**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.**

1. **taxi.py**

**we first use random to determine the action will choosed**

**if the random number less than self.epsilon, we choose the action by random**

**ifthe random number greater than self.epsilon, we choose the action with maximun value in qtable**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

**from the formula of Q-learning we update the qtable by the equation.**

**一張含有 文字, 字型, 白色, 行 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**

just return the maxvalue in the specific state in q-table. **一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述**

1. **cartpole.py**

**we use function linspace to separate the upper bound and lower bound into n interval, but**

**we don’t the first and last element, so we eliminate the first and last element with**

**ans[1:-1]**

1. **一張含有 文字, 螢幕擷取畫面, 字型 的圖片

   自動產生的描述**
2. **DQN.py**

**Part II. Experiment Results:**

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

1. **taxi.png:**

**一張含有 文字, Rectangle, 螢幕擷取畫面, 行 的圖片

自動產生的描述**

1. **cartpole.png**

**一張含有 螢幕擷取畫面, 繪圖, 文字, 圖表 的圖片

自動產生的描述**

**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 螢幕擷取畫面, 文字, 行 的圖片

自動產生的描述

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

Because the data we observe iscontinuous, but we need to make it into discrete

states, so we have to discretize it to get the data in which interval.

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

The performance will become greater, because we will have more states, and the

data after discretize will be more closer to the true data.

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

If we increase “num\_bins”the process speed may become slower, because we increase the number of states and also make the size of Q-table bigger and need more space to calculate.

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%)**

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%)**