Deep Learning Homework 3

0510894 電機 4D 翁紹恩

一 Generative adversarial network (GAN)

1 Data augmentation (Describe how you preprocess the dataset and explain why.)

a The customized dataset

自定義一個新的 dataset，由於一開始欲使用 google colab 作為使用環境但 google colab 無法一次讀入大量圖片，而整個 dataset 有 22 萬多張經過 align 的圖片，因此將所有圖片分成 34 個資料夾，每個資料夾約有 6000 張圖片，檔名分別為 1~34。Customized dataset 最重要的為儲存每一張圖片路徑及需要放入 transform 以處理資料。讀取照片方法使用 for loop 以讀取 34 個資料夾再分別 glob 路徑中資料，在__getitem__中則透過 index 再加上 root 形成完整路徑，利用 matplotlib.image 讀取檔案並將之轉換成 PIL 形式，並套入指定 transform，最後回傳圖片。Transform 將 image resize 成大小 64*64，並對數據 normalize 將之換成[-1, 1]，最後轉成 tensor。

b Dataloader

使用 torch.utils.data.DataLoader，將上面處理完的 dataset 及 batch size 放入，並 shuffle，最後可將 dataloader 放入 train。

2 Setup main

a 利用 customized dataset 和 torch.utils.data.DataLoader create dataloader，並將 generator 和 discriminator 送到 device 中，使用 Adam 當作 optimizer，learning rate = 0.0002，beta = (0.5, 0.999)，criterion 使用 BCELoss 並加入參數 reduction="mean"以將輸出除以輸入元素個數，讓 loss 曲線變得平滑以方便觀察。

3 Setup train

a 每個 epoch 會跑過所有的資料，設置兩個 label，real 為 1、fake 為 0。

b 訓練一個 batch 時，初始化 generator 和 discriminator 的 gradient 並將 data 放入 device

c 先將原始圖片放入 discriminator，並將相對應 label 設置為 1；產生 random noise 放入 generator，給予 generator output label 為 0 並丟入 discriminator，分別獲得的 loss 相加為實際 discriminator 的 loss，利用此 loss 做 adam optimize 訓練 discriminator；由於 noise generate 的 output result 放入新的 discriminator 的結果對於 generator 來說為真，因此將之對應 label 設為 1，更新 generator network，反覆訓練至收斂，在此設定 50 個 epochs。

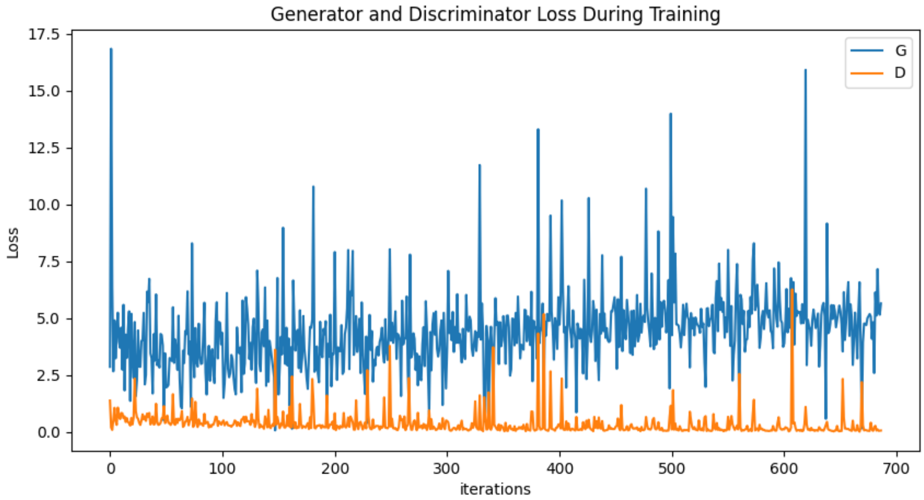
4 Setup Visualization

a 每迭代 500 次或是達到 dataloader、epoch 的最後一步則儲存 loss 的圖片及展示實際上的圖片和用 noise 產生出來的圖片。

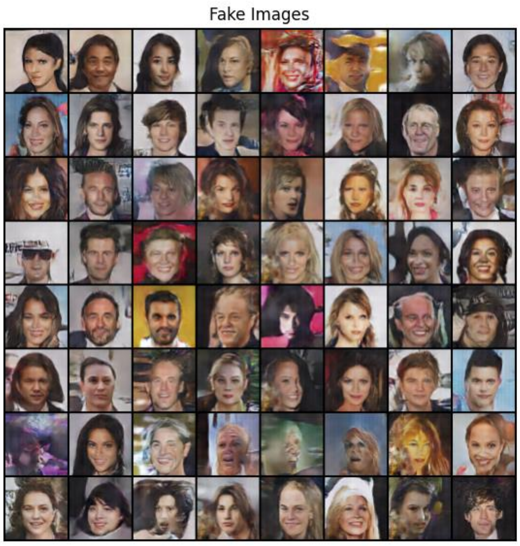
b 取出 dataloader 中其中一個 batch 的前 64 張圖片當作 real image，並拿取在 train 中儲存的 img_list 的最後一張圖為當下最新的生成圖片最為展示。

5 Plot the learning curves for both generator and discriminator, and draw some samples generated from your model.

a learning curves for both generator and discriminator



b samples generated from my model

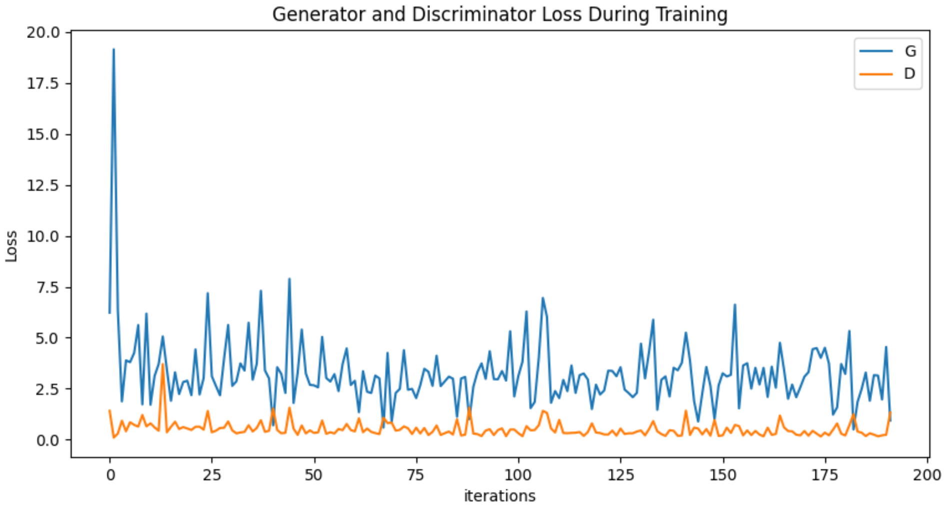


可以看到產生的圖片已經具備人形，但放大圖片看這細微的地方仍有需多地方需要加強，像是頭髮會對不上臉型等等。

6 The difficulties you face in this homework

a 這次的 GAN 訓練一次滿久的，原始只訓練五個 epochs 的結果十分不清晰，將原始 model 訓練至 21 個 epoch 時才可以達到上述結果。原始 5 個 epochs 結果如下：

若在 discriminator 的 conv2d 前加上 torch.nn.utils.spectual_norm 則在 epoch=5 時結果如下:



可以發現圖片有達到訓練了 21 個 epochs 的效果，訓練曲線也有收斂的傾向，這是依照[1]一文指出 spectral normalization 就是透過 Lipschitz constant 對每層的輸出做限縮，可以有效的解決 GAN 訓練不穩定的問題。

[1] Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

二 Deep Q Network (DQN)

1 Indicate the code paragraph about the updating from the given source code, explain the purpose of the following hyperparameters:

```
expected_state_action_values = (next_state_values * self.GAMMA) + reward_batch

loss = F.smooth_l1_loss(state_action_values, expected_state_action_values)
```

a updating step α

α 代表這一輪的 state 對比前一輪的更新程度

b discount factor γ

γ 代表的是給 reward 的影響，時間間隔越遠所給的 reward 影響越來越小，當下的 reward 最大

```
expected_state_action_values = (next_state_values * self.GAMMA) + reward_batch
```

c target network update period τ

τ 表示周期以更新 network 參數

```
    if self.update_count % args.target_step == 0:
        self.update_target_net()

def update_target_net(self):
    with torch.no_grad():
        self.target_net.load_state_dict(self.policy_net.state_dict())
```

d ε for ε-greedy policy

ε 是一個在[0, 1]之間的 threshold 決定是否可以使用 greedy algorithm。選擇動作時隨機選擇一個[0, 1]間的數值，在本次作業中，若是隨機選出的數值小於 ε ，則進行探索性的動作，即忽略 Q 隨機選取動作；反之，高於 ε 會執行 greedy algorithm。

```
def select_action(self, state):
    self.interaction_steps += 1
    self.epsilon = self.EPS_END + np.maximum( (self.EPS_START-self.EPS_END) * (1 - self.interaction_steps/self.EPS_DECAY), 0)
    if random.random() < self.epsilon:
        return torch.tensor([random.choices([0,1,2], weights = [3,6,1], k = 1)], device=device, dtype=torch.long)
    else:
        with torch.no_grad():
            return self.policy_net(state).max(1)[1].view(1, 1)
```

2 The total reward of sample episodes for changing the probability of random agent: [NOOP (0.3), UP (0.6), DOWN(0.1)]

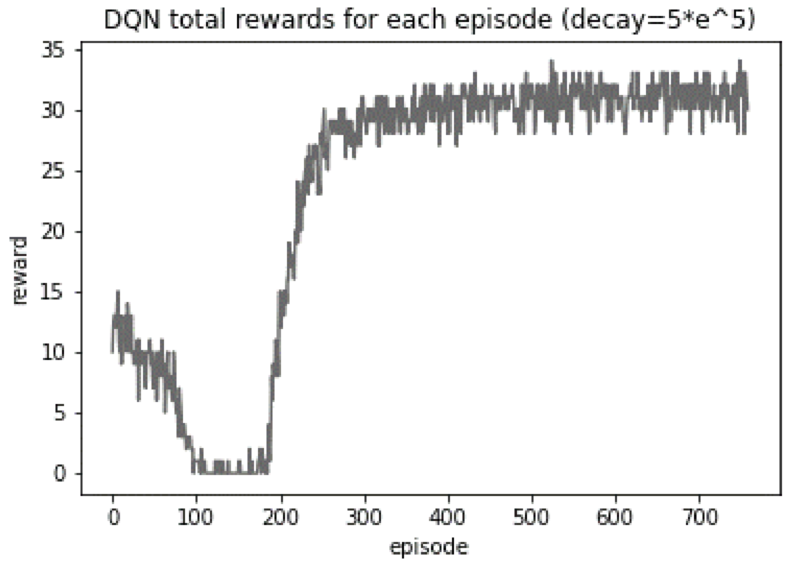
```
Evaluation: True, Episode:      0, Interaction_steps:   2048, evaluate reward: 23.200000
Episode:      1, interaction_steps:   4096, reward: 11, epsilon: 0.996314
Episode:      2, interaction_steps:   6144, reward:  9, epsilon: 0.994470
Episode:      3, interaction_steps:   8192, reward: 10, epsilon: 0.992627
Episode:      4, interaction_steps:  10240, reward: 12, epsilon: 0.990784
Episode:      5, interaction_steps:  12288, reward: 11, epsilon: 0.988941
Episode:      6, interaction_steps:  14336, reward: 12, epsilon: 0.987098
Episode:      7, interaction_steps:  16384, reward: 13, epsilon: 0.985254
Episode:      8, interaction_steps:  18432, reward: 10, epsilon: 0.983411
Episode:      9, interaction_steps:  20480, reward: 12, epsilon: 0.981568
Episode:     10, interaction_steps:  22528, reward:  9, epsilon: 0.979725
Evaluation: True, Episode:     10, Interaction_steps:  22528, evaluate reward: 22.200000
Episode:     11, interaction_steps:  24576, reward: 12, epsilon: 0.977882
Episode:     12, interaction_steps:  26624, reward: 13, epsilon: 0.976038
Episode:     13, interaction_steps:  28672, reward: 11, epsilon: 0.974195
Episode:     14, interaction_steps:  30720, reward: 10, epsilon: 0.972352
Episode:     15, interaction_steps:  32768, reward: 12, epsilon: 0.970509
Episode:     16, interaction_steps:  34816, reward:  7, epsilon: 0.968666
Episode:     17, interaction_steps:  36864, reward: 12, epsilon: 0.966822
Episode:     18, interaction_steps:  38912, reward: 11, epsilon: 0.964979
Episode:     19, interaction_steps:  40960, reward: 12, epsilon: 0.963136
Episode:     20, interaction_steps:  43008, reward: 11, epsilon: 0.961293
```

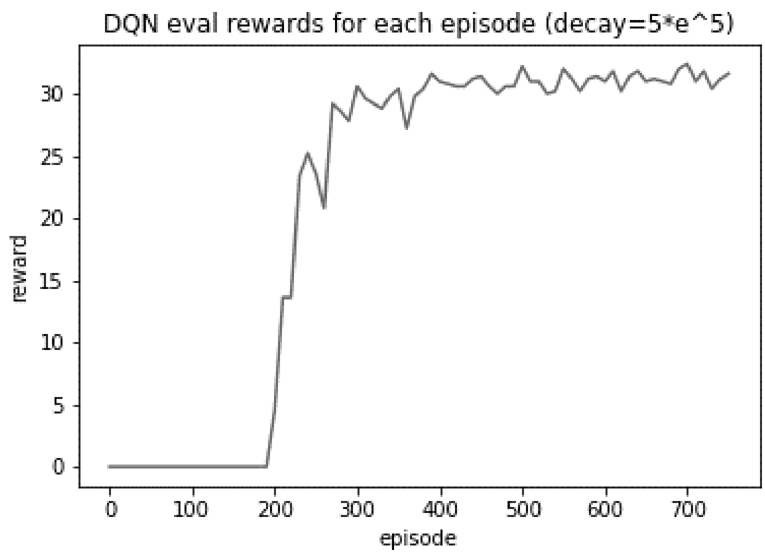
```
Evaluation: True, Episode:    460, Interaction_steps: 944128, evaluate reward: 31.600000
Episode:    461, interaction_steps: 946176, reward: 31, epsilon: 0.148442
Episode:    462, interaction_steps: 948224, reward: 30, epsilon: 0.146598
Episode:    463, interaction_steps: 950272, reward: 31, epsilon: 0.144755
Episode:    464, interaction_steps: 952320, reward: 31, epsilon: 0.142912
Episode:    465, interaction_steps: 954368, reward: 27, epsilon: 0.141069
Episode:    466, interaction_steps: 956416, reward: 31, epsilon: 0.139226
Episode:    467, interaction_steps: 958464, reward: 30, epsilon: 0.137382
Episode:    468, interaction_steps: 960512, reward: 30, epsilon: 0.135539
Episode:    469, interaction_steps: 962560, reward: 30, epsilon: 0.133696
Episode:    470, interaction_steps: 964608, reward: 31, epsilon: 0.131853
Evaluation: True, Episode:    470, Interaction_steps: 964608, evaluate reward: 30.200000
Episode:    471, interaction_steps: 966656, reward: 31, epsilon: 0.130010
Episode:    472, interaction_steps: 968704, reward: 31, epsilon: 0.128166
Episode:    473, interaction_steps: 970752, reward: 33, epsilon: 0.126323
Episode:    474, interaction_steps: 972800, reward: 30, epsilon: 0.124480
Episode:    475, interaction_steps: 974848, reward: 31, epsilon: 0.122637
Episode:    476, interaction_steps: 976896, reward: 30, epsilon: 0.120794
Episode:    477, interaction_steps: 978944, reward: 31, epsilon: 0.118950
Episode:    478, interaction_steps: 980992, reward: 30, epsilon: 0.117107
Episode:    479, interaction_steps: 983040, reward: 30, epsilon: 0.115264
Episode:    480, interaction_steps: 985088, reward: 32, epsilon: 0.113421
```

可以看到和一般隨便 random 數值比起來，一開始 training reward 即可達到 10 左右，並在大約 500 epochs 即差不多收斂，反之，隨便 random 大約要在 700、800 epochs 才會收斂。

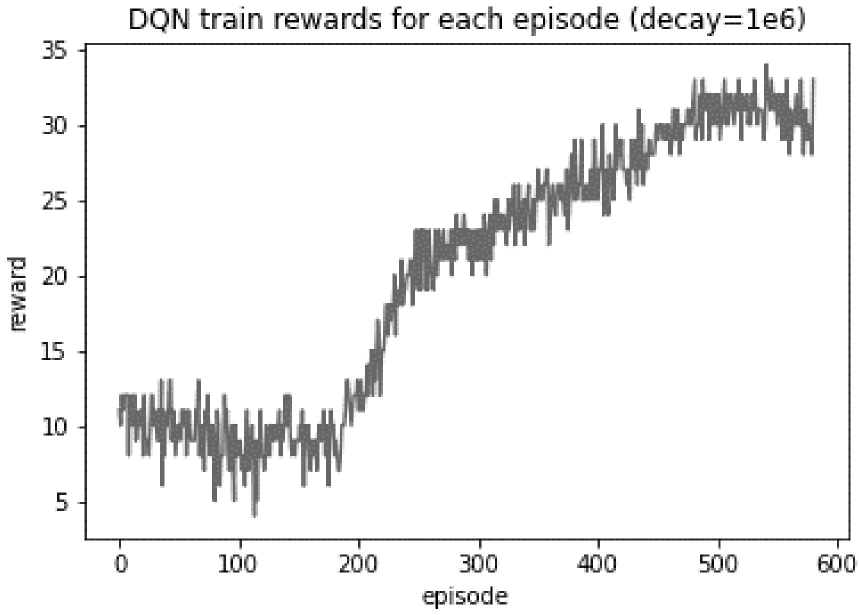
3 Plot the episode reward in learning time and evaluation time (ε = final epsilon) (2 charts). Show your configuration and discuss what you find in training phase.(沒有特別指出的即為原本設置參數)

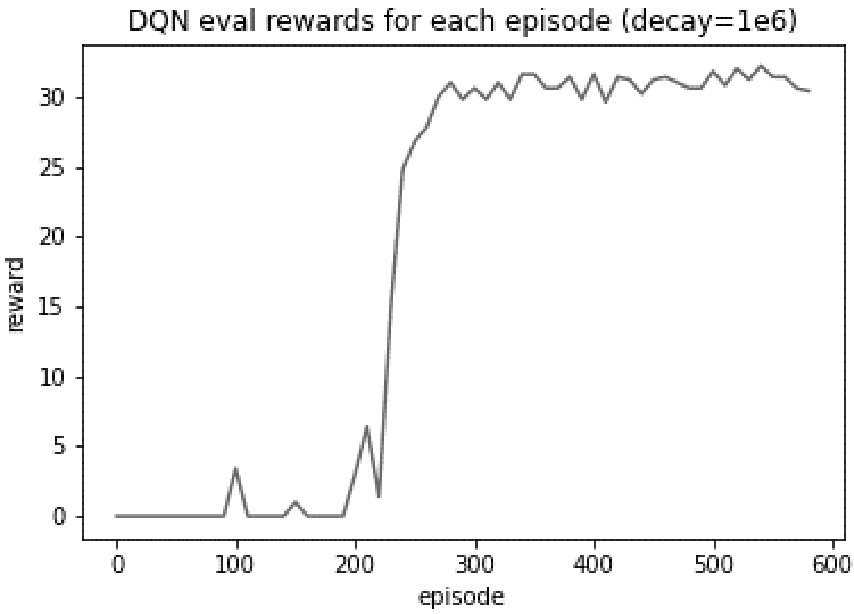
a epsilon_decay = 500000



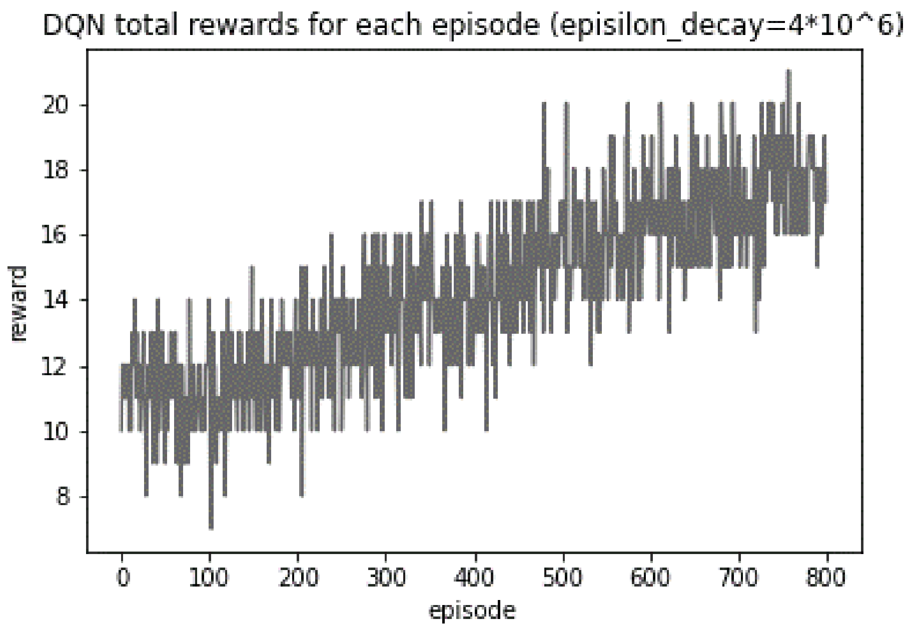


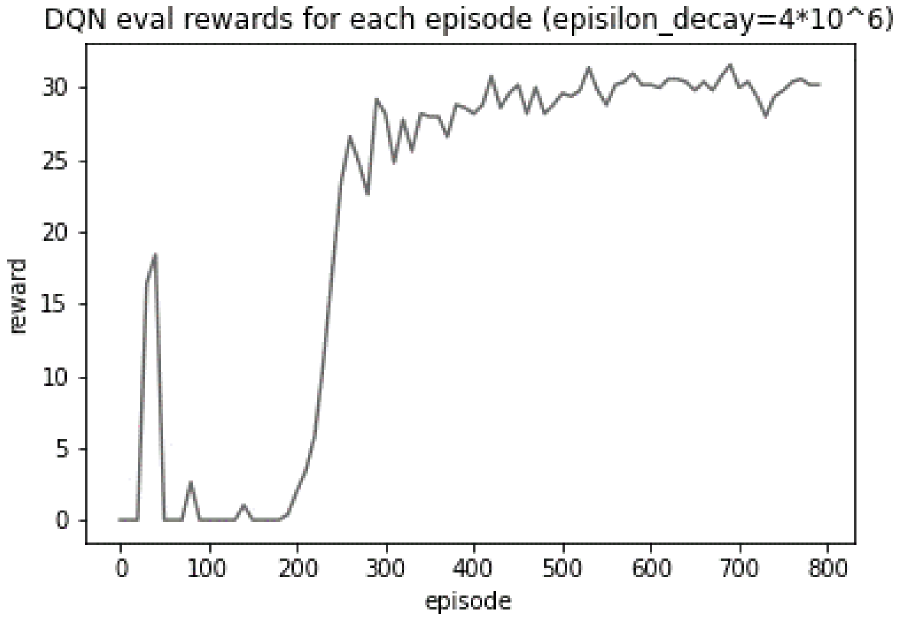
b epsilon_decay = 1000000



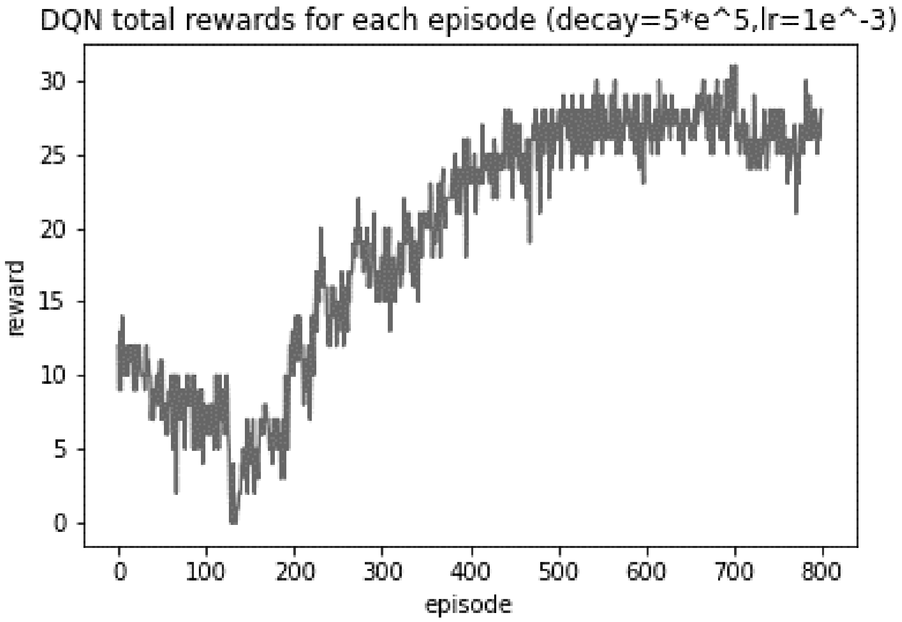


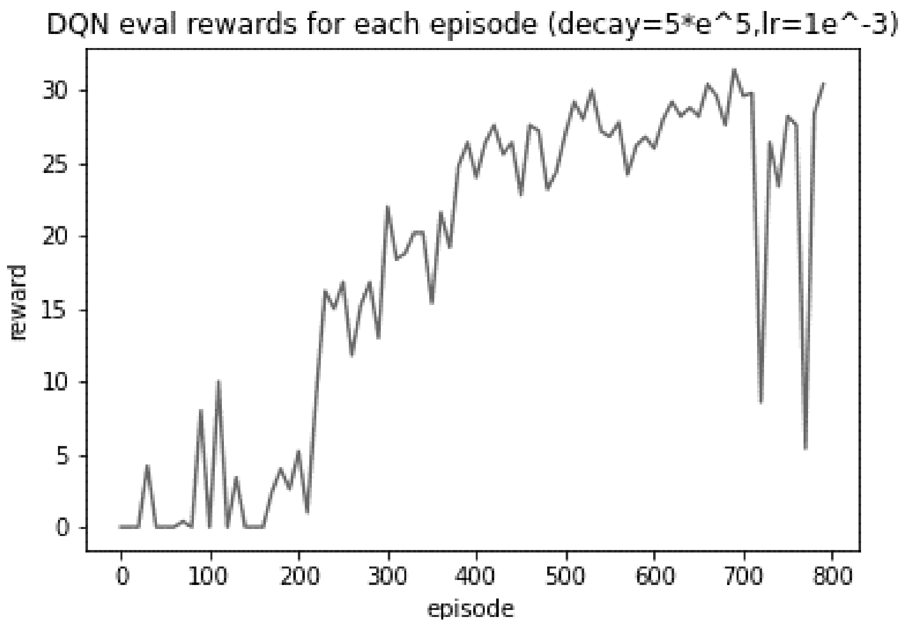
c epsilon_decay = 4000000



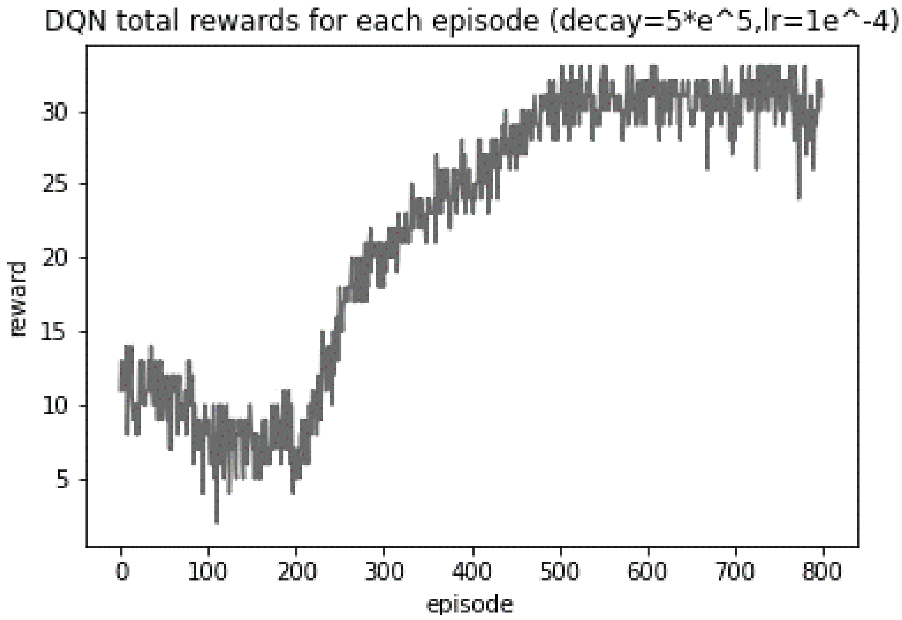


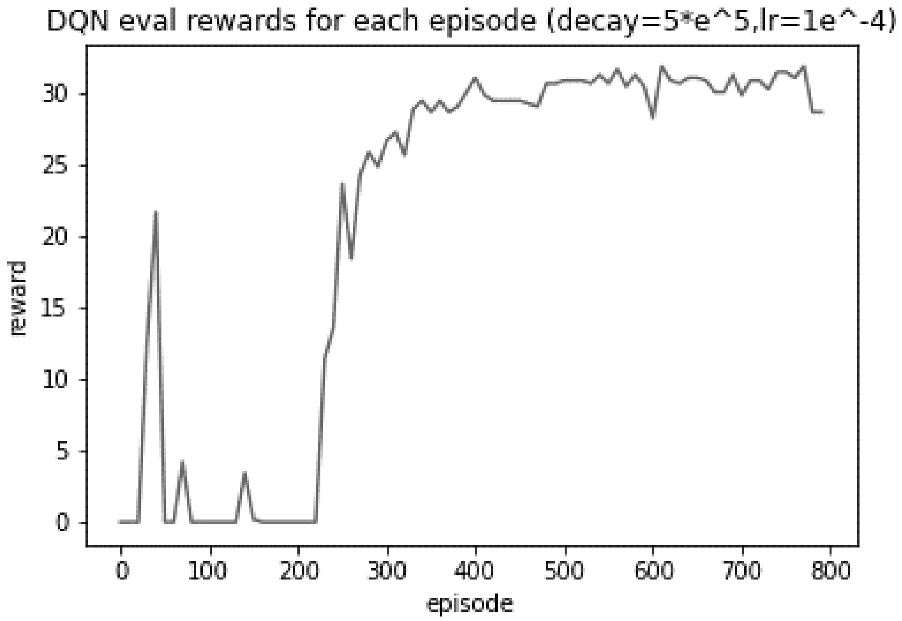
d epsilon_decay = 500000, learning rate = 0.001



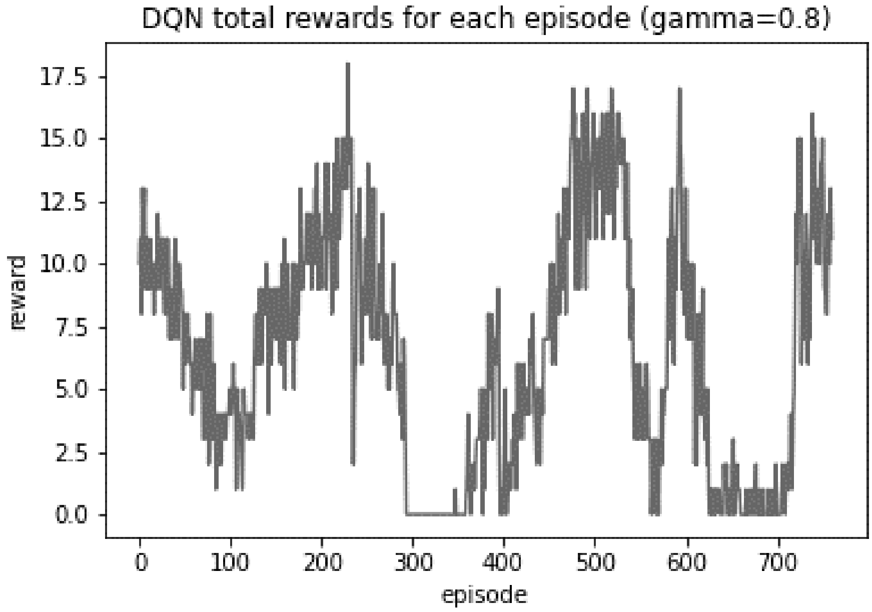


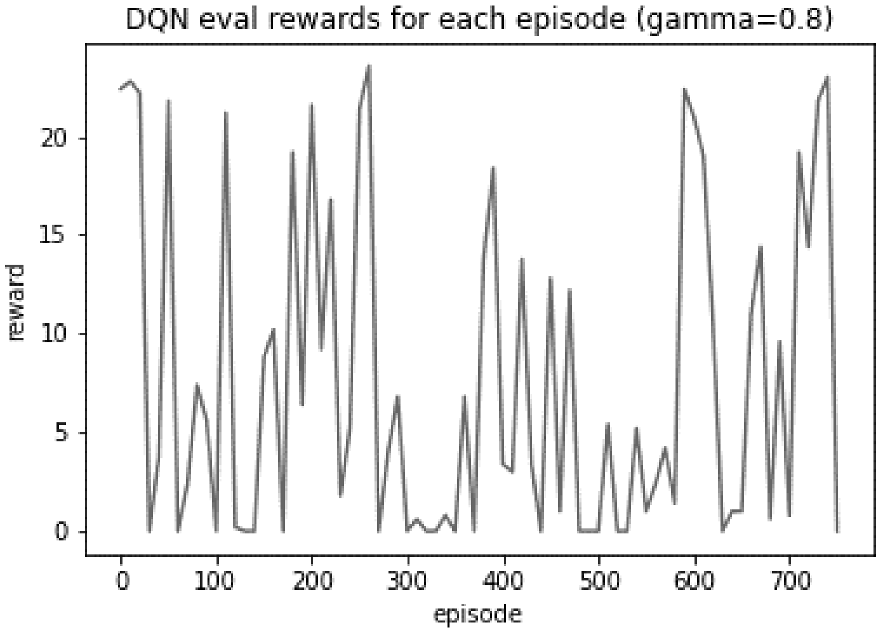
e epsilon_decay = 500000, learning rate = 0.0001



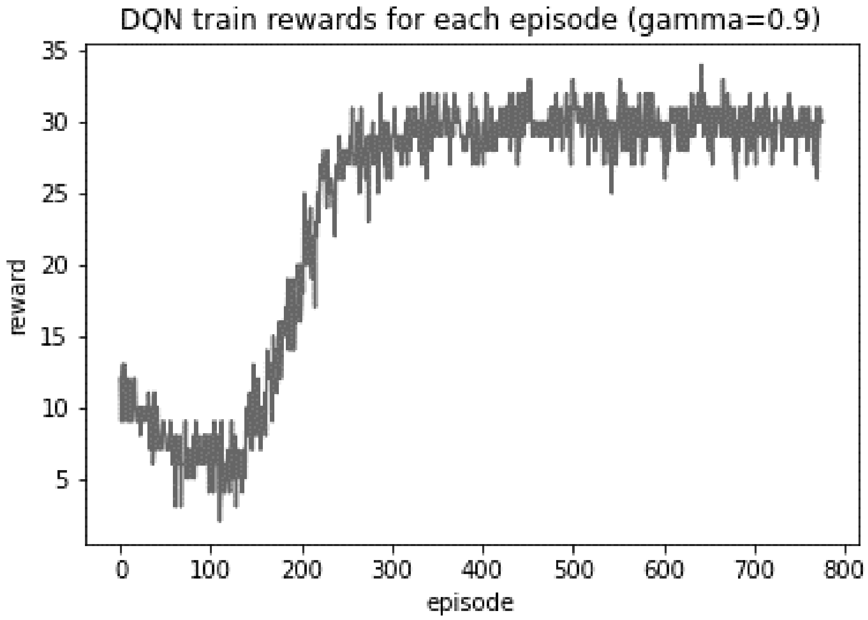


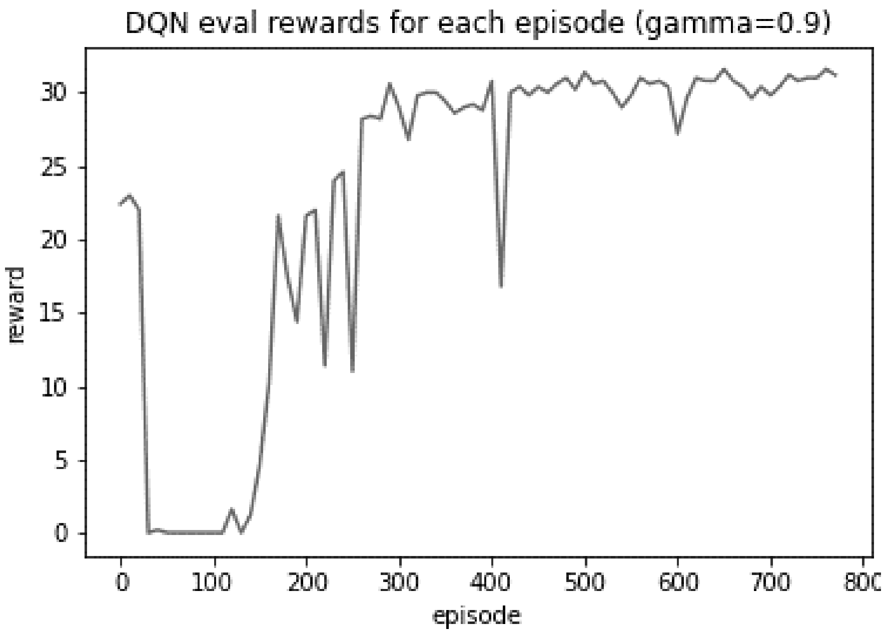
f gamma = 0.8




DQN eval rewards for each episode (gamma=0.8)
reward
episode

g gamma = 0.9


DQN train rewards for each episode (gamma=0.9)
reward
episode


DQN eval rewards for each episode (gamma=0.9)
reward
episode

根據以上結果可以發現當 epsilon decay 越大時，epsilon 下降的速度越慢，反而 eval 時使用的是 epsilon=0.1，所以會形成一個 eval 時 reward 已經達到 30 幾且有收斂傾向但 training reward 反而只有 20 幾的現象，例如： c 。並且 epsilon decay 越小時訓練速度越快；而當 gamma = 0.8 時會有無法順利收斂的情形，因為 gamma 是時間數據給的影響，因此需要一定以上的數值，例如 0.9 才能正常收斂；至於 learning rate 需要小一點才比較穩定。

4. After training, you will obtain the model parameters for the agent. Show total reward in some episodes for deep Q-network agent.

   a. epsilon_decay = 500000

```
[Info] Restore model from './decay5e5/q_target_checkpoint_1538048.pth' !
Episode:      0, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      1, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      2, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      3, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      4, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      5, interaction_steps:      0, reward: 31, epsilon: 0.100000
Episode:      6, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      7, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      8, interaction_steps:      0, reward: 31, epsilon: 0.100000
Episode:      9, interaction_steps:      0, reward: 30, epsilon: 0.100000
```

   b. epsilon_decay = 1000000

```
[Info] Restore model from './decay1e6/q_target_checkpoint_1128448.pth' !
Episode:      0, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      1, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      2, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      3, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      4, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      5, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      6, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      7, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      8, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      9, interaction_steps:      0, reward: 31, epsilon: 0.100000
```

   c. epsilon_decay = 4000000

```
[Info] Restore model from './decay4e6/q_target_checkpoint_1538048.pth' !
Episode:      0, interaction_steps:      0, reward: 29, epsilon: 0.100000
Episode:      1, interaction_steps:      0, reward: 31, epsilon: 0.100000
Episode:      2, interaction_steps:      0, reward: 29, epsilon: 0.100000
Episode:      3, interaction_steps:      0, reward: 28, epsilon: 0.100000
Episode:      4, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      5, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      6, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      7, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      8, interaction_steps:      0, reward: 31, epsilon: 0.100000
Episode:      9, interaction_steps:      0, reward: 32, epsilon: 0.100000
```

   d. epsilon_decay = 500000, learning rate = 0.001

```
[Info] Restore model from './DQN_lr1e-3/q_target_checkpoint_1538048.pth' !
Episode:      0, interaction_steps:      0, reward: 29, epsilon: 0.100000
Episode:      1, interaction_steps:      0, reward: 29, epsilon: 0.100000
Episode:      2, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      3, interaction_steps:      0, reward: 28, epsilon: 0.100000
Episode:      4, interaction_steps:      0, reward: 31, epsilon: 0.100000
Episode:      5, interaction_steps:      0, reward: 29, epsilon: 0.100000
Episode:      6, interaction_steps:      0, reward: 26, epsilon: 0.100000
Episode:      7, interaction_steps:      0, reward: 29, epsilon: 0.100000
Episode:      8, interaction_steps:      0, reward: 28, epsilon: 0.100000
Episode:      9, interaction_steps:      0, reward: 31, epsilon: 0.100000
```

   e. epsilon_decay = 500000, learning rate = 0.0001

```
[Info] Restore model from './lr1e4/q_target_checkpoint_1538048.pth' !
Episode:      0, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      1, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      2, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      3, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      4, interaction_steps:      0, reward: 31, epsilon: 0.100000
Episode:      5, interaction_steps:      0, reward: 29, epsilon: 0.100000
Episode:      6, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      7, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      8, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      9, interaction_steps:      0, reward: 33, epsilon: 0.100000
```

f gamma = 0.8

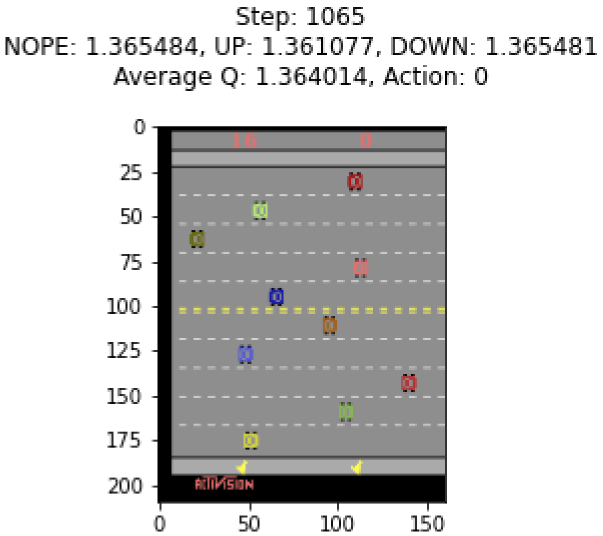
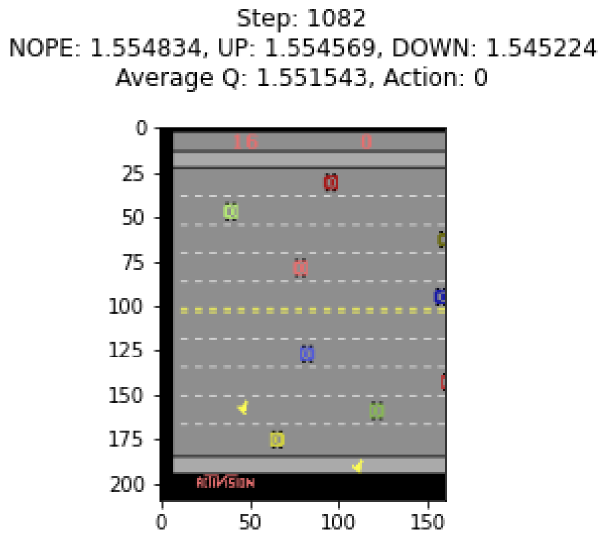
```
[Info] Restore model from './gamma08/q_target_checkpoint_1538048.pth' !
Episode:      0, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      1, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      2, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      3, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      4, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      5, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      6, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      7, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      8, interaction_steps:      0, reward:  0, epsilon: 0.100000
Episode:      9, interaction_steps:      0, reward:  0, epsilon: 0.100000
```

g gamma = 0.9

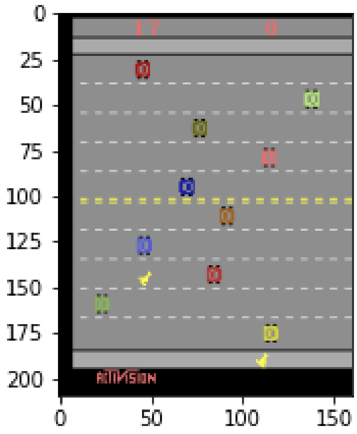
```
[Info] Restore model from './gamma09/q_target_checkpoint_1538048.pth' !
Episode:      0, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      1, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      2, interaction_steps:      0, reward: 33, epsilon: 0.100000
Episode:      3, interaction_steps:      0, reward: 30, epsilon: 0.100000
Episode:      4, interaction_steps:      0, reward: 32, epsilon: 0.100000
Episode:      5, interaction_steps:      0, reward: 29, 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: 33, epsilon: 0.100000
```

可以看到除了 gamma=0.8 外，其餘的在 epsilon=0.1 都幾乎有達到 reward=30

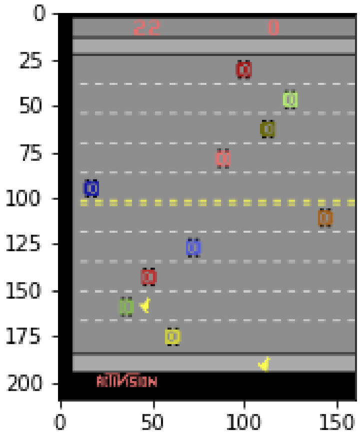
5 Sample some states, show the Q values for each action, analyze the results, and answer

<table>
<tr><th colspan="2">NOOP(lower side)</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Step: 1144
NOPE: 1.590589, UP: 1.587643, DOWN: 1.587901
Average Q: 1.588711, Action: 0

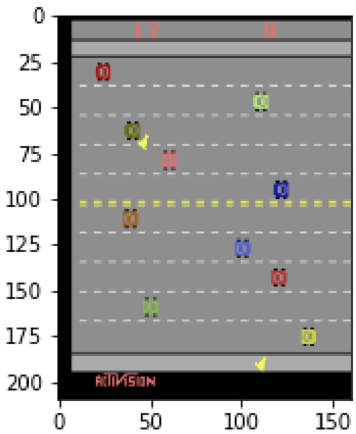


Step: 1477
NOPE: 1.494347, UP: 1.491878, DOWN: 1.487549
Average Q: 1.491258, Action: 0

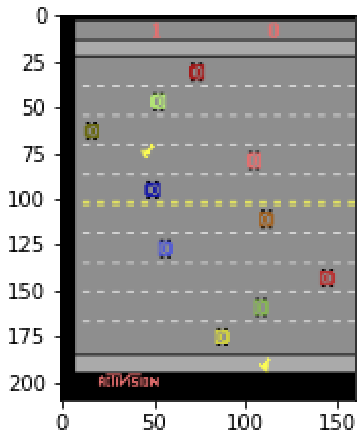


NOOP(upper side)

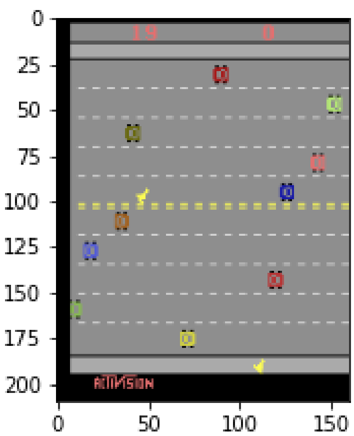
Step: 1171
NOPE: 1.992428, UP: 1.974762, DOWN: 1.918598
Average Q: 1.961929, Action: 0



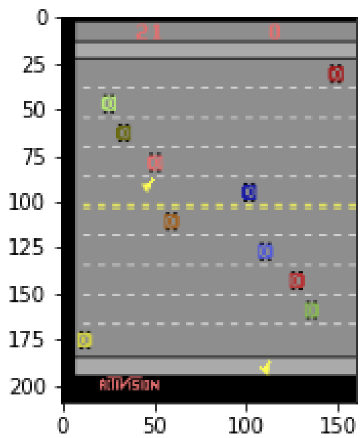
Step: 0109
NOPE: 1.890743, UP: 1.885659, DOWN: 1.884539
Average Q: 1.886981, Action: 0

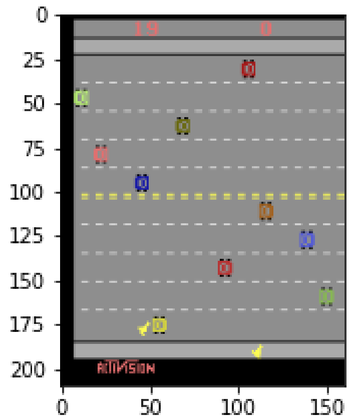
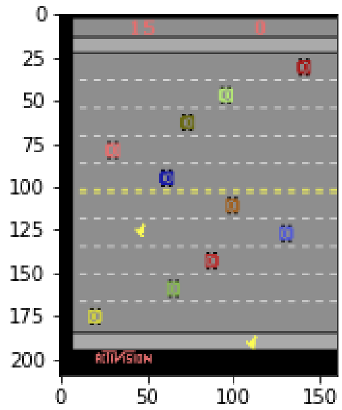
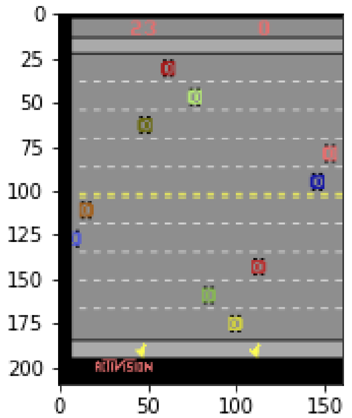
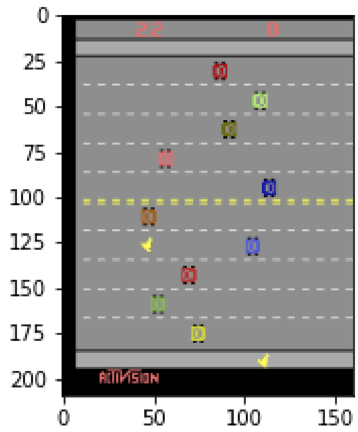


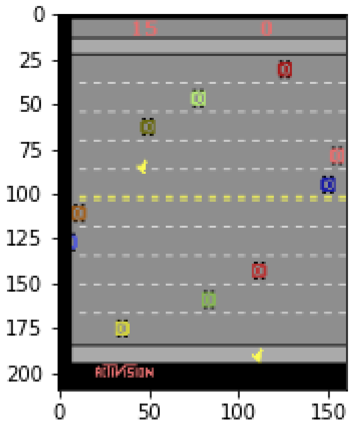
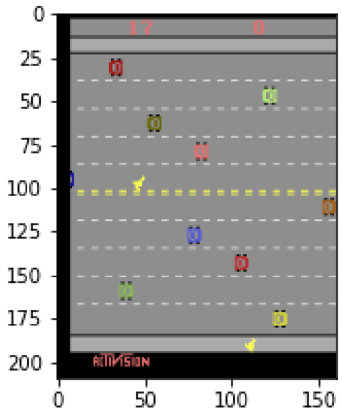
Step: 1290
NOPE: 1.875576, UP: 1.875026, DOWN: 1.869004
Average Q: 1.873202, Action: 0

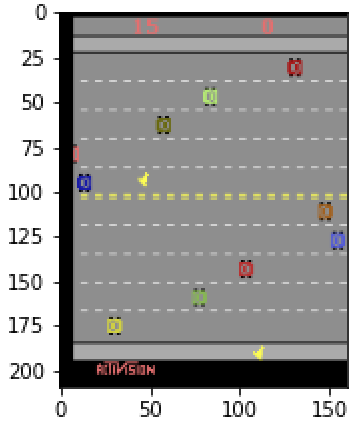
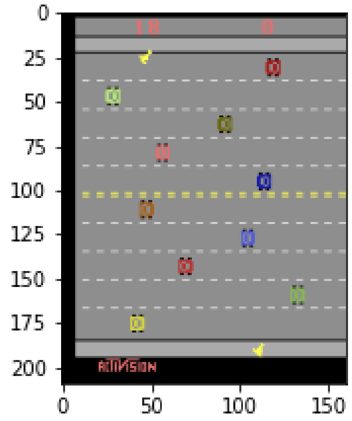


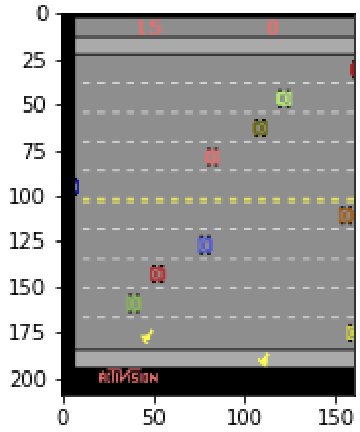
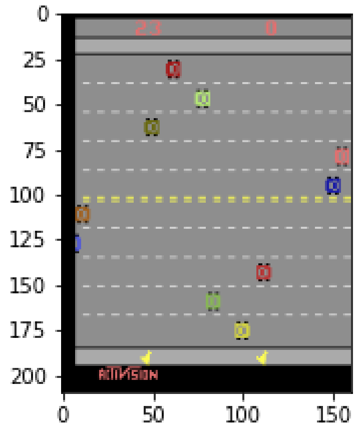
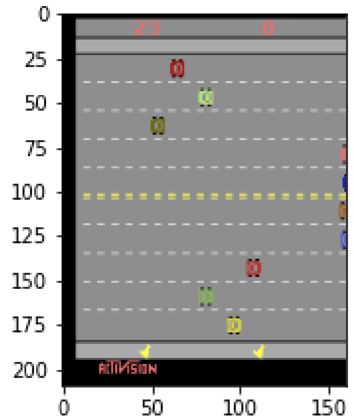
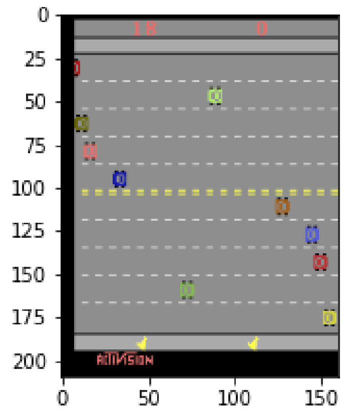
Step: 1416
NOPE: 1.731435, UP: 1.697172, DOWN: 1.726002
Average Q: 1.718203, Action: 0



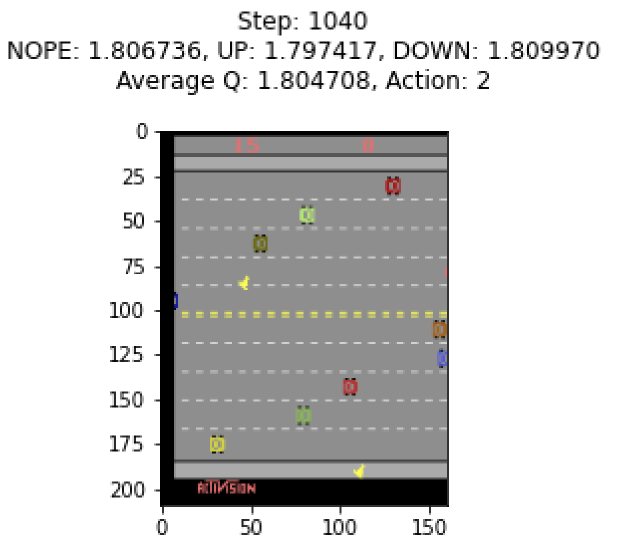
UPPER(lower side)	
Step: 1270 NOPE: 1.465109, UP: 1.473536, DOWN: 1.457437 Average Q: 1.465361, Action: 1 	Step: 1026 NOPE: 1.619185, UP: 1.635577, DOWN: 1.606043 Average Q: 1.620268, Action: 1 
Step: 1525 NOPE: 1.399002, UP: 1.399967, DOWN: 1.396841 Average Q: 1.398603, Action: 1 	Step: 1493 NOPE: 1.707941, UP: 1.717828, DOWN: 1.689856 Average Q: 1.705209, Action: 1 

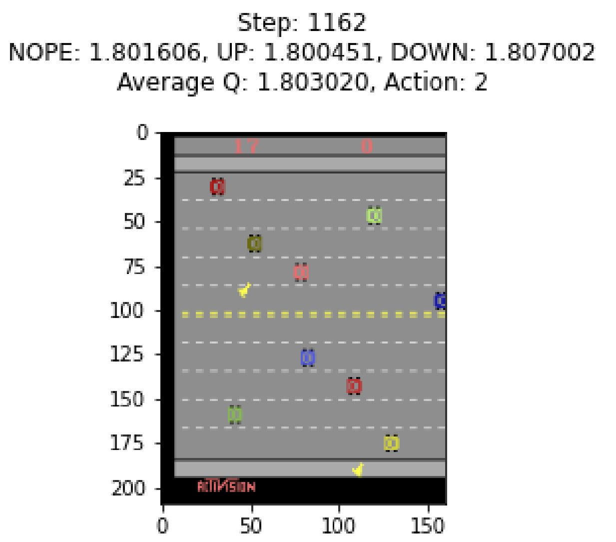
UPPER(upper side)	
Step: 1044 NOPE: 1.906360, UP: 1.909484, DOWN: 1.905280 Average Q: 1.907042, Action: 1 	Step: 1160 NOPE: 1.780041, UP: 1.783939, DOWN: 1.774823 Average Q: 1.779601, Action: 1 

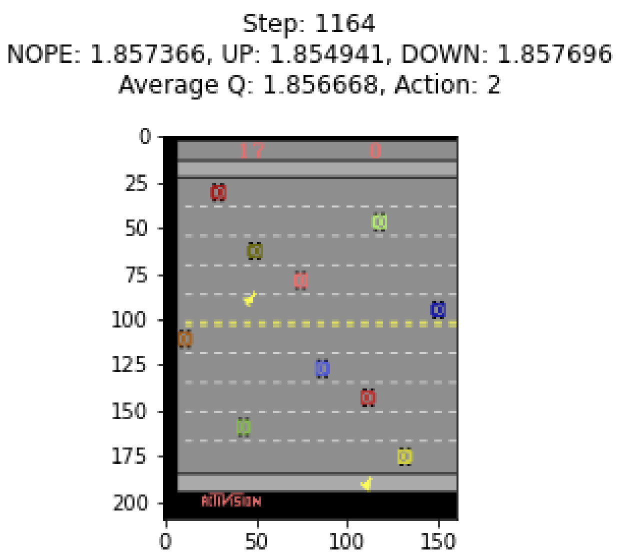
Step: 1038 NOPE: 1.768048, UP: 1.771819, DOWN: 1.761009 Average Q: 1.766959, Action: 1 	Step: 1253 NOPE: 2.310244, UP: 2.320221, DOWN: 2.289747 Average Q: 2.306737, Action: 1 

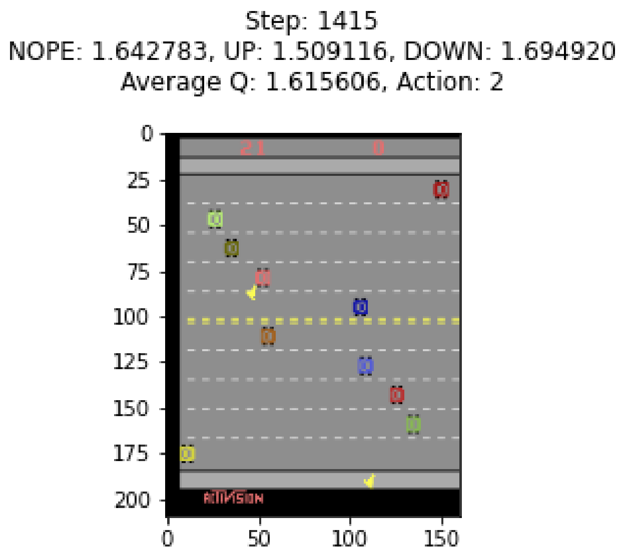
DOWN(lower side)	
Step: 1000 NOPE: 1.320663, UP: 1.316234, DOWN: 1.322482 Average Q: 1.319793, Action: 2 	Step: 1524 NOPE: 1.387036, UP: 1.386472, DOWN: 1.387311 Average Q: 1.386939, Action: 2 
Step: 1521 NOPE: 1.378362, UP: 1.374818, DOWN: 1.378502 Average Q: 1.377227, Action: 2 	Step: 1193 NOPE: 1.366366, UP: 1.365475, DOWN: 1.368232 Average Q: 1.366691, Action: 2 

DOWN(upper side)









a　Is DQN decision in the game the same as yours? Any good or bad move?

經過觀察後我認為使用 DQN 大部分的移動方式都比我會下的決定還要好，因為有許多 step 是我可能會判定會撞到因而不動或多次往下移動，但 DQN 多次選擇上移且都有順利通過，且實際玩遊戲時會因為一些緊張等等的心理因素而操作失誤，但 DQN 不帶任何情緒操作，所做的判斷幾乎是最理智的決定。

b　Why the averaged Q-value of three actions in some state is larger or less than those of the other states?

由以上結果可以發現，當要橫越上半部馬路時 Q-value 都比較大，因為接近終點 target，因此可以獲得的 reward 較高。

雲端硬碟 Q1、Q2 checkpoint 連結：(因為使用多個帳號訓練因此僅附上最終結果)

https://drive.google.com/drive/folders/1dC2ccq5jcHLESN05Kyn8Esg7zJasPwUj?usp=sharing