**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.**

**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, 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Part 1**

**Part II. Experiment Results:**

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

**一張含有 文字, Rectangle, 螢幕擷取畫面, 圖表 的圖片

自動產生的描述1. taxi.png:**

**一張含有 文字, 螢幕擷取畫面, 繪圖, 圖表 的圖片

自動產生的描述2. cartpole.png**

**3. DQN.png**

**4. compare.png**

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

1. **一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

   自動產生的描述**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%)**
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) **(10%)**
3. Why do we need to discretize the observation in Part 2? **(3%)**

We nned to discretize the observation to let us making the Q table easier

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

The performance will be better, because the accuracy of the observations will be

improved.

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

Dimension of the Q table would grow, and the runnning time and space complexity are going to increase.

1. Which model (DQN, discretized Q learning) performs better in Cartpole-v0, and what are the reasons? **(5%)**
2. What is the purpose of using the epsilon greedy algorithm while choosing an action? **(3%)**

When the Q table is just initialized, the agent don’t know any imformation to

choose the action, so the algorithm can make the agent try some actions to learn some data.

1. What will happen, if we don’t use the epsilon greedy algorithm in the CartPole-v0 environment? **(3%)**
2. Is it possible to achieve the same performance without the epsilon greedy algorithm in the CartPole-v0 environment? Why or Why not? **(3%)**
3. Why don’t we need the epsilon greedy algorithm during the testing section? **(3%)**
4. Why does “with torch.no\_grad():“ do inside the “choose\_action” function in DQN? **(4%)**