ADLxMLDS HW3 Report

資工所 碩一 R06922055 吳均庭

Model Description

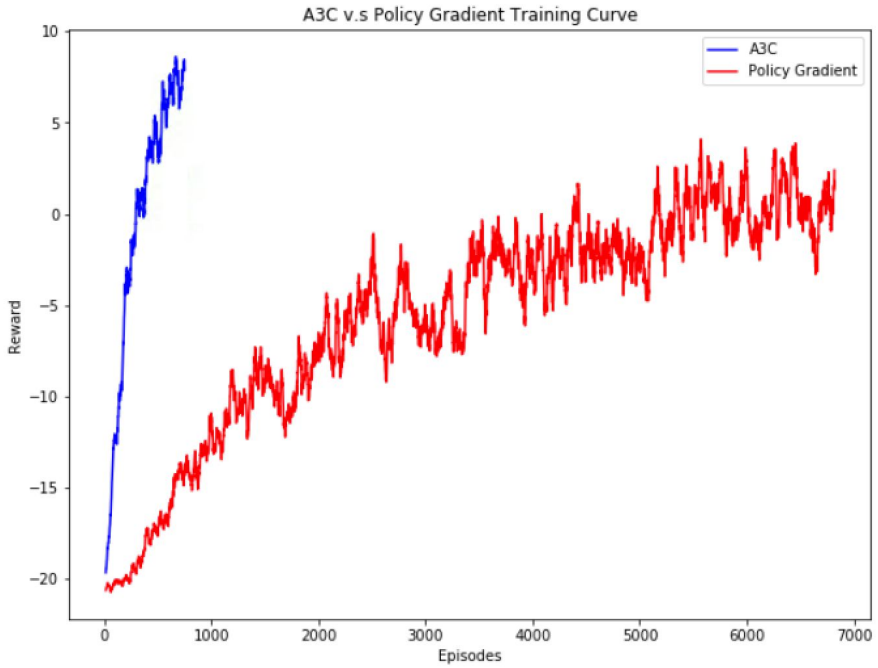
- Pong:
    - Policy Gradient

      將env拿到前後兩個frame的observation相減，形成 80 * 80 * 1 的tensor作為input，通過兩層Fully connected layers，第一層通過actvation function relu，第二層經過sigmoid輸出一個up action的機率p。訓練方法為每供經過N個step作為一個batch，將batch內每個step的reward加上未來reward 的總和乘上一個discount gamma表示對未來reward的不確定性。網路目標為最大化action所得reward 的期望值，實作上用網路output p與實際sample出來的action 還有discounted reward 來計算loss和gradient，並使用Adam optimizer來進行網路更新訓練。

    - A3C Asynchronous Advantage Actor-Critic

      因網路收斂緩慢，決定改成實作A3C的網路架構，網路架構可分為兩個部分，一個Global network 與數個agent network，兩種網路具有相同的架構，input先經過兩層cnn layer分別為： [8 x 8] 16 stride=4，activation = rele、 [4 x 4] 16 stride=2，activation = relu、flaten後經過一層dense 行程256D vector，接著分為actor network 與 value network 兩個部分，Actor network 將256D 過一層fc成長度=action space的vector，通過softmax 計算每種action的機率、value network則將256D過fc 計算一個scalar評估當前state的好壞。

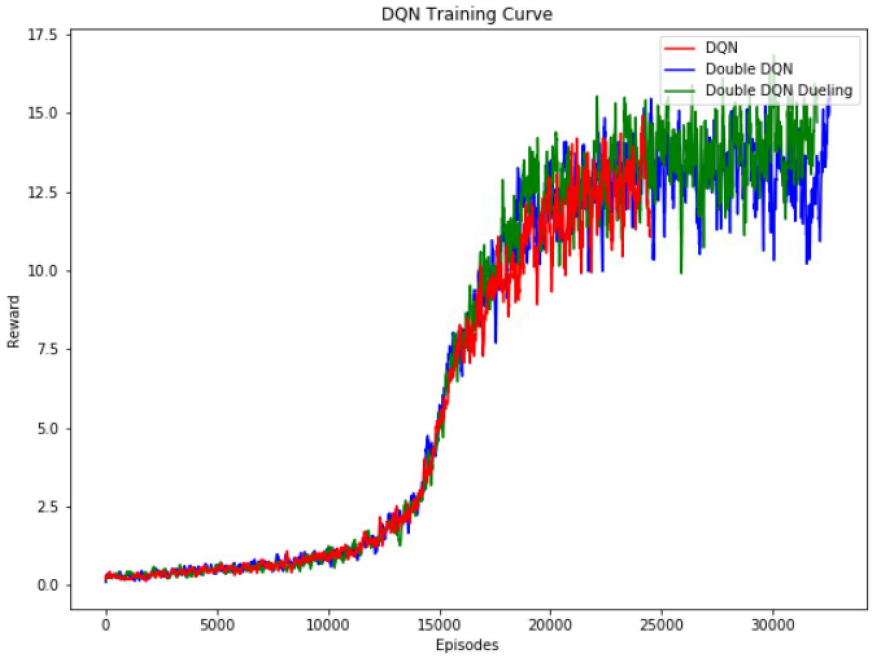
      在作業實作中，訓練方法為，先產生一個global network與 8條thread的 agent network，每個worker 會分別與環境互動，每五球或episode結束，利用自己的value_loss與polic_loss計算total loss = policy_loss + value_loss * 0.5，並計算gradient，使用RMSprop 來更新global network 的參數，再將global network 最新的參數copy回agent local network，用最新的參數繼續探索環境。



- Breakout:
    - DQN

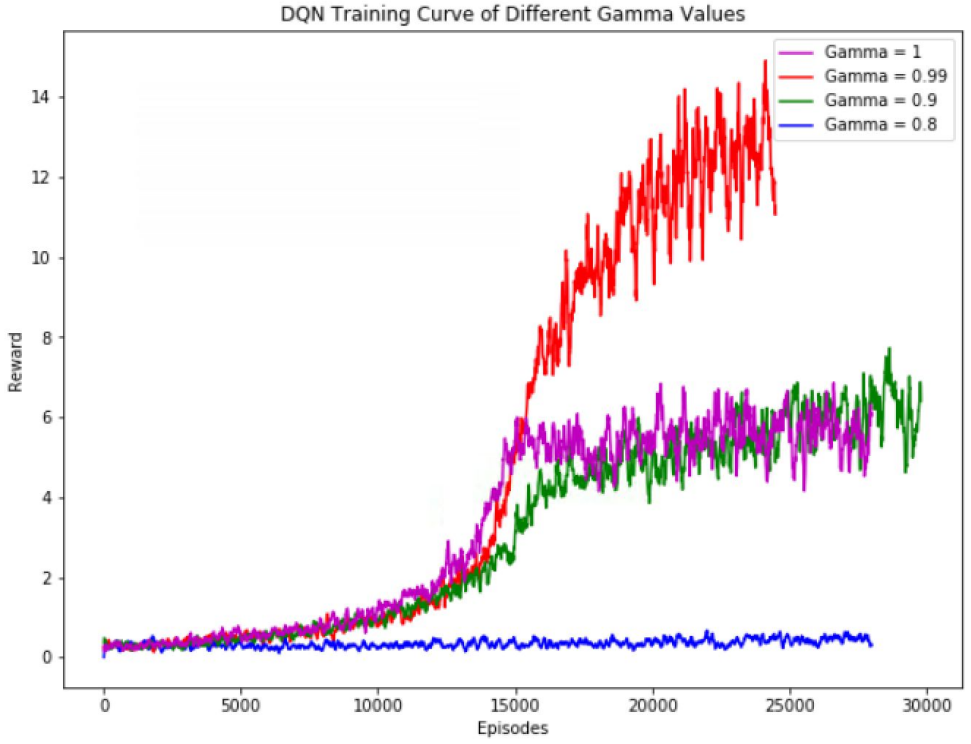
      網路架構為 [8 * 8] * 32 stride = 4、 [4* 4] * 64 stride = 2、 [3* 3] * 64 stride =1 三層CNN activation = relu，拉直之後通過fc成為512-D vector 通過 leaky-reluactivation最後再經過一層fc 轉為 3維的vector表示每種action的Q 值。DQN包含target network、online network 兩個相同架構的網路，訓練方式為，將observation 84 * 84 * 4的 tensor 餵進online network估計Q值，並以一定機率隨機選擇action來探索環境，或者對Q取argmax選擇action，將 observation與結果一起存進memory，供網路學習使用。

      在一定數量的step(4)之後，會從memory中sample出32個observation，將前後的observation分別餵進target與 online network。把action的reward 加上前一個 state的Q值，與目前online network所估計的Q值，可以計算MSE loss再透過 RMSprop (decay = 0.99) 更新online network參數，在一定間隔之後(1000) 將 online network 參數assign 給target network，即可完成網路訓練。



Experimenting with DQN hyperparameters

- 選擇不同 gamma值的Training Curve：

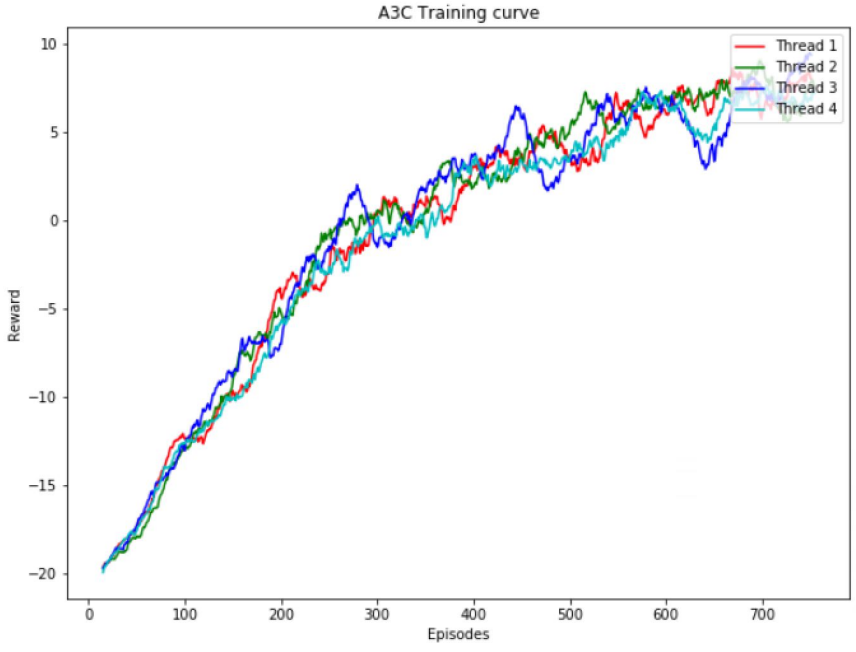


- Explanation
    - 我們發現gamma 對training結果有很大的影響，由dqn所使用的loss function 是將當前reward 加上一個衰減的上一個state 的Q值來當作學習的目標，這個衰減係數，表示當下的reward 較為重要，越遠之後的reward對目前的影響越小。

gamma = 0表示只考慮當前的reward，而 >=1 會造成期望值不收斂，可能造成網路訓練失敗。從圖上可以發現gamma 在 = 0.99時的學習效果最好，而gamma = 0.8時會造成網路訓練完全失敗，gamma=0.9時會造成網路到中途就無法上升，而gamma =1 時reward有上升，但同樣到中途就無法上升。從本圖可以得知，gamma值對dqn網路的訓練十分重要，一般選擇0.99來進行網路訓練。

- Bonus:
  - A3C:

    A3C(Asynchronous Advantage Actor-Critic) 網路架構與訓練比較圖如報告第一部分所示，比起原始的policy gradient多了 Advantage、Actor-Critic與並使用多條thread 同時進行訓練。從比較圖中，可以發現A3C網路架構，可以更快達到收斂，下圖為訓練時每個thread reward對 episode的變化，



  - DQN
    - Double DQN

      原本的網路，target network會取每個action Q 值的max 來估計當前Q值，但可能會造成高估，在doube dqn 則利用原本就有兩個網路的優勢，將前一個state 餵進online network得到一組action的Q值，取argmax之後再從target network output 中取得相對應的Q值來計算loss，從上圖中可以發現，training curve比起原來的DQN有些許的提升。

    - Dueling

原本的網路output 即為每個action的Q值，但實際情況中，在某些state之下，選擇什麼action沒有太大差異。透過在網路上加上兩個branch，一個predict 對於當前state 每個 action 的advantange 以及對當前state的value，接著計算 value + (advanage- mean(advantage))，來作為最後網路的output。使用dueling可以使網路不用硬學對應state 的action，而是學到action 相對於state的關係，從上圖可以看到加上 Dueling 可以使 Double DQN的 learning curve 有些許的提升。