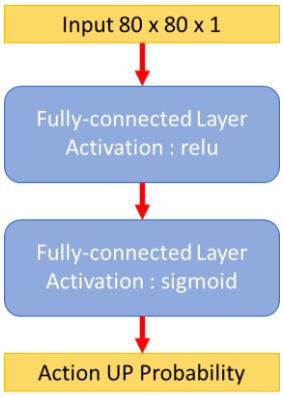
MLDS HW 4-1 Policy Gradient

A. Model description (Policy Gradient model)

首先將從 environment 拿到的前後兩個 frame 相減，形成 size 為 80 x 80 x 1 的 tensor 作為 input 丟進如下的 model；model 將輸出 action UP 的機率。



而訓練方式為:

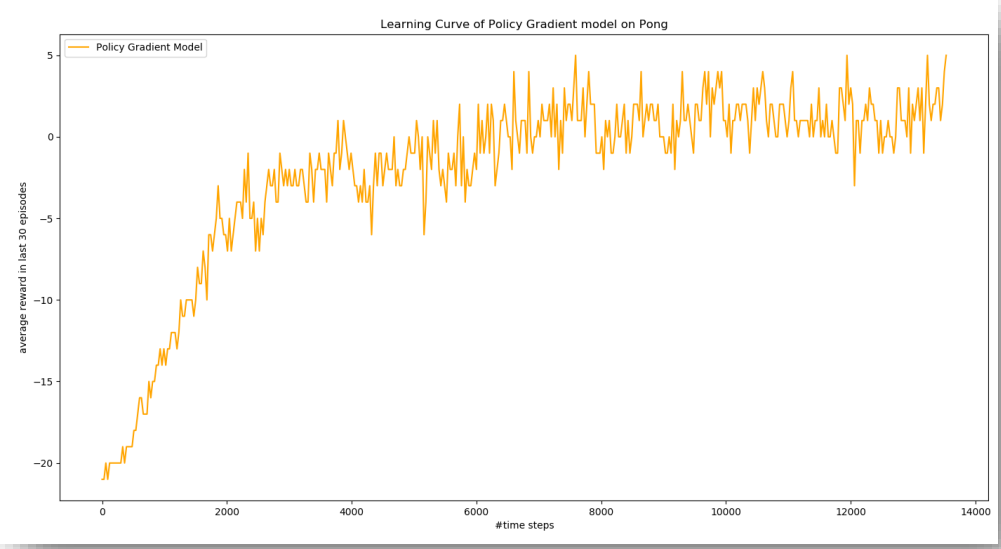
i. 每經過若干個 step 作為一個 batch，對每個 batch 計算一個 discounted reward value: 將 batch 中每個 step 所得的 reward 加上 [從這 step 算起未來 steps 的 rewards 總和* discount factor(用以表示對於未來的不確定性)]。而此 model 的訓練目標為最大化該 discounted reward value 的 expectation value

ii. Model 會利用 output 的機率、sample 出的 action 和計算而得的 discounted reward value 來計算 loss 和 gradient，並利用 optimizer Adam 來做 training。

Details of Model

Parameter	Value
Learning Rate of Adam Optimizer	1.0e-3
Discount (Gamma)	0.99

B. Performance of Policy Gradient model on Pong



C. Improvement

a. Tip Description: Proximal Policy Optimization

Proximal Policy Optimization 用以解決 policy gradient model 不好確定 learning rate(or step size)的問題；因為如果 step size 過大，policy 會一直跳動不易收斂，但若 step size 過小，則會花費過多時間等待 training 的完成。PPO 利用 New Policy 和 Old Policy 的比例限制 New Policy 的 update 幅度來讓 policy gradient 對稍微大一點的 step size 較不那麼敏感。

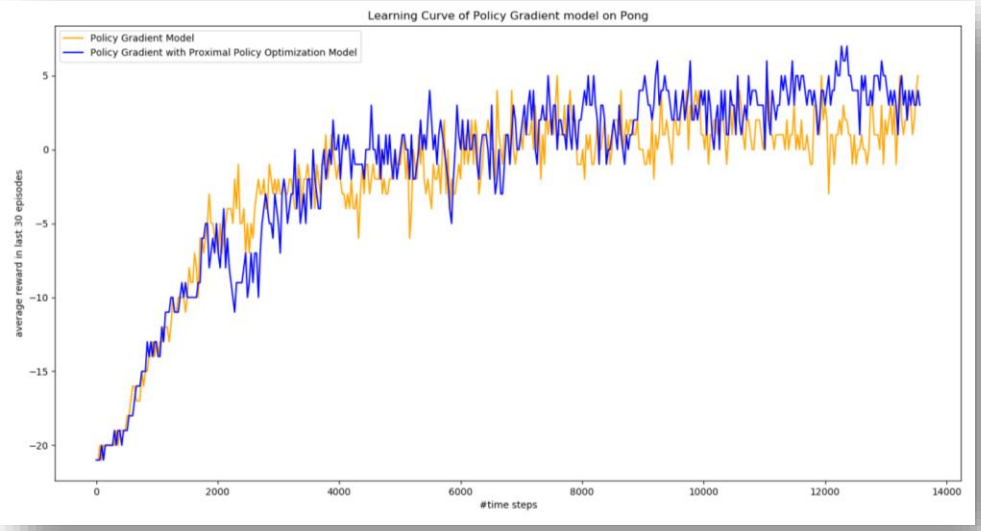
而在基礎的 PPO 上，使用新的 objective function 如下:

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)\right]$$

- θ is the policy parameter
- $\hat{E}_t$ denotes the empirical expectation over timesteps
- r_t is the ratio of the probability under the new and old policies, respectively
- $\hat{A}_t$ is the estimated advantage at time t
- ε is a hyperparameter, usually 0.1 or 0.2

此 objective function 實現了與 Stochastic Gradient Descent 相容的 Trust Region update 方法，同時移除了 KL penalty 和建構 adaptive updates 以簡化演算法。

b. Performance of Proximal Policy Optimization model on Pong



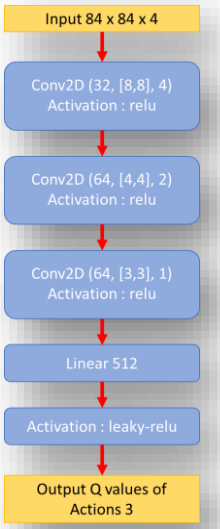
c. Compare to the vallina policy gradient

由 b.中的 learning curve 觀察得知，使用 PPO 的 model 其收斂速度相較原先 vallina policy gradient model 的收斂速度較快，雖然並不顯著；而在兩個 Model 皆進入穩定收斂過程後，可明顯觀察出使用 PPO 的 model 能比 vallina policy gradient model 獲得更高的 reward，確實能使 agent 在遊戲過程中進行更好的 action 判斷。

MLDS HW 4-2 Deep Q Learning

A. Model description (DQN Model)

DQN Model 包含 target network 以及 online network 兩個擁有相同 network 架構的網路；其 network 架構如下:



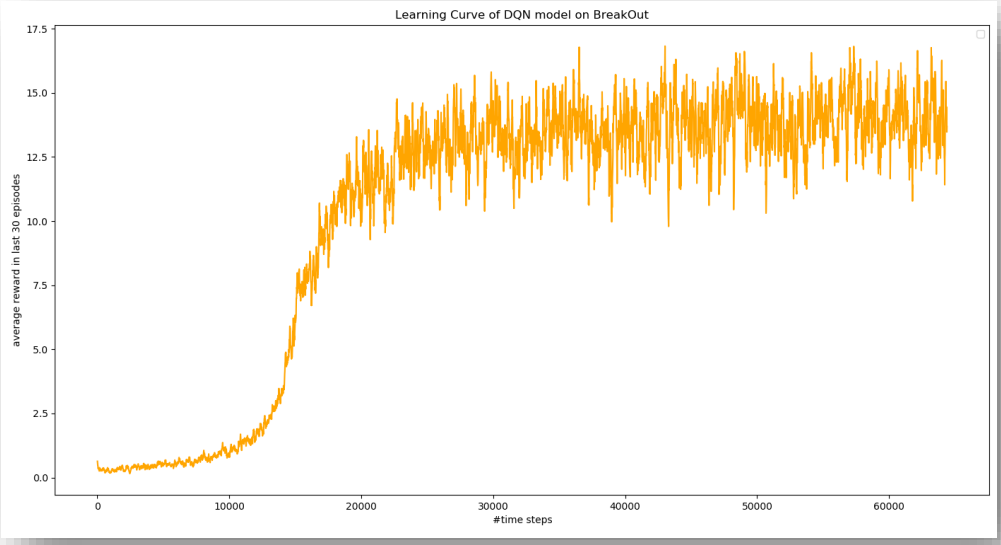
而訓練方式為:

a. 首先，先將 84 x 84 x 4 的 input tensor 餵進 online network 中以估計 Q 值，以某一 probability distribution 隨機選取 action 或對 Q 值取 argmax 選擇 action 以探索環境，將 take action 後所得之 observation 存進 buffer 中，以供後續 network learning 使用。
b. 在每 #Perform Update Current Network Step (=4)後，由 buffer 中 sample 出 32 個 observations
c. 將 take action 後所得的 reward 加上前一個 state 的 Q 值、與 current online network 估計而得的 Q 值以計算 MSE loss value y
d. 利用 c. 所得 y 值利用 optimizer RMSProp (decay=0.99)以 update online network 的參數
e. 在 #Perform Update Target Network Step(=1000)後將 online network 的參數 copy 給 target network

Details of Model

Parameter	Value
Learning Rate of RMSProp Optimizer	1.0e-4
Discount (Gamma)	0.99
Start Exploration	1.0
Final Exploration	0.05
Exploration	1000000
Replay Memory Size	10000
Perform Update Current Network (Online Net) Step	4
Perform Update Target Network Step	1000
Batch Size	32

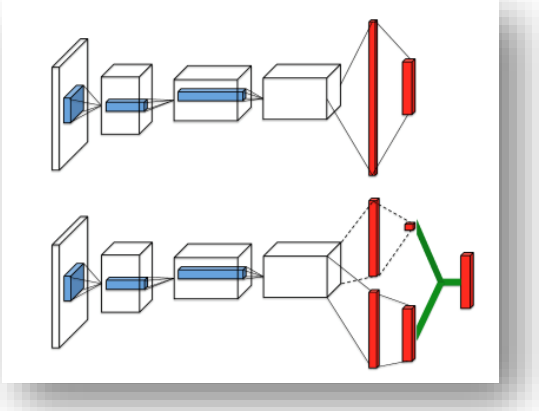
B. Performance of DQN model on Breakout



C. Improvement

a. Tip Description: Dueling DQN model

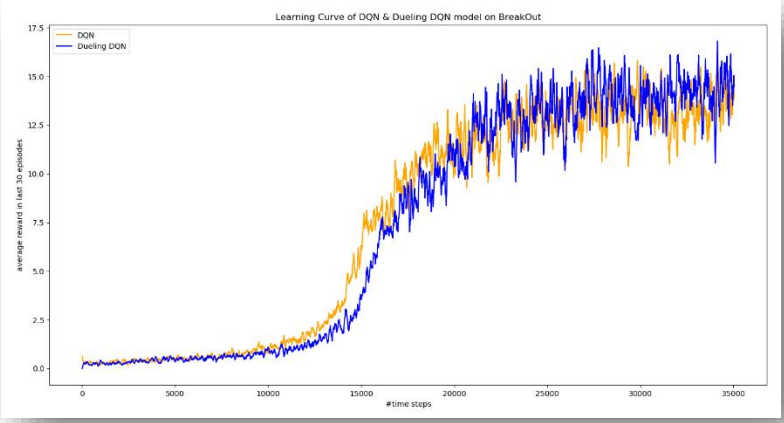
在遊戲中存在著許多 state，不同的(state, action) pair 具有不同的 Q value，但在某些 state 下，無論採取何種 action 都對 Q value 影響不大；在這些情形下計算 action 的價值函數意義並沒有 state 的狀態函數來得大。Dueling DQN 便是基於以上的想法而被提出的一種競爭式的 DQN model.



上圖中的第一個模型為一般的 DQN model，而第二個模型則為 Dueling DQN model；一般的 DQN model 直接輸出的是每個 action 的 Q value，但 Dueling DQN model 則將原先的 output 一分為二，用以分別預測 state value $V(s;\theta,\beta)$ 和 advantage for each action $A(s,a;\theta,\alpha)$，最後再將兩者結合$Q(s,a;\theta,\alpha\,\beta) = V(s;\theta,\beta) + A(s,a;\theta,\alpha)$ 輸出 Q value。

由於 Dueling DQN model 是一個 end to end 的 training network，並不存在單獨訓練 state value V or advantage for each action A 的 value functions，因此 Dueling DQN model 的訓練方式與原先 DQN model 是沒有區別的。

b. Performance of Dueling DQN model on Breakout



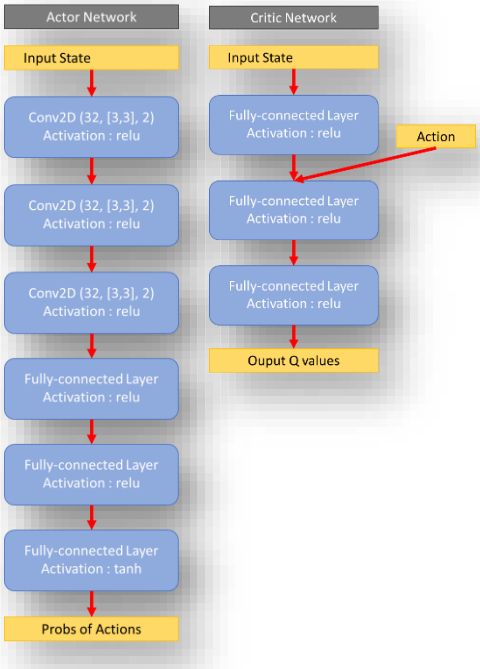
c. Comparison to original DQN model

由 b.中的 learning curve 觀察得知，Dueling DQN model 與原先 DQN model 的收斂速度相差無幾，並不如原先所預期的 Dueling DQN model 會帶來較為顯著的改進；推測其因可能為 Breakout 遊戲較不具備" 但在某些 state 下，無論採取何種 action 都對 Q value 影響不大"的情形，故 Dueling DQN model 在學習速度上的表現不顯著。但在兩個 Model 皆進入穩定收斂過程後，可明顯觀察出 Dueling DQN model 能比 DQN model 獲得更高的 reward，確實能使 agent 在遊戲過程中進行更好的 action 判斷。

MLDS HW 4-3 Actor Critic

A. Model description (Actor-Critic model on Pong & BreakOut)

整個 model 分為如下的 Actor network 和 Critic network:



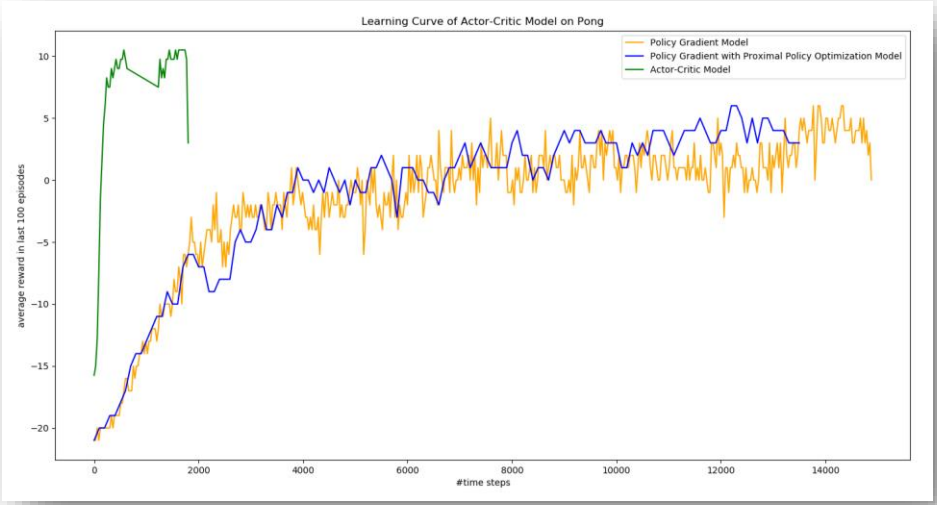
而訓練方式為:

i. 先累積 experience replay buffer 到達 minibatch 的指定個數，然後根據 sample 分別訓練以上的 Actor Network 和 Critic Network
ii. Model 會先透過 reward 來更新 Critic Network 的參數θ^Q，再依據 Critic Network 對於 action 的評分(Q value)調整 Actor Network 的參數 θ^μ
iii. 利用 update 過後的 Critic Network 參數θ^Q以及 Actor Network 參數 θ^μ，透過參數 τ 按照比例更新到 target network

Details of Model

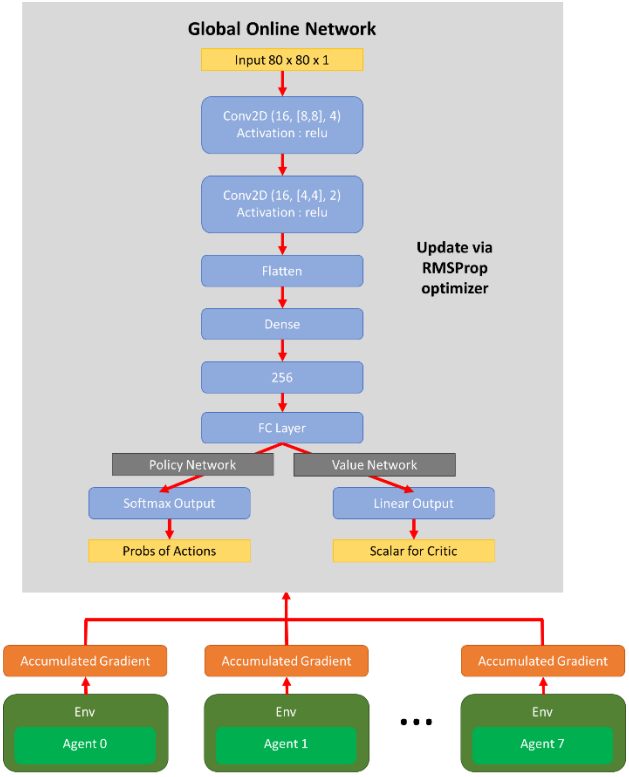
Parameter	Value
Learning Rate of RMSProp Optimizer	1.0e-3
Discount (Gamma)	0.99

B. Performance of Actor-Critic model on Pong



C. Improvement

a. Tip Description: A3C (Asynchronous Advantage Actor-Critic) model



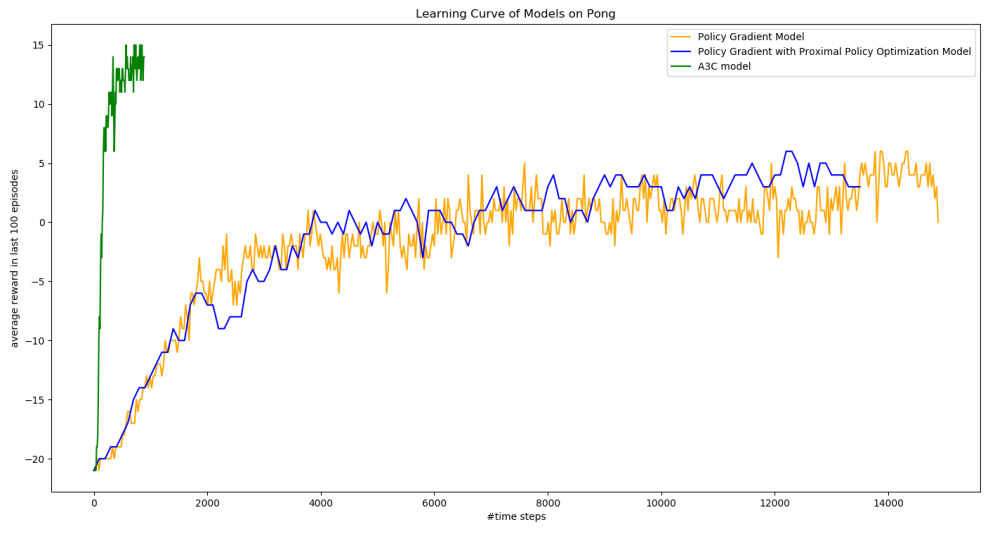
可將整個 model network 架構分為兩個部分: 一個 global network 及數個 agent networks；而兩種 network 具有相同的 structure 如上圖中的 global online network 所示: input 會先經過兩層 Conv2D layer, 再經過一層 Flatten 及 Dense layer 後形成 256 維的 vector。接著該 network 會分成 policy network 及 value network 兩部分；policy network 會將 256 維的 vector 過一層 FC(Fully-connected) layer，並利用 softmax function output 出一包含各 action probability 的 vector；而 value network 則會將 256 維

的 vector 經過一 FC (Fully-connected) layer 後計算出一個 scalar 作為評估 current state 好壞的依據。

而訓練方式為:

- Model 會先產生一個 global network 與其他 8 個 thread 的 agent network
- 每個 agent network 會各自與環境作互動，再分別計算各自的 loss (= policy loss + value loss* 0.5)及 gradient
- 利用 accumulated gradients 和 RMSProp Optimizer 去 update global network 的參數，再將 global network 的最新參數 copy 回各 agent network 中，使各個 agent network 得以繼續 explore environments.
- 利用 update 過後的 Critic Network 參數θ^Q以及 Actor Network 參數 θ^μ，透過參數 τ 按照比例更新到 target network

b. Performance of Proximal Policy Optimization model on Pong



D. 分工

由於另兩位同學決定退選，故此次作業皆由一人 b03901156 完成