作業三 Game Playing

劉彥廷 B03902036

---

1. 模型敘述

1.1. Policy Gradient

模型參照了 [3] 的介紹。由兩層 fully-connected layer 所構成，第一層由 200 個單元所構成，第二層則對應至輸出的動作數量（強制縮限到只剩下「Up」跟「Down」兩種動作）。影像經過預處理縮小為 80×80 的方格，並且二值化。

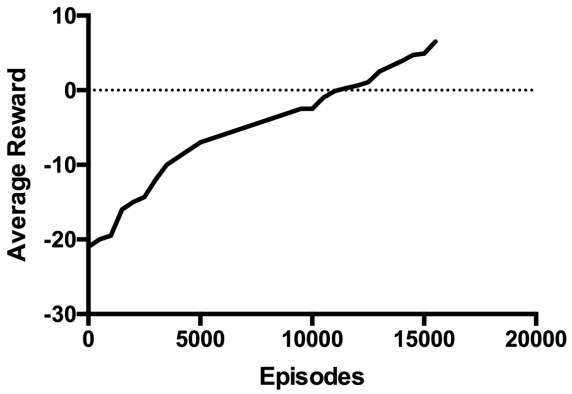

Figure 1: 學習曲線

繪圖的時候沒注意到只存了 episode 數而不是 time step，但已經來不及了。

1.2. DQN

參考了 [1] 的 GitHub 專案與助教所提供的提示。總共有三層 Conv2D，單元數分別為 32/64/64，kernel 大小依序為 8/4/3，間隔取樣距離分別為 4/2/1（均為方陣）。Conv2D 的 activation 方式均為 relu，而緊接在後的兩層 FC 則分別有 512 個單元與 6 個（總操作數量），activation 方式分別為 ReLU 與 softmax。影像經過預處理為 8-bit 灰階並縮小至 84×84 的方格。

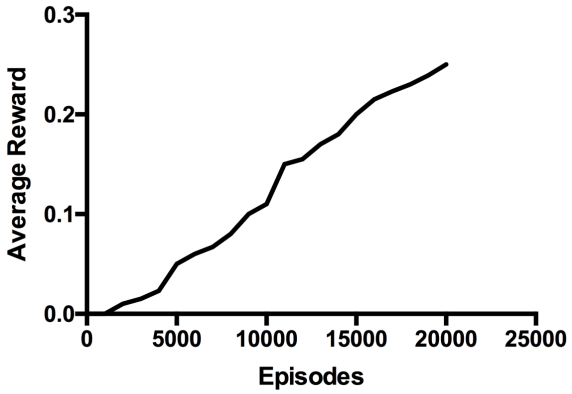

Figure 2: **學習曲線（未達** baseline）

繪圖的時候沒注意到只存了 episode **數而不是** time step**，但已經來不及了。**

2. 參數調整

選擇了 TARGET_UPDATE_INTERVAL **（投影片當中的** target network update frequency **）進行調整。共計四種參數，**1k/2k/5k/10k **剛好覆蓋了一個數量級之內的範圍，並且停在固定的** episodes **數。**

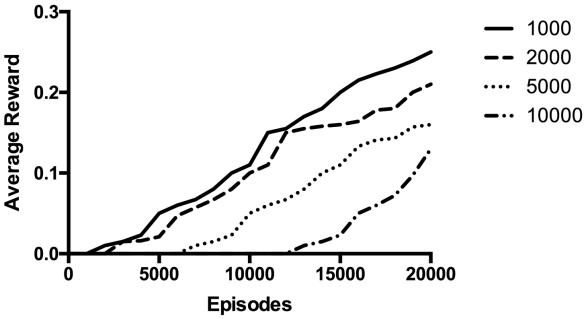

Figure 3: **學習曲線（未達** baseline）

在搜尋參考用的模型，發現到這篇 [2] **討論串，似乎這個參數會影響到模型學習的速度，考慮到時間有限，覺得這個參數可能會影響比較顯著。不過原始文章似乎也說明了，更新頻率拉高的時候，會讓分數上升速度快了些，但後期上升的幅度會變慢。這似乎可以從** Fig. 3 **看到，但似乎需要更多的** epsiodes/timesteps **才能更確定是否真的如此。**

3. 參考文獻

[1] Dqn implementation in keras + tensorflow + openai gym. `https://github.com/tokb23/dqn`. (Accessed on 12/16/2017).
[2] The test score is different from the deepmind paper · issue #32 · tambetm/simple_dqn. `https://github.com/tambetm/simple_dqn/issues/32`. (Accessed on 12/16/2017).
[3] Karpathy, A. (2016). Deep reinforcement learning: Pong from pixels. `http://karpathy.github.io/2016/05/31/rl/`. (Accessed on 12/16/2017).