Machine Learning Engineer Nanodegree

Capstone Proposal : Stock Closing Price Prediction

Nitish Kumar
November 13th, 2018

Proposal

Domain Background

Stock market prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange (1). The successful prediction of a stock's future price could yield significant profit. This is one of the oldest field which have a tremendous amount of data, hence the use of Data Science.

Most of the manual analysis and predictions get influenced by the intuitions, which may mislead to bad decisions. This was observed way back in time and since then Algorithmic approach is been is used to predict the stock data for long term as well as short term periods. With the development of the technologies the process of data collection is also developed and as the complete prediction process is based on the data, hence better data leads to better prediction.

These stock data is highly influenced by the human behaviour as even after a good prediction its a humans decision to buy or sell a stock, and these behaviours are also dependent on various other factors as well, like NEWS related to that entity. Another factor that determine the stocks is how the industry and other companies in that domain are performing.

With these many sources of data and complex correlation between them, it becomes mandatory to have a good machine learning algorithm to have a good prediction on the data, like in 1997, the prior knowledge and neural network was used to predict stock price (5). Later in 2009, Tsai used a hybrid machine learning algorithm to predict stock price (6). And now in 2018, popular machine learning algorithms were used to predict stock price such as pattern graph (7), convolutional neural network (8), recurrent neural network (9).

Problem Statement

For this project the problem statement is to predict the closing price of the stock (Apple and Microsoft) of the day using stock details like opening pice, max price and close price, and to check that how much of the news (from New York Times) related to these stocks really effect the predictions. In this the benchmark model will be provided with the data of the particular stock and its news, and will use the polynomial regression to predict the closing price of the stock.

Although it would be really amazing to make a model to predict the closing prices of upcoming days in the future and using Reinforcement Learning to continuously learn the new data, but I am afraid that these are out of the scope of this project.

Datasets and Inputs

For this project the main data of the stock is downloaded from kaggle (2) and only the data of the Apple and Microsoft stocks were used. These data are also available at the Yahoo finance website (https://in.finance.yahoo.com)

Then to get the data of the news related to these I used the New Your Times search_api (3) and the get_news.ipynb notebook. In the notebook I specially searched the news related to Apple and Microsoft, then extracted the title and abstract of the news and concatenated them into a single string and stored them into the AppleNews.csv and MicrosoftNews.csv files respectively, for further processing. If the api returned more than one news on a given date than all the titles and abstract were combined to make the final string and if there were no news than the field was left blank.

For the final data set these news string was processed through the Natural Language Toolkit (specifically SentimentIntensityAnalyzer from nltk.sentiment.vader) which gave me the compound, negative, positive and neutral values of the news sentiment. These data were stored into AppleFinalData.csv (of size (2517,10)) and MicrosoftFinalData.csv (with size (2517,10)) with the stock data.

For the input as training I will be using the data from 2006 Dec to 2014 Dec and remaining for the test. Before final splitting into training and testing I also added one column, Month (contains only the month from the date) so that the model can also analyse the yearly patterns of the Price. Finally Y/ target will contain the 'Adj Close' or 'Price' and X will contain stock data as 'Month', 'Open', 'High' & 'Low' and sentiment data as 'compound', 'neg', 'neu' and 'pos'. These columns will be combined and converted into a NumPy matrix to feed the network. Following will be the shape of the train and test data sets:

- Train X : (2033,8)
- Test X : (483,8)
- Train Y : (2033,1)
- Test Y : (483,1)

Solution Statement

The final solution will be based on the Recurrent Neural Network Structure. To overcome the problem of vanishing gradient problem it will contain LSTM (Long Short Term Memory) layers to store the gradient and have a short history of the stock data, followed by the combinations of the Dense layers to have a reliable prediction. For the single company model the final output will be a single value and for the combine model it will return a vector containing prediction for both of the companies. And the final result will be, the model which performed better than the benchmark model and have a reasonably small error values for the test data set.

Benchmark Model

For the bench mark model I will be using the polynomial regression to predict the data for the stocks using the same data set. This will be implemented using the neural network containing combinations of Dense layers in keras and will have separate model for each company. It will give us the basic idea that how much can a simple regression can predict and what improvements can be made on that.

Evaluation Metrics

For the evaluation metrics relative error of the actual Adjusted Close and final prediction of the model for the testing data set will be used .i.e how reliable is the prediction with respect to the actual data. Where Relative error is:

$$\text{Relative Error} = |\text{Adjusted Close} - \text{Predicted Close}| \div \text{Adjusted Close}$$

Project Design

This project is to make a model that can predict the price of the stocks with the basic details provided. Hence to train the model I gathered the data from Kaggel and the NY times api. The stock data can be downloaded from other sources like Yahoo and Google. To use the news into model I convert it into numbers using the 'nltk' (4) library.

After gathering the data it's time to visualise and understand the data and for that I am using marplot lib. From the graph we can see a yearly pattern but its not that clear, hence this makes me use the months of the year as a training feature. 'Month' is not added in the final data file but the final training matrix will contain it, I will have to split the 'Date' at '-' using split() function and store it into the 'Month' filed. And since I would not be using the 'Close' column, it will also be removed and at last the 'Adj Close' will be

renamed as 'Price'. Then I will rearrange the columns in the data frame and convert it into numpy array using 'pd.DataFrame.values' . Something like following:

```
[3]  apple=pd.read_csv('AppleFinalData.csv')

[4]  apple=apple.rename(columns={'Adj Close':'Price'})

[5]  apple=apple.drop('Close',axis=1)

[6]  apple['Month']=int

[7]  apple=apple[['Open', 'High', 'Low', 'compound', 'neg', 'neu', 'pos',
            'Month','Date','Price']]

[8]  apple.columns

     Index(['Open', 'High', 'Low', 'compound', 'neg', 'neu', 'pos', 'Month', 'Date',
            'Price'],
           dtype='object')

[9]  # add the data into the month column
     for i in range(len(apple)):
       apple['Month'][i]=int(apple['Date'][i].split('-')[1])

[10] apple_x_train=apple.iloc[:2033,:8].values
     apple_y_train=apple.iloc[:2033,9:].values

[11] apple_x_test=apple.iloc[2034:,:8].values
     apple_y_test=apple.iloc[2034:,9:].values

[18] apple_x_train.shape,apple_y_train.shape

     ((2033, 8), (2033, 1))

[19] apple_x_test.shape,apple_y_test.shape

     ((483, 8), (483, 1))
```

Then it's time to select the fields of the data that we are going to use for the benchmark and the final model. For this project I have divided the data into 3 categories:

(1) Only stock data of companies into separate models.

(2) Stock with news data into separate models.

(3) Stock with news data into a single model with a vector output for both companies.

All of these will be compared with the benchmark model to see which one actually works for both of the companies. The benchmark model will use the stock and news data of both the companies into 2 separate models for each of them.

Finally for the main model I am planning to use different combination of the LSTM layers for fully connected network followed by Dense layers with most likely 'Relu' as the

activation function. In the end it will contain a single node layer to get a single output for both companies separately. Currently I am planning to use the 'adam' as a my optimiser for the network, but I will also test it with other optimisers to see which works best with the data. In all the cases I will be keeping the learning rate low, near 0.01 for the initial trails. At the time of the compilation I will also try different loss function combinations with these optimisers, but for the initial I will keep it simple with ' mean_squared_error '. In the training I will also use the 'ModelCheckpoint' to store the best results during the training and this will really help me to decide that which combination of optimiser and loss function actually works for the data. For the debugging I am also going to plot the error value with the number of trails to see that set of hyper-parameters are actually working out or not. Finally the main model will have the minimum of the error in prediction the final price and will obviously perform better than the benchmark model.

Resources and References

(1) Wikipedia : https://en.wikipedia.org/wiki/Stock_market_prediction

(2) Kaggle Stock data set : https://www.kaggle.com/mgkmgk/stock-price

(3) New Your Times: http://developer.nytimes.com/article_search_v2.json

(4) NLTK : http://www.nltk.org/

(5) Kohara, K., Ishikawa, T., Fukuhara, Y., & Nakamura, Y. (1997). Stock price prediction using prior knowledge and neural networks. Intelligent systems in accounting, finance and management, 6(1), 11-22.

(6) Tsai, C. F., & Wang, S. P. (2009, March). Stock price forecasting by hybrid machine learning techniques. In Proceedings of the International MultiConference of Engineers and Computer Scientists (Vol. 1, No. 755, p. 60).

(7) Jeon, S., Hong, B., & Chang, V. (2018). Pattern graph tracking-based stock price prediction using big data. Future Generation Computer Systems, 80, 171-187.

(8) Lee, M. S., Ahn, C. H., Kwahk, K. Y., & Ahn, H. (2018). Stock Market Prediction Using Convolutional Neural Network That Learns from a Graph. World Academy of Science, Engineering and Technology, International Journal of Business and Economics Engineering, 5(1).

(9) TORRES, D. G., & QIU, H. (2018). Applying Recurrent Neural Networks for Multivariate Time Series Forecasting of Volatile Financial Data.