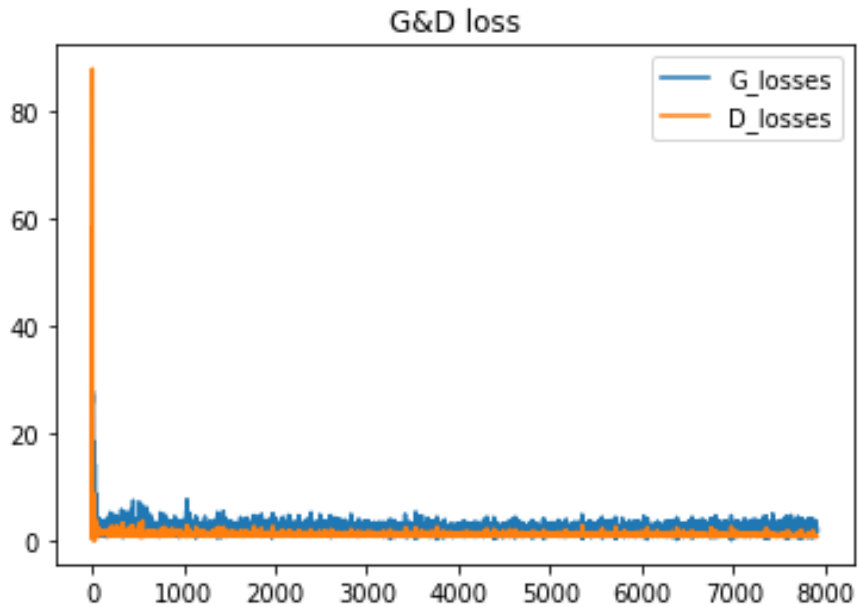


DL HW3

1. GAN

(1) 對照片進行 resize 成 64×64 與 crop 也是 64×64

(2)



(3) 在 train 的過程中，試過非常多改法，有比較 generator 與 discriminator 的

參數是否有調整起始值，也是過很多 seed，但發現在大部分過程中 generator 的 loss 都非常難下降，上面是唯一比較穩定的，所以照片出來也相較清楚，但近看還是有很多模糊的地方，可能可以嘗試其他 GAN 的寫法如 cycle GAN。

2. DQN

(1)

updating step α : 這個值為衡量過往訓練與現在考慮哪個比較多，若 α 越大則表示較不重視過去訓練的結果

discount factor γ : 這個值代表你偏好記憶中的利益還是眼前的利益，越大則越重視記憶中的結果，反之則重視眼前利益

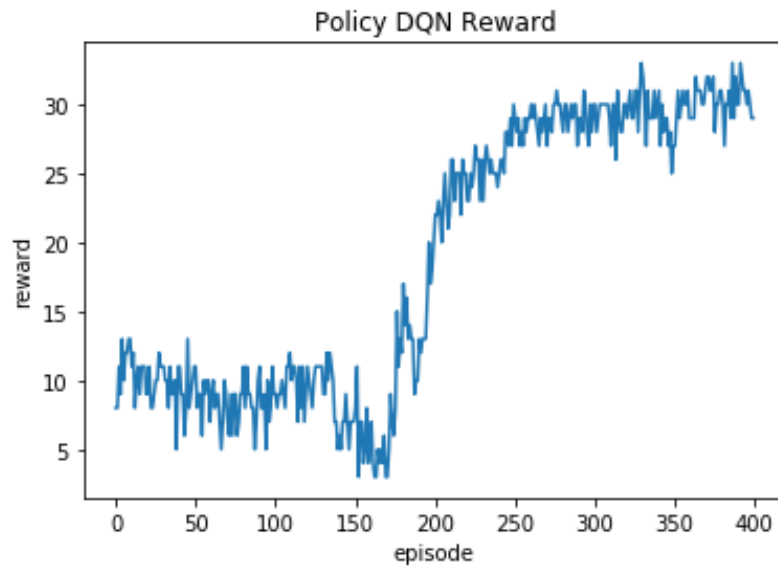
target network update period τ : 表示多久更新一次 target net

ϵ -greedy: 讓選擇可以有可變性，因為若某個 state 的 A action reward 很高，那每次到那個 state 都會選擇 A action，為了避免這個情況，因此增加了一點機率進去，而 ϵ 就類似 threshold 的概念，如果抽樣結果小於 ϵ 就機率選 action，否則選 reward 最大的，且會隨著迭帶次數增加變小，減少向外探索的機會

(2)

```
Episode: 392, interaction_steps: 802816, reward: 33, epsilon: 0.100000
Episode: 393, interaction_steps: 804864, reward: 32, epsilon: 0.100000
Episode: 394, interaction_steps: 806912, reward: 31, epsilon: 0.100000
Episode: 395, interaction_steps: 808960, reward: 31, epsilon: 0.100000
Episode: 396, interaction_steps: 811008, reward: 30, epsilon: 0.100000
Episode: 397, interaction_steps: 813056, reward: 31, epsilon: 0.100000
Episode: 398, interaction_steps: 815104, reward: 30, epsilon: 0.100000
```

(3)



從圖可以發現在前面學習的時候 reward 都沒什麼增長，需要到 episode 約 200 次時才會開始急速上升，但上升到 30 幾又開始停滯了

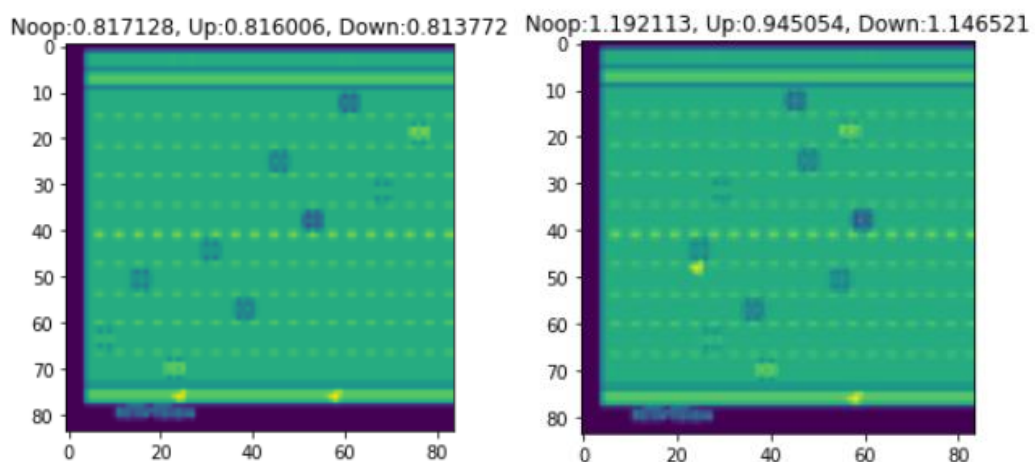
(4)

[Info] Restore model from 'model/' !

| | | |
|----------|-----------------------|----------------------------------|
| Episode: | 0, interaction_steps: | 0, reward: 31, epsilon: 0.100000 |
| Episode: | 1, interaction_steps: | 0, reward: 31, epsilon: 0.100000 |
| Episode: | 2, interaction_steps: | 0, reward: 32, epsilon: 0.100000 |
| Episode: | 3, interaction_steps: | 0, reward: 29, epsilon: 0.100000 |
| Episode: | 4, interaction_steps: | 0, reward: 32, epsilon: 0.100000 |
| Episode: | 5, interaction_steps: | 0, reward: 33, epsilon: 0.100000 |
| Episode: | 6, interaction_steps: | 0, reward: 31, epsilon: 0.100000 |
| Episode: | 7, interaction_steps: | 0, reward: 32, epsilon: 0.100000 |
| Episode: | 8, interaction_steps: | 0, reward: 32, epsilon: 0.100000 |
| Episode: | 9, interaction_steps: | 0, reward: 31, epsilon: 0.100000 |

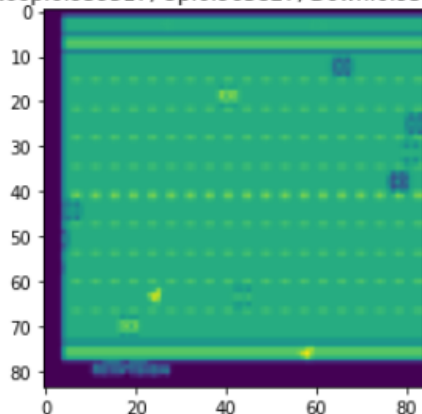
(5)

NOOP

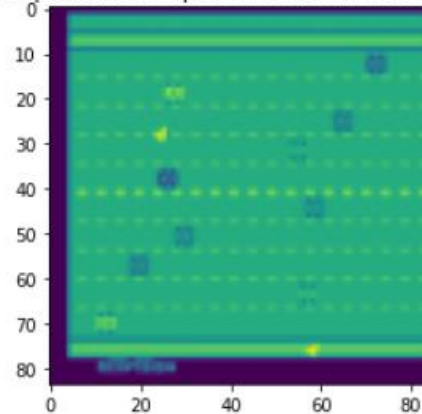


UP

Noop:0.959517, Up:0.963827, Down:0.957861

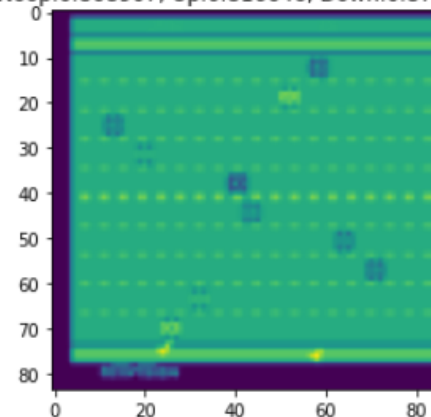


Noop:1.288916, Up:1.296872, Down:1.282673

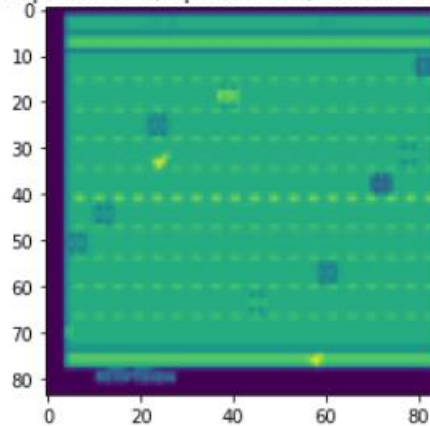


DOWN

Noop:0.868907, Up:0.816646, Down:0.871354



Noop:1.333297, Up:1.336198, Down:1.340450



1. 因為圖片為四張拼接在一起，所以我都選第四張的影像當作輸出，然後可能沒看到 RGB 在 code 的哪個位置，因此顏色出來很奇怪，但隱約可以看出形象，從上面的圖片看起來他做的選擇跟我會做的蠻像的，但因為程式是一次考慮四張，所以有些選擇可能會不一樣
2. 因為 Q-value 值會受 reward 和參數影響，因此若那時的 reward 越高那 Q-value 就會越大