CS230

Deep Learning Prediction on Price Movement of NASDAQ

Will Vithayapalert
Department of Management Science & Engineering
Stanford University
nopsinth@stanford.edu

Abstract

Throughout recent years, deep Learning techniques have been applied to predict the price movement of securities in order for investors to create more powerful trading strategies. This paper aims to explore how Recurrent Neural Network (RNN) models can inform investors to more likely make a winning decision. The author investigates Invesco QQQ Trust Series 1 as a proxy of NASDAQ index. The dataset includes daily QQQ's prices since its inception in 1999 and augmented technical indicators. With hyperparameter tuning and regularization, Multi-layer GRU appears to outperform other models due to its highest Recall and high AUC score.

1 Introduction

Stock price prediction has been one of the well-known techniques that investors use to make investment decisions, as the financial market is highly erratic. In contrast to the Efficient Market Hypothesis where no investor is believed to be able to take advantage of the short-term fluctuations, short-term market timing suggests that investors can make profits based on the knowledge that foresees the price change. In this paper, we aim to create a model that can accurately predict the price movement of a well-known Exchange Traded Fund (ETF) named Invesco QQQ Trust Series 1, which replicates the performance of NASDAQ.

Unlike S&P500 whose price movement has been analyzed by a lot of existing research, Invesco QQQ Trust Series 1 has been relatively less studied since this index only tracks around 100 companies. However, the performance of QQQ is less weighted by some companies with very large market caps, as its composite is formed by a certain rule that excludes large financial companies. Therefore, QQQ can be an ideal proxy for the performance of non-financial companies, most of which are rapidly-growing tech companies, as well as the health of the U.S. financial market. The predictability of the model can reflect how well investors would capture the opportunities to make profits from the market.

Among various machine learning techniques, we believe that an accurate prediction can be achieved by an effective Recurrent Neural Networks (RNN's) that incorporates historical information. Therefore, this project aims to use RNN's models to predict whether the price of QQQ index will go up or down over a specific time horizon, using the information regarding the daily price and technical indicators. To be specific, provided the time-series values, we will predict whether P(t+s) is above or below P(t) where s is a look-forward window. (If the price on the next s periods increases, the response variable would be labeled to be 1 and 0 otherwise).

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

With the increase of deep learning applications in stock price prediction, Long Short Term Memory (LSTM) model is considered one of the models with high predictive power. There is a widely accepted conclusion that only a simple linear model cannot capture the non-linearities of stock prices, as the stock market is highly volatile and driven by erratic investor sentiment. Sequence models as LSTM is well-suited to use financial time-series data to recognize the trend of the data, as it incorporates historical information to compute the outputs.

However, most of research frame the stock price prediction to be a regression problem, which doesn't obviously assist investors to make a decision. This paper attempts to instead frame the task to be classification which can quickly and thoughtfully inform investors to buy or sell the security. When it comes to be a classification problem, it is interesting to investigate whether sophisticated models as RNN can outperform simple logistic regression.

3 Dataset and Features

The dataset consists of QQQ's daily Closing price and the other a number of technical indicators that are augmented based on the close price. These indicators represent volatility, momentum, and trending strength of price movement. Each technical indicator will be elaborated in details later in this section.

Closing Price: the first features included is a close price since it indicates an opportunity point where investors would buy or sell during the trading day. If the current price increases as anticipated, an investor might make profits from "sell" action today.

Trading Volume: this feature reflects how many shares of ETF are traded during the day, which can roughly indicate investor sentiment towards a particular security. If Volume is high, it can reflect high interest in that security.

Moving Average: this indicator measures trendiness of the price, widely used in trend-following strategies to determine both buy and sell signals. Moving Average can be represented in two measures: Simple Moving Average (SMA) and Exponential Moving Average (EMA), which can be calculated as following. We also create SMA and EMA over many different lookback periods (5, 10, 12, 15, 26, 50, 100 days) to observe how long the future price is related to historical prices.

$$SMA_n = \frac{P_t + P_{t-1} + \ldots + Pt - n}{n} \quad n : \text{lookback window}$$
$$EMA_n = (P_t - EMA_{n-1}) \cdot \frac{2}{n+1} + EMA_{n-1} \quad n : \text{lookback window}$$

Moving Average Convergence Divergence (MACD): this indicator shows the relationship between two moving averages of the securities, which can be calculated by the difference between the exponential moving average over 12 days and the exponential moving average over 26 days.

$$MACD_t = EMA_{12} - EMA_{26}$$

Binary response variable: the indicator of whether the price on the next day increases.

$$y_i = \begin{cases} 0 & P_{t+1} < P_t \\ 1 & P_{t+1} \geq P_t \end{cases}$$

4 Methods

After data preprocessing step, our dataset have 4974 data points and 16 covariates. We split the dataset into train, dev, and test set by periods. The recent periods should be included in the test set, as they most represent how the current stock market goes. The train set lasts from 1999 to 2013, and the dev set contains data only in 2014. Meanwhile the test set is the data from 2015 to 2018. Each dataset is served for different purposes. Specifically, we use the train set to build RNN models, the dev set for hyperparameter tuning, and the test set for evaluating the performance of models.

4.1 Loss Function

As the goal is to predict whether the future price goes up or down, we use the binary cross entropy as the loss function to reflect the nature of binary classification.

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log\left(p_i\right) + (1-y_i)\log\left(1-p_i\right)\right] = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log\left(p_{ij}\right)$$

4.2 RNN Architecture

In this project, we have built 3 deep learning models: single-layer LSTM, multi-layer LSTM, and multi-layer GRU. These three models have 128 hidden units in each layer with RELU activation function. We also regularized these models with early stopping and drop-out techniques. For the single-layer model, we applied 50% dropout between LSTM block and FCC layer. Meanwhile, we applied 50%, 20%, and 70% dropout to 3-layer models respectively. The details of deep-learning architecture are illustrated below.

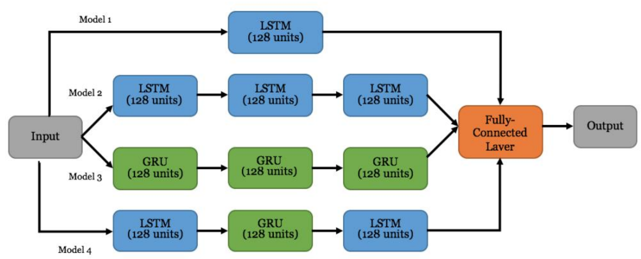

Figure 1: 4 Different RNN Models

4.3 Hyperparameter Tuning

After the models were successfully constructed, we also tuned hyperparameters to produce the best results by using Grid Search method. The tuned hyperparameters include learning rate, batch size, number of hidden units, and number of epochs. We constructed the grid that contains different values of those 4 hyperparameters and employed GridSearchCV function from sci-kit learn package to perform the task. The values filled in the grid are listed below.

Learning rates varying over log scale: 0.0001, 0.001, 0.01, 0.1
Batch sizes: 32, 64, and 128
The number of epochs: 10, 50, and 100
The number of hidden units in each layer: 64, 128, 256, 512

Furthermore, we also tuned different optimizers (RMSprop, Adam, and Adaptive Subgradient) to choose the optimizer that produces the best result. The best performing optimizer is Adam Optimization.

5 Results and Discussion

As a reference, our baseline model is a linear classifier featured with Ridge and Lasso regularization. We have tested total 7 models including 3 logistic regression and 4 RNN models to predict the probability where the next day's price goes up. After fitting all models, we performed hyperparameter tuning by using Grid Search. The best hyperparameters are summarized as following:

The best learning rate is 0.001.
The optimal batch size is 32.
The best number of epochs is 10.
the optimal one is 128 hidden units.

Then, we use 4 evaluation metrics to select the most predictive model. Our results demonstrate that the models with LSTM tend to underperform ones that are less complex, as they suggest that logistic regression performs better in general. Meanwhile, GRU relatively has higher recall and AUC Score, compared to others. If every metric is equally weighted, the best model will be logistic regression with ridge regularization. However, in the stock prediction, investors care about precision and recall, as every true positive can potentially generate substantial excess returns.

Model	Accuracy	Precision	Recall	AUC Score
Logistic Regression	0.6066	0.6066	0.6066	0.605
Logistic Regression (Ridge)	0.6318	0.6318	0.6318	0.677
Logistic Regression (Lasso)	0.5	0.556	0.556	0.5
Single-layer LSTM	0.6089	0.6086	0.78313	0.6318
Multi-layer LSTM	0.543	0.6860	0.34217	0.611
Multi-layer GRU	0.583	0.5766	0.9518	0.601
Stacked GRU and LSTM	0.5914	0.578	0.9398	0.5828

6 Conclusion and Future Works

In this project, we have built multiple deep learning models with various architecture and three traditional classifiers as our baseline models in order to predict whether QQQ's price on the next day will increase or decrease. Out of 7 models, 3-layer GRU is the best model that allow investors to make the most accurate prediction, as it relatively has high AUC Score and highest recall. Our experiments that we used single-layer LSTM, 3-layer LSTM and stacked LSTM with GRU demonstrate that more complexity added in the architecture tends to worsen the performance. This is sensible because our dataset used still has too few observations. Thus, highly complex models might be skewed when we enforced them to capture the trend of the data over relatively short periods. In contrast to other regression problem regarding stock predictions, deep learning methods in classification can be less accurate.

For further improvements, our model currently incorporates only trend-following technical indicators. However, the stock prices also depend on news and various macroeconomic factors. Therefore, for future works, we would like to input more features including the factors that measure investor sentiment in the market as well as relevant fundamental characteristics such as Book Value/Market Value or Price to Earnings Ratio.

In addition, we suspect that our dataset used in this project might have too few data points that allow some sophisticated deep learning models to capture the "trend" of prices historically. We might try using larger dataset, which consists of approximately millions of data points (stock prices in seconds or minutes) to explore how well LSTM or GRU can make prediction.

7 Contributions

Since this project has been done by only me, I have fully contributed to all steps of the project including data preprocessing and building all deep learning architecture.

References

[1] Girbal, R., Stronov, G. (2019) Predicting stock price dynamics using stacked GRU's and LSTM's.

[2] Contesti, R., Hurubaru, S., Yan, J. (2019) Deep Learning for relative stock price prediction.

[3] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). doi:10.1109/icacci.2017.8126078

[4] Faustryjak, D., Jackowska-Strumillo, L., Majchrowicz, M. (2018). Forward forecast of stock prices using LSTM neural networks with statistical analysis of published messages. 2018 International Interdisciplinary PhD Workshop (IIPhDW). doi:10.1109/iiphdw.2018.8388375