1. Introduction

## Overview

The aim of this project is to predict the future values of the stock prices based on its past values using neural network back-propagation learning.

Stock market prediction has been an important issue in the field of finance, engineering and mathematics due to its potential financial gain. As a vast amount of capital is traded through the stock market, the stock-market is seen as a peak investment outlet. Researchers have strived for proving the predictability of the financial market. Henceforth, Stock Market prediction has always had a certain appeal for researchers. While numerous scientific attempts have been made, no method has been discovered to accurately predict stock price movement. Even with a lack of consistent prediction methods, there have been some mild successes.

With the advent of faster computers and vast information over the Internet, stock markets have become more accessible to either strategic investors or the general public. Information from quarterly reports or breaking news stories can dramatically affect the share price of a security. As the Internet provides a primary source of event information which has a significant impact on stock markets, the techniques to extract and use information to support decision making have become a critical task. To predict the stock market accurately, various prediction algorithms and models have been proposed by many researchers in both academics and industry. In addition, for accurate stock market prediction, we investigate various global events and their issues on predicting stock markets.

## Motivation

There are several motivations for trying to predict stock market prices. The most basic of these is financial gain. Any system that can consistently pick winners and losers in the dynamic market place would make the owner of the system very wealthy. Thus, many individuals including researchers, investment professionals, and average investors are continually looking for this superior system which will yield them high returns.

Neural networks have been chosen to model the stock market because of their ability to learn the non-linear mappings between inputs and outputs. The stock market is considered to be highly chaotic model, which cannot be understood by the human mind, and thus can’t be easily expressed. However, with the neural networks’ ability to learn chaotic and nonlinear systems, it is possible to solve many more problems which could not be solved by simple traditional analysis or current computer methodologies.

In addition to financial market predictions, neural networks have been already put to use by many organizations for handling different financial tasks. Some banks use neural networks to scan credit and loan applications to predict bankruptcy probabilities. Though the neural networks were primarily used as a tool in the financial domain, many improvements, with respect to the design issues and the training of neural networks have been achieved during research and the implementation.

2. Literature Review

When predicting the future prices of Stock Market securities, there are several theories available.

Fama, et al. [7], Efficient Market Hypothesis (EMH), in EMH, it is assumed that the price of a security reflects all of the information available and that everyone has some degree of access to the information. Fama’s theory further breaks EMH into three forms: Weak, Semi-Strong, and Strong. In Weak EMH, only historical information is embedded in the current price. The Semi-Strong form goes a step further by incorporating all historical and currently public information into the price. The Strong form includes historical, public, and private information, such as insider information, in the share price. From the tenets of EMH, it is believed that the market reacts instantaneously to any given news and that it is impossible to consistently outperform the market.

Malkiel, et al. [7], proposed different perspective on prediction comes from Random Walk Theory. In this theory, Stock Market prediction is believed to be impossible where prices are determined randomly and outperforming the market is infeasible. Random Walk Theory has similar theoretical underpinnings to Semi-Strong EMH where all public information is assumed to be available to everyone. However, Random Walk Theory declares that even with such information, future prediction is ineffective.

Kimoto, et al. [7], designed a network that was to predict the Tokyo stocks. They used daily data (prices) for training the network for about 33 months. They proposed a variation of back-propagation for the system, called supplementary learning. The weights in the network were updated based on the sum of all errors for all the patterns. During training they used moving simulation, where prediction is done while moving the target learning and prediction periods. They used a modular network consisting of 4 backpropagation neural networks, trained on different data sets.

Ajith Abraham, et al. [9], used soft computing techniques and connectionist paradigms for modeling chaotic behavior of the stock markets. They experimented with the system on the data of NASDAQ stock market and NIFTY stock exchange. They used four different methods to model the markets and compared the results from each of them.

Chan Man-Chung, et al. [10], proposed a variation of learning algorithm, the conjugate gradient learning algorithm with a restart procedure.

A huge amount of research is being done on the application of neural network to stock market. Some of the applications include prediction of IBM daily stock prices, a trading system based on prediction of the daily index, short term trend prediction using dual module networks, weekly index prediction, monthly index prediction using radial basis function or neural network, etc.

The work done in the area of **stock market prediction using neural network** can be classified into broad categories:

* Prediction using past stock index values only, and
* Prediction using past stock index values and other fundamental factors.

## Similar ongoing project

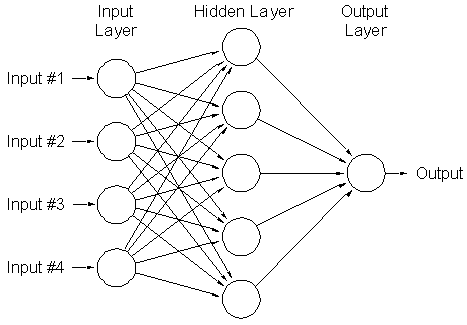
Now days, lots of project running on future prediction such as:-

* Inflation rate prediction
* Property price prediction
* Market crisis prediction
* Oil demand prediction
* Country’s economy prediction
* The recession probabilities
* Global Consciousness Project

3. Theoretical Framework

## Neural Network:

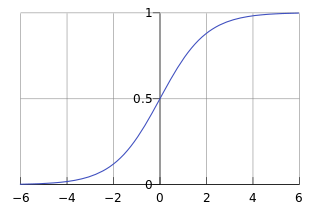
There are many different types of neural networks and techniques for training them but we are just going to focus on the most basic one of them all – the classic back propagation neural network (BPN).  The back propagation refers to the fact that any mistakes made by the network during training get sent backwards through it in an attempt to correct it and so teach the network whats right and wrong.



Input Signal

Error Signal

## Sigmoid Function



Ysigmoid = 1/(1+e-X)

Calculate the actual outputs of the neurons in the output layer:

yk(p) = sigmoid

where ***m*** is the number of inputs of neurons ***k*** in the output layer:

The threshold θ can be used to shift the decision boundary.

Calculate the actual outputs of the neurons in the hidden layer:

yj(p) = sigmoid

where *n* is the number of inputs of neuron *j* in the hidden layer, and

*sigmoid* is sigmoid activation function.

## Error Gradient

This is the formula to calculate the basic error gradient for each output neuron k:

C:\Users\vikash\Desktop\untitled.pngk **=** yk (1-yk)(dk-yk)

where yk is the value at output neuron k

and dk is the desired value at output neuron k.

There is a difference between the error gradients at the output and hidden layers. The hidden layer’s error gradient is based on the output layer’s error gradient (back propagation) so for the hidden layer the error gradient for each hidden neuron is the gradient of the activation function multiplied by the weighted sum of the errors at the output layer originating from that neuron

C:\Users\vikash\Desktop\untitled.pngk **=** yk (1-yk)

## The weight adjustment

The final step in the algorithm is to update the weights, this occurs as follows:

wij = wij + Δwij and wjk = wjk + Δwjk

where Δwij = α × inputNeuron**i** × C:\Users\vikash\Desktop\untitled.pngj

and Δwjk = α × hiddenNeuron**j** × C:\Users\vikash\Desktop\untitled.pngj

**α –** learning rate

C:\Users\vikash\Desktop\untitled.png **-** error gradient

The alpha value you see above is the learning rate, this is usually a value between 0 and 1. It affects how large the weight adjustments are and so also affects the learning speed of the network. This value need to be careful selected to provide the best results, too low and it will take ages to learn, too high and the adjustments might be too large and the accuracy will suffer as the network will constantly jump over a better solution and generally get stuck at some sub-optimal accuracy.

## The learning algorithm

The BPN learns during a training epoch, you will probably go through several epochs before the network has sufficiently learnt to handle all the data you’ve provided it and the end result is satisfactory. A training epoch is described below:

*For each input entry in the training data set:*

* feed input data in (feed forward)
* check output against desired value and feedback error (back-propagate)

*Where back-propagation consists of*:

* calculate error gradients
* update weights

## Basics of stock markets:

A stock is a small share that represents a partial ownership of a company. Stocks are issued by companies in order to raise capitals and are bought by investors in order to acquire a portion of the company. Even a small share of the company will give the investors the right to have a say in how the company is run. Although they gain a portion of the company’s profits, investors do not carry an obligation to the company in cases of defaults or lawsuits.

Stocks are issued by companies to raise capital. A cash injection is needed for either property acquisition or company expansion. Every stock is limited to a particular number of shares. The growth potential and perceived health of the company influences the market adjustment of the par value of the stocks.

When the stock market trends are uncertain, the investor should take care to have diversity in his investments. Otherwise effect of uncertainty will be there on the investments in stock market. Therefore an intelligent investor should take investment decision only after deciding on the schemes, investment tools and how long the investments are to be continued. For this purpose, people are being helped by software which predicts the stock price.

4. Data Collection

There all indices which we tried out for

<http://www.ziddu.com/download/3972399/Niftydailydata.xls.html>

**There are all stock exchanges of the world**

African stock exchanges, Asian stock exchanges, European stock exchanges, Middle Eastern Stock Exchanges, North American Stock Exchanges, and South American Stock Exchanges

<http://www.tdd.lt/slnews/Stock_Exchanges/Stock.Exchanges.htm>

We can also collect data from:

<http://download.finance.yahoo.com/>

<http://www.nseindia.com/content/indices/ind_histvalues.htm>

5. Proposed Approach

The implementation is divided into three phases:

* Design the neural network
* Training the network with past data
* Using the network for forecasting the future trends of the stock based on the present and past data.

## Designing the neural network:

The neural network architecture has to be decided upon before going on with the designing of the mining system for forecasting. Feed-forward networks are the most widely used architecture as they provide good generalization capabilities and are readily implementable. So, a feed-forward network with three layers is chosen as the architecture to model the forecasting task.

The parameters defining the architecture of the neural network are:

Number of layers: The number of layers in the Neural network is three. They are one Input Layer, one Hidden Layer, one Output Layer. The Hidden layer is used to transform the inputs to the neural network using a transfer function, described later.

Number of nodes (neurons): If the input layer consists of ***n*** nodes (neurons), one for each of the stock prices for the last ***n*** days then hidden neurons

Number of hidden neurons = 2n+1.

There is a single output node in the output layer, corresponding to the predicted stock price. The outputs of all of the nodes in the hidden layer are passed on to the output node, each with certain weight.

Weights of the links: All of the weights are initially randomly set in the range [-0.5,0.5].During the training phase, the weights are adjusted according to the error of the nodes in the next layer.

Activation function: It is the function that is fired at each node, after receiving inputs from the previous layer. This is the transfer function that was mentioned earlier. This function transforms the input to a neuron (node) into its output. The activation function used in the implementation is the sigmoid function. This function maps any input x to the range [0,1]. So the outputs of all the nodes belong to the range [0,1].

Learning rate: It defines the rate of learning of the neural network. It can also be interpreted as the acceleration of the network while training. In the implementation the value of learning rate has been set to 0.5.

Input mapping: While taking the inputs to the neural network the maximum of all the inputs is calculated and stored. Then all the inputs are normalized by dividing them by the maximum value. This is done so that during training the predicted output and the expected output both lie in the range [0,1].

Output scaling: The predicted outputs lie in the range [0,1], because of the application of the sigmoid function at all the nodes. To get the actual predicted price, the output is multiplied with the maximum of all the inputs. This scales the output from the range [0,1] to the original scale of the stock price

## Training a Neural Network

The number of hidden layers and the number of neurons in hidden layer are larger if the number of input data is larger too. Training a neural network to learn in the data patterns in the data involves iteratively presenting it with examples of the correct known answers. The objective of training is to find the set of weights between the neurons that determine the global minimum of the error function. Unless the model is overfitted, this set of weights should provide good generalization. The BP network uses a gradient descent training algorithm which adjusts the weights to move down the steepest slope of the error surface. Find the global minimum is not guaranteed since the error surface can include many local minima in which the algorithm can become ‘stuck’.

Using the Network for forecasting

After training the network, it has to be tested with current data, to evaluate its performance. Some of the training data will be kept aside for testing, to avoid over-fitting by the neural network.

The network is used to predict the future stock prices, after training and testing. In the implementation, the network is created so that it predicts the next days’ closing price of the stock. The closing prices of the next days are predicted by adding to the input. Then the trading strategy is used to provide the user with the response of the system in a much understandable manner. The ultimate decision, of buying/holding/selling the stock, is completely onto the user

## Snapshots of the GUI

6. Plan of Work

Phase-1(8 Weeks) ***Completed***

* Getting familiar with Machine Learning , Data Mining and Stock Index.
* Understanding the basic functionalities of the Application.

Phase-2(8 Weeks)  ***Completed***

* Implement the application.
* Developed GUI based interaction for the user.
* Testing of the implemented application.

7. Conclusion

We are using feed-forward neural network for the stock market prediction on the basis of past data and training set data provide to the neural network from archive of stock value index.

8. Future Work

A lot of scope for research still exists as this is a very hot topic and a little improvement is also very valuable. We can also include sentiment analysis in our text analysis to improve the prediction accuracy.

The network predicts the future values only on the basis of the past and current price values. In reality, the prices are affected by many unquantifiable entities, like political, economic changes, etc. They can be taken into consideration to achieve better accuracy in the prediction.

We can also use other machine learning process for prediction of stock market like recurrent neural network, genetic algorithm, etc. to improve the prediction accuracy and efficiency.

9. References

[1]Amol S. Kulkarni(1996), Application of Neural network to stock market prediction. Available:

<http://machine-learning.martinsewell.com/ann/Kulk96.pdf>

[2] Michael Negnevitsky, [Artificial Intelligence: A Guide to Intelligent Systems (2nd Edition)](http://www.amazon.com/Artificial-Intelligence-Guide-Intelligent-Systems/dp/0321204662/ref=dp_ob_title_bk)

[3] Ramon Lawrence(1997), Using Neural Networks to Forecast Stock Market Prices. Available:

<http://people.ok.ubc.ca/rlawrenc/research/Papers/nn.pdf>

[4] Alexendra Cristea, Stock exchange forecasting with the help of Neural Networks. Available:

<http://www.dcs.warwick.ac.uk/~acristea/HTML/stock.html>.

[5] I. Kaastra and M. Boyd. (1996) , Designing a neural network for forecasting financial and economic time series.

[6] M. R. Hassan, B. Nath, and M. Kirley. (2006) A fusion model of hmm, ann and ga for stock market forecasting. [Online]. Available:

<http://dx.doi.org/10.1016/j.eswa.2006.04.007>

Suggestions