HW5

Basic

1. Describe how you implement the Q-learning algorithm.

令 N_EPISODES = 180 與 EPISODE_LENGTH = 200，N_bucket 設為(1, 1, 6, 3)代表有 4 個 state 分別切成幾份，接著創造 q_table，size 為 state 數量*action 數量。

每一輪 training 剛開始時，首先獲取這一輪 epsilon 以及 lr。以及把整個 environment 重置，獲得起始的 state，將 reward 設為 0。在進入 training 時，先選擇 action。在 action 中，如果 random_sample 比 epsilon 還要小，就隨機選擇 action，如果不是，就選擇 q table 中這個 state 最大的 action。接著做這個 action，獲得 observation，reward，done 與 info。累積 rewards，得到 nextstate。接下來尋找 q table 中下一個 state 中 action 的最大 reward，並且帶入以下 function 更新 q table。

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha \left(R(s,a) + \gamma \max_{a'} Q\left(s', a'\right) - Q_{t-1}(s,a) \right)$$

2. Describe difficulties you encountered.

基本上因為助教寫好了要填什麼東西，所以沒遇到什麼問題。

3. Summarize your implementation.

因為旁邊註解寫好要填什麼東西，所以主要能調整的只有 N_EPISODES，n_bucket。多方嘗試後發現 N_EPISODES = 180，N_bucket 為(1, 1, 6, 3)即可達成 rewards 200.0

Advanced-report

1. Describe difficulties you encountered

主要遇到的問題是理解 DQN 的原理花了比較多的時間，上網找教學後，後續打 code 就快很多了。

2. Summarize your implementation.

主要是依照助教旁邊寫的註解去實作，在建造 fully-connected layers 時，分別是 input->hidden，hidden->output。接著將傳入 DQN 的所有參數設定完成，在選擇 action 時，於 q_table 相同的做法，獲得真正的 reward，接著將結果存起來。直到 memory 儲存得夠多後，進入 learning process，最後進入下一個 state。