Stock Market Prediction using Neural Networks

Vaibhav V. Shah, Smitkumar J. Mirani, Yashvardhan V. Nanavati, Vishal Narayanan, Sheetal I. Pereira

Abstract— In this paper we present our efforts to predict the stock market using Artificial Neural Networks. We study different types of Neural Networks, their salient features along with the internal working of these networks and the various configurations that they can be run with. We go to comment on the advantages and disadvantages of these networks. Finally we select the one network with specific configurations and use it to predict the stock prices of a few selected companies from the National Stock Index. We achieve best case accuracy of 98% on the dataset.

Keywords: Artificial Neural Networks, Neurons, Back propagation algorithm, Prediction methods, Stock markets.

I. INTRODUCTION

Inspired by central nervous system, particularly the brain, Artificial Neural Networks try to create a system to mimic the brain and achieve similar processing and computing power.[1] Warren McCulloch and Walter Pitts were the first scientists to talk about artificial neural networks in 1943. They suggested how neurons in a brain may work and tried to create artificial neurons to function similarly. But they did this using electrical circuit. Donald Hebb made significant contributions to field like publishing that every time a particular neural path is used, its strength increases. This is what is happening when the brain learns. After initially giving high promises and hopes to this field, funding greatly declined by the 1970s. This was partly because of the rise of the von Neumann architecture that was taking over the computing scene. The other major reason was that initial hype over the possibilities of the neural network led to high expectations. But hardware limitations and lack of proper research meant that those expectations were not met.

However, fearing that it would be left behind in the field after Japan declared efforts to develop the Fifth Generation computing using neural networks, the United States increased funding and promoted research in the area. Problems such as implementing Windrow-Hoff rule to multiple layers were solved and new algorithms such as back-propagation were discovered.

Today neural networks are used in many aspects. Deep Mind, acquired by Google in 2014, is one company that is trying to make practical applications using neural networks. Deep Mind's Alpha Go is the first program to beat a professional Go player[3]. Google itself has implemented neural networks with its email service, Gmail. Using neural networks, Google claims to filter out spam better and also be able to automatically generate replies to emails. [4][5]

Revised Version Manuscript Received on March 04, 2016.

Vaibhav V Shah Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India.

Smitkumar J Mirani Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India.

Yashvardhan V Nanavati Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India.

Vishal Narayanan Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India.

Prof. Sheetal I Pereira Computer Engineering, K. J. Somaiya College of Engineering, Mumbai, India.

In section 2, we talk about Artificial Neural Networks, their history and current developments. In section 3, we talk about the algorithm we have implemented in our neural network. We discuss our networks in section 4 and our dataset in section 5. Section 6 consists of our results and our conclusion is in section 7.

II. ARTIFICIAL NEURAL NETWORKS

In this section we describe the structure of Artificial Neurons and how they are connected to form Artificial Neural Networks.

A. Artificial Neurons

"The perception is a mathematical model of a biological neuron. While in actual neurons the dendrite receives electrical signals from the axons of other neurons, in the perceptron these electrical signals are represented as numerical values. At the synapses between the dendrite and axons, electrical signals are modulated in various amounts. This is also modelled in the perceptron by multiplying each input value by a value called the weight. An actual neuron fires an output signal only when the total strength of the input signals exceed a certain threshold. We model this phenomenon in a perception by calculating the weighted sum of the inputs to represent the total strength of the input signals, and applying a step function on the sum to determine its output. As in biological neural networks, this output is fed to other perceptrons."[6]

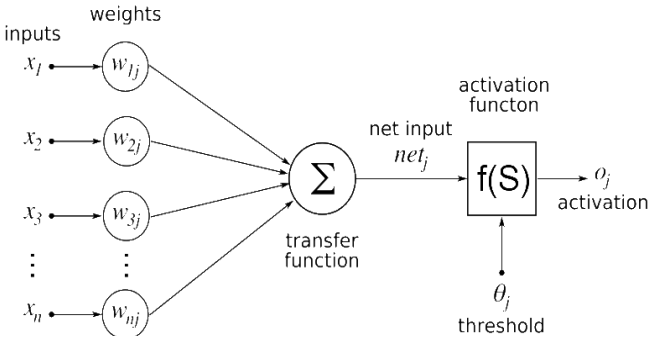

Figure 1. Example of an Artificial Neuron.

As stated, artificial neurons, or perceptrons, take inputs from the previous layer. This is then multiplied by the weight for that input line. The transfer function then sums up all the input-weight products.

$$net_j = \Sigma x_i w_{ij}$$

The activation function then goes on to produce an output for the perceptron based on the input it has received. Activation function can be linear, sigmoidal, ramp step, etc.

$$O_j = f(net_j)$$

The output of this neuron is fed to a neuron in the next layer.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

B. *Artificial Neural Networks*

A number of artificial neurons connected together to mimic the human brain is called a artificial neural network.

An artificial neural network consists of layers. There is one input layer that is responsible for accepting input to the system. The input can be from the user, environment, other systems, etc. There is one output layer that is used to present the output of the network. Depending on the objective of the network, there may be one or more hidden layers in the network.

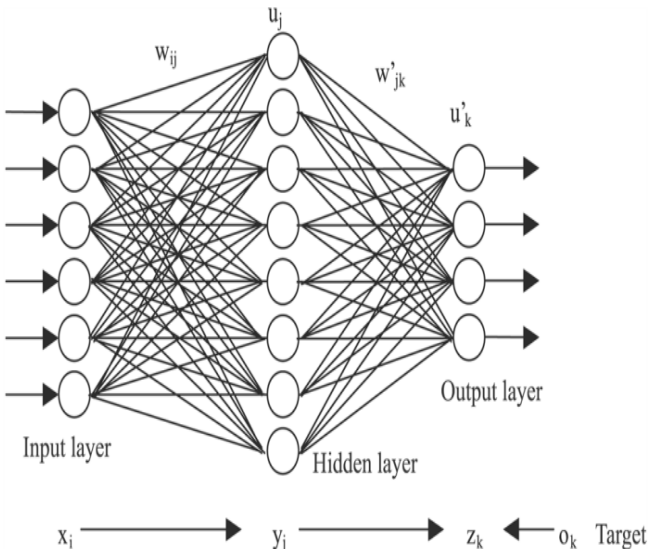

Figure 2. Example of an Artificial Neural Network.

C. *Learning Algorithm*

Learning algorithm dictates how the network learns. Knowledge in the network is represented by the weights. When a network learns, it updates its knowledge. It does this by updating the weights. How it updates the weight depends on the algorithm. The algorithm is chosen on the basis of the functionality of the network, the complexity of the network, the type of inputs and outputs, etc.

All algorithms fall in either of the 3 types of learning: Supervised, Unsupervised, Reinforced.

- In supervised learning, the network is given input-output pairs in the training phase. This helps the machine to identify the output based on the input vector and training samples.
- In unsupervised learning, the network is given only a series of inputs. Using these input, it must develop the output.
- Reinforced learning works just like unsupervised learning, except the output for each input vector is given. This output is used to determine if the network is being trained in the right direction.

III. OUR ALGORITHM

There were many algorithms that were considered for our network. We focused on back propagation. We used the back propagation algorithm in our network because it has many advantages.

While it may be complex and time consuming, back propagation is accurate and very versatile. In back propagation, the error is calculated at the output node and backwards through all the different layers.

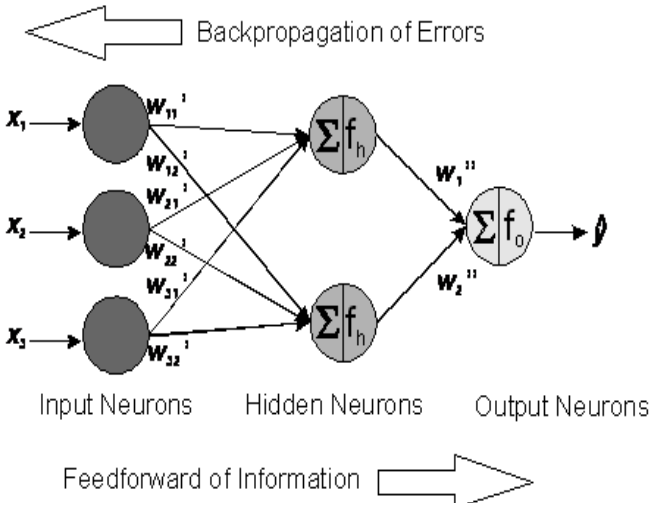

Figure 3. Example of a neural network using backpropagation algorithm.

The output nodes represent all the possible outputs of the network. So if you have only 2 outputs, then you only need one node i.e. binary output. Similarly, you need at least one input for all the different outputs. If you have six outputs, then you need at least six input vectors, one for each output.

In backpropagation, the number of nodes in the hidden layer determine the nature of the network. More nodes in the hidden layer is advantageous for highly fluctuating functions. Conversely, fewer nodes are required for smoother, more predictable functions.

Two important things to note:

Too few neurons in the hidden layer, means less "brain power". Hence the accuracy of the predictions is lost.

Too many neurons in the hidden layer, means instead of working out, or figuring out the right answer, the network simply remembers the answer from the inputs

Either case, we would like to avoid. Hence the number of neurons in the hidden layer play a very critical role in the network.

As far as activation functions go, backpropagation algorithm requires the it to be continuous. This is because differentiation of the activation function is taken when propagating the error backwards.

One important concept relating to backpropagation is momentum. Momentum allows the network to continue to train in one direction if it get better results in that direction. While this is not necessary, it certainly helps to speed up the processing ability of the network.

IV. OUR NETWORK

We will state the specifics of our network in this section.

A. *Training Function*

We will be using the Train LM or Levenberg- Marquardt optimization to train the network. Although it requires more memory as compared to other algorithms, it is fastest method to train a Back-Propagation network. As such it is often the first choice for supervised training[10].

B. *Adaption Learning Function*

So for adapting the network we will be using the LearnGDM method. LearnGDM stands for Learning with

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

gradient descent and momentum[11]. Gradient Descent allows us to find out the minimum value of our function in an efficient manner. The momentum allows the network to improve its learning in a particular direction if there are successive successful minimums of a function in a direction.

C. *Performance Function*

Mean Square Error is what we will be using to calculate the performance of our system. This is a limitation set upon us since we are using TrainLM which uses the Jacobian for calculations.

D. *Hidden layers and Neurons*

There are 3 hidden layers. The first layer has 20 neurons as input. The second layer has 10 neurons. The last layer is out output layer and thus has only 1 neuron.

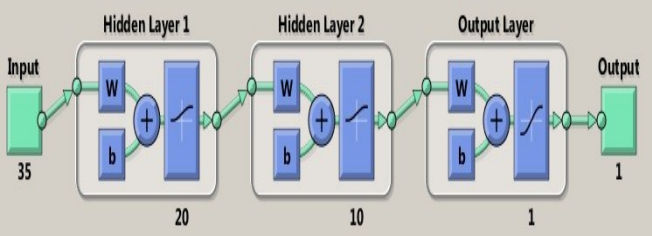

Figure 4. An abstract view of our neural network.

E. *Transfer Function*

For Back-Propagation, the transfer function must differentiable. As such we have selected a log-sigmoid transfer function.

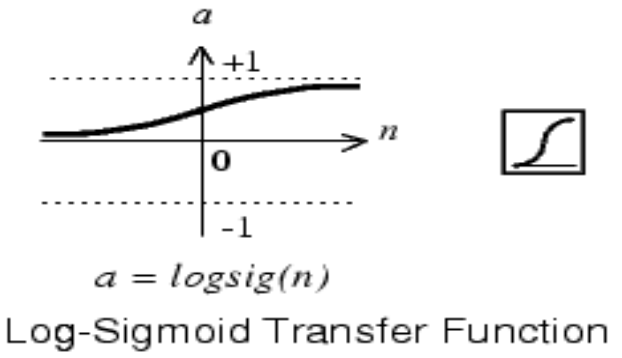

Figure 5. A graphical view of Log-Sigmoid Transfer Function.

V. OUR DATASET

The data sets we acquired were from the website of Quandl[14]. Quandl acts as a search engine for numerical data. There are millions of Economic, social and financial datasets. The data is collated from various sources and is available to be used in multiple formats and with multiple APIs.

We downloaded the data for several companies from Quandl. We tried to look for companies that are still active in the stock market. We went for bigger companies that tend not to be affected by spooks in the market and rumours. This makes the stock price more stable and less prone to fluctuations. As far as time line is concerned, we have datasets ranging from 3 years to 5 years. This gives us about 300 to 700 tuples.

When we download the datasets, we get 8 attributes: Date, Opening Price, High Price, Low Price, Last Price, Closing Price, Total Trade and the Turnover. Out of all of these we are only taking into everything except the date. After we extracting this data we are normalizing the data using the zscore function of MATLAB. Zscore subtracts the mean value of the column from each tuple and divides by the standard deviation from the result[15]. The processed dataset contains the normalized data which is ranges from -1.96 to +1.96.

$$z = \frac{(\text{data point - mean})}{\text{standard deviation}}$$

Figure 6. Formula for zscore function for data normalization

This data is used as input to train the network. However for each day we use these 4 information from the previous 5 working days. So in effect, we have 35 inputs for each prediction. The type of learning performed is supervised. So for the target we use is the next day's Closing Price as our desired output.

VI. RESULTS

We ran on our network on multiple companies with multiple sizes of the datasets. We got varying results. Our accuracy improved when we reduced the number of neurons from 35 to 20. This can be attributed to the system over fitting the data points.

Our system more or less predicted within Rs9 of the actual closing price all the time. But more importantly, it was always able to predict the direction of the stock. It would always predict whether the closing price would be higher or lower than the previous working days accurately.

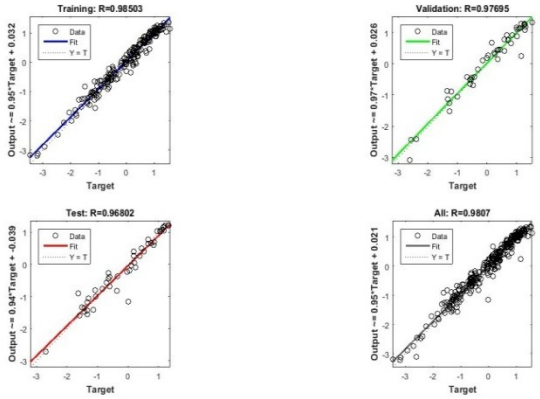

Figure 7. Regression Chart of our Model

The case scenario was that we were off by Rs3. The actual closing price for Cogniti Technology on 1st February, 2016 was around Rs 396 and our model predicted a prince of Rs399.Based on the regression model of our system for that share, it was able to predict the prices with 98% accuracy.

VII. CONCLUSION

Although we can accurately predict the prices of a share to a certain extent, a vast amount of volume of data is required. For our example we have taken more or less 5 points of the share from the entire day. If we can get our hands on these 5

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

data points for every hour in the day we can get a batter more accurate prediction. Also, having more neuron not necessarily results in a better network, as this may lead to over fitting. For the most accurate prediction, a balance must be struck by not having too many neurons or too few neurons. It should work on thousands and thousands of data points. The direction of prediction is also a key to consider while examining the results.

REFERENCES

1. http://wsc10.softcomputing.net/ann_chapter.pdf
2. https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html
3. https://en.wikipedia.org/wiki/AlphaGo
4. http://gizmodo.com/gmail-now-uses-artificial-neural-networks-to-sniff-out-1716975952
5. http://gizmodo.com/googles-neural-network-can-now-reply-to-gmail-messages-1740260404
6. https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html
7. https://www.google.co.in/imgres?imgurl=http://www.extremetech.com/wpcontent/uploads/2015/07/NeuralNetwork.png&imgrefurl=http://www.extremetech.com/extreme/215170-artificial-neural-networks-are-changing-the-world-what-are they&h=590&w=1008&tbnid=ru1iSgMkF619M:&docid=uw9rq85fHW8ClM&ei=LtW1Vrb7IoqeugTZ85fwDg&tbm=isch&ved=0ahUKEwi2ypfhgOPKAhUKj44KHdn5Be4QMwhCKBEwEQ
8. http://mnemstudio.org/neural-networks-backpropagation.htm
9. http://www.frank-dieterle.de/phd/images/image016.gif
10. http://in.mathworks.com/help/nnet/ref/trainlm.html?refresh=true
11. https://en.wikipedia.org/wiki/Gradient_descent
12. http://in.mathworks.com/help/nnet/ref/logsig.html
13. http://matlab.izmiran.ru/help/toolbox/nnet/logsig.gif
14. https://www.quandl.com/data/NSE?keyword=
15. http://www.measuringu.com/zcalc.htm

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.