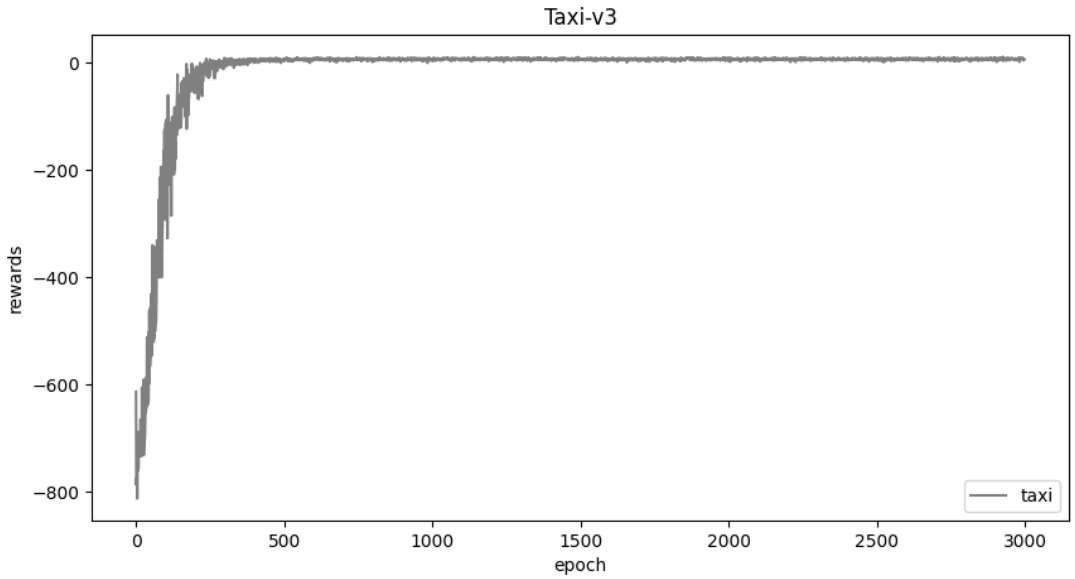
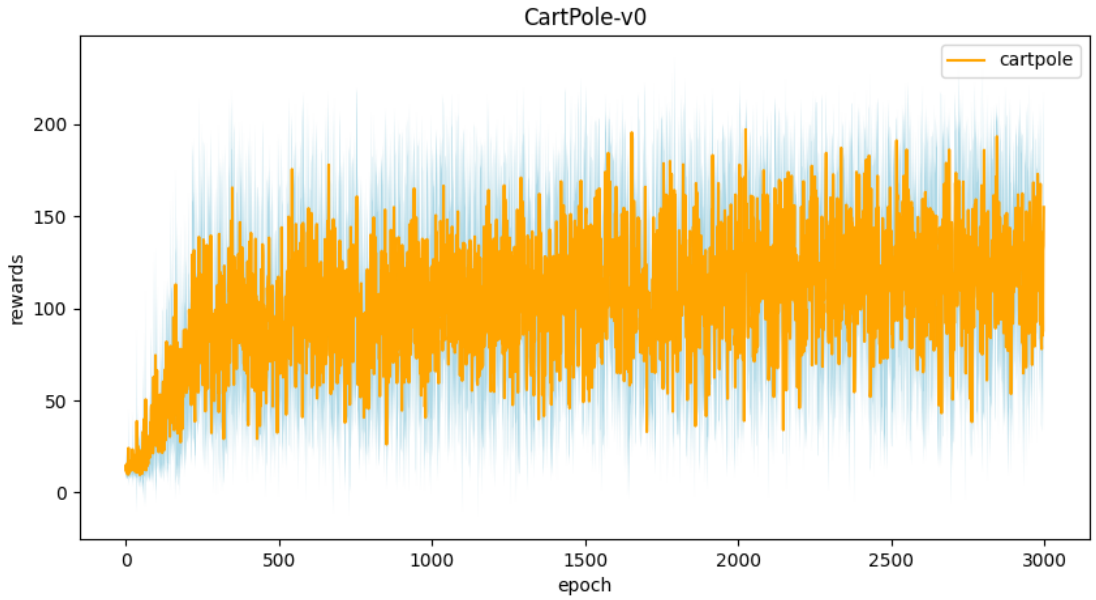
Homework 4: Reinforcement Learning Report

Part I. Experiment Results:

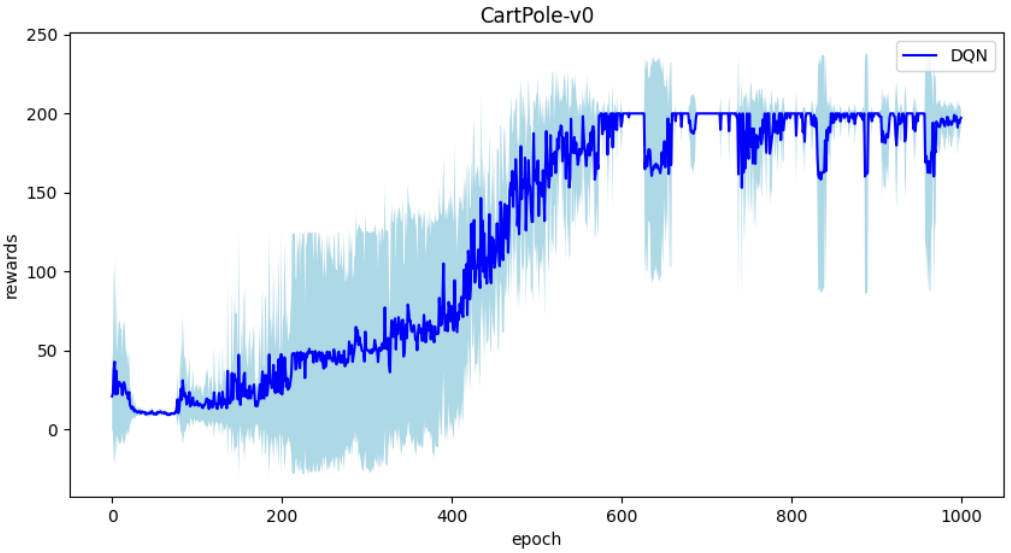
1. taxi.png:



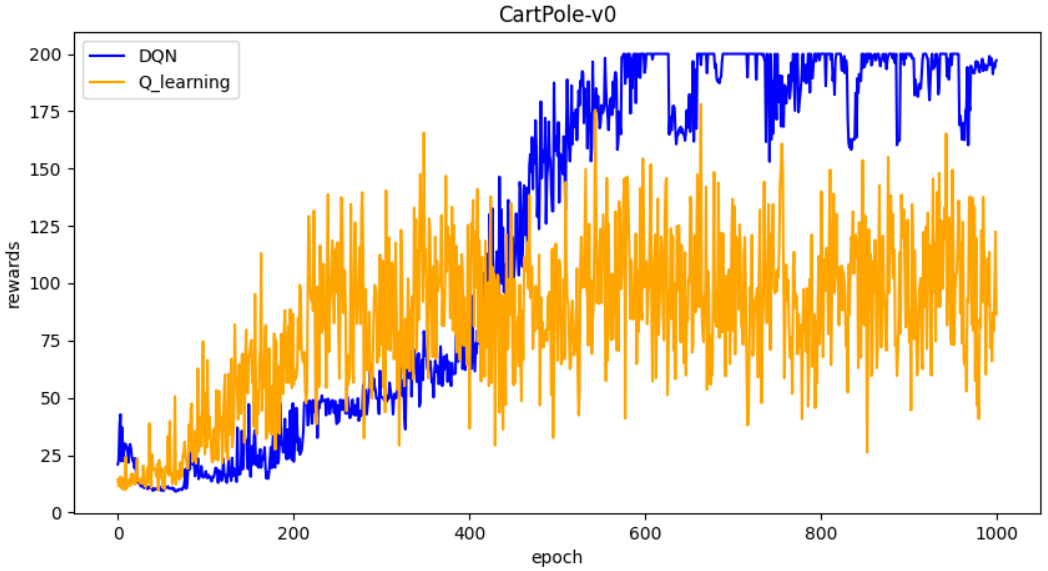
2. cartpole.png:



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: 7.93
Initail state:
taxi at (2, 2), passenger at B, destination at R
max Q:-0.5856821172999982
q_value:-3.958333333333331
```

The above screenshot shows the Q-value that the q-learning estimated. Using the formula provided in the slide, I calculated the optimal q-value for exploited actions, since the random actions are too much to compute each value. The computed result is -3.9583, which is lower than the approximated value.

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: 153.55
max Q:30.452802815369232
```

```
q_value:1.8095238095238093
```

The above screenshot shows both the Q-valued that the q-learning estimated and the one (1.8095) that is originally defined in the course slide (I calculated the optimal value for exploited best actions only due to limitations of the recursion). The originally defined optimal value is much lower than the approximated one.

3.

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

The states in the cartpole environment are continuous values. Directly using these values will result in infinitely many state-action pairs. Thus we have to discretize the observations to make building the look-up table possible.

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

The performance might be better if we use more "num_bins", since doing so makes the discretized states more similar to the original continuous states.

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

Increasing the “num_bins” requires more space for the look-up table, thus the space complexity is increased.

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

At first the discretized Q learning performs slightly better, but after a few episodes DQN performs significantly better than discretized Q learning. The reason behind of this is discretized Q learning’s “brain” is a q-table made by limited states, so its performance is limited to a certain level; while DQN use a deep neural network to approximate q-values, using this method can get the better performance for predicting the q table for very many states.

5.

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

The main purpose of the epsilon greedy algorithm is to balance the rate of exploration and exploitation, selecting the one with higher estimated rewards.

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

The average reward will be decreased significantly, since the agent will not be able to explore for new actions that may lead to better rewards, and only focus on currently better actions.

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, without the epsilon greedy policy, the agent can not explore for new actions that likely leads to better rewards while first training the agent. The exploration is critical especially when initially exploring the environment.

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

During the testing section, the q-table is already optimized in the training process, so the agent can choose the best action using exploitation.

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

We do not need to update the parameters of the evaluation net when we extract the q-values by forwarding the states to it, so use the “torch.no_grad()” to skip the process of calculating the gradient.

7.

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

Not necessary. The older 2013 version of DQN uses only one network to calculate q values.

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

Using the extra target network helps prevent the q-network from becoming unstable, since the prediction q-value and the target one are dependent on weights; also, if one network chooses an over-estimating action, the other one can provide a less over-estimating one for it. That’s why it’s better to store the target values in the seperate network.

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

First of all, the target network does not update very often, so the target q-values are wrong for a long time, which is one of the reasons for a very long training time for DQN; also, using seperate networks requires an extra amount of the 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 is a buffer space to store the correlated transition data. The training data for deep learning have to be independent, but the data for reinforcement learning are correlated. To solve the issue, a replay buffer is implemented to store these correlated data, and use random sampling to make the data uncorrelated. Thus, using the replay buffer has the benefit of speeding up learning and breaking undesired correlations.

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

The batch size determines the number of samples that will be propagated through the network for each iteration. Because the dataset can be huge for deep learning, we need to divide these data into batches of samples that are small enough for the training process.

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%)**

Using the larger replay buffer makes it less likely to sample correlated elements, hence the network is more stable, however doing this requires more memory and is likely to slow down the process of training, also has the downside that the old transitions will be kept in the buffer for a long time; as for using the smaller batch size has the advantage of faster training and less memory requirement, but at the cost of less accurate estimate of gradient.

9.

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

Once an optimization step is done, the neural network is saved.

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

Because it can not be sure which iteration will the terminal states appear, so I decide to save the updated network every time the training process is done.

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

I have learned the very basics of deep neural networks, some concepts and examples of reinforcement learning, including the main idea of related algorithms and how to implement them. I also learned how to make use of some important python libraries such as pytorch and numpy. Last but not least, the patience and techniques required for debugging are also learnt.