Homework 4

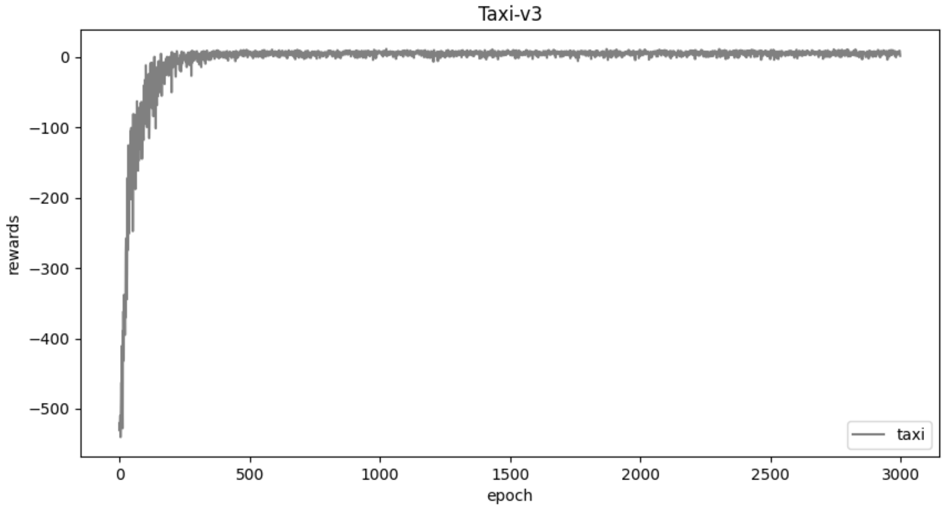
Reinforcement Learning

Please keep the title of each section and delete examples. Note that please keep the questions listed in Part II.

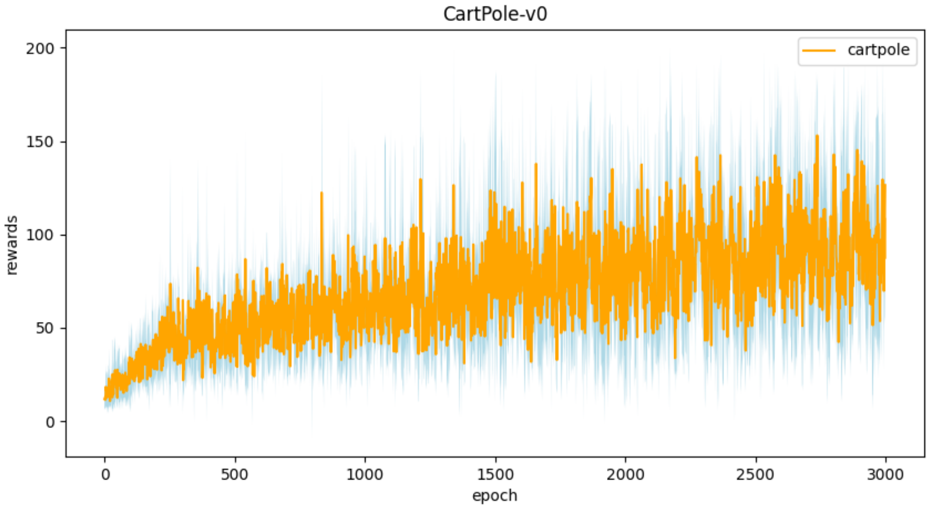
Part I. Experiment Results (the score here is included in your implementation):

Please paste taxi.png, cartpole.png, DQN.png and compare.png here.

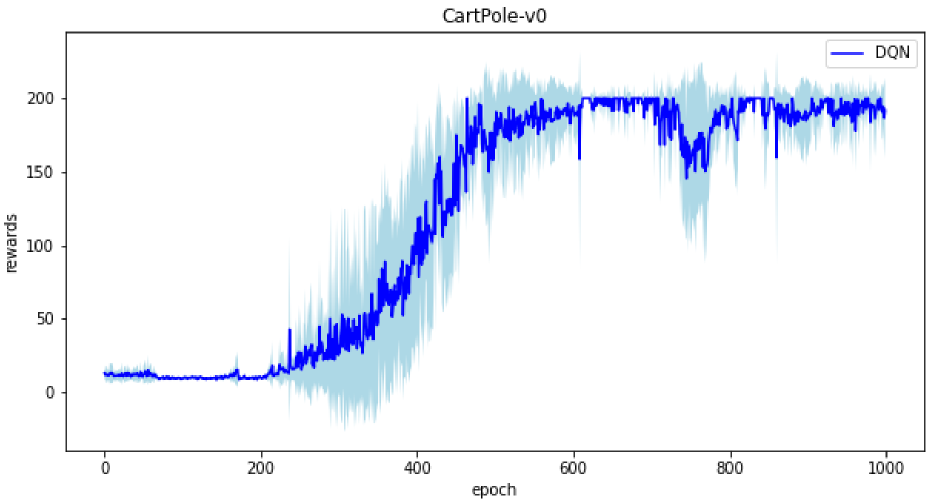
1. taxi.png:



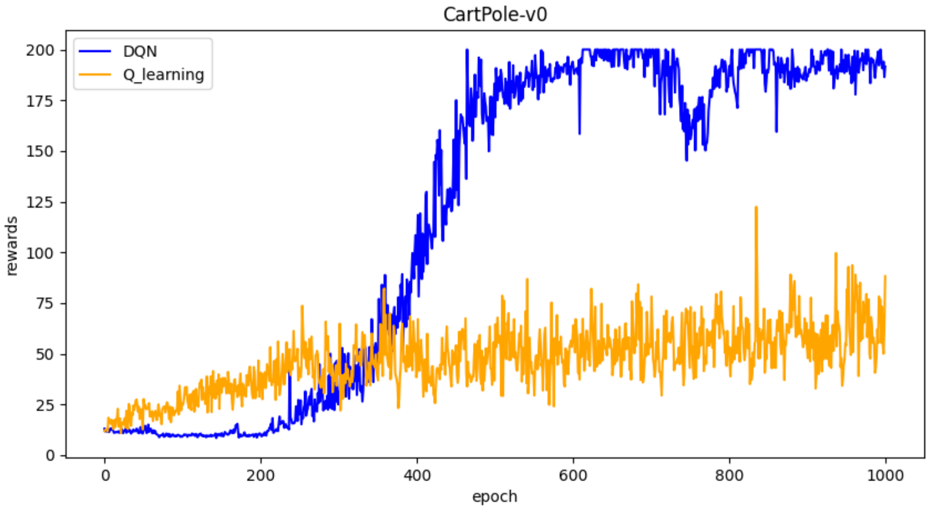
2. cartpole.png



3. DQN.png



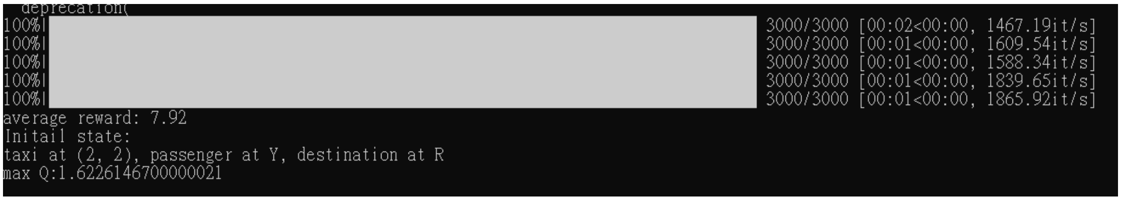
4. compare.png



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%)

$$Q_{otm} = -1 * \sum_{x=0}^{8} 0.9^{x} + 20 * 0.9^{9} = 1.62261467$$



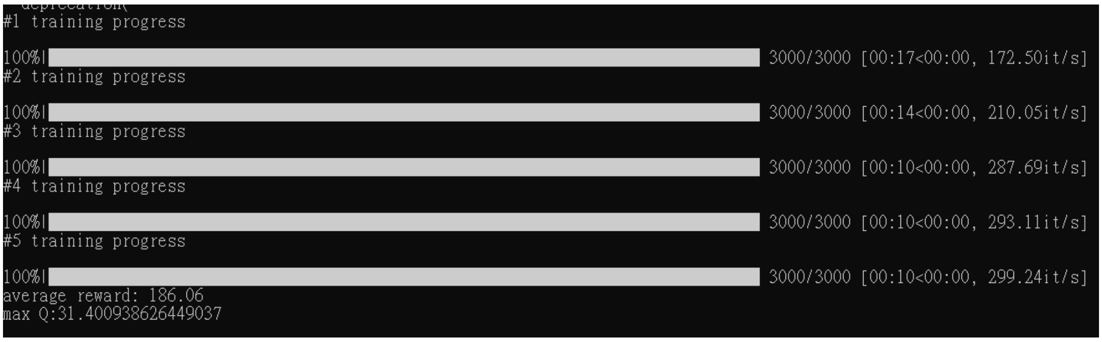
```
  deprecation(
100%|                                        3000/3000 [00:02<00:00, 1467.19it/s]
100%|                                        3000/3000 [00:01<00:00, 1609.54it/s]
100%|                                        3000/3000 [00:01<00:00, 1588.34it/s]
100%|                                        3000/3000 [00:01<00:00, 1839.65it/s]
100%|                                        3000/3000 [00:01<00:00, 1865.92it/s]
average reward: 7.92
Initail state:
taxi at (2, 2), passenger at Y, destination at R
max Q:1.6226146700000021
```

→ $Q_{otm} = Q$

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%)

$max\ Q_{\text{initial state}} = \sum_{x=0}^{200} 0.97^{x} = 33.11444889$



```
  deprecation(
#1 training progress

100%|██████████████████████████████████████████████████| 3000/3000 [00:17<00:00, 172.50it/s]
#2 training progress

100%|██████████████████████████████████████████████████| 3000/3000 [00:14<00:00, 210.05it/s]
#3 training progress

100%|██████████████████████████████████████████████████| 3000/3000 [00:10<00:00, 287.69it/s]
#4 training progress

100%|██████████████████████████████████████████████████| 3000/3000 [00:10<00:00, 293.11it/s]
#5 training progress

100%|██████████████████████████████████████████████████| 3000/3000 [00:10<00:00, 299.24it/s]
average reward: 186.06
max Q:31.400938626449037
```

→ $max\ Q_{\text{initial state}} > Q$

3.

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

State is a continuous value, so there are infinitely many possible state-action pairs, so we need to discretize these values to build a lookup table.

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

Performance will be worse as discrete observations work increases, building the table takes more time.

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

When we increase num_bins, the time to perform discrete observations will increase, and the program needs more time to process, the program may be overloaded.

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

DQN. DQN gives better average reward, use less memory.

5.

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

Selects the action with the highest estimated reward most of the time, the balance between exploration and exploitation, generates the maximum reward possible for the given state.

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

The choice of action will be random, there will be no more exploration and exploitation, the average reward will be random.

c. Is it possible to achieve the same performance without the epsilon greedy

algorithm in the CartPole-v0 environment? Why or Why not? (3%)

It is not possible to achieve the same performance. Because program doesn't have the balance between exploration and exploitation, no matter how many episode times the training is performed, the average reward does not improve.

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

Because we have trained many times in training section. In the learn section, the best value is stored in the qtable, so in the testing section we just need to exploit the already existing value.

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

To perform inference without gradient calculation, to let PyTorch know the model no need parameter update.

7.

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

It is necessary to have two networks.

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

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.

c. What are the disadvantages? (2%)

Use more memory.

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%)

Replay buffer(memory) use to store trajectories of experience when executing a policy in an environment. During training, replay buffer(memory) are queried for a subset of the trajectories.

It is necessary to implement a replay buffer.

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

Because batch size used in the estimate of the 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%)

There is effect if adjust the size.

Advantages:

It requires less memory.

Disadvantages:

The smaller batch size the less accurate the estimate of the gradient will be.

9.

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

When done enough episodes, start checking current best reward with new reward, if new reward is better than current best reward,we save target network and update best reward value.

b. What are the reasons? (2%)

It doesn't take long to execute the program.

Save the best target network, making it possible to find the average reward.

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

Getting to know and using openAI gym environment.

How to install python libraries and how to resolve errors.

Different random commands can affect average reward.

How to use Q-learning and DQN algorithm.