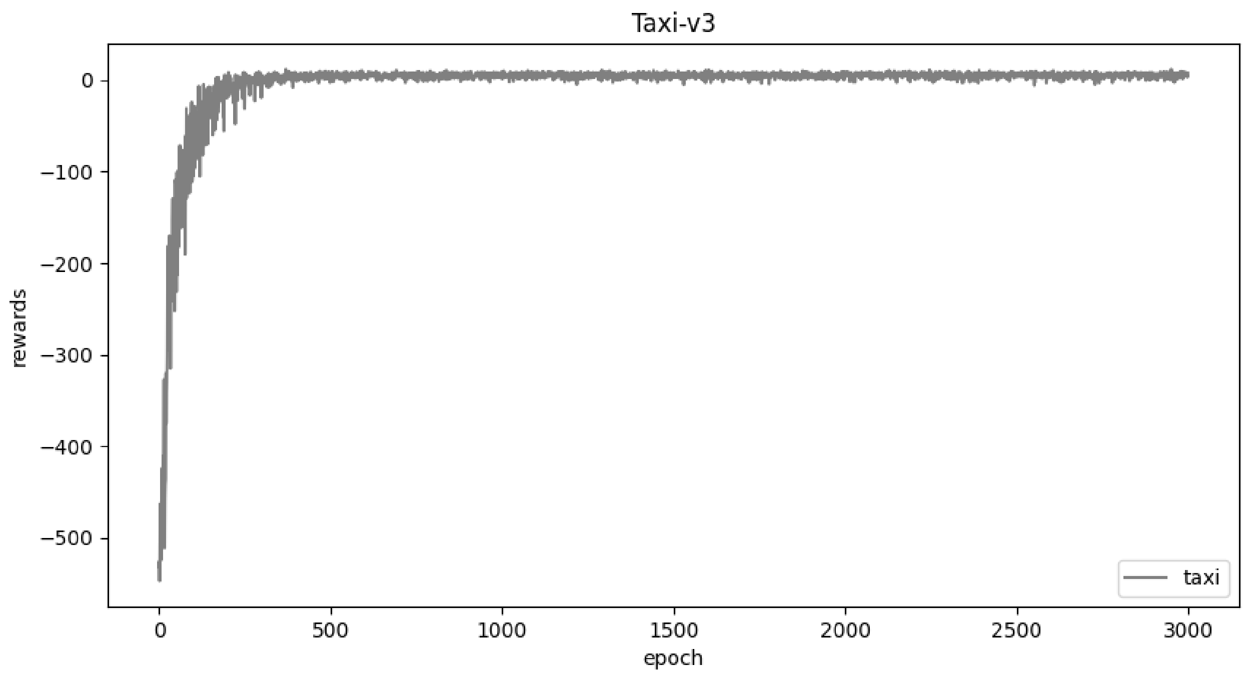
HW4 Report

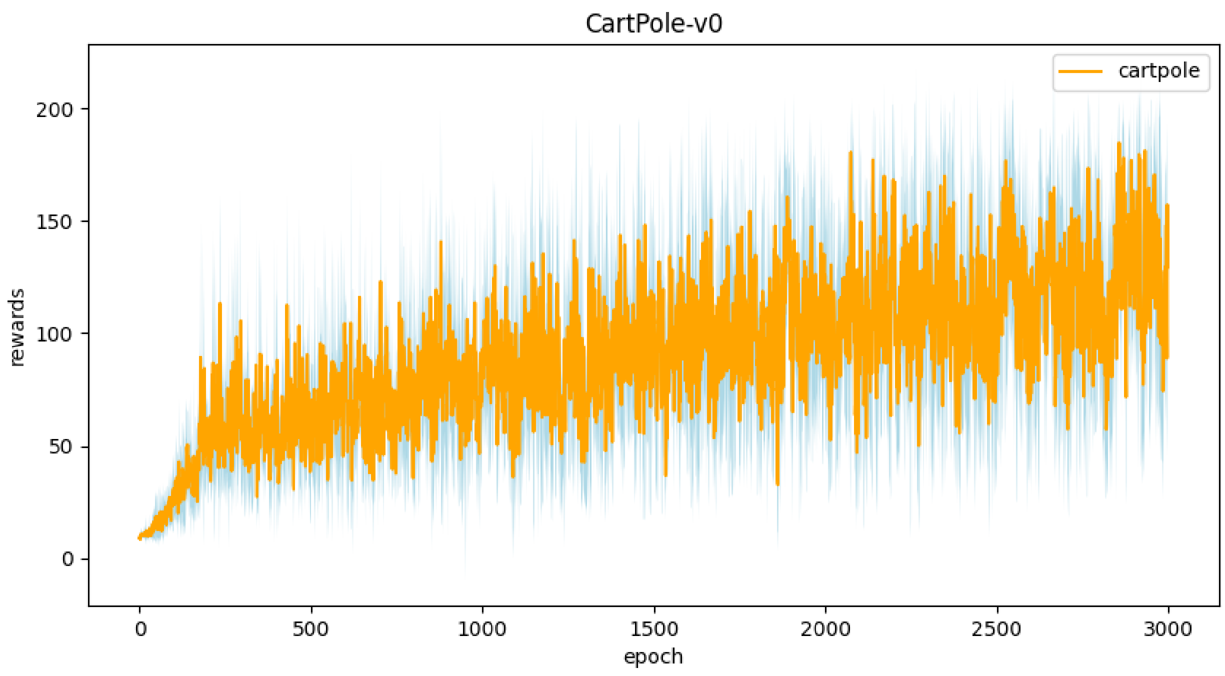
109550206 陳品劭 Self link

Part I. Experiment Results

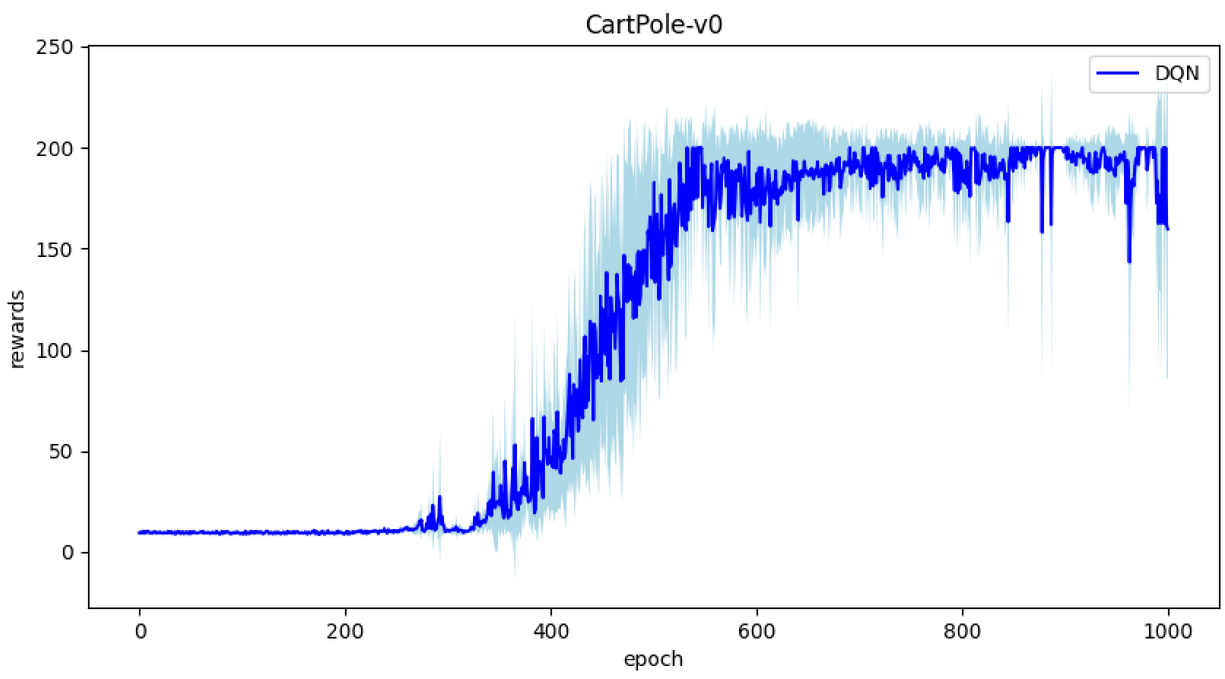
1. taxi.png



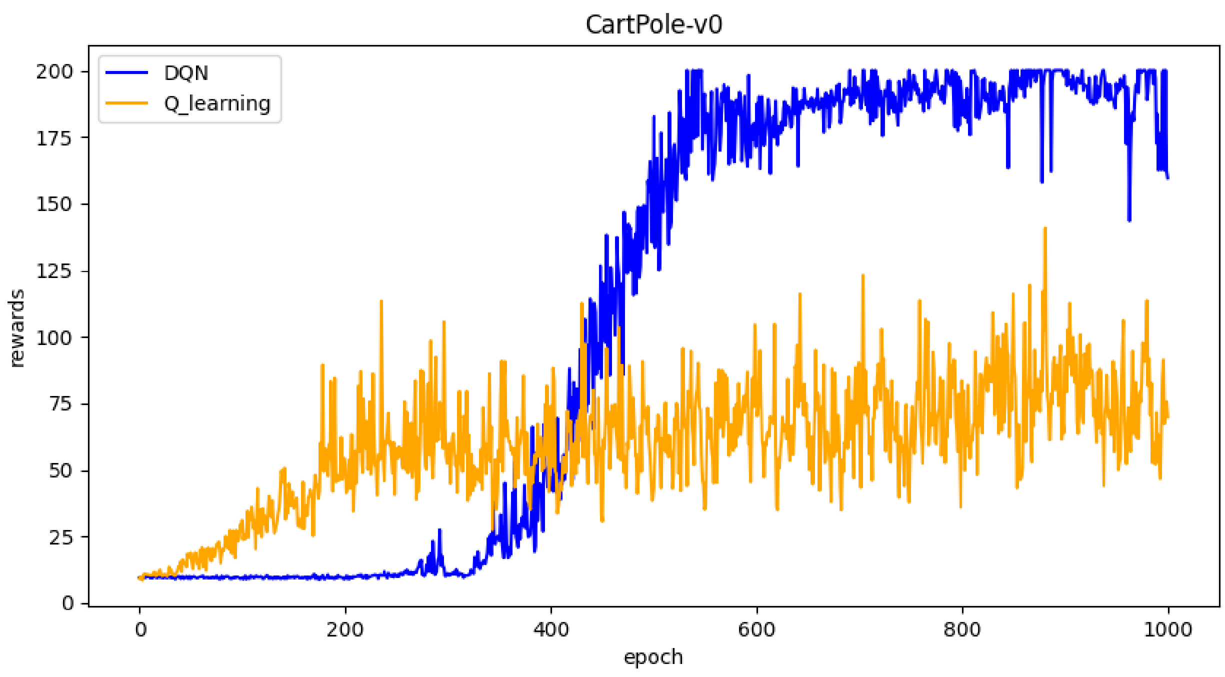
2. cartpole.png



3. DQN.png



4. compare.png



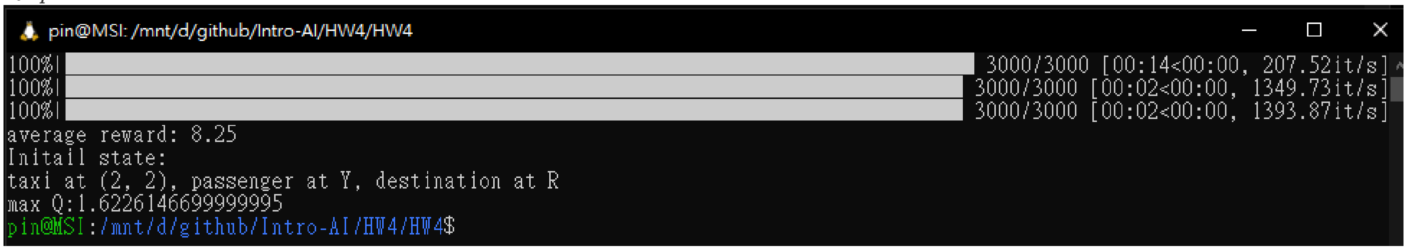
Part II. Question Answering

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%)**

```
+---------+
|R: | : :G|
| : | : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
Rewards:
- -1 per step unless other reward is triggered.
- +20 delivering passenger.
- -10  executing "pickup" and "drop-off" actions illegally.
```

$$\hat{Q}_{\text{opt}}(s,a) = \sum_{s'} \hat{T}(s,a,s')[\widehat{\text{Reward}}(s,a,s') + \gamma \hat{V}_{\text{opt}}(s')]$$

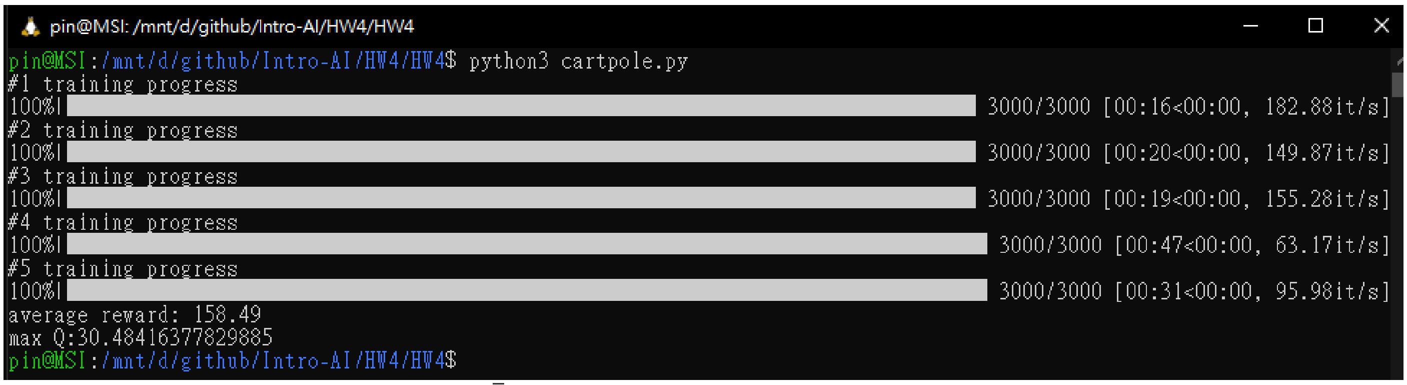
$\hat{Q}_{opt} = 1.6226...$



```
pin@MSI: /mnt/d/github/Intro-AI/HW4/HW4
100%|████████████████████████████████████████| 3000/3000 [00:14<00:00, 207.52it/s]
100%|████████████████████████████████████████| 3000/3000 [00:02<00:00, 1349.73it/s]
100%|████████████████████████████████████████| 3000/3000 [00:02<00:00, 1393.87it/s]
average reward: 8.25
Initail state:
taxi at (2, 2), passenger at Y, destination at R
max Q:1.6226146699999995
pin@MSI:/mnt/d/github/Intro-AI/HW4/HW4$
```

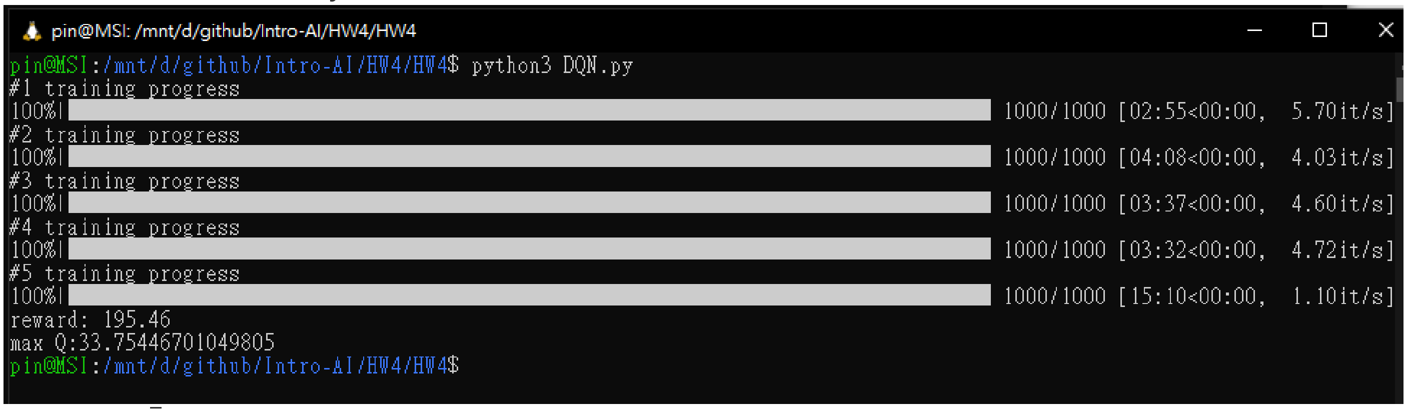
Very close with max Q.

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%)**
   $\frac{1}{1-\gamma} = \frac{1}{1-0.97} = 33.\bar{3}$



```
pin@MSI: /mnt/d/github/Intro-AI/HW4/HW4
pin@MSI:/mnt/d/github/Intro-AI/HW4/HW4$ python3 cartpole.py
#1 training progress
100%|████████████████████████████████████████| 3000/3000 [00:16<00:00, 182.88it/s]
#2 training progress
100%|████████████████████████████████████████| 3000/3000 [00:20<00:00, 149.87it/s]
#3 training progress
100%|████████████████████████████████████████| 3000/3000 [00:19<00:00, 155.28it/s]
#4 training progress
100%|████████████████████████████████████████| 3000/3000 [00:47<00:00, 63.17it/s]
#5 training progress
100%|████████████████████████████████████████| 3000/3000 [00:31<00:00, 95.98it/s]
average reward: 158.49
max Q:30.48416377829885
pin@MSI:/mnt/d/github/Intro-AI/HW4/HW4$
```

Since discretize it is not very close to $33.\bar{3}$



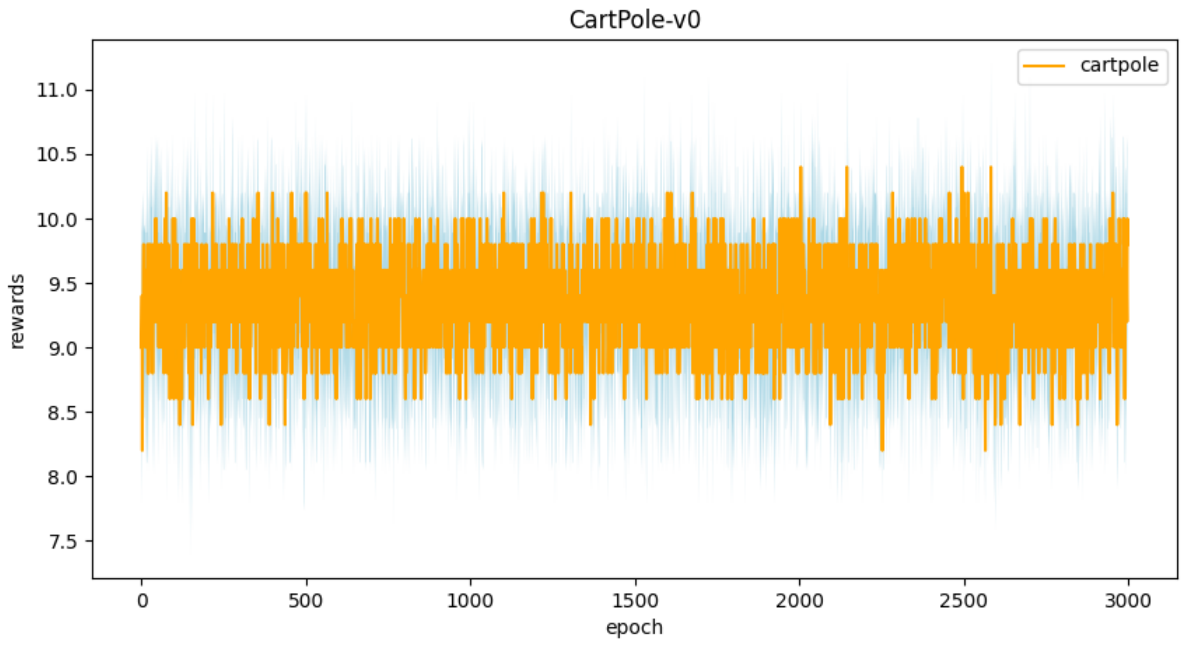
```
pin@MSI: /mnt/d/github/Intro-AI/HW4/HW4
pin@MSI:/mnt/d/github/Intro-AI/HW4/HW4$ python3 DQN.py
#1 training progress
100%|████████████████████████████████████████| 1000/1000 [02:55<00:00,  5.70it/s]
#2 training progress
100%|████████████████████████████████████████| 1000/1000 [04:08<00:00,  4.03it/s]
#3 training progress
100%|████████████████████████████████████████| 1000/1000 [03:37<00:00,  4.60it/s]
#4 training progress
100%|████████████████████████████████████████| 1000/1000 [03:32<00:00,  4.72it/s]
#5 training progress
100%|████████████████████████████████████████| 1000/1000 [15:10<00:00,  1.10it/s]
reward: 195.46
max Q:33.75446701049805
pin@MSI:/mnt/d/github/Intro-AI/HW4/HW4$
```

Close to $33.\bar{3}$

3. **a. Why do we need to discretize the observation in Part 2? (2%)**
   We can use a lighter architecture to learn at the cost of a less efficient policy.
   b. How do you expect the performance will be if we increase "num_bins"? (2%)
   Better.
   c. Is there any concern if we increase "num_bins"? (2%)

We need more resource to train this machine.

4. **Which model (DQN, discretized Q learning) performs better in Cartpole-v0, and what are the reasons? (3%)**
    - DQN
    - 1. Since discretized Q learning use discretized. (See above question)
    - 2. Deep Q Learning tends to overestimate the reward, which leads to unstable training and lower quality policy. Let's consider the equation for the Q value: The last part of the equation takes the estimate of the maximum value. This procedure results in systematic overestimation, which introduces a maximization bias. In DQN, the Q values will be taken from the target network, which is meant to reflect the state of the main DQN. However, it doesn't have identical weights because it's only updated after a certain number of episodes. The addition of the target network might slow down the training since the target network is not continuously updated. However, it should have a more robust performance over time.
5. **a. What is the purpose of using the epsilon greedy algorithm whilechoosing an action? (2%)**
    In the beginning, the agent has none or limited knowledge about the environment, so we need use some to let it discover its environment. Here we use epsilon greedy algorithm. It will let agent try some actions are never taken before and let it know these environment. On the other hand, let it learn which action is better and choose that in that moment.
    b. What will happen, if we don't use the epsilon greedy algorithm in the CartPole-v0 environment? (3%)



    It may get the result look like above. Since it can not get the information of the environment, it will stay in some well-known action and result.

    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 just need some to konw the environment, epsilon greedy algorithm is a simple and efficient way. But we may use other similar way to achieve that. For example, softmax may reach same performance.
    d. Why don't we need the epsilon greedy algorithm during the testing section? (2%)
    Since the agent has enough information about the environment.
6. **Why is there "with torch.no_grad():" in the "choose_action" function in DQN? (3%)**
    Since we only want to use network to choose action here, we do not want to update it.
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 have two network, the Q values can be taken from the target network, which is meant to reflect the state of the main DQN. However, it doesn't have identical weights because it's only updated after a certain number of episodes. It should have a more robust performance over time.
    c. What are the disadvantages? (2%)

The addition of the target network might slow down the training since the target network is not continuously updated.

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%)**
   Relay buffer is a way to remember some samples for agent.
   No.
   The agent will get a relatively good result in a single step, and then affect the subsequent actions, which can be solved by remembering through the relay buffer.
   b. Why do we need batch size? (3%)
   Let us know how many samples we can remember and we have how many samples for every train.
   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 small buffer might force your network to only care about what it saw recently. A large buffer might take a long time to "become refreshed" with good trajectories, when they finally start to be discovered. Conversely, their (large, small) respective advantages

9. **a. What is the condition that you save your neural network? (1%)**
   We save my neural network when done is true and this time is a great one of 5 trainings.
   b. What are the reasons? (2%)
   We can let it train faster by this condition.
   In addition, I try to use test() to be condition to get better result. But it will let train slower. I think that is not a best way.

10. **What have you learned in the homework? (2%)**
    I have learned many thing about deep Q learning, one way of RL. Some implement and detail with DQN.
    In fact, after finishing part 2, I think I know nothing and I don't have any idea with part 3. Then I had to look up a lot of information, and then I felt like I learned something during part 3. And these question let me know many detail about DQN, some I've thought about, some haven't. These questions let me know more thing about DQN.