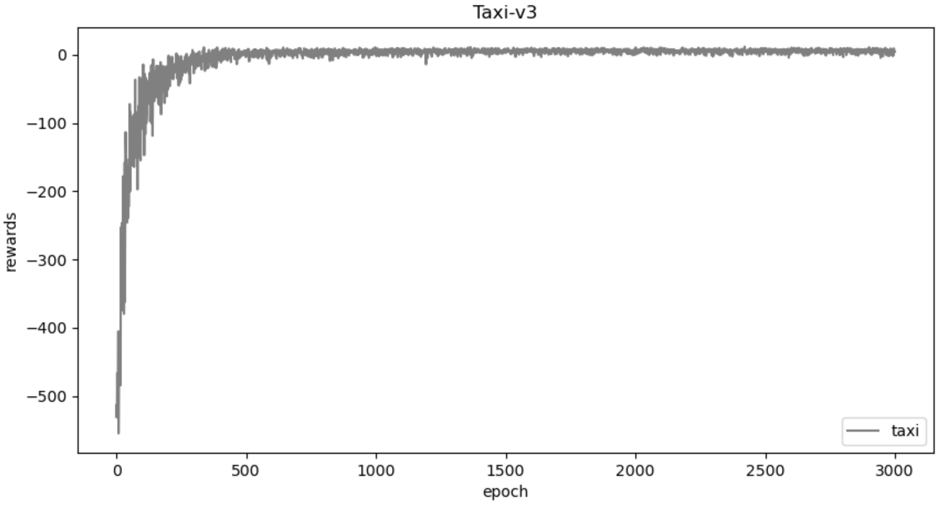
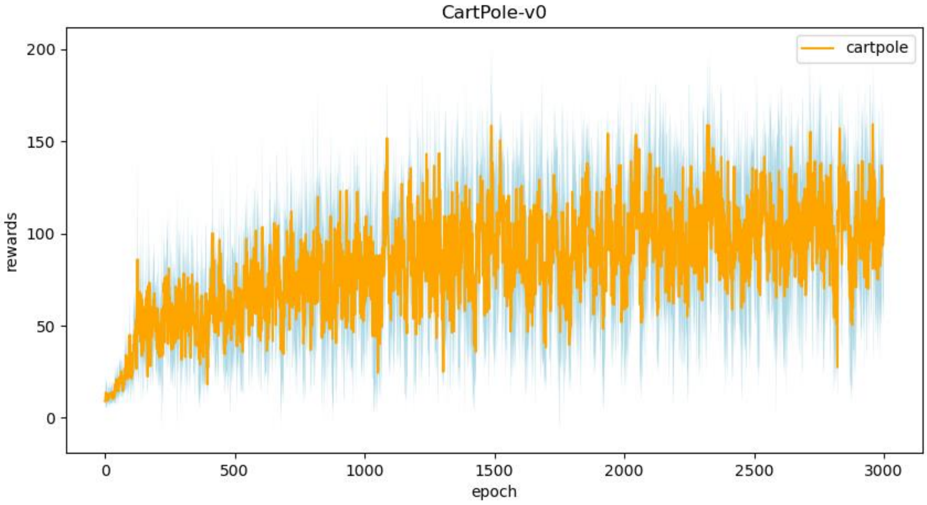
Report

Part 1. Experiment Results

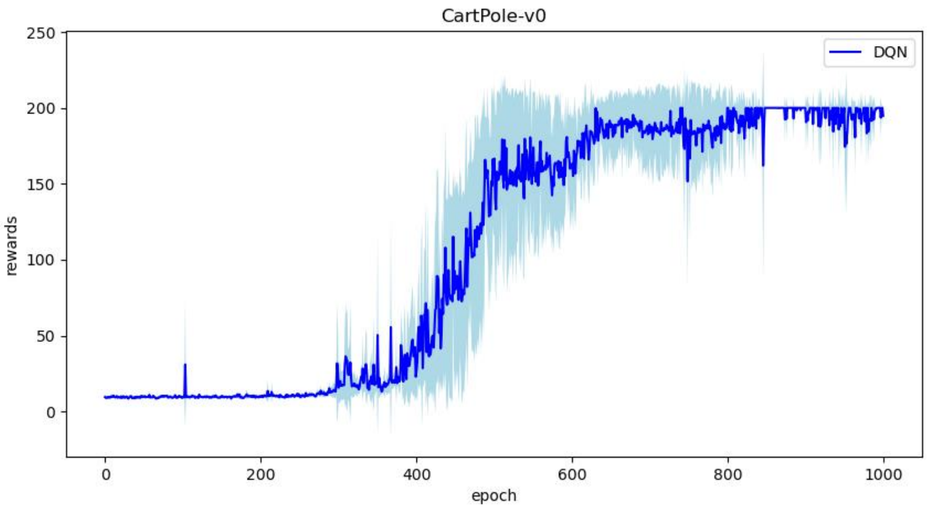
1. Taxi.png



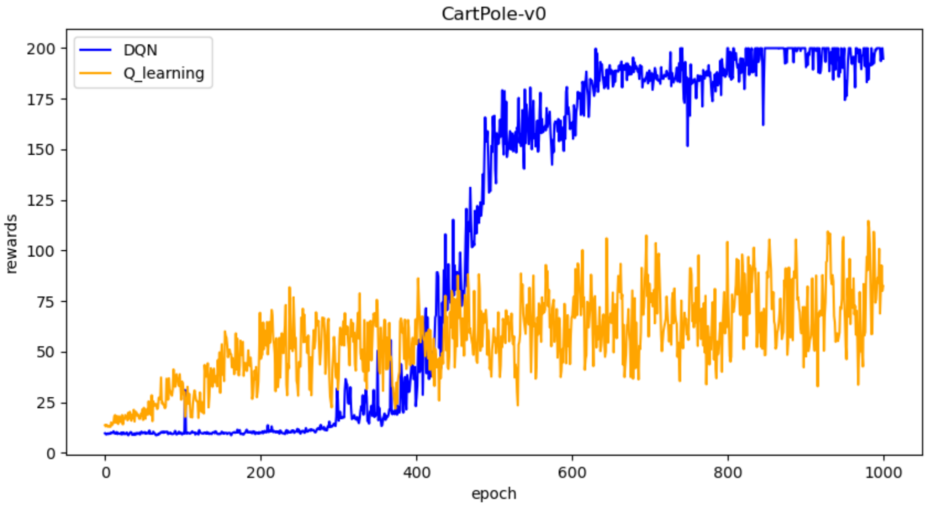
2. Cartpole.png



3. DQN.png



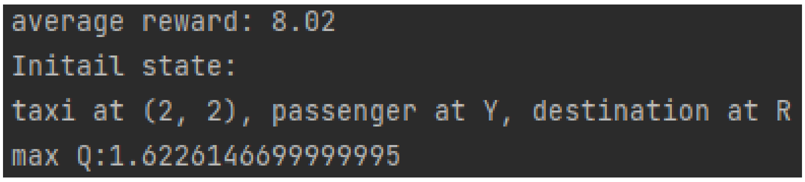
4. Compare.png



Part 2. Question Answering

1. Calculate the optimal Q-value of a given state in Taxi-v3, and compare with the Q-value you learned.

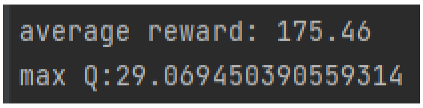
$$Q_{opt} = R_1 + \gamma R_2 + \gamma^2 R_3 + \ldots = (-1) + 0.9(-1) + \ldots + 0.9^9(20) = 1.62261467$$



```
average reward: 8.02
Initail state:
taxi at (2, 2), passenger at Y, destination at R
max Q:1.6226146699999995
```

2. Calculate the max Q-value of the initial state in CartPole-v0, and compare with the Q-value you learned. (max episode lengeth = 200)

$$Q_{max} = R_1 + \gamma R_2 + \gamma^2 R_3 + \ldots = 1 + 0.97 + \ldots + 0.97^{199} = 31.7482497358$$



```
average reward: 175.46
max Q:29.069450390559314
```

3.

a、 Why do we need to discretize the observation in Part 2?

Because the state space is continuous, and we can't store infinitely many values to record the Q-value of every state. So, we need to discretize the observation.

b、 How do you expect the performance will be if we increase "num_bins"?

I think the performance of average reward would be better.

c、 Is there any concern if we increase "num_bins"?

Since it would produce more states, so it'd take more space to store the Q-table.

4. Which model performs better in CartPole-v0, and what are the reasons?

DQN performs better. DQN uses function, instead of Q-table, to approximate Q-values, and DQN doesn't have to discretize the states, so DQN's accuracy can be better than discretized Q-learning.

5.

a、 What is the purpose of using the epsilon greedy algorithm while choosing action?

In order to evaluate other potential and not evaluated actions.

b、 What will happen if we don't use the epsilon greedy algorithm in the CartPole-v0 environment?

If we always choose the best action when learning, we'll always choose the same action for evaluated states and don't have the chance to explore other potential actions.

c、 Is it possible to achieve the same performance without the epsilon greedy algorithm in the CartPole-v0 environment? Why or why not?

I think it is not possible. Because the rewards are always positive, so once we evaluate the state and choose an action, it will always choose the same action.

d、 Why don't we need the epsilon greedy algorithm during the testing section?

Because we have already known the best action of all states after thousands of episodes, so there is no need to use epsilon greedy algorithm.

6. Why is there “with torch.no_grad():” in the “choose_action” function in DQN?

   Because we don’t have to calculate the gradients and we shouldn’t affect the gradients when choosing actions, so no_grad() is required.

7.
   a、 Is it necessary to have two networks when implementing DQN?

      No, it isn’t necessary.

   b、 What are the advantages of having two networks?

      It can increase the stability of the training progress and avoid over-estimating, prevent altering the target value too often.

   c、 What are the disadvantages?

      It needs extra space to store the target network.

8.
   a、 What is a replay buffer? Is it necessary to implement a replay buffer? What are the advantages of implementing a replay buffer?

      i. Replay buffer is a container that stores training experience and perform experience replay.
      ii. No, it is not necessary.
      iii. It can utilize same experience multiple times and make better approximation by sampling the experiences, increasing its training efficiency.

   b、 Why do we need batch size?

      To sample a batch of data and calculate the gradient.

   c、 Is there any effect if we adjust the size of the replay buffer or batch size? Please list some advantages and disadvantages.

	Advantage	Disadvantage
Increase the batch size	1. Estimations are more effective. 2. Approximation more accurately.	1. Harder to calculate the gradient.
Increase the replay buffer size	1. Better represents overall states. 2. Can sample more replay at the same time.	1. Takes more space. 2. Make sample speed slower.

9.

a、 What is the condition that you save your neural network?

I simulate a test function and runs for one episode, if the reward is better than the rewards tested before, then save the network.

b、 What are the reasons?

Because our goal is to maximize the reward, so testing for the network that yields the best reward seems reasonable.

10. What have you learned in the homework?

I've truly understood how Q-learning works and learned how to implement Q-learning in practical, because I didn't know the details of the algorithm by only listening to the lecture. After completing part 1 and 2 in this homework, I can really imagine how do the algorithm work.

I didn't quite understand the theory of DQN as well as PyTorch before working on this homework. During the work on part 3, I've read lots of documents, tutorials, and introduction of PyTorch and DQN. Thanks to this homework and TAs, I have a better understanding on the process of DQN and the usage of PyTorch.