Universe
Scientific Publishing

Research on Optimal Portfolio of DDPG Strategy Based on Deep Reinforcement Learning

Kai Jing

North China University of Water Resources and Hydropower, Zhengzhou 450000, China.

Abstract: With the continuous development of artificial intelligence and large numbers, people are no longer satisfied with the traditional investment portfolio method. This paper proposes a way of investment portfolio based on reinforcement learning, and researches stocks through deep deterministic strategy gradient algorithm. We set up stock virtual account assets, set rewards and punishments, and charge certain transaction fees, so that the agent can carry out autonomous transactions to achieve the effect of decision optimization.

Keywords: Deep Deterministic Strategy Gradient Algorithm; Agent; Decision Optimization

Introduction

Portfolio theory was first put forward by American economist Markowitz in 1952, and he conducted systematic, in-depth and fruitful research on it, and established appropriate portfolio strategies to effectively diversify risks[2]. In today's financial market, this method of risk diversification and return maximization has been widely used, no longer only for financial investment and risk management, but its application in the stock market is also controversial.

There is a lot of uncertainty in the financial markets, so it is difficult to build a portfolio model that is suitable for investors. The depth deterministic strategy gradient algorithm is a model-free strategy algorithm that learns continuous actions, which can make the robot better reach the optimal control equilibrium point [3]. It can not only be used to solve the continuous decision problem of asset weight allocation in portfolio, but also can be used to deal with the multi-dimensional state space problem that needs to be considered in the process of portfolio. Therefore, this paper uses DDPG algorithm for portfolio management and effectively studies the stock market.

1. Background introduction

Reinforcement learning was first proposed in 1954 and was originally used to control dynamic systems for control purposes[1]. With the rapid development of deep learning, Google deep Mind uses deep networks to approximate Q-value functions to solve spatial problems in continuous states, which is the origin of modern deep reinforcement learning, known as deep Q networks. The development track chart is as follows:

Fig 1

Although the DDPG algorithm was originally designed to handle some continuous action problems, it can also handle discrete action problems in some special cases, including portfolio management problems. We add experience playback to it, and the application in the portfolio allows the agent to store previous experiences and randomly extract small batches of data from the stored experiences for training, which makes the agent learn more efficiently.

2. Problem description

In the process of investing a portfolio, the first thing to do is to find a balance between maximum return and risk, so investors need to

change the weight of each asset class by judging the form of the current moment. Therefore, investors need to continuously research, learn and adapt to changes in the market, adopt scientific methods and tools, as well as continuous risk management strategies. In this paper, our investment object is agent. By making agent constantly learn to adapt to changes in the market, we can find a balance point between the maximization of returns and the maximization of risks.

2.1 Markovian decision process

Markov decision process is referred to as MDP, it contains five elements, S, A, P, R, and means:

S: A finite set of states $S_t = [s_1, s_2, \cdots, s_t]$

A: A limited set of actions $A_t = [a_1, a_2, \cdots, a_n]$

P: State transition probability matrix, $P_{ss'}^{a} = P[S_{t+1} = s' \mid S_t = s, A_t = a]$

R: Reward function, $R_s^a = E[R_{t+1} \mid S_t = s, A_t = a]$

γ: $\gamma \in [0,1]$ is a discount rate.

Stock data is a common type of financial time series data that we record as the closing price of the stock over the past N days, as $S_t = [x_{t-N}, x_{t-N+1}, \cdots, x_{t-1}, x_t]$,Here A may have three actions, which are buy, sell, hold, that is $A_t = [a_b, a_s, a_h]$. The agent needs to find an optimal strategy, which is represented by π. When each state outputs only one deterministic action, that is, the probability of this action occurring is 1, and the probability of other actions occurring is 0. When the probability of the action is random, he outputs the probability distribution of the action in each state and samples the action according to that distribution. We use $v^{\pi}(s)$the policy-based state value function in MDP.

In the strategy,the value of taking action a in state s is equal to the product of the immediate reward plus the state transition probability of the next state of all possibilities after decay and the corresponding value,which we write as:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) V^{\pi}(s')$$

In Bellman's equation, strategy π defines the probability of taking an action in a given state, and the value function $V^{\pi}(s)$ describes the expected cumulative return of following strategy π and starting from state s. We can get the equation as follows:

$$\begin{aligned} V^{\pi}(s) &= E_{\pi}[R_t + \gamma V^{\pi}(S_{t+1}) \mid S_t = s] \\ &= \sum_a \pi(a \mid s) \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma V^{\pi}(s')] \end{aligned}$$

$$\begin{aligned} Q^{\pi}(s,a) &= E_{\pi}[R_t + \gamma Q^{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] \\ &= R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) \sum_{a' \in A} \pi(a' \mid s') Q^{\pi}(s', a') \end{aligned}$$

Our goal is to find the optimal strategy π^* so that for all states s, $V^{\pi^*}(s) \geq V^{\pi}(s)$.

2.2 Depth deterministic strategy

In DDPG algorithm, actors output actions and critics evaluate actions[5]. Critics are used to estimate Q values, actor networks are used to determine the best action for a given state, while critics are used to guide actor updates, and actors are used to select actions. The Actor network inputs state S, where state S is a vector containing stock prices, portfolio weights, technical indicators, macroeconomic indicators, and so on, which we write:

$$S_t = [P_{t,1}, P_{t,2}, \cdots, P_{t,n}, \omega_{t,1}, \omega_{t,2}, \cdots, \omega_{t,n}, \cdots]$$

For the reward function, we can set it to the actual return of the portfolio, but considering the transaction cost and risk, we can define it as follows:

$$r_t = \omega_t^T \cdot r_{t+1} - \lambda \cdot C(\omega_t, \omega_{t+1}) - \alpha \cdot Risk(\omega_{t+1})$$

In the Critic network, the neural network is used to estimate Q value, which is marked as $Q(s_t, a_t \mid \theta^c)$. In the Actor network, given a state s, the Actor outputs a continuous action: $a_t = Actor(s_t \mid \theta^a)$. the update of the target Q value of the Critic network is as follows:

$$Q_{target}(s_t, a_t) = r_t + \gamma \cdot Q'(s_{t+1}, \text{Actor}'(s_{t+1} \mid \theta^{Actor'}) \mid \theta^{Ctric'})$$

Updating the Critic network using the mean square error is the loss function, as follows:

$$L = E[(Q_{target}(s_t, a_t) - Q(s_t, a_t \mid \theta^{Critic}))^2]$$

Update the Actor network through the Q value gradient of Critic, and guide how to update the Actor's strategy to maximize the expected cumulative return, the formula is as follows:

$$\nabla_{\theta^{Actor}} \mathrm{J} \approx E[\nabla_a Q(s, a \mid \theta^{Critic}) \times \nabla_{\theta Actor}(s \mid \theta^{Actor})]$$

We use DDPG algorithm for strategy exploration, evenly distribute the weights of 4 stocks, and adjust the shares in five steps to prevent the agent from frequently adjusting the stock weights. At the same time, we charge a certain fee, set at 0.1%, and replay the experience when the agent adjusts the stock weights, which helps to provide a more stable data flow. In the process of agent exploration, the ε-greedy strategy is added to make the algorithm select the algorithm strategy of the current optimal scheme at each decision point.

3. Experiment

3.1 Data and parameter configuration

We took stocks of Apple, Amazon, Google, and Microsoft, and downloaded closing price data from 2010 to 2022 from Yahoo Finance.

We selected two years of data from the graph to train the agent, and by training the data, the model was better able to handle these extreme cases. Training a model to deal with this uncertainty may make it more robust in the face of the unknown.

We intercept the closing price data from 2021 to 2022 and calculate the yield data of four stocks according to the yield formula.

As can be seen from the picture, the closing price data of the four stocks generally show a trend of rising first and then falling. We set up a reward and punishment mechanism in the trading environment to encourage agents to reweight stocks, while charging a 0.1% processing fee. When the learning rate is too high in the training process, the training speed will be accelerated and the model will converge more quickly, but the optimal solution may be skipped. If the learning rate is too small, the convergence speed will slow down, and the training times of the model will be too long, which is difficult to reach the optimal.

We introduce ϵ here to prevent the denominator from being 0 while recording the portfolio's volatility and total return.

3.3 Experimental result

3.3.1 Asset change

After training the code, we find the optimal model, that is, the model with the greatest returns, and plot the changes in the asset account.

The above figure shows the training results of the agent. We conducted the test by randomly selecting the returns of four stocks. This paper selected the data from December 7, 2018 to December 3, 2019 for testing, and the results are as follows:

	Value
Final portfolio value	126591.88141083531
Average daily return	0.0011106020606180497
Average maximum Sharpe ratio	0.05309490749322492

Table 1

After the closing of the transaction, the final asset value is 126591.88141083531 and the model yield is approximately 26.6%. Let's plot the change in the asset account.

From the asset account change chart, we can see that the effect of our portfolio with DDPG algorithm is quite good.

3.3.2 Stock weight change

As for the investment portfolio, our purpose is to diversify investment risks and increase returns. A good investment strategy should spread the proportion of assets to each stock, rather than putting all assets into one stock.

As can be seen from the above four pictures, the agent successfully dispersed assets into four stocks and dispersed risks.

3.3.3 Sharpe Ratio

We record the Sharpe ratio at each step in the model to observe the agent's strategy, as shown below:

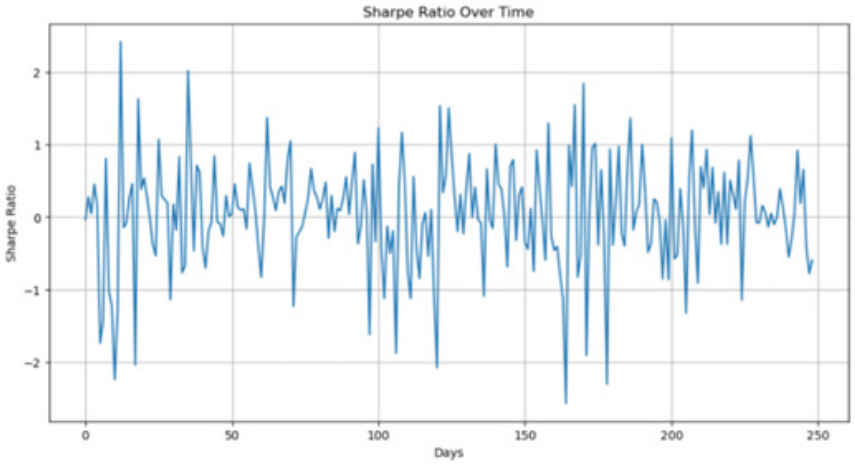

Fig 2

It can be seen from the above figure that when we use DDPG algorithm to carry out investment portfolio, the return of agent's strategy selection in most steps is higher than the risk, indicating that our research is feasible.

4. Conclusion

In this paper, we choose DDPG algorithm to study the portfolio problem in the financial field, and have achieved remarkable results. Some effective results have been obtained by adding cost to the transaction and adjusting the model parameters, but there are still many shortcomings to be improved. In the course of the experiment, we found that the training results of the agent were also very different with different input data. Therefore, we have chosen large oscillations in this paper to make the model better able to cope with these extreme cases.

Therefore, it is also necessary to continue to study the deep reinforcement learning algorithm to better solve various problems encountered in the real financial market.

References

[1] Shaokang D, Jiarui C, Yong L, et al. Reinforcement Learning from Algorithm Model to Industry Innovation Innovation: A Foundation Stone of Future Artificial Intelligence[J]. ZTE Communications,2019,17(03):31-41.

[2] Wang Kang, Bai Di. Research on Portfolio Management Based on Deep Reinforcement Learning [J]. Modern Computer,2021(01):3-11.

[3] Jiao Yuming. Stock portfolio Management and empirical research based on deep reinforcement learning [D]. Northwestern University, 2021.

[4] Chen Jia. Application of Deep reinforcement Learning in Stock Portfolio Management [D]. Huazhong University of Science and Technology, 2022.

[5] Xu Jie, ZHU Yukun, Xing Chunxiao. Research on Financial transaction Algorithm based on deep reinforcement Learning [J]. Computer Engineering and Applications, 2022, 58(07): 276-285.

[6] Li Ming. Research on Stock Market Prediction Based on Deep Learning [D]. Nanjing University of Posts and Telecommunications,2019.

[7] Liu G. Research and application of stock market prediction model and evaluation method based on deep learning [D]. Beijing University of Posts and Telecommunications, 2020.