An Application Predicting Stock Price Trends using Neural Network

Yi Man
yi.man@colorado.edu

Han Ngo
han.ngo@colorado.edu

1 Problem Motivation

Stock price trend prediction is an active research area for data scientists. Therefore, in recent years, significant efforts have been put into developing models that can predict for future trend of a specific stock or overall market. Also, some of the researchers showed that there is a strong relationship between news article about a company and its stock prices fluctuations. We create Stockalyzer to help us project the tendency in stock price changes with regards to past datasets and public reactions. Using deep learning model and text mining techniques to forecast the interactivity between change in price and released news will help us foresee stock trend movement. The overall purpose of our work will help clarify such following questions:

1. Given a set of symbols (company indexes), how would the model predict stock price change in a specified time frame?

2. How efficient does the neural network work towards different sets of data? Does it cause too much over-fitting or under-fitting?

2. What would be a good decision in different situations, e.g. if we decided to previously buy the stock and there is an increase in the stock price?

3. Once decision is made, based on historical data source, how does the program generate analysis for us to see if the action yields any profits?

2 Literature Survey

Financial stock analysis is a well-studied area that has seen much prior work. There are approaches that have been utilized to build efficient predictive models from Deep Learning technologies such as Neural Network [1], Support Vector Machines [2], Q-Learning, etc. to identify the correlation between past datasets and current public opinions' impact on stock movements. A much more detailed and well-developed version has been developed by Facebook team using additive models to predict certain sets of stocks in current years. Moreover, collaborative open-source work has also been performed by Kaggle community members and competition candidates interested in exploring this domain. While many of these projects provide bases analysis of stock trends as a function of time, Facebook's Prophet model seems to be the most popular library to be used currently for its animating visualization and high validation accuracy.

3 Proposed Work

Data Collection and Integration

We have collected constituent financials and download the full S&P 500 data from Yahoo! Finance-GSPC to our dataset directory. The Python script load.py will fetch the prices of individual stocks in S&P 500 for 4000 days maximum, each saved to our dataset directory.

Collecting tweets and analyzing sentiments

We will fetch the filtered news from Reuters via the Twitter streaming API. The sentiment analysis takes the input from package then looks up the sentiment using a sentiment analysis API and stores the scores as a new feature in our tables.

Training data preprocessing

We will use content in one sliding windows to make prediction for the next and ensuring there is no overlap between two consecutive windows. Layer inputs and outputs will form an unrolled version of recurrent neural network for training on Tensorflow. After that, for model validation and accuracy evaluation, we will use the latest 10% of data as the testing data.

Data normalization

We will normalize the close prices in each sliding window. The task becomes predicting the relative change rates instead of the absolute values. In a normalized sliding window at time t, all the values are divided by the previous price to help prevent out-of-scale issues.

Model construction

The model is expected to learn the price sequences of different stocks in time. Due to the different underlying patterns, we will use embedding. We will create a Tensorflow graph using input variables from datasets that we have and the available optimizing algorithm built in TF library for gradient descent optimization

Training/Testing

We will initiate different hyperparameters, start a running tf session, and saved our train model at the end which will be later used to evaluate on the last 10% of our datasets.

There are a lot of approaches that machine learning enthusiasts pursue to forecast stock trends as the aforementioned in literature survey section. For the sake of learning the whole process of data mining, we decided to try Neural Network approach. Moreover, reading researches about the relevance between public opinions and stock movement motivates us to discover add on a new feature to the training data sets.

4 Dataset Source

https://raw.githubusercontent.com/datasets/s-and-p-500-companies/master/data/constituents-financials.csv/
https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC/

5 Evaluation Methods

We will use mean square error as the loss metric and the RMSPropOptimizer algorithm which is built in Tensorflow for gradient descent optimization

6 Tools

For communication and sharing purposes, our group will work on the project through Github. The code will be mostly based on Python. We plan to utilize Pandas and Numpy libraries to aid data cleaning process. In addition, we plan to use WEKA for text analysis, Tableau and Matplotlib for basic graphing visualizations. Most interactions will be based on Ubuntu CLI and data training process will also make use of our GPU instance on AWS.

7 Milestones

March 15th	Data collection, preprocessing + cleaning
March 30th	Data integration + normalization
April 7th	Model construction + set up TF environment Work on progress report
April 14th	Training + testing on our dataset
April 21th	Finalize visualizations, write up our findings
April 28th	Finish group presentation slides and report

8 Summary of Peer Review Session

We were advised to give more elaborations on each step we do, based on the first general presentation in class. Compared to other group projects, we lean towards more of a financial market using predictive Machine Learning model to forecast stock trends while other groups in the same cluster use more of regression models and displaying results as an overlaying heatmap.

REFERENCES

[1] Monik V.2017. RNN-LSTM in stock prediction. (2017). http://monik.in/a-noobs-guide-to-implementing-rnn-lstm-using-tensorflow/

[2] Ruchi Desai. 2015. Stock Market Prediction Using Data Mining. (2015). https://www.ijedr.org/papers/IJEDR1402237.pdf

[3] Mark Dunne. *Stock Market Prediction*,

[4] Sarvajanik Sudeep. 2017. A simple deep learning model for stock price prediction using tensorflow. *Medium*(2017).