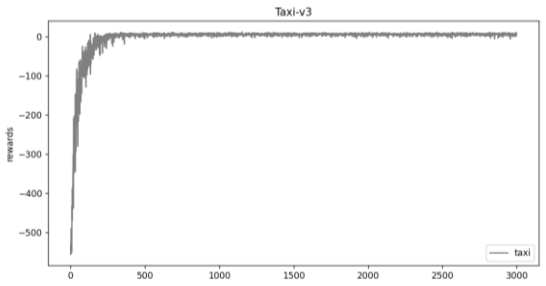
Homework 4 Report 109550171 陳存佩

Part I. Experiment Results

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

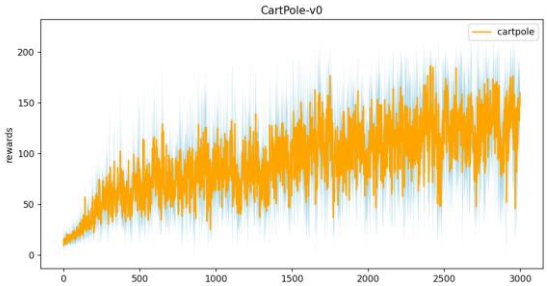
taxi

```
PS D:\Pei\.AI intro\HW4\AI_HW4_updated> python taxi.py
100%|
100%|
100%|
100%|
100%|
average reward: 8.07
Initail state:
taxi at (2, 2), passenger at B, destination at G
max Q:1.6226146699999995
```



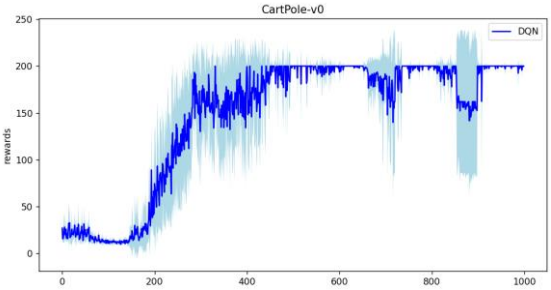
cartpole

```
PS D:\Pei\.AI intro\HW4\AI_HW4_updated> 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: 189.56
max Q:30.001170989087267
```

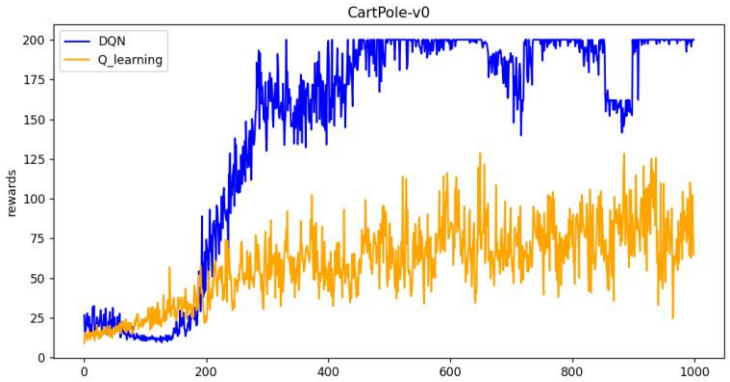


DQN

```
PS D:\Pei\.AI intro\HW4\AI_HW4_updated> python DQN.py
#1 training progress
100%|
reward: 146.33
#2 training progress
100%|
reward: 164.342
#3 training progress
100%|
reward: 153.363
#4 training progress
100%|
reward: 122.621
#5 training progress
100%|
reward: 143.07
reward: 200.0
max Q:tensor([32.7384, 32.8014], grad_fn=<UnbindBackward0>)
```



compare



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%)**

I calculated the optimal Q by coding.

The Optimal Q and Q value we learned are almost the same because the learning curve is steep in early episodes.

```
power = np.power(self.gamma, 9)
optimal_Q = (-1)*(1 - power) / (1 - self.gamma) + 20 * power
print("optimal_Q:",optimal_Q)
```

```
average reward: 8.07
Initail state:
taxi at (2, 2), passenger at B, destination at G
optimal_Q: 1.6226146700000017
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. (Please screenshot the result of the "check_max_Q" function to show the Q-value you learned) (4%)**

I calculated the optimal Q by coding.

The Q value we learned is smaller than optimal Q value because the learning curve is not very steep, and it results in bad estimation for early episodes.

```
power = np.power(GAMMA, np.mean(rewards))
print(f"optimal Q:{(1 - power) / (1 - GAMMA)}")
```

```
average reward: 189.56
max Q:30.001170989087267
optimal Q:33.22974074298835
```

3.

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

Because continuous data costs the memory space too much, and it causes too many states to visit during training.

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

I expect increasing "num_bins" will get better performance.

Because the more bins can let the model account for more specific state space.

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

Yes, it might result in costing too much storage space, time, and computational power.

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

DQN performs better. Because neural network can match different environment and automatically extract feature from environment. However, Q table is fixed.

5.

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

Because we have to take into account exploration and exploitation and get balance.

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

We will only consider the things we have already learned, and don't have the chance to explore new actions which might be better than learned action.

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

Yes, since we can balance exploration and exploitation using other algorithm such as the randomized probability matching algorithm. This algorithm samples a probability distribution which represents the probable mean of the payoff.

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

Because we have finished the training. In testing section, we only use the actions which are explored when training to get best reward.

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

It can take less time to finish the training.

When we select the action, we don't have to train the network, because our goal in "choose_action" is picking one action using current weights.

7.

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

No.

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

If we overestimate on one Q network, then we have the second will hopefully control this bias when we would take the max.

It not only improves accuracy in the action-values but also improves the policy learned.

c. What are the disadvantages? (2%)

We have to maintain two networks, which increase space cost and little time cost.

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 can remember the things which have learned.

It is not necessary to implement a replay buffer, but if we have it, we can make the training more robust.

It can make better use of our experience.

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

It's hard for us to send all data to train, so we divide the data into batches.

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%)

Bigger:

More data can store, resulting in more accurate approximation.

However, needs more memory and more time for each step.

Smaller:

Less data can store, resulting in less accurate approximation.

It needs less time for each step.

Decrease the noise in the gradients and get better gradient estimate.

9.

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

Loss value (using loss function).

b. What are the reasons? (2%)

I want the loss value to be as small as possible.

The loss value is smaller, the module is better.

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

I. I browsed lots of websites which let me get familiar with this homework.
II. The concept of neural network and how to implement it with PyTorch.
III. The loss function concept. I used it to design the condition to save the model.
IV. Answering the questions in the report helps me a lot to understand the differences between Q-learning and DQN and the extend issues about these algorithm.