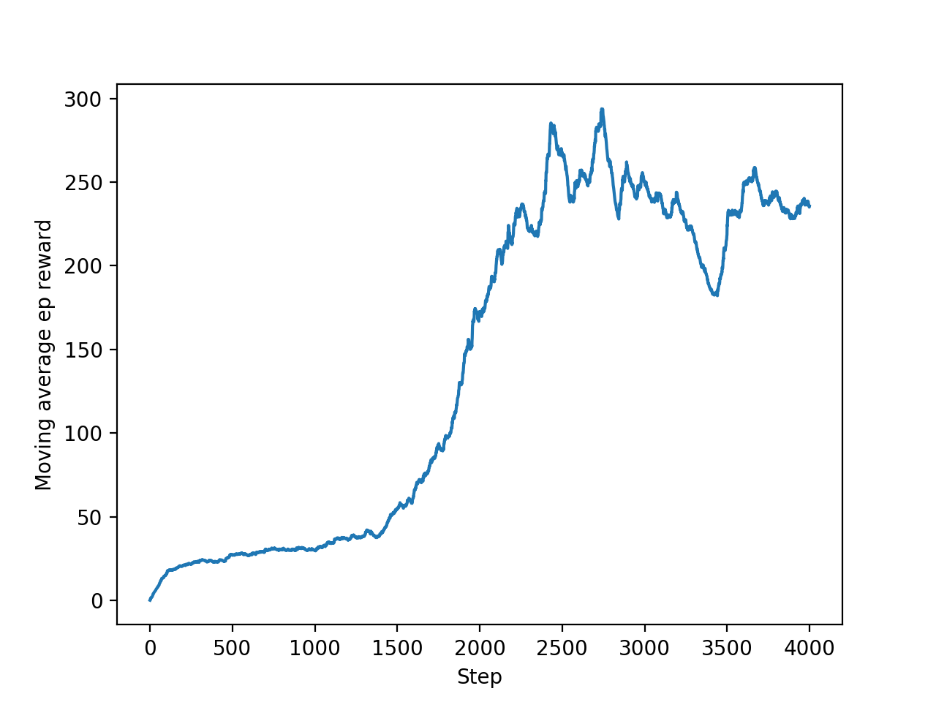
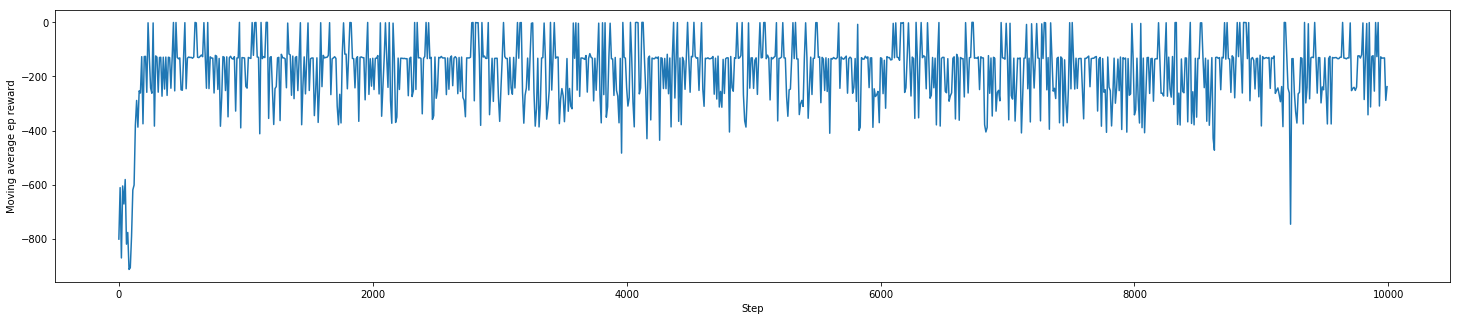
**Lab 8: DQN and DDPG**

* **A plot shows episode rewards of at least 1000 training episodes in the game CartPole-v0 (DQN)**



* **A plot shows episode rewards of at least 10000 training episodes in the game Pendulum (DDPG)**



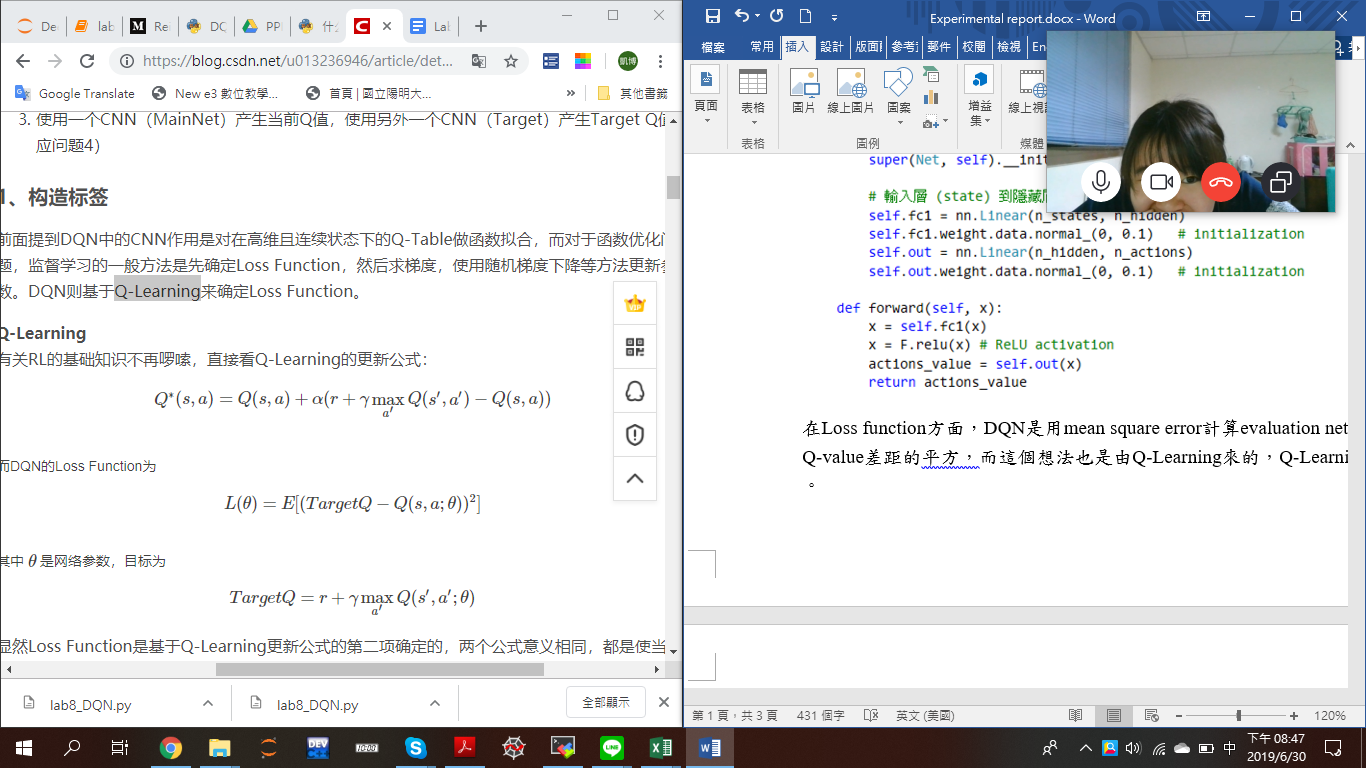
* **Describe your implement/adjustment of the network structure & each loss function (DQN + DDPG)**

1. **DQN**

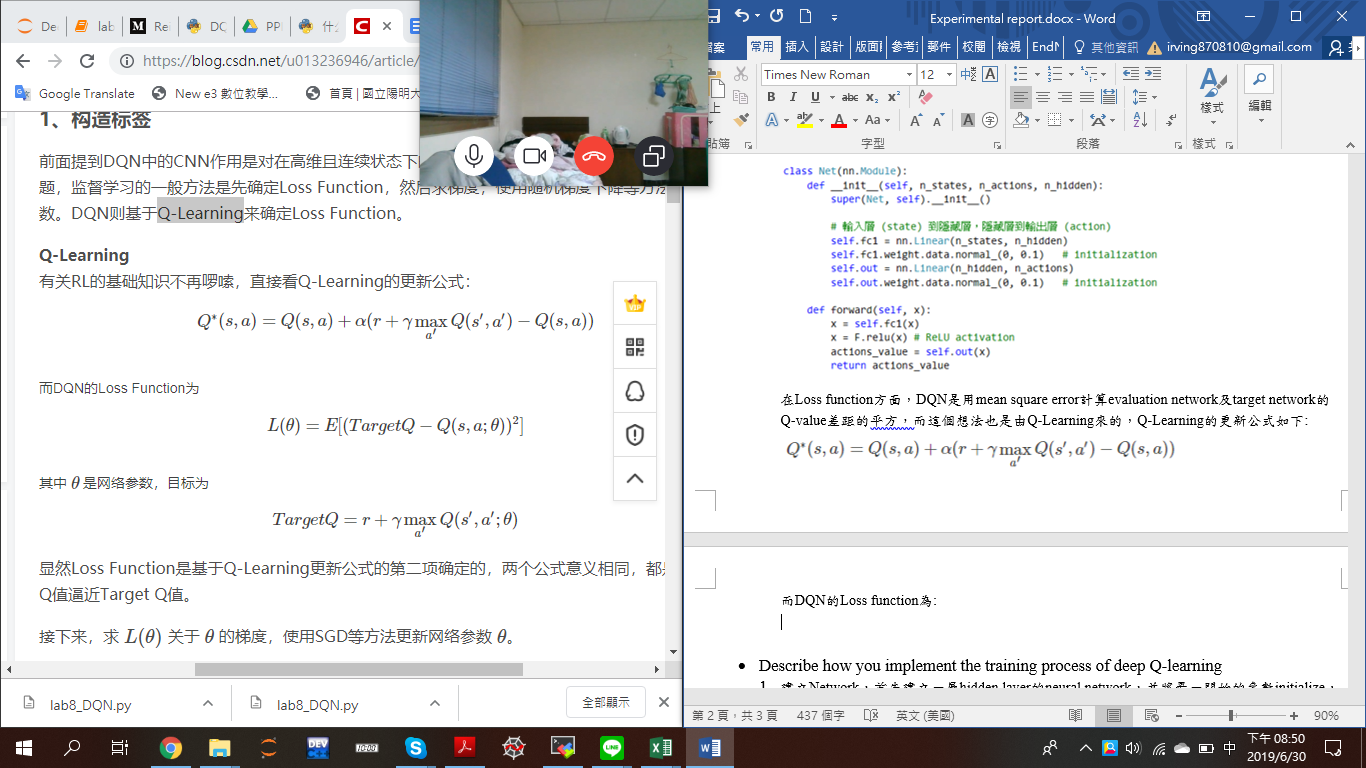
DQN可以利用neural network取代Q-learning用Q-table記憶state和action，好處是neural network有更好generalization的效果，可以從龐大的state space中自動提取特徵。其中我將Network都調整程簡單的Fully connected linear layer，參數用normal distribution的分布initilize到0~1之間，最後用relu當activation function輸出。



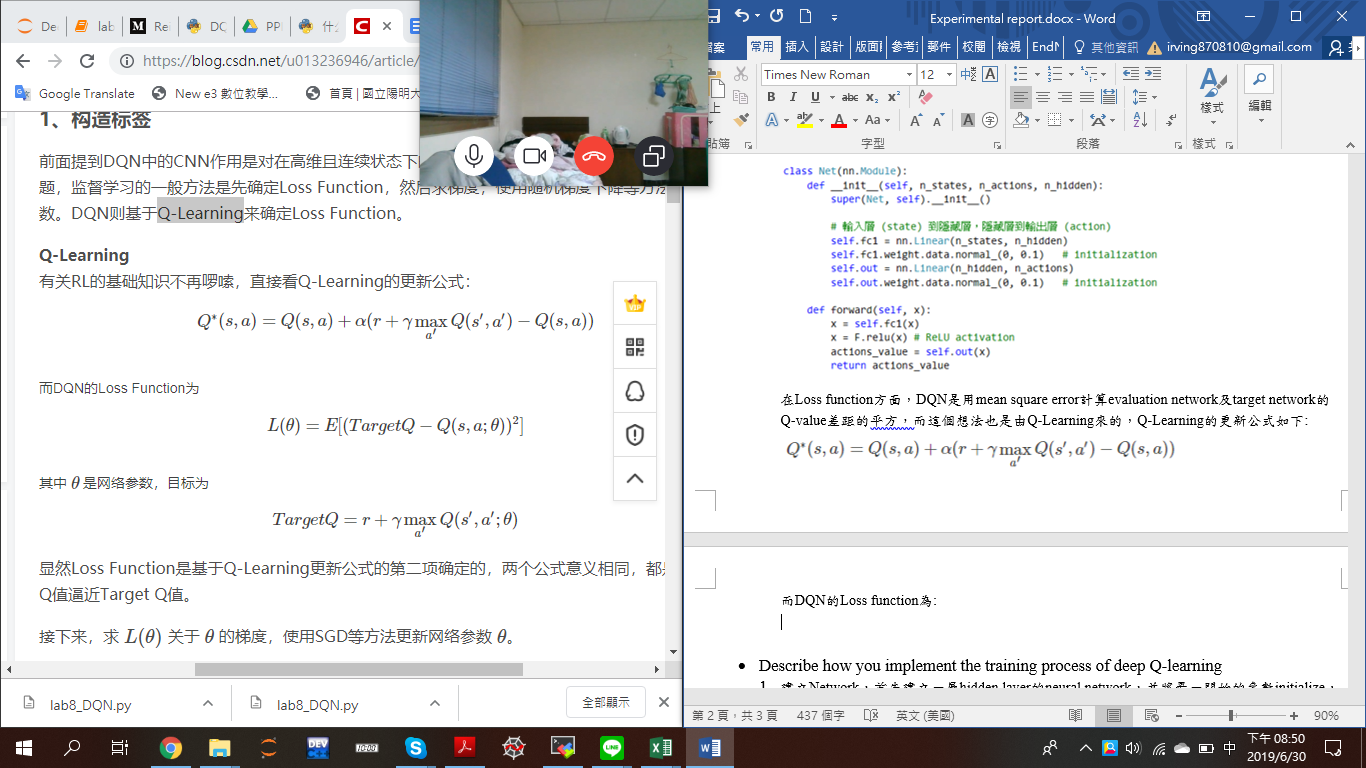
在Loss function方面，DQN是用mean square error計算evaluation network及target network的Q-value差距的平方，而這個想法也是由Q-Learning來的，Q-Learning的更新公式如下:



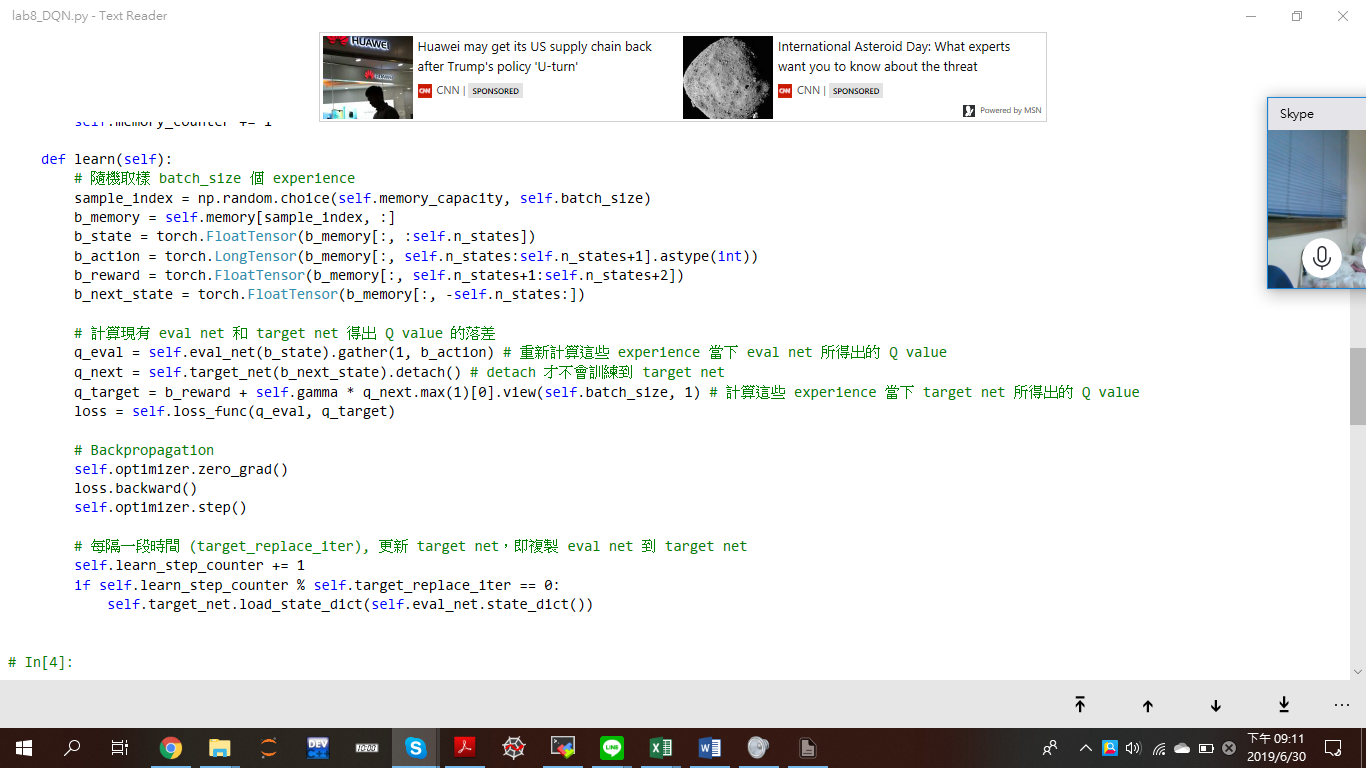
而DQN的Loss function為:



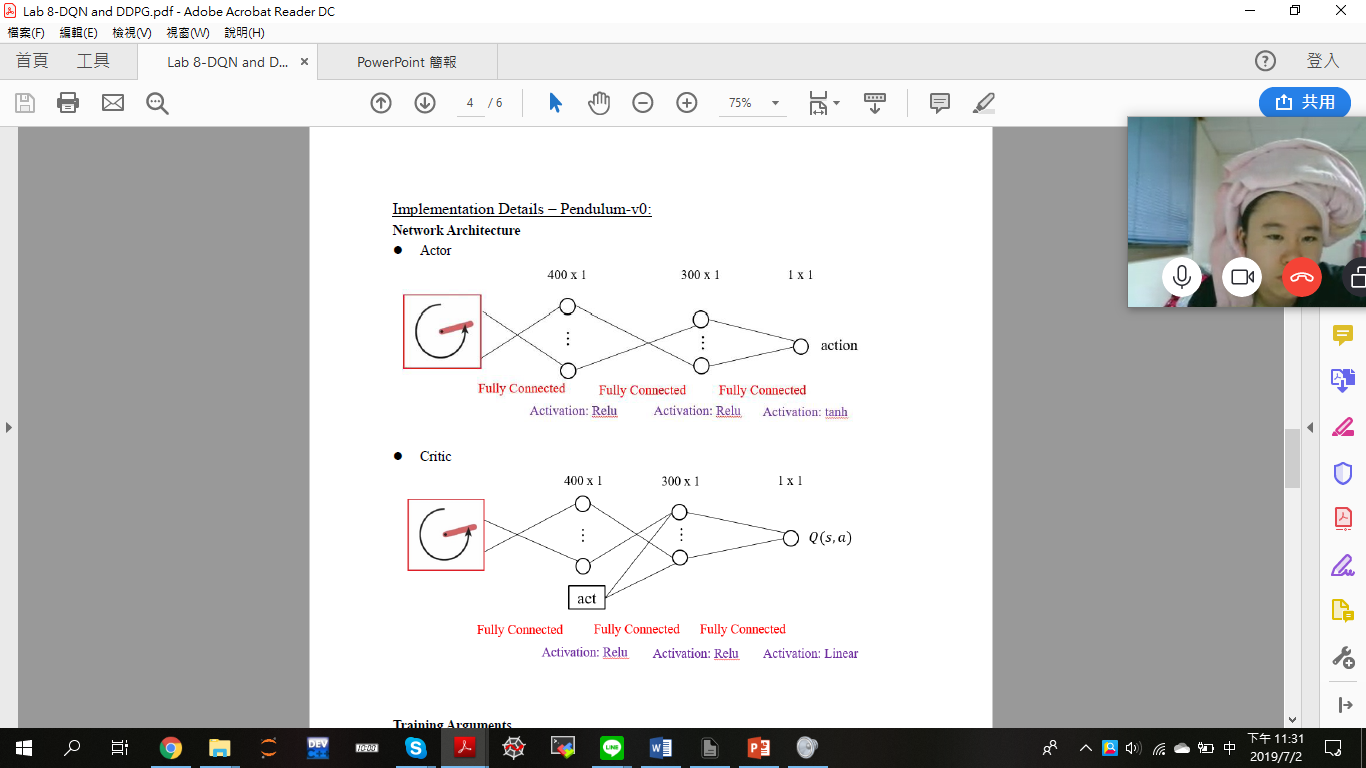
其中θ是神經網絡的參數，所以可以得知Target Q是:



所以DQN是想要是當前的Q(s, a；θ)去逼近Target Q值，程式碼如下:



1. **DDPG**

DDPG採用actor-critic架構，actor輸出動作，critic評判動作後輸出價值，並且借鑑DQN的思想，所以共有四個神經網絡。即: critic部分有兩個神經網絡，target network Q' 和critic network Q；actor部分有兩個神經網絡：target network u' 和actor network u。首先actor和critic的架構如講義。在actor的部分actor network u的輸入為當前狀態，target network u'的輸入為下一狀態；在critic的部分 critic network Q的輸入為當前狀態和當下動作，target network Q'的輸入為下一狀態和下一動作。兩部分的hidden layer也都分別是400和300並以fully connected layer，最後actor輸出action，critic輸出Q-value值。

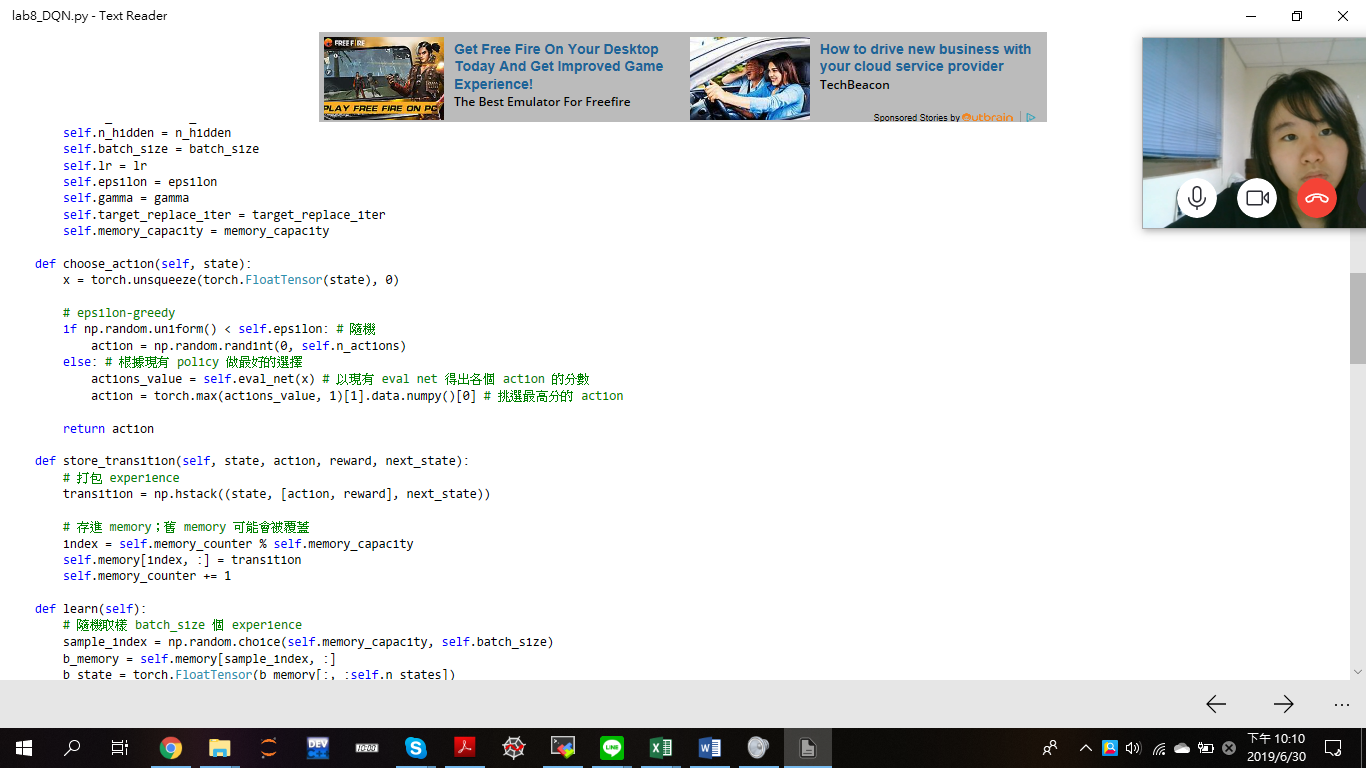
* **Describe how you implement the training process of deep Q-learning (DQN)**

在訓練DQN的過程中，要設定max\_steps以比免遊戲無法終止，在每一輪一開始的時候，會先刷新環境後再選擇action，依epslion的機率決定是否要依照最大的Q-value值走，之後就執行行動返回下一個狀態和reward等資訊，之後修改reward使得agent可以更抓到遊戲的訣竅，分別是使車子的水平位移變小，並讓棒子越正越好，儲存經驗 (儲存Q-value值)，直到有足夠的experience後再進行訓練，接著進入到下一個state，反覆循環直到遊戲終止或是達到max\_steps。



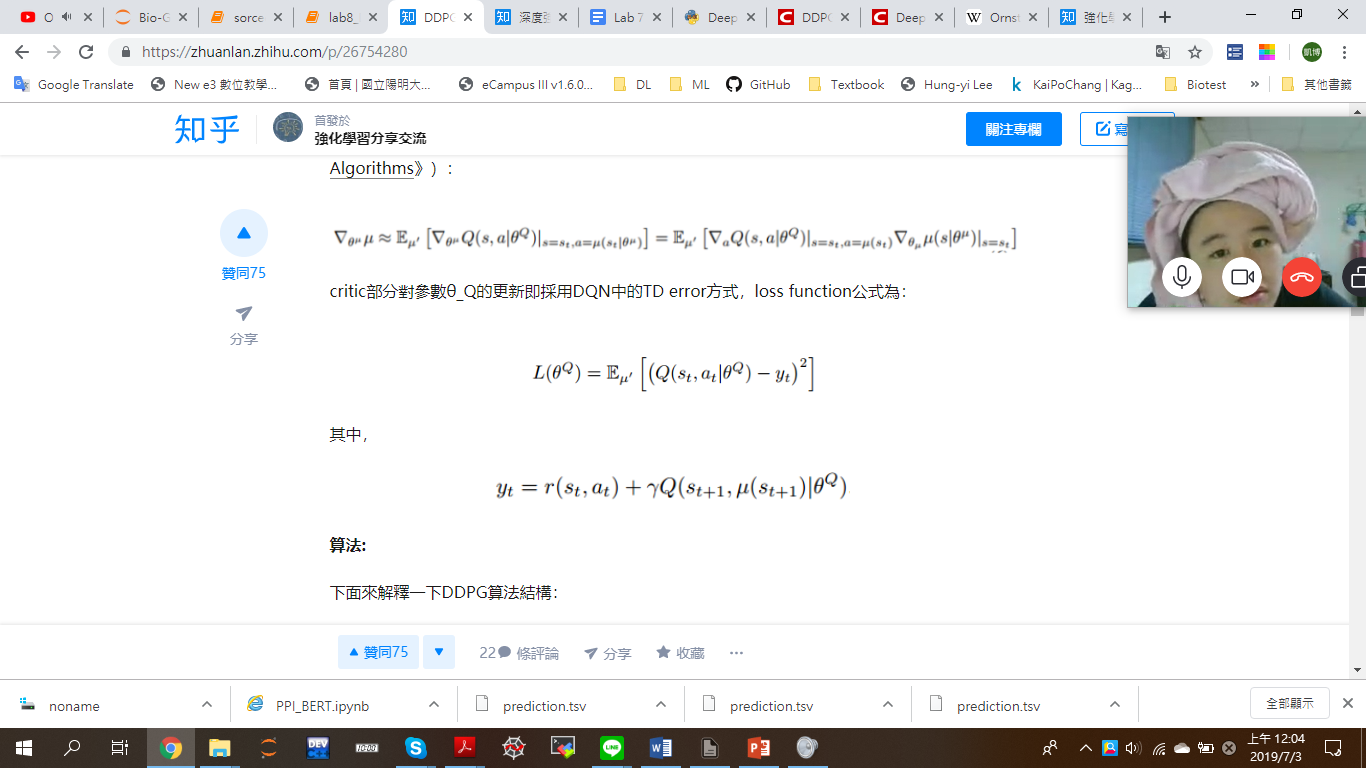
* **Describe the way you implement of epsilon-greedy action select method (DQN)**

在這一次程式碼裡面的epsilon = 0.1，若是用np.random.uniform的方式去生成的值小於0.1，那agent會隨機往左或往右移 (carpole)，如此便有一定的機率會去走沒有走過的路；反之，若是用np.random.uniform生成的值大於1，agent便會根據Q-value最大的值去做出action。

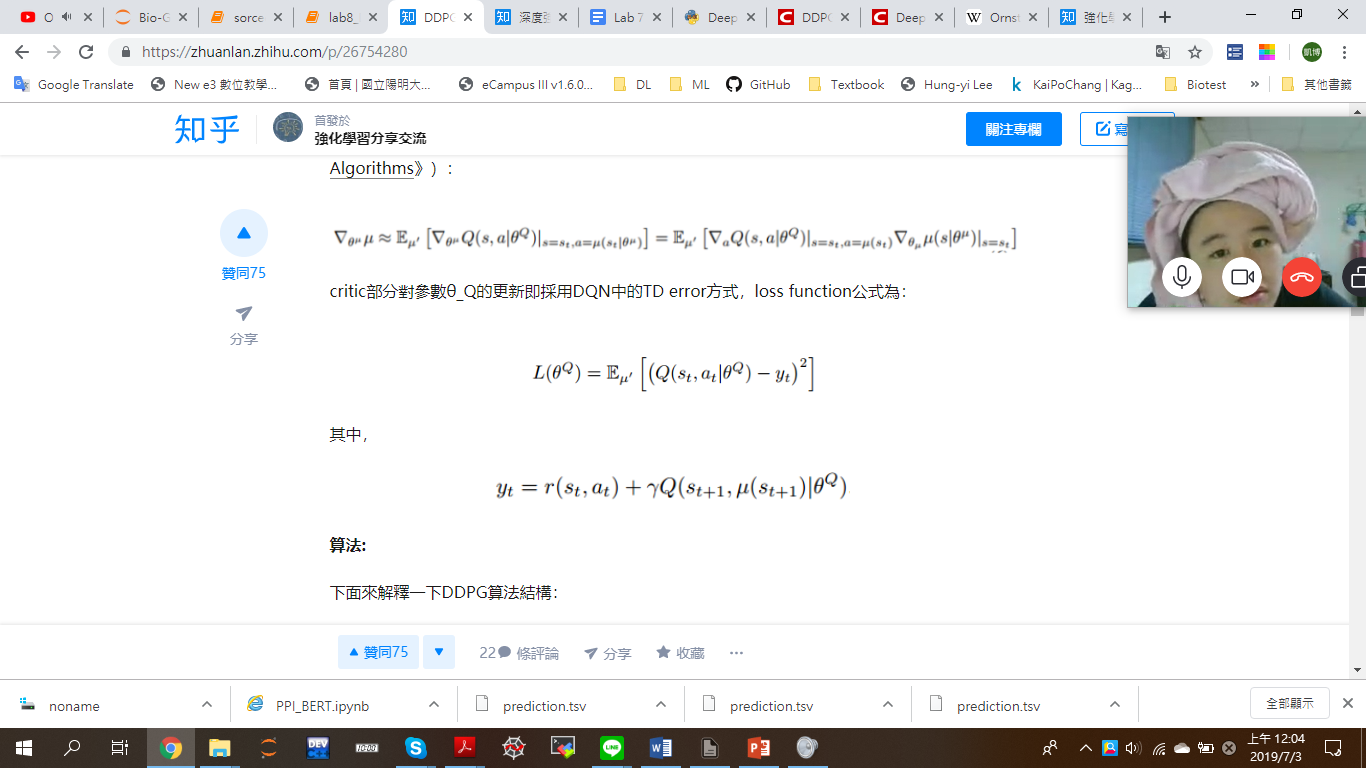


* **Explain the mechanism of critic updating (DDPG)**

在DDPG critic部分對參數θ\_Q的更新，是採用DQN中TD error的Loss function:

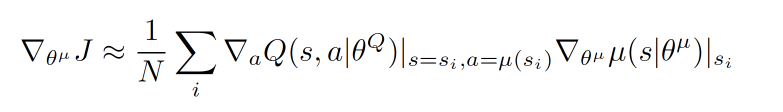


其中，



* **Explain the mechanism of actor updating (DDPG)**

actor參數的更新涉及到critic，其中grad[Q]代表受到critic影響，而後半自身的θ\_u代表actor會將自身參數optimization，整合在一起就是actor會朝著Maxium Q-value的方向更新參數。



* **Describe how to calculate the gradients? (DDPG)**

對於計算actor的policy gradient是利用David Sliver在2014年提出DPG（Deterministic Policy Gradient，DDPG改善DQN策略動作以stochastic去執行動作，變成是以derministic的方式去執行動作，使得Q-value的Bellman function由:

https://pic4.zhimg.com/80/v2-9da2792a518f070ad5c4ec58b13948c7_hd.png

改為:

https://pic2.zhimg.com/80/v2-3fc9ef97e14ed1ec4e93b20d08dcf151_hd.png

* **Describe how the code work (the whole code) (DDPG)**

DDPG一開始要建立environment，一開始先隨機選擇action並為了之後的exploration在action上增加randomness，得到下一步狀態的資訊後先儲存在replay memory buffer中作為online training的dataset，之後會隨機sampling N個 transition data作為一個mini-batch的training data，此時可以得知更新actor和critic網絡參數的狀態、動作、reward、下一狀態。再來分別更新actor network和critic network的網絡參數，最後依據soft update以更新target network的參數 (Q-target和現有的critic network output Q-value)。

* **Other study or improvement for the project.**
* **Performance**

1. DQN: 由於訓練訓練較多場 (4000場)，後期reward普遍較高，平均每一百輪的testing episodes為213.5，除以2後是106.75。

2.