MLDS HW3 - Game Playing

學號：B04901066 系級：電機三 姓名：洪國嘵

Basic Performance (6%)

Policy Gradient model (1%)

reference：https://github.com/mrahtz/tensorflow-rl-pong

前處理：210x160x3轉成一維6400向量

model：

input：一維6400

hidden layer：200unit的Fully Connected Layer

output：向上的機率

training：

不斷重複：

初始化。

每個episode中：

前處理observation(變成與上個observation之delta)，計算向上機率，依其機率step，取得新的狀態，紀錄batch、state、action、reward。

若episode_n是batch_size_episodes的整數倍：將reward作discount處理再標準化，fit model，清空紀錄。

testing：

前處理observation(變成與上個observation之delta)，model預測之機率高者為輸出。

DQN model (1%)

reference：https://keon.io/deep-q-learning/

前處理：無

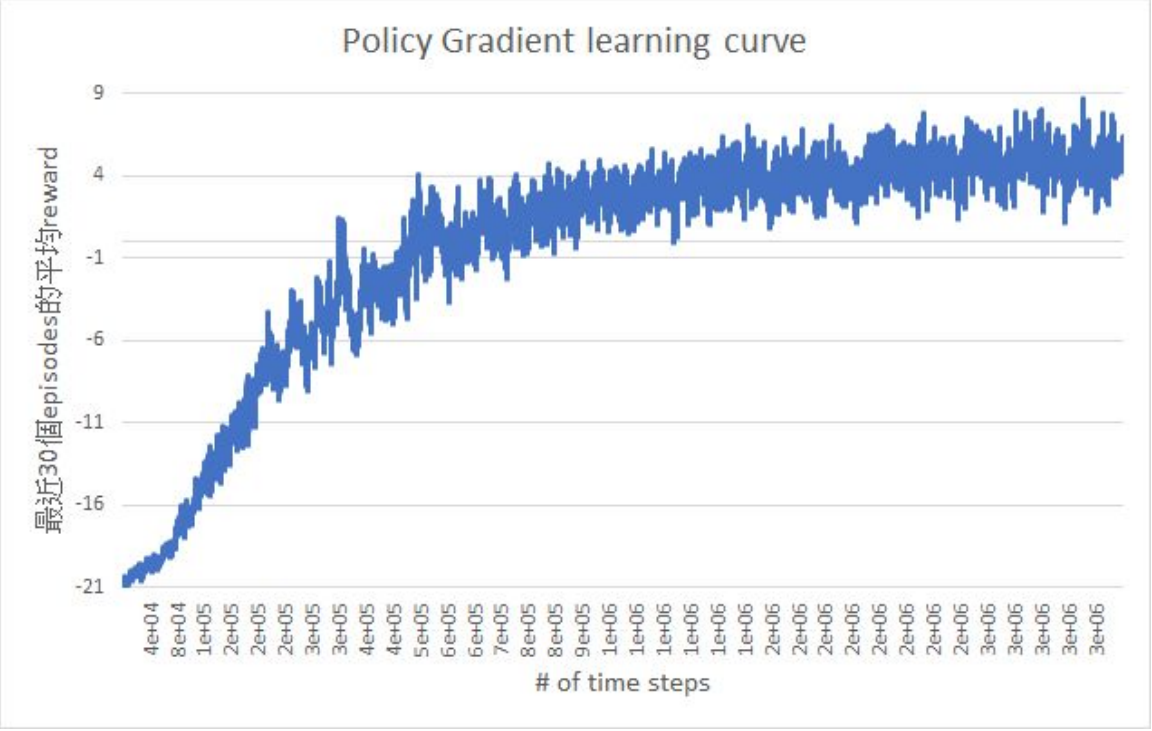
model： input：84*84*4；output：四個動作之機率

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 20, 20, 32)	8224
conv2d_2 (Conv2D)	(None, 9, 9, 64)	32832
conv2d_3 (Conv2D)	(None, 7, 7, 64)	36928
flatten_1 (Flatten)	(None, 3136)	0
dense_1 (Dense)	(None, 512)	1606144
leaky_re_lu_1 (LeakyReLU)	(None, 512)	0
dense_2 (Dense)	(None, 4)	2052

Total params: 1,686,180
Trainable params: 1,686,180
Non-trainable params: 0

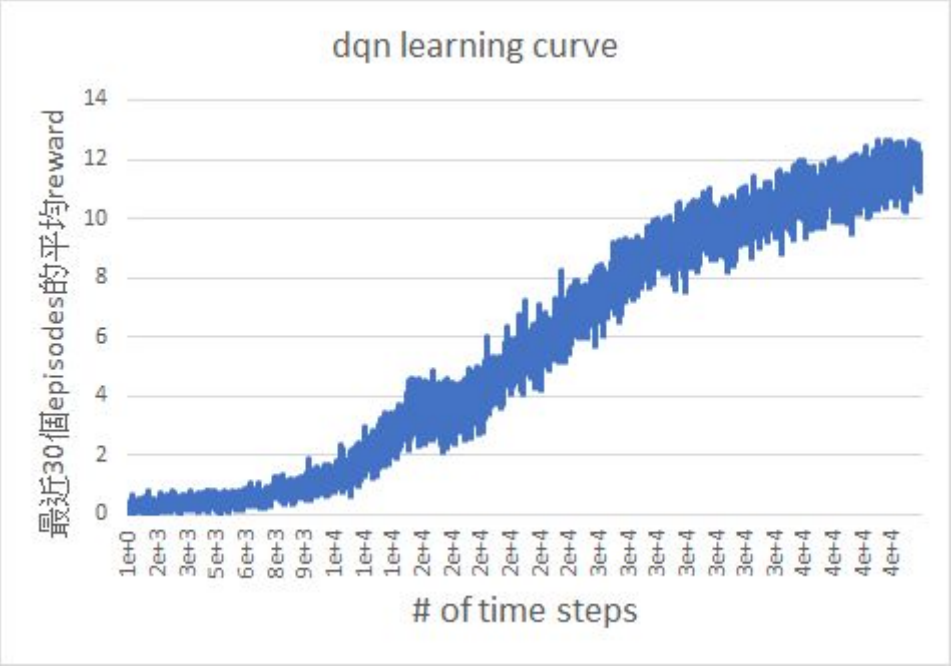
training：
不斷重複：
初始化。
每個episode中：
前處理observation(變成與上個observation之delta)，計算向上機率，依其機率step，取得新的狀態，紀錄batch、state、action、reward。
若total_step大於10000且為4的整數倍：
從記憶體中隨機取出batch_size的紀錄，用這些紀錄給線上model預測，若遊戲結束，這個動作的reward=reward；反之reward=reward加上 以線上model一下個觀察預測之輸出率最大者來從目標model輸出中選擇，乘上gamma。
上述輸出當作目標fit線上model，最後降低epsilon rate。
若total_step大於10000且為1000的整數倍：
更新目標model。
testing：
model依照觀察，預測之機率高者為輸出。

Plot the learning curve to show the performance of your Policy Gradient on Pong (2%)



X-axis: number of time steps；Y-axis: average reward in last 30 episodes

Plot the learning curve to show the performance of your DQN on Breakout (2%)



X-axis: number of time steps；Y-axis: average reward in last 30 episodes

Experimenting with DQN hyperparameters (4%)
未完成。

Bonus
未完成。