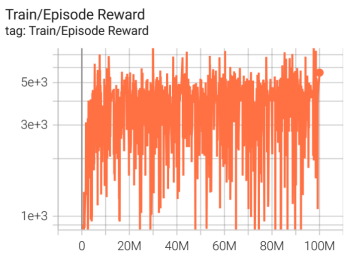
Lab2 - DQN

313551153 施儒宏

September 2024

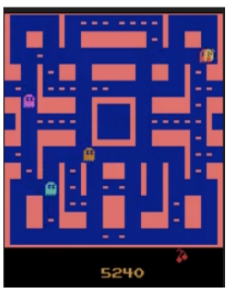
1 Screenshot of Tensorboard training curve and Testing results on DQN (30%).

1.1 Training curve



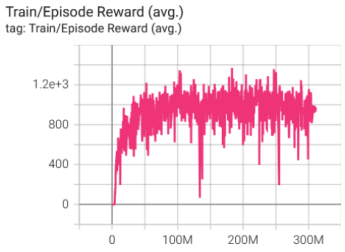
1.2 Testing results

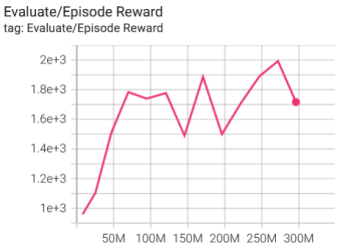
```
total reward: 5240.0
================================================
Evaluating...
episode 1 reward: 6030.0
episode 2 reward: 5190.0
episode 3 reward: 2260.0
episode 4 reward: 7090.0
episode 5 reward: 5300.0
average score: 5174.0
================================================
Moviepy - Building video ./record.mp4.
Moviepy - Writing video ./record.mp4
```



2 Screenshot of Tensorboard training curve and Testing results on Enduro-v5 using DQN (10%).

2.1 Training curve

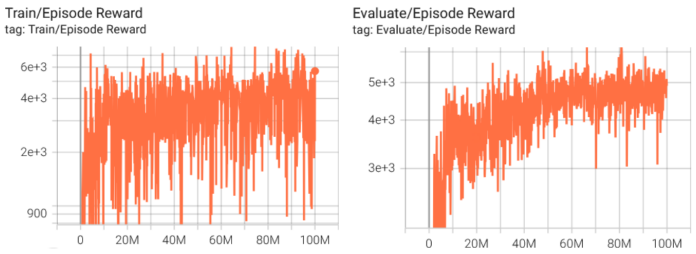




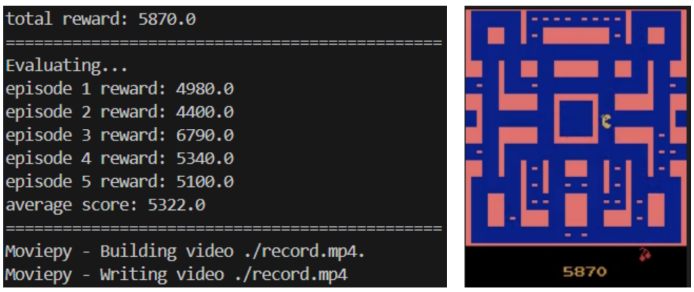
2.2 Testing results

3 Screenshot of Tensorboard training curve and Testing results on DDQN, and discuss the difference between DQN and DDQN (3%).

3.1 Training curve



3.2 Testing results



3.3 Difference between DQN and DDQN

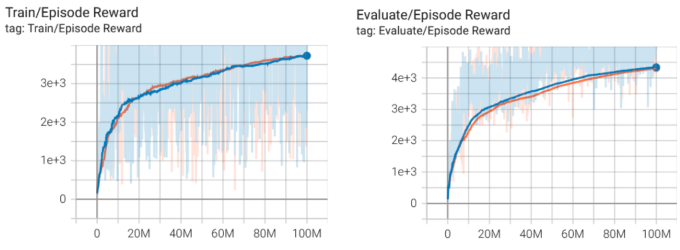


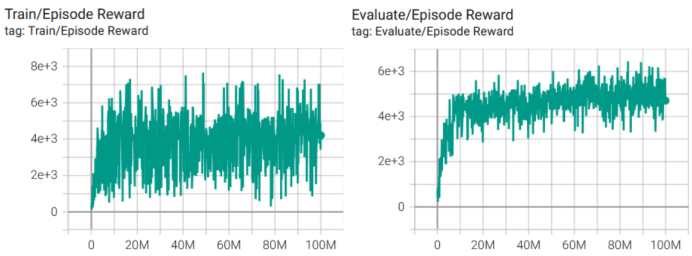
Figure 1: DDQN(Blue), DQN(Orange)

DDQN 和 DQN 的差別在於 Target 計算的方式不一樣。DDQN 爲了解決 overestimation 的問題，利用 Behavior network 選擇 S_{next} 的最佳動作 a_{next}, 再利用 Target network 和 a_{next} 得出每個動作的 Q_{value}，最後乘上 discount factor 再加上 reward 得到 Q_{target}。
在 Learning Curve 上，其實兩者的是差不多的, 但在 Evaluation curve 中, 可以看到 DDQN 在 20M 之後其實都是優於 DQN 的，且波動幅度較小。這可能是因爲 DDQN 在訓練上可以得到相對可靠的 Q 值，讓 Behavior network 在 evaluation 時都能得到稍微好的結果。

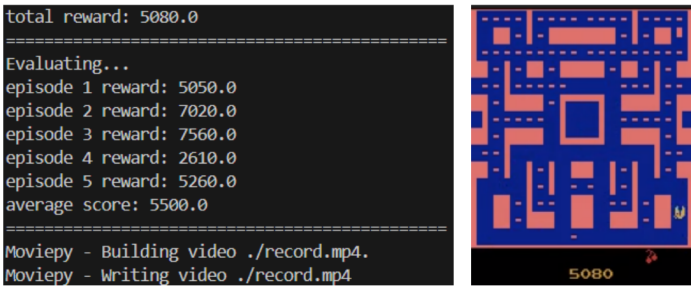
Overestimation：在原公式中, 永遠是取 Q_{value} 值最高的動作來更新 Q_{value}，但有些情況其實先選次優才能得到最優的解。另外這種算法會在錯誤的方向上反覆嘗試，從而導致收斂速度降低。

4 Screenshot of Tensorboard training curve and Testing results on Dueling DQN, and discuss the difference between DQN and Dueling DQN (3%).

4.1 Training curve



4.2 Testing results



4.3 Difference between DQN and Dueling DQN

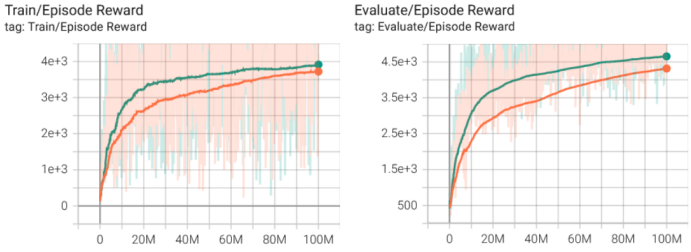
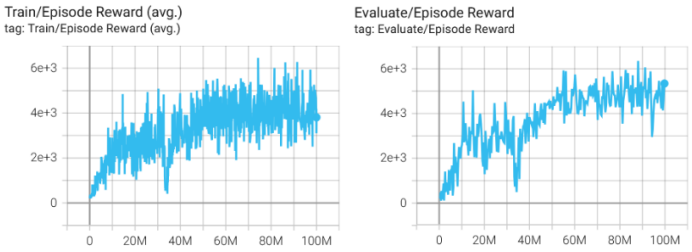


Figure 2: Dualing(Green), DQN(Orange)

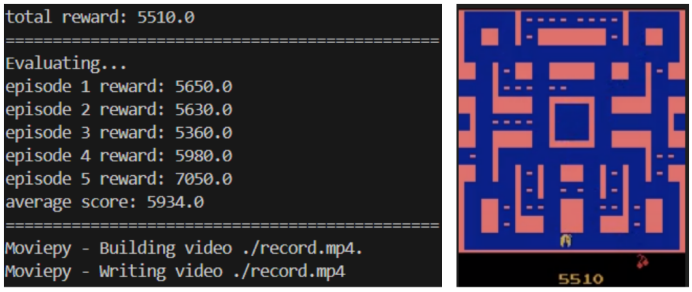
Dualing DQN 和 DQN 的差別在於 Network layer 實作的方式不同，在我的 Dualing DQN 的 network 實作中，我是將原本的 Classfier 層直接加上一個額外的 Output 作爲 value 值，其他的作爲 advantage，最後跟著公式對他們作處理並得到 Q_value 輸出。這樣的做法透過減去均值降低網路的學習的 variance，直覺上來說也能透過這種方法去學每個 Action 對於 Observation 的重要程度。在結果上我們也可以看到 Dueling DQN 的訓練雖然結果是差不多的，但是在訓練初期 Dualing DQN(深紅色) 具有更好的上升速度，間接證明了這種網路對於 Observation 和 Action 的關係能給予更準確地學習方向。

5 Screenshot of Tensorboard training curve and Testing results on DQN with parallelized rollout, and discuss the difference between DQN and DQN with parallelized rollout (4%).

5.1 Training curve



5.2 Testing results



5.3 Difference between DQN and DQN with parallelized rollout

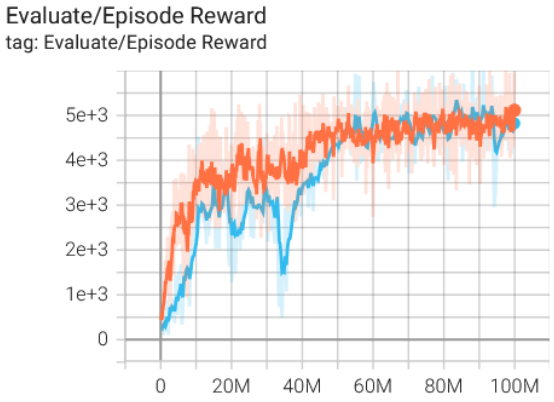


Figure 3: DDQN(Blue), DQN(Orange)

本質上兩者在梯度的 Update 上並無區別，兩者的區別主要在於系統方面的實作，對於 Parallelized rollout 來說，我這裡的實作方法是 overwrite training function。對於每個環境，雖然是異步執行於不同 threads 上，但爲了能夠呈現於 tensorboard，我會讓等待所有環境都死亡才一起 reset 並進入下一個 episode。每個 episode 我會取每個環境的 reward 平均作爲 output。對於 replay-buffer, 每個環境的 experience 都會被 enque 是 replay-buffer 等待 sampling。可以看到 DQN 大部分情況都優於 Parallelized rollout DQN 的，因爲這種設計方式會把 DQN 永遠選當下最優解的缺點再放大，在 Replaybuffer 中，同一時間可能每個 Environment 都往錯誤的方向去嘗試。從曲線上看，Parallelized Rollout 的方式直到 60M 才趨於穩定，我認爲應該要搭配 Dueling DQN 或者 Double DQN 才能得到更好的效果。