Large Language Models and Safe Reinforcement Learning For Short-Term Stock Trading

Stanford CS229 Project
Project Category: Theory & Reinforcement Learning
Project Mentor: Jeff Z. HaoChen

Irina Alexandra Marton
SUNet ID: imarton
Stanford University - Department of Computer Science
`imarton@stanford.edu`

Abstract

Exploring the potential of a system of models for short-term stock trading. We use finBERT to asses stock market sentiment, and then feed the sentiment to a Safe DRQN model for training. Further experiments are needed with a larger training dataset and with longer training to fully probe the ability of the proposed system to generate significant profit.

1 Introduction

There have been many studies regarding the implementation of machine learning for predicting stock market returns. While recent research has shown that large language models have a higher potential for predicting stock market movements than previous models, they are still not enough on their own. Therefore, according to Lopez-Lira and Tang (2023), we need to investigate if stock market price fluctuations can be predicted based on stock market news headlines with the use of large language models enhanced with other machine learning techniques and/or models.

For the experiment, **finBERT** is used to perform sentiment analysis on the **Key Developments** dataset, then merging the sentiment dataset with the **Yahoo Finance Daily Updates OHLC** dataset we create the training dataset for the reinforcement learning algorithm **Safe financial DRQN**. The trained model will output the predicted profit/loss for *hold, buy* and *sell* actions.

2 Related work

2.1 Sentiment analysis with Large Language Models

Lopez-Lira and Tang (2023) document a positive correlation between the sentiment score output by ChatGPT on news headlines and daily stock market returns. Moreover, Araci (2019) and Wu et al. (2023) agree that using a LLM trained on financial data outperforms other models when performing financial tasks such as sentiment analysis of financial news headlines. Therefore, the use of BloombergGPT or finBERT, which are LLMs trained on financial documents, will greatly improve the capability of an AI trading agent to make good trading decisions.

2.2 Reinforcement learning

The proposed modifications by Huang (2018) create a deep recurrent Q-network learning algorithm suitable for financial trading that delivered on average a 23.8% return on investment on the test dataset during the experiment. On the other hand, Thomas et al. (2022) developed a formula and algorithm that creates a reward penalty framework that heavily penalizes undesired states, further improving the performance of reinforcement models. By applying the safe reinforcement reward policy to Huang's financial algorithm, the resulting model has the potential to outperform all existing trading models.

3 Dataset and Features

3.1 Key Developments Dataset

The dataset contains more than 100 types of important events for a public company including executive changes, changes in corporate guidance, delayed filings, and SEC inquiries gathered from 20,000 news sources including press releases, regulatory filings, company websites, web mining, investor Conference Organizer Websites, and call transcripts. Each item in the dataset includes the announced date, headline, situation summary, type, source, and other identifiers. Duplicates of the headlines were kept in order to feature engineer the feature 'virality' that attempts to measure the amount of media exposure the mentioned company had during each day.

3.2 Yahoo Finance Daily Updates OHLCV Dataset

Yahoo Finance Daily Updates OHLCV Dataset is a database that contains all important market information on all public companies. The items selected were closing price, adjusted closing price, high, low, open price, and volume for each day.

3.3 Training Dataset

The training dataset contains all information from both selected datasets for the period spanning from 1-1-2022 to 1-1-2023, a total of 290 trading days for the selected public company. After processing the **Key Developments** and **Yahoo Finance Daily Updates** datasets, when merging the two, the following assumptions were made:

- for the days where there were multiple headlines per day, a mean sentiment was computed
- for the days where the market was closed, but there still were news reports, the sentiment for the next open day was considered to be the mean sentiment of all closed days along with the next open day sentiment
- if an open market day had no headlines, it was assumed that no change in sentiment occurred and the sentiment for the previous trading day was copied
- if the first trading day from the dataset had no news reports, a value of 0 as the base sentiment was inserted

The training dataset will have the following format:

date	open	high	low	close	adj close	volume	pos	neg	neutr	vir
2022-01-8	177.83	182.88	177.71	182.01	180.43	1192700	0.25	0.06	0.67	19

However, before feeding this dataset into the model for training it is preprocessed as follows:

- The volume column is normalized and now contains values between 0 and 1.
- The "virality" column is normalized and now contains values between 0 and 1.
- The date column is one-hot encoded and now contains an integer $\in \{0, ...290\}$
- The open, high, low, close, and adjusted close columns have been normalized together and now contain values between 0 and 1.

The preprocessed training dataset will have the following format:

date	open	high	low	close	adj close	volume	pos	neg	neutr	vir
0	0.912	0.998	0.973	1	1	0.479	0	0	0	0

3.4 Test Dataset

The test dataset contains all information from both selected datasets preprocessed as before, for the period spanning from 1-1-2023 to 1-3-2023, a total of 38 trading days for the selected public company.

4 Methods

4.1 finBERT

Even though the total corpora size of finBERT is 4.9B tokens compared to BloombergGPT's which has 363B tokens, making BloombergGPT the superior LLM, finBERT is made available for use by ProsusAI through Huggingface and thus was chosen for the experiment. With the use of HuggingFace Transformers, the pre-trained model finBERT and its tokenizer is loaded, then used to convert the Key Developments dataset into sentiment predictions, outputting positive, negative, and neutral sentiment values.

4.2 Financial S-DRQN

When investigating methods of applying reinforcement learning to stock trading, it was decided to use Huang's (2018) Financial DQRN with minor adjustments, resulting in the **Financial S-DRQN** algorithm. The most significant modification is the implementation of the terminal cost *C* for undesired outcomes as defined by Thomas et al. (2022).

4.2.1 Algorithm

```
Algorithm Financial S-DRQN Algorithm
Initialize $T \in \mathbb{N}$, recurrent Q-network $Q_\theta$, target network $Q_{\theta^-}$ with $\theta^- = \theta$,
    dataset $\mathcal{D}$, environment $E$ and $\epsilon = 1$
Simulate env $E$ from dataset $\mathcal{D}$
Observe initial state $s$ from env $E$
for each episode do
    for each time step do
        Select $\epsilon$ greedy augmented action w.r.t. $Q_\theta(s, a)$ and apply to env $E$
        Receive safe reward $r$ and next state $s'$ from env $E$
        Form $T = (s, a, r, s')$ and store $T$ to memory $\mathcal{D}$
        if memory $\mathcal{D}$ is larger than batch size then
            Sample a sequence of batch size length from $\mathcal{D}$
            Train network $Q_\theta$
            Decay $\epsilon$
        end if
        Soft update target network $\theta^- \leftarrow (1 - \tau)\theta^- + \tau\theta$
    end for
end for
```

Where T is a day with all its trading features, E is the merged dataset, and the reward function is the returns at each time step according to the reward penalty framework. The model will make short-term decisions of buy, sell, or hold and be evaluated by the total profit at the end of the episode.

4.2.2 Action Space

A simple discrete action space was chosen: labels *Hold, Buy, Sell* are one-hot encoded to 0, 1, 2. In order to simplify the trading environment, the model is allowed to hold only 1 stock in inventory. Therefore, after buying, the model is forced to choose the action for its 2nd highest predicted reward (either sell or hold) if the prediction was again buy. Similarly, if the prediction was *sell* and the inventory is empty, the model is forced to either buy or hold.

4.2.3 Reward

As *Safe Reinforcement* focuses on planning ahead for avoiding unsafe states, it is a perfect fit for mitigating the risk of undesirable investing outcomes (losses). By defining the state where we obtain a profit lower or equal to 0 and the state where the model refuses to engage in trading as the unsafe state, we have the following reward penalty framework:

$$(\tilde{r}(s, a), \tilde{T}(s, a)) = \begin{cases} (r(s, a), T(s, a)) & \text{if } s \notin \mathcal{S}_{unsafe} \\ (-C, s) & \text{if } s \in \mathcal{S}_{unsafe} \end{cases}$$

$$(r(s,a), T(s,a)) = \begin{cases} c_t - o_t & \text{if } a \text{ is Buy} \\ p - (c_t - o_t) & \text{if } a \text{ is Sell} \\ \begin{cases} c_t - o_t & \text{if } \mathcal{I} > 0 \\ -(c_t - o_t) & \text{if } \mathcal{I} < 0 \end{cases} & \text{if } a \text{ is Hold} \end{cases}$$

Here, c_t and o_t are the closing and open prices at timestep t, and $p = o_t - o_b$ is the profit obtained by subtracting from the open price at timestep t, the open price of when the stock was bought. $\mathcal{I}$ represents the length of the inventory. The terminal cost $C \in \mathbb{R}$ was calculated according to the formula developed by Thomas et al. (2022):

$$C > \frac{r_{max} - r_{min}}{\gamma^H} - r_{max}$$

Since all the data from the training dataset is normalized, the r_{max} and r_{min} are always 1 and -1 respectively. Despite the model being trained for short-term trading, we still care about the long-term reward and γ will be ≈ 1. However, since our focus is short-term trading, the horizon will be 1. Therefore C must be greater than 1, and our unsafe state rewards should be lower than -1.

5 Experiment

5.1 Model Architecture

The model architecture is slightly different from the one proposed by Huang (2018), as the intention is to make the model focus on short-term trading, rather than long-term, therefore the LSTM layer was changed to a dense layer while maintaining the same number of units.

Model Architecture		
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	2816
dense_1 (Dense)	(None, 256)	65792
dense_2 (Dense)	(None, 256)	65792
dense_3 (Dense)	(None, 3)	771
		Total params: 135,171
		Trainable params: 135,171
		Non-trainable params: 0

Hyperparameters	
optimizer	adam
loss	mse
batch size	16
learning rate	0.00025
memory size ($\mathcal{D}$)	480
discount factor (γ)	0.99
terminal cost (C)	4
target network (τ)	0.001
ϵ decay rate	0.95

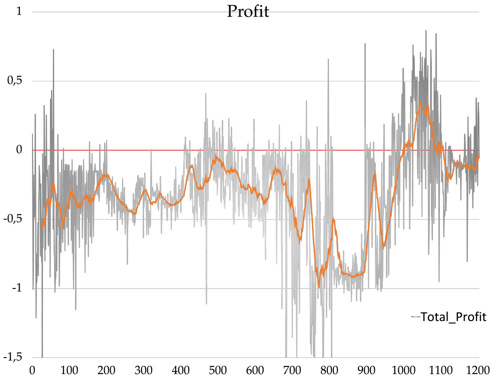
5.2 Hyperparameters

Some of the hyperparameters were chosen according to the Huang (2018) experiment, while the others were selected via an expedited grid search.

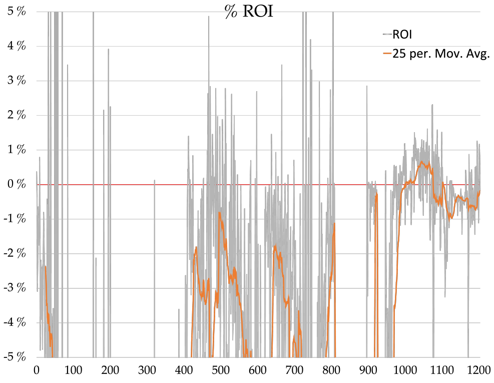
5.3 Results

The experiment was run for only 1200 episodes due to machine limitations. As the objective was to formulate a system of models that can generate profit when engaging in short-term stock trading, our primary financial metrics are profit and % return on investment. While we can observe the modest results obtained so far from the graphics generated with the training data for each episode, we can also make a few other observations:

- Overall the trend is ascending, therefore we might get better results with longer training.
- Despite the fact that, as expected from a reinforcement learning model, there are exploratory moves, and the algorithm implements a reward-to-go strategy, there still is a lot more exploration than anticipated.
- Further hyperparameter tuning might be necessary.



Profit per Episode



Return on Investment per Episode

Below we have the model evaluation metrics. The model was evaluated using 4 different datasets. As our model was trained exclusively on Apple Inc. data, 2 of our test datasets are Apple Inc data, while Alphabet Inc. was added to test the model's ability to generalize to other companies what it learned.

	Training	AAPL Jan-Feb 2023	AAPL Jan-March 2023	GOOGL Jan-Feb 2023
loss	0.0052	0.0126	0.0072	0.0175
accuracy	0.0280	0.0270	0.0000e+00	0.0000e+00
mae	0.0550	0.0929	0.0558	0.0956
% roi	1.4550	14.0990	6.4330	0.9040

From the metrics, we can conclude that the model did slightly overfit to the training data, as despite it outputting better financial results, its loss, accuracy, and mae are inferior. This disparity between the model evaluation metrics and financial evaluation metrics can be due to overall market fluctuation as in the training time interval the AAPL stock had a fall of -26.8%, while in the Jan-Feb 2023, it had an increase of 11.9% and a 26.91% increase in the interval Jan-March. Surprisingly, the model generalizes well, considering the circumstances, on the Alphabet Inc. test dataset, which had only a 4.3% stock price increase in the interval Jan-Feb 2023.

6 Conclusion

The proposed system of models shows potential, however, it is hard to say at this point if it can outperform by a significant margin existing financial models. It outputs positive financial results, which may be proof of the benefit of using a terminal cost, however, its evaluation metrics are lacking by comparison. It needs to be trained for longer and with a broader training dataset.

6.1 Future Work

- Incorporating time-series forecasting. As Huang (2022) used a LSTM layer in the Financial DRQN algorithm, this might improve current results.
- Include more than one company in the data and have the model pick an action for each company.
- Explore the impact of different terminal costs C on the performance of the Safe - DRQN algorithm.
- Perform an extensive grid search and cross-validation for all hyperparameters.

References

[1] Lopez-Lira, Alejandro and Tang, Yuehua, *Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models* (April 6, 2023). Available at: `https://ssrn.com/abstract=4412788`

[2] Araci, Dogu Tan *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models* (June 25, 2019). Available at: `http://arxiv.org/abs/1908.10063`

[3] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, Gideon Mann *BloombergGPT: A Large Language Model for Finance* (May 9, 2023). Available at: `https://arxiv.org/abs/2303.17564`

[4] Huang, Chien Yi *Financial Trading as a Game: A Deep Reinforcement Learning Approach* (July 8, 2018). Available at: `https://arxiv.org/abs/1807.02787`

[5] Garrett Thomas, Yuping Luo, Tengyu Ma *Safe Reinforcement Learning by Imagining the Near Future* (February 15, 2022). Available at: `https://arxiv.org/abs/2202.07789`

[6] Google Brain team *TensorFlow* (2023). Available at: `https://www.tensorflow.org`

[7] Goncharov, Ivan *Financial Sentiment Analysis on Stock Market Headlines With FinBERT & Hugging Face* (December 11, 2022). Available at: `https://shorturl.at/ehyJY`

[8] Jansen, Stefan *Deep Reinforcement Learning: Building a Trading Agent* (September 7, 2017). Available at: `https://github.com/stefan-jansen/machine-learning-for-trading`