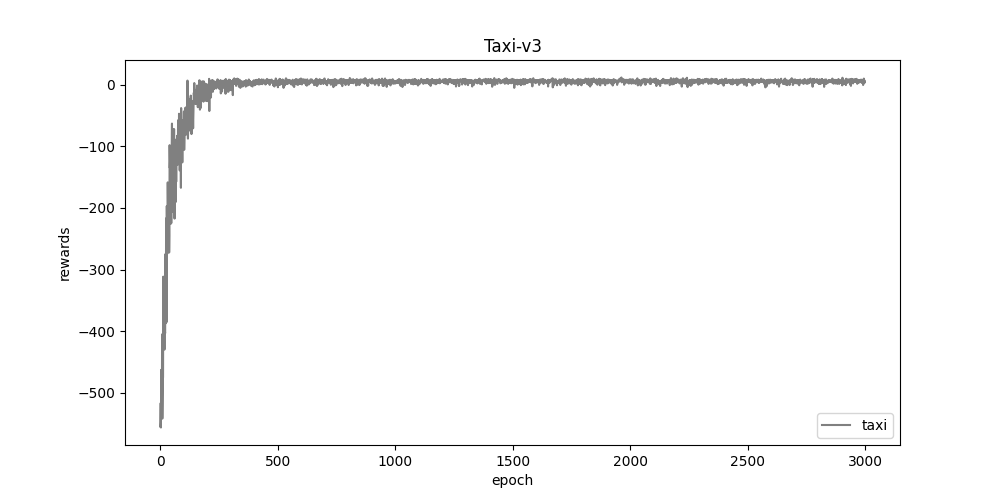
**Report**

**Part I. Experiment Results**

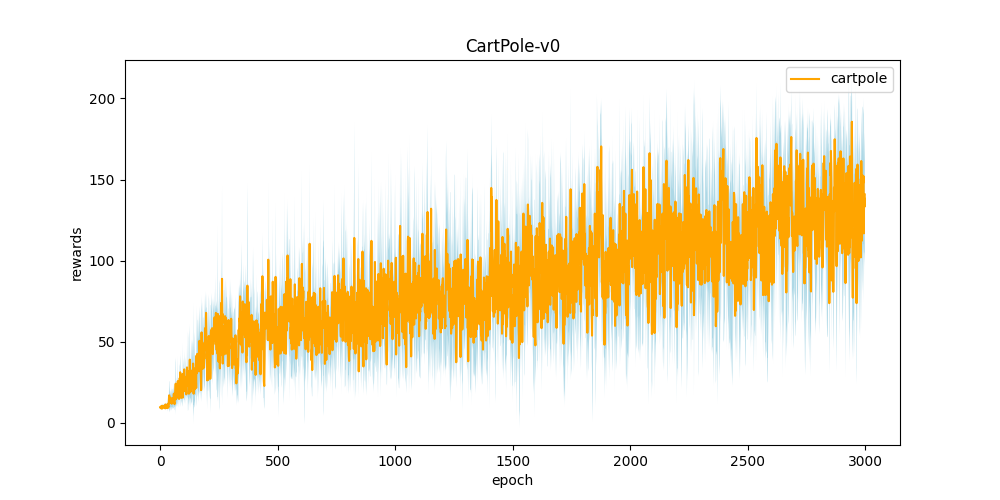
1. taxi.png



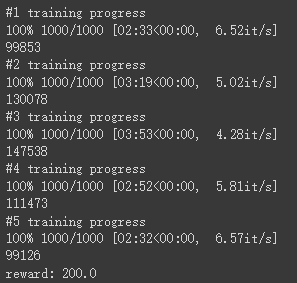


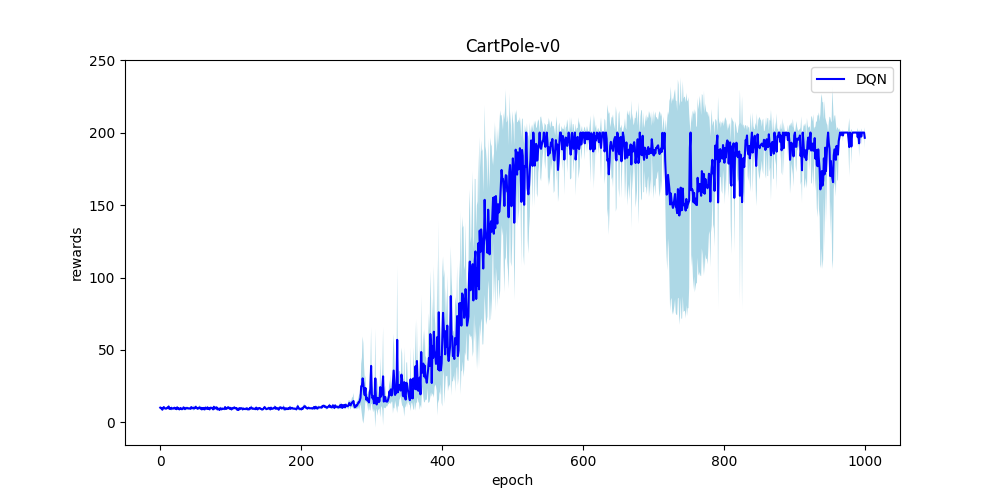
1. cartpole.png



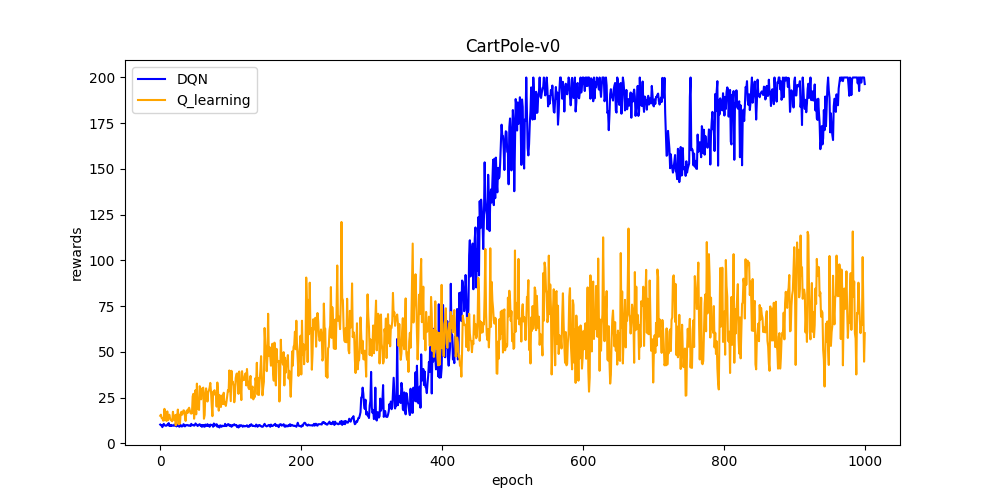


1. DQN.png





1. 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%)

A:



The max Q-value I get from training matches the optimal Q-value which means I trained my agent well.

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%)

A:



The max Q-value I get from training is less than the optimal Q-value which means my agent still needs improvement.

3.

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

A: The observation we observed is a continuous value, so there are infinitely possible state-action pairs. Thus, we need to discretize these values to build a lookup Q-table.

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

A: I expect the performance will be better since the number of states will grow when we increase num\_bins.

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

A: Yes. If we increase num\_bins, Q-table will be larger which makes storing and maintaining Q-table difficult.

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

A: DQN. The Q-learning agent initially performs better than the DQN. This is because the DQN needs a certain amount of data before it can train a reasonable model of the Q-values. The precise amount of data required depends on the complexity of the deep neural network and the size of the state space. The Q-learning agent sometimes performs poorly due to the greedy random action. In contrast, DQN is capable of generalizing to states that it hasn’t seen before, so performance is more stable.

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

A: The aim is to have a balance between exploration and exploitation. Exploration allows us to have some room for trying new things, sometimes contradicting what we have already learned.

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

A: The epsilon greedy algorithm ensures all the action space is explored. If we use the greedy algorithm instead of the epsilon greedy algorithm, the agent will always choose the action with the maximum expected return and the action space will be explored very little.

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

A: It is possible. Just by pure luck, the greedy algorithm can end up selecting the optimal action from the very beginning. In that case, the epsilon greedy algorithm will never match the performance of the greedy algorithm.

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

A: Because we saved the best Q-table after we trained our agent for five times, the action we take during the testing section just takes the index of the maximum Q-value of each state.

1. Why is there “with torch.no\_grad():“ in the “choose\_action” function in DQN? (3%)

A: torch.no\_grad() prohibits the calculation of gradients. It can reduce the use of computational memory.

7.

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

A: Yes. The second modification to online Q-learning aimed at further improving the stability of our method with neural networks is to use a separate network for generating the targets y\_j in the Q-learning update. More precisely, every 100 steps update we clone the network Q to obtain a target network Q^ and use Q^ for generating the Q-learning targets y\_j for the following C updates to Q. This modification makes the algorithm more stable compared to standard online Q-learning, where an update that increases Q(s\_t, a\_t) often also increases Q(s\_{t+1}, a) for all a and hence also increases the target y\_j, possibly leading to oscillations or divergence of the policy. Generating the targets using an older set of parameters adds a delay between the time an update to Q is made and the time the update affects the targets y\_j, making divergence or oscillations much more unlikely.

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

A: 1. It makes our training more stable. 2. Break the correlation between states s and s’.

1. What are the disadvantages? (2%)

A: It uses 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%)

A:

1. Replay buffer is used to store trajectories of experience when executing a policy in an environment. During training, replay buffers are queried for a subset of the trajectories to "replay" the agent's experience.
2. As a supervised learning model, deep Neural Network requires data to satisfy independent and identical distribution and when the agent interacts with the environment, the sequence of experience tuples can be highly correlated. Replay buffer breaks this correlation through the storage-sampling method. In addition to breaking harmful correlations, experience replay allows us to learn more from individual tuples multiple times, recall rare occurrences, and in general make better use of our experience. If there is no experience replay, the algorithm basically performs gradient descent in the same direction for a continuous period of time, so the direct calculation of the gradient under the same steps may not converge.
3. Advantages: 1. High data usage and computation efficiency ── a sample is used multiple times. 2. The correlation of consecutive samples will make the variance of the argument update relatively large, this mechanism can reduce this correlation.

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

A: The batch size defines the number of samples that will be propagated through the network and impacts how quickly a model learns and the stability of the learning process. The size of the batch size affects the time required to complete each epoch and the smoothness of the gradient between each iteration in the deep learning 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%)

A: (1) The larger the replay buffer is, the less likely we will sample correlated elements, hence the more stable the training of the neural network will be. However, a large replay buffer also requires a lot of memory and it might slow training.

When the batch size is larger, the difference between adjacent batches is smaller, and the gradient is smoother during training (loss oscillation will also be smaller). On the contrary, when the batch size is smaller, the gradient oscillation is larger. For the model, a large batch size is more benefit to model convergence because the gradient is smooth and stable, but the randomness of iteration during training will also be smaller and affect the generalization ability of the model. The relatively smaller batch size has a lot of randomness and the generalization ability of the model training will be better.

(2) Advantages:

* Reduces training time and improves stability when increases the size
* With appropriate increase in batch size, the accuracy of gradient descent direction increases and the amplitude vibration during training decreases

Disadvantages:

* Larger batch size leads to a decrease in the generalization ability of the model
* Requires more memory capacity when increases the size

9.

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

A: At the end of each round of training, judge done and max reward. If done and the reward of this round is larger than max reward which is recorded in previous rounds.

b. What are the reasons? (2%)

A: Each round of training may lead to different reward, so I choose the round which has the best performance to store the neural network. Therefore, I can get a better result.

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

A: I get familiar with the environment of Open AI, algorithm of Q-learning and DQN. Because I got different average reward each train, I need to determine the timing to save my best Q-table. In the homework, I learned I could use a global variable to store the maximum value of reward and save Q-table when the current reward is larger than the maximum value of reward which stored before.