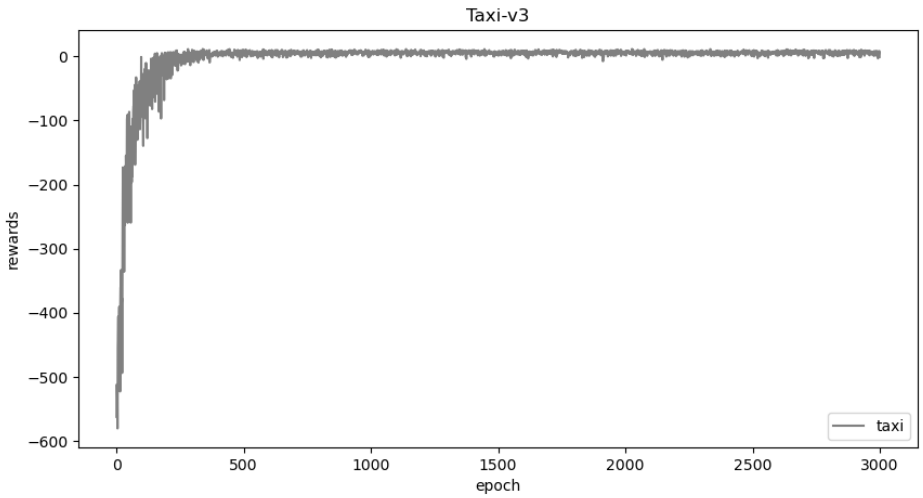
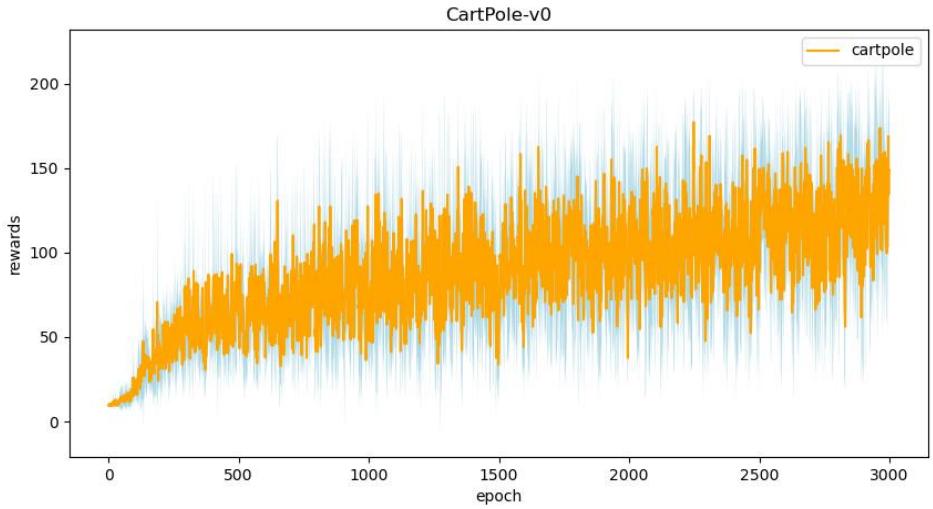
Homework 4:
Reinforcement Learning
Report Template

Part I. Experiment Results:

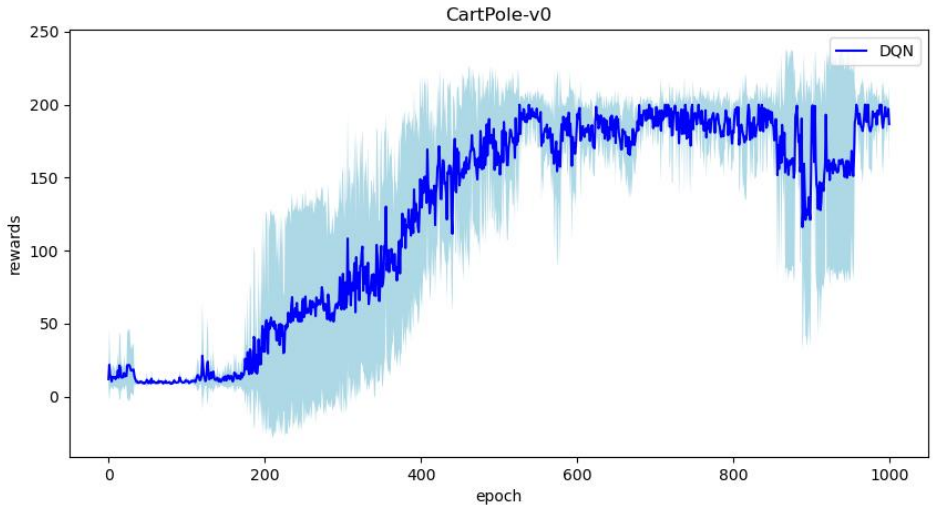
1. taxi.png



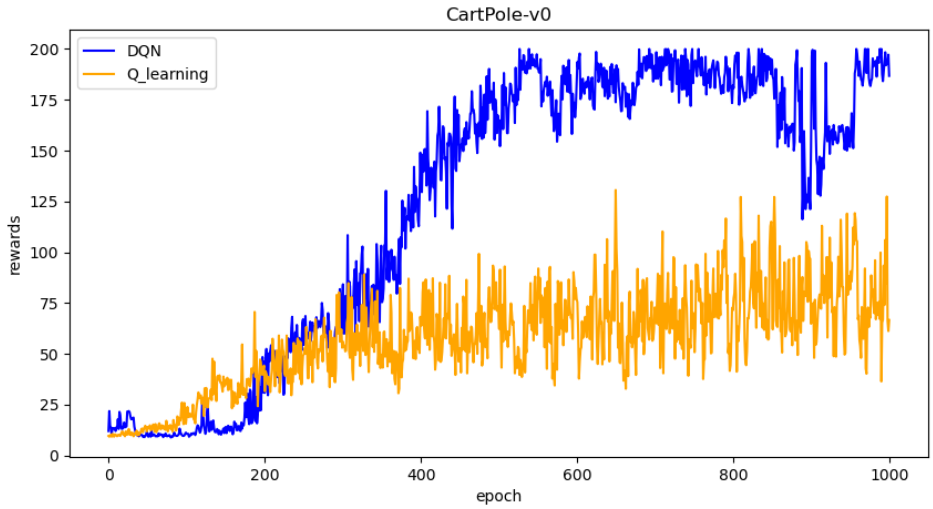
2. cartpole.png



3. DQN.png



4. compare.png



Part II. Question Answering (50%):

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: 8.18
Initail state:
taxi at (2, 2), passenger at Y, destination at R
opt_q = 1.6226146700000017
max Q:1.6226146700000021
```

```
power = np.power(self.gamma, 9)
q = (-1) * (1 - power) / (1 - self.gamma) +  power * 20
```

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: 180.61
opt_q = 33.19472449836912
max Q:30.633980519153994
```

```
power = np.power(self.gamma, 180)
q = (1 - power) / (1 - self.gamma)
```

3.

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

Because the original data is continuous, we need to discretized the observation first to do further computation.

b. How do you expect the performance will be if we increase "num_bins"? **(2%)**

The performance will become better since we have more precise state.

c. Is there any concern if we increase "num_bins"? **(2%)**

The time and space needed will also increase if we increase num_bins.

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

DQN, because DQN uses exact value while discretized Q learning use a range of value.

5.

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

Choose between from the best action and random action.

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

If we don't use the epsilon greedy algorithm, we can only choose either exploitations or explorations. This would lead to some problems: if we only choose the best action, the initial q_table is all zero; on the other hand, if we always choose random action, we cannot ensure the action we chose is a better action, thus the computation may be useless.

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, as I mentioned above, using only exploitations is meaningless, the process won't learn anything; as for using only explorations, it would lead to a bad q_table.

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

Because we only use the best action according to the q_table, there is no need to perform the epsilon greedy algorithm.

6. Why is there "with torch.no_grad():" in the "choose_action" function in DQN? **(3%)**

Because we won't use the autograd engine, so we use with torch.no_grad() to deactivate it. This way, we can reduce memory usage and speed up computations.

7.

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

Yes.

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

Can makes training more stable by prevent short-term oscillation.

c. What are the disadvantages? **(2%)**

Need more memory space.

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%)**

A replay buffer is used to realize experience replay; no; more efficient use of previous experience and better convergence behavior when training a function approximator.

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

To limit the number of the stored state.

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%)**

The larger the size is, the more stable the training will be. However, a large size also requires a lot of memory and it might slow down the training process.

9.

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

if self.count mod 100 = 0

b. What are the reasons? **(2%)**

Because the network difference is not significant between every loop, so saving for every 100 times is enough.

10. What have you learned in the homework? **(2%)**

I have learned much more about machine learning. Also, I learned that Q-learning is a useful strategy to perform in ML. Some easier ways we performed before are using Monte Carlo method. Although it is not a bad way, sometimes we can't know every stat or condition. So here comes Q-learning. I think after this homework, I have finally explored part of reinforcement learning, which is totally different, and much deeper than those I have known before.