**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 in the possible action**

**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]**

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

**we can easily use function digitize in numpy to calaulate the target value in which interval.**

**一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述**

**when we get the observation value, we need to turn it into discrete value. So for every value in observation, we use the discretize function we write before to turn it into discrete value. 1**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

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

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

**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, 字型 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 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. **DQN.py**

**from the steps in spec:**

**1.every 100 time we load(update) the target\_net (write by TA)**

**2.we first get the state,action reqard next\_state and done from self.buffer**

**and we turn this values into nparray and turn into tensor matrix and ensure they have the**

**same batch size**

**3.we put the state and next state into the network to get the Q value and next Q value note**

**that we only consider the next Q value with haven’t done. Last we use the formula to**

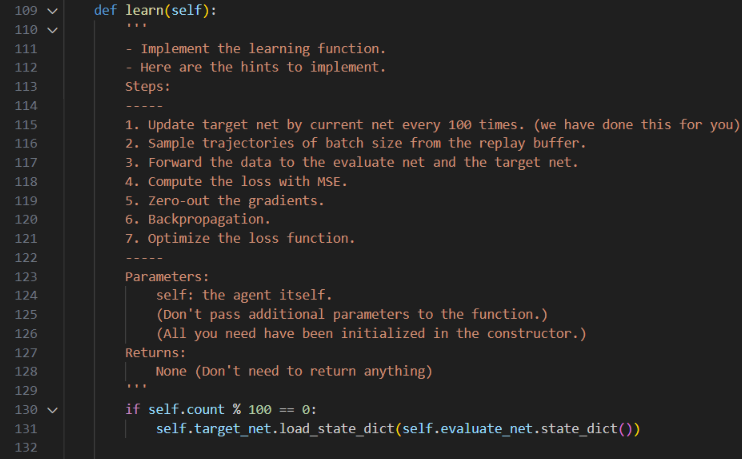
**calculate the target Q value**

**4. we call the loss function from network and put Qvalue and target Q value into it.**

**5.we use self.optimizer.zero\_grad() to zero-out the gradiendts.**

**6.we use loss.backward() to reach backprogation**

**7.we use self.optimizer.step() to optimize the loss function**

****

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

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

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

**if the random number greater than self.epsilon, we first turn the state into tensor matrix and use unsqeeze to make it to higher dimension. Then we can put it into the nueral network to see what may the value get after action, we choose the action with max value.**

**一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述**

**we first turn the initial state(self.env.reset ) into tensor matrix and use unsqeeze to make it to higher dimension. and use the neural network target\_net to calculate the values we get after the calculation and we return the max Q value.**

**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 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**

1. **DQN.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**

**4. compare.png**

**一張含有 文字, 螢幕擷取畫面, 繪圖, 字型 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%)**

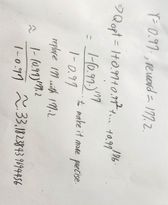
一張含有 螢幕擷取畫面, 文字, 行 的圖片

自動產生的描述

一張含有 文字, 筆跡, 信, 書法 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%)**

1. 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%)**

**DQN perform better in cartpole-v0,in Qlearning we should devided the observation to limited discrete data and DQN can directly use the continuous data, which is more familier to real data. so DQN can have higher performance.**

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

**the purpose is to balance exploration and exploitation, it can prevent the extreme cases, always choose the rangom action to get more information, or choose the best action in with max Q value,(there may be other better action we havn’t explore).**

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

**if we always use explore, choose action randomly, the episode will be easily ended and since we don’t choose the best action in qtable , we may not construct a**

**good Q table.**

**if we always use exploit, we only choose the action that is known, this may cause us can’t not explore the better performance action that we haven’t explore and can’t get a good Q table .**

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 it is possible. Since the CartPole-v0 environment is easily, if we can find another method to balance the explore and exploit rate, we can get the same performance in the CartPole-v0 environment.**

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

**Because in the training section we need to find new action that we haven’t explore. But in the testing section we only want to test the q table we got after training, so we not need the epsilon greedy algorithm during the testing section.**

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

**we don’t need to calculate the gradient and back paropagation in choose action, so we use with torch.no\_grad() to disable gradient tracking.**