ADLxMLDS2017 HW3 report

姓名：徐有慶
學號：R05922162

1. Model description

- **Policy Gradient**
  - **Hyperparameters**

Gamma = 0.99

  - **Model Architecture**

Input image = (80, 80, 1)

Conv2d, out_channels=16, kernel_size=[8,8], stride=4, activation=relu

Conv2d, out_channels=32, kernel_size=[4,4], stride=2, activation=relu

Dense, neurons=128, activation=tanh

Dense, neurons=actions number, activation=softmax

  - **Optimizer**

RMSprop, lr=1e-4, decay=0.9

  - **Comment**

原先有 6 個 actions number，把它降為 3 個 actions number(左、右和停止)。丟入 model 的 observation 會先經過助教提供的前處理方法，轉成(80, 80, 1)的圖片。助教提供的三個 tips 也都有用上

- **Deep Q Learning**
  - **Hyperparameters**

Environment step: 沒限制，直到它 train 起來

Experience replay size: 10000

Learning start step: 10000

Target network update frequency: 1000

Online network update frequency: 4

Gamma: 0.99

Batch size: 32

Exploration rate: 1 to 0.05 in the first 1e6 env step

■ **Model Architecture**

Conv2d, out_channels=32, kernel_size=[8,8], stride=4, activation=relu

Conv2d, out_channels=64, kernel_size=[4,4], stride=2, activation=relu

Conv2d, out_channels=64, kernel_size=[3,3] stride=1, activation=relu

Dense, neurons=512, activation=leaky_relu

Dense, neurons=actions number, activation=leaky_relu

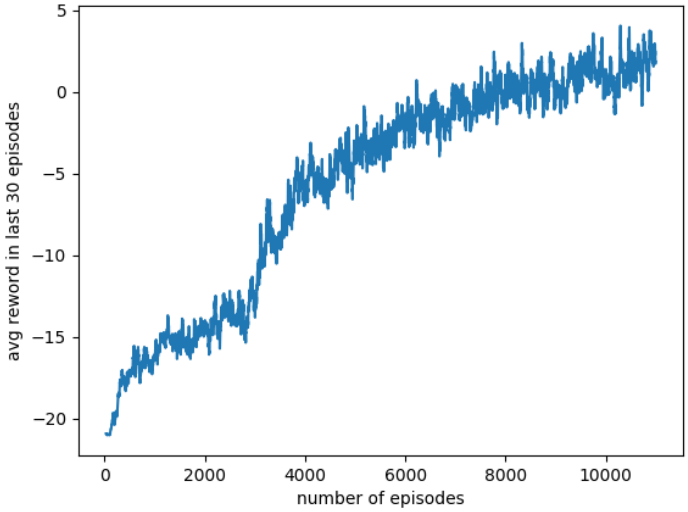
■ **Optimizer**

RMSprop, lr=1e-4, decay=0.99

■ **Comment**

和 PG 一樣，原先有 4 個 action numbers，把它降為只有 2 個 action numbers

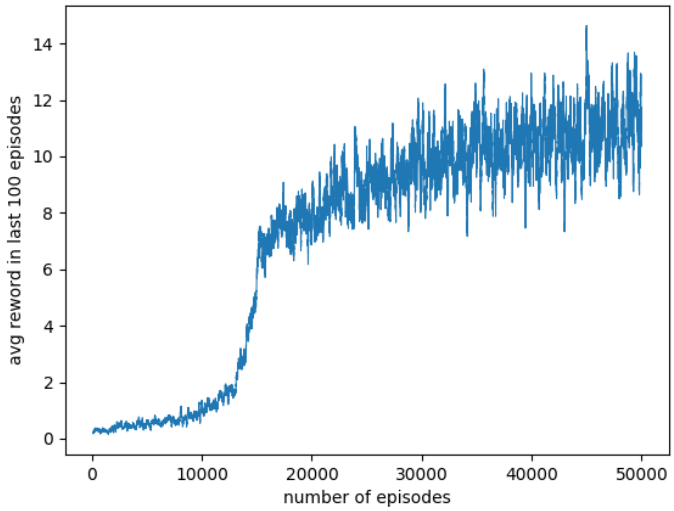
2. Learning curve of Policy Gradient on Pong



X 軸為第幾個 episodes

Y 軸為當前 episode 的前 30 個 episodes 的平均 reward

3. Learning curve of Deep Q Learning on Breakout



X 軸為當第幾個 episodes

Y 軸為當前 episode 的前 100 個 episodes 的平均 reward

4. Experimenting with DQN hyperparameters

以 model description 中描述的參數及架構當作 base，再對其他的參數做調整，總共做了三個不同的實驗

- **Test1**

將 Experience replay size 及 Learning start step 降低為 5000，看到有人說調大了 Experience replay size 就 train 起來了，所以想試試看在 batch size 不變的情況下，Experience replay size 對於 training 會不會有影響，而 Learning start step 感覺是跟 Experience replay size 綁定的，所以一起降低

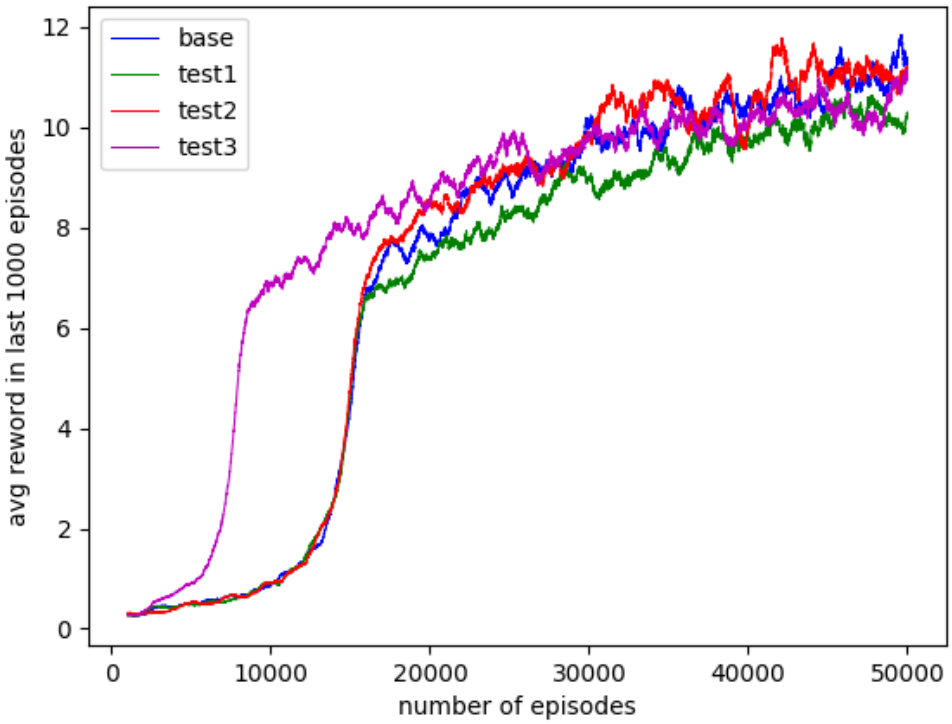
- **Test2**

Target network update frequency 調到 3000，target network 的作用就是算出 Q target，當它更新頻率下降時，也就表示和 online network 算出的 Q eavl 差距會越來越大，因為裡面存的參數是比較久遠的，想試試看這樣會不會降低收斂的速度

- **Test3**

調整 exploration rate 降低的速度，Exploration rate: 1 to 0.05 in the first 500000 env step，exploration rate 降低的速度變快，代表的就是更早讓 random 的機率降低，提早讓每次動作是根據 online network 出來的 Q 值做決定，想看看這樣子的收斂速度會不會快一點

- **Learning Curve**



X 軸一樣是第幾個 episode

Y 軸更改為當前 episode 的前 1000 個 episodes 的平均 reward

這樣會讓圖形看得比較清楚一點，如果只取前 100 個的話，圖形會震盪的很嚴重，而每條線幾乎都會疊在一起

- **Analysis**

  由 learning curve 來分析的話

  - **Test1**

    memory size 確實會影響到 training 出來的結果，雖然在前 15000 個 episodes 看起來差異不大，但隨著時間推進，較大的 memory size 可以得到較佳的 average reward

  - **Test2**

    Target network update frequency 的影響看起來是時好時壞，感覺如果再將 target network update frequency 的值調高一點可能比較能看出差異，如：5000, 1000

  - **Test3**

    調整 exploration rate 降低的速度，確實能讓收斂速度快一點，但隨著時間的推進，好像也不會比原先的結果還要佳