



Research on Quantitative Trading Based on Deep Learning

Zhengyan Wang¹(✉) and Yisong Zhao²

¹ Paris-Saclay University, 92330 Sceaux, France
zhengyan.wang@universite-paris-saclay.fr

² Sorbonne University, 75005 Paris, France

Abstract. Traditional quantitative trading strategies are widely used in stocks, futures and other financial markets, but the manual extraction method of features makes it lack the ability to effectively adjust strategies dynamically, and deep reinforcement learning can effectively simulate complex market environments and solve dynamic quantitative trading problems. Based on the development status of the financial industry, this paper introduces the deep reinforcement learning algorithm into the field of stock trading to build an intelligent trading model. Its goal is to discover the laws of the market in the learning of massive data, so as to carry out effective transactions, effectively avoid market risks and improve investors' returns. On the basis of the traditional DQN algorithm, corresponding to the actual requirements, we propose RB_DRL deep reinforcement learning algorithm model to improve the network structure. The experimental analysis results show that the improved model also shows good results in multi-group comparative experiments.

Keywords: Deep Learning · Neural Networks · Quantitative Trading · Data Mining

1 Introduction

The dramatic changes in financial markets in recent decades have been largely reflected in the gradual transformation of a market structure dominated by manual trading to one dominated by automated computerised trading. Developments in order generation, transmission and execution technology have facilitated this rapid transformation, greatly increasing the speed, capacity and complexity of the trading functions available to market participants. Trading markets facilitate capital formation and allocation by setting the price of securities and allowing investors to enter and exit positions in securities whenever and wherever they wish [1]. The great changes in the financial market in recent decades are mainly reflected in the gradual transformation of the market structure dominated by manual transactions into a market structure dominated by automated computer transactions [2]. The development of technology to generate, transmit, and execute commands has driven this rapid shift, greatly increasing speed, capacity, and complexity of trading features available to market participants. Traditional strategies typically use manual

extraction of financial features to remove data noise and uncertainty [3]. However, traditional machine learning mainly takes prediction as the direction of supervised learning, and cannot achieve direct transaction decisions or direct transaction costs. Therefore, this paper combines the current research status to enter the field of financial quantitative trading from the perspective of deep reinforcement learning [4]. While providing a new model for the field of financial quantitative analysis, it also expands another application scenario of deep reinforcement learning in the financial field, which has important academic research and practical application value.

2 Deep Reinforcement Learning

Deep reinforcement learning (DRL) is a product of the convergence of deep learning as well as reinforcement learning. Deep learning takes raw data such as images and feeds them directly into a multi-layer deep neural network, which gradually extracts higher-level features through non-linear transform learning and has a strong perceptual capability [5]. The latter has a bias towards learning problem-solving decision making capabilities, where the intelligence and the environment are constantly interacting and trial and error, using reward and punishment mechanisms to uncover optimal control strategies. The two complement each other's strengths, providing a general new way of thinking about solving complex problems, as shown in Fig. 1. In essence, deep reinforcement learning still follows the algorithmic thinking of traditional reinforcement learning [6]. The value-based approach approximates the state action value function through a neural network such as a DNN, which provides guidance for the selection of actions by the intelligence. Representative algorithms for such deep reinforcement learning methods include deep Q-networks [7]. Representative algorithms for this deep reinforcement learning approach include the Action Critic algorithm [8].

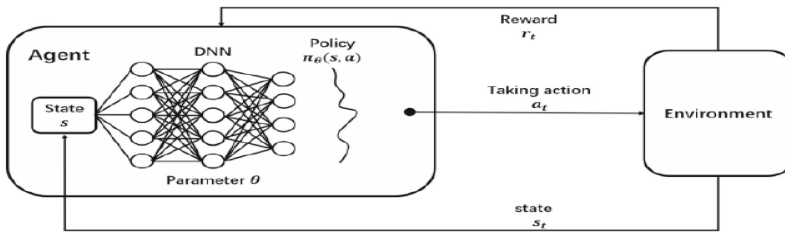


Fig. 1. Deep reinforcement learning process

Model-based deep reinforcement learning algorithms require that an intelligence be given a model of the environment or be asked to learn a model of the environment in order to perform a task in a particular environment. The most significant advantage of model-based deep reinforcement learning algorithms is that planning is considered in advance, where the intelligence tries out the available actions in advance of each step and then selects from the candidates, providing higher sampling efficiency compared to model-free deep reinforcement learning algorithms [9]. However, when the real environment

is more complex, there is a gap between the model explored by the intelligences and the actual valid model, which can lead to poor performance of the intelligences in the real environment [10]. In contrast, although the model-free deep reinforcement learning algorithm gives up the potential gain in sample efficiency, it is easier to achieve the learning of the intelligent body to the optimal policy/optimal value function and to adjust the learning process of the intelligent body after a lot of interaction between the intelligent body and the real environment, with progressively better performance in a trial-and-error manner. The DQN algorithm's structure is illustrated in Fig. 2.

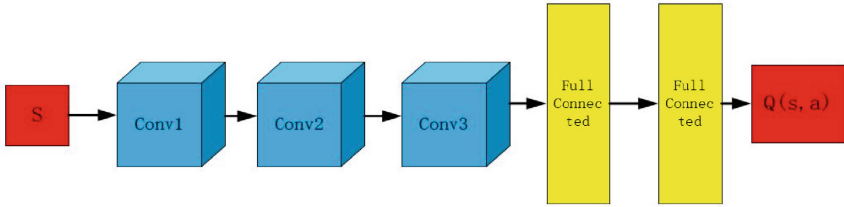


Fig. 2. Schematic diagram of the network structure of the DQN algorithm

Model-based DRL requires the construction of a model of the environment, relies on control theory and is often interpreted in terms of different disciplines. Model-free deep reinforcement learning learns policies based on information from direct interactions with an unknown environment, and so does not require knowledge of state transfer functions. Value function-based DRL algorithms learn an approximation of the optimal value function and indirectly learn deterministic policies [11]. The policy-based deep reinforcement learning algorithm learns an approximation of the optimal policy directly, outputting the probability of the next action and selecting the action based on the probability, but not the one with the highest probability, which is applicable to both continuous and discontinuous actions.

Model-based deep reinforcement learning algorithms require the construction of a model and the generation of a training set from the constructed model, so the complexity of the samples is not as high as model-free deep reinforcement learning algorithms, but the process of constructing the model is somewhat challenging. The classical algorithms for model-free deep reinforcement learning based on value functions are: the Deep Q-Network algorithm, and the Double Deep Q-Network algorithm. The Deep Q-Network algorithm is an offline algorithm based on value functions that combines the advantages of DNN and traditional Q-learning, differing from traditional Q-learning in that Deep Q-Networks use deep neural networks to fit Q-values and use an experience pool to store data obtained from interactions with the environment, whose empirical data can be reused.

3 Research on the Design of DRL Algorithm Based on Improvement

This study aims to improve the traditional deep reinforcement learning algorithm model and construct a new algorithm model. It learns the trading rules in the financial market through the model, so as to train an agent that can trade, and further build the corresponding quantitative trading system according to the trained algorithm model to carry out automatic quantitative trading.

3.1 Deep Reinforcement Learning Model Construction Based on Value Functions

In traditional reinforcement learning, such as the Q-learning algorithm, tables are used to record the results of each exploration. This method works better when both the state and the action space are low in dimension and the state space is not continuous. When states and actions become more dimensional, it is impossible to continue using tables, so the fitting problem of converting such tables into a function is raised. In deep neural networks, the state is used as the input to the network and the $Q(s, a)$ action value function is used as the output of the network. Based on this idea, the DQN model adopts a two-network structure, that is, one network is used to calculate the current value function, and the other network is used to generate the target value. And use the current value, and the target value to build a loss function to optimize the deep neural network.

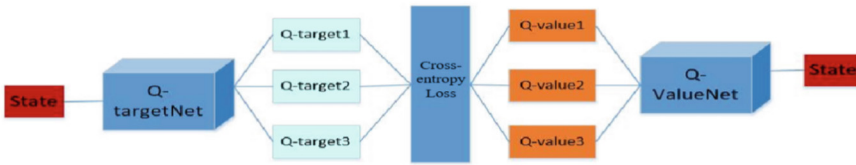


Fig. 3. RB-DRL model block diagram

Figure 3 is a structure diagram based on the improved DRL model, which consists of two parts in general. In the left half, a Q-targetNet is constructed to calculate the Q value of the target, and the input of the network uses the results of reinforcement learning exploration. The right half uses the same structure of Q-ValueNet to usher in the evaluation of the calculated action value function, and the last layer of the network uses the Softmax layer to input a 1×3 -dimensional action value function vector, and build a cross-entropy loss function for network updates.

Figure 4 shows the overall flow chart of the RB-DRL algorithm model. In the model, on the one hand, the Q-ValueNet, is used to interact with the environment, carry out the exploration process of reinforcement learning, and store the results of each experiment in the memory playback unit. The target value network, that is, Q-TargetNet, generates the target Q value, which is used as the target of learning to evaluate the current value function. After the introduction of the target value network, the model can ensure that the target Q value is constant for a period of time during training. On the other hand, due to the existence of the memory playback unit, the exploration results of the experiment are persistently stored, and the data in the unit can be taken out for further training and learning every once in a while, which is convenient for further updating the network.

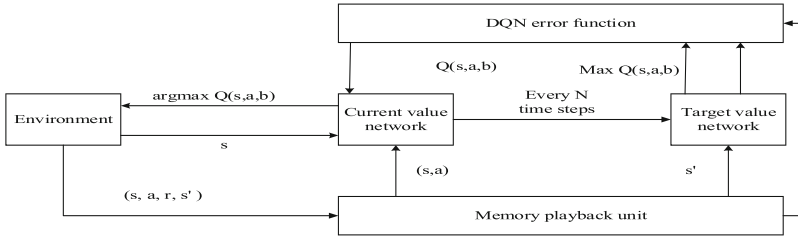


Fig. 4. RB-DRL algorithm flowchart

3.2 Design of Convolutional Neural Network with Residual Blocks

An improved convolutional neural network is introduced in the experimental study of this paper, and the structure diagram is shown in Fig. 5. The entire network is divided into 10 layers, including one input and output layer, three convolutional layers, two pooling layers, and three fully connected layers. The input layer inputs 20×20 -dimensional sample data, and after a series of convolution calculations, outputs 3-dimensional result data in Softmax, which can also be considered to divide the samples into 3 categories. In the convolutional layer, the first layer uses a $3 \times 3 \times 64$ convolution kernel for the input data for convolution, and the “SAME” method is used to fill the data edges during the convolution process to ensure that the overall dimension size remains unchanged during the convolution process. The sampling window of the pooling layer is 2×2 , which adopts the maximum pooling method and fills the data like the convolutional layer. The fully connected layer accepts the output of the final pooling layer and calculates, uses the dropout method to randomly discard some parameters on the calculation results, and introduces the idea of residual in the fully connected layer, and the results of the first layer of the fully connected layer are transmitted to the third layer in addition to the second layer, so as to amplify the signal and enhance the fitting effect of the network model.

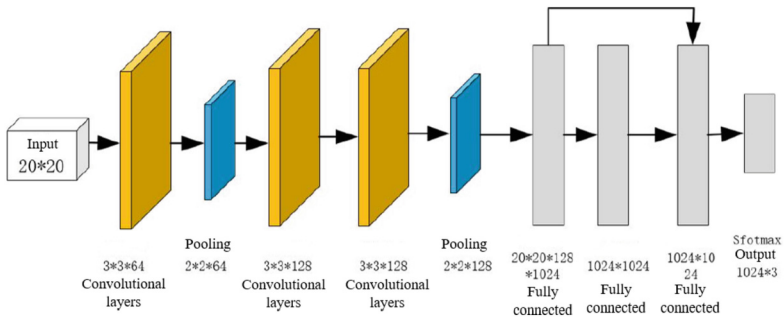


Fig. 5. RB-DRL network block diagram

4 Experimental Verification

The hardware environment of the experiment is Intel(R) Core(TM)i7-10210U, implemented in Python. Python 3.7 is the current more popular version of Python, compared to the previous Python 2 in all aspects of great improvement, Tensorflow is the next popular deep learning framework, providing a variety of neural network building interfaces.

The experiments in this paper were divided into two groups to verify the effectiveness of the improved model in this paper. Mainly select two stocks of Ping An and Shandong Gold for in-depth research and analysis. In terms of data, we select the data of the decade from 2010-01-01 to 2020-12-31, and divide it into training set and test set, and the training set selects the data of the eight years from 2010-01-01 to 2018-12-31, and the remaining two years are used as test set data to verify the learning results of the model.

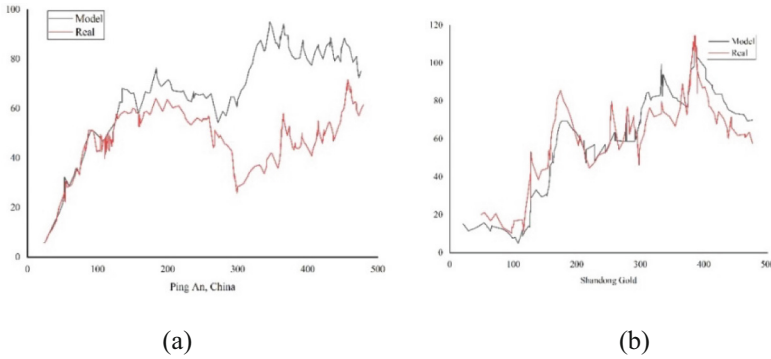


Fig. 6. Model yield curve based on fully connected neural network

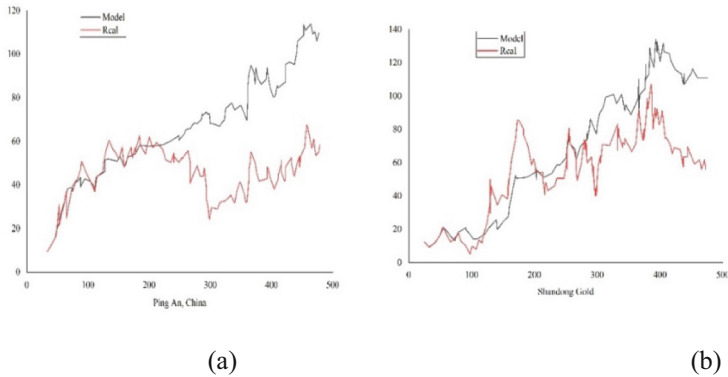


Fig. 7. RB_DRL Model yield curve

Figure 6 is the model yield curve based on the fully connected neural network, and Figure 7 is the experimental results of the improved RB_DRL deep reinforcement

learning algorithm. Looking at the overall yield curve, the improved model performed significantly on both stocks compared to the no improvement experiment. The final returns were able to reach the level of the leading benchmark, the drawdown of individual stocks was reduced, and the risk level of the strategy model was further controlled. Ping An of China earned 126.26%, an increase of 47.76%, and Shandong Gold earned 116.70%, an increase of 48.73%. From the yield curve, we can also see that the fitting effect of the improved model has been improved, which can further seize the market opportunities. Buy in time to get a higher yield. And can sell stop loss in time during the decline process to reduce the risk of the model.

Experimental results show that the return after using the deep reinforcement learning algorithm can exceed the return of the benchmark buy and hold. Therefore, the experimental results show that on the one hand, the introduction of deep reinforcement learning in the field of stock trading is effective and has high research value. On the other hand, in the use of the model, it is also necessary to improve the model according to the actual results. Experimental results show that the improved deep reinforcement learning algorithm model (RB_DRL) in this paper is effective.

5 Conclusion

With the development of various algorithm technologies and the advent of the Internet 2.0 era, computers are gradually integrating into traditional industries with their powerful functions. In the financial field, the related algorithm technology has shined and achieved unprecedented achievements. On the basis of the traditional dDRL algorithm, RB_DRL deep reinforcement learning algorithm model is proposed, and the effectiveness of the model is verified through a series of comparative experiments. This paper proposes RB_DRL based on DRL, which solves the problems of insufficient fitting degree of financial data by traditional DQN algorithms, as well as the correlation between features and the importance of factors to a certain extent. However, in the field of combining artificial intelligence and finance, the current research is still in its infancy, and the follow-up research in this area will continue.

References

1. Ahmadi, E., Jasemi, M., Monplaisir, L., Nabavi, M.A., Mahmoodi, A., Jam, P.A.: New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the support vector machine and heuristic algorithms of imperialist competition and genetic. *Expert Syst. Appl.* **94**, 21–31 (2018)
2. Cagliero, L., Garza, P., Attanasio, G., Baralis, E.: Training ensembles of faceted classification models for quantitative stock trading. *Computing* **102**, 1213–1225 (2020)
3. Huang, B., Huan, Y., Xu, L., Zou, Z.: Automated trading systems statistical and machine learning methods and hardware implementation: a survey. *Enterprise Information Systems* **13**, 132–144 (2019)
4. Abel, D., Dabney, W., Harutyunyan, A., Ho, M.K., Littman, M.L., et al.: On the expressivity of markov reward. *Adv. Neural. Inf. Process. Syst.* **34**, 7799–7812 (2021)
5. An, B., Sun, S., Wang, R.: Deep reinforcement learning for quantitative trading: challenges and opportunities. *IEEE Intell. Syst.* **37**(2), 23–26 (2022)

6. Houssein, E.H., Dirar, M., Hussain, K., Mohamed, W.M.: Assess deep learning models for Egyptian exchange prediction using nonlinear artificial neural networks. *Neural Comput.* **33**(11), 5965–5987 (2020)
7. A. Ntakaris, J. Kannianen, M. Gabbouj, A. Iosifidis, (2018) “Mid-price prediction based on machine learning methods with technical and quantitative indicators”, *Proc. SSRN*, pp. 1–40
8. Wang, Q., Xu, W., Huang, X., Yang, K.: Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning. *Neurocomputing* **347**, 46–58 (2019)
9. Kumar, G., Jain, S., Singh, U.P.: Stock market forecasting using computational intelligence: a survey. *Arch. Comput. Methods Eng.* **28**, 1069–1101 (2021)
10. Spooner, T., Savani, R.: Robust market making via adversarial reinforcement learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 4590–4596 (2021)
11. Saini, A., Sharma, A.: Predicting the unpredictable: an application of machine learning algorithms in Indian stock market. *Ann. Data Sci.* **9**, 791–799 (2022)