**Homework 4:**

**Reinforcement Learning**

**Report Template**

**Please keep the title of each section and delete examples. Note that please keep the questions listed in Part III.**

**Part I. Implementation (-5 if not explain in detail):**

* **Please screenshot your code snippets of Part 1 ~ Part 3, and explain your implementation.**

**Part 1: taxi.py**

**一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述**

**一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

自動產生的描述**

**Part 2: cartpole.py**

**一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述**

**Part 3: DQN.py**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

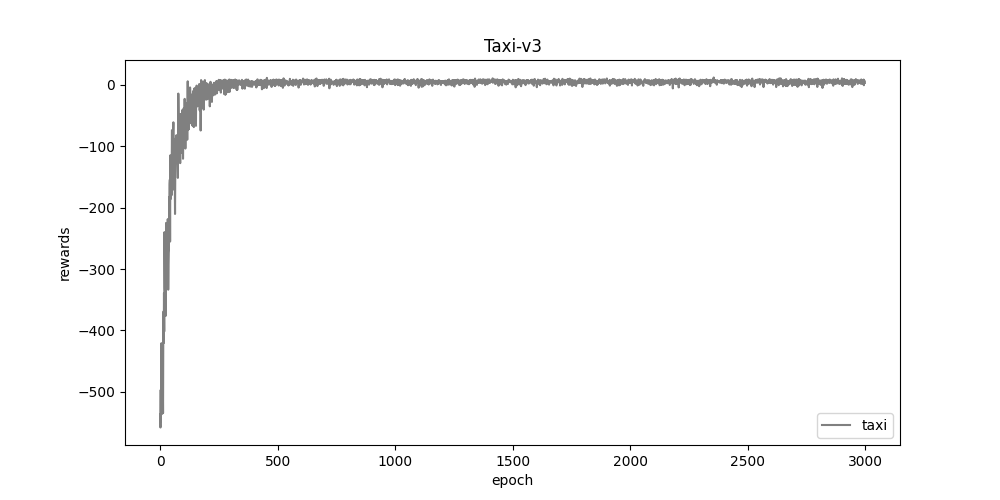
**一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

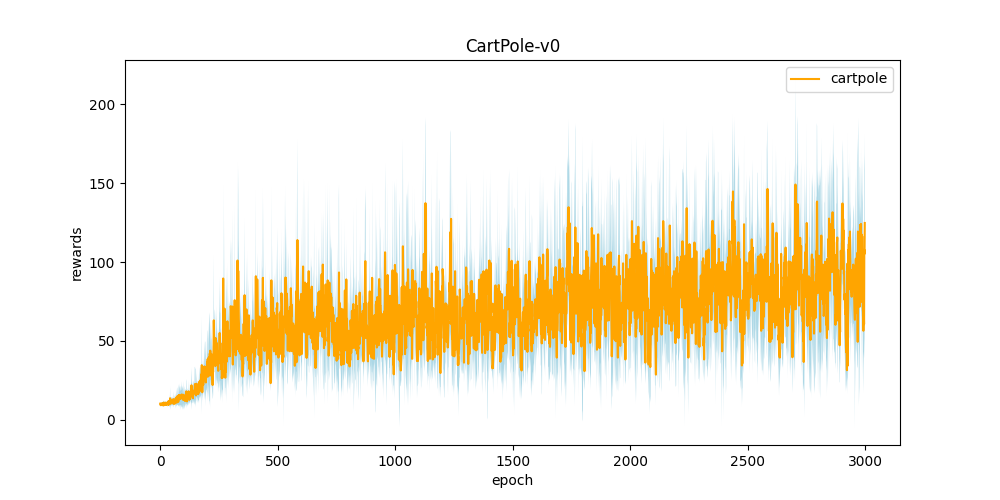
自動產生的描述一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

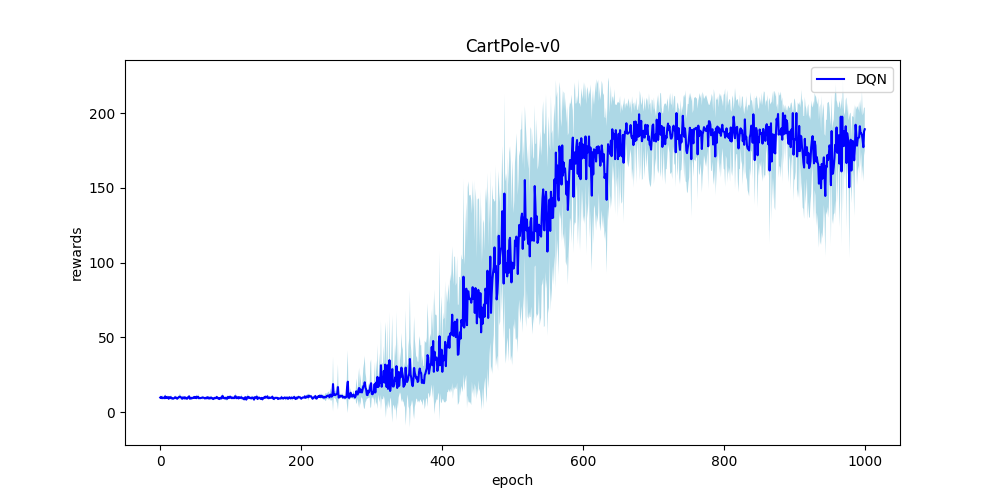
自動產生的描述**

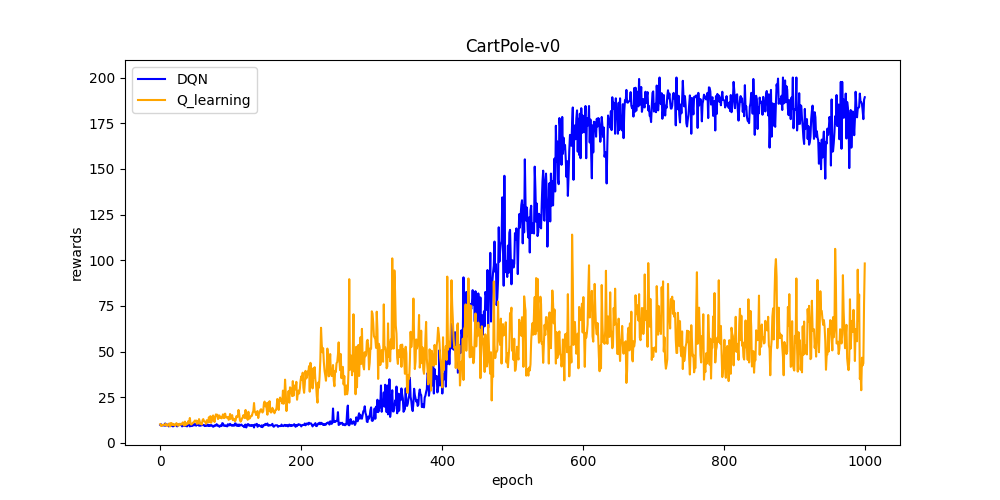
**Part II. Experiment Results:**

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

**1. taxi.png**

**2. cartpole.png**

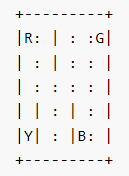
**3. DQN.png**

**4. compare.png**

**Part III. Question Answering (50%):**

1. Calculate the optimal Q-value of a given state in Taxi-v3, 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). **(10%)**

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述 

Q = (-1)\*(1-r^num\_steps)/(1- r)+(20\*r^num\_steps)= -1\*(1-0.9^9) / 0.1 + 20\*0.9^9 = 1.622

optimal Q: L -> L -> D -> D -> pickup -> U -> U -> U -> U -> dropoff

1. 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) **(10%)**

一張含有 文字, 字型, 螢幕擷取畫面, 圖形 的圖片

自動產生的描述

Q = (1 - r^ave.reward) / (1 - r) = 31.74...

1. Why do we need to discretize the observation in Part 2? **(3%)**

The observed data is continuous, so we discretize it to several parts.

1. How do you expect the performance will be if we increase “num\_bins”? **(3%)**

I think it may be better because it can separate to more parts so that it can be more precise.

1. Is there any concern if we increase “num\_bins”? **(3%)**

The efficiency may be slower and it need more cost on qtable.

1. Which model (DQN, discretized Q learning) performs better in Cartpole-v0, and what are the reasons? **(5%)**

DQN model.

From compare.png, we can find that whem we try over about 400 opoches, the reward of DQN will be higher than Q-learning. I think it is because that Q-learning need to discretize to several parts and DQN can use the continuous data.

1. What is the purpose of using the epsilon greedy algorithm while choosing an action? **(3%)**

It can balance exploration and exploitation by choosing them randomly.

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

All exploration: (All random) Cannot remain the known best situation.

All exploitation: Can only work on known situation, may miss some case.

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

Yes, epsilon greedy alogithm select randomly. I think there might be some ways more reliable, maybe use the probability or something else.

1. Why don’t we need the epsilon greedy algorithm during the testing section? **(3%)**

We use this alogorithm to train the model and we don’t need to do that while testing.

1. Why does “with torch.no\_grad():“ do inside the “choose\_action” function in DQN? **(4%)**

In “choose\_action”function, “with torch.no\_grad()”can disable gradient calculation during a block of code. It will set the “requires\_grad” to False so that it can save many memory and computation.