

MAFS 5370 Assignment 2 Report

Zihua SHE 20921494

Peng GAO 20931528

1 Question

1.1 Problem Statement

In a binomial model of a single stock with non-zero interest rate, assume that we can hedge any fraction of a stock, use policy gradient to train the optimal policy of hedging an ATM American put option with maturity $T = 10$. When do you early exercise the option? Is your solution same as what you obtain from delta hedging?

1.2 Substitute

For the really advanced students: suppose the stock follows a GBM, construct an algorithm to train a NN that hedges an ATM American put option.

2 Solution

2.1 PPO Introduction

This part's result and code can be find in the notebook [PPO.ipynb](#).

The two question are nearly the same. The first difference is the using method (gradient policy and Neural Network), the second difference is the model setting (binomial tree and Geometry Brownian Motion).

For Policy Gradient method,

$$\hat{g} = \hat{\mathbb{E}}_t \left[\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{A}_t \right]$$
$$L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_{\theta}(a_t | s_t) \hat{A}_t \right]$$

The conventional Policy Gradient method can just use the samples to update one time. Then we need to drop the sample and conduct sampling one more time to achieve updating.

After some literature review (Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms), we find the PPO method using the Policy Optimization and Neural Network to solve the problem. So in this part for assignment study, we use PPO method to solve the American put option hedging problem under binomial and GBM setting.

Algorithm 1 PPO, Actor-Critic Style

```
for iteration=1,2,... do
  for actor=1,2,...,N do
    Run policy  $\pi_{\theta_{old}}$  in environment for  $T$  timesteps
    Compute advantage estimates  $\hat{A}_1, \dots, \hat{A}_T$ 
  end for
  Optimize surrogate  $L$  wrt  $\theta$ , with  $K$  epochs and minibatch size  $M \leq NT$ 
   $\theta_{old} \leftarrow \theta$ 
end for
```

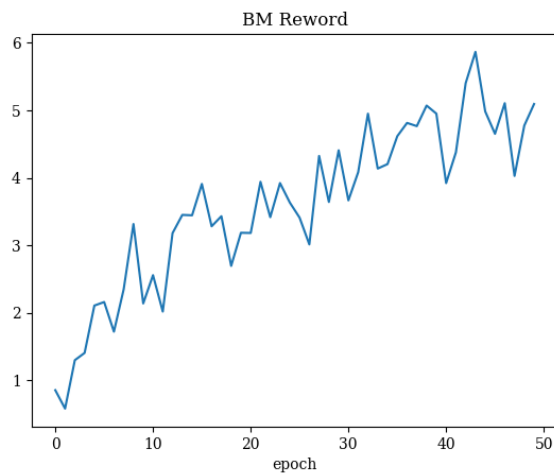
The parameter setting is, $S_0 = 100.0$, $K = 100.0$, $T = 10.0$, $r = 0.05$, $\sigma = 0.2$. The state is the stock price, the action is how many stocks to buy or sell, and the reward is the payoff.

We define the class `PPO`, `CRR` (for binomial tree case) and function `gbm` (for GBM case). And our goal is to maximize the reward. The point for early exercise need that exercise would maximize the reward.

In the code, we train 1000 episodes, and try to plot the average of 20 rewards in one epoch (50 epoch, 20 episodes in one epoch).

2.2 CRR (Binomial tree case)

Here are the graph for reward.



We can find that when episodes increasing, the reward becomes larger and closer to its optimal value.

Since we can find the reward function of each state, we would early exercise the american option when we would maximize the reward with exercise.

2.3 GBM

Here are the graph for reward.



We can find that when episodes increasing, the reward becomes larger and closer to its optimal value.

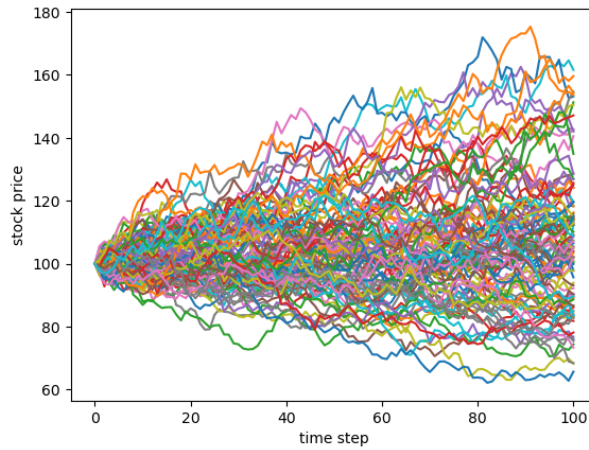
Since we can find the reward function of each state, we would early exercise the american option when we would maximize the reward with exercise.

2.4 NN hedging for Substitute question

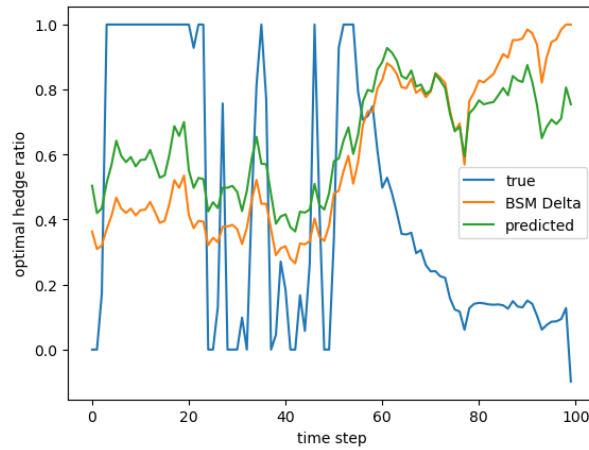
This part's result and code can be find in the notebook [NN_Hedging.ipynb](#).

In this part we use Neural Network to hedge an ATM American put option under GBM.

Firstly we Generate Stock Path. We set $S_0 = 100$, $r = 0.05$, $\sigma = 0.2$, $T = 1$, $n_{sim} = 100$, $n_{steps} = 100$.



Based on this, we try to train a Neural Network model. After training we test the NN and repeat to generate different test price path.



The above plot shows the true optimal hedging ratio, BSM hedging ratio and hedging ratio predicted by NN model. During the training process, the input of the neural network is the spot price and the volatility and the output is the hedge ratio. We never use the data from the BSM. However, after training the NN model with the simulation data, we find that the predicted result follows the hedging ratio given by BSM, which means NN model finds the pattern that is quite similar to the BSM. Hence, under GBM using the NN to train the model, the option price will still converge to the BSM formula.