Deep Learning Prediction on Price Movement of NASDAQ

Will Vithayapalert

Overview

Motivation: The price movement of Invesco QQQ Trust Series Series 1 has been less studied, compared to that of SPY. QQQ is one of the most common ETFs included in portfolios, especially for ones that focus on companies in a technology sector. To develop the model that can accurately predict the QQQ's future prices, investors will potentially form profitable trading strategies that generate excess returns from this index.

Approach: We frame this problem to be a classification task (whether the price on the next day increases). Using logistic regression as a baseline model, we employ multiple RNN models namely single-layer and multi-layer LSTMs, and multi-layer GRU.

Dataset and Features

The dataset consists of QQQ's daily Close price and the other a number of technical indicators that are augmented based on the close price. These indicators include Simple Moving Average, Exponential Moving Average over different historical period (10, 12, 15, 26, 50, 100 days), Moving Average Convergence Divergence (MACD), and Trading Volume.
The response variable is binary and can be augmented as following:

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

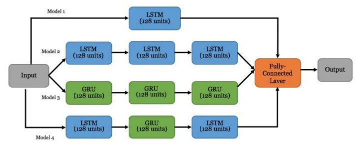
0000

Loss function and Evaluation metrics

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. To evaluate the performance of the model, we choose multiple metrics including accuracy, precision, and area under ROC curve (AUC score).

Models

As a reference, our baseline model is a linear classifier featured with Ridge and Lasso regularization. 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. The details of deep-learning architecture are illustrated below.



Results

The results of both baseline and deep learning models are summarized in the table below.

Model	Accuracy	Precision	Recall	AUC Score
Logistic Regression	0.6066	0.6066	0.6066	0.605
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

Hyperparameter Tuning

After the models were successfully constructed, we also tuned hyperparameters to produce the best results measured by AUC score by using Grid Search method. The tuned hyperparameters include learning rate, batch size, number of hidden units, and number of epochs.

Learning rate = [0.0001, **0.001**, 0.01, 0.1]
Batch size = [32, **64**, 128]
Number of Epochs = [**10**, 50, 100]
Number of hidden units in each layer = [64, **128**, 256, 512]

Best performing hyperparameters were bolded.

Furthermore, we also tuned different optimizers to choose the optimizer that performs best.

Best Optimizer: **Adam Optimization**

Future works

Further improvement:
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.