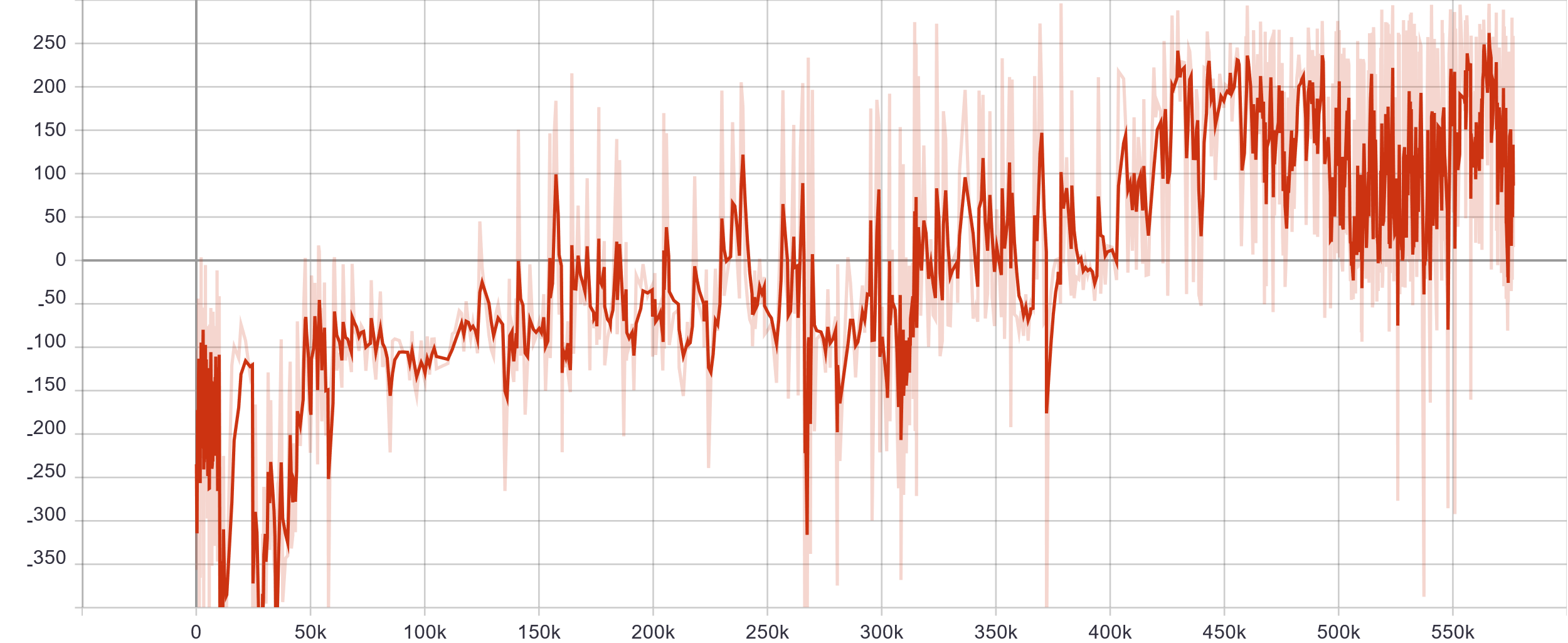
DLP Lab6 Deep Q-Network and Deep Deterministic Policy Gradient

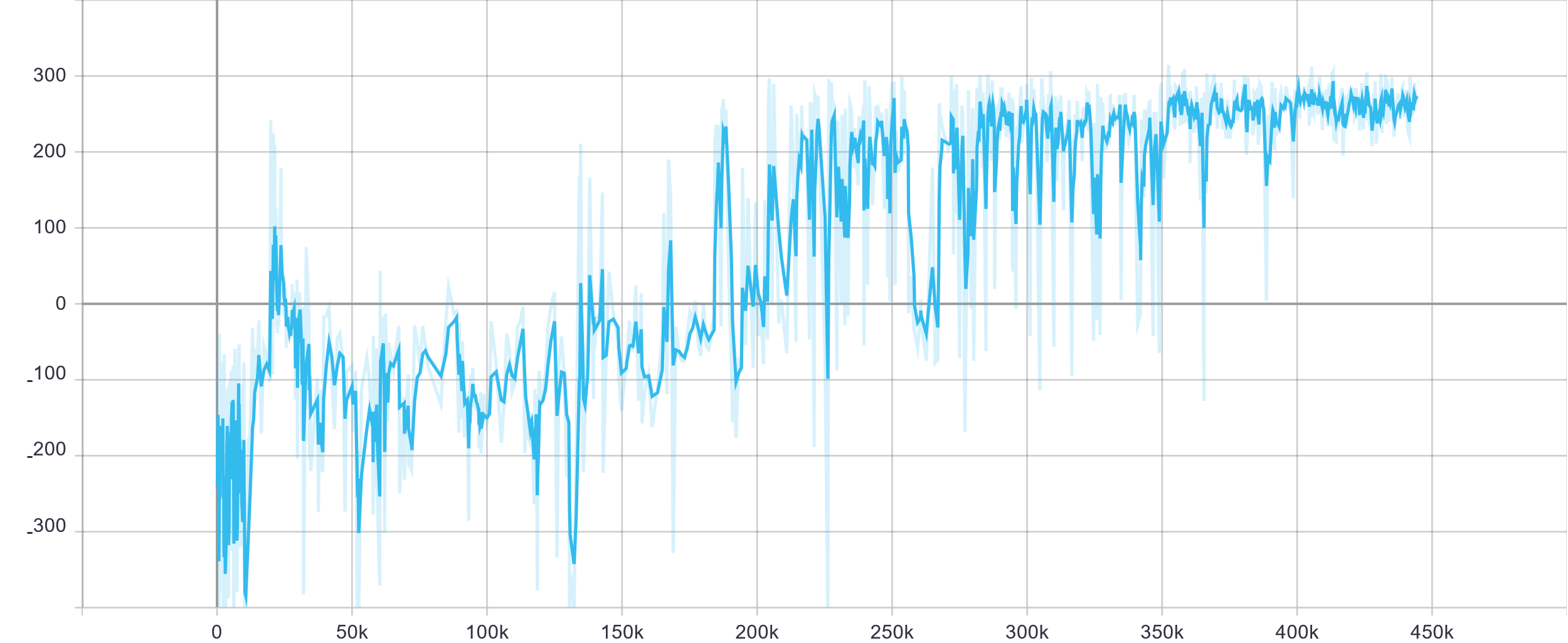
0711529 陳冠儒

**Report**

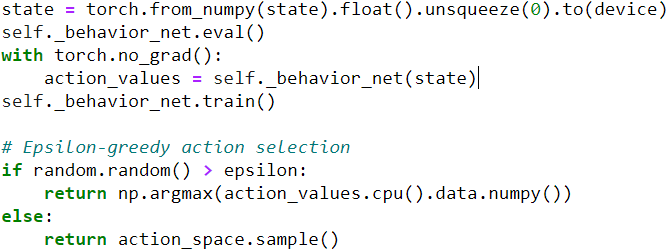
* A tensorboard plot shows episode rewards of at least 800 training episodes in LunarLander-v2(5%).
* Using Soft updating target network



* A tensorboard plot shows episode rewards of at least 800 training episodes in LunarLanderContinuous-v2(5%)

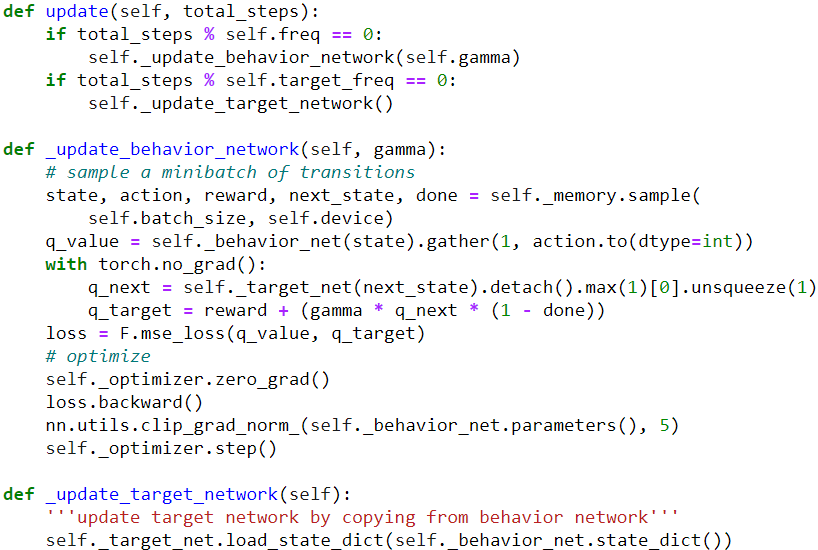


* Describe your major implementation of both algorithms in detail. (20%)
* DQN
  + Selection Action with ε-Greedy Exploration



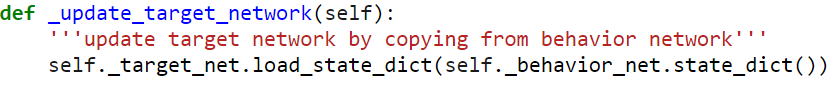
使用selection action來選擇在training時要做的動作，會先使用state丟到behavior net的eval mode來得到該state各action的value，再依照ε-Greedy看是要選擇最大value的action進行exploitation，或是進行隨機action的exploration。

* + Experience replay & Update the behavior network



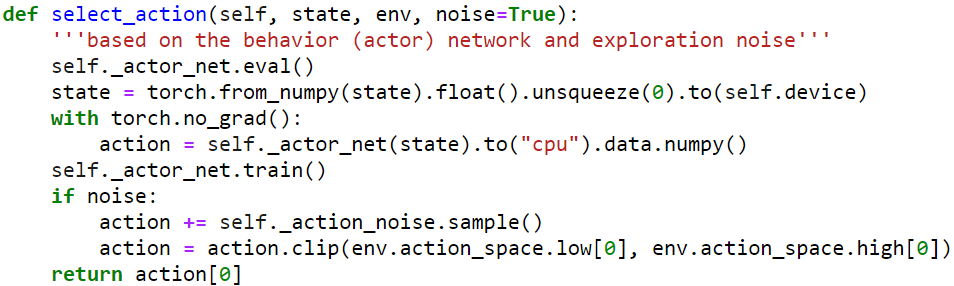
使用replay experience來獲得存在memory buffer中以前算過的資料，再帶入behavior net和target net來求得Q learning的loss function並update behavior net。

* + Update Target Network



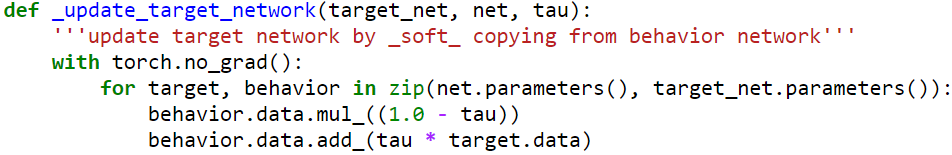
原始的algorithm在訓練了1000步後更新target network，將behavior network的參數複製到target network中，在下面會將其改成soft updating的形式。

* Network Architecture
* Input: 8-dimension observation
* First layer: fully connected layer (ReLU)
  + input: 8, output: 32
* Second layer: fully connected layer (ReLU)
  + input: 32, output: 32
* Third layer: fully connected layer
  + input: 32, output: 4
  + Hyperparameters
* Memory capacity (experience buffer size): 10000
* Batch size: 128
* Warmup steps: 10000
* Optimizer: Adam
* Learning rate: 0.0005
* Epsilon: 1200
* Gamma (discount factor): 0.99
* Update network evert 4 iterations
* Update target network every 100 iterations ( updating by copying )
* DDPG
  + Selection Action with exploration noise



跟DQNε-Greedy不一樣的地方在於DDPG因為是用於continuous action spaces，所以他在exploration時是用添加noise的方式。

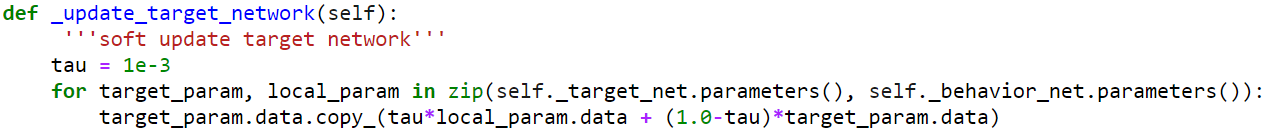
* + Target Network Updating



使用soft updating的方式，讓他可以緩慢差距不大的更新。

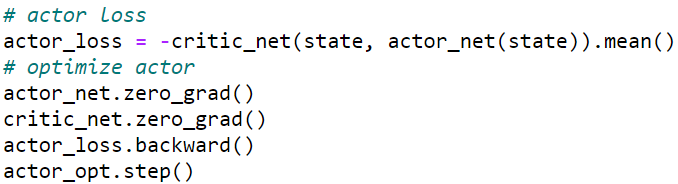
另外DDPG的experience replay方式跟DQN相同故就不再說一次，而actor跟critic updating的方式將在下面詳述。

* + Network Architecture
    - Actor
* Input: 8-dimension observation
* First layer: fully connected layer (ReLU)
  + input: 8, output: 400
* Second layer: fully connected layer (ReLU)
  + Input: 400, output: 300
* Third layer: fully connected layer (tanh)
  + input: 300, output: 2
* Critic
* Input: 8-dimension observation & action with dim 2
* First layer: fully connected layer (ReLU)
  + input: 10, output: 400
* Second layer: fully connected layer (ReLU)
  + Input: 400, output: 300
* Third layer: fully connected layer (ReLU)
  + input: 300, output: 300
* Forth layer: fully connected layer
  + input: 300, output: 1
  + Hyperparameters
* Memory capacity (experience buffer size): 50000
* Batch size: 64
* Warmup steps: 10000
* Optimizer: Adam
* Learning rate (actor): 0.0001
* Learning rate (critic): 0.0001
* Gamma (discount factor): 0.99
* Tau: 0.005
* Describe differences between your implementation and algorithms. (10%)
* DQN
  + Using Soft Updating for Target Network



從原本的直接copy換成了使用soft updating，原理跟上面所述的DDPG soft updating target network一樣。在這邊DQN是每隔4步更新一次target network，而更新的方式變成原先的target network \* 0.997 + behavior network \* 0.003，緩慢的更新，就不會有像是原本copy updating一樣如此劇烈的更動。

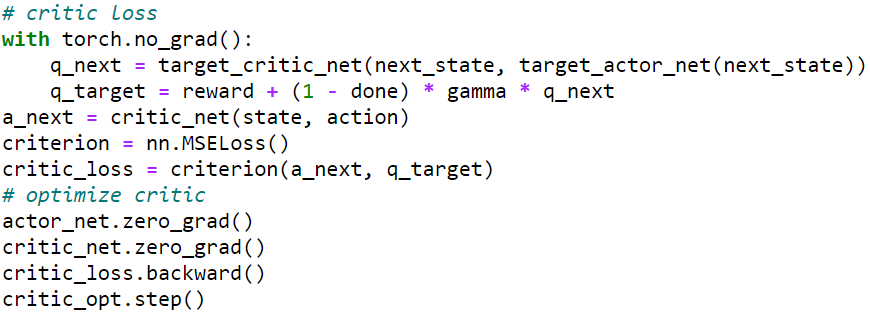
* + Hyperparameters
* tau = 0.001
* Update target network every 4 iterations
* Describe your implementation and the gradient of actor updating. (10%)



actor network用於policy參數的更新，使用policy objective function J(θ)來估測一個policy的好壞，並希望去maximize J(θ)，故將其微分求倒數J(θ) 也就是policy gradient。

在這裡使用Off-Policy Deterministic Actor-Critic，將actor\_net(state)的結果用於critic net的action來計算，並且因為是希望要maximize policy objective function故在前面加負號。

* Describe your implementation and the gradient of critic updating. (10%)



critic用於q value function的近似，他的update方式與DQN中的類似。先求出下一個state的q value，在將他乘上gamma後與reward相加得到target，就可以用原來的state q value與target進行MSE算出loss用梯度下降來更新critic network。

* Explain effects of the discount factor. (5%)

discount factor是一個介於[0, 1]的hyperparameters，當discount factor小的時候，會較重視及時的reward；當discount factor大的時候會較重視遠程的reward。以這次的LunarLander-v2為例，因為他是一個會結束的遊戲，所以即使discount factor很大也不太會有造成總和無限大的問題，且他的最終目標是希望能平穩的降落，因此遠程會是比較好的選擇，因此discount factor就會選擇很大的0.99。

* Explain benefits of epsilon-greedy in comparison to greedy action selection. (5%)

Greedy action selection就是每次都選擇擁有最大action value的選擇，但是這樣會產生Exploration-Exploitation Dilemma，greedy action selection沒有Exploration的步驟，因此也就是說現在選擇的可能並不是最佳的值；而epsilon-greedy會利用epsilon當作一個機率threshold，有一部分的機率會去進行exploration，另一部分的機率是進行exploitation，且可以保證經過epsilon-greedy後policy value一定會增加。

* Explain the necessity of the target network. (5%)

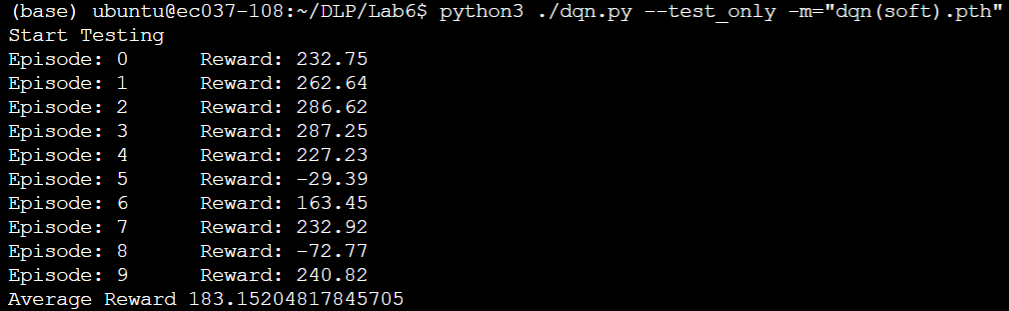
如果沒有target network而只有一個network的話，那個network要同時update和產生target，因此等號兩邊都會是變數，造成訓練會很不穩定，所以需要讓式子的一邊並成定數，故使用target network來當作update時的目標。

* Explain the effect of replay buffer size in case of too large or too small. (5%)

在大多數實驗中，replay buffer設置為默認容量106。實際上replay buffer的大小會嚴重影響到學習速度和agent的quality。如果replay buffer太小則違背了他的用意，幾乎沒有用處，因為replay buffer是希望能夠儲存下之前跟環境互動後的結果並反覆再利用，但如果太小的話，基本上沒有多少可以儲存。而如果replay buffer太大，會需要很大量的memory來儲存，並且也會減慢訓練的速度。

**Performance**

* [LunarLander-v2] Average reward of 10 testing episodes: Average ÷ 30
* Using Soft updating target network  
  Average reward：183.137



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

Average reward：231.80