Homework 4

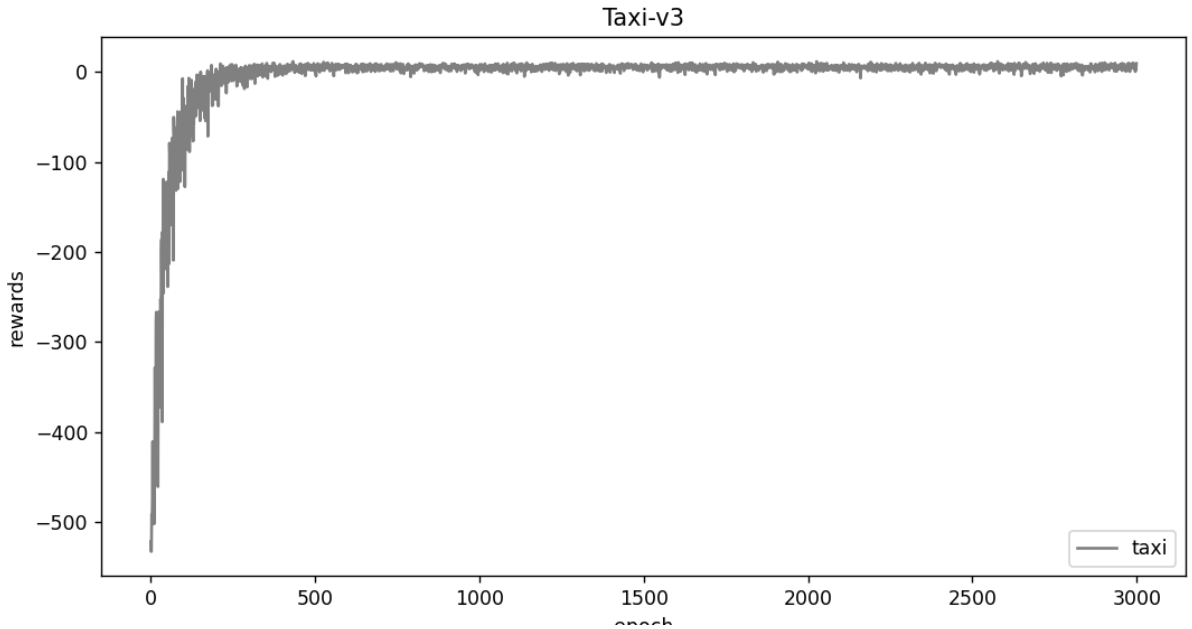
109550134

Reinforcement Learning

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

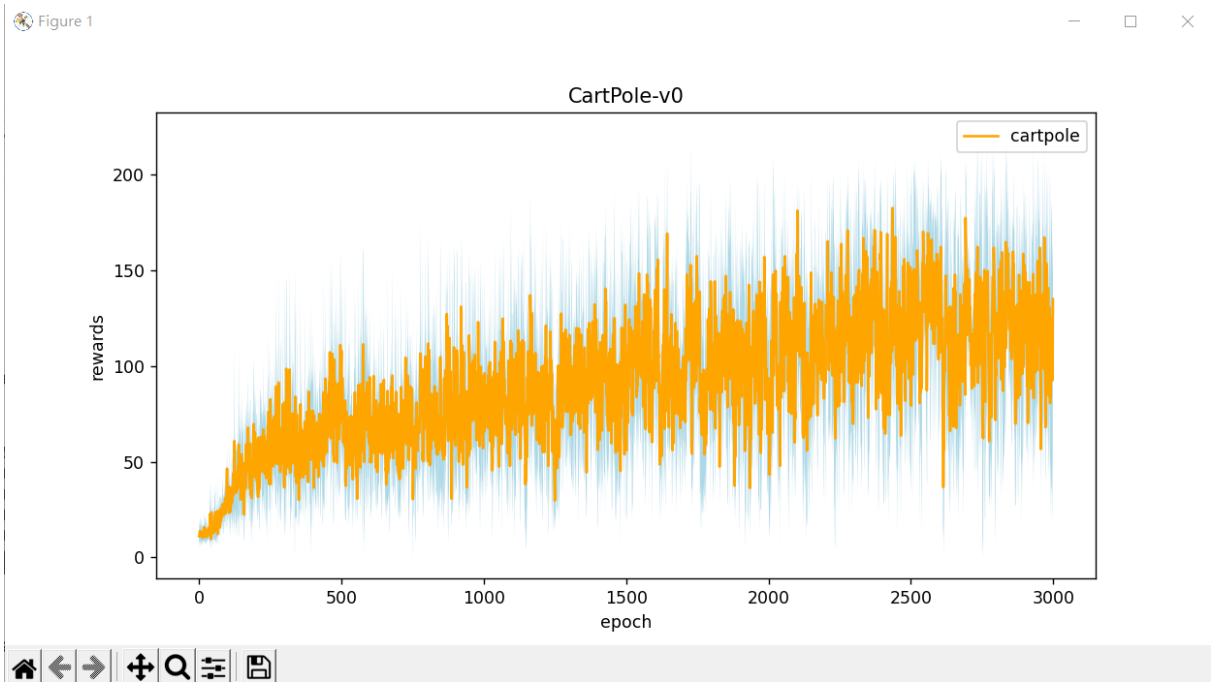
1. Taxi.png:

```
PS D:\AI_2022_spring\AI_HW4> python taxi.py
100%|| 3000/3000 [00:03<00:00, 843.07it/s]
100%|| 3000/3000 [00:03<00:00, 864.06it/s]
100%|| 3000/3000 [00:03<00:00, 868.64it/s]
100%|| 3000/3000 [00:03<00:00, 848.41it/s]
100%|| 3000/3000 [00:03<00:00, 877.78it/s]
average reward: 7.35
Initail state:
taxi at (2, 2), passenger at Y, destination at R
max Q:1.622614669999995
```



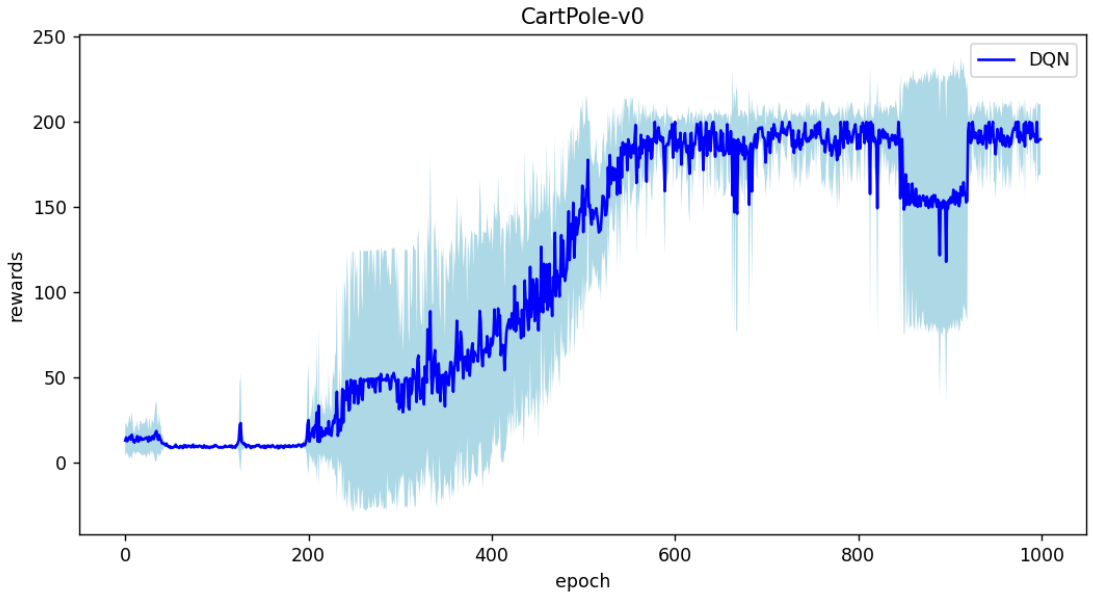
2. cartpole.png:

```
PS D:\AI_2022_spring\AI_HW4> python cartpole.py
100%|| 3000/3000 [00:24<00:00, 122.42it/s]
#2 training progress
100%|| 3000/3000 [00:20<00:00, 146.29it/s]
#3 training progress
100%|| 3000/3000 [00:21<00:00, 142.85it/s]
#4 training progress
100%|| 3000/3000 [00:20<00:00, 146.08it/s]
#5 training progress
100%|| 3000/3000 [00:19<00:00, 155.75it/s]
average reward: 189.57
max Q:30.197457438871318
```

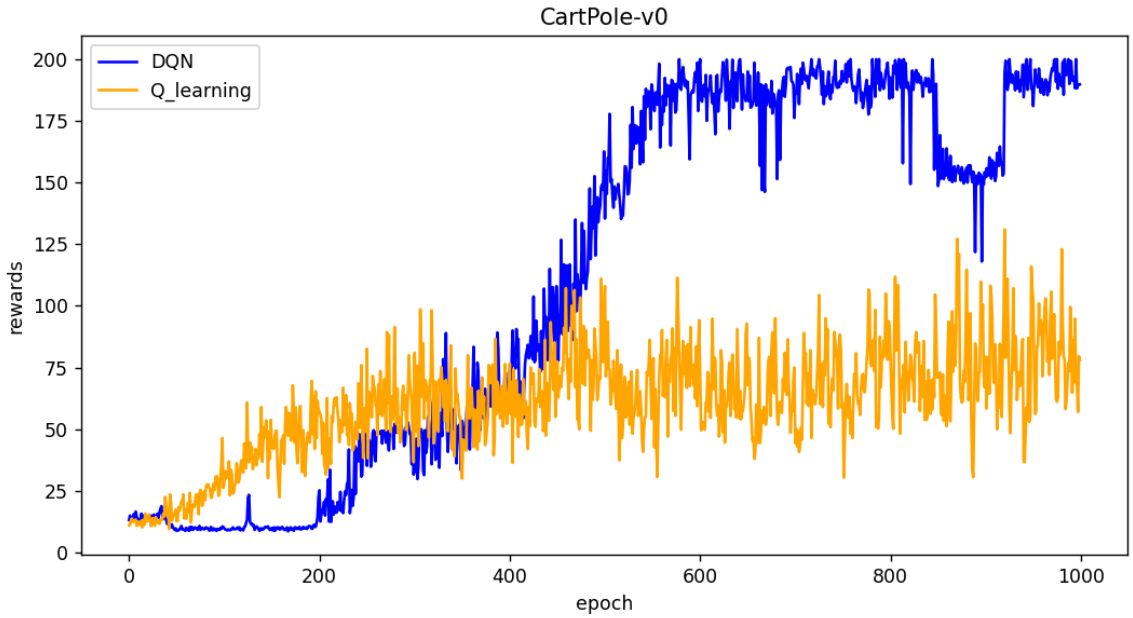


3. DQN.png

```
PS D:\AI_2022_spring\AI_HW4> python DQN.py
100%|| 1000/1000 [06:31<00:00,  2.56it/s]
#2 training progress
100%|| 1000/1000 [05:36<00:00,  2.97it/s]
#3 training progress
100%|| 1000/1000 [08:59<00:00,  1.85it/s]
#4 training progress
100%|| 1000/1000 [06:36<00:00,  2.52it/s]
#5 training progress
100%|| 1000/1000 [05:25<00:00,  3.08it/s]
reward: 199.45
max Q:0.12019764631986618
```



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%)

```
def check_max_Q(self, state):
    """
    - Implement the function calculating the max Q value of given state.
    - Check the max Q value of initial state

    Parameter:
        state: the state to be check.
    Return:
        max_q: the max Q value of given state
    """
    # Begin your code
    #pass
    print("opt Q:" , (-1)*(1-np.power(self.gamma,9))/(1-self.gamma)+20*np.power(self.gamma,9))
    return np.max(self.qtable[state])
    # End your code
```

```
100%|
100%|
100%|
100%|
100%|
average reward: 7.35
Initail state:
taxi at (2, 2), passenger at Y, destination at R
opt Q: 1.6226146700000017
max Q:1.6226146699999995
```

left -> left -> down -> down -> pick -> up -> up -> up -> up -> drop
9 step with reward (-1), 1 step with reward (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%)

```python
def check_max_Q(self):
    """
    - Implement the function calculating the max Q value of initial state(self.env.reset()).
    - Check the max Q value of initial state

    Parameter:
        self: the agent itself.
        (Don't pass additional parameters to the function.)
        (All you need have been initialized in the constructor.)

    Return:
        max_q: the max Q value of initial state(self.env.reset())
    """
    # Begin your code
    #pass
    state = self.discretize_observation(self.env.reset())
    print("opt Q:" ,(1-np.power(self.gamma,189.57))/(1-self.gamma))
    return np.max(self.qtable[state[0]][state[1]][state[2]][state[3]])
    # End your code
```

```
PS D:\AI_2022_spring\AI_HW4> python cartpole.py
#1 training progress
100%|
#2 training progress
100%|
#3 training progress
100%|
#4 training progress
100%|
#5 training progress
100%|
average reward: 189.57
opt Q: 33.22977229166539
max Q:30.197457438871318
```

Calculate the max q value with average reward 189.57

3.

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

Because the observed data is continuous, thus we have to discretize it to get num_bin states depending on which interval its value is in.

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

The performance may become better if we increase “num_bins”, this means the bounded intervals were separated into more states, and the discretization is closer to the observation.

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

The process may become slower and it may cost more to save the Q-table, because the number of states will increase, thus Q-table has to contain much more data and requires more space and operation.

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

DQN performs better than Q-learning. I think that it's because the Q learning has to convert the observed data into discrete states, while DQN can use the continuous data without discretization. Which means DQN can preserve more details in the data, therefore increase performance.

5.

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

Balance exploit and explore : when exploit, remain the known best(max) action of the state, and when explore, choose a random action in order to get more information. Thus when remaining known best actions, also try new actions wishing to get better reward.

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

If all explore : can't remain the known best situation, all random

If all exploit : can only work on the known information, may miss unknown high-performance conditions without randomness.

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 may be possible that if we can find another method to replace the learning rate in epsilon greedy algorithm, and remain the same proportion of explore/exploit rate. With the same distribution, I think there exist possibility to achieve the same performance.

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

Because we use epsilon greedy algorithm to explore and exploit while training, and at the test section we only have to find the best path, and get the best reward , therefore there's no need to train the data with epsilon greedy algorithm.

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

In the "choose_action" function, every tensor has requires_grad set to False, which means gradient calculation will be disabled.

7.

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

No, DQN with only one net can also work.

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

In DQN with one net, max Q values will be chosen, however there exist errors of the estimated value, and the processes will cause an overestimated reward. Double DQN separates target and behavior networks, which becomes more stable, and less overestimated.

c. What are the disadvantages? (2%)

May need to spend more space when using two networks of the same size.

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%)

(1.) Replay buffer contains experience tuples : (state, action, reward, next_state), and training data is selected from the tuples instead of using only the latest data.

(2.) Not necessary needed to implement a replay buffer

(3.) Using this “experience replay” implemention, we can learn more from a tuple multiple times, which is more efficiently. And because the data is randomly selected, it is more similar to independent and identically distributed data.

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

When the dataset is too large that it’s hard to use all data every time, we need batch size to decide how much data will be used in every turn.

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%)

When the size is larger, advantages are : less iterations are needed to run all data, and it converges more accurately; disadvantages are : more space needed, and more time to finish each iteration.

When the size is smaller, advantages are : less time to finish each iteration, less space consumed ; disadvantages are : the convergence oscillates more, need more iteration to converge.

9.

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

If done = true, then save the network

b. What are the reasons? (2%)

Only save the networks that are successfully done, because I think those networks may have the higher probability to contain the positive features that will affect the results.

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

It’s my first time to use the environment in openAI gym, and I learned that besides writing code by myself, it’s also important to read and analyze the given code so that I can work based on the environment and get information from it.