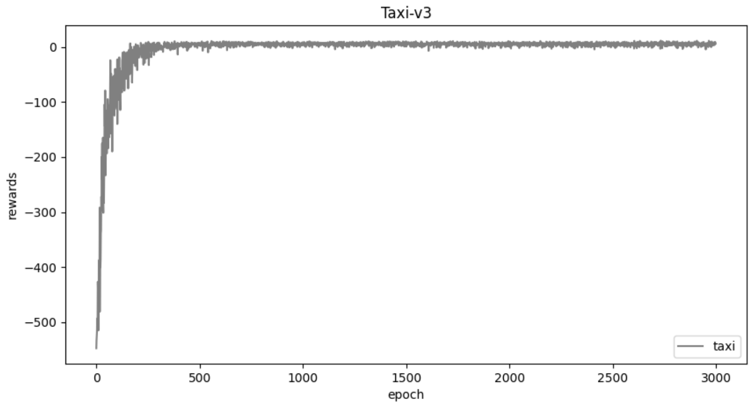
109550069_林宇柔

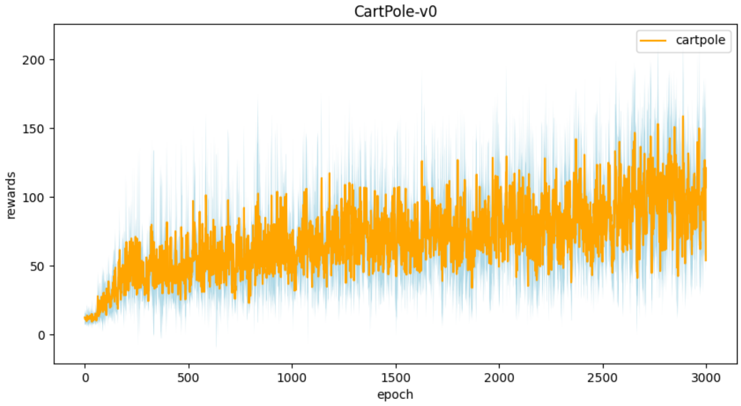
Report HW4

Experiment Results

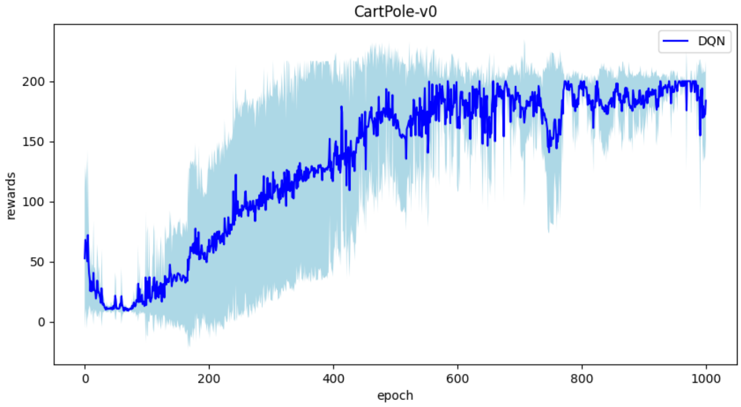
1. taxi.png



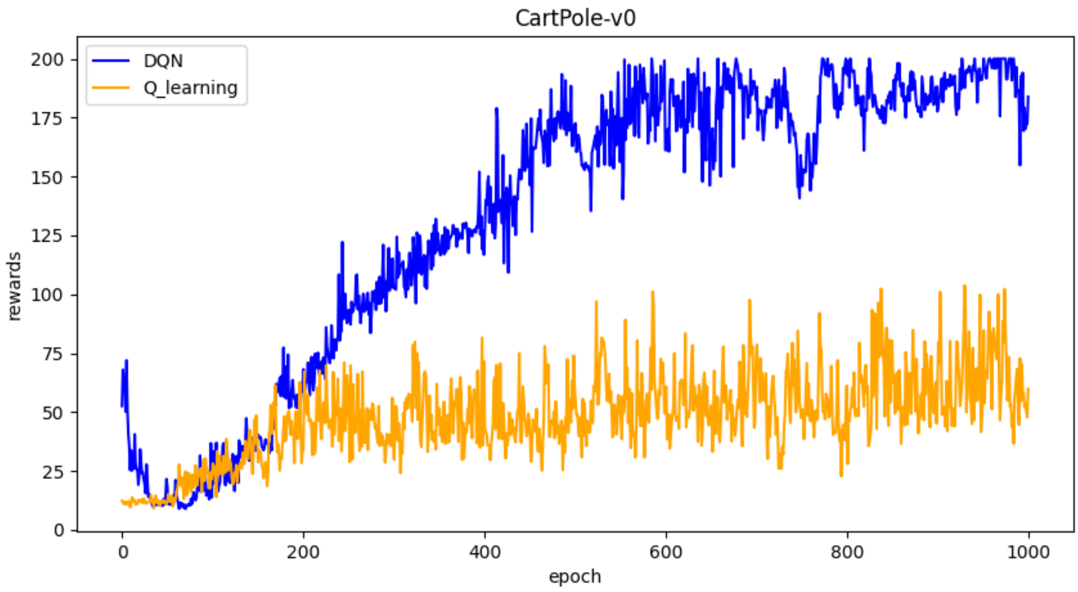
2. cartpole.png



3. DQN.png

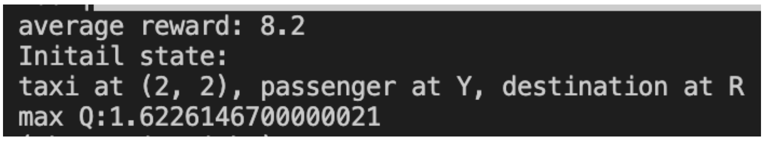


4. compare.png



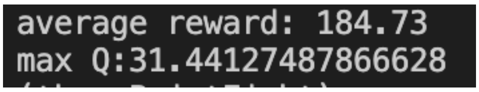
Question Answering

1. Calculate the optimal Q-value of a given state in Taxi-v3 (the state is assigned in google sheet), and compare with the Q-value you learned (Please screenshot the result of the “check_max_Q” function to show the Q-value you learned). (4%)

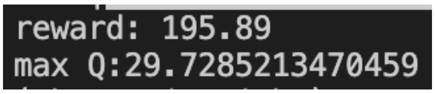


```
average reward: 8.2
Initail state:
taxi at (2, 2), passenger at Y, destination at R
max Q:1.6226146700000021
```

2. Calculate the max Q-value of the initial state in CartPole-v0, and compare with the Q-value you learned. (Please screenshot the result of the “check_max_Q” function to show the Q-value you learned) (4%)



```
average reward: 184.73
max Q:31.44127487866628
```



```
reward: 195.89
max Q:29.7285213470459
```

3. a. Why do we need to discretize the observation in Part 2? (2%)

The observation was continuous but we need discrete numbers to represent the number of buckets.

b. How do you expect the performance will be if we increase “num_bins”? (2%) The training process would be more efficient.

c. Is there any concern if we increase “num_bins”? (2%)

Time might be wasted on useless calculations since we don’t need every variable to be precise.

4. Which model (DQN, discretized Q learning) performs better in Cartpole-v0, and what are the reasons? (3%)

Discretized Q learning performed better. DQN needs a certain amount of data before generating a reasonable model of Q-value.

5. a. What is the purpose of using the epsilon greedy algorithm while choosing an action? (2%)

To balance between exploration and exploitation.

b. What will happen, if we don’t use the epsilon greedy algorithm in the CartPole-v0 environment? (3%)

There might be infinite trials with each action taken infinite times.

c. Is it possible to achieve the same performance without the epsilon greedy algorithm in the CartPole-v0 environment? Why or Why not? (3%)

No, the possibility would be different.

d. Why don’t we need the epsilon greedy algorithm during the testing section? (2%)

We have already done it in the learning section.

6. Why is there “with torch.no_grad():“ in the “choose_action” function in DQN? (3%) To stop the calculation of gradient to speed up the process.

7. a. Is it necessary to have two networks when implementing DQN? (1%)

Yes, using a separate target network, updated every so many steps with a copy of the latest learned parameters, helps keep runaway bias from bootstrapping from dominating the system numerically, causing the estimated Q values to diverge.

b. What are the advantages of having two networks? (3%)

The training process would be more stable, reducing overestimation.

c. What are the disadvantages? (2%)

Might slow down the learning process and increase the sample complexity.

8. a. What is a replay buffer(memory)? Is it necessary to implement a replay buffer? What are the advantages of implementing a replay buffer? (5%)

It is used to store experiences when executing, it is not necessary, but the training process can use a more diverse mini-batch for performing updates.

b. Why do we need batch size? (3%)

Controls the accuracy of the estimated error gradient.

c. Is there any effect if we adjust the size of the replay buffer(memory) or batch size? Please list some advantages and disadvantages. (2%)

Advantages: Larger batches can be processed faster, smaller batches can be better regularized.

Disadvantages: The size of update depends on which samples are drawn from the dataset while this depends on the batch size.

9. a. What is the condition that you save your neural network? (1%)

Save when the rec (0.9*count + 0.1*rec) is bigger than the maximum count.

b. What are the reasons? (2%)

I used the way I save cart pole but I found the results are not ideal, and my classmate suggested to calculate it with the way we calculate q-values.

10. What have you learned in the homework?

By implementing this homework, I have gained a deeper understanding on Q-learning and DQN, and again increased the ability to code in python.