FinRL Contest 2024 Task I: Cryptocurrency Trading with Ensemble Learning

Can You
cyoucandice@gmail.com
FTI Consulting
New York, NY, USA

Bing Cheng
bcheng33@gmail.com
Ernst & Young
New York, NY, USA

Abstract

Reinforcement learning (RL) is a powerful technique to derive optimal strategies for various financial tasks such as algorithmic trading and portfolio optimization. In this study, we developed robust and effective trading agents for cryptocurrencies trading using ensemble reinforcement learning (ERL). Specifically, we use Bitcoin as an example to demonstrate the effectiveness of the RL agents. We conduct a detailed analysis on model selection using both on-policy learning and off-policy learning algorithms, and explore the application of various ensemble learning methods. Subsequently, we analyze the performance of both individual and ensemble agents using various risk and performance metrics. Finally, we discuss potential research directions to improve the training process and performance of the agents.

Keywords

Reinforcement Learning, Ensemble Learning, Cryptocurrency Trading, DDQN, D3QN, PPO, A2C, SAC, Vectorized Environment

ACM Reference Format:
Can You and Bing Cheng. 2024. FinRL Contest 2024 Task I: Cryptocurrency Trading with Ensemble Learning. In *Proceedings of 5th ACM International Conference on AI in Finance (ICAIF '24).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

In this analysis, we utilize second-level Limit Order Book (LOB) data for Bitcoin provided in the starter kit. The reinforcement learning state consists of the position, holding period, and 8 strong RNN features, which are trained to predict future price movements based on 101 formulaic alphas. [1][3].

We explored ensemble reinforcement learning approach that combines state-of-the-art reinforcement learning algorithms to improve training stability and performance. We also take advantage of the power of massively parallel simulations by utilizing the provided vectorized environments.

ICAIF '24, November 14–17, 2024, New York, NY

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

2 Individual Reinforcement Learning Agents

This section provides an overview of on-policy and off-policy reinforcement learning algorithms employed to train individual reinforcement learning models. The training procedures and parameter configurations are also documented.

In addition to the Double Deep Q-Network (DoubleDQN)[10] and the Dueling Double Deep Q-Network (D3QN)[11] provided in the starter kit [1], we incorporated three additional agents: Proximal Policy Optimization (PPO)[7], Advantage Actor-Critic (A2C)[6], and Soft Actor-Critic (SAC)[2]. Given the discrete nature of the action space, other algorithms, such as Deep Deterministic Policy Gradient (DDPG)[4][9], were explored but ultimately not selected. To be compatible with evaluation framework, the models are implemented following the style and conventions of ElegantRL[5].

2.1 Off-Policy Approaches

We trained off-policy agents based on DoubleDQN, D3QN and SAC algorithms.

For DoubleDQN and D3QN, the neural network architectures provided are used for training.

The models are trained using 4096 parallel simulations of environments, which are randomly sampled from the available price and order book features dataset. The number of training time steps is increased from 320,000 to 1,280,000 to facilitate more exploration of the environment and to improve the convergence of the algorithms. This applies to training for all agents unless specified otherwise.

We incorporated additional neural network architectures for SAC and modified the traditional SAC method to accommodate discrete action spaces. To handle the increased memory usage, we simulate 1024 environments in parallel during training of SAC agent.

2.2 On-Policy Approaches

Off-policy learning provides better sample efficiency while on-policy learning provides more stability. To take advantage of both, we added on-policy agents using PPO and A2C algorithms.

PPO and A2C are implemented for both continuous and discrete action spaces, providing the flexibility to convert trading actions into a continuous space.

To balance exploration and exploitation, various learning rates are tested. Different architectures for the actor and critic neural networks are also explored.

The entropy term is added on top of the traditional A2C objective to avoid premature convergence to suboptimal deterministic policies and to encourage exploration.

3 Strategies for Ensemble Learning

Ensemble reinforcement learning is a method that combines different reinforcement learning training methods into a common framework to make joint predictions. Ensemble learning can reduce both the variance and bias of individual reinforcement learning (RL) agents by leveraging multiple models to improve prediction stability and accuracy. This section presents an overview of the ensemble methods used in the analysis.

3.1 Majority Voting

Majority voting method is the default approach where the action taken by the largest number of agents is selected as the ensemble action.

3.2 Confidence Level Based

The confidence level based approach add confidence score as a voting weight. Each action use the corresponding Q-value as the confidence score, and the cumulative Q-values from all agents are aggregated for each action. The action with the highest cumulative Q-value is selected as the ensemble action. This approach will not only account for the top action but also the confidence on the optimal action or the remaining actions.

3.3 Boltzmann Addition

To address the differences in objective and Q-value calculations across various approaches, it is beneficial to transform the Q-values into Boltzmann probabilities[8][12].

The sum of Boltzmann probabilities from all agents is then used to evaluate each action, with the action having the highest score selected as the ensemble action.

3.4 Boltzmann Multiplication

An alternative method is Boltzmann Multiplication [8][12]. Similar to Boltzmann Addition approach, the Q-values for each agent are first transformed to Boltzmann probabilities. Similar to the Boltzmann Addition approach, the Q-values for each agent are first converted into Boltzmann probabilities. The joint probability for each action is then computed as the product of the probabilities from all agents, similar calculating the joint probability of independent events. This joint probability is then used to score actions, and the action with the highest joint probability is chosen as the ensemble action.

4 Results

We evaluated individual agents and ensemble methods using extensive risk and return metrics. As seen in Figure 1, all agents achieved over 2% Sharpe ration and over 75% win rate. The Sharpe ratio is reasonable considering that the return horizon is one second. Note that the performance metric would also depend on market conditions in the test set.

The ensemble approaches provide a good balance between various risk and return metrics. The ensemble methods based on Q-values and Boltzmann probability provides more stable results and achieved Sharpe ratio of 8% while maintaining the return over max drawdown at 11.8 and win rate at 75%. The performance is better than majority voting as the probability distribution of all actions is utilized to determine the ensemble action, leading to more informed decision-making.

5 Discussions

There are several potential steps we can take to further improve the reinforcement learning trading algorithm.

Firstly, we could consider aligning the reward function used for training with the one used for evaluation. For instance, while the training is based on cumulative return, in evaluation we considers additional metrics such as Sharpe ratio and maximum drawdown. By incorporating these metrics into the training reward at the completion of each episode, we can achieve a balance between optimizing returns and minimizing risk, and improved performance during evaluation.

Furthermore, we could introduce more flexibility in the environment to enable the discovery of more optimal strategies. For instance, allowing the trading of fractional bitcoins within a continuous action space, instead of a discrete space with only three possible values, would support more refined optimization and yield better strategies. Additionally, permitting short selling of bitcoin, which can be executed in futures markets, could also expand the action space and achieve better risk adjusted returns. Relaxing these constraints could potentially enhance the algorithm's overall performance.

References

[1] ACM ICAIF 2024 FinRL Contest. 2024. Task 1 Starter Kit. Retrieved November 8, 2024 from https://open-finance-lab.github.io/finrl-contest-2024.github.io/

[2] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. 2018. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905* (2018).

[3] Zura Kakushadze. 2016. 101 formulaic alphas. *Wilmott* 2016, 84 (2016), 72–81.

[4] TP Lillicrap. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[5] Xiao-Yang Liu, Zechu Li, Zhuoran Yang, Jiahao Zheng, Zhaoran Wang, Anwar Walid, Jian Guo, and Michael I Jordan. 2021. ElegantRL-Podracer: Scalable and elastic library for cloud-native deep reinforcement learning. *NeurIPS, Workshop on Deep Reinforcement Learning* (2021).

[6] Volodymyr Mnih. 2016. Asynchronous Methods for Deep Reinforcement Learning. *arXiv preprint arXiv:1602.01783* (2016).

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[8] Yanjie Song, Ponnuthurai Nagaratnam Suganthan, Witold Pedrycz, Junwei Ou, Yongming He, Yingwu Chen, and Yutong Wu. 2023. Ensemble reinforcement learning: A survey. *Applied Soft Computing* (2023), 110975.

[9] Haining Tan. 2021. Reinforcement learning with deep deterministic policy gradient. In *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*. IEEE, 82–85.

[10] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[11] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1995–2003.

[12] Marco A. Wiering and Hado van Hasselt. 2008. Ensemble Algorithms in Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 38, 4 (2008), 930–936. https://doi.org/10.1109/TSMCB.2008.920231

Agent	Initial Asset	Final Asset	PnL	Mean Return	Volatility	Sharpe Ratio	Max Drawdown	Return over Max Drawdown	Win Rate
AgentDoubleDQN	1000000	1002668	2668.4	0.00011%	0.00492%	2.290%	-0.013%	21.13	76.8%
AgentD3QN	1000000	1000546	546.2	0.00002%	0.00028%	8.283%	-0.005%	10.50	75.4%
AgentDiscretePPO	1000000	1005158	5158.2	0.00022%	0.00983%	2.214%	-0.032%	16.21	77.2%
AgentDiscreteA2C	1000000	1005193	5193.0	0.00022%	0.00983%	2.229%	-0.037%	14.06	76.0%
AgentDiscreteSAC	1000000	1000290	289.6	0.00001%	0.00035%	3.476%	-0.012%	2.38	77.8%
Ensemble: Majority_Voting	1000000	1005015	5015.3	0.00021%	0.00983%	2.152%	-0.043%	11.59	75.7%
Ensemble: Confidence_Based	1000000	1000556	555.7	0.00002%	0.00028%	8.429%	-0.005%	11.84	75.3%
Ensemble: Boltzmann_Addition	1000000	1000554	554.2	0.00002%	0.00028%	8.406%	-0.005%	11.81	75.4%
Ensemble: Boltzmann_Multiplication	1000000	1000556	555.7	0.00002%	0.00028%	8.429%	-0.005%	11.84	75.3%

Figure 1: Performance for individual agents and ensemble methods.