Spring 2022

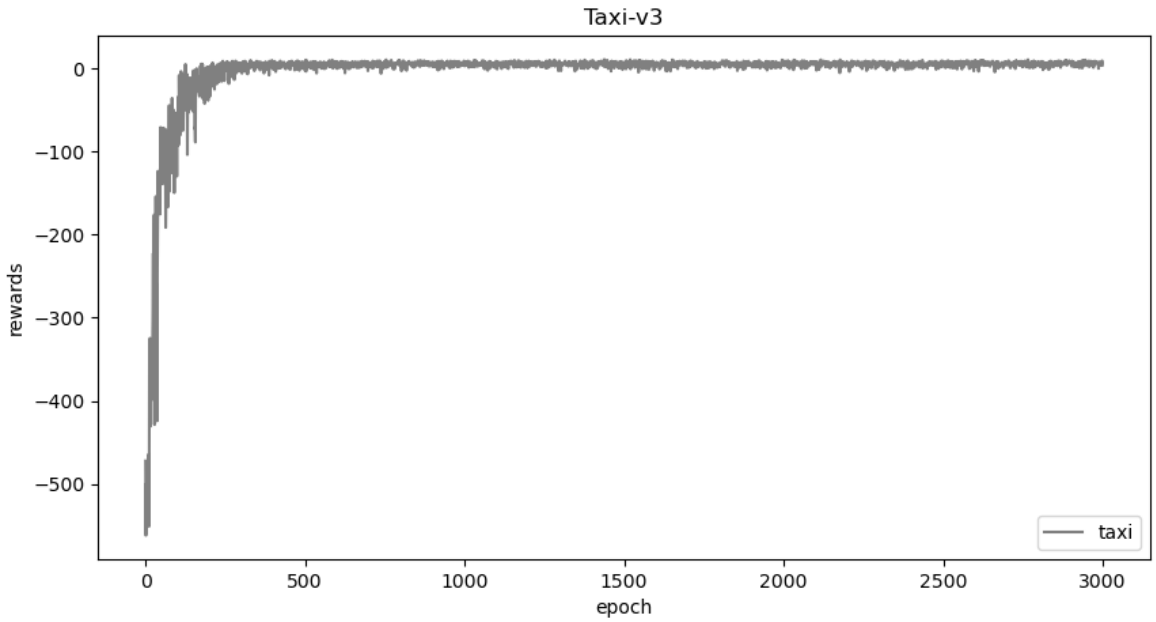
Introduction to Artificial Intelligence

Homework 4: Reinforcement Learning

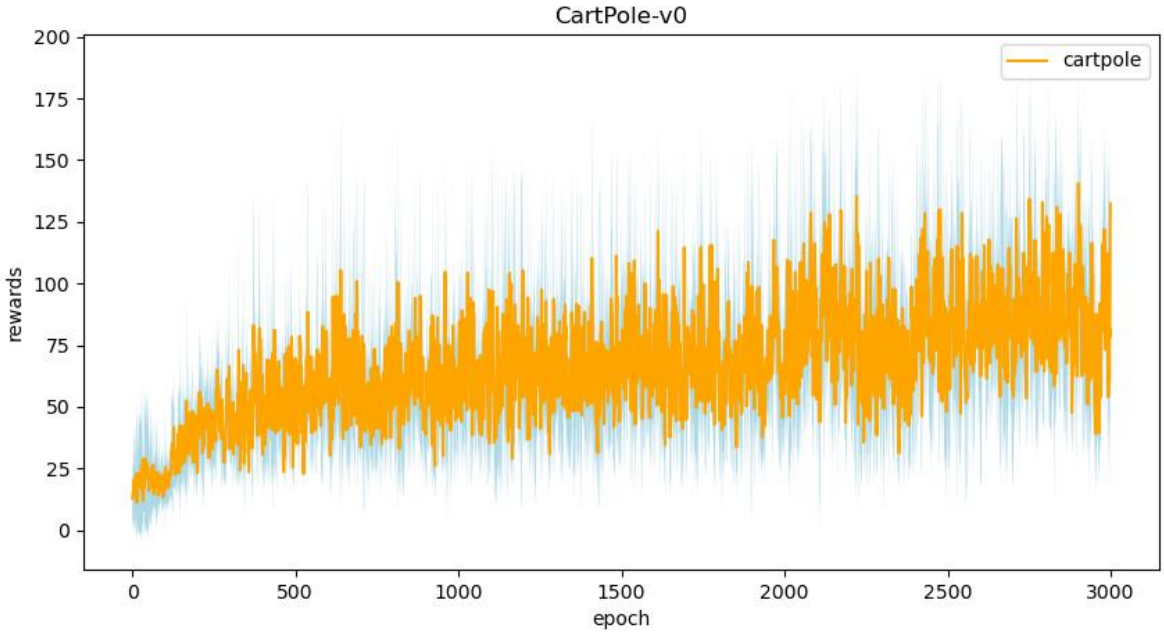
109550159 李驊恩

Part I. Experiment Results

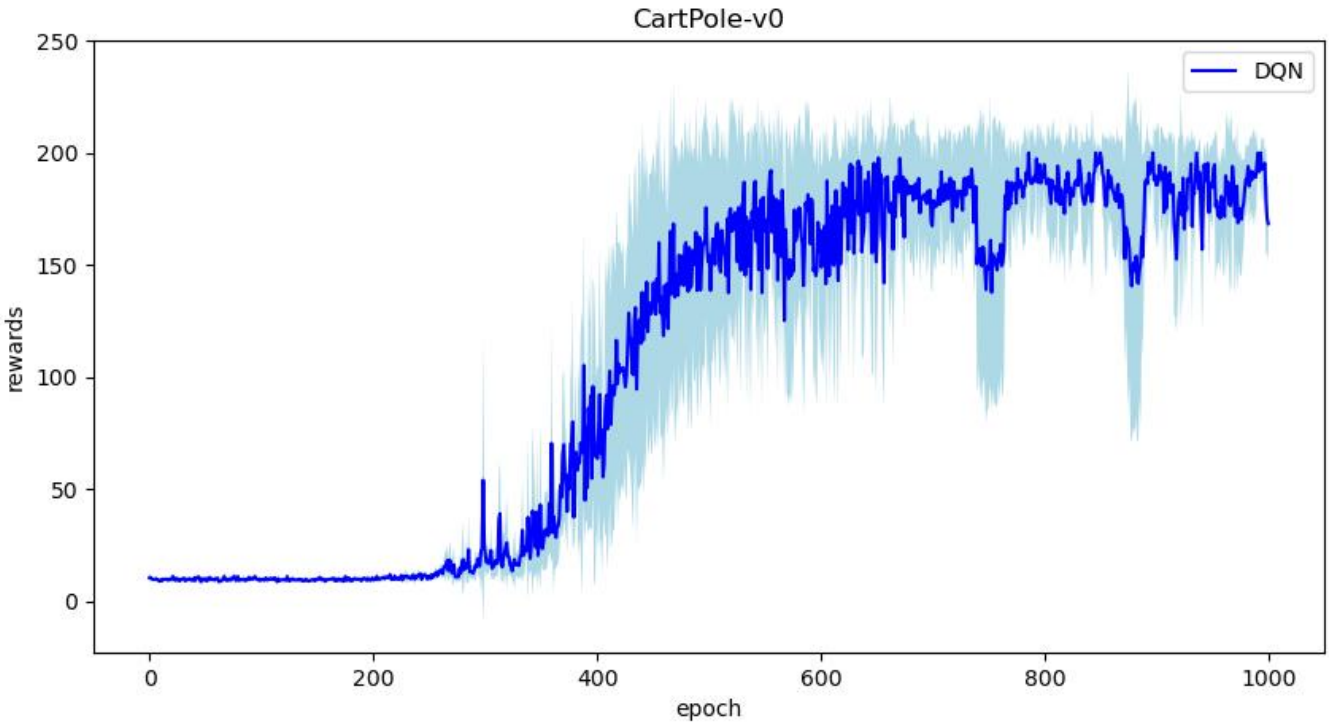
Taxi:



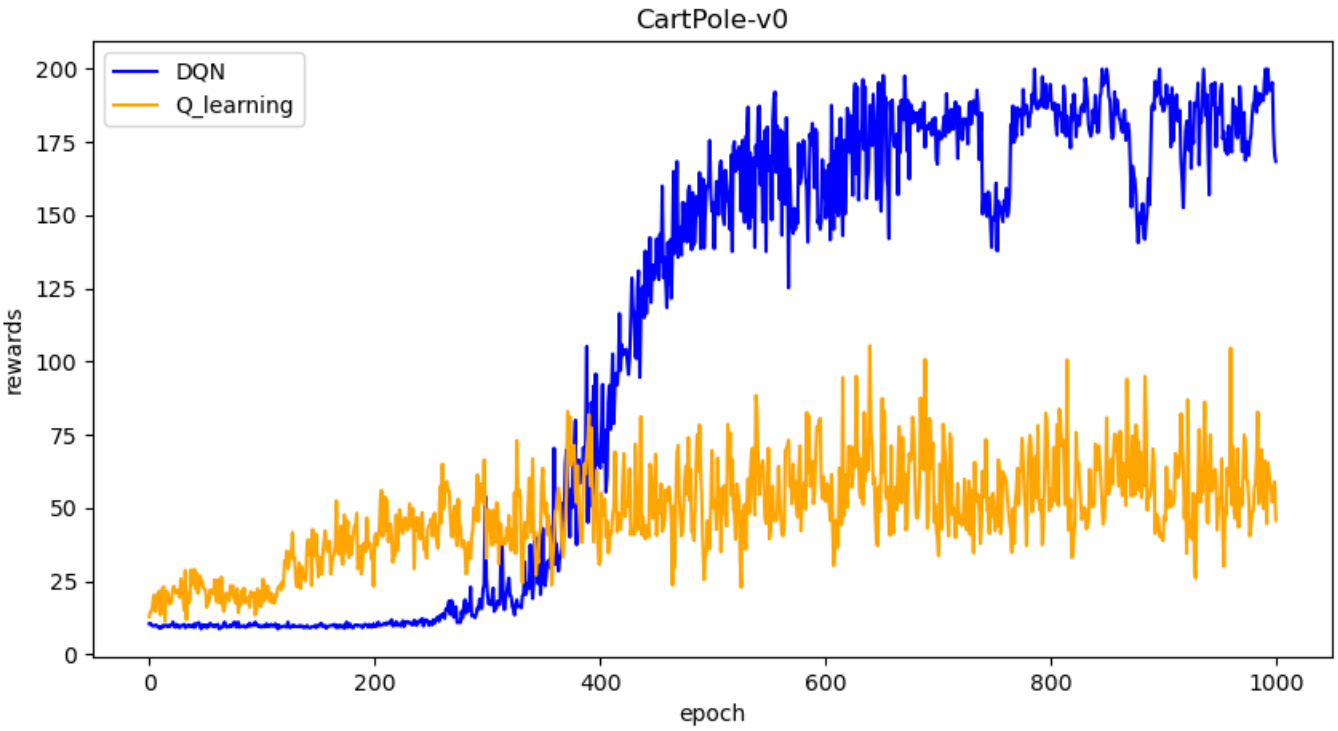
Cartpole:



DQN:


CartPole-v0
DQN
rewards
epoch
250
200
150
100
50
0
0
200
400
600
800
1000

Compare:


CartPole-v0
DQN
Q_learning
rewards
epoch
200
175
150
125
100
75
50
25
0
0
200
400
600
800
1000

Part II. Question Answering

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%)

```
100%|██████████| 3000/3000 [00:39<00:00, 76.84it/s]
100%|██████████| 3000/3000 [00:35<00:00, 83.62it/s]
100%|██████████| 3000/3000 [00:33<00:00, 89.09it/s]
100%|██████████| 3000/3000 [00:33<00:00, 88.96it/s]
100%|██████████| 3000/3000 [00:33<00:00, 89.73it/s]
average reward: 7.62
Initial state:
taxi at (2, 2), passenger at Y, destination at R
max Q:1.6226146699999995
```

The computation of Q-value is based on the formula:

$$\hat{Q}_{opt}(s,a) \leftarrow (1-\eta)\hat{Q}_{opt}(s,a) + \eta\left(r + \gamma \max_{a' \in Actions(s')} \hat{Q}_{opt}(s',a')\right)$$

where η = learning rate = 0.8, γ = gamma = 0.9, reward = 1

$$\Rightarrow \hat{Q}_{opt}(s,a) \leftarrow 0.2\,\hat{Q}_{opt}(s,a) + 0.8\left(1 + 0.9 \max_{a' \in Actions(s')} \hat{Q}_{opt}(s',a')\right)$$

For state = 248, the initial state:

taxi at (2,2), passenger at Y, destination at R

After the computation of computer we get:

max Q-value ≈ 1.62 which is the same with the screen shot.

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%)

```
In [67]: runfile('C:/Users/user/Desktop/AI_HW4/
cartpole.py', wdir='C:/Users/user/Desktop/AI_HW4')
#1 training progress
100%|██████████| 3000/3000 [01:15<00:00, 39.79it/s]
#2 training progress
100%|██████████| 3000/3000 [01:25<00:00, 35.08it/s]
#3 training progress
100%|██████████| 3000/3000 [01:35<00:00, 31.49it/s]
#4 training progress
100%|██████████| 3000/3000 [01:31<00:00, 32.73it/s]
#5 training progress
100%|██████████| 3000/3000 [01:24<00:00, 35.40it/s]
average reward: 199.27
max Q:30.57291651214969
```

The computation of Q-value is based on the formula:

$$\hat{Q}_{opt}(s,a) \leftarrow (1-\eta)\hat{Q}_{opt}(s,a) + \eta\left(r + \gamma \max_{a' \in Actions(s')} \hat{Q}_{opt}(s',a')\right)$$

where η = learning rate = 0.8, γ = gamma = 0.9, reward = 1

$$\Rightarrow \hat{Q}_{opt}(s,a) \leftarrow 0.2\,\hat{Q}_{opt}(s,a) + 0.8\left(1 + 0.9 \max_{a' \in Actions(s')} \hat{Q}_{opt}(s',a')\right)$$

After the computation of computer we get:

max Q-value ≈ 30.6 which is the same with the screen shot.

3.

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

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

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

a.

Parameters of cartpole (position, velocity, angle, angular velocity) are continuous. If the states are continuous value, there will be infinite possible state-action pair. Thus, we need to discretize the observation to build a table.

b.

If we increase ”num_bins”, we can discretize our observation more precisely, and thus we can expect that there will be a better learning process.

c.

If we increase ”num_bins”, there will be more optimal policies need to be found, which leads to more training time.

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

According to the graph we have plotted, DQN performs better in Cartpole-v0.
We use Q-table in Q-learning, and deep neural network in DQN. When using DQN, we don’t need to discretize the observation since DQN is available with continuous state spaces.

5.

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

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

c. Is it possible to achieve the same performance without the epsilon greedy algorithm in the CartPole-v0 environment? Why or Why not? (3%)

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

a.

The purpose of using the epsilon greedy algorithm is to balance exploration and exploitation. Agent can learn new experience if it has a chance to choose a random action.

b.

if we don’t use the epsilon greedy algorithm, the agent will always select the best known action, but will never explore a better one.

c.
No, we won't get such performance without the epsilon greedy algorithm.
There is only two action to choose (left or right) in cartpole. Different picking of actions may leads to a totally different results.
d.
We have saved the best Q-table in the learning section with epsilon greedy algorithm. In the test section, we just have to choose action according to the best Q-table.

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

If we use "with torch.no_grad():" , gradients in the "choose_action" function will not be traced. The "choose_action" function here only choose an action according to the current state. We don't need to train the network after it, and we don't need to backward here. Thus, using "with torch.no_grad():" is a faster way.

7.

a. Is it necessary to have two networks when implementing DQN? (1%)
b. What are the advantages of having two networks? (3%)
c. What are the disadvantages? (2%)

a. Yes, it is necessary to have two networks when implementing DQN.
b. Having two networks helps to converge the Q-value, and it makes the algorithm be stable.
c. There might be some slight increase on the computational time and also extra 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%)
b. Why do we need batch size? (3%)
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%)

a. Replay buffer stores all of the experience during the process that the agent choosing action, these choice may come from different strategies. It is necessary to implement a replay buffer. The advantage of implementing a replay buffer is that there will be more choice for mini-batch when training. It also replays the experience that comes from different strategy which just been stored.

b. Batch size is needed because the data is very large in the hole process, we have to split it into small part, so we can set a batch size and choose data in the replay buffer randomly.
c. If we adjust the size of the replay buffer(memory) or batch size, the stability of the training algorithm might change. If we enlarge the size, it will be more stable, but the time cost of training will increase and so will the memory cost.

9.

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

b. What are the reasons? (2%)

a. I save my neural network after the data of experiences stored in the replay buffer are large enough.

b. We need to store large enough of data to save our neural network because large data size contains large variety of strategy to choose action. Thus our neural network will be stable during the computation.

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

I have learned many ways to implement reinforce learning, such as Q-learning and DQN. I also learn many programming syntax of using Pytorch, which is very useful for neural network.