MLDS hw4 Reinforcement Learning

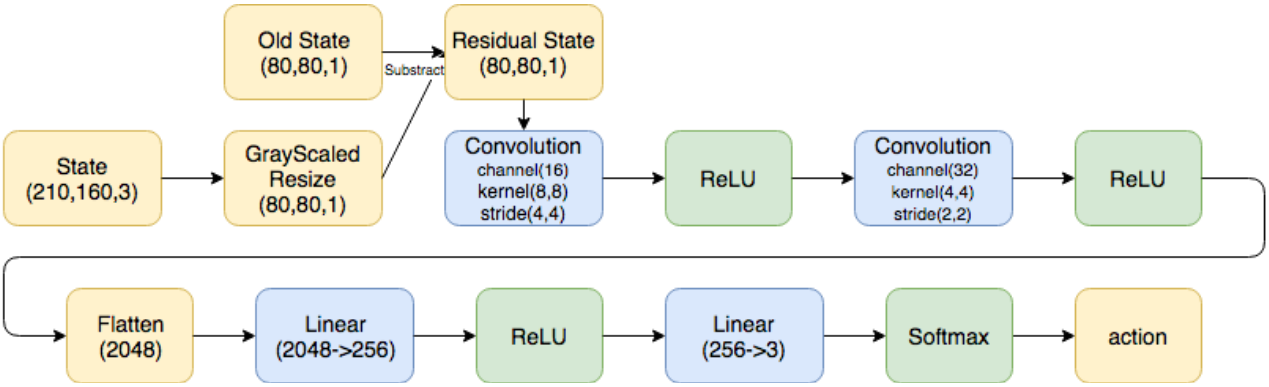
b03902101 資工四 楊力權 40% 4-1 4-2 4-3
b03902093 資工四 張庭維 30% 4-1
b03902102 資工四 廖廷浩 30% 4-2

4-1 Policy Gradient

一、Model description

(a.) 架構

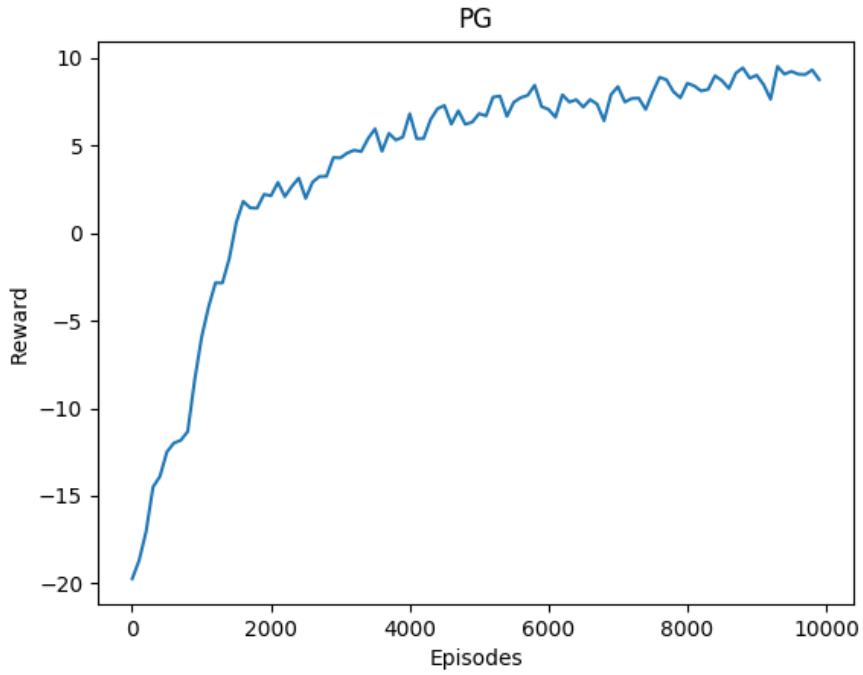
- 先將3個channel的state做灰階與resize。
- 把每個時間點的state減去上個時間點的state得到Residual State。
- 將Residual State餵作CNN input並經過output flatten後，再過hidden layer與action space layer最後過softmax輸出至action space。



(b.) 訓練細節

- Optimizer : RMSprop
- Learning Rate : 0.0001
- Gamma : 0.99

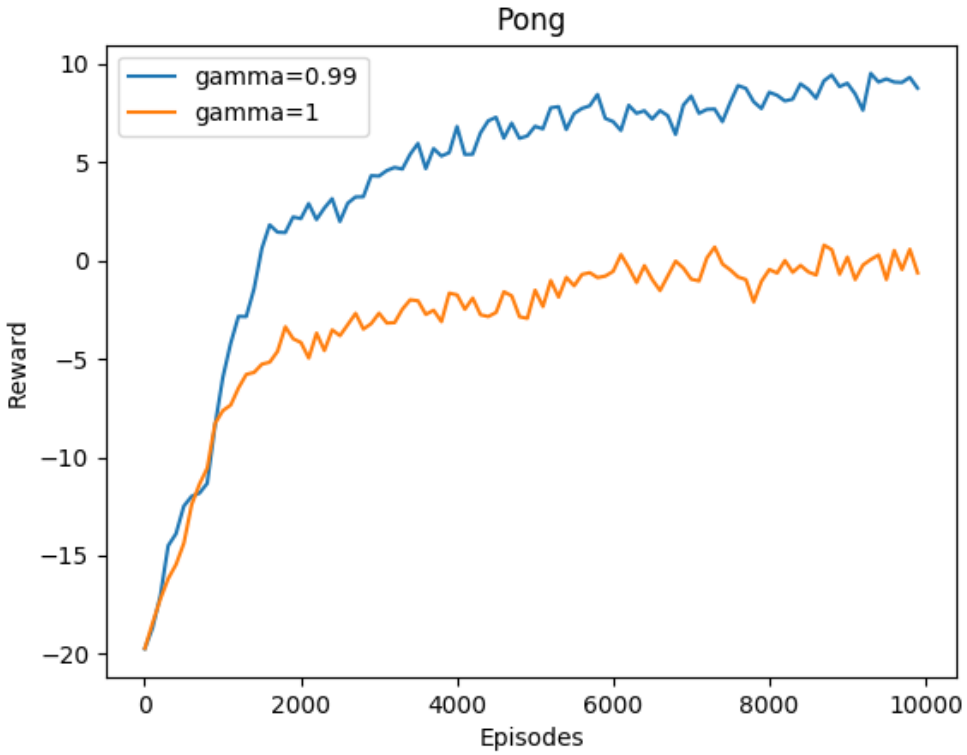
二、Learning Curve



三、Variance Reduction with Discount

在做Policy Gradient時，我們會把每個action後的reward情況全部考慮，而action前的reward不列入考慮，這是基本的reduce variance。但真實情況是往往離action愈遠的reward，跟action關係已經沒那麼大，因此我們在計算每個action的reward時，可以把後面累加的reward乘上一個折扣gamma，所以對每個action而言，之後每個action點的reward都要乘上折扣gamma加上採取該action的reward，得到total reward。

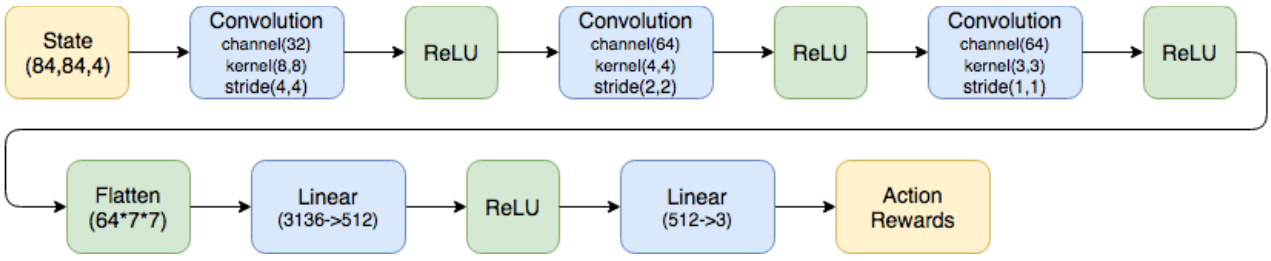
我們試了不做Discount也就是gamma=1，以及做了Discount也就是gamma=0.99時的比，可知道做discount的結果明顯好出很多。



4-2 Deep Q-Learning

一、Model description

(a.) 架構

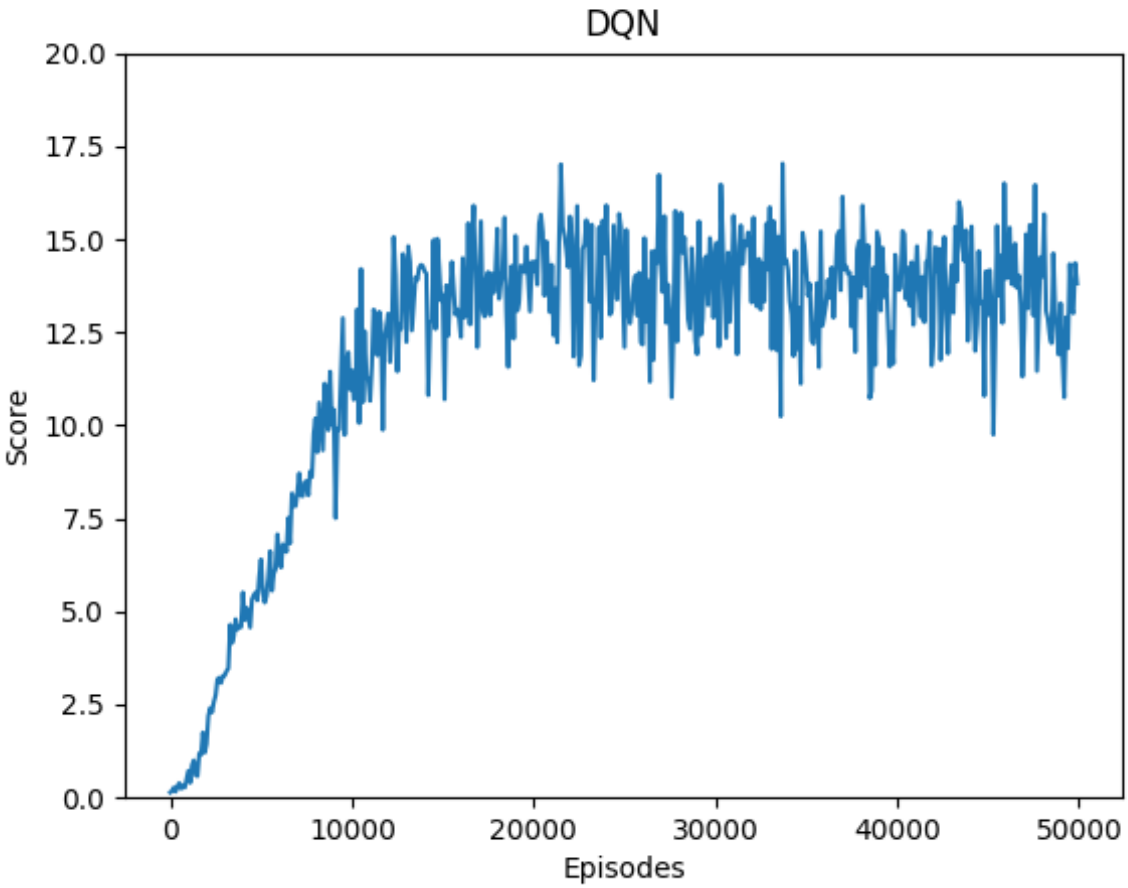


(b.) 訓練細節

- Optimizer : RMSprop
- Learning Rate : 0.00015

- Batch Size : 32
- Gamma : 0.99
- Memory Size : 10000
- Epsilon : 由1.0線性遞減至0.025
- 更新率：DQN每4 steps更新一次；target network每1000steps更新一次

二、Learning Curve

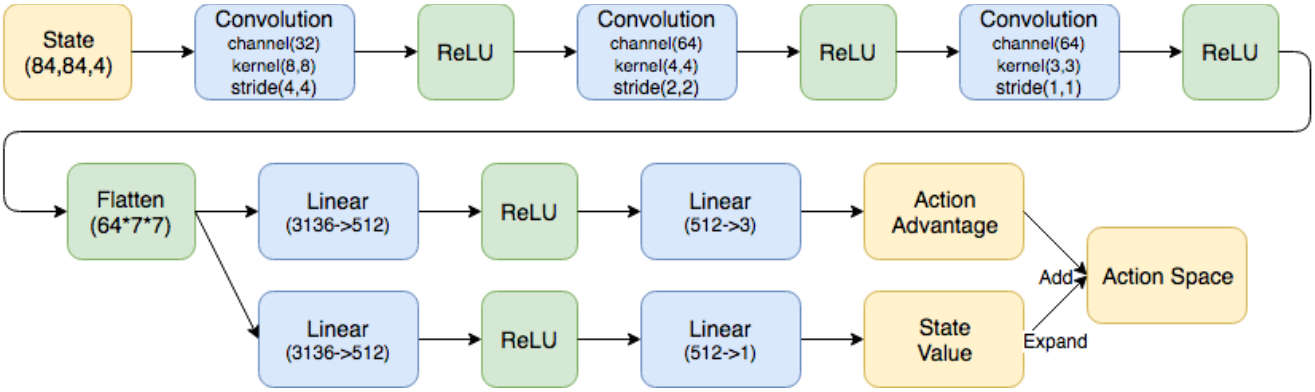


三、Dueling

(a.) 架構與想法

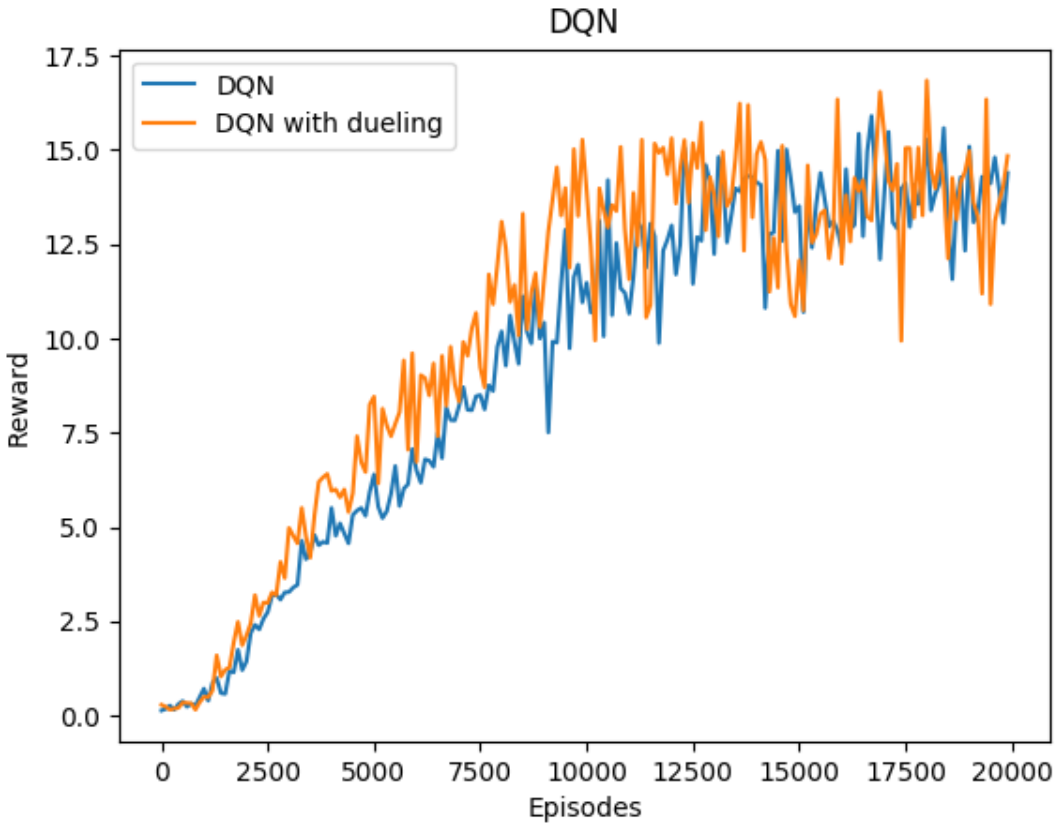
因為在breakout遊戲中，很多時候做什麼action可能對之後的結果幾乎沒有影響。例如當球在所有方塊上面狂彈的時候，往往選擇什麼action對於造成的結果是沒有關係的。

因此在model中分成action advantage希望模型能從state知道採取某action能得到的優勢，又分成state value希望模型能學習到每個state都有該state的分數，所以在很多與action無關state的training情況下，在model拿到reward時，action的reward可以因為無關而很低，此時state value就會對model的reward輸出佔較大的比例，能幫助機器明白一些這種狀況，應該能較容易訓練。



(b.) Learning Curve與結果比較

很明顯在幾乎相同的架構下，dueling訓練得較快，比一般DQN還快收斂，但是最終的收斂結果其實是差不多的。在testing時dueling DQN可以打到最高400多分，vanilla DQN也是能達到390左右，兩者平均也都有60幾分，其實是差不多的，都是不錯的結果。

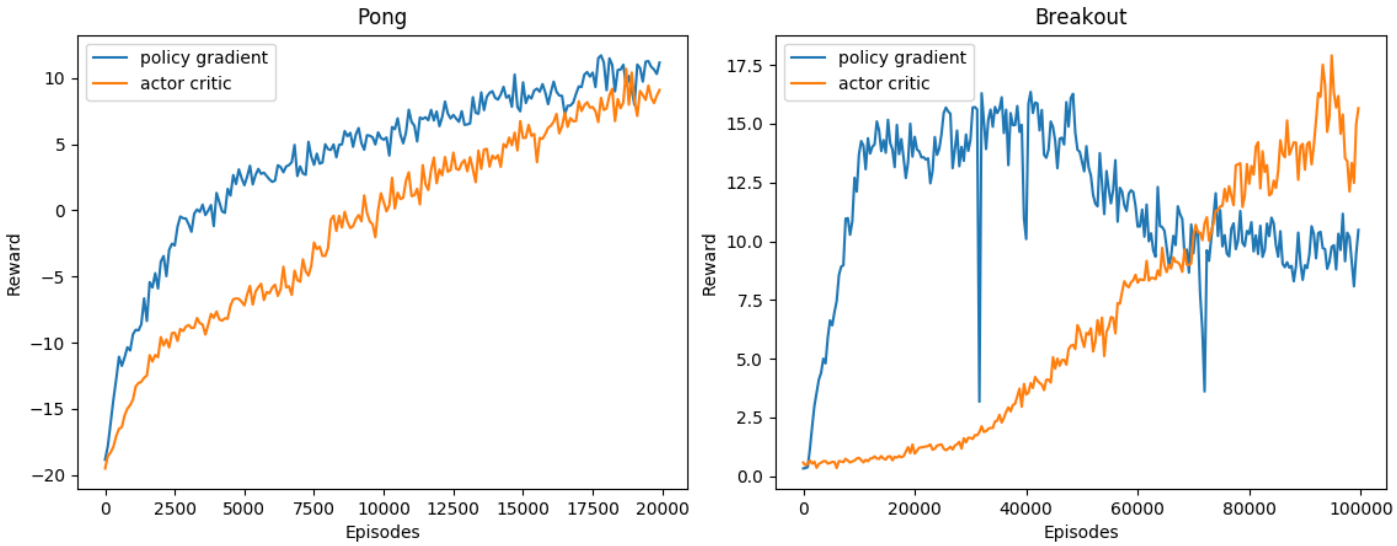


4-3 Actor Critic

Actor Critic

我們對4-1的遊戲Pong與4-2的遊戲breakout做了Actor Critic，**架構上**與4-1與4-2的policy gradient架構類似，只是多了一個layer也就是value function，把hidden layer的output經過一個fully connected network得到一個state value，該state value是state的accumulate reward。更新參數時，用真實sample的accumulate reward與value function出來的reward算mse並backward更新；同

時用value與reward差，當作policy gradient更新actor的reward並更新。可以發現actor critic雖然一開始起步較慢，但是最後收斂的結果會比Policy gradient與DQN還好(右圖有一點錯誤，藍線應該是DQN而不是policy gradient)



A3C(Asynchronous A2C)

A3C是把A2C的方法，做平行處理，一次多個CPU做action sampling並在結束後各自更新共用的model，此方法可以快速收斂且往多個sample組合進行探索，得到很好的結果。

我們使用A3C論文Asynchronous Methods for Deep Reinforcement Learning(https://arxiv.org/pdf/1602.01783v1.pdf)的架構用10個平行線程跑了Pong與Breakout，A3C在Pong與Breakout上結果都遠超過policy gradient, DQN, actor critic，不論從訓練速度或結果都是A3C完勝。注意右圖有分成兩子圖，因為A3C效果實在遠超一般reinforcement與actor critic，只需1000 episode就可以在training拿到約60分的結果，而DQN actor critic要到60000才達到15分左右，尺度相差很大所以特別分開作圖。