A Deep-Reinforcement-Learning Based High Frequency Trading Strategy

Group Number: A390 Group Member: 周光祈、歐羿辰、黃霈紳 Professor: 馬席彬

Abstract

With computer tech and AI advancement, high-frequency trading's stability and efficiency gain attention. Prior research shows deep reinforcement learning's use here. We explore its potential, aiming to create trading strategies.

We suggest a trading agent using proximal policy optimization, an actor-critic algorithm. It processes tick-by-tick trading messages, generating trading behavior. Our goal is good performance considering trading costs. We optimize strategy model efficiency using quantized fixed-point arithmetic, testing various quantization bit widths. Experiments show benefits in simulated trading.

Overview on PPO

The Proximal Policy Optimization (PPO) algorithm employs an off-policy method, aiming to maximize the objective function as shown in Eq. 1.

$$L^{CLIP+VF+S}(\theta) = E_{(s_t,a_t)\sim\pi_{\theta'}}[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)], \quad (1)$$

$$L_t^{VF}(\theta) = \left(V_\theta(s_t) - V_{\theta'}(s_t)\right)^2, \quad (2)$$

$$L_t^{CLIP}(\theta) = \min\left(\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)} A^{\theta'}(s_t,a_t),\ clip\left(\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)},\ 1-\varepsilon,\ 1+\varepsilon\right) A^{\theta'}(s_t,a_t)\right), (3)$$

Research Methodology

Financial markets feature limit and market orders. A limit order sets a buy/sell price, matched using the limit order book (LOB). Unmatched orders go to the bid/ask side of the LOB. Bid orders rank by highest price, with the first termed the best bid price. Ask orders rank by lowest price, with the first termed the best ask price. Usually, the best bid price is lower than the best ask price, forming a bid-ask spread (see Fig 1).

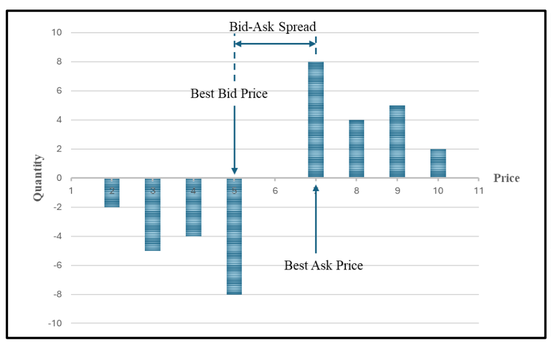

Fig. 1 Sketch of LOB

Action Number	Description
0	Do nothing
1 (Buy)	Buy one lot at the current P_{Ask}
2 (Sell)	Sell one lot at the current P_{Bid}

Table 1 Action Space of Model

The Actor-Critic neural network adopts a three-layer Fully-Connected (FC) Layer architecture. Leaky ReLU is chosen as the Activation Function for the Hidden Layer. Unlike ReLU, Leaky ReLU prevents neuron shutdown with negative inputs, avoiding "Dying ReLU." This suits scenarios with both positive and negative input values. A schematic diagram is illustrated in Fig. 2.

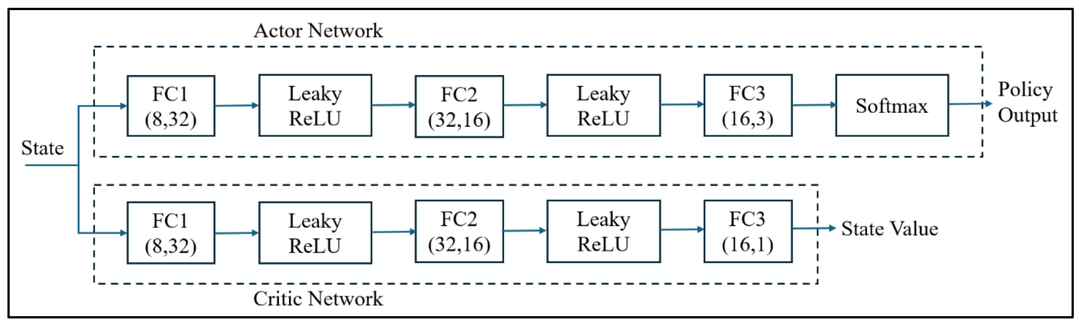

Fig. 2 Framework of Actor-Critic Neural Network

Our reward function assumed actions occurred before a_t, leading to potential losses from buying high and selling low due to rapid position accumulation. To mitigate this, we introduce "Action-Reversing." By observing price movements, we set a small range around the opening price. If the price falls within this range, the model's buy action becomes sell, and vice versa, creating a reverse position. This continues until the price breaks this range, allowing normal actions to close these reverse positions. Adjusting the range is crucial, and typically, sharp price rises lead to higher closing prices and vice versa.

Experimental Result

The training results of the model are depicted in Fig. 2. It can be observed that due to the slightly larger learning rate, there is still a small amount of oscillation in the model performance towards the end. However, overall, the model converges and the reward is positive, indicating that the model has learned the expected behavior of the reward function.

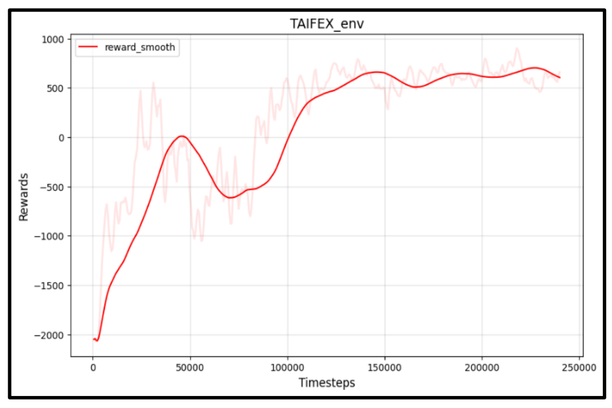

Fig. 2 Training Result (Converge)

Simulation Parameter	Value
Initial cash	10,000,000
Maximum position	±223
Trading days	2022/06/13~14、2022/10/24~26、2022/10/31
Trading time	09:30 ~ 13:00

Table 2 Information about Testing

During testing, we start with an initial capital of 10,000,000 NTD. With Mini-TAIEX futures' margin set at 44,750 NTD per contract, falling below this triggers a margin call. Hence, position size aligns with available capital. Regardless of long or short positions, the maximum size is 223 lots. Table 2 details simulation info, tested daily. Model performance is shown in Fig. 3, while Fig. 4 compares performance with and without action-reversing.

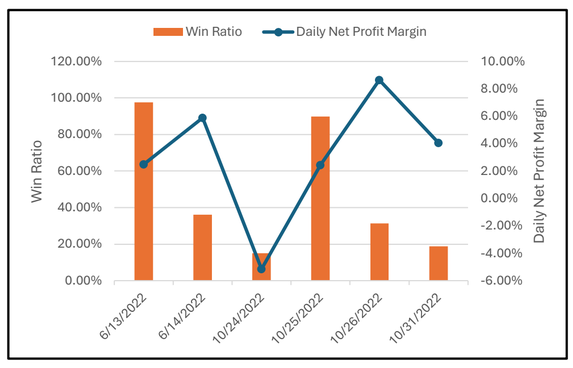

Fig. 3 Performance on 6 Trading Days

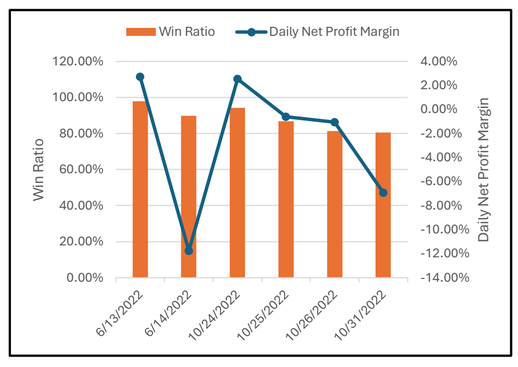

Fig. 4 Performance on 6 Trading Days Without Action-Reverse

Quantizing the actor network aims to utilize efficient operations with integers or fixed-point numbers. Plotting the results using different numbers of bits and finding the minimum bit number that hardly affects performance, as shown in Fig. 5, it can be observed that it is approximately between 8 to 10 bits.

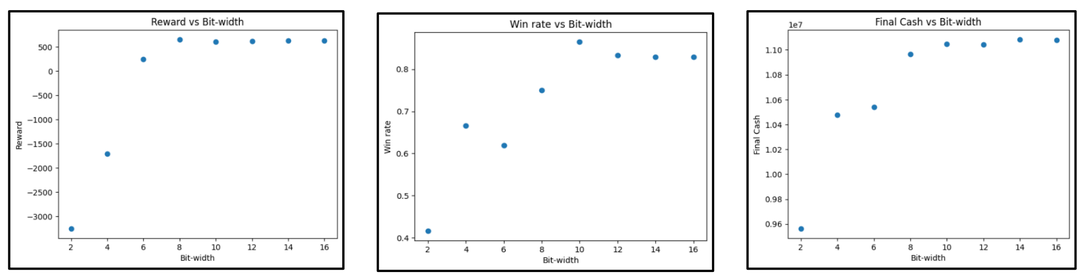

Fig. 5 Performance Corresponded with Arbitrary Quantization bits

Finally, testing different combinations of bit width for each layer, we obtained the minimum bit width that hardly affects performance. It can be seen that from originally having full precision (32-bit) for each layer to eventually using only (8-bit, 5-bit, 8-bit), a significant amount of computation is saved, and in terms of test results, the performance is hardly affected. If we relax the constraint to only preserve over 80% of the performance (targeting daily net profit margin), we can further reduce it to (8-bit, 3-bit, 8-bit), as shown in Table 3. Result 2 in the table represents the result allowing a 20% performance loss.

	Full precision			Result 1			Result 2		
Bit width	FC1	FC 2	FC 3	FC1	FC 2	FC3	FC1	FC 2	FC 3
	32	32	32	8	5	8	8	3	8
Win rate	82.93%			82.86%			86.05%		
Net Profit	10.59%			9.86%			8.06%		

Table 3 Comparison of Performance Between Full Precision and Quantized Ones

Conclusion

This project merges deep reinforcement learning with Priority Queue data structure, analyzing past trading messages with conditional judgment methods. Initial tests on tick-by-tick trading data show promise. The PPO agent learns action strategies effectively, optimizing the reward function. Experiments highlight its potential. Furthermore, in subsequent model quantization, we determine the optimal bit width, enhancing computational efficiency and reducing model size with minimal performance impact.