Capstone Proposal

Kazuma Kamiaka

May 4th, 2017

Domain Background

In recent years, many trading firms use automated trading systems and look for better algorithm. Machine learning algorithm is one of the powerful option. For example, Yudong Zhang used neural network to predict S&P 500[1]. Some approach uses sentiment analysis. Johan Bollen used Twitter Sentiment Analysis to predict the daily up and down changes in the closing values of the DJIA. [2]

In this research, I try to predict S&P 500 companies historical stock prices using neural net approach. The reason why I've chosen this area is that I'd like to put my money in investment vehicle.

Problem Statement

In this research, I try to build one day ahead prediction system for stock market. I can use historical daily stock prices of S&P 500 companies to build system. Most of data spans from 2010 to the end 2016.

Datasets and Inputs

I use S&P 500 company's historical stock prices with fundamental data as datasets. Prices were fetched from Yahoo Finance and spans from 2010 to the end 2016. All daily data contains opening price, closing price, lowest price, highest price, and volume. Fundamentals are from Nasdaq Financials, extended by some fields from EDGAR SEC databases. It contains enough data to derive most of popular fundamental indicators, for example, After Tax ROE, Capital Surplus. This data set is oftained from Kaggle. [3]

This data set is used as training data of machine learning model. Specifically, I build model which can predict one day ahead stock price by given historical stock prices and fundamental data. This input data is appropriate because it is well known that almost all analysis of the financial markets bases on historical prices or fundamental data.

Solution Statement

The solution which I'll use in this research is “Long short-term memory” (LSTM). LSTM is a special kind of a recurrent neural network (RNN) proposed in 1997. [4] It can learn long-term dependencies much better than traditional RNN can. It is used not only in

time series prediction area, but in machine translation area, such as google translate. [5]

Benchmark Model

In this research, I'll use three very simple time-series analysis models as benchmark. First models outputs the historical mean value. Second models outputs the very last observation out. Third models outputs result of simple linear regression. If possible, I'll use an autoregressive integrated moving average (ARIMA) model as benchmark in addition to above three.

Both the benchmark model and the solution model output numeric as predicted value. So the performance can be measured by comparing these output and correct value. The metrics I'll use is mentioned in next section.

Evaluation Metrics

I'll use Root Mean Squared Error (RMSE) as evaluation metrics. RMSE is one of the most common metrics used to measure accuracy for time series prediction. RMSE means square root of the average of squared differences between prediction value and correct value. The formula which shows RMSE is as follows.

$$\sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \quad (1)$$

Project Design

A theoretical workflow for approaching a solution can be carried out just like previous project of this nanodegree.

First, I'll make a cursory investigation about S&P 500 datasets. This process is called as Data Exploration and it'll help me better understand about data and problem. In this process, I'll use visualization and calculate statistics about dataset.

Second step is preprocessing. The type of data preprocessing should be determined in data exploration process. For example, dropping null value, excluding outliers, normalizing data to 0-1 scale, and so on.

In third step, I'll build benchmark models mentioned in Benchmark Model section. After building benchmark models, I'll calculate RMSE of each models. The final model built in this research has to be at least better than these benchmark models.

The objective of last step is building better prediction model. This step is expressed as following and carried out recursively until I'm satisfied with the result.

* Choose input data. (feature engineering included)
* Building LSTM layers and choose hyperparameter.
* Training and Testing
* Compare result to benchmark model and consider what to be improved.

Reference

[1] http://www.sciencedirect.com/science/article/pii/S095741740800852X

[2] https://arxiv.org/abs/1010.3003

[3] https://www.kaggle.com/dgawlik/nyse

[4] https://www.researchgate.net/publication/13853244_Long_Short-term_Memory

[5] https://arxiv.org/abs/1609.08144