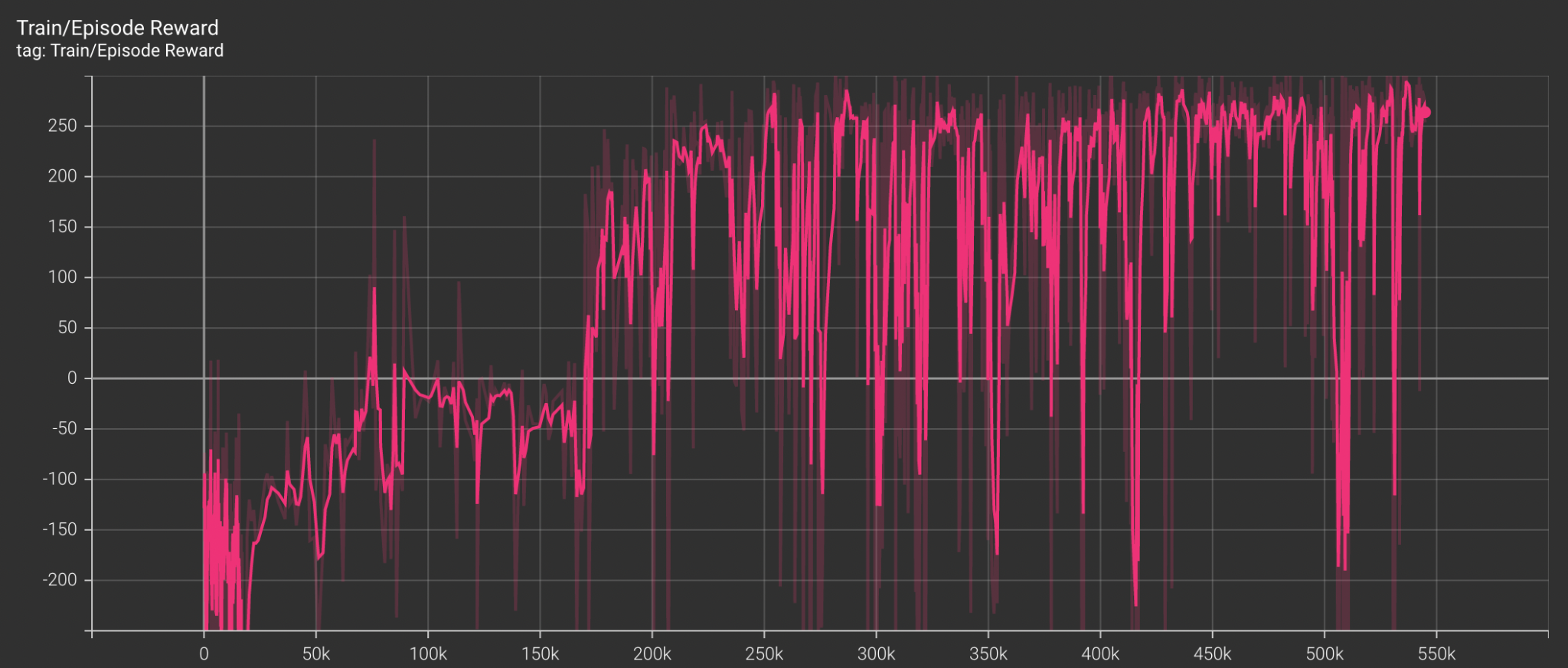
◼A tensorboard plot shows episode rewards of at least 800 training episodes in LunarLander-v2

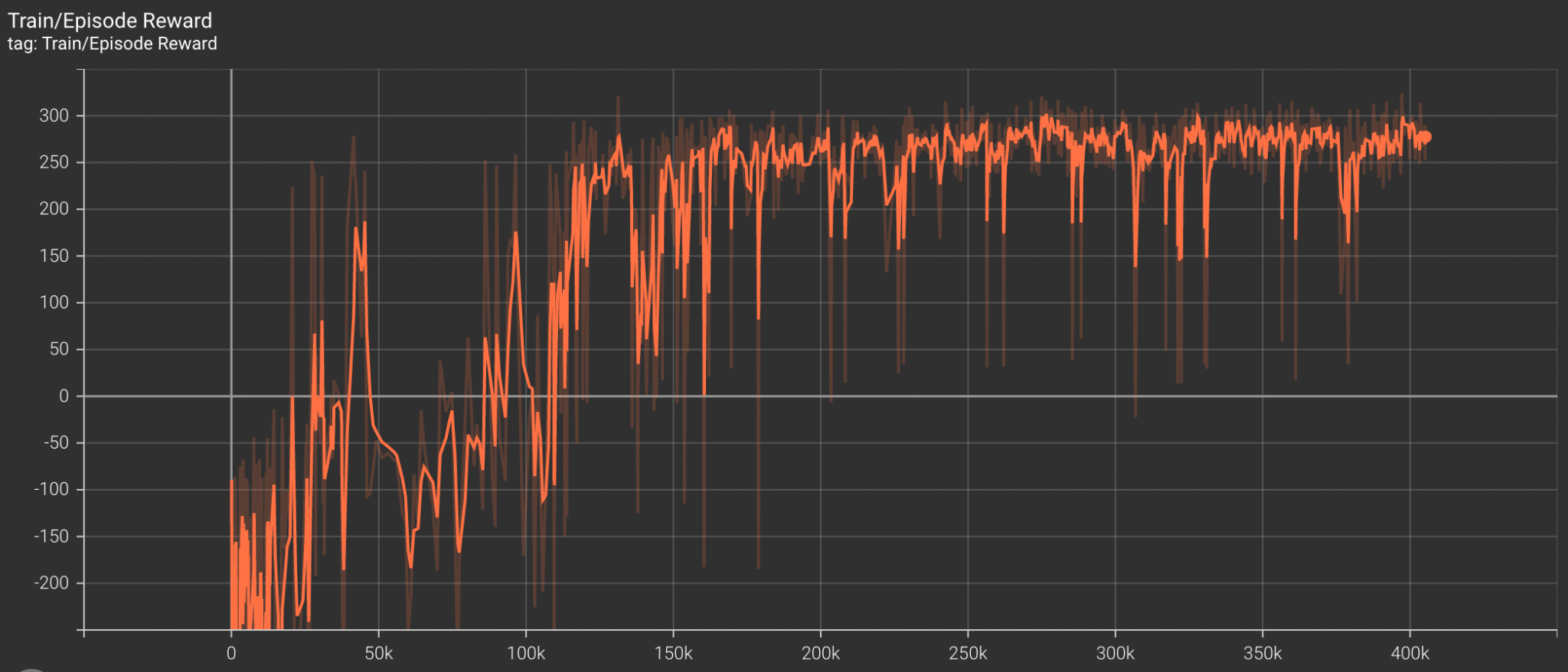
DQN:

episode=1500

◼ A tensorboard plot shows episode rewards of at least 800 training episodes in LunarLanderContinuous-v2

DDPG:

episode=1500



◼ Describe your major implementation of both algorithms in detail.

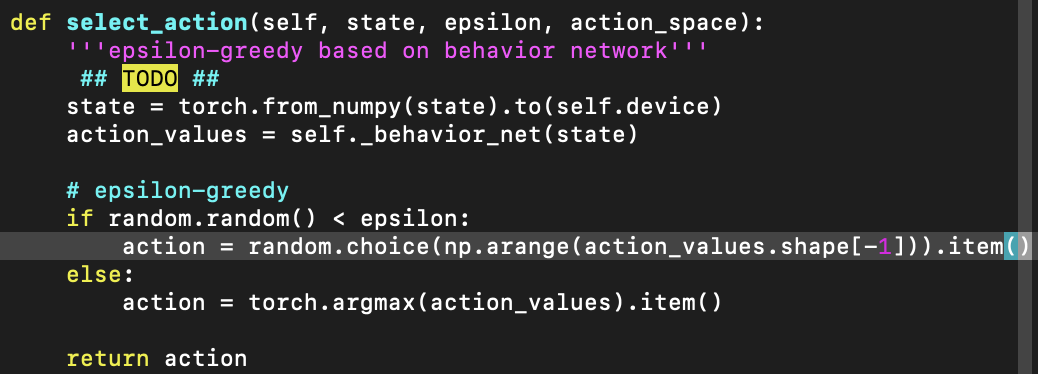
DQN:

DQN是value based algorithm。

首先要先建個behavior network和target network下去計算action的value。

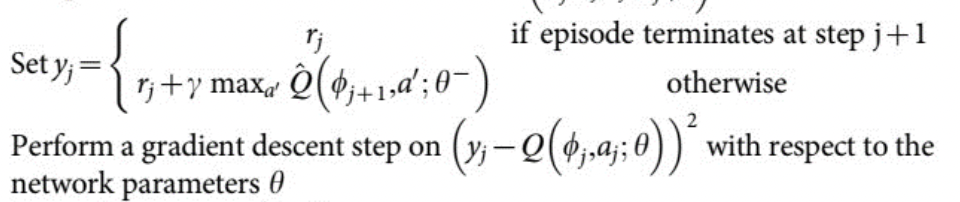


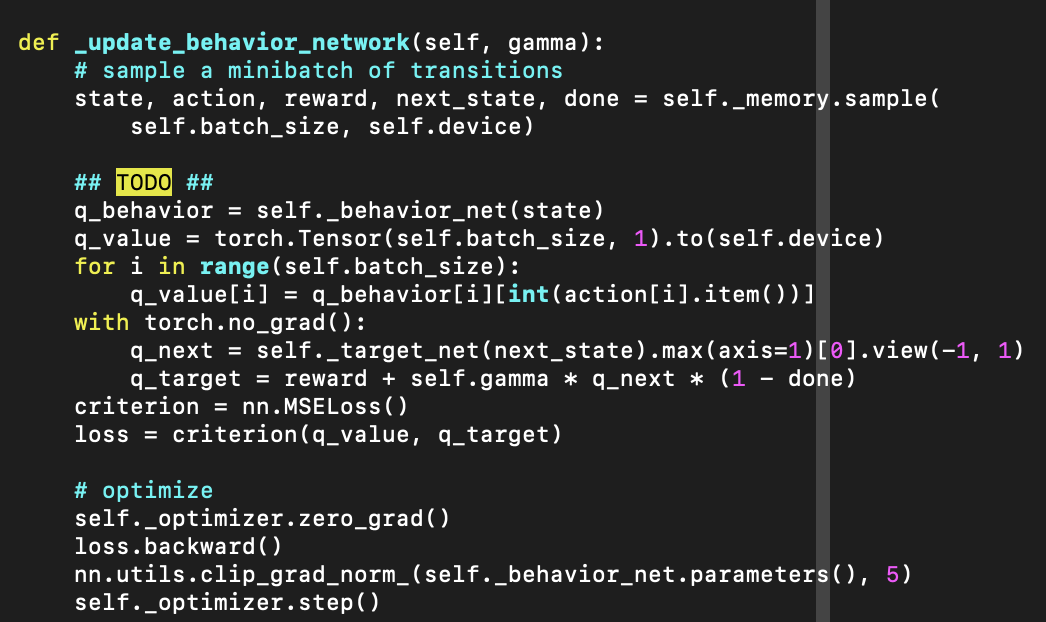
接著是action的選擇，由於是value based，所以這邊先去計算每個action的value後，再做epsilon greedy，也就是有epsilon的機率隨便選一種action，1-epsilon的機率選value最大的機率。



接著是update behavior net的部分，就照著演算法下去改。

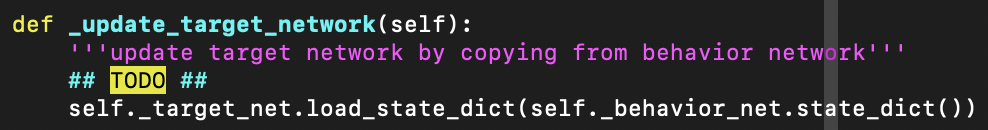
具體來講就是reward+q\_target(next\_state)=q\_target(state)，然後和q\_behavior(state)下去做mse。





接著是update target net，也就是每經過一定次數的step後，將behavior net的參數直接更新到target net上。



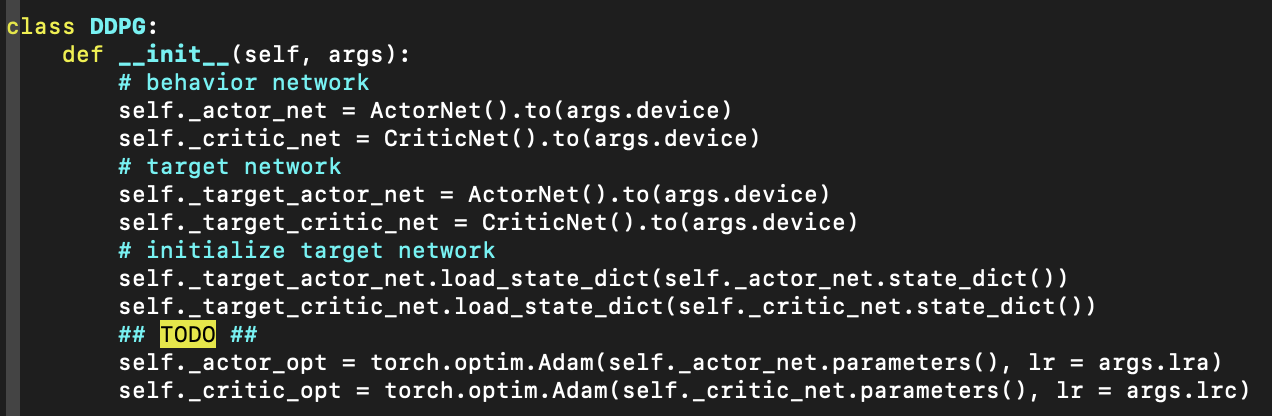


至於network和hyperparameter魔改的部分，留到後面說明。

DDPG:

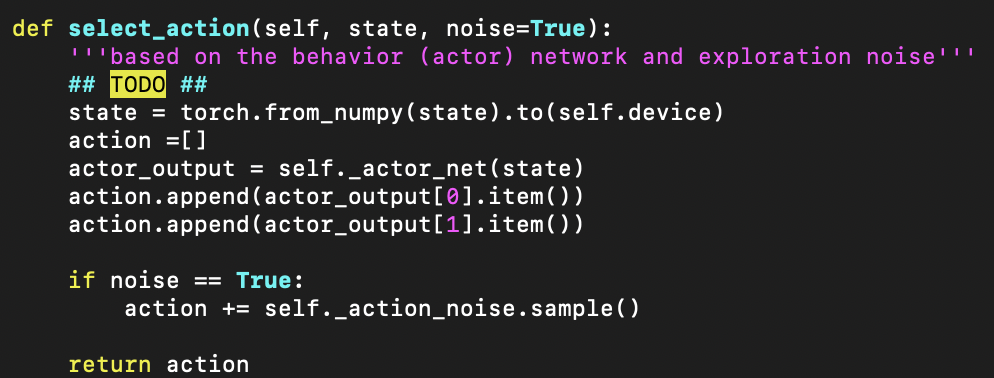
DQN是method based algorithm，也就是policy based和valued based的混合版。

首先要先建個critic network和actor network下去計算value和action。  
然後這兩個還要在分別建個target network，總共共4個network。



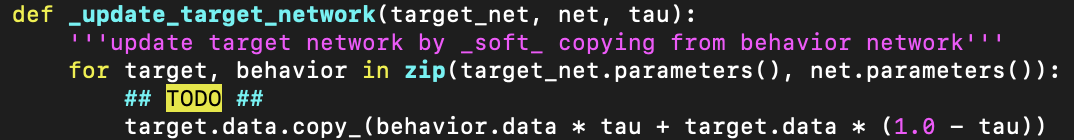
至於動作的選擇，就是靠actor network來決定。

然後為了能做到探索的效果，所以就在output加上noise。



然後critic 和 actor的update，留到後面做說明。

接著講一下如何update target network，跟dqn不同的是，ddpg是做soft update，也就是每次更新時，是照著原target network和local network的parameter下去做加權平均。其中tau是一個很小的數字，代表更新時，target network不會有很大的變動。

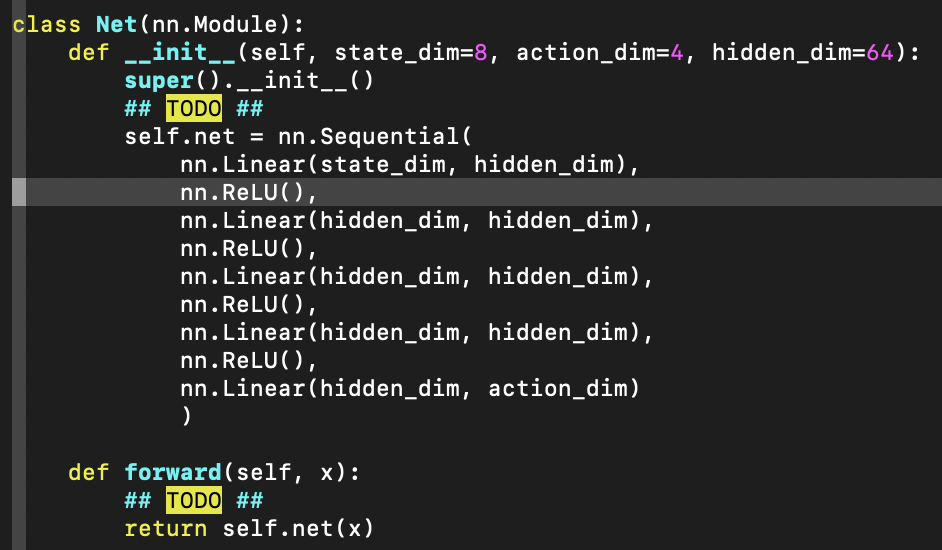


至於network和hyperparameter魔改的部分，留到後面說明。

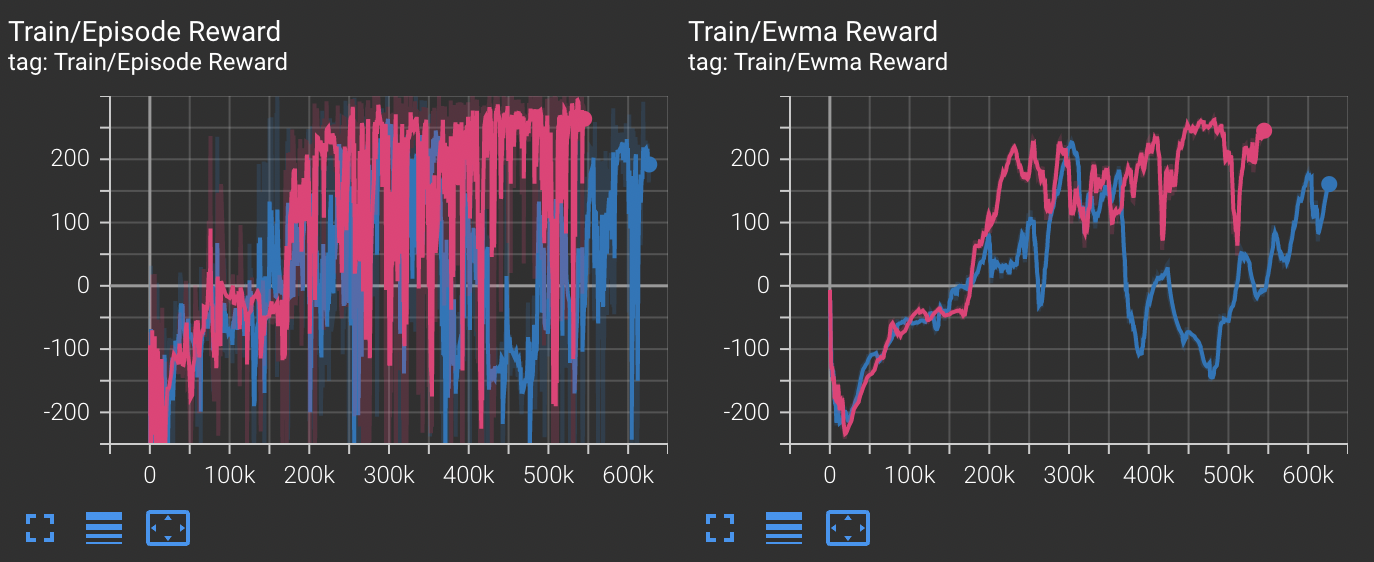
◼ Describe differences between your implementation and algorithms.

主要是改network和hyperparameter，hyperparameter留到bonus做說明。

首先是dqn的network:



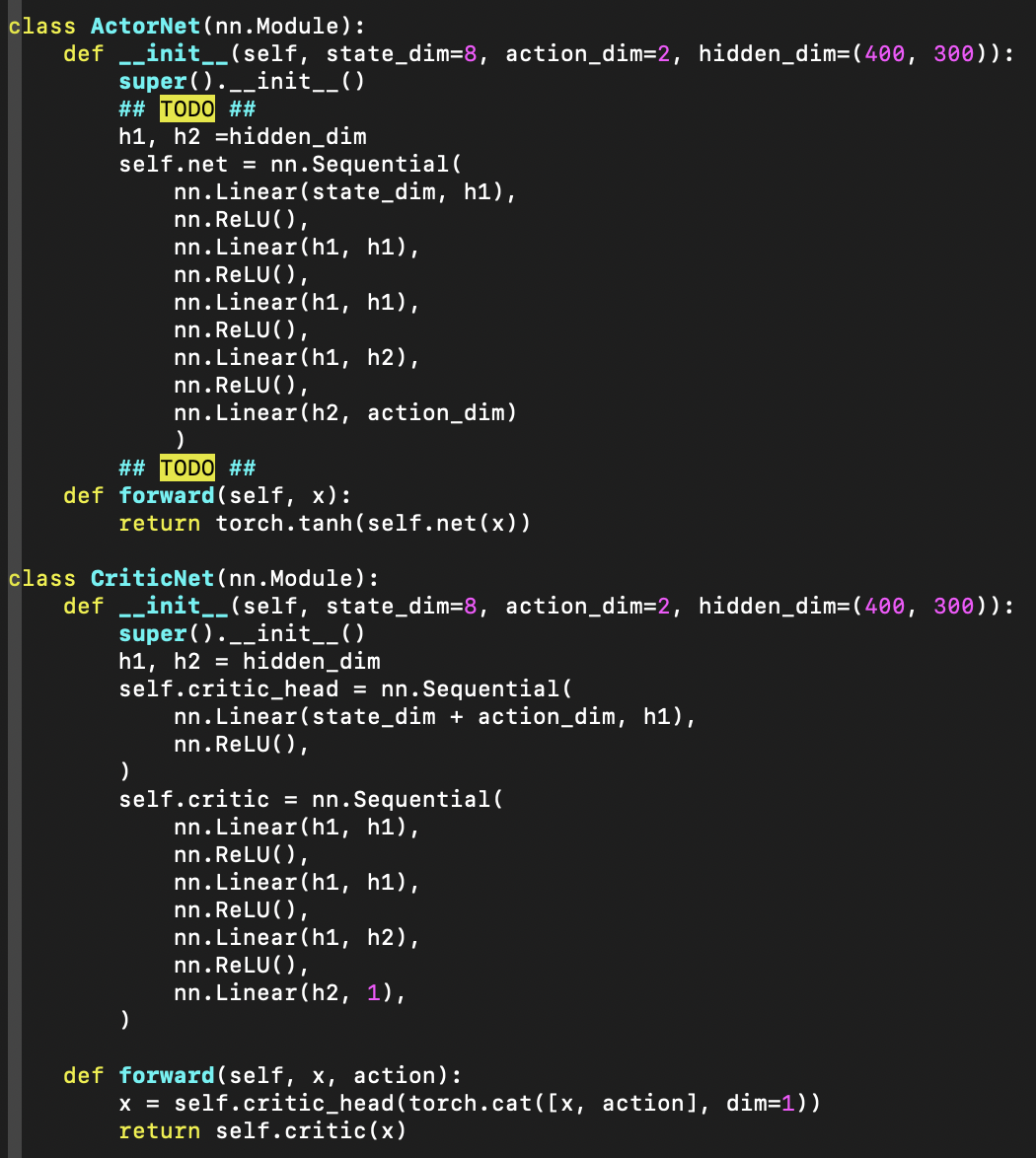
把原本的三層網路變成五層，然後hidden\_dim從32變成64。



可以看到紅色練起來果然有比原本藍色好一些。

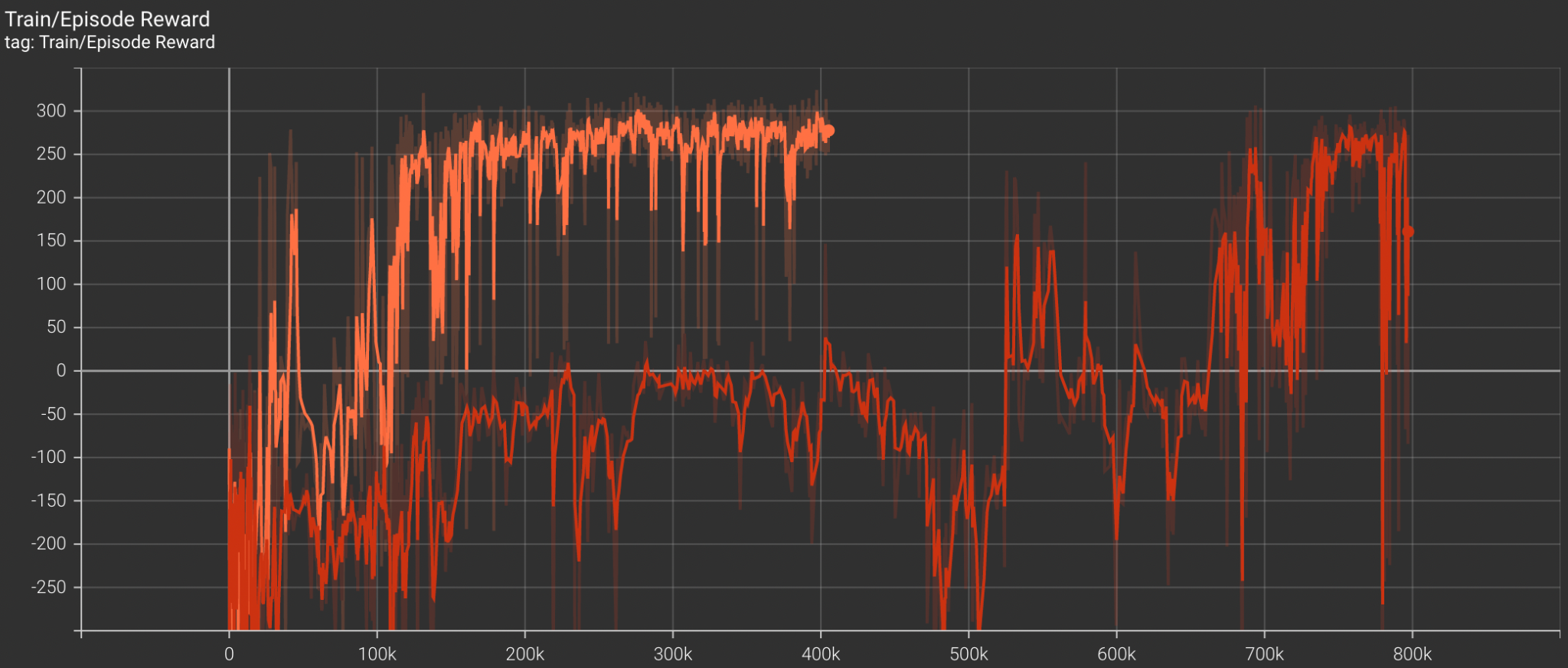
接下來是ddpg:

我就直接將network無腦堆個5層和4層fc，效果顯著。



紅線是原本的network，橘線是我魔改後的，可以發現reward穩定很多。

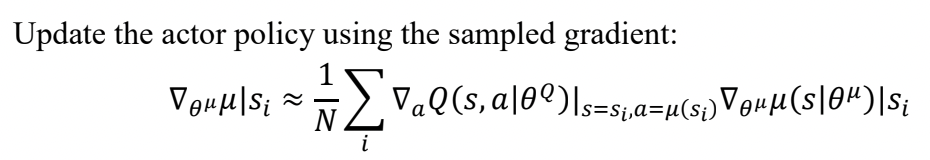
至於hyperparameter怎麼調，留到後面談。



◼ Describe your implementation and the gradient of actor updating.

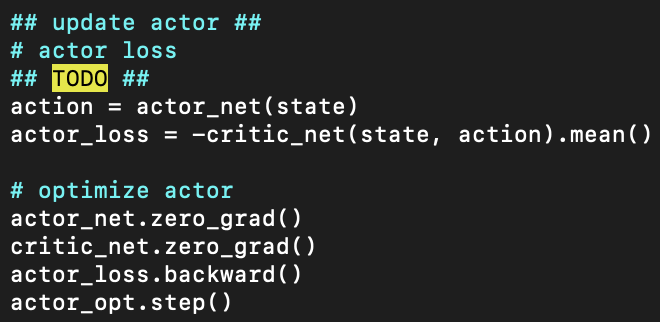
藉由計算policy gradiant，去更新actor network。

policy gradiant公式如下：



然後因為希望能收斂到max Q，反過來說就是希望能做成min -Q，所以要做gradiant ascent，要多加個負號。

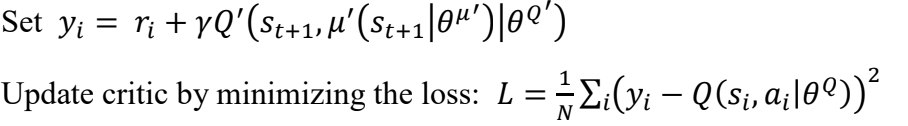
實作如下：



◼ Describe your implementation and the gradient of critic updating.

公式如下：

其實精神跟dqn更新behavior net幾乎一模一樣，只差在behavior net選action時是挑value最大的那個action，而ddpg是直接用actor output action。

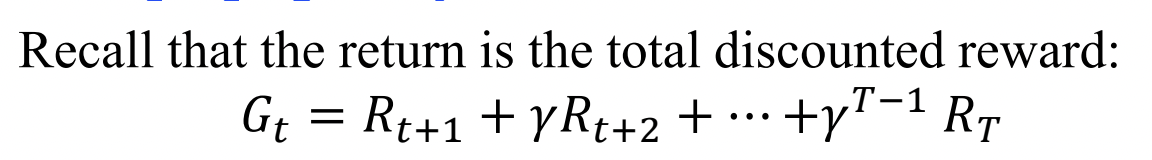


實作如下：



◼ Explain effects of the discount factor.

在RL中，常會用對未來的期望值來做出action，但問題是理論上現在的action對越未來的reward應該影響會越小，好比說國小用功讀書不代表升大學學測能考好，所以加上個discount factor比較貼近符合現實。反過來說，如果discount factor接近1，那代表reward較不易因為時間而改變。



◼ Explain benefits of epsilon-greedy in comparison to greedy action selection.

Greedy actoi selection是照著max estimate value下去選擇action，最後不一定能帶來最好的結果，因為estimate value不見得跟value差不多。所以需要額外一點隨機探索的機會， 於是epsilon greedy就誕生了，epsilon greedy會分一點機率epsilon作為隨機選擇action。

◼ Explain the necessity of the target network.

如果沒有target network，這就代表每次更新時，都是用同個網路來estimate value，會造成收斂的不穩定，所以要用一個target network先固定部分的value。然後如果update target network是直接copy的話，那最好update target network的時間長一點比較好。

◼ Explain the effect of replay buffer size in case of too large or too small.

buffer size太大的話，會浪費空間，training的時間也會過長。反之buffer size太小的話，sample太少，可能會overfitting。

⚫ Report Bonus

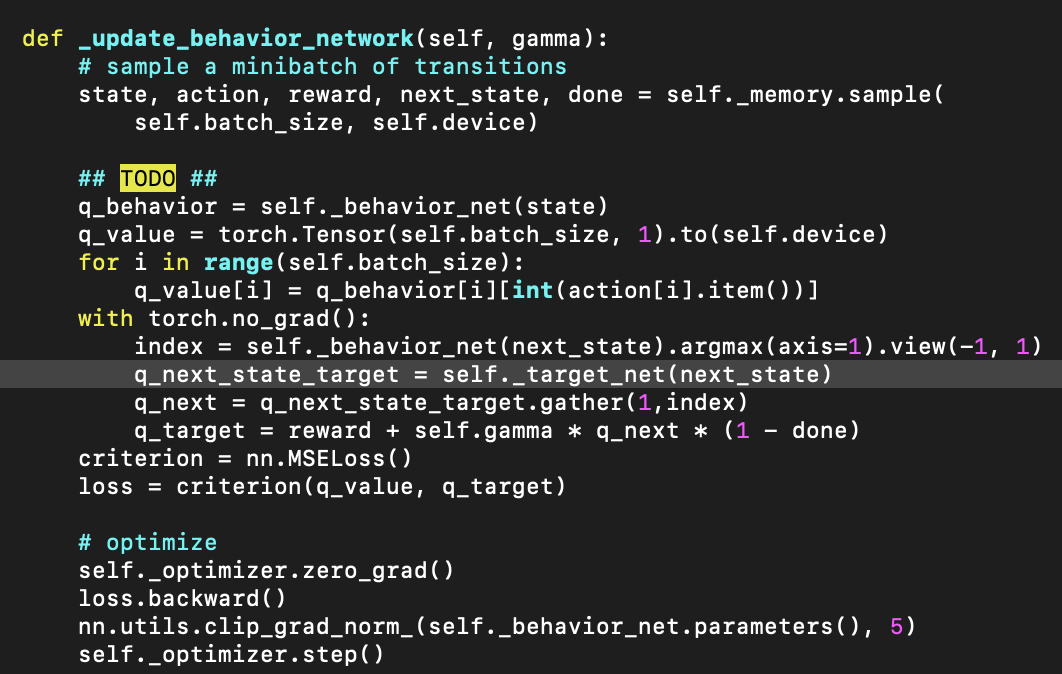
◼ Implement and experiment on Double-DQN

DDQN和DQN的差別只差在update behavior net。

因為DQN的behavior net選了覺得value最大的動作，然後又對其動作後的state取behavior net的對應value，下去對q\_next做update，但實際上這樣的算法會造成overestimate。

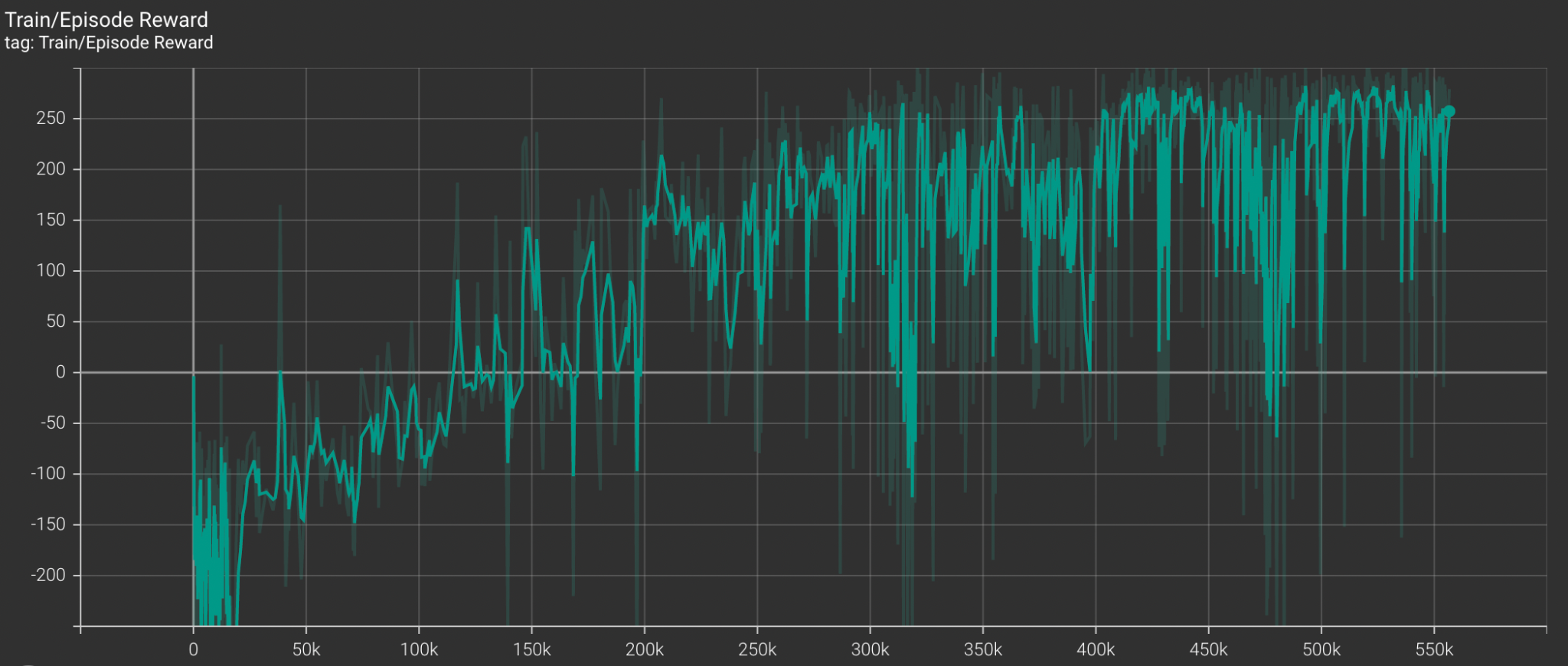
為了不造成overestimate，所以就讓behavior net選了覺得value最大的動作，然後又對其動作後的state取target net的對應value。

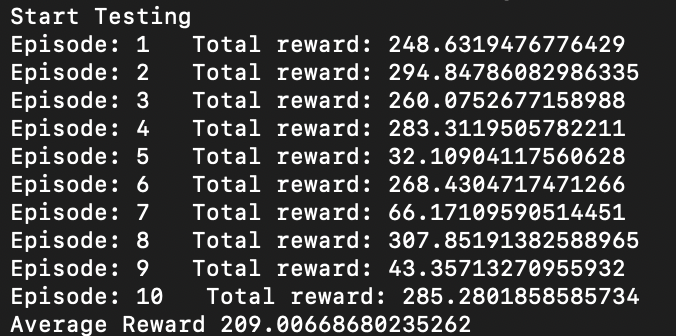
這樣就不會有overestimate的問題。



然後DDQN我沒有魔改network，所以結果就比較普通。

1500 episode:





◼ Extra hyperparameter tuning

DQN:

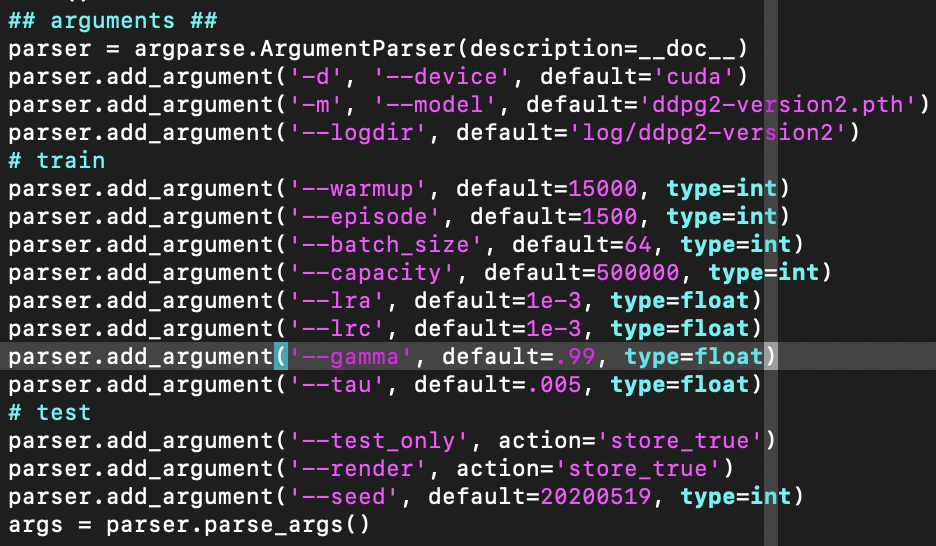
除了調網路，我還有調一些參數。我覺得影響最大的是將target-freq調成1000這件事，沒調這個前是完全train不起來的。此外我把warmup和episode train調長一點，防止有時候收斂太慢的問題。



DDPG:

我把warmup和episode train調長一點，防止有時候收斂太慢的問題。

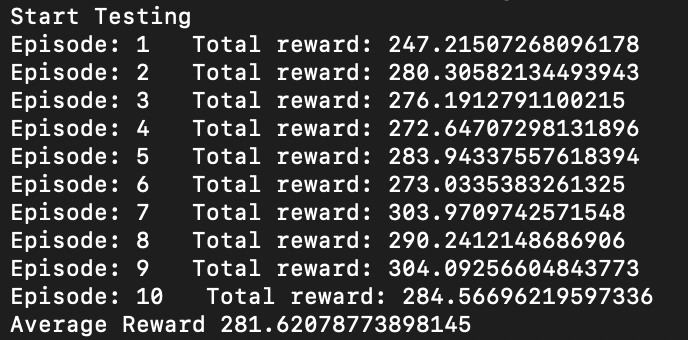
原本想對lr做調整，後來發現lr=1e-3剛剛好。



⚫ Performance

◼ [LunarLander-v2] Average reward of 10 testing episodes: Average ÷ 30

DQN:



◼ [LunarLanderContinuous-v2] Average reward of 10 testing episodes: Average ÷ 30

DDPG:

