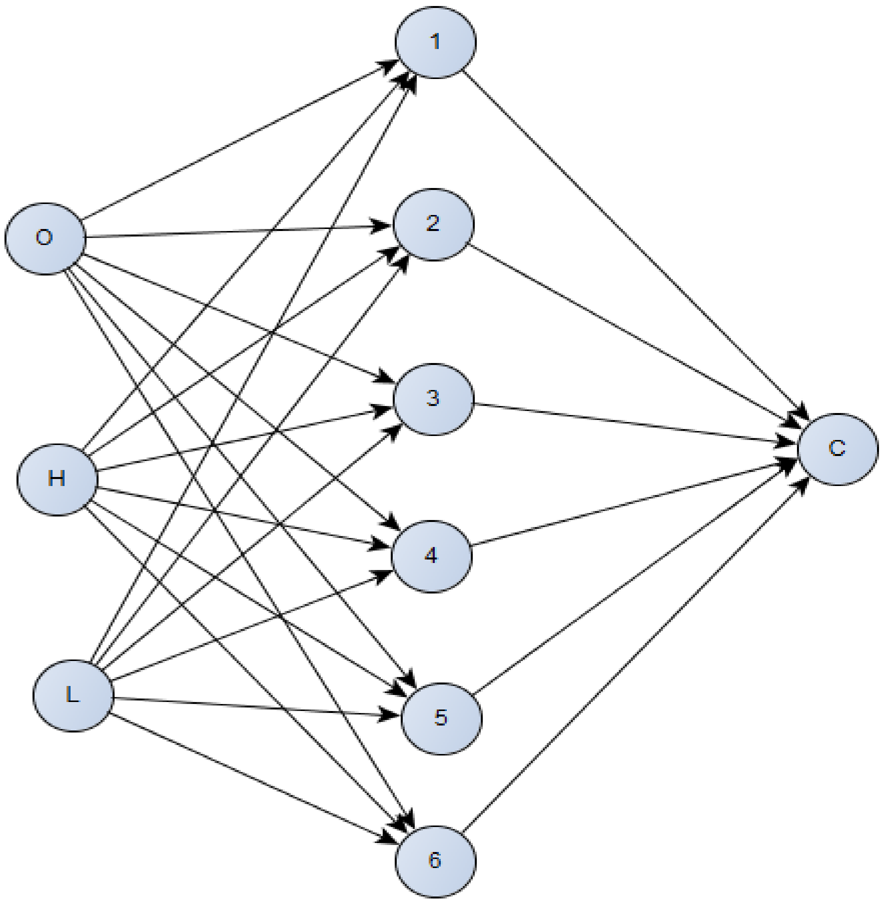


ARCHITECTURE DIAGRAM:

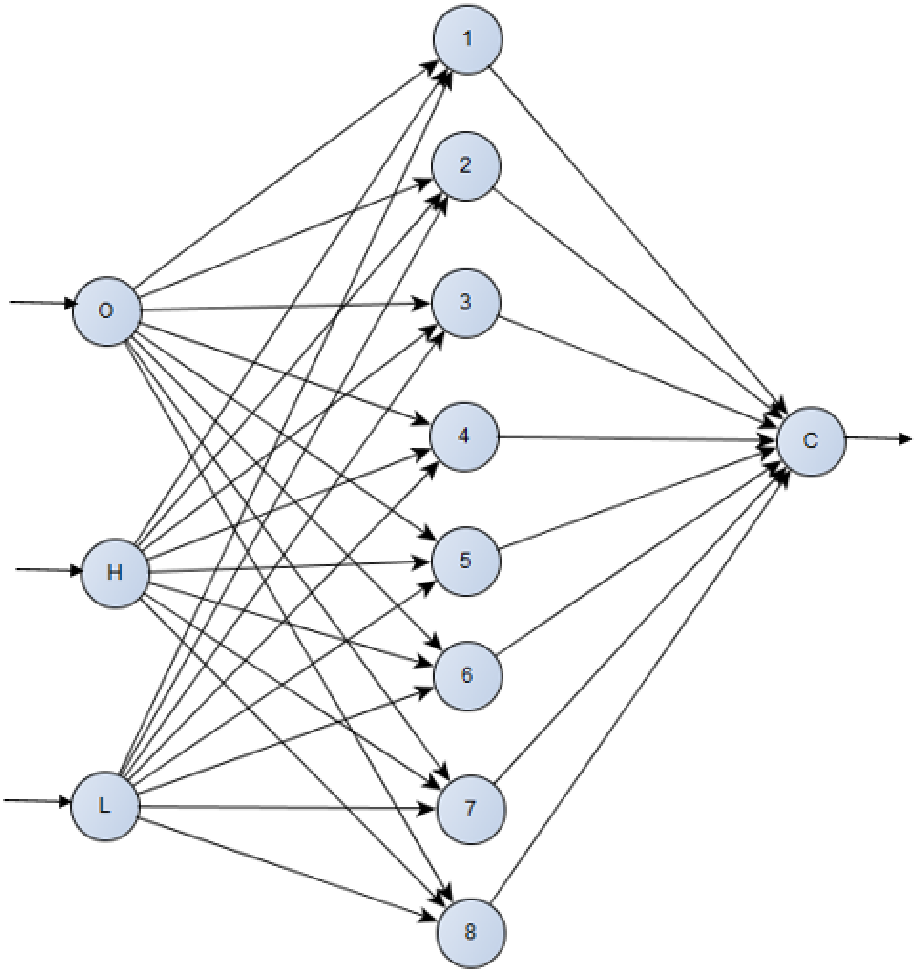
3:2:1



3:6:1



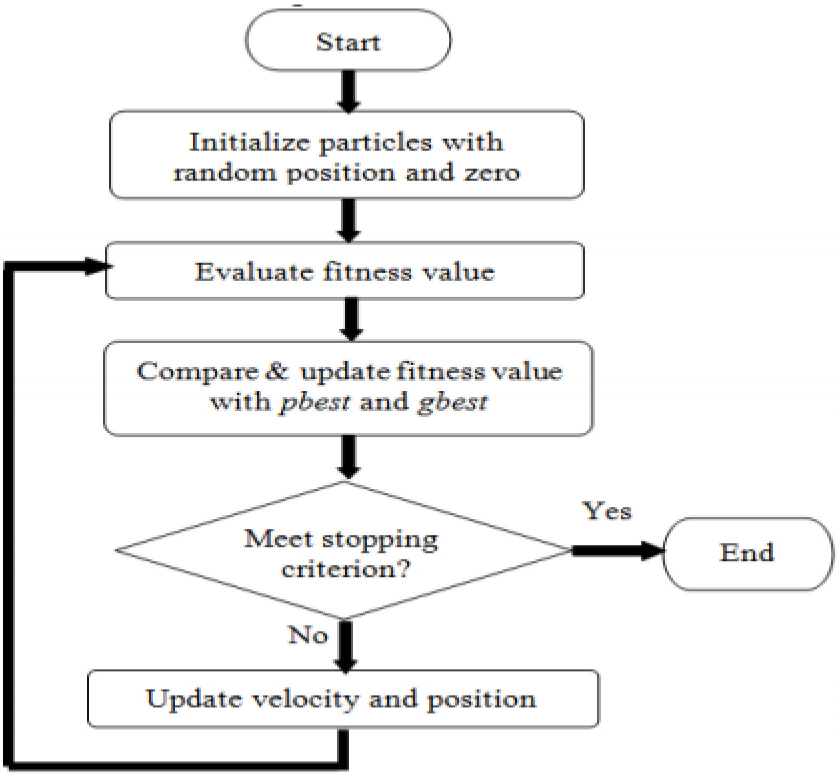
3:8:1



ERROR CALCULATION:

Calculating Root Mean Square,
Let RMS is denoted as Root Mean Square, E is denoted as Error of difference between actual value and predicted value
GE means Global Error.
E=$\sqrt{GE}$/SIZE
Updating, error value,
GE+=Δ*Δ
Where, delta=expected value-actual value
Activation function: Sigmoid Result=1/ $(1 + \theta^{-d})$
Tan hyperbolic: Result= *2.0-1.0/ 2.0+1.0

DEEP LEARNING FLOW CHART:



MATLAB CODE:

Code for training the network:

```
% BACKPROPAGATION ALGORITHM TRAINING: ONLY FOR SINGLE
HIDDEN LAYER
% Data Set
pattern=csvread('train.csv');
fid = fopen('wih.dat','w'); % Weights stored of input-
hidden layer
fid1 = fopen('who.dat','w'); % Output stored of hidden-
output layer
alpha =0.9;           % Momentum
%Convergence is made faster if a momentum factor is added
to the weight updation process.
eta = 0.8;            % Learning rate
tol = 0.001;              % Error tolerance
Q = 800;                  % Total no. of the patterns to be
input
n = 3; q = 6; p = 1;      % Architecture

% Initializing the values and weights
```

```
Wih = 2 * rand(n,q) - 1;     % Input-hidden random weight
matrix
Whj = 2 * rand(q,p) - 1;     % Hidden-output random weight
matrix
DeltaWih = zeros(n,q);   % Weight change matrices
DeltaWhj = zeros(q,p);   % matrix of qxp of zeros
DeltaWihOld = zeros(n,q);
DeltaWhjOld = zeros(q,p);
Si = [pattern(:,1:3)];   % Input signals
D = pattern(:,4);        % Desired values
Sh = [zeros(1,q)];       % Hidden neuron signals
Sy = zeros(1,p);         % Output neuron signals
deltaO = zeros(1,p);     % Error-slope product at output
deltaH = zeros(1,q);     % Error-slope product at hidden
sumerror = 2*tol;        % To get in to the loop
i=0;
itt=0;

figure;
%x=linespace(0,1);
tr=plot(D);
tr.Color=[0 0 0.5];
tr.Marker='o';
%hold on;

% Training BPA network

while sumerror>tol && itt<3000  % Iterate(Stops when
error tolerance = 0.001 or when iteration reaches 20,000
   sumerror = 0;
   for k = 1:Q % for loop of input data (Q=99 times)
      Zh = Si(k,:) * Wih;         % Hidden activations
      Sh = [1./(1 + exp(-Zh))]; % Binary sigmoid function
Hidden signals
      Yj = Sh * Whj;              % Output activations
      Sy = 1./(1 + exp(-Yj));     % Binary sigmoid function
Output signals
      Ek = D(k) - Sy;             % Error vector
      deltaO = Ek .* Sy .* (1 - Sy);% Delta output
      for h = 1:q                              % Delta W:
hidden-output
         DeltaWhj(h,:) = deltaO * Sh(h);
      end
      for h = 2:q                              % Delta
hidden
         deltaH(h) = (deltaO * Whj(h,:)') * Sh(h) * (1 -
Sh(h));
```

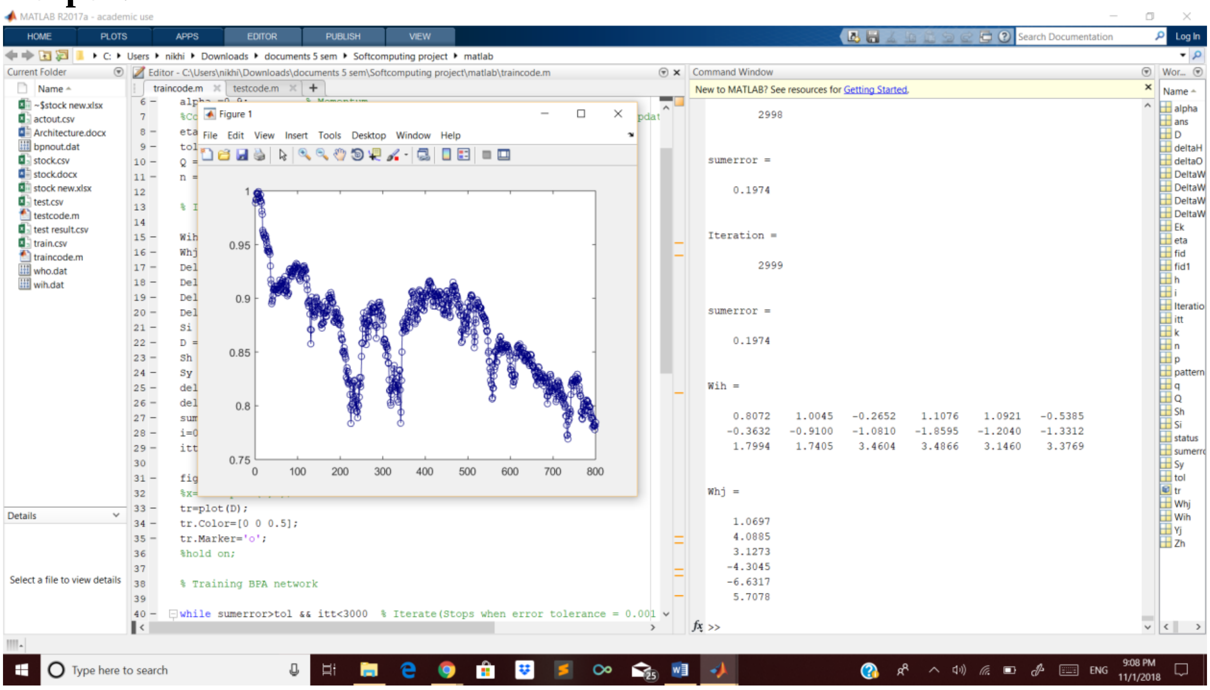
```
        end
        for i = 1:n                                    % Delta W:
input-hidden
            DeltaWih(i,:) = deltaH(q:q) * Si(k,i);
        end
        Wih = Wih + eta * DeltaWih + alpha * DeltaWihOld;
        Whj = Whj + eta * DeltaWhj + alpha * DeltaWhjOld;
        DeltaWihOld = DeltaWih;                            % Update
weights(or)Store changes
        DeltaWhjOld = DeltaWhj;
        sumerror = sumerror + sum(Ek.^2); % Compute error

    end

Iteration = itt
    sumerror % Print epoch error
    itt=itt+1;
end

Wih
Whj
fprintf(fid,'%12.8f\n',Wih);
fprintf(fid1,'%12.8f\n',Whj);
status = fclose(fid);
status = fclose(fid1);
```

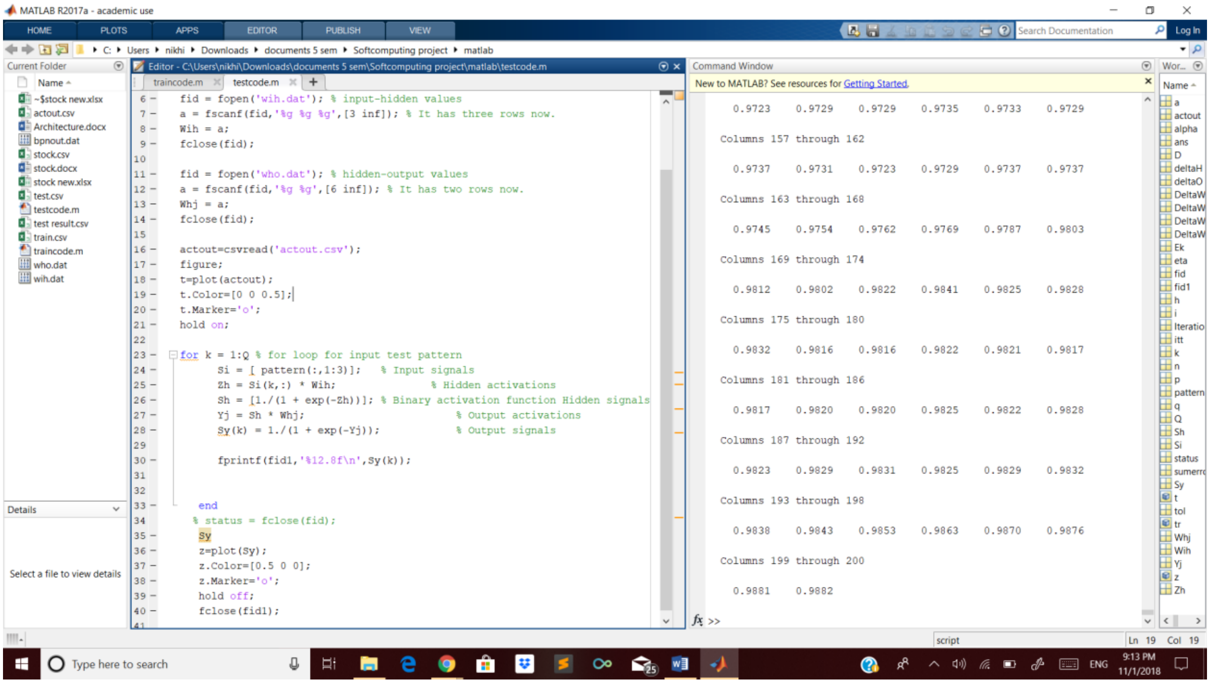
Code for testing the network:

```
% BACKPROPAGATION ALGORITHM TESTING: ONLY FOR SINGLE
HIDDEN LAYER

pattern=csvread('test.csv'); %Open test data file
Q=200;
fid1 = fopen('bpnout.dat','w'); % Store the output of
test data
fid = fopen('wih.dat'); % input-hidden values
a = fscanf(fid,'%g %g %g',[3 inf]); % It has three rows
now.
Wih = a;
fclose(fid);

fid = fopen('who.dat'); % hidden-output values
a = fscanf(fid,'%g %g',[6 inf]); % It has two rows now.
Whj = a;
fclose(fid);

actout=csvread('actout.csv');
```

```
figure;
t=plot(actout);
t.Color=[0 0 0.5];
t.Marker='o';
hold on;

for k = 1:Q % for loop for input test pattern
      Si = [ pattern(:,1:3)];    % Input signals
      Zh = Si(k,:) * Wih;                % Hidden activations
      Sh = [1./(1 + exp(-Zh))]; % Binary activation function Hidden signals
      Yj = Sh * Whj;                         % Output activations
      Sy(k) = 1./(1 + exp(-Yj));             % Output signals

      fprintf(fid1,'%12.8f\n',Sy(k));


   end
  % status = fclose(fid);
   Sy
   z=plot(Sy);
   z.Color=[0.5 0 0];
   z.Marker='o';
   hold off;
   fclose(fid1);
```

Output:




MATLAB R2017a - academic use
Editor - C:\Users\nikhi\Downloads\documents 5 sem\Softcomputing project\matlab\testcode.m
fid = fopen('wih.dat'); % input-hidden values
a = fscanf(fid,'%g %g %g',[3 inf]); % It has three rows now.
Wih = a;
fclose(fid);
fid = fopen('who.dat'); % hidden-output values
a = fscanf(fid,'%g %g',[6 inf]); % It has two rows now.
Whj = a;
fclose(fid);
actout=csvread('actout.csv');
figure;
t=plot(actout);
t.Color=[0 0 0.5];
t.Marker='o';
hold on;
for k = 1:Q % for loop for input test pattern
Si = [pattern(:,1:3)]; % Input signals
Zh = Si(k,:) * Wih; % Hidden activations
Sh = [1./(1 + exp(-Zh))]; % Binary activation function Hidden signals
Yj = Sh * Whj; % Output activations
Sy(k) = 1./(1 + exp(-Yj)); % Output signals
fprintf(fid1,'%12.8f\n',Sy(k));
end
% status = fclose(fid);
Sy
z=plot(Sy);
z.Color=[0.5 0 0];
z.Marker='o';
hold off;
fclose(fid1);
Command Window
0.9723 0.9729 0.9729 0.9735 0.9733 0.9729
Columns 157 through 162
0.9737 0.9731 0.9723 0.9729 0.9737 0.9737
Columns 163 through 168
0.9745 0.9754 0.9762 0.9769 0.9787 0.9803
Columns 169 through 174
0.9812 0.9802 0.9822 0.9841 0.9825 0.9828
Columns 175 through 180
0.9832 0.9816 0.9816 0.9822 0.9821 0.9817
Columns 181 through 186
0.9817 0.9820 0.9820 0.9825 0.9822 0.9828
Columns 187 through 192
0.9823 0.9829 0.9831 0.9825 0.9829 0.9832
Columns 193 through 198
0.9838 0.9843 0.9853 0.9863 0.9870 0.9876
Columns 199 through 200
0.9881 0.9882

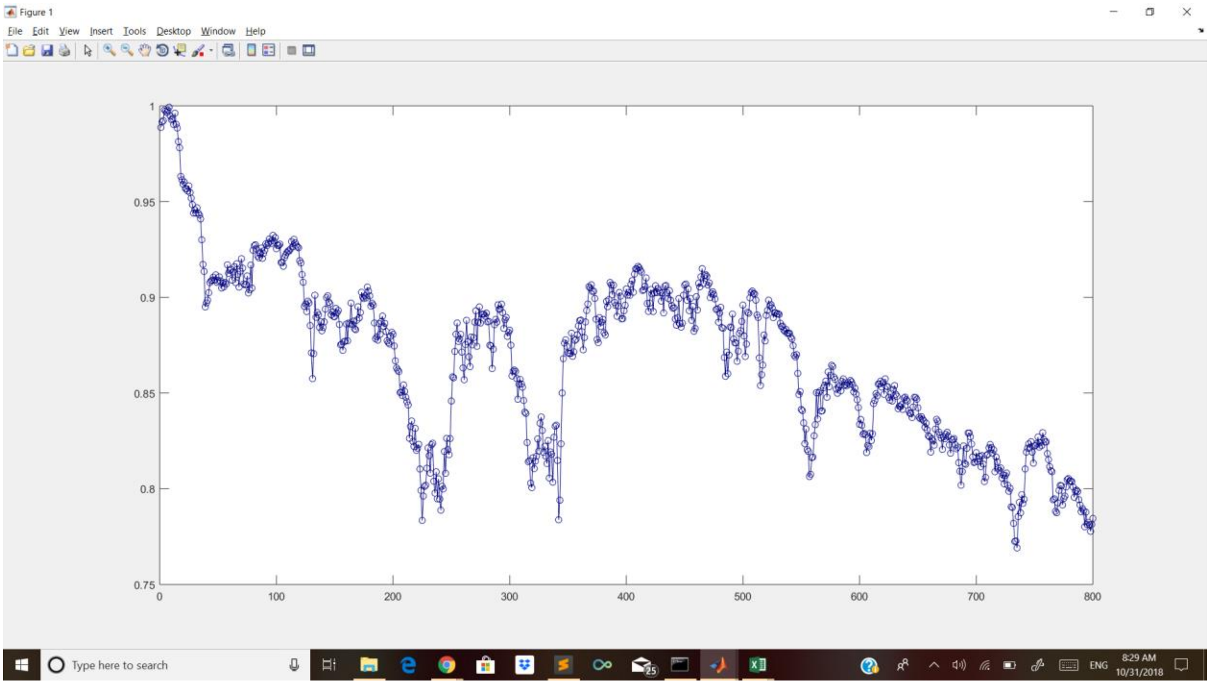


Figure 1
1
0.95
0.9
0.85
0.8
0.75
0
100
200
300
400
500
600
700
800

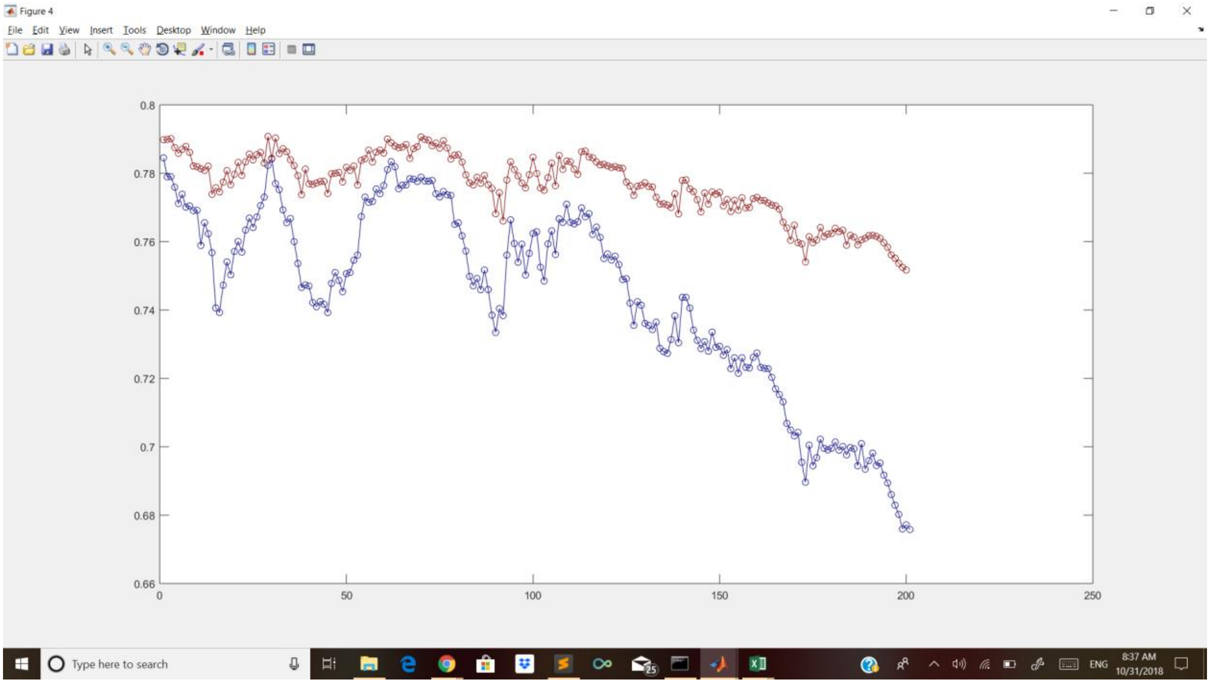


Figure 4
File Edit View Insert Tools Desktop Window Help
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0
50
100
150
200
250
Type here to search
ENG
8:37 AM
10/31/2018

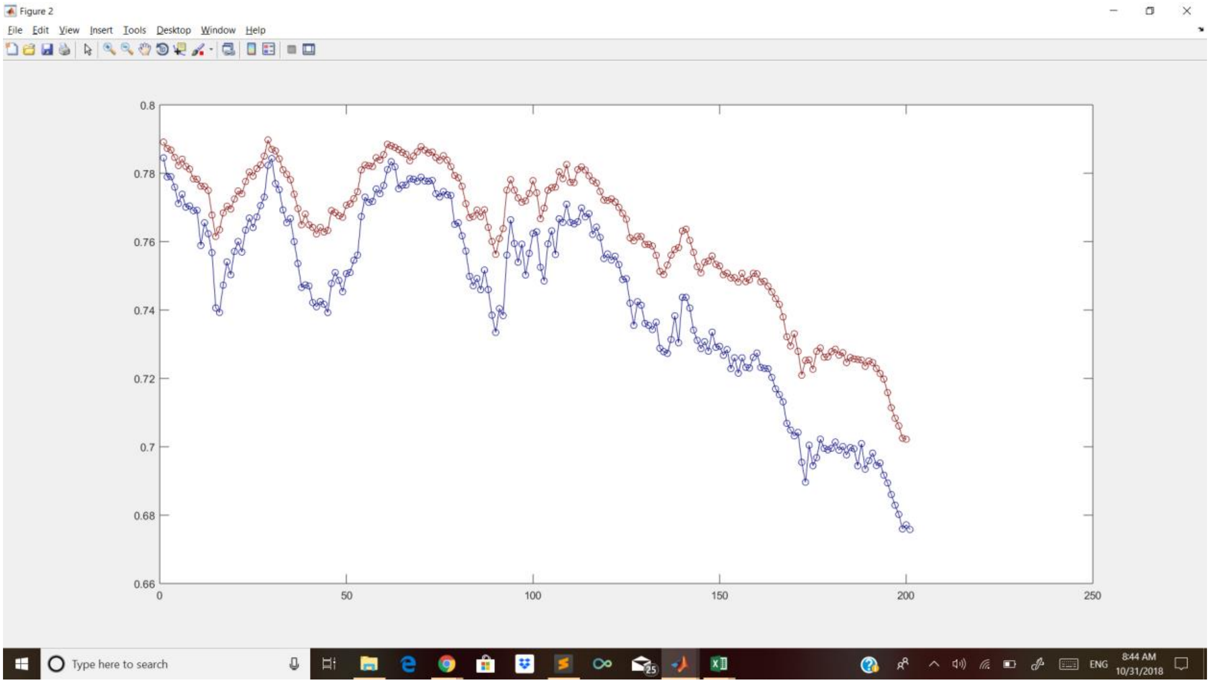


Figure 2
File Edit View Insert Tools Desktop Window Help
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0
50
100
150
200
250
Type here to search
ENG
8:44 AM
10/31/2018

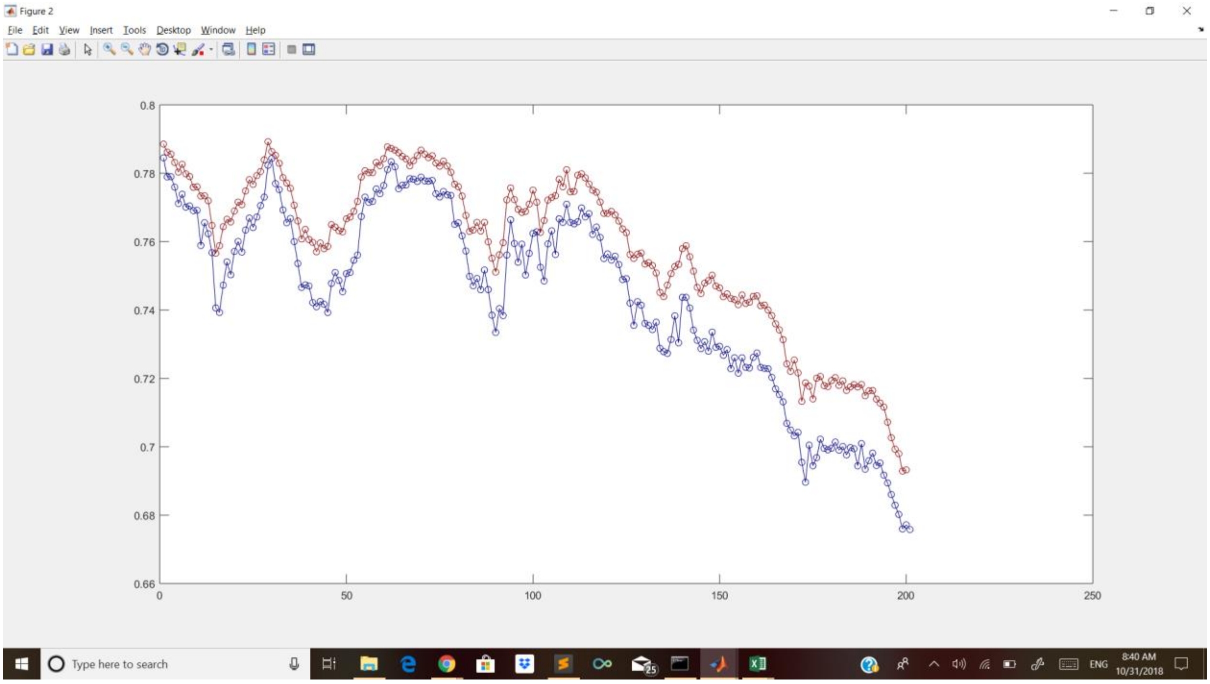


Figure 2
File Edit View Insert Tools Desktop Window Help
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0
50
100
150
200
250
Type here to search
ENG
8:40 AM
10/31/2018

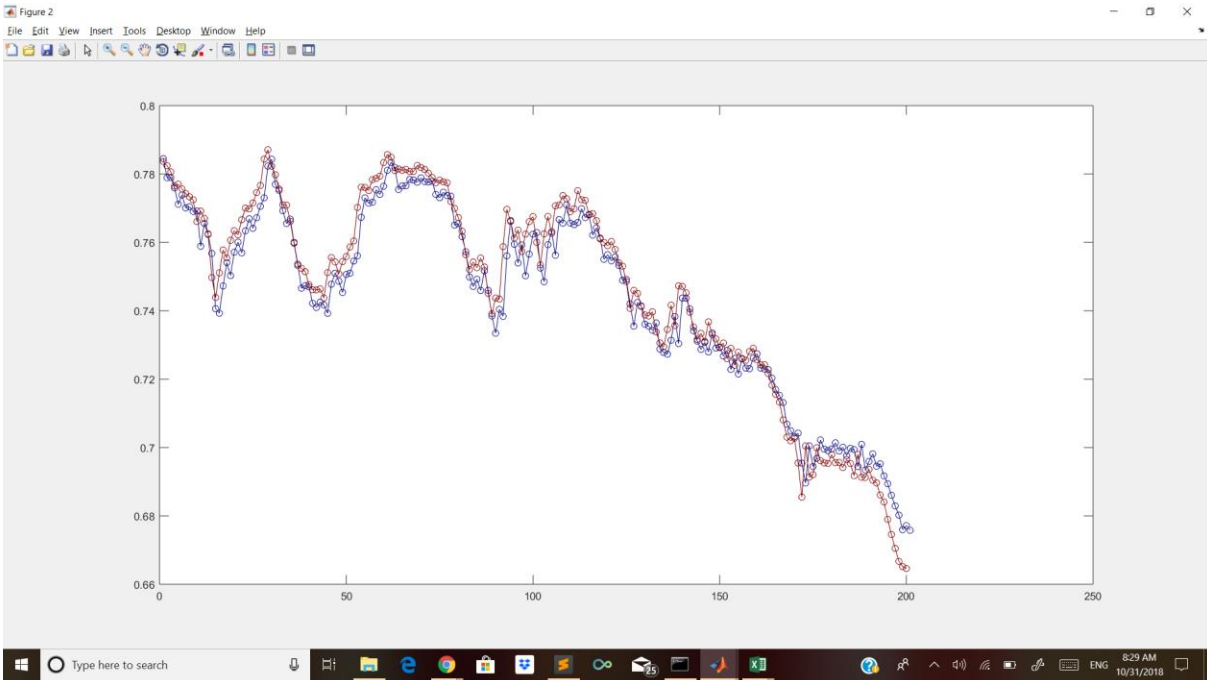
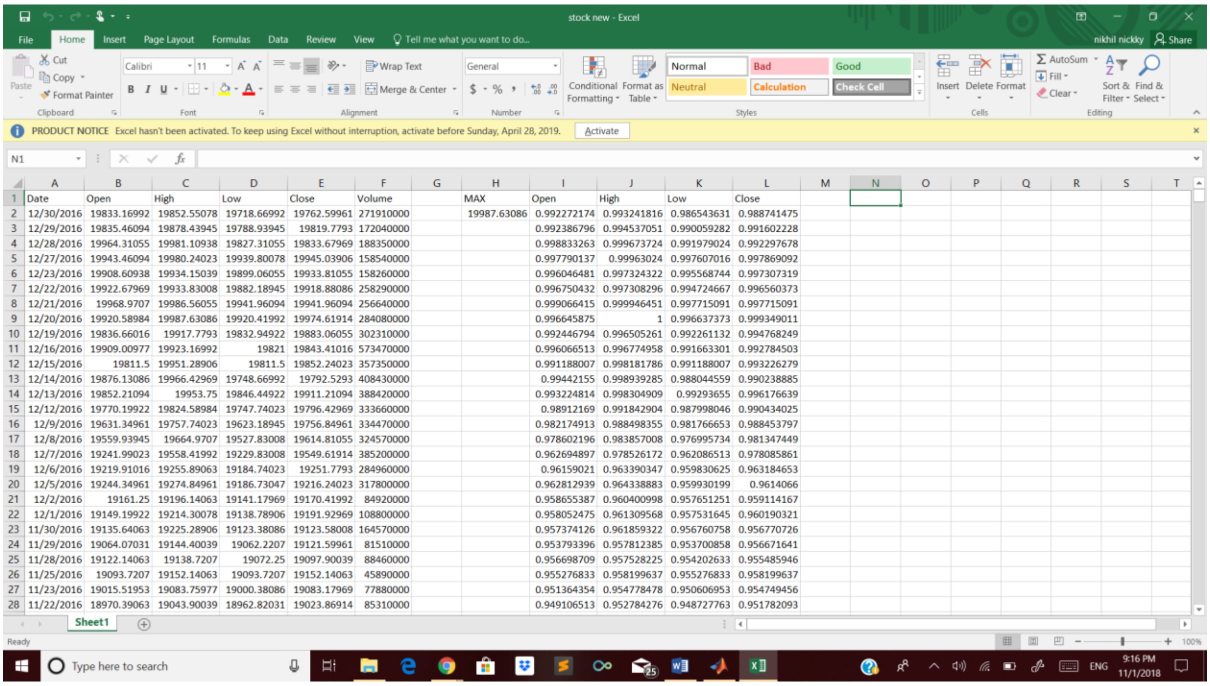
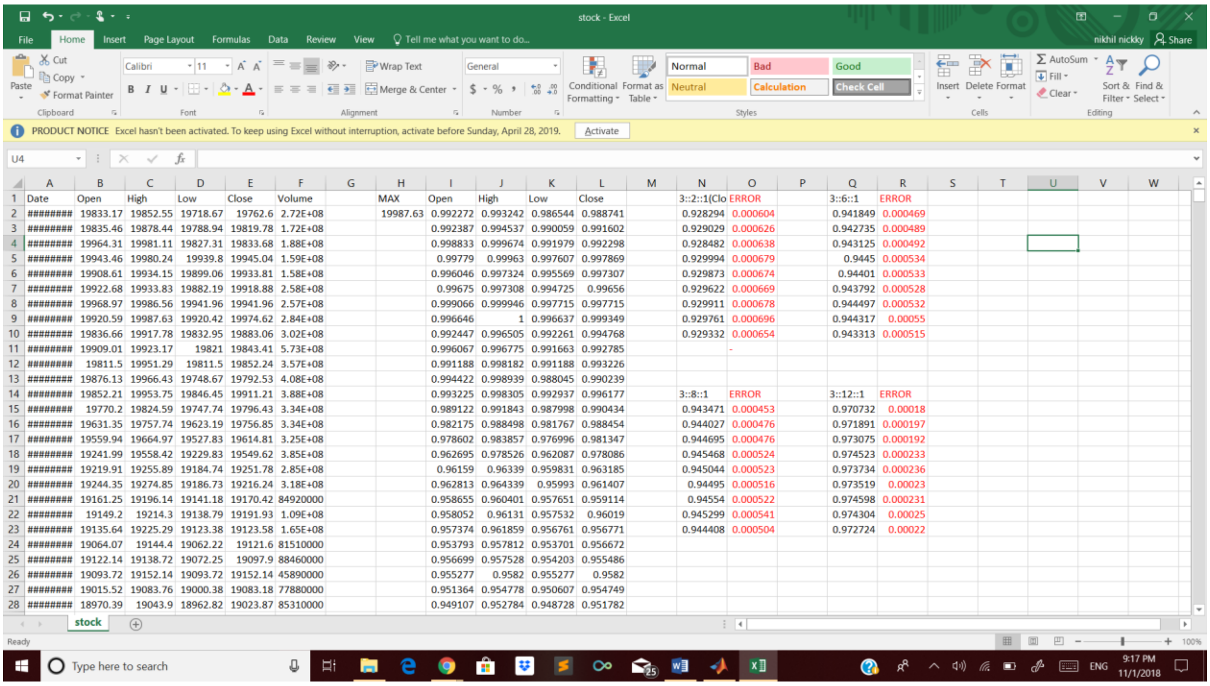


Figure 2
File Edit View Insert Tools Desktop Window Help
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0
50
100
150
200
250
Type here to search
ENG
8:29 AM
10/31/2018





CONCLUSION:

Back propagation algorithm is the best algorithm to be used in Feed forward neural network because it reduces an error between the actual output and desired output in a gradient descent manner.

The LSTM neural network with character level embedding for stock market prediction using only financial news as predictors. Our results suggest that the use of character level embedding is promising and competitive with more complex models which use technical indicators and event extraction methods in addition to the news articles. Character embedding models are simpler and more memory ecient than word embedding and are also able to keep sub-word information. With character embedding the risk of seeing unknown tokens in the test set is diminished, since the data sparsity is much lower than with word embedding. In the future we consider testing the use of character embedding with more complex architectures and possibly the addition of other sources of information to create richer feature sets.

REFERENCES

[1] Prakash Ramani, Dr.P.D.Murarka,"Stock market Prediction Using Artificial Neural Network", International Journal of Advanced Research in Computer Science and Software Engineering, volume 3 issue 4, April 2013.

[2] Neelama Budhani, Dr.C.K.Jha, Sandeep K. Budhani "Application Of Neural Network In Analysis Of Stock Market Prediction", International Journal Of Computer science And Engineering Technology, volume 3 no.4 April 2012.

[3] Zhou Yixin Jie Zhang, Stock data analysis based on BP neural network,2010 second international conference on communication Software and Network.

[4] Zabir Haider khan, Tasnim Sharmin Alin md. Akter Hussain,price prediction of share market using Artificial Neural Network(ANN), International Journal of computer application(09758887) volume 22 no.2, May 2011.

[5] K. K. Sureshkumar, Dr. N. M. Elango, Performance analysis of Stock price prediction using Artificial neural Networks, Global journal of computer science and Technology, volume 2 issue 1 version 1.0 January 2012.

[6] B. Manjula, S.S.V.N. Sharma, R. Lakshman Naik, G. Shruthi, Stock Prediction using Neural Network, International journal of advantage engineering sciences and technologies vol.no 10,issue no 1,013018.

[7] K. K. Sureshkumar, Dr. N. M. Elango ,An Efficient Approach to forecast Indian Stock Market Price and their Performance analysis,International journal of computer application (09758887) volume 34, no 5, November 2011.

[8] Md. Syedul Amin,Md. Mamun, Fazida Hanim Hashim,Jubayer Jalil

and HazahHusain,Design and Implementation of Novel Artificial Neural Network Based StockMarket Forecasting System on Field-Programmable Gate Arrays, American Journalof Applied Sciences 8 (10): 1054-1060, 2011 ISSN 1546-9239.
Bhagwant Chauhan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 904-907
www.