Stock Recommandation System

Model

UjashKumar Leuva ujashkumar.leuva@student.uts.edu.au

# Project Introduction

"Attention is a scarce cognitive resource" − Daniel Kahneman

Literature on attention and its asset pricing implications suggests that people pay attention to stocks that appear in the media and people tend to buy attention-grabbing stocks. Stock recommendations from the TV show Mad Money and Wall $treet Week have proven to catch people’s attention and create price pressure on the stocks recommended, triggering a positive overnight excess return followed by price reversal in 2-5 days as was found in Beltz and Jennings (1997) and Engelberg, Sasseville, and Williams (2012). This is the so-called "price pressure hypothesis." Barber and Odean (2008) shows that since investors face a formidable search problem in buying stocks but not in selling stocks, people buy stocks that are recommended in the media while there isn’t a huge media effect on negative stock recommendations. Accurately recommendation of the stocks is a complex task as there are millions of events and pre-conditions for a particular stock to move in a particular direction. So we need to be able to capture as many of these pre-conditions as possible. We also need make several important assumptions:

1) markets are not 100% random

2) history repeats

3) markets follow people’s rational behaviour

4) the markets are ‘perfect’.

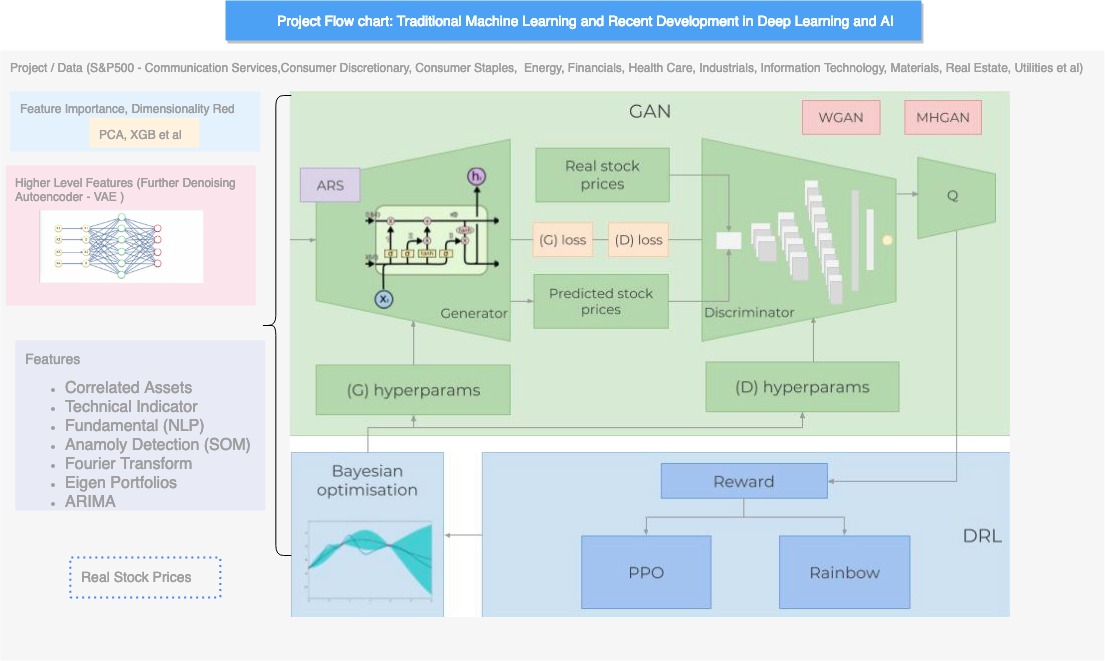
The objective of this Project is to develop stock recommendation system of s&P500 Organisation. For this purpose, I will use the daily closing price from 1st January, 2013 to 31st December, 2018 (4 years for training purpose and one years for validation purpose).

We will use a lot of different types of input data. Along with the stock’s historical trading data and technical indicators, we will use the newest advancements in NLP (using ‘Bidirectional Embedding Representations from Transformers’, BERT, sort of a transfer learning for NLP) to create sentiment analysis (as a source for fundamental analysis), Fourier transforms for extracting overall trend directions, stacked auto- encoders for identifying other high-level features, Eigen portfolios for finding correlated assets, autoregressive integrated moving average (ARIMA) for the stock function approximation, and many more, in order to capture as much information, patterns, dependencies, etc, as possible about the stock. As we all know, the more (data) the merrier. Predicting stock price movements is an extremely complex task, so the more we know about the stock (from different perspectives) the higher our changes are. For the purpose of creating all neural nets we will use MXNet and its high-level API — Gluon, and train them on multiple GPUs and Tensor Flow as Backend.

# Data Retrieval and feature discussion

We need to understand what affects whether the S&P500 stock prices will move up or down. We are using Quandl, for retrieving all the S&P500 organisations stock prices for last five years (January 1st, 2013 to December 31st, 2018) including all the sectors i.e. S&P500 - Communication Services, Consumer Discretionary, Consumer Staples, Energy, Financials, Health Care, Industrials, Information Technology, Materials, Real Estate, Utilities etc.

The Project Architecture is shown below:



**Figure 1: Overview of the complete architecture**

We need to incorporate as much as data as possible (depicting the stock from different aspects and angles). We will use daily data- 4 years of data to train the various algorithms and predict the next 365 days (test data). Then we will compare the predicted results with the test data. For forecasting the stocks prices, we need to use features that are listed below:

1. **Correlated Assets:** Correlated Assets are other types of such as Commodities, FX, Indices or Fixed income securities. A S&P 500 Organisations, doesn’t ‘live’ in an isolated world- it depends on, and interact with, many external factors, including competitors, clients, the global economy, turnover, the geo-political situations, fiscal and monetary policies, access to capital, etc.
2. **Technical Indicators:** Almost all the analysts and investors are follows the technical indicators which can be used as independent features. For Example, we will take- 7 and 21 days moving average, Exponential moving average, momentum, Bollinger bands, MACD.
3. **Fundamental Analysis:** A very important feature indicating whether a stock might move up or down. There are two features that can be used in fundamental analysis: 1) Analysing the company performance using 10-K and 10-Q reports, analysing ROE and P/E, etc (we will not use this), and 2) News — potentially news can indicate upcoming events that can potentially move the stock in certain direction. We will read all daily news for S&P500 Organisations and extract whether the total sentiment about S&P 500 Organisations on that day is positive, neutral, or negative (as a score from 0 to 1). As many investors closely read the news and make investment decisions based (partially of course) on news, there is a somewhat high chance that if, say, the news for S&P500 Organisations today are extremely positive the stock will surge tomorrow. One crucial point, we will perform feature importance (meaning how indicative it is for the movement of S&P500 Organisations) on absolutely every feature (including this one) later on and decide whether we will use it. More on that later. For the purpose of creating accurate sentiment prediction, we will use Neural Language Processing (NLP). We will use BERT — Google’s recently announced NLP approach for transfer learning for sentiment classification stock news sentiment extraction.
4. **Fourier transforms:** Along with the daily closing price, we will create Fourier transforms in order to generalize several long- and short-term trends. Using these transforms we will eliminate a lot of noise (random walks) and create approximations of the real stock movement. Having trend approximations can help the LSTM network pick its prediction trends more accurately.
5. **Autoregressive Integrated Moving Average (ARIMA):** This was one of the most popular techniques for predicting future values of time series data (in the pre-neural networks ages). Let’s add it and see if it comes off as an important predictive feature.
6. **Stacked Auto-encoders**: most of the aforementioned features (fundamental analysis, technical analysis, etc) were found by people after decades of research. But maybe we have missed something. Maybe there are hidden correlations that people cannot comprehend due to the enormous amount of data points, events, assets, charts, etc. With stacked auto-encoders (type of neural networks) we can use the power of computers and probably find new types of features that affect stock movements. Even though we will not be able to understand these features in human language, we will use them in the GAN.
7. **Deep Unsupervised learning for anomaly detection in options pricing**. We will use one more feature — for every day we will add the price for 90-days call option on S&P500 Organisation stocks. Options pricing itself combines a lot of data. The price for options contract depends on the future value of the stocks (analysts try to also predict the price in order to come up with the most accurate price for the call option). Using deep unsupervised learning (Self-organized Maps) we will try to spot anomalies in every day’s pricing. Anomaly (such as a drastic change in pricing) might indicate an event that might be useful for the LSTM to learn the overall stock pattern.

Next, having so many features, we need to perform a couple of important steps: Perform statistical checks for the ‘quality’ of the data. If the data we create is flawed, then no matter how sophisticated our algorithms are, the results will not be positive. The checks include making sure the data does not suffer from hetero-skedasticity, multi-collinearity, or serial correlation. Create feature importance. If a feature (e.g. another stock or a technical indicator) has no explanatory power to the stocks we want to predict, then there is no need for us to use it in the training of the neural nets. We will using XGBoost (eXtreme Gradient Boosting), a type of boosted tree regression algorithms. As a final step of our data preparation, we will also create Eigen portfolios using Principal Component Analysis (PCA) in order to reduce the dimensionality of the features created from the auto encoders.