R06725048 資管碩一 陳信豪

MLDS HW3 Report

- **Describe your Policy Gradient & DQN model**

  [Policy Gradient]

  我的 deep learning model 如下，與助教大致相同

  2 CNN + 1 DNN 最後再 output

  Activation 全使用 relu (除最後一層是 softmax)

  optimizer 為 Adam (learning rate = 1e-4)，loss 為 categorical_crossentropy

  kernel_initializer 為 he_uniform

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
reshape_1 (Reshape)          (None, 80, 80, 1)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 20, 20, 16)        1040
_________________________________________________________________
activation_1 (Activation)    (None, 20, 20, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 10, 10, 32)        8224
_________________________________________________________________
activation_2 (Activation)    (None, 10, 10, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               409728
_________________________________________________________________
activation_3 (Activation)    (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 387
_________________________________________________________________
activation_4 (Activation)    (None, 3)                 0
=================================================================
Total params: 419,379
Trainable params: 419,379
Non-trainable params: 0
_________________________________________________________________
```

  我們餵 action 給 env 後會拿到 observation、reward、done 等資訊，而我餵給 DL model 的是兩個 env_step 回傳的 observation 的差值 (state)，輸出為動作（我有將重複的六個 action 簡化成三個，所以 DL model 最後輸出只有三維）

  train 時，要 make action 時，會先取出 model prediction 的機率分佈，然後再根據這個機率分佈 sample 出一個動作出來

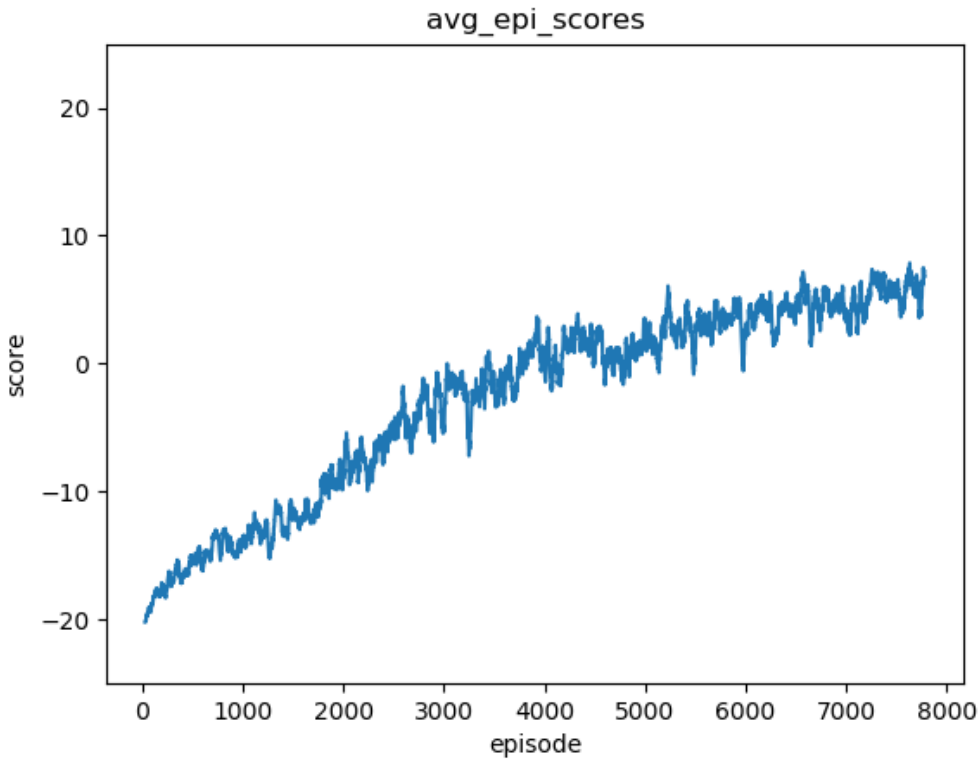
  （test 時，則是直接取機率最大的）

然後為了使可以獲得比較好的 reward 的 action，下次被選中的機率更高，我們從真正選擇的 action 與 model predict action 的機率分佈算出 gradient，並將這個 gradient 乘上該次的 discounted-normalized reward，再將之乘上一個 learning rate 後加回原本的 model predict action 機率分佈當作 model 要吃的 label

最後我的 model 會在每次 done 時，將此次 episode 視為一個 mini-batch，將其中 step 的 state （observation 差）當作 X，以及上面所說的 label 當作 y 來 train

[DQN]
train 不起來 …

- **Plot the learning curve to show the performance of your Policy Gradient on Pong**



- **Plot the learning curve to show the performance of your DQN on Breakout**
- **Experience with DQN hyperparameters**