Q1: Models (2%)

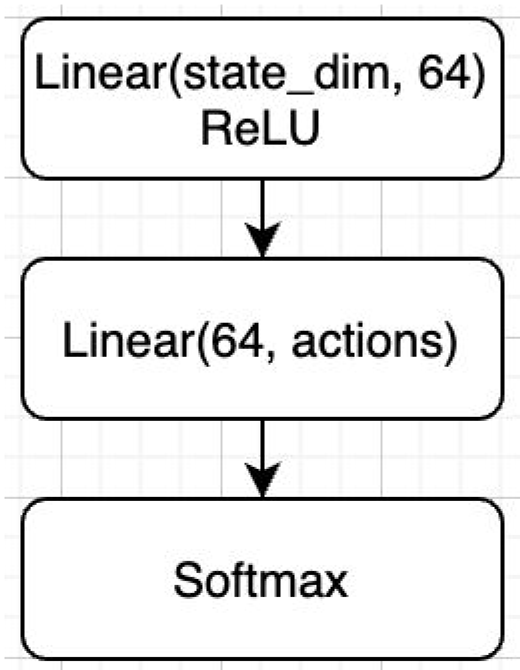
- Describe your Policy Gradient & DQN Model
- Plot the learning curves of rewards
    - You may need to use Moving Average when plotting the curves

Policy Gradient:
我的模型是採用助教給的tutorial code。
Optimizer: Adam

整體流程如下：
假設模型為 P

1. For each episode
    a. Given state s_t, take an action a_t
        i. $a_t = P(s_t)$
    b. Obtain reward r_t, s_t
    c. Store reward r_t to Rs, store action $log(a_t)$ to As
2. Update model
    a. discount rewards
    b. Update Rs: [$R_i = r_i + \gamma * R_{i+1}$ for r_i in Rs]
    c. Normalize Rs
    d. loss=sum([-r*log_p for r, log_p in (Rs, As)])
3. Calculate average rewards
4. If average reward > 50, stop, else go back a.

我的 PG 模型架構如下：



DQN:

我的模型是採用助教給的tutorial code，因助教所給的是 double DQN，所以就基於此來繼續實作。

Optimizer: RMSprop

整體流程如下：

假設模型為Q，目標模型為T，全部可執行的actions為A

1. Initialize Q, T=Q
2. For each episode
    a. Given state s_t, take an action a_t from Q by epsilon greedy
        i. Given random probability p
        ii. $threshold = end + (start - end) * e^{-\frac{step}{decay}}$
        iii. if p > threshold, action $a_t = max(Q(s_t))$
        else action $a_t = random(A)$
    b. Obtain reward r_t and next state s_{t+1}
    c. Store info (s_t, a_t, r_t, s_{t+1}) to buffer
    d. If the buffer is full, sample info and update Q else go back to a.
    e. Sample info (s_t, a_t, r_t, s_{t+1}) from buffer as batch data

f. Update Q

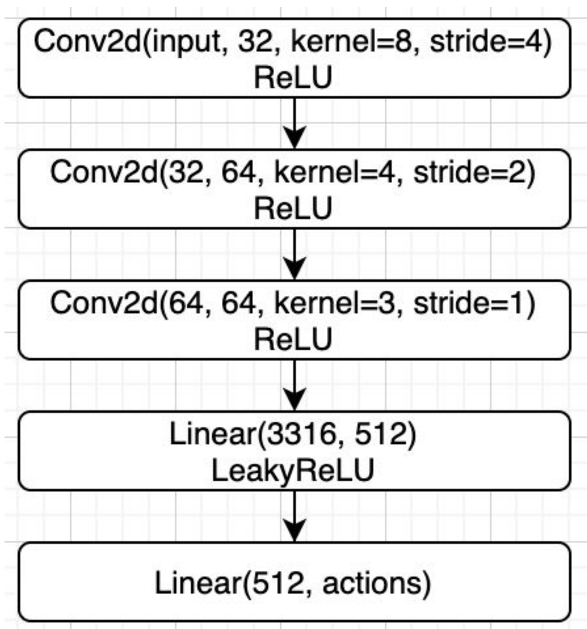
i. $v_{except} = r_t + \gamma * T(s_{t+1}, max(Q(s_{t+1}, A)))$

ii. $v_{current} = Q(s_t, a_t)$

iii. $loss = \left\| action_{except} - action_{current} \right\|_1$

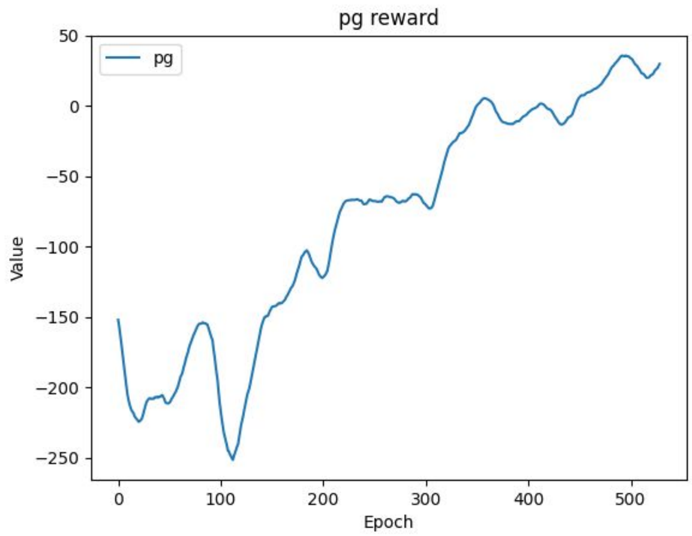
g. For every C step, assign T=Q

我的 DQN 模型架構如下：

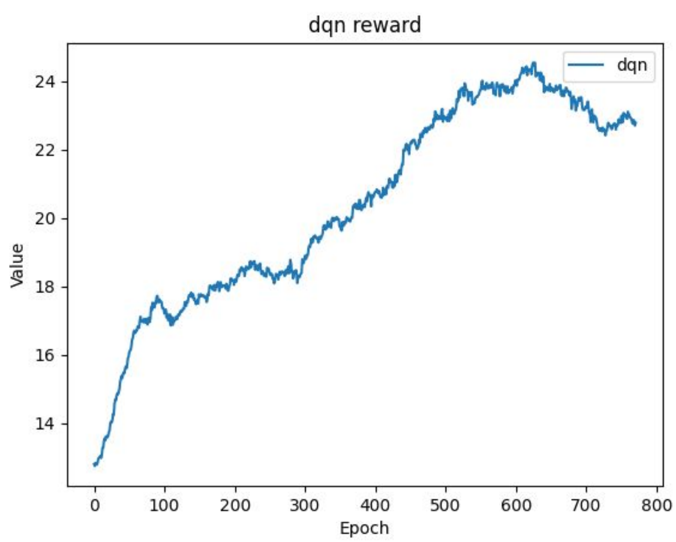


下圖為 policy gradient 的 reward curve，我設定 window size = 20 來

平滑 moving average



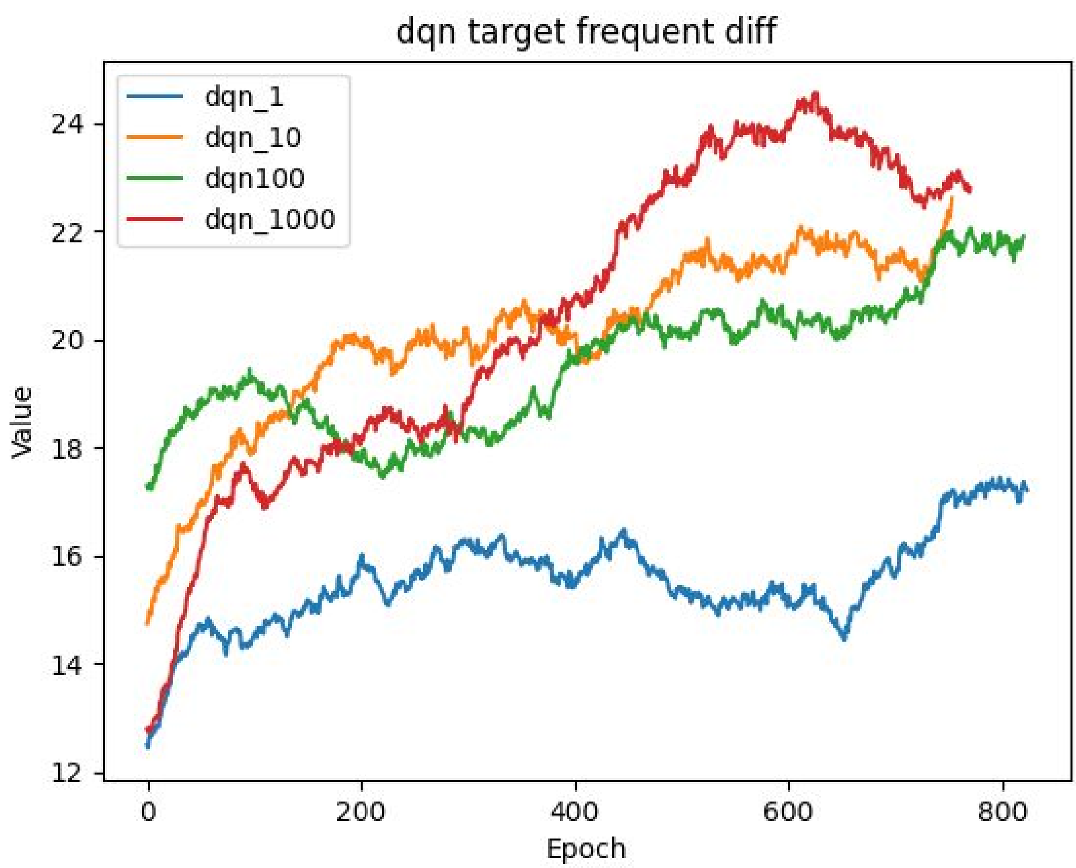
下圖為 DQN 的 reward curve，我設定 window size = 200 來平滑 moving average



Q2: Hyperparameters of DQN (4%)

- Choose one hyperparameter of your choice and run at least three other settings of this hyperparameter

- Plot all four learning curves in the same figure
- Explain why you choose this hyperparameter and how it affect the results
- Candidates: gamma, network architecture, exploration schedule/rule, target network update frequency, etc.
- You can use **any environment** to show your results



我選擇改變的參數是 target network update frequency，選擇的環境是小精靈。這個參數所影響的是 target net 多久會更新一次，target net 所代表的是下個 state 的期望值，也就是說改變這個參數會影響之後下個 state 所回傳的數

值。

上圖是不同 frequency 畫出來的圖，分別有 1，10，100，1000，為了方便呈現，只考慮前面 200000 steps 的 reward，我設定 window = 200 來平滑曲線。

從圖中可以發現，更新的愈頻繁，如藍色的線，每個 step 都在更新，可能會讓 online model 和 target model 太過接近，無法給予一個好的期望值；反倒是更新較為緩慢的紅色線，每 1000 step 才更新一次，給予了非常好的效果。

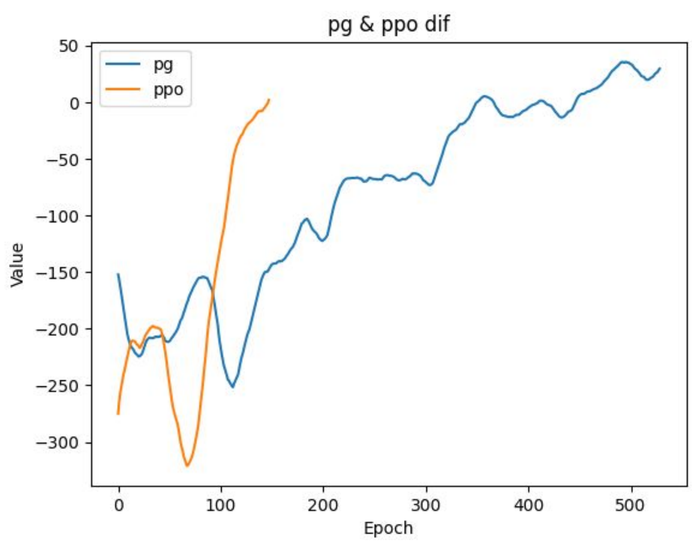
Q3: Improvements of Policy Gradient / DQN (4%)

- Choose **two** improvements of Policy Gradient or DQN
    - Describe the improvements and why they can improve the performance
    - Plot the learning curves and compare results with and without improvement
- You can train in **any environment** to show your results, so you should better choose an environment where you can see significant differences between those methods.
- You **do not** need to submit the code of this part

我選擇 PPO (Proximal Policy Optimization) 來優化 PG 和 Dueling

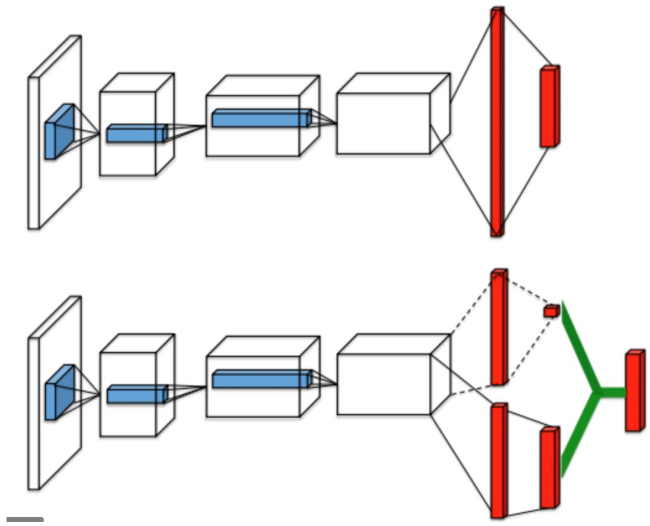
Network 優化 DQN

原本的 PG 是 on-policy 的方法，而PPO 使用 off-policy ，利用 important sampling 的方式達成，這樣可以增進每次更新時 sample 不均衡的現象，用兩個 network, actor, critic 來計算出 advantage ，判斷此 sample 的權重。再來利用 clip 把 loss 限制在一定範圍 [-1, 1]，讓兩個 network 不會差異太大。
下圖是 pg 和 ppo 的 reward 比較圖，訓練停止條件都是在當 reward > 50 ，我設定 window = 10 去平滑化，從圖中可以發現 ppo 非常快就達到停止條件，而 pg 要花大約三倍左右的時間才能緩慢提升。

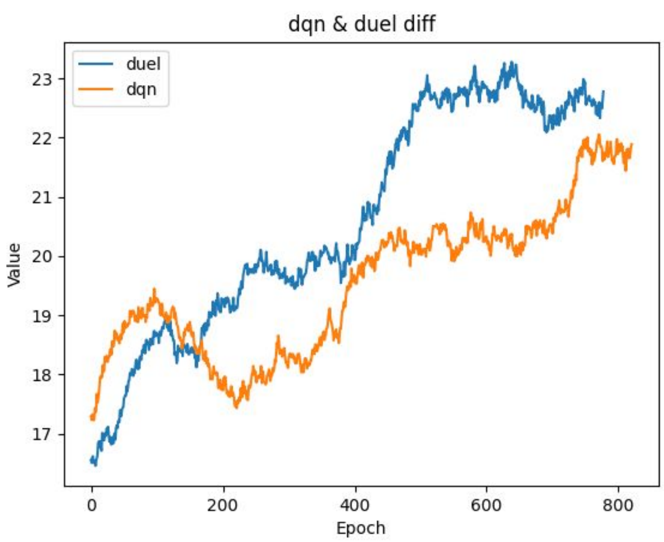


原本的 DQN 已經是 Double DQN 了，這個改良是基於 double 的情況下實作，所以是 Double Duel DQN

Duel DQN 把模型架構多輸出一個 value ，如下圖，最後再把兩個加起來，去限制模型直接從 Q 去找答案，如此可以讓 Q 裡面的每個值不會變動的太獨立。



下圖是 double dqn 和 double duel dqn 的 reward 比較，我設定 window = 200 去平滑化，從圖中可以發現 duel 提升的速度非常明顯，但兩著的差異我覺得沒有 pg，ppo 之間差距明顯，可能是 duel 的 variance 比較大，相比起來 double dqn 比較穩定，不過仍然有不少提升。



Bonus: Fine-tuning Your HW1 Summarization (2%)

- Describe the RL algorithm(s) you use.
- Analyze the results between RL / supervised learning.
    - If you get a better (or worse) performance, try to explain why.
    - Sample some summaries from both models, analyze the human readability and sentence quality.
- We will grade this part according to the experiment and analysis (eg: is your experiment setup reasonable). **You do not need to outperform your best result in HW1.**