ADLxMLDS2017 HW3 Report
B03b02014 張皓鈞

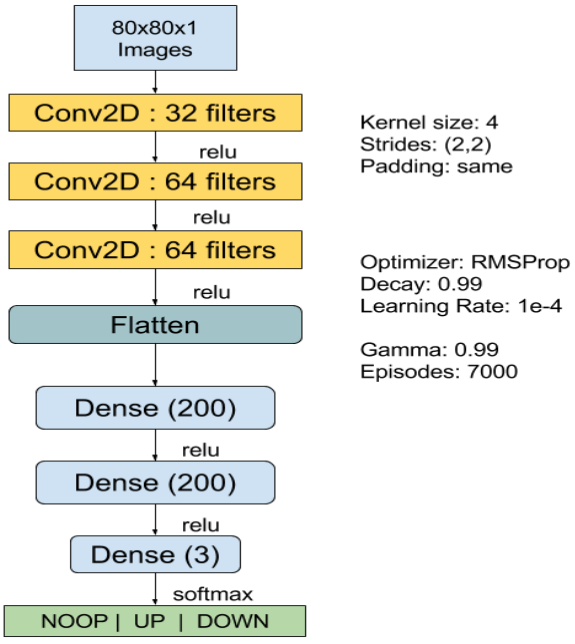
- **Policy Gradient**
  - 前處理：

將圖片只取一個**channel**，經過**downsample**以及去除邊界畫面、把雙方板子和乒乓球的顏色轉為**1**，其餘轉成**0**。在輸入模型前，取該時間點畫面和前一個時間點畫面的差值。

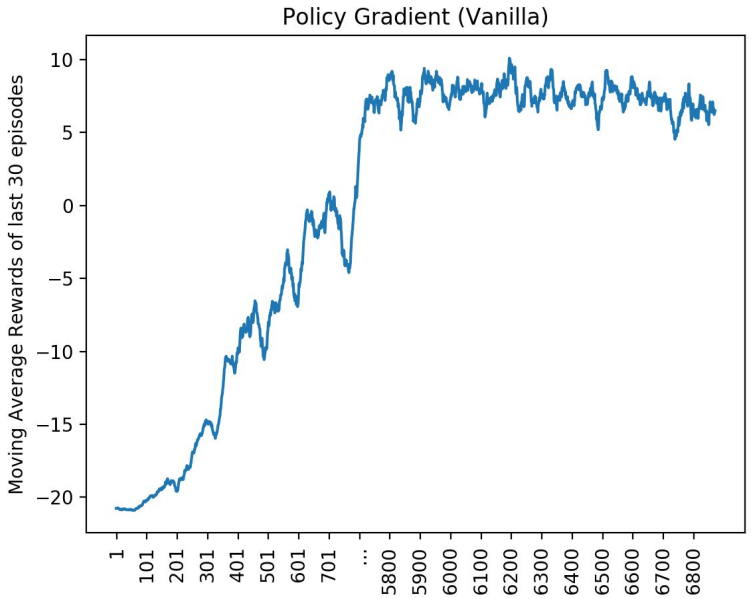
  - 獎勵處理：

將每一場的獎勵乘以**gamma**進行打折。
每次更新網路的時候將打折後的獎勵標準化以確保**gradient**的數值穩定。

  - 詳細**Policy Network**模型架構：



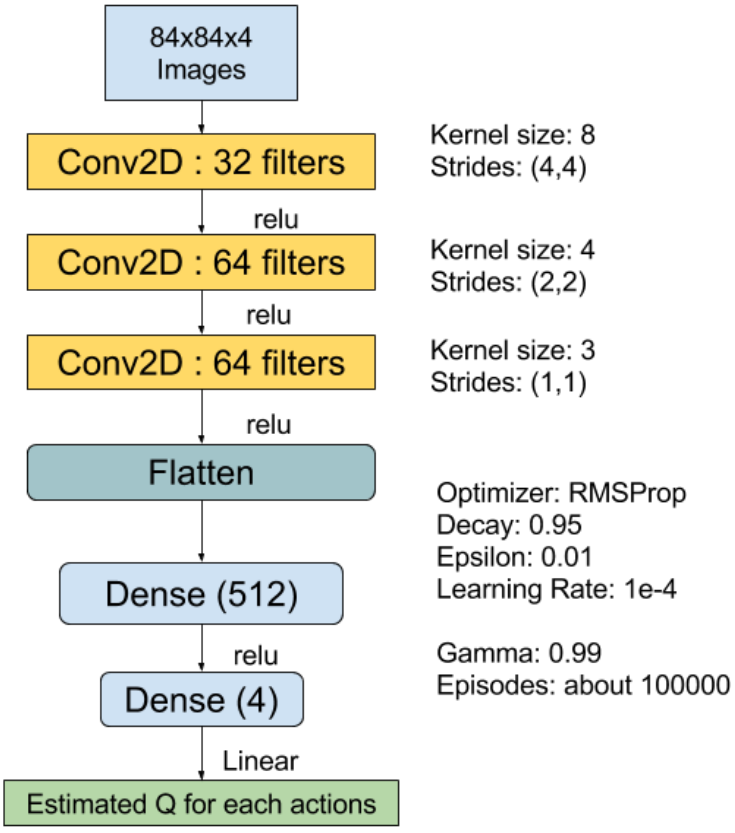
  - 學習曲線：



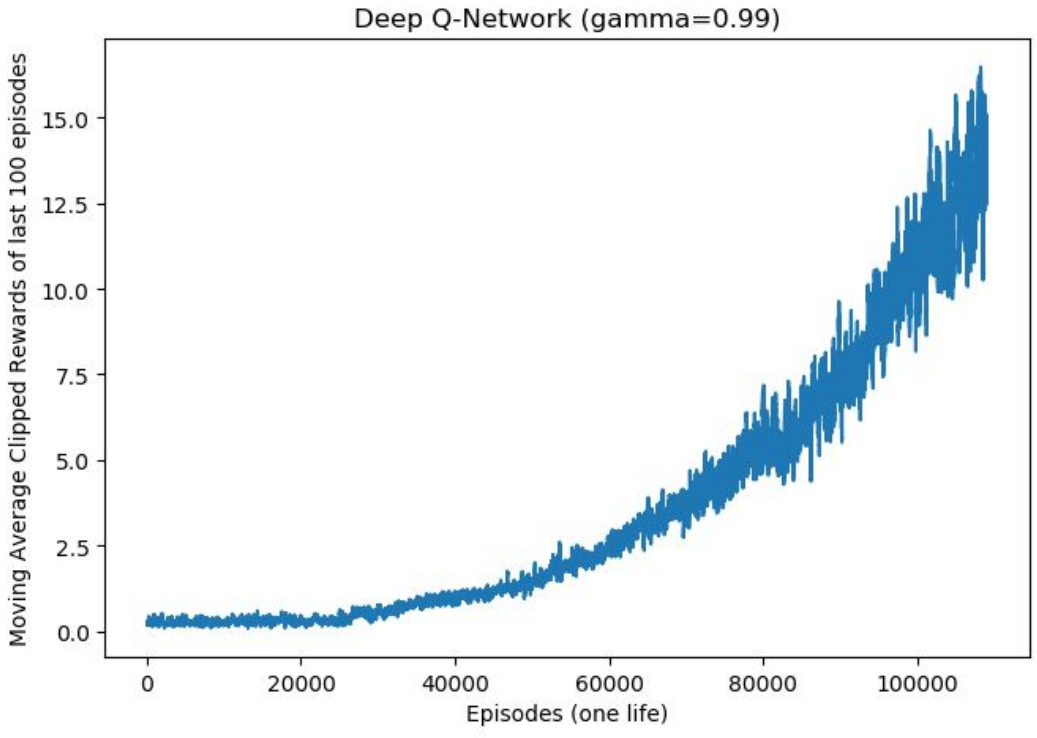
- **Deep Q-Network (DQN)**
  - 前處理：

感謝助教事先使用**atari_wrapper**將畫面疊成 **84x84x4**的矩陣。且將獎勵修剪到**-1, 0, 1**的值。

  - 詳細**Deep Q Network**模型架構：



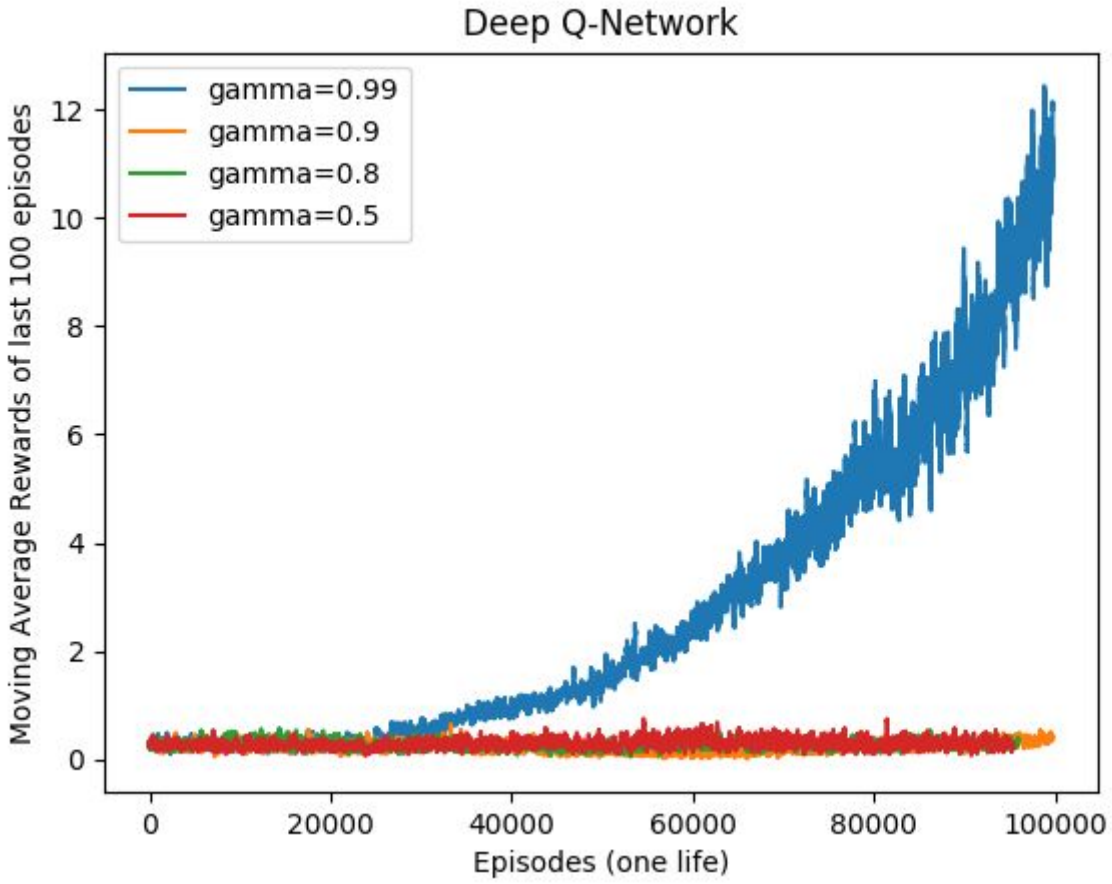
  - 學習曲線：



- **Experiment of DQN Hyperparameter**
  - 我選擇將獎勵打折的**打折係數(gamma)**作為實驗**DQN**的參數：

    原因是**gamma**的值提供了我們在訓練**agent**時它所看到的**"視野"**，**gamma**值越高表示所考慮的步數越多，**gamma**若等於零則表示只看現在畫面下所得到的分數。而在多數的訓練環境中，包括這次的**Breakout**遊戲，通常都是需要考慮到未來的獎勵，才能學出現在應該要做的動作。所以我想在這探討**agent**所考慮的視野遠近對它在**Breakout**這個遊戲中的表現有何影響。

  - 學習曲線比較：



  - 討論：

    從學習曲線的比較可以看到**gamma**在另外三種設定都無法成功有效學習到**Breakout**這個環境所給的獎勵和畫面的關係，顯示**gamma**的設定在訓練模型中扮演重要的角色。若是將**agent**的**"視野"**設定成太近，則會造成**agent**無法有效判斷該畫面的預期獎勵，即使是**gamma=0.9**的模型仍舊和另外兩個**gamma**值小的模型表現差不多。

    由於時間關係，沒有再進一步探討當**gamma**大於**0.99**時所訓練出來的模型表現。

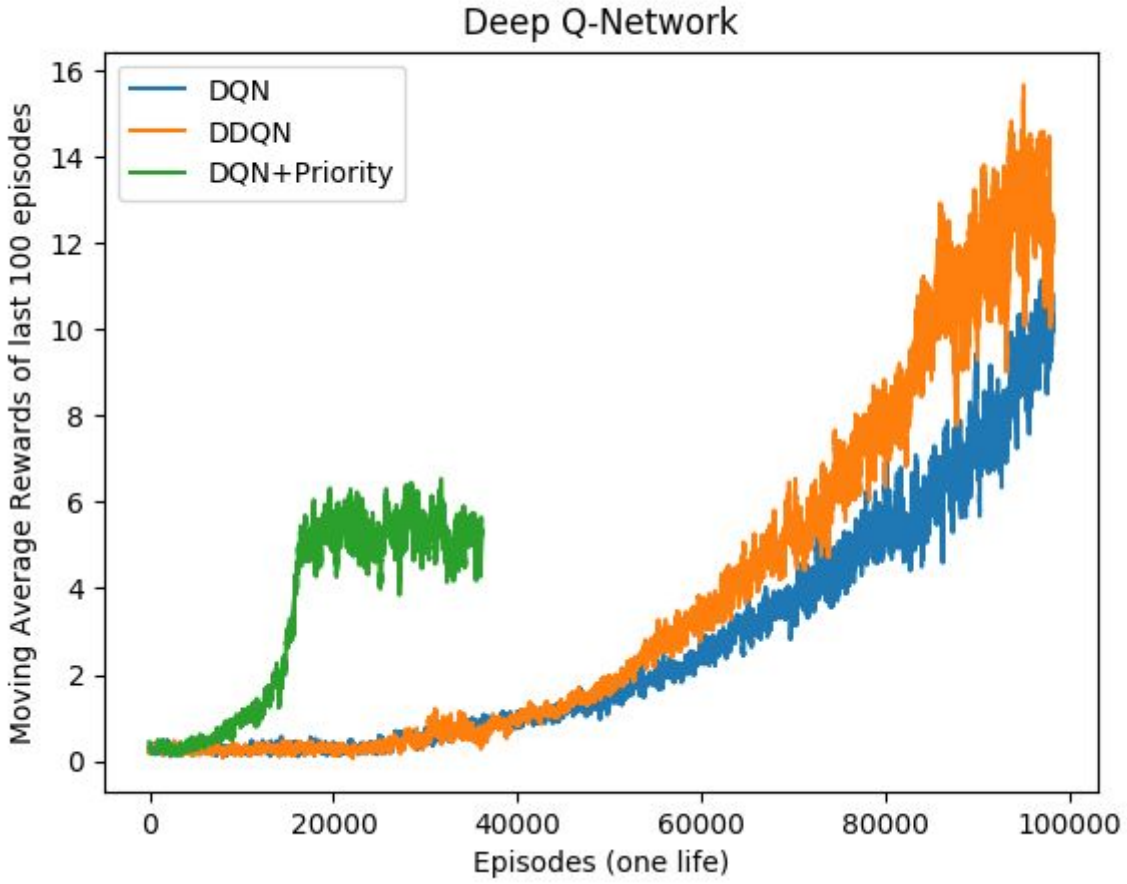
- **DQN Improvement:**
  1. **Double DQN**

     和上述**DQN**的差別在於：**next state**預測出的**Q-value**是有經過另一個**Q-network**來評估的，可以去除掉每次為了讓**Q**值最大化的偏差，此評估用的**Q-network**就如同**DQN**的**target Q-network**，固定每幾個**step**才更新一次，讓真正的**Q-network**可以有比較穩定的學習方向。

  2. **Prioritized Replay**

     和上述**DQN**的差別在於：在**Q-network**學習存取起來的記憶時，優先抽取較大**loss**的經驗，也就是模型比較不擅長的訓練例子有比較高的權重。每次更新**Q-network**時以比較大的機率選取模型不擅長的例子來更新，使梯度可以更有效率地指向最佳方向。

     Improvement與原本**DQN**的學習曲線比較：



     討論：

     可以看到**DDQN**有比較快的學習速度，平均獎勵上升的速度比起一般的**DQN**來得快。符合我們對**DDQN**去除掉偏差後可以更有效的學習之期待。

     而一般**DQN**加上**Priority**的學習記憶，在**Breakout**的效果上可以看到一開始學習速度比另外兩者還要迅速，但在**20000~40000**之間開始震盪，且沒有進步的趨勢。