HW5

r12922016 葉丞勛

1. **Describe the Deep Q-Network**

在 DQN 裡面，除了 __init__ 外，主要有三個 method：act()、process()、learn()，以下會一一介紹：

(1) act()：輸入一個 state，輸出 action。首先會去做判斷：若是目前在 training 並且 random number 小於 epsilon 或是目前 total_steps 小於 warmup_steps 的話，就隨機選擇一個 action；否則，會先透過 network 取得目前 state 對應的 q_values，然後找出具有最大 q_value 的 action，選擇這個 action 作為 output。

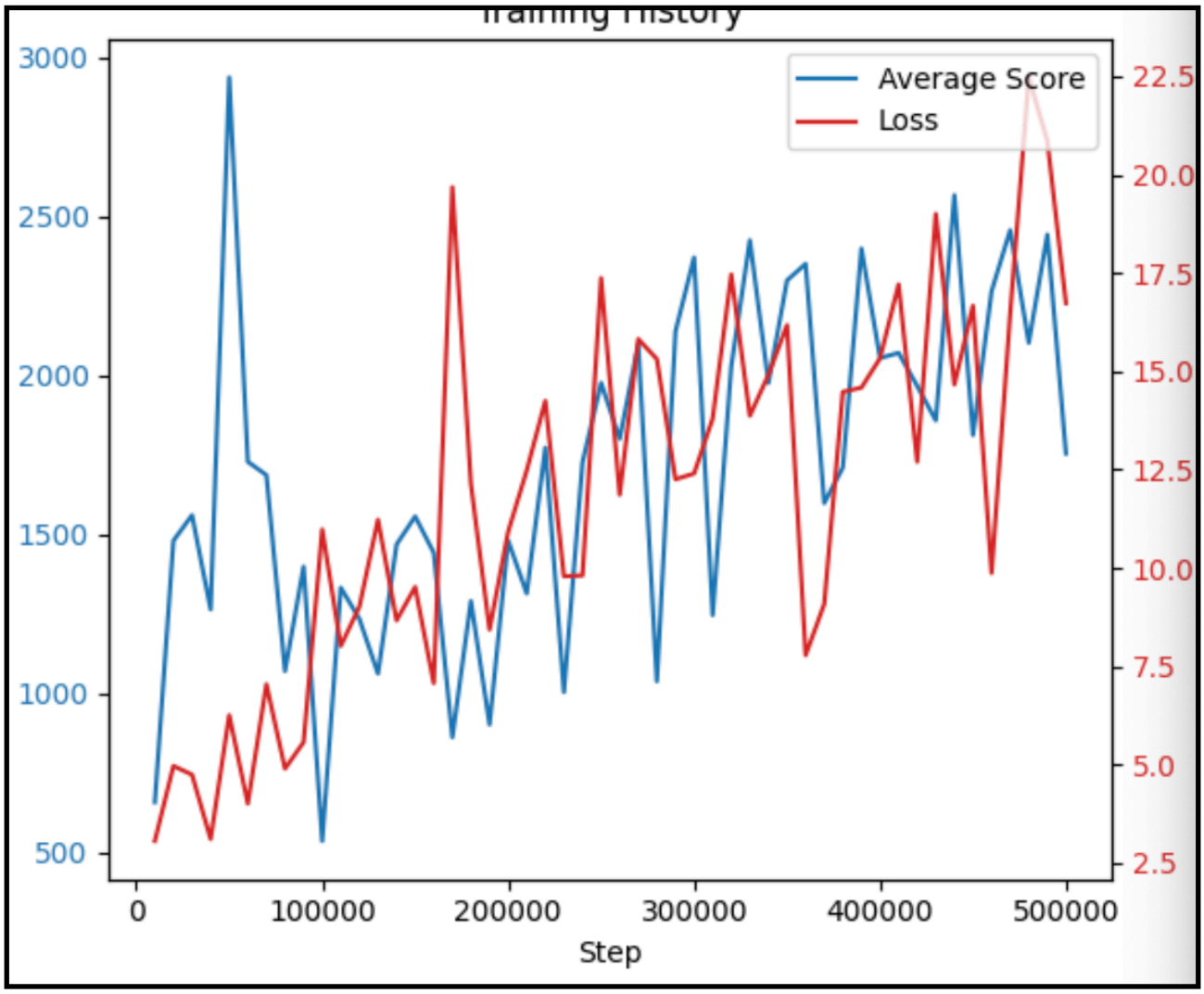
(2) process()：輸入一個 transition，然後去決定 agent 目前要做什麼事。首先，會將目前的 transition 丟進 buffer 裡面去做 update，接著判斷若是 total_steps 大於 warmup_steps，就去 learn()；若是 total_steps 是 target_update_interval 的倍數，就把當前 network 的參數更新到 target_network；之後再將 epsilon 乘上 0.97，讓數值隨著時間慢慢降低。

(3) learn()：從 replay buffer 隨機 sample mini_batch 來訓練 q_network。首先從 buffer sample mini_batch，然後將剛剛 sample 得到的 state 跟 action，透過 network 取得 action 對應的 q_values。接著將剛剛 sample 得到的 next_state 丟進 target_network，將最大的 q_values 作為 next_q_values。然後透過 td_target = reward + (1 – terminated) * self.gamma * next_q_values 算出 td_target，再透過 Huber loss 去算出 q_values 跟 td_target 的 loss，然後做 backpropagation 去更新 network。

2. **Describe the architecture of your PacmanActionCNN**

總共有 3 層 convolutional layer 加上 2 層的 fully connected layer，其中每層 convolution layer 後面會再接 batchNorm2d 去做正規化，並且第一層的 convolution layer 後面還會再去做 max pooling， 提取重要的特徵。

3. **Plot your training curve, including both loss and rewards**



4. **Show screenshots from your evaluation video**