**Machine Learning Techniques for Stock Price Prediction**

**Machine Learning, Fall 2011**

Authors:

Anuj Jain

Anuroop Yadav

Hildingur Mahanti

Sampad Mishra

Contents

[1. Executive Summary **Error! Bookmark not defined.**](#_Toc293442984)

[2. Technical Details **Error! Bookmark not defined.**](#_Toc293442985)

[3. Prodecure Details **Error! Bookmark not defined.**](#_Toc293442986)

[4. Implementation Details **Error! Bookmark not defined.**](#_Toc293442987)

[Appendix **Error! Bookmark not defined.**](#_Toc293442989)

**1. Executive Summary**

In this report we test Machine Learning techniques for stock price prediction. Stock prices are usually very dynamic and profit can be made if we can predict stock price movements. Different kinds of algorithms have been implemented to predict stock price movements. Such kinds of algorithms usually fall into one of the two broader categories: Fundamental analysis and Technical analysis. Fundamental analysis deals with predicting the value of the company based on company’s fundamental factors like P/E ratio etc. Technical analysis mainly focuses on finding patterns in the trade prices. Machine learning techniques are very helpful in pattern recognition and hence various Machine learning algorithms have been implemented for stock price prediction.

The main goal of this project was to study and apply as many machine learning techniques as possible, as opposed to coming up with a newer or better strategy for stock price prediction.

This paper starts by first detailing the technical model being used for stock price prediction. After that each machine learning technique and its exact details and simulation results are provided. Finally the conclusions of the project are provided.

**2. Technical details**

Support Vector Machines, Neural Network are used to predict the up and down movements of the stock price. Neural Networks and Linear Regression are applied to predict the stock price itself.

To classify the up or down movement of the price, we use two technical indicators. We use VIX Index and a combination of Short Term Moving Average and Long Term Moving Average. VIX is market indicator for expected volatility and it generally increases in times of downward market. Moving averages are average of stock price during last N days. When N is large, Moving Average is Long term MA else Short Term MA. If STMA is greater than LTMA, we can say that market is upward trending and vice versa.

For predicting the actual price of the stock, we use Exponential Moving Average as an Indicator. EMA is a form of a moving average with more weight to recent values. We make a hypothesis that EMA value tomorrow can be expressed as some function of previous EMA values. Mathematically

EMA(t)= w1\*EMA(t-1) + w2\*EMA(t-2)+.....wn\*EMA(t-n)

Also EMA(t)= (1-alpha)\*EMA(t-1)+alpha\*S(t) ( S(t) is the value of the stock price at time t)

Thus if we can predict the value of EMA(t), we can predict value of S(t)

1.4 Procedure

Data Cleaning -> Machine Learning Technique- Output- Final Performance

**4 Implementation Details**

**4.1 Stock Price Movement Classification**

VIX Index is used and Moving Averages are used. Input Vector will consist of two values. If VIX Increases in value from previous day, input is 1 else -1. Similarly if STMA is greater than LTMA, +1 is passed as second input otherwise -1. As output +1 or -1 is passed depending on stock price movement next day. We are trying to find if movements in inputs can predict the stock price movement of tomorrow. In this section, SVM and Neural Networks are implemented which are described below.

**4.1.1 Support Vector Machines for Classification**

SVM technique aims to classify inputs into two categories. There are two kind of SVM technique. One is Linear Classifier and other is Non Linear Classifier. In case of Linear Classifier, the aim is to find a Hyper plane which divides inputs into two classes. Hyper Plane is chosen in such a way that distance of plane from nearest point on each side is maximum. Other type is Non Linear, where instead of finding Hyper plane, we find a multi dimensional curve to differentiate inputs. Different kind of kernels can be used such as Polynomial, Hyperbolic Tangent, Gaussian Radial Basis Functions. All other than Linear kernels are used for Non Linear Classification. Mathematically, Kernel function above can be described as follows:

Linear: K(xi; xj) =

Polynomial: K(xi; xj)=

Radial Basis Function K(xi; xj)= exp(-

The technique has been implemented in MATLAB . MATLAB procedure used followed is explained below.

Results:

1. With RBF kernels- We use a training set of 2000 data points and then validate it 1000 instances. In this method, we have two parameters C ( cost of error term) and . We vary these parameters while training and compare the performance in validation data.

C=0.001, Accuracy=50%

C=256, Accuracy=69%

Thus we see that as we increase the cost of error term, accuracy increases up to a certain point after which it stabilizes, which can be thought of as maximum accuracy that can be reached.

**4.1.2 Neural Networks for Classification**

**4.2 Stock Price Prediction**

10 day EMA is used as an indicator and it is modeled as a function of previous 10 days EMAs.

EMA(t)=f( EMA(t-1),…..EMA(t-10))

Once we have predicted the EMA, we use EMA(t)=(1-)\*EMA(t-1)+\*S(t) where S(t) is the stock price

Where

**4.2.1 Neural Network for EMA prediction**

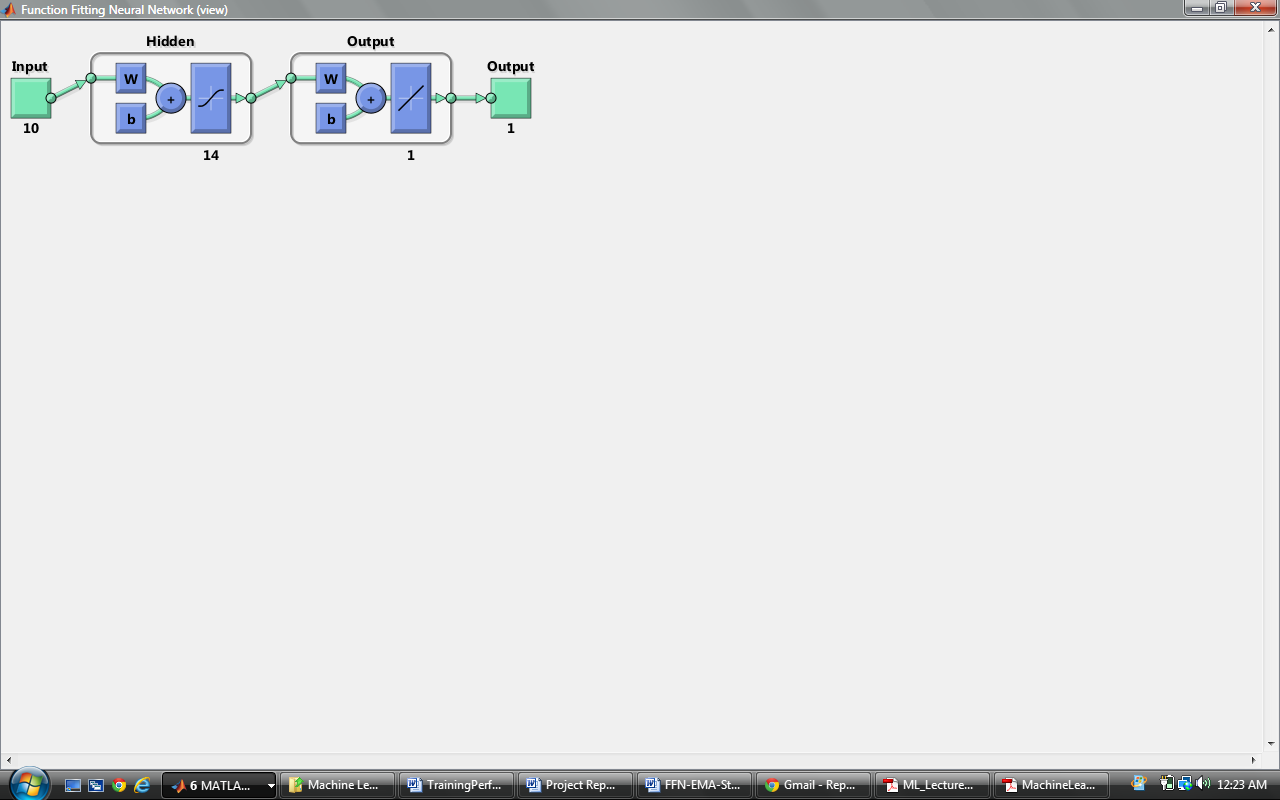
We used a Feed Forward Neural Network (FNN) to predict the value of EMA on day 11 based on values of EMA on previous 10 days.

In theory, a Prediction Problem can be considered as evaluating a function F at time t based on the previous values of F at times t-1,t-2,t-n while assigning corresponding weight function w at each point to F.

*F (t) = w1\*F (t-1) + w2\*F (t-2) + … + w\*F (t-n)*

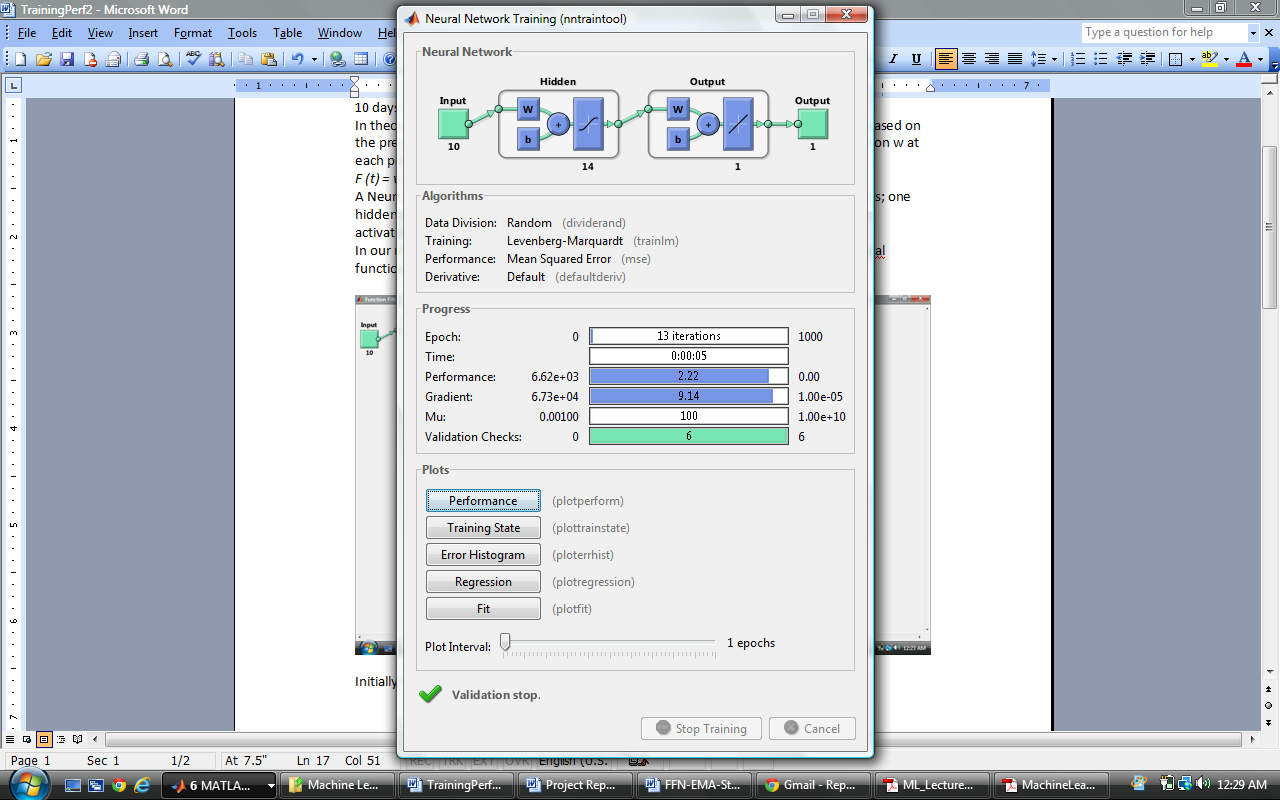
A Neural Network is ideal for this kind of a problem as a Neural Network with two layers; one hidden layer with non-linear and smooth activation function and an output layer linear activation is sufficient to universally approximate any function.

In our model we used a FNN with 14 hidden layer nodes with a tanh (TRANSIG) sigmoidal function and linear output activation. Figures for architecture and performance are presented below.

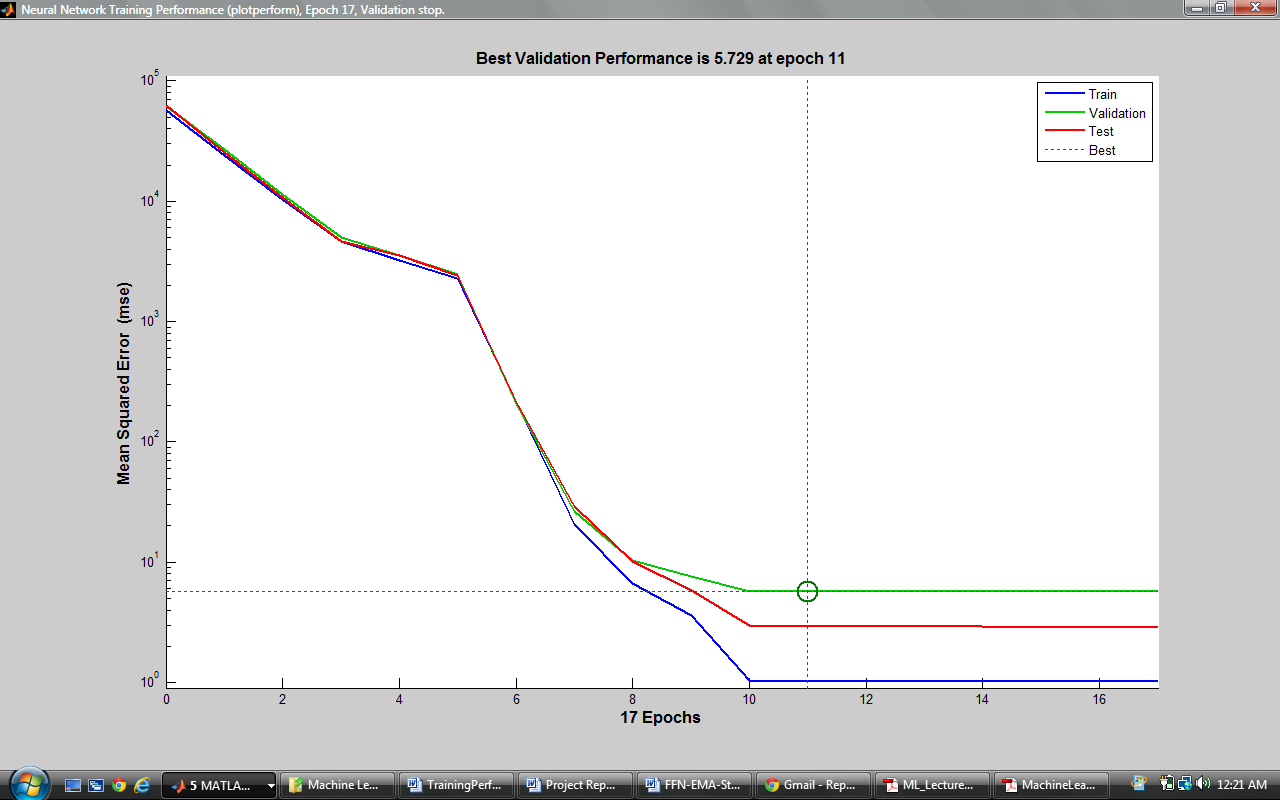


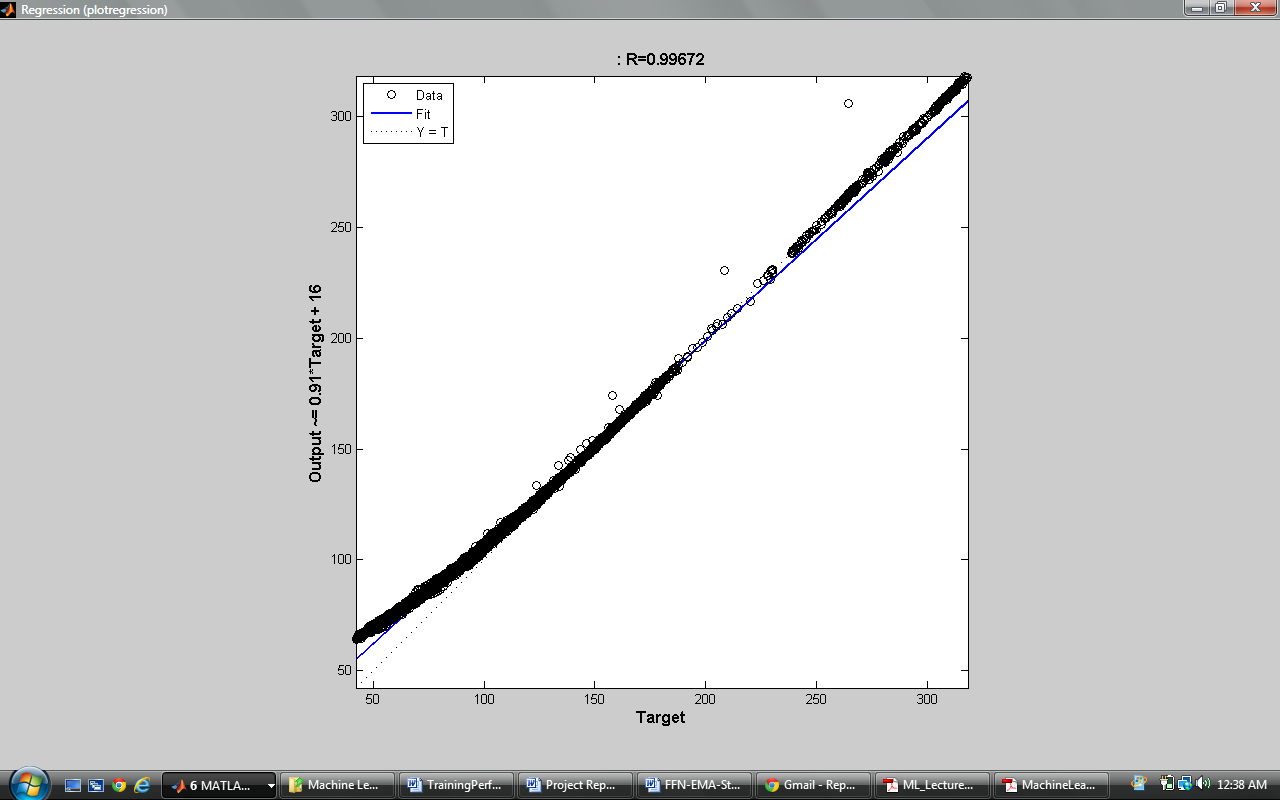
We used the past 50 years of IBM 10 day EMAs. Data was set up the data such that we had 50 years worth of 10 day EMAs for the past 10 days in each row.

The approach was to use the 47 years 10 day EMAs until the past 3 years for training the FNN.



The Training performance indicates the following:





Using the past 3 years of 10 day EMA day 11 EMA was predicted. The predicted EMA for the next day was then compared with the observed 10 day EMA on those dates and a MSE was reported.

**4.2.2 Regression:**

We apply liner least square regression to predict the EMA and therefore the stock price.

We use 2000 days as training set and apply it to predict the values for next 2000 days. We then plot the MSE of the stock price as a function of Number of instance in testing set. Results and plot of MSE is presented below.

Results:

|  |  |
| --- | --- |
| *Regression Statistics* | |
| Multiple R | 0.999784 |
| R Square | 0.999568 |
| Adjusted R Square | 0.999566 |
| Standard Error | 1.79493 |
| Observations | 2000 |

As we can see, regression works very well upto a certain point and then error increases very rapidly.

The spike in the error initially can be attributed to less instance used.