Reinforcement Learning Strategy for Quantitative Trading

Written by Xinyu Wang, Jingwei Zhou, Xiaoxu Bai

[1]xw3080@nyu.edu
jz6022@nyu.edu
xb2057@nyu.edu
https://github.com/VioletGo319/rl_trading/blob/main/rl-strategy.ipynb/

Abstract

This study employs a quarterly rolling window approach to evaluate and select from three reinforcement learning models—PPO, A2C, and DDPG—based on their Sharpe Ratios, with the intent to dynamically choose the model exhibiting the highest Sharpe Ratio at the end of each period for our trading strategy. This approach is substantiated through comprehensive data, as illustrated in Figure 1 and the associated table, which shows the fluctuations in Sharpe Ratios over several quarters from March 2016 to June 2020. For instance, the A2C model displayed significant peaks such as a Sharpe Ratio of 0.513 on 2017-03-31 and reached a high of 0.561 on 2018-03-31, indicating robust performance during these periods. Conversely, there were notable dips, such as -0.377 on 2020-06-30, reflecting its volatility.

The quantitative analysis reveals that our adaptive strategy, which prioritizes models based on the highest quarterly Sharpe Ratio, successfully leverages the periodic strengths of each model. This strategic approach has proven effective in optimizing returns by managing risks associated with market volatility. Specific examples from the data include periods like 2017-12-31, where the DDPG model led with a Sharpe Ratio of 0.389, and 2019-09-30, where the A2C model outperformed others with a ratio of 0.450. This dynamic model selection not only enhances the overall performance but also underscores the necessity of continuous model evaluation and risk management to adapt to rapidly changing market conditions. The success of this methodology in this volatile financial trading environment highlights the potential of reinforcement learning in developing sophisticated, adaptive trading algorithms.

Introduction

Automated stock trading is gaining popularity in finance for its ability to make smarter investment choices and increase profits. Recent advances in reinforcement learning (RL) have made trading strategies more advanced and effective. This project uses RL methods to create an automated stock trading system. We're using a big dataset with stock info from 30 American companies in different industries. Our goal is to use RL algorithms to analyze this data and come up with smart trading strategies. This project is important because it could change how people trade stocks by using machine learning to adapt to market changes and find patterns in the data for better decisions. Ultimately, we hope this work will improve automated trading systems and help investors make more money in the stock market.(Yang et al. 1997)

This dataset comprises stock information from 30 different companies spanning various industries in the United States. It covers the period from January 2, 2009, to August 17, 2020. These companies include tech giant Apple Inc. (AAPL), financial services provider American Express Company (AXP), aerospace manufacturer The Boeing Company (BA), industrial machinery company Caterpillar Inc. (CAT), and consumer goods company The Coca-Cola Company (KO), among others. This dataset consists of 13 variables and 87,780 rows of data. Each row represents a unique observation, while the variables provide details such as date, stock ticker symbol, adjusted closing price, opening price, highest price, lowest price, trading volume, MACD (Moving Average Convergence Divergence), RSI (Relative Strength Index), CCI (Commodity Channel Index), ADX (Average Directional Index), and turbulence.

For this project, we aim to harness the capabilities of deep reinforcement learning (RL) to develop sophisticated strategies for automated trading. The core of our strategy will involve dynamically adjusting the portfolio by determining optimal weights for various stocks and deciding whether to buy or sell at each decision point. We plan to use a state space that includes historical price data, technical indicators (such as moving averages, RSI, and MACD), market volume, volatility, recent actions, and current holdings of the portfolio. The action space will be discretized into three main actions: buying, selling, or holding stocks. To optimize these decisions over the long term, we will implement the Proximal Policy Optimization (PPO), Deep Deterministic Policy Gradient (DDPG), and Advantage Actor-Critic (A2C) algorithm, known for its stability and effectiveness in enhancing policy learning through gradient ascent. We will assess the model's performance by calculating the Sharpe Ratio and cumulative returns after each transaction period, experimenting with one quater cycle to understand the impact of trading frequency on returns. This quantitative measurement will help determine the risk-adjusted return, crucial for navigating the volatile stock market. By directly linking the model's output to the trading agent, we enable real-time decision-making, thereby minimizing response times to market fluctuations.

Literature survey

Automated Trading

In recent years, automated trading in financial market has been increasingly widespread. Unlike traditional methods, automated trading relies on algorithms designed by traders based non-selected factors, with trades executed by pre-set code. Today, one can conduct trading even from a laptop at home.

Automated trading systems enhance efficiency of trading by reducing the probability of mistakes, increasing the frequency of trades, and improving the accuracy of executions.(Li, Burns, and Hu 2016)

Reinforcement Learning for Automated Trading System

Reinforcement learning is a crucial component of automated trading, where the environment is the dynamic financial market itself,

and agents are the trained models or algorithms. These reinforcement learning models autonomously make decisions by analyzing various states in the stock market, such as opening and closing prices, momentum factors and reference indices like RSI(Relative Strength Index) and MACD(Moving Average Convergence Divergence). They then take actions following a trained policy.

Several key studies demonstrate the successful application of reinforcement learning (RL) in the financial domain, specifically in the context of stock trading strategies. For instance, Zihao Zhang et al. (2019) in their paper "Deep Reinforcement Learning for Trading" explore the use of Deep Q-Networks (DQN) for trading. This research highlights how deep reinforcement learning can adapt to various market conditions and outperform traditional quantitative strategies in terms of profitability.For the financial trading task, the state-of-the-art deep recurrent Q-network (DRQN) algorithm is suitable.(Huang 2018)

Another notable study is "A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem" by Jiang, et al. (2017). This research introduces a portfolio management framework that utilizes the Deep Deterministic Policy Gradient (DDPG) model. The findings indicate that this reinforcement learning model can effectively manage and rebalance portfolios in real-time, achieving significant returns compared to traditional benchmarks.

Methodology

We applied a progressive training and validation method that uses a rolling window strategy to continuously expand the training set and incrementally update the validation set to adapt to the dynamic changes in financial markets. We started with data from 2009 to 2015 as the initial training set, followed by the next quarter as the first validation set. During this phase, we trained three reinforcement learning models: A2C, PPO, and DDPG. We selected the best-performing model based on the Sharpe Ratio, a measure of risk-adjusted return.This method not only enhanced the model's adaptability to market changes but also effectively prevented overfitting, ensuring stable performance on unseen data. Through this strategy, we ensured that our model could capture and reflect the key features and dynamics of financial time series data.

Description of Environments

In our study, we have developed three specialized OpenAI Gym environments for simulating stock trading during training, validation, and actual trading phases.In the environment we set up, each state is defined by several key components: the adjusted closing prices of the day, the number of shares held for each stock, and four important technical indicators—Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Commodity Channel Index (CCI), and Average Directional Index (ADX). These indicators collectively provide detailed information about the market conditions and the investment portfolios. The system's reward mechanism is defined as the net growth of assets from the beginning to the end of a stage, specifically the final total assets minus the initial total assets. This motivates the model to adopt strategies that increase the total assets.This approach allows for a comprehensive evaluation of the trading strategies developed.Brockman et al.2016

Training Environment (StockEnvTrain) In the training environment, we simulate real market data to train our trading models. This environment provides real-time feedback including price movements and various technical indicators such as MACD, RSI, CCI, and ADX. Participants can buy and sell stocks with the goal of maximizing total account assets through trading actions.

Validation Environment (StockEnvValidation) Our validation environment employs a separate dataset to evaluate the robustness and generalization of the trading strategies outside their training data. we have set a market volatility threshold to test the strategy's performance under extreme market conditions, ensuring that the trading strategies remain stable across different market states.

Trading Environment (StockEnvTrade) In the trading environment, strategies undergo a real-world test, where they make live trading decisions based on current market data and historical performance. This environment not only captures all trading activities and associated costs but also computes performance metrics like the Sharpe ratio, aiding in the evaluation of the strategies' long-term investment value.

Training Algorithms

We employ several reinforcement learning algorithms to train models within a custom training environment (`env_train`). Each algorithm is utilized with its respective implementation to evaluate its performance over a set number of timesteps (10,000). Below is a detailed description of the training process for each algorithm.

Advantage Actor-Critic (A2C) The A2C algorithm is an enhancement of the traditional Actor-Critic method, incorporating synchronous updates to stabilize training.[1]The algorithm maintains two neural networks: the actor, which updates the policy parameters in the direction suggested by the advantage function, and the critic, which evaluates the action taken by the actor using the value function. The objective function for the actor is given by:

$$\nabla_\theta J(\theta) = \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A(s_t, a_t) \tag{1}$$

where $\pi_\theta(a_t|s_t)$ is the policy network, and $A(s_t, a_t)$ is the advantage function, which can be expressed as:

$$A(s_t, a_t) = r(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t) \tag{2}$$

(Mnih 2016)

Deep Deterministic Policy Gradient (DDPG) The DDPG algorithm is designed for environments with continuous action spaces. It combines the actor-critic approach with deterministic policy gradients. The actor network outputs a deterministic action, while the critic network evaluates this action. To encourage exploration, DDPG adds noise to the action taken by the actor. In this study, we use Ornstein-Uhlenbeck noise, which is temporally correlated and suitable for physical control problems. The deterministic policy gradient is given by:

$$\nabla_{\theta^\mu} J \approx \mathbb{E}\left[\nabla_a Q(s, a|\theta^Q) \nabla_{\theta^\mu} \mu(s|\theta^\mu)\right] \tag{3}$$

At each time step, the DDPG agent performs an action at state s_t, receives a reward r_t, and arrives at state s_{t+1}. The transitions (s_t, a_t, r_t, s_{t+1}) are stored in the replay buffer R. A batch of N transitions are drawn from R and the Q-values are updated as:

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}\right), \quad i = 1, \ldots, N \tag{4}$$

The critic network is then updated by minimizing the loss function $L(\theta)$, which is the expected difference between outputs of the target critic network Q' and the critic network Q:

$$L(\theta) = \mathbb{E}_{s_t, a_t \sim R}\left[\left(y_i - Q(s_i, a_i|\theta^Q)\right)^2\right] \tag{5}$$

(Lillicrap 2015)

Proximal Policy Optimization (PPO) The PPO algorithm optimizes the policy by ensuring that the updates are not too large, maintaining a balance between exploration and exploitation. It achieves this by using a clipped surrogate objective. The PPO objective function is:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{6}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the advantage estimate, and ϵ is a hyperparameter that controls the clipping range. This approach helps in maintaining the stability of updates by restricting the change in the policy within a specified range.(Schulman 2017)

In summary, this methodology leverages various reinforcement learning algorithms, each with its unique approach to optimizing policy learning. The implementation of these algorithms in a controlled environment allows for a comparative analysis of their performance and training efficiency.

Result

We established a rolling window of one quarter as the test set to evaluate three selected reinforcement learning models—PPO, A2C, and DDPG—based on their Sharpe ratios. We then selected the best-performing model at each validation point to formulate our strategy.

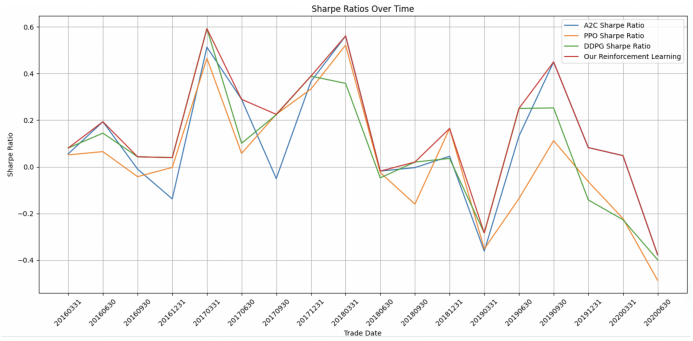

Figure 1: Sharpe ratios of three different reinforcement learning models—A2C, PPO, and DDPG—across various rolling validation periods

Figure 1 illustrates the performance of three reinforcement learning models—A2C, PPO, and DDPG—across various quarterly validation periods, as reflected on the x-axis marking the end of each quarter. The y-axis measures the Sharpe Ratio, a gauge of risk-adjusted return. Notably, the A2C model frequently leads in performance with peaks surpassing 0.5 Sharpe Ratio in early 2017 and a significant trough dipping to approximately -0.4 in mid-2017, indicating its high volatility but occasionally superior performance. The PPO model, on the other hand, exhibits more moderate fluctuations, reaching its highest at around 0.4 Sharpe Ratio in early 2018, while the DDPG closely tracks PPO but sometimes falls behind, as seen in late 2020 when its Sharpe Ratio approached -0.3.

The strategy outlined by the "Our Reinforcement Learning" line (red), which adapts by selecting the model with the highest Sharpe Ratio at each validation point, is evident in its close tracking of the best-performing model curve at any given time. This adaptive approach aims to optimize returns by efficiently managing risk and highlights the importance of dynamically adjusting to changing market conditions. For instance, in late 2018, this strategy closely mirrored the A2C model when it rebounded to a Sharpe Ratio of approximately 0.6, demonstrating an effective response to market opportunities. This strategy underlines the necessity for continuous refinement of models and robust risk management to maintain a competitive edge in the volatile realm of financial trading.

Table 1: Sharpe Ratios for A2C, PPO, and DDPG Over Time

Date	A2C SR	PPO SR	DDPG SR	Max SR
2016-03-31	0.056	0.051	0.081	0.081
2016-06-30	0.193	0.065	0.144	0.193
2016-09-30	-0.010	-0.042	0.043	0.043
2016-12-31	-0.138	-0.003	0.039	0.039
2017-03-31	0.513	0.465	0.592	0.592
2017-06-30	0.289	0.058	0.101	0.289
2017-09-30	-0.050	0.225	0.223	0.225
2017-12-31	0.366	0.333	0.389	0.389
2018-03-31	0.561	0.521	0.358	0.561
2018-06-30	-0.018	-0.025	-0.047	-0.018
2018-09-30	-0.004	-0.160	0.020	0.020
2018-12-31	0.045	0.164	0.036	0.164
2019-03-31	-0.362	-0.352	-0.284	-0.284
2019-06-30	0.131	-0.136	0.250	0.250
2019-09-30	0.450	0.112	0.254	0.450
2019-12-31	0.082	-0.064	-0.142	0.082
2020-03-31	0.048	-0.221	-0.227	0.048
2020-06-30	-0.377	-0.487	-0.398	-0.377

Table 1 provides a detailed comparison of Sharpe Ratios for three different reinforcement learning models—A2C, PPO, and DDPG—over multiple quarterly test periods from March 2016 to June 2020. Notably, the A2C model frequently achieves high Sharpe Ratios, peaking at 0.513 in March 2017 and again reaching 0.561 in March 2018, indicating periods of strong performance relative to risk. However, it also shows significant volatility, with its Sharpe Ratio dropping to as low as -0.377 in June 2020. The PPO and DDPG models exhibit less extreme fluctuations but generally underperform compared to the A2C in terms of the maximum Sharpe Ratio achieved in each period.

From the analysis, it is evident that the selection strategy of adopting the model with the highest Sharpe Ratio at the end of each quarter can dynamically adjust to market conditions, optimizing returns relative to risk. For instance, the highest overall Sharpe Ratios in the table, marked under "Max SR", typically correspond to the best-performing model for each period, reflecting the effectiveness of this model-switching strategy. This approach minimizes periods of underperformance and leverages the strengths of each model, as demonstrated in late 2017 and early 2018, where switching between models captured upward shifts in performance notably during quarters where certain models like DDPG outperformed others.

Conclusion

At the outset of this project, our primary objective was to develop a dynamic trading strategy that could adapt to varying market conditions by leveraging the strengths of different reinforcement learning models. Specifically, we aimed to evaluate the performance of three distinct models—A2C, PPO, and DDPG—over sequential quarterly periods, using Sharpe Ratios as a metric to assess the risk-adjusted returns of each model. Our strategy was designed to select the model with the highest Sharpe Ratio at the end of each validation period, thereby optimizing our investment returns while managing risk.

Over the course of this study, we have successfully implemented

and evaluated the proposed strategy. The results, detailed in the accompanying table, demonstrate the efficacy of our approach. By systematically selecting the model that exhibited the highest Sharpe Ratio at the end of each quarter, we were able to capitalize on the best-performing model's strengths during subsequent periods. This method proved particularly beneficial during times of significant market fluctuations, as it allowed our strategy to adapt to changing market dynamics and maintain a competitive edge.

Through this research, we have not only validated the feasibility of using Sharpe Ratios for model selection in a rolling training and validation framework but have also highlighted the potential of reinforcement learning models to significantly enhance trading strategies. The adaptability of our approach, demonstrated by its capacity to switch models based on their performance, underscores its practical application in real-world trading scenarios where market conditions are constantly evolving.

In conclusion, the project has met and exceeded our initial objectives, providing a robust framework for dynamic model selection in algorithmic trading. Our findings offer valuable insights into the application of machine learning in financial markets and set a strong foundation for future research to build upon, particularly in optimizing these models further and exploring additional metrics for model evaluation.

References

Yang, Hongyang and Liu, Xiao-Yang and Zhong, Shan and Walid, Anwar, Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy (September 11, 2020). Available at SSRN: https://ssrn.com/abstract=3690996 or http://dx.doi.org/10.2139/ssrn.3690996.

Li, Y., Burns, C., and Hu, R. (2016). Representing Stages and Levels of Automation on a Decision Ladder:The Case of Automated Financial Trading. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 60(1), 328-332. https://doi.org/10.1177/1541931213601074

Zhang, Z., Zohren, S., and Roberts, S.J. (2019). Deep Reinforcement Learning for Trading. The Journal of Financial Data Science..

Chien Yi Huang. (2018). Financial Trading as a Game: A Deep Reinforcement Learning Approach. arXiv.Org. https://doi.org/10.48550/arxiv.1807.02787

Jiang, Z., Xu, D., and Liang, J. (2017). A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem. arXiv.Org. https://doi.org/10.48550/arxiv.1706.10059

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W. (2016). OpenAI Gym. arXiv preprint arXiv:1606.01540.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. arXiv preprint arXiv:1602.01783.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D. (2019). Continuous Control with Deep Reinforcement Learning. arXiv preprint arXiv:1509.02971.

Recht, B., Ré, C., Wright, S., Niu, F. (2017). Scalable and Sustainable Deep Learning via Randomized Hashing. arXiv preprint arXiv:1707.06347.