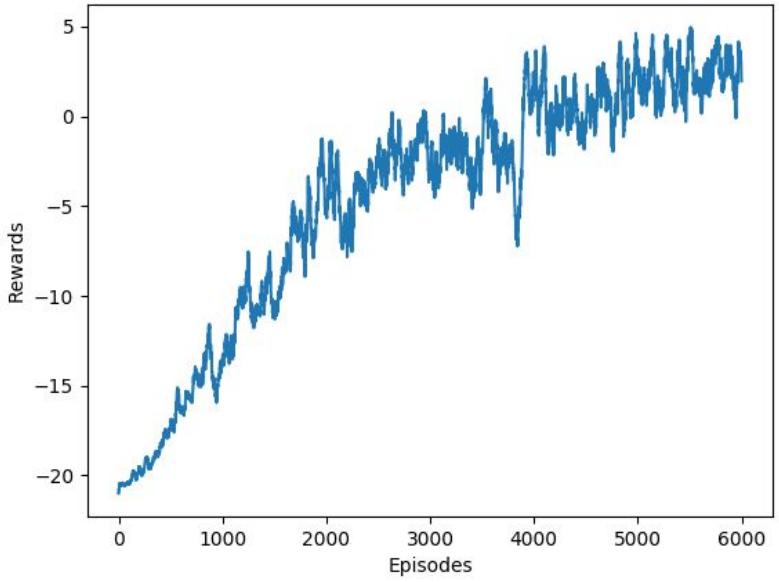
MLDS HW4 Report

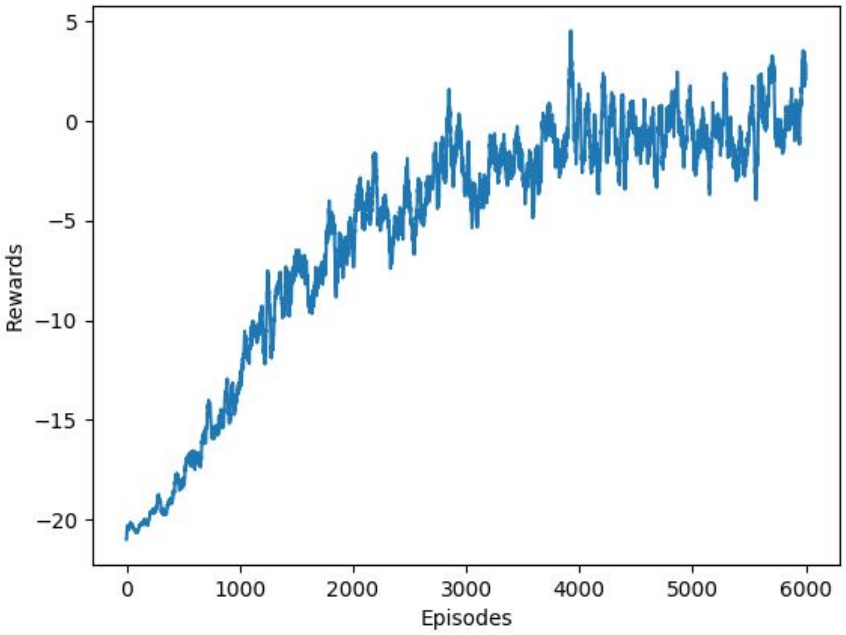
許芯瑜、李祐賢、熊展軒

HW4-1:

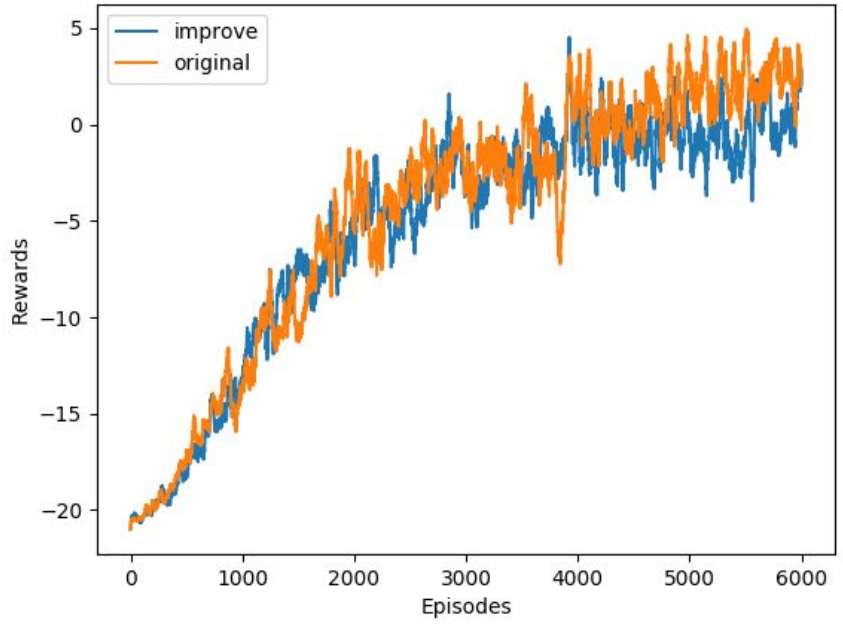
- **Policy Gradient model**
  首先，先對畫面做預處理，將210x160的畫面crop成160x160，也就是將上面比分crop掉，再對160x160做down-sample成80x80，最後將畫面中的背景都設為0，球和板子設為1，即可將原本彩色的input改成黑白input，這有助於訓練。再來，將80x80的畫面做flatten成6400的vector，過一層256-dim的fully connected layer，再過一層1-dim的fully connected layer輸出球拍向上的機率。在training過程中，我們將當一方進球時設為一個batch，將一個batch內的每個step的reward加上未來的每一個step的reward的總和乘上一個discount，減小step對未來reward的影響力。網路的目標是最大化action得到reward的期望值，實作方式為利用網路的output probability與sample出的action及discount reward求得loss及gradient，再利用adam求optimization。
- **Learning curve**



- **Improvement**
  我們選用variance reduction的方法來improve。我們在原本的network多加了一個分支去預測reward的baseline，再對已做過standardize的reward及預測出的baseline求得mse loss，並利用這個loss對分支網路做optimize。我們將原本的reward減去預測出的baseline作為我們新的reward，新的reward的variance較低，可以讓訓練過程較平穩，也較快收斂。
- **Learning curve**



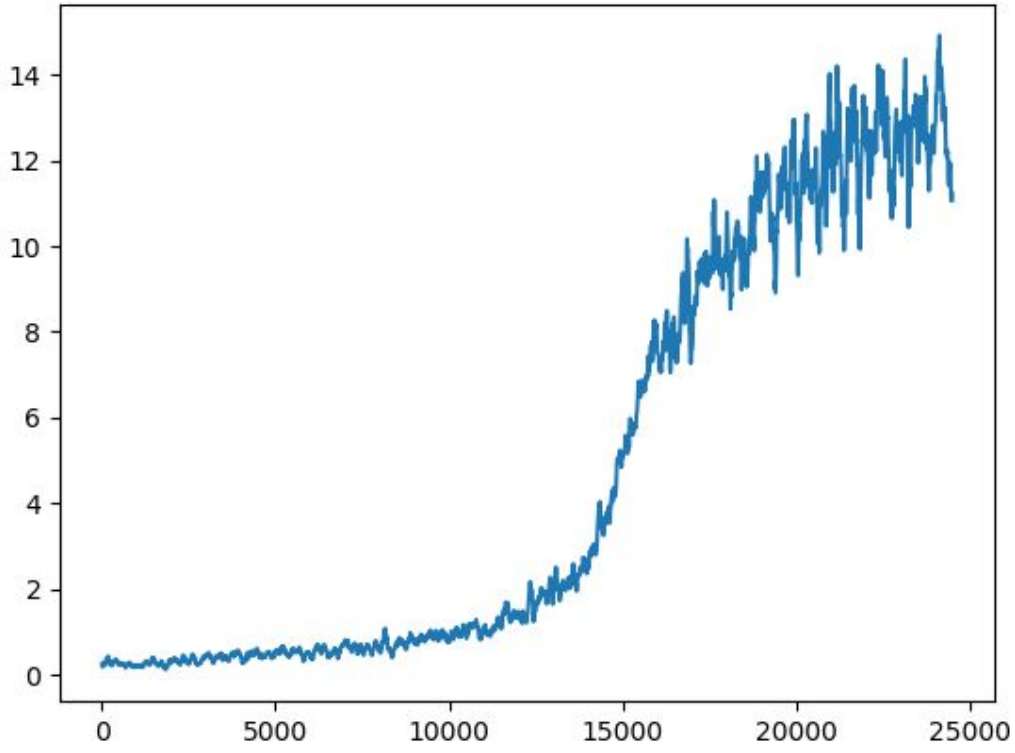
- **Compare to the vallina policy gradient**
  單從learning curve可以看出使用variance reduction方法的curve會比較平穩，但若將兩個learning curve畫在同一張圖，如下圖，可以看出不使用variance reduction方法的rewards會高一點點。我們認為會出現這樣的情況是因為我們只訓練了6000個episodes，在這麼少episode的情況下無法將baseline學得很好，導致rewards不升反減。如果將訓練時間拉長可能可以看到variance reduction真正的效果。



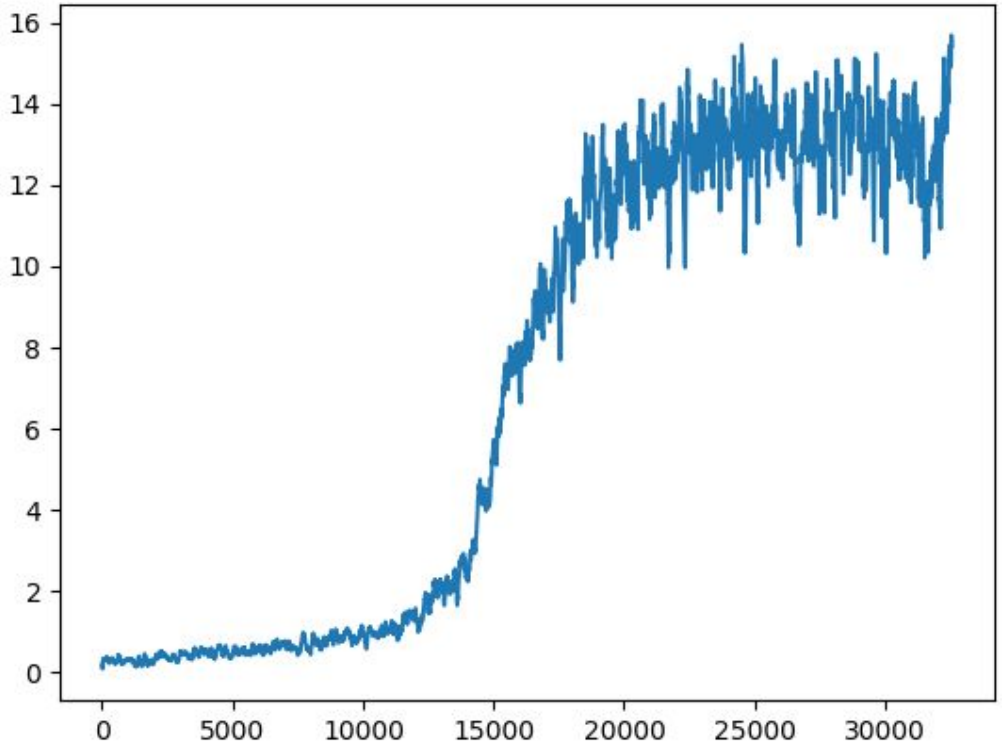
HW4-2:

- **Deep Q learning:**
  首先將初始畫面reahpe成(1, 84, 84, 4), 然後丟進一個CNN model, ouput為三維的vector, 分別對應到遊戲操作"左"、"不動"、"右"的value. 我使用的計算rewar方式是TM, 因此CNN model會分別給出這個stage和下個stage的value,目標是他們之間的差要盡量等於下個stage所得到reward.

  另外,為了避免Q model偏袒某個遊戲操作, 在一開始會有比較大的機率所給出的value是隨機的,為的就是希望在一開始model還沒有很強時盡量讓它嘗試使用不同的方法
- **Learning curve:**



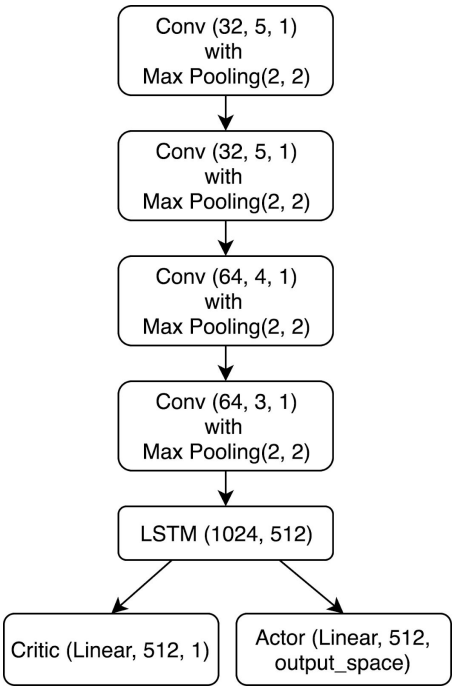
- **Improvement:**
  我使用**dueling** 的方法來做improve. 將原本的Q(s,a) 拆解成 V(s) + A(s,a).另外給A(s,a)的限制是action的加總要是0.
- **Learning curve:**



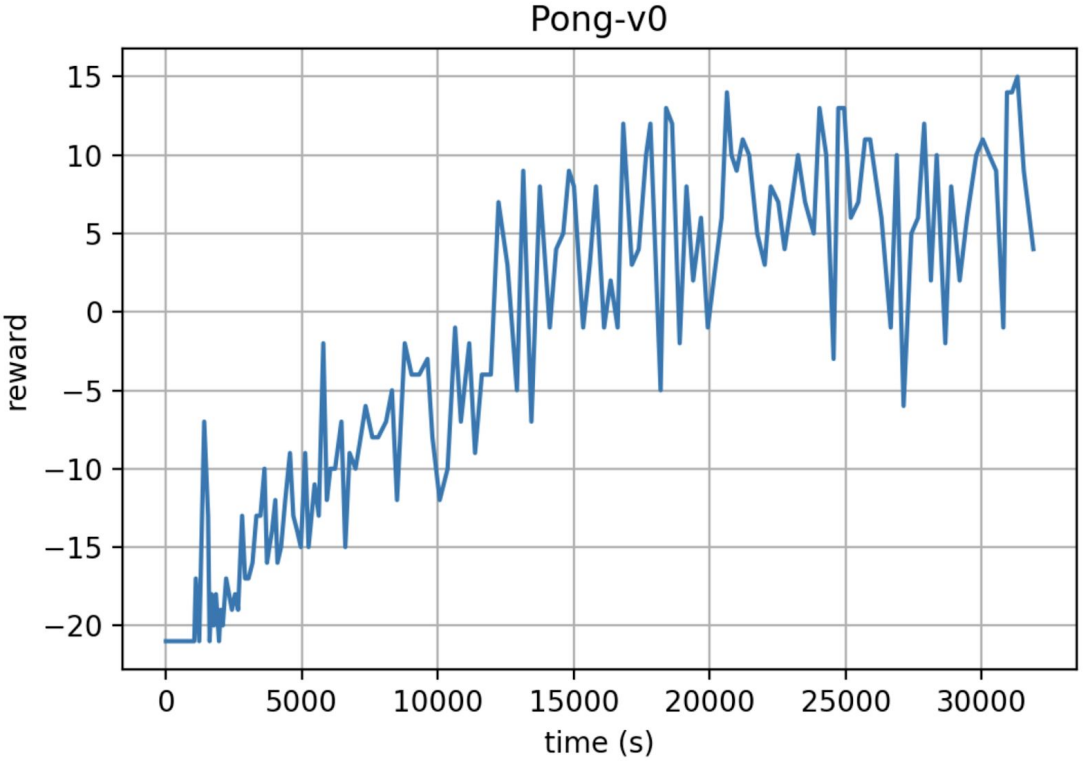
- **Compare to original deep Q learning**
  我們每次都sample 32 筆data做訓練, 但其它沒有被sample但情況相似的data難以一併被更新,而V(s)的機制使得全部的data都可以一併做更新.

HW4-3:

- **Model Description:**
  兩者皆是用以下Model，先用CNN處理影像，在經過LSTM處理過，最後餵給Critic和Actor Network（皆為單層線性）。

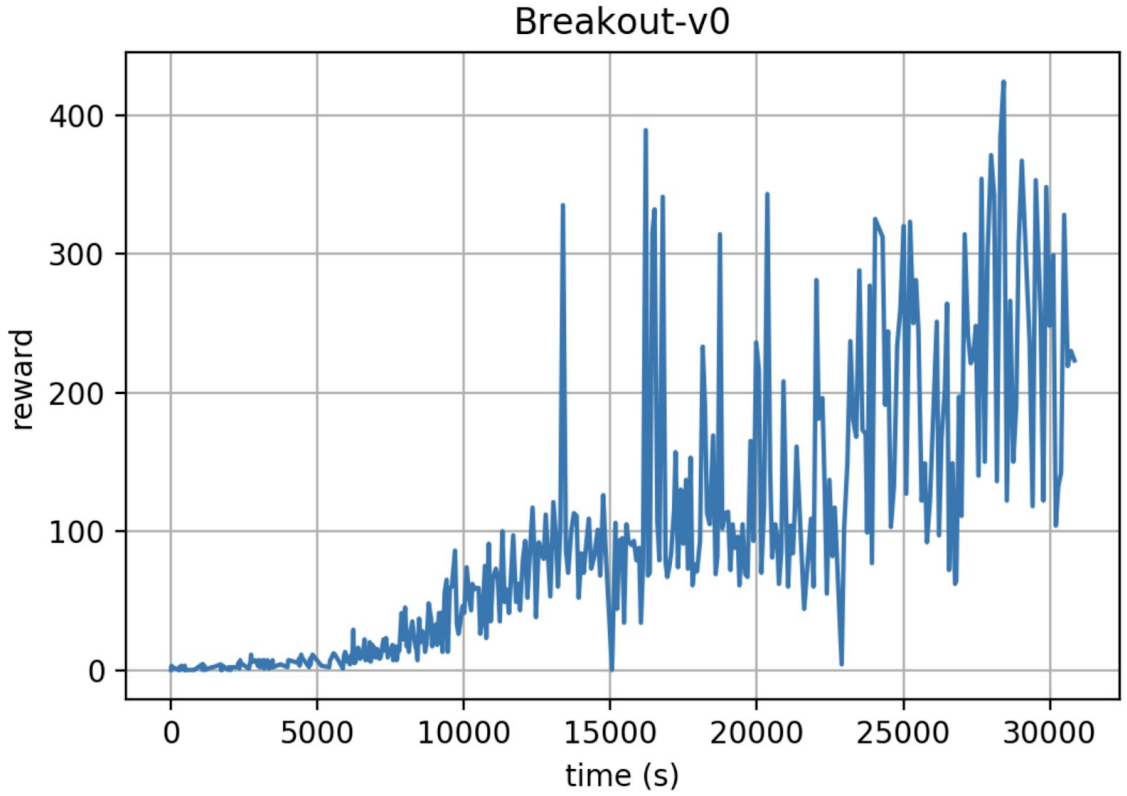


- **Pong-v0 training curve**



使用A3C跟[HW 4-1]相比有較好的performance，reward在約五小時 (16000秒時)的訓練後可以經常達到10以上，[HW 4-1]總共Training時間約為一天，相比之下有非常顯著的提升。

- **Breakout-v0 training curve**



使用A3C跟[HW 4-2]相比同樣有較好的performance，reward在約兩個多小時(10000秒)的訓練後可以穩定的突破baseline (40)，在四個多小時(16000秒)後reward大多都可以破百以上，[HW 4-2]約共training 15小時。

- **Improvement**

使用A3C同步的去玩遊戲（前面Learning Curve皆為A3C，故在Improvement加入A2C作比較），相對於A2C有較好的performance，以下為A2C對比A3C之learning curve之差異（左為A2C，右為A3C，X軸為訓練時間，單位為秒），在同樣的時間下，A2C reward約莫落在一百左右，A3C則達兩到三百。