Deep Q Learning in Stock Trend Prediction

1st Ti-Wen Chen
Institute of International Business
National Taiwan University
Taipei, Taiwan
r10724050@ntu.edu.tw

***Abstract*—In this research, I present a Deep Q-Learning agent trained to analyze candlestick charts for the purpose of forecasting stock market trends. The results demonstrate that this agent is capable not only of accurately forecasting trends but also of consistently outperforming the Moving Average (MA) Breakout strategy, which is frequently employed in elementary algorithmic trading scenarios.**

***Index Terms*—Algorithmic Trading, Deep Q-Learning, Deep Learning**

I. Introduction

Extensive research has been conducted on stock market data with the objective of identifying patterns to forecast future stock trends. The evolution of machine learning and deep learning technologies in recent times has significantly enhanced the capabilities for such investigations. Shen et al. [1] employed Support Vector Machines (SVM) to predict the subsequent day's stock trend. Meanwhile, Rather [2] explored the use of Long Short-Term Memory (LSTM) networks for predicting stock prices and optimizing portfolios. Kelly et al. [3] utilized Convolutional Neural Networks (CNN) to analyze candlestick charts, classify cross-sectional stocks, and generate alpha that is not attributable to risk factors. Furthermore, the domain of reinforcement learning in finance has seen substantial growth. Jiang et al. [4] introduced a reinforcement learning-based Kelly strategy for optimal investment decisions in stock markets. Moser et al. [5] proposed a Double Deep Q-learning algorithm for trading on the S&P 500. Additionally, Liang et al. [6] integrated adversarial techniques with Deep Deterministic Policy Gradients (DDPG) and Proximal Policy Optimization (PPO) to develop a more robust stock portfolio.

However, most research on training deep reinforcement learning trading agents relies on historical time series data of opening, high, low, and closing prices. This approach does not fully leverage the advantage of deep learning in analyzing high-dimensional data. Moreover, dividing the opening, high, low, and closing prices into four separate data channels and using CNN for analysis is not intuitively ideal. Therefore, in this research, I directly use candlestick charts as the state that the agent will observe, and train the agent to execute different actions based on different states, which include buying, doing nothing, and selling.

The structure of the remainder of this paper is as follows. Section 2 commences with an introduction to the data utilized in this study, followed by a detailed discussion on the methodology employed for re-scaling the data, thereby rendering the candlestick charts comparable across different stocks. Section 3 is divided into three parts: firstly, it introduces the states, actions, and reward function encountered by the agent; secondly, it details the convolutional network architecture used for analyzing these states; and thirdly, it describes the Deep Q-Learning framework implemented. Section 4 presents the results obtained during the testing period. Finally, Section 5 provides the conclusion of this research.

II. Data

The stock price data for this study were sourced from yfinance. The designated period for training the model spanned from January 1, 2000, to December 31, 2016, while the testing period extended from January 1, 2017, to December 25, 2023.

Owing to computational constraints, this study limited its focus to ten tickers from the S&P 500 for training the agent. The selected tickers include AAPL, INTC, JPM, AEP, NKE, NVDA, MSFT, T, SBUX, and GS. Although only ten tickers were utilized for training, the features used were derived from candlestick charts formed by open, high, low, close prices, and trading volume. In addition to re-scaling the price data as per the method outlined by Kelly et al. [3], the volume data was also standardized. Consequently, from a technical morphology perspective, it is believed that these candlestick chart features are applicable to any stock. Fig. 1 illustrates an example of a state encountered by the agent. Nevertheless, should more computational resources and time be available, a larger selection of tickers could be included in the training process.

Fig. 1. Re-scaled training state.

III. Methodology

In this section, I will introduce the development of the stock trend prediction agent. The first subsection will focus on the actions and reward function, with special emphasis on the reward function. The design of the reward function is crucial as it aims to ensure that the agent avoids making trades when the predicted future 20-day price fluctuation is less than the market costs, such as transaction fees and taxes. Following this, the architecture of the CNN will be detailed. Lastly, the section will concentrate on introducing the Deep Q-Learning framework.

A. States, Actions and Reward Function

As previously mentioned, the state faced by the agent is represented by candlestick charts derived from the past 40 days of re-scaled open, high, low, close, and volume data, as illustrated in Figure 1. The agent can take one of three different actions based on the observed states: bullish, flat, and bearish. In the implementation, these actions are correspondingly assigned values of 1, 0, and -1, representing the predicted direction. The 'flat' action indicates that if the agent predicts a price change over the next 20 days that is less than the market friction (i.e., transaction fees and taxes), it should refrain from any buying or selling activities.

Each action is associated with a specific reward function. The reward functions for 'buy' and 'sell' actions are straightforward to define, being the 20-day return multiplied by 100 and the predicted direction. For example, if the 20-day return is -0.06 and the agent's prediction is bearish, the reward would be 100 × -0.06 × -1 = 6. Conversely, if the prediction is bullish, the reward would be 100 × -0.06 × 1 = -6. However, for a 'flat' prediction, the reward system is different. If the actual return over the next 20 days is greater than the friction, the agent is penalized with a reward of -5 for an incorrect prediction. Conversely, if the return is less than or equal to the friction, the agent is encouraged with a reward of 5. The various scenarios and their corresponding reward functions are summarized below.

$$r = \begin{cases} return \times 100\ (\%) & \text{if} \quad correct, \\ -return \times 100\ (\%) & \text{if} \quad wrong, \\ 5 & \text{if} \quad flat,\ abs(return) < fee, \\ -5 & \text{if} \quad flat,\ abs(return) > fee. \end{cases}$$

B. CNN Architecture

In the majority of existing literature that employs deep reinforcement learning for portfolio optimization, features are often treated as channels within a 3D matrix. The dimension of the matrix is structured as features × time series data × stocks. CNN are then employed to analyze this data. However, I argue that this method of interpreting 3D panel data as image data is not intuitive and does not effectively align with the fundamental concept of employing filters in CNN for image analysis. Furthermore, Kelly et al. [3] have noted that utilizing graphical data, as opposed to time series data, enables CNN to discern non-linear spatial associations among various price curves Therefore, this research adopts candlestick charts as the training data, enabling the CNN to more effectively leverage its analytical strengths.

However, in contrast to the approach of Kelly et al. [3], who converted candlestick charts into black and white, this paper retains the color representation indicative of daily price movements. Specifically, a day where the closing price is higher than the opening price is represented in red, equal prices in gray, and a closing price lower than the opening price in green. I believe that this approach allows the agent to better discern the past trend of a given stock. Figure 2 illustrates the complete architecture of the CNN used in this paper.

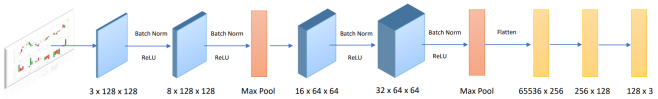

Fig. 2. CNN Architecture.

C. Deep Q-Learning

Deep Q-Learning was originally developed by Mnih et al. [7] in 2013 for use in agents playing Atari games. Given the notable performance of this agent in applications involving discrete actions, this study has elected to employ this architecture. The complete flowchart of the architecture is depicted in Figure 3. [1]

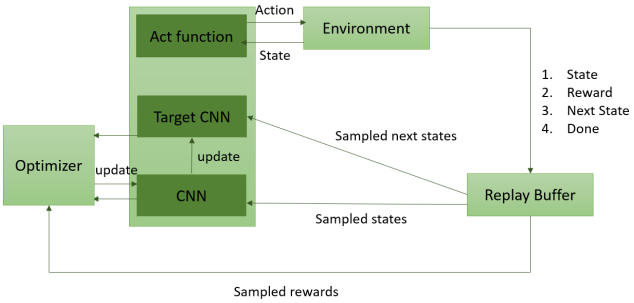

Fig. 3. Deep Q-Learning Architecture.

The loss function utilized for updating the Convolutional Neural Network (CNN) is defined as follows:

$$Loss = \frac{1}{n}\Sigma(Q_{current} - Q_{expected})^2 \tag{1}$$

where the $Q_{current}$ refer to the output Q values of the CNN upon receiving the current states. On the other hand, the output of the target CNN upon receiving the subsequent states, termed as Q_{next}. These next Q values are then transformed

[1] In this agent, the batch size sampled from the replay buffer is set to 32. The Adam optimizer is employed, with a learning rate of 10^{-4}.

to expected Q values according to the method outlined in Eq. (2).

$$Q_{expected} = r + \gamma \times Q_{next} \times 1_{dones}. \quad (2)$$

r represents the reward function as previously defined, the discount factor, $\gamma = e^{-r_f}$, where r_f denotes the market risk free rate.

IV. Empirical Results

A. Reward-Step Function

Most literature focusing on the training of reinforcement agents for financial trading omits the inclusion of a reward-step chart. This omission raises concerns regarding the stability of these agents. Figure 4, however, demonstrates that the reward in this study tends to stabilize over time and maintains positive.

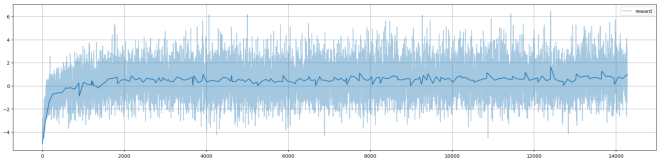

Fig. 4. Reward-Step Function.

B. Agent Performance in Testing data

In their study, Kelly et al. [3] attempted to use training data from a specific time period to predict the trend of stocks in a future period of the same duration. However, they also noted the potential of using shorter-term training data to predict longer-term stock movements, which could address the issue of limited financial data availability. Building upon this idea, in the testing period of this study, the agent was tasked with predicting trends in the Taiwanese stock market. The following are the tickers of ten companies used for this experiment: 2330.TW, 2881.TW, 2454.TW, 2455.TW, 3406.TW, 2412.TW, 2327.TW, 3034.TW, 2884.TW, 2886.TW.

Panel A of Fig. 5 displays the cumulative returns generated by the agent, accompanied with those from the buy-and-hold strategy and the MA breakout strategy, Panel B illustrates the decisions made by the agent at each respective time point.

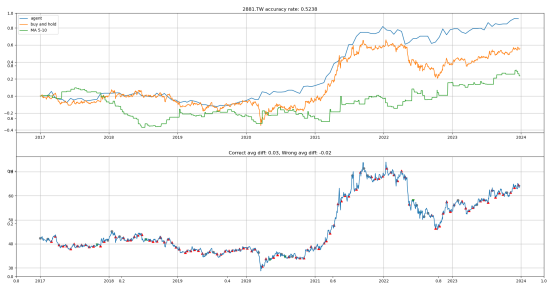

Fig. 5. cumulative Returns of 2881.TW.

Additionally, as of December 25, 2023, the mean cumulative return achieved by the agent across these 10 companies stands at 147.6%, compared to 103.8% for MA breakout strategy.

V. Conclusion

In their study, Kelly et al. [3] applied a supervised learning method, extracting factors from the candlestick charts of cross-sectional stocks to build a long-short portfolio, thereby achieving significant alpha. However, the approach of this research diverges from the aforementioned methodology. In this study, a deep Q-learning agent was trained to autonomously identify patterns within candlestick charts. Due to computational constraints, the training was restricted to data from 10 companies. The efficient market hypothesis suggests that generating alpha from past data is challenging, particularly when making predictions for individual stocks, as opposed to utilizing a long-short strategy. Despite these challenges, the results of this research demonstrate that the cumulative returns of the trading strategy formulated by this agent not only surpass those of a conventional buy-and-hold approach but also significantly outperform the MA breakout strategy, which is frequently employed in introductory algorithmic trading.

Regarding future research directions, several promising avenues exist. First, a detailed analysis could be conducted to elucidate the mechanisms by which the agent captures market trends. This would involve dissecting the agent's decision-making process to understand the underlying factors influencing its predictions. Secondly, there is potential for optimizing pair trading strategies. This could be achieved by employing a deep Q-learning agent to examine the relationships between the candlestick charts of stock pairs that exhibit cointegration. Such an approach may offer insights into more nuanced aspects of market dynamics and could enhance the effectiveness of algorithmic trading strategies.

References

[1] S. Shen, H. Jiang and T. Zhang. 2012. Stock market forecasting using machine learning algorithms. Department of Electrical Engineering, Stanford University, Stanford, CA, 1-5.

[2] A. M, Rather. 2021. LSTM-Based deep learning model for stock prediction and predictive optimization model. EURO Journal on Decision Processes, 9, 100001.

[3] J. Jiang, B. Kelly and D. Xiu. 2023. (Re-) Imag (in) ing price trends. The Journal of Finance, 78(6), 3193-3249.

[4] R. Jiang., D. Saunders and C. Weng. 2022. The reinforcement learning Kelly strategy. Quantitative Finance, 22(8), 1445-1464.

[5] F. Zejnullahu, M. Moser and J. Osterrieder. 2022. Applications of reinforcement learning in finance–trading with a double deep q-network. arXiv. arXiv preprint arXiv:2206.14267.

[6] Y. Y. Chen, C. T. Chen, C. Y. Sang, Y. C. Yang and S. H. Huang. 2021. Adversarial attacks against reinforcement learning-based portfolio management strategy. IEEE Access, 9, 50667-50685.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.