**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:**

**一張含有 文字, 電子產品, 螢幕擷取畫面, 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, 作業系統 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:**

**一張含有 文字, 電子產品, 螢幕擷取畫面, 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, 作業系統 的圖片

自動產生的描述**

**一張含有 文字, 螢幕擷取畫面, 陳列, 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, 作業系統 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 3:**

**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 的圖片

自動產生的描述**

**一張含有 文字, 螢幕擷取畫面, 軟體, 作業系統 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:**

**一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述**

1. **cartpole.png**

**一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述**

1. **DQN.png**

**一張含有 文字, 陳列, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

**4. compare.png**

**一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述**

**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, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述**

一張含有 筆跡, 文字, 字型, 書法 的圖片

自動產生的描述

As we can see, the optimal Q value I calculate is very close to that in Taxi-v3.

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, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述**

一張含有 筆跡, 文字, 字型, 書法 的圖片

自動產生的描述

Since in Cartpole-v0 the states are discrete, therefore, the error between code and what I calculate is a little larger than Taxi-v3.

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

The original states should be continuous which produce infinite actions. Thus we cut them into discrete groups to build a table containing limited number of members.

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

Increasing “num\_bins”causes more groups in table mentioned above. If an interval we cut is thinner, the estimation is more closer to the real value. Thus, we have better result if getting higher “num\_bins”.

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

Increasing “num\_bins”indicates we get more groups to calculate, which accumulates tasks for computer that increases space complexity, and also takes much more time to execute.

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

I think DQN will perform better. First, discretized Q learning utilizes discrete data that causes error between estimations and real values. Second, discretized Q learning executes based on Q table that is confined with the number of groups we provide. However, DQN constructs a deep network that can calculate estimations with much more states than Q learning. Thus, DQN handles more cases than Q learning, performing better.

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

We need to find a balanced ratio of exploitation and exploration and produce higher value of reward.

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

It will exploit all the time, and won’t find a new better action as exploration. In this case, we may obtain a low reward.

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 algorithm is aimed to enhance the possibility of randomly choosing other states. It’s still possible to form a totally same combination as a combination from all exploration or all exploitation without algorithm.

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

During the testing section, our agent has been already familiar with the environment, which we finish optimizing Q table. Thus, agent can always choose the best actions.

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

There is a parameter “requires\_grad”in tensor causing automatical update of the net. But we only need the action, which we set the parameter to be false, that is “no\_grad,”to avoid calculation of gradient.