Application of Neural Network Actor-Critic Method in Hedging of ATM European Call Option

Group members:
MA Rongyue 20826084
SHI Hengtao 20823252

May 3rd, 2022

Abstract

In our project, our main methodology is to use the actor-critic model to train the optimal policy of hedging an ATM European call option. As we learned, the actor is a reference to the learned policy, and the critic refers to the learned value function, usually a state-value function.

Because of the non-linear relationship between c and S, critic and actor functions are parameterized with neural networks, which is good to handle non-linearity and multiple time steps.

The project contains four main parts:
1. Stock Price Simulation and Option Pricing
2. Neural Network Actor-Critic Method
3. Result Evaluation
4. Comparison Between NN Actor-Critic Method with Delta Hedging Method Under a Binomial Model

Group Work Contribution:
We all actively participated in the initial discussion on the overall structure and ideas. SHI Hengtao is mainly responsible for coding, and MA Rongyue is mainly responsible for report writing. And we also supported and conducted cross-checking for others' parts and give suggestions.

1 Stock Price Simulation and Option Pricing

1.1 Stock Price Simulation

We simulate stock prices by the binomial model as follows.

```
# obtain the next stock price with binomial pricing method
def nextStock(self, S):
    isUp = np.random.rand() >= 0.5
    if isUp:
        S_next = S * self.su
    else:
        S_next = S * self.sd
    return S_next
```

We assume the probabilities of stock prices going up or down in a certain time interval are equal, and *su* and *sd* capture the volatility of underlying stock with the condition *su***sd* = 1. We set *su* as 1.1 and *sd* as 1/1.1 in our case.

1.2 ATM European Call Option Pricing

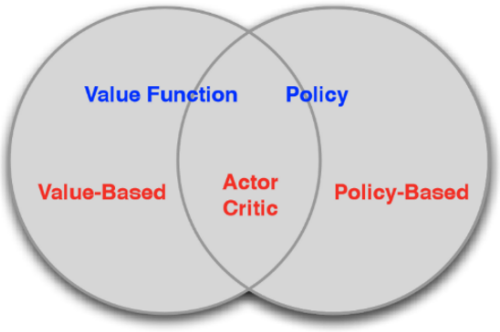
Formula of value of European Call Option Pricing at time t

$$c = 0.5 * (max(0, S * u - K)) + 0.5 * (max(0, S * d - K)) \tag{1}$$

S: spot price at time t
K: strike price
u*d = 1

2 Neural Network Actor-Critic Method

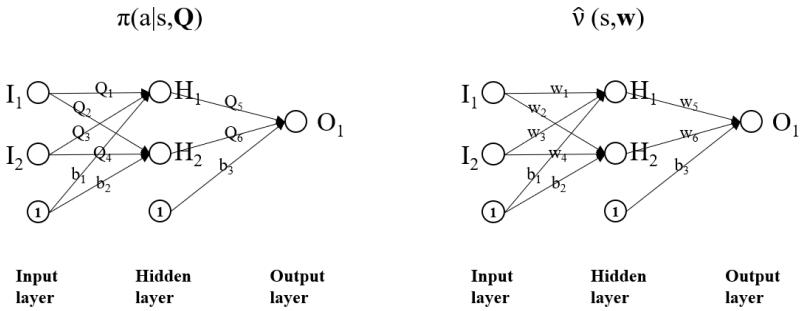
Compared with the value-based method and policy-based method, the Actor-Critic method has both policy and value functions.



2.1 Parameterizing Policy and State Value by NN

From the formula of ATM European Call Option, we find the relationship between c and S is non-linear. Among other methods, Neural Network is good at the non-linear mapping between the inputs and response variables, especially under the situation of multiple time steps. Therefore, we use NN to parameterize policy and state-value.

We denote the $\pi(a|s, \mathbf{Q})$ and $\hat{\nu}(s, \mathbf{w})$ by the two simplest neural networks with only one hidden layer.



The corresponding codes are as follows.

```
# construct value neural network
def valueNetwork(self, I1, I2):
    # construct hidden layer
    self.H_w[0] = I1 * self.W[0] + I2 * self.W[2] + self.b_w[0]
    self.H_w[1] = I1 * self.W[1] + I2 * self.W[3] + self.b_w[1]
    # construct output layer
    O1 = self.H_w[0] * self.W[4] + self.H_w[1] * self.W[5] + self.b_w[2]
    return O1

# construct action/Policy neural network (pi)
def actionNetwork(self, I1, I2):
    # construct hidden layer
    self.H_Q[0] = I1 * self.Q[0] + I2 * self.Q[2] + self.b_Q[0]
    self.H_Q[1] = I1 * self.Q[1] + I2 * self.Q[3] + self.b_Q[1]
    # construct output layer
    O1 = self.H_Q[0] * self.Q[4] + self.H_Q[1] * self.Q[5] + self.b_Q[2]
    return O1
```

2.2 Actor-Critic Method with Eligibility Traces

Action space: Because Neural Network can generate continuous outputs, we have continuous action space, which means we can hedge any fraction of a stock.

Goal: to perfectly hedge the ATM European call option

Reward function:
Reward = -1 * | European call option price in $t+1$ - European call option price in t – action *(stock price in $t+1$ – stock price in t |

Based on the goal, we pursue no gain and loss (profit = 0).

```
# define a reward function manually
def reward(self, C, C_next, S, S_next, act):
    diff = -1 * abs(C_next - C - act * (S_next - S))
    return diff
```

Parameter updates:
The parameter updating formulas are as follows. We train the parameters by the Gradient Temporal-Difference method (GTD method).

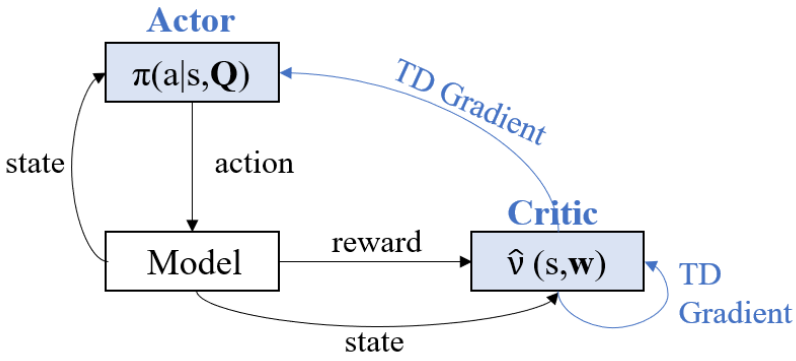
Update parameters in $\pi(a|s, \mathbf{Q})$:

$$Q_{t+1} = Q_t + \alpha \cdot (R_t - \bar{R} + \nu(S_{t+1,\mathbf{w}}) - \nu(S_{t,\mathbf{w}})) \cdot \frac{1}{\pi(A_t|S_t, \mathbf{Q})} \cdot \frac{\partial \pi(A_t|S_t, \mathbf{Q})}{\partial \mathbf{Q}} \tag{2}$$

Update parameters in $\hat{\nu}(s, \mathbf{w})$:

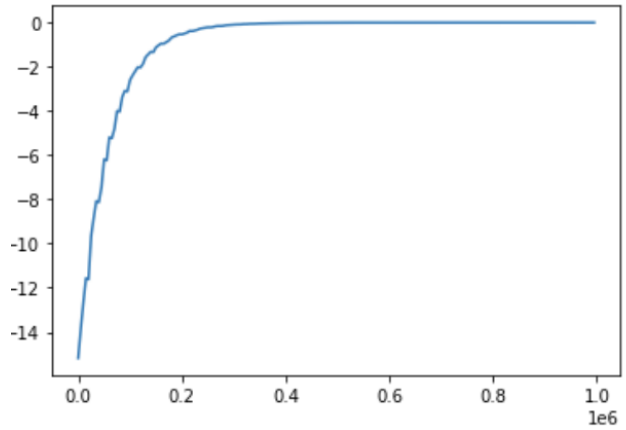
$$w_{t+1} = w_t + \beta \cdot (R_t - \bar{R} + \nu(S_{t+1,\mathbf{w}}) - \nu(S_{t,\mathbf{w}})) \cdot \frac{\partial \nu(S_t, \mathbf{w})}{\partial \mathbf{w}} \tag{3}$$

Summary of our model:



3 Result Evaluation

Case 1: Time step T=1
Please find below the reward graph. We can infer that the model performs quite well as the reward value converges to near-zero quickly.



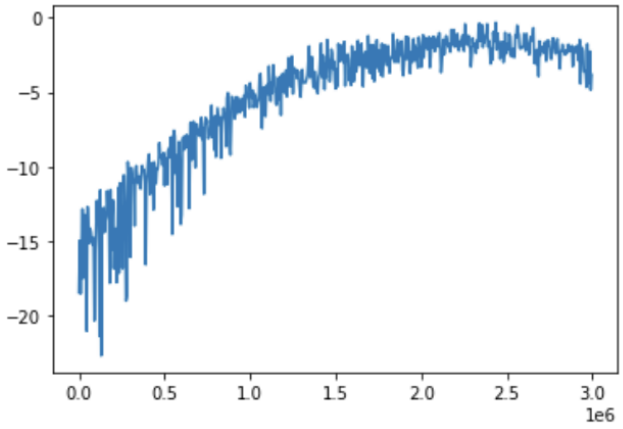
We can also output the average reward numbers in the last 10 episodes, and find they are very small.

```
In  [51]: model.avg_reward_track[-10:]

Out[51]: [-0.0029181677504395154,
          -0.0026538659988410984,
          -0.002652886915575081,
          -0.0026519078476594515,
          -0.00291602167460425 2,
          -0.002651915053901277,
          -0.002650936001225901,
          -0.0026499569639000242,
          -0.0029138757361213408,
          -0.0029149606572058673]
```

Case 2: Time step T=5

Please find below the reward graph. We can see the reward function has much higher oscillation and slower convergence compared with the simplest case of T=1. To improve the model, we might deepen our neural network and conduct parameter tuning further.



The reward values are shown below.

```
In  [69]: max(model.avg_reward_track)

Out[69]: -0.8515228715708707

In  [65]: model.avg_reward_track[-10:]

Out[65]: [-3.51538650267288,
          -2.0262195508530954,
          -2.2708733156818908,
          -2.0892546115300554,
          -2.1894037602267042,
          -2.9278844662330257,
          -2.216241634959423,
          -3.803458581087638,
          -2.2104027349507356,
          -2.958214662079879]
```

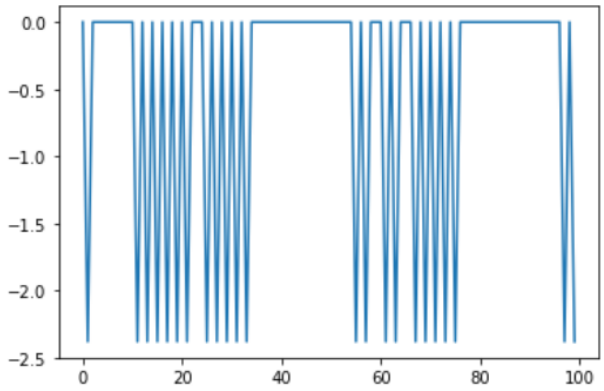
4 Comparison Between NN Actor-Critic Method with Delta Hedging Method Under a Binomial Model

Summary of Delta Hedge:

In the reward graph, we can find the reward value is highly volatile, and several points have zero rewards (which is most preferred).

Those zero-reward points refer to the fully hedged situation where the stock price is very high or low, s.t. delta = 1 or 0.

From the reward graph, we can see delta hedge doesn't work out well. We think it may be because the price change is discrete, therefore, delta = *dc/dS* can't perfectly fit the change of European call option price along with stock price.



5 Code File

The code can be found in Github via https://github.com/NicoleMa1220/RL-Project3.git