Stock Market Prediction

1 Introduction:

Stock Markets have always intrigued people and the ability to predict it's moment is a rare and valued skill. Thus we have attempted to do just that.
The Standard & Poor's 500 often abbreviated as S&P 500 is an American stock market index based on the market capitalisation of 500 large companies having common stock listed on the NYSE or NASDAQ. It is one of the most commonly followed equity indices, and many consider it one of the best representations of the U.S. stock market. Thus the closing prices of the index are a representative of the movement of the American markets. We will use Recurrent Neural Networks to predict these closing prices.

2 Data Preprocessing:

Data preprocessing is the first and foremost step in any machine learning problem. The data is a time series of closing prices of the S&P index, sourced from Yahoo Finance.The series starts from January 1950 and has entries up until November 2017. The future movement of these indices depend upon their recent performance in the past. Thus we have prepared the data in non-overlapping slots of say 'n' days. So that the closing prices in $\mathbf{n_t}$ slot will predict the closing prices in the $\mathbf{n_{t+1}}$ slot. We use all the values from $\mathbf{n_0}$ to $\mathbf{n_{t-1}}$ to predict the $\mathbf{n_t}$ slot closing prices. But this doesn't work very well in the long run as the overall prices increase and then are out of scale for the previous years, and the network has to predict numbers never seen before. To solve this problem we normalize the data.

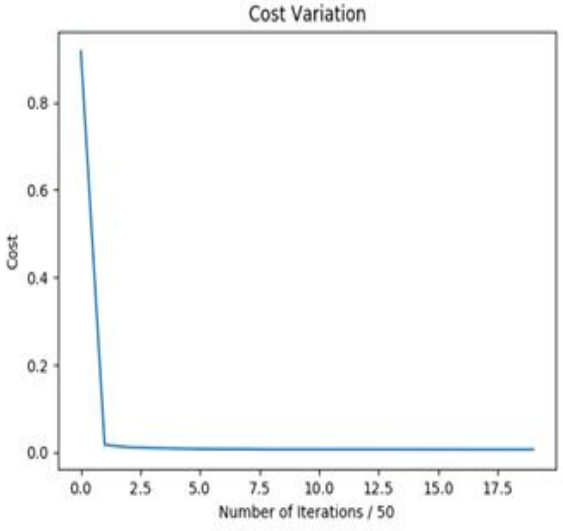
3 Normalization:

Due to growing magnitude of the prices, we see the percent change in the prices and not the actual prices. For that we divide all the prices in a slot with the last price from the previous slot.

4 Long short term memory (LSTM):

Recurrent neural networks are a great way to predict prices as it retains some of the recent inputs. Retaining the inputs models the dependence of stock prices on past behaviour.
But, RNNs suffer from the vanishing gradient problem where the long term dependencies get negligible and fail to affect the current prediction. We use LSTM(Long Short term memory) cells. To train the networks we use Back Propagation through time algorithm. We train RNN in an “unrolled” version so that we don’t need to propagate too far back in the series.

We have used RMSProp optimization.
Below is the plot of the cost(mean squared error) as the model is trained.



5 References:

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
https://towardsdatascience.com/lstm-by-example-using-tensorflow-feb0c1968537
https://r2rt.com/recurrent-neural-networks-in-tensorflow-i.html
https://lilianweng.github.io/lil-log/2017/07/08/predict-stock-prices-using-RNN-part-1.html#input_format_example

TEAM MEMBERS:
Atharva Khandait
Yash Kothari
Vikramaditya Patil
Abhay Jindal