Comparison of Deep Q-Learning Network and Double Deep Q-Learning Network for Trading Strategy

Yang Chen

Sc21y2c@leeds.ac.uk

School of Computing University of Leeds Leeds, LS2 9JT, United Kingdom

Abstract—This study introduces a unique program designed to provide a specialized trading environment for the purpose of training agents. Two principal agents, namely Deep Q-Leaning Network (DQN) and Double DQN (DDQN), were taught and afterwards evaluated, with their performance measured using profit metrics, providing insights into their effectiveness in formulating stock policies. The comparative examination of these agents, based on their respective training periods, consistently demonstrated the higher performance of DQN in comparison to DDQN. As for the progress into the future, there are several potential optimizations that might be considered. These optimizations encompass fine-tuning the elements of neural networks, extending the duration of training sessions, and integrating emerging brain designs and training paradigms. The suggested application demonstrates versatility and adaptability by offering opportunities to incorporate a wide range of financial market data sets. This feature holds the potential to provide a more comprehensive training environment and im-prove the trading techniques employed by the agents.

Keywords: Deep Q-Learning Network, Double Deep Q-Learning Network, Trading Policy

1 Introduction

The field of stock trading has seen substantial changes in modern financial markets, leading to a multifaceted and dynamic environment. Investors and traders continually endeavour to gain advanced trading strategies in order to maintain a competitive edge within this turbulent industry.

Machine learning and statistical models, such as Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA), are frequently utilised in conventional stock trading techniques. These models are designed to predict future stock prices based on historical data. However, as the complexity and volatility of the financial market grow, the efficacy of these models becomes a topic of debate.

Consider, for example, the ARIMA algorithm, which is a widely used technique for forecasting time series data. Although it exhibits competence in collecting linear correlations within stable datasets, its effectiveness is impeded when the market exhibits non-linear behaviours or through sudden oscillations. The susceptibility of ARIMA models to unanticipated market events, such as geopolitical incidents or important economic statements, may arise due to their dependence on data stationarity [1]. The analytical capacities of a system are inherently weakened when the future diverges dramatically from established past trends.

ICEMBDA 2023, October 27-29, Tianjin, People's Republic of China

DOI 10.4108/eai.27-10-2023.2341940

On the other hand, LSTM, which falls under the category of recurrent neural networks, has been specifically developed to identify and capture prolonged dependencies within datasets. Although this characteristic renders it a potent instrument for numerous predictive endeavours, it is not devoid of its own set of difficulties. Firstly, it is worth noting that LSTM models may encounter challenges when attempting to comprehend data that has a well-defined sequential structure [1]. Moreover, the architectural design of the system does not facilitate parallel operations, hence posing a potential bottleneck in real-time trading situations. Additionally, the problem of gradient vanishing might negatively impact the predicted accuracy of the model, particularly when the length of the sequences exceeds a specific threshold.

Furthermore, it is important to acknowledge that the efficacy of both models relies on the calibre of the data used for their training. In a dynamic market context, the efficacy of utilising historical data as a dependable predictor for forecasting future market trends may be constrained. Numerous exogenous factors, including as policy changes, shifts in investor sentiment, and global events, can introduce variables that these models are unable to adequately account for. Although classic models like as ARIMA and LSTM provide vital insights into market behaviours by analysing historical data, it is important to acknowledge that these models are not without limitations. It is advisable for investors and financial experts to exercise caution when making forecasts, as they should employ a combination of quantitative models and qualitative insights in order to effectively traverse the dynamic realm of stock trading.

Although classic forecasting methods like ARIMA and LSTM have provided valuable insights in the past, the advancements in artificial intelligence have introduced novel techniques that have the potential to be more successful for analysing the stock market. An exemplary illustration of this transformative change led by artificial intelligence is the integration of deep reinforcement learning techniques, specifically DQN and DDQN. The DQN algorithm leverages deep learning techniques to optimize the decision-making process through the estimation of Q-values. On the other hand, the DDQN improves upon the DQN by utilizing two separate networks to estimate Q-values, hence reducing the potential for overestimation. Nevertheless, what are the reasons for employing Q-Networks in the context of trading? The complications as-sociated with anticipating stock prices or trends are emphasized in the research paper entitled [2]. The stock market poses significant hurdles to conventional prediction models due to its complex and multifaceted nature, influenced by a multitude of factors. This research presents a Deep Q-Learning agent that demonstrates proficiency in trend-following trading, relying exclusively on observable market data to make trading decisions. The findings demonstrate that the suggested methodology yields greater profitability compared to standard prediction-based tactics. Previous article explores the complexities of Q-Learning and its extension in deep learning [3]. This study provides clarification on the primary objective of Q-Learning, which is to derive the best control strategy, denoted as π^*, with the goal of maximizing the expected cumulative reward.

In light of the provided scenario, a crucial inquiry emerges: which of the two approaches, namely DQN and DDQN, demonstrates superior efficacy in generating more profits within the volatile stock market? The hypothesis put out in this study suggests that the utilization of DQN and DDQN algorithms can result in the development of an enhanced stock trading strategy. Among the several options considered, it is believed that DDQN exhibits a stronger potential for generating bigger profits owing to its more sophisticated methodology.

2 Methodology

2.1 Problem Setupt

Let $\boldsymbol{M}$ denote the stock market, specifically referring to its historical data. Let $\boldsymbol{A_1}$ and $\boldsymbol{A_2}$ denote the Deep Q-Network (DQN) and Double Deep Q-Network (DDQN) reinforcement learning approaches, respectively.

The objective function may be defined as:

$$\max_{A \in \{A_1, A_2\}} \textbf{Profit}\,(\boldsymbol{A}, \boldsymbol{M}) \tag{1}$$

The function **Profit(A, M)** represents the financial gain obtained through the utilization of reinforcement learning approach $\boldsymbol{A}$ in the context of the stock market $\boldsymbol{M}$.

The primary aim of an objective function is to maximize a given quantity. This is denoted by the symbol "max" in mathematical notation. This implies that we are seeking the maximum value of the given function. The function to be optimised is denoted as **Profit(A, M)**. The aforementioned metric denotes the financial gain produced by a reinforcement learning methodology. When applied to the stock market, the variable M.

The hypothesis can be empirically examined through the process of comparing:

$$\textbf{Profit}(\mathbf{A_1}, \mathbf{M}) \text{ and } \textbf{Profit}(\mathbf{A_2}, \mathbf{M}) \tag{2}$$

If:

$$\textbf{Profit}(\mathbf{A_1}, \mathbf{M}) > \textbf{Profit}(\mathbf{A_2}, \mathbf{M}) \tag{3}$$

The evidence supports the premise that the Double Deep Q-Network (DDQN), represented by $\boldsymbol{A_2}$, exhibits a greater capacity for generating larger profits.

2.2 DQN

The integration of deep learning with Q-learning in the form of Deep Q-learning has significantly revolutionized the field of reinforcement learning. The system demonstrates a high level of proficiency in addressing the difficulties presented by sensory inputs with many dimensions [4]. Deep Q-learning utilizes a technique known as "experience replay" as a fundamental component. This mechanism involves the storage and random selection of previous experiences by the agent in order to update its knowledge, hence facilitating a consistent learning process. This approach not only enhances the efficiency of learning but also mitigates the uncertainty associated with updates. The significance of off-policy learning in effectively handling parameter changes during sampling was underscored [5]. However, like other approaches, this method is not devoid of its own set of obstacles. The combination of deep neural networks and Q-Learning has been seen to occasionally result in learning instability. In order to tackle this issue, scholars propose the utilization of two Deep Q-Networks, wherein one network serves the purpose of making predictions, while the second network functions as a reference or target [3].

Now, let us proceed to explore the DQN in further detail. This approach combines Q-learning, a reinforcement learning algorithm that does not require a model, with deep learning approaches

that are well-suited for managing large state spaces. In the conventional approach of Q-learning, a tabular representation is employed to store and update Q-values associated with each feasible action within a specific state. However, this approach becomes unfeasible when dealing with extensive state spaces. The DQN is a reinforcement learning algorithm that combines deep neural networks with the Q-learning algorithm. The utilization of a profound neural network is employed for the purpose of forecasting the Q-value function, a theoretical construct that has been investigated [6]. The learning process of this network involves the minimization of the discrepancy between the anticipated and intended Q-values. The current condition is inputted into the system, which then undergoes processing to generate Q-values corresponding to potential actions. In order to promote consistent learning, DQN employs a mechanism called experience replay, wherein past experiences are stored and subsequently sampled in a random manner during the training process [7]. In addition, the DQN employs a target network, which is a secondary network that updates at a slower rate compared to the primary network. This target network is utilized to compute target Q-values by applying the Bellman equation [8]. The integration of a comprehensive methodology, along with the exceptional capabilities of deep learning in managing extensive state spaces, distinguishes DQN as a prominent technique in the field of reinforcement learning.

2.3 DDQN

DDQN algorithm is an improved version of the DQN algorithm. By combining Q-learning with deep neural networks, the DDQN algorithm has demonstrated exceptional performance in a range of reinforcement learning tasks. One significant drawback of the DQN algorithm is its inclination to overestimate action values as a result of employing the maximum operator in the Q-value update process [9]. The DDQN algorithm resolves this issue by separating the process of selecting an action from evaluating its value [10]. In contrast to the utilization of the maximum predicted Q-value of the subsequent state in the DQN algorithm, the DDQN approach involves the utilization of the present Q-network for action selection and the target Q-network for evaluation purposes. This alteration mitigates the tendency to overestimate, hence resulting in more consistent and precise estimations of Q-values. Empirical research has shown evidence that DDQN consistently outperforms DQN in terms of faster convergence and higher quality policy outcomes [11]. Similar to its precursor, DDQN employs experience replay as a mechanism to preserve and selectively sample previous experiences. This approach effectively disrupts the correlation between consecutive samples, hence enhancing the stability of the training process. In addition, the utilization of a target network that is updated less frequently serves to maintain stability in the updates of Q-values, reducing the likelihood of divergence [9].

2.4 Evaluation Indicator

This study utilises a methodical methodology to assess the efficacy of two well-known deep reinforcement learning algorithms: Deep Q-Network (DQN) and Double Deep Q-Network (DDQN). The training protocol is organised into successive epochs of 500, reaching a final total of 2,000 epochs. In particular, subsequent to every 500, 1,000, 1,500, and 2,000 epochs, both algorithms undergo a testing phase. The evaluation criteria employed during these test phases mostly consist of indicators such as loss, reward, and profit. This study aims to offer a detailed analysis of the effectiveness and convergence patterns of the DQN and DDQN algorithms over

a prolonged training period. This will be achieved by systematically comparing their performance at specific training intervals.

3 Result

In the conducted research, a simulation platform running on a Mac equipped with an M2 Pro chip was employed. The experimentation and instruction were carried out utilising the Spyder integrated development environment in conjunction with the PyTorch framework. During the course of our research, it was seen that PyTorch's compatibility with Apple's chip series was limited. As a result, the computational tasks were predominantly performed on the central processing unit (CPU) of the M2 Pro chip.

3.1 Data Set

The dataset supplied is a collection of data points organized in chronological order, specifically focusing on the values of stocks. The dataset was acquired via Kaggle and covers a duration of nearly 13 years, commencing on February 25, 2005, and concluding on November 10, 2017. The dataset provided is extensive in nature, encompassing a range of daily factors like the opening price, highest price, lowest price, closing price, and trade volume. During the observed period, it was noticed that the stock's opening price exhibited fluctuations within the range of 2.7595 to 37.85. Similarly, the daily trading volume demonstrated a range spanning from 2,691 to 6,370,578. The mean opening value was calculated to be 12.5343, accompanied with a mean trading volume of 204,258. One notable characteristic of the dataset is the presence of the "open interest" variable, which is noteworthy due to its exclusive inclusion of zero values. This issue merits consideration as it could suggest the possibility of data exclusion in further analysis. The dataset's extensive content renders it a highly suitable asset for the development and evaluation of trading strategies, particularly those that employ computer approaches to uncover patterns and gain valuable insights.

The framework proposed in this study establishes a trading environment based on a Deep Q-Network (DQN) approach. Within this environment, an agent engages with historical stock price data through interactions. The state of the environment encompasses both the present position value and the historical price variations. The agent has the ability to choose from three possible actions: hold, purchase, or sell. Rewards are granted in accordance with the trading decisions made by the agent, while penalties are imposed for acts that are deemed suboptimal. The agent utilises an ε-greedy approach, wherein experiences are saved for replay and periodic training is conducted. The primary goal is to maximise the agent's cumulative earnings over an extended period by engaging in a series of iterative exchanges and refining its trading strategy accordingly.

3.2 Result Display

In the trading simulation, an assessment was conducted to compare the performance of the DDQN and the standard DQN, as shown in Fig. 1. The results indicated that DDQN achieved a cumulative reward of 1,438,184, surpassing the performance of DQN, which obtained a cumulative reward of 1,166,264. It is worth mentioning that during epoch 443, the DDQN had an unusually high loss value of 368,520.55. In contrast, the peak loss observed for the DQN was far lower at 170.52, occurring during epoch 1473. Although DDQN has shown improved

profitability, its significant variations in losses indicate possible issues in learning or stability during some periods. In contrast, the DQN exhibited a more stable learning trajectory, but accompanied by a decrease in profitability. Moreover, the rewards and profits of the training and testing set are demonstrated in Fig 2, Fig 3, Fig 4, and Fig 5 respectively. The observations highlight the intricate equilibrium between stability and profitability in both algorithms, underscoring the necessity for more optimization in practical trading situations..

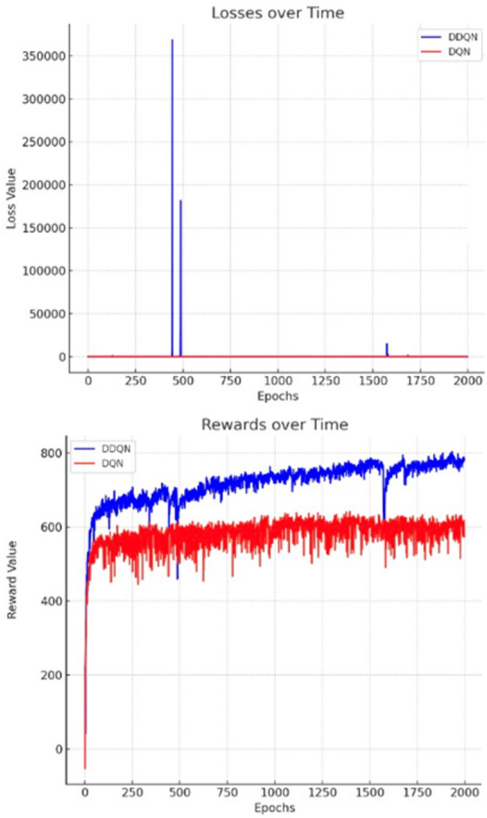

Figure 1. Comparison between two agents (Figure credit: Original).

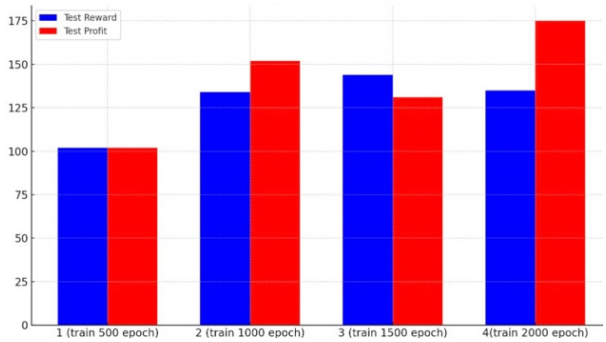

Figure 2. Test reward and profit for DQN (Figure credit: Original).

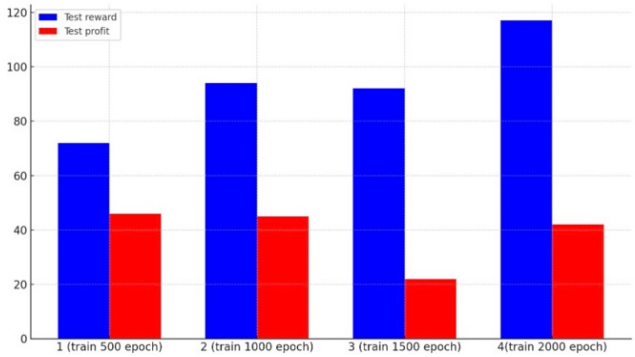

Figure 3. Test reward and profit for DDQN (Figure credit: Original).

Based on the test findings, it is apparent that the trained DQN exhibits superior performance compared to the DDQN in this particular activity, as evidenced by higher levels of reward and profit. While the DDQN exhibited a greater reward during the training phase, its overall profitability remains inferior to that of the DQN when assessed from an academic standpoint.

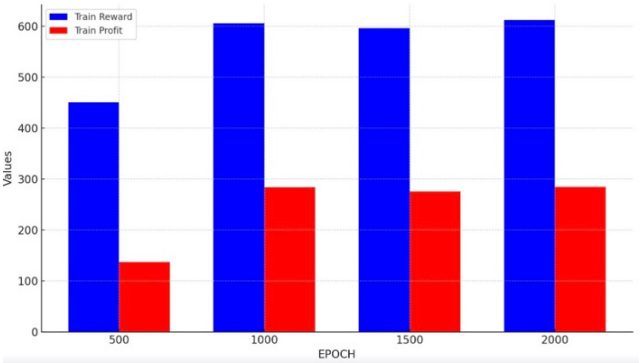

Figure 4. Train reward and profit for DQN (Figure credit: Original).

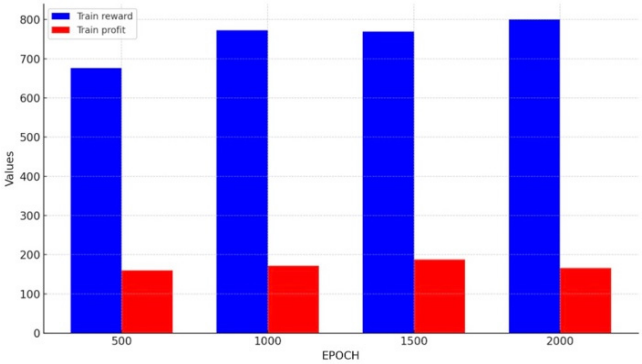

Figure 5. Train reward and profit for DDQN (Figure credit: Original).

4 Discussion

The study delved deeply into the application of the DQN and DDQN algorithms in stock trading strategies. From the results, while the DDQN showed potential during the training phase, the DQN outperformed in terms of overall profitability. This raises questions about the stability and consistency of DDQN in real-world trading scenarios. The significant loss observed for DDQN during a specific epoch suggests potential vulnerabilities in its learning mechanism. On the other hand, DQN, despite its simpler architecture, demonstrated a more stable learning trajectory. This juxtaposition under-scores the intricate balance between complexity and stability in trading algorithms.

Both DQN and DDQN heavily rely on parameters derived from historical data for training. The efficacy of these parameters, especially in a dynamic market, is crucial for the success of the models. For instance, the "experience replay" mechanism in DQN, which stores and randomly selects past experiences, plays a pivotal role in ensuring consistent learning. Similarly, the dual-network structure in DDQN, designed to separate action selection from value evaluation, is pivotal in reducing overestimation biases. However, as observed, this added complexity might introduce other challenges. The study's results, especially the variations in profitability and learning stability, can be attributed to how these models utilize and optimize their parameters.

The world of artificial intelligence and machine learning is ever-evolving, promising newer algorithms and strategies that could further revolutionize stock trading. There's potential in exploring hybrid models that combine the strengths of multiple algorithms, such as integrating DQN or DDQN with other advanced machine learning approaches. Real-time data incorporation, like news events or social media sentiments, could further enhance the predictive power of these models. Additionally, as the study utilized a specific dataset from a defined timeframe, future research could benefit from diverse datasets spanning different market conditions and timeframes. This would provide a more comprehensive understanding of the models' adaptability and performance consistency.

5 Conclusion

In summary, the trading environment designed for agent training has been carefully developed within this program. Once an agent fulfills the predetermined training conditions, it is subjected to a comprehensive evaluation process. The evaluation of the DDQN and DQN agents centers on the profit measures, which serve as a measure of their effectiveness in formulating stock policies. Moreover, a comprehensive comparative study is undertaken, wherein agents are juxtaposed based on different lengths of training. The results obtained from the series of trials consistently demonstrate the higher performance of the DQN algorithm compared to the DDQN algorithm.

During the evaluation phase, the DQN agents exhibited an average profit increase of 4.87%, whereas the DDQN agents had an average profit decrease of -0.27%. During a period of 500 epochs, it was observed that the DQN agents exhibited superior performance compared to the DDQN agents in 5.14% of the epochs.

Furthermore, a thorough comparative analysis is conducted, in which agents are juxtaposed according to varying durations of training. The findings derived from the consecutive epochs consistently indicate that the DQN algorithm outperforms the DDQN method in terms of reward. Specifically, the DQN agent achieved an average reward of around 128.75, whereas the DDQN agent earned an average reward of 93.75. In addition, it is observed that the average loss of the Deep Q-Network (DQN) is 105.55, but the Double Deep Q-Network (DDQN) exhibits an average loss of 369.91.

Looking towards the future, there exists a multitude of prospective improvements that are calling for attention. The optimization of the training regimen's efficacy could be enhanced by implementing subtle modifications to the variables within the neural network. Exploration of prolonging the duration of the training represents an additional viable approach. With the constant evolution of technology and data analysis approaches, it is expected that emerging neural network topologies and training paradigms have the potential to enhance the performance of the trading agents to a greater extent. Possible avenues of exploration could include the implementation of more complex network structures, the incorporation of innovative optimization methods, or the fusion with other advanced machine learning approaches. Furthermore, the program's adaptability can be expanded by integrating it with additional data sets that have not been utilized in the current version. An in-depth analysis of various financial markets, the examination of numerous trading methods, and the evaluation of complex market factors such as news events or macroeconomic data may offer a more comprehensive environment for training and testing agents.

References

[1] H. Ji, "Stock price prediction based on SVM, LSTM, ARIMA," BCP Business & Management, vol. 35, pp. 267-272, 2022.

[2] A. A. Grover and R. S. Gabriel, "Analysis of Algorithmic Trading with Q-Learning in the Forex Market," presented at the 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021.

[3] J. Chakole and M. Kurhekar, "Trend following deep Q-Learning strategy for stock trading," Expert Systems, vol. 37, no. 4, 2019.

[4] K. Kim, "Multi-Agent Deep Q Network to Enhance the Reinforcement Learning for Delayed Reward System," Applied Sciences, vol. 12, no. 7, 2022.

[5] V. Mnih et al., "Playing Atari with Deep Reinforcement Learning," p. arXiv:1312.5602, 2013.

[6] M. Furukawa and H. Matsutani, "Accelerating Distributed Deep Reinforcement Learning by In-Network Experience Sampling," presented at the 2022 30th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), 2022.

[7] B. Zigon and F. Song, "Accelerating Experience Replay for Deep Q-Networks with Reduced Target Computation," presented at the Computer Science and Machine Learning Trends 2023, 2023.

[8] J. Yu, J. Chen, Y. Chen, Z. Zhou, and J. Duan, "Double Broad Reinforcement Learning Based on Hindsight Experience Replay for Collision Avoidance of Unmanned Surface Vehicles," Journal of Marine Science and Engineering, vol. 10, no. 12, 2022.

[9] L. Ji, R. Zhao, Y. Dang, J. Liu, and H. Zhang, "Query Join Order Optimization Method Based on Dynamic Double Deep Q-Network," Electronics, vol. 12, no. 6, 2023.

[10] L.-W. Feng, S.-T. Liu, and H.-Z. Xu, "Multifunctional Radar Cognitive Jamming Decision Based on Dueling Double Deep Q-Network," IEEE Access, vol. 10, pp. 112150-112157, 2022.
[11] Z. Zhu, C. Hu, C. Zhu, Y. Zhu, and Y. Sheng, "An Improved Dueling Deep Double-Q Network Based on Prioritized Experience Replay for Path Planning of Unmanned Surface Vehicles," Journal of Marine Science and Engineering, vol. 9, no. 11, 2021.