Deep Reinforcement Learning on Stock Trading

Ke Chen
Katz School of Science and Health
Yeshiva University
New York, USA
kchen6@mail.yu.edu

***Abstract*—Deep Reinforcement Learning is a subfield of machine learning that combines reinforcement learning (RL) and deep learning. A tentative implementation of stock trading applying Deep Q Learning and other variants were practiced to show the ideas of DRL.**

***Index Terms*—Deep Reinforcement Learning, Deep Q Learning, Stock Trading**

I. Introduction

Deep Neural Network is the most used method for stock price prediction, leveraged on the ability of automatically finding the corresponding representation of the lower dimension by extracting the higher dimension input data. The prediction result is a powerful reference for trading stock manually. The Deep Neural Network methods include LSTM, CNN and many more which rely on the stock price history [2]. However, the regular DNN methods internally use the supervised learning logic for prediction. Stock trading is not only the prediction of price, but also could be seen as an associate pressing the sell or buy button based on his or her own decision and experience. As the current stock price changes with a variety of factors, such as news, accidents and other unpredictable influences, the profits made after trading seems more convincing and intelligent. The trading activity, which is the interaction of the person and current situation concept, matches the idea of Reinforcement Learning very well. Therefore, the stock trading could be applied with Reinforcement Learning. Studies show that applying deep reinforcement learning on stock trading can obtain higher profits compared with other trading strategies [3].

II. Reinforcement Learning ideas on stock trading strategies

A. Background

The basic idea of Reinforcement Learning is to create an agent to simulate human cognition and learning to take actions. The agent learns how to operate in the environment through an iterative feedback loop. The agent takes an action and gets a result, the state will change based on the rewards either increased or decreased. Then the agent will learn which action should be taken to receive the most reward. In stock trading scenario, the agent could decide to buy, sell, or stay on the current situation. After taking the action decided, observe how much profit it may get and learn which action should take in the future.

An important concept here is the Q value of a state-action pair, which means the expected reward if taking a certain action under a certain state. The Q value is higher, the more reward will get, the action under the current state is the right one. In this stock trading case, we would like to acquire the largest Q value action to receive the most profits. The method requires and updates the Q value of a state-action pair is called Q learning, which is based on the Bellman Equation.

As the DNN and Reinforcement Learn both has distinct advantage, they could be a complimentary of each other and combined as Deep Reinforcement Learning (DRL). The core idea is to use the neural network to predict the Q value rather than update the state-action Q value table according to the Bellman Equation. Deep Q Network (DQN), Double Deep Q Network (DDQN) and Dueling Deep Q Network (Dueling DQN) are examples of DRL.

B. Algorithm and Architecture

Take Deep Q Network as an example, the method architecture could be divided into 3 phases:

- 1. Firstly, the raw data will be processed before the DQN starts. the basic analysis and processing will be done in this phase. Then, the data go to the second phase, the core network phase.
- 2. For DQN, the purpose of the network part is to calculate the Q value of each action may take under the status. Therefore, the input of the network is the status, and the Q values with different actions, which is 3 (buy, sell, and stay) under the stock trading case, will be the output.
- 3. Finally, depending on the Q values, the agent could select the action getting the largest Q value.

As the second phase is the center of the architecture, it is necessary to clear the mind of the core network phase.

Recall the regular neural networks, the main task is to update the weights of each layer. The weights are updated based on the difference of prediction and target, considered as the loss. After that, apply backpropagation, gradient descent, and update the weights. In deep reinforcement learning, two neural networks are needed, the main Q network and a target network.

Getting back to the neural network in Q learning, the loss is the sum of squared differences of the Q-values and the targets. After N steps, the weights from the main network will be copied to the target network. Then the weights in the target network will be the temporary standard of main network to reach for weights updating.

Another concept involved in DQN is the memory, which is for experience replay. The experiences the agent learned will not be directly used for neural networks. They will be stored in a memory as the learning source. When the threshold reached, the experiences of the memory will be put in the neural network for training, as batch processing in regular NN.

In the stock trading scenario, the status is the stock price and/or the volume of stock you are possessing. After several hidden layers of computation, the final output are the buy-reward, sell-reward, and stay-reward Q values. The workflow is shown in the picture below.

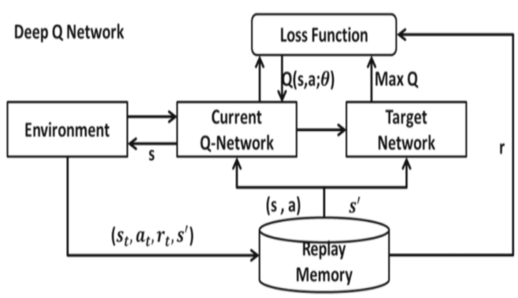

Fig. 1. Fig1.Deep Q Network Architecture [1]

III. DRL Implementation and Result

Historical daily prices and volumes of all US stocks dataset were used to reproduce the stock trading work based on three different DRL methods. In our implementation, we randomly select one stock for showing the workflow. Take DQL as example, the environment determined the trading actions and the corresponding reward and profit if taken. In the environment, the observation was set as 60 days by default. Secondly, the neural networks. The neural network could be implemented by Keras, Tensorflow and other libraries. The input size of the neural network is the days we are willing to observe, which is 60 days in this case. The network was arbitrarily created as 1 input layer + 2 hidden layers + 1 output layers. The trading model was trained based on the 4-layer neural networks.

According to the training and testing Time-market Q value pairs, the magenta points represent the time to take the ”buy” action, cyan points for ”sell” and gray points for ”hold” action. The plot shows the action selection is generally similar to human being’s common sense that ”high sell, low buy”. The decision seems locally effective; however, the future reward was considered in the algorithm. Therefore, it is more long-sighted than the pure stock price prediction.

Even the three methods had different performance on the action points plot, the loss and rewards plots show convergent

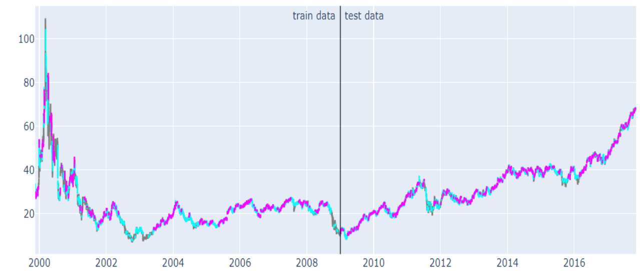

Fig. 2. Fig2. Time-market Q value pairs

trends. The loss gets lower, and the reward gets higher with the epoch of training increasing. The regular DQN had a better performance of convergence in our attempt.

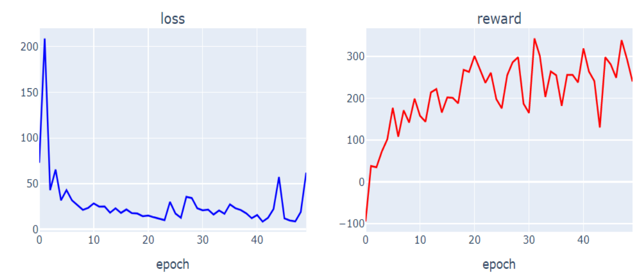

Fig. 3. Fig3. Loss and reward after each epoch for DQN

IV. Conclusion and Future Work

Unlike Q learning, Deep Q Network and other variant DQN methods update the Q table with neural networks rather than the Bellman equation, DQN utilize neural networks to approximate the Q value, which is more powerful than the pure Q learning. In the future, more variant DQN could be attempted, such as Noisy DQN and DQN with Prioritized Experience Replay. In this paper, we applied steady epsilon-greedy method. For better convergence performance, decayed-epsilon-greedy could be considered in the future on stock trading.

References

[1] Y. Li, P. Ni, and V. Chang, “Application of deep reinforcement learning in stock trading strategies and stock forecasting,” Computing, vol. 102, no. 6, pp. 1305–1322, 2019.

[2] L.-C. Cheng, Y.-H. Huang, M.-H. Hsieh, and M.-E. Wu, “A novel trading strategy framework based on Reinforcement Deep Learning for financial market predictions,” Mathematics, vol. 9, no. 23, p. 3094, 2021.

[3] Y. Li, P. Liu, and Z. Wang, “Stock trading strategies based on Deep Reinforcement Learning,” Scientific Programming, vol. 2022, pp. 1–15, 2022.

Attached code Link: https://github.com/chenkecoco1/Reinforcement-Learning/blob/main/Mini_Project.ipynb