Lab 3 Deep Traffic Report A061054 蔡昀芸

# Model Design

Q1 :

1. 首先，設定車可以看到的視線範圍，lanesSide = 3使車子可以看到左右各三道，PatchesAhead = 40 可以看到前方四十個grids的視線，這樣可以在training時候，預先判斷前方比較廣範圍的車輛狀況，而PatchesBehind = 5，因為測試之後發現後方視野廣對學習並未有甚麼改變，但增加input會導致training變慢所以只設5而已。總共Iteration為100000次。Temporal\_window = 0 是因為想要車子不透過先前的紀錄去學習。
2. Layer design設成兩層，增加一層Tanh在Relu前面使負的值也可以被傳回。
3. learning\_rate: 0.001, batch\_size: 128, l2\_decay: 0.01，這邊batch\_size調大的原因是因為input變多想要讓他training快一點，同一個batch裡的值代表較相近也可以學習效果也較好。
4. gamma折扣函數 = 0.95，對於每個result的值乘上折扣函數避免over-fitting。值高一點可以讓學習更有效果。而epsilon\_min = 0.2，讓他做決策時有二十趴是以隨機的決定做決策，這樣可以讓車子學習的時候學到原本沒有的決策。

Q2 :

DQN訓練過程是將Q 值(q)向目標 Q 值(target\_q)逼近，Q值代表車子從環境跟自身經驗學習到的經驗，會以一個Award分數呈現。loss = (target\_q - q)^2。

一開始會先初始車子狀態(s)，初始記憶並設置觀察值，在每一次iteration中車子根據記憶裡有的策略，選擇一個策略(a)並執行，執行完後得到獎勵(r)，狀態改變為(next\_s)，將s, a, r, next\_s存到記憶裡之後，再從記憶裡抽出一部分當作訓練樣本，剛剛所改變的next\_s會當成神經網路的輸入值，進行批次處理，再得到新的獎勵跟狀態。

**Q(s, a) = r + Gamma \* Max[Q(s`, all actions)]**

搜尋策略並做決策 : 透過隨機參數Epsilon，增加探索的機會，車子可以選擇要用神經網路計算出來的結果當決策，還是用隨機的結果當決策，可以避免DQN陷入局部最小值

# Question & Discussion & Analysis

## Difficulty

1. 在Q1的第二個步驟設計Layer的時候，因為我發現Input變多如果用Relu的話會deactivation，在back-prop的時候負的值會變為0，所以沒有辦法再使值活化。另外，如果今天有很多層layer，而每個layer都用relu會造成梯度爆炸，gradient vanish，值無法回傳，所以加上Tanh這一層layer可以避免這種事情發生，
2. Gamma值如果設比較低，相對折扣會比較大，主要是要解決over-fitting問題，但在學習上會比較慢效果也不好，所以後來就嘗試了0.9以上，發現0.95效果最好。

## Analysis Hyperparameter

1. 隨機函數epsilon\_min 我覺得是很重要的參數，因為在training的時候，有關查到車子一開始會no action，也就是對於任何狀況都沒有判斷只是一直向前，但之後車子便會開始知道可以加速這件事，但是遇到正前方有車還是會慢下來，並不會作轉跑道的動作，但是因為有隨機參數的關係，讓車子做決定能有隨機的可能性，而因此學到轉跑道這件事情。所以車子之後的行徑方式便會多了轉跑道這個決策。而在了解到可以轉跑道後，車子也會開始學習前方哪一區塊的車子比較多，而轉到車子較不多的區塊，因此速度便可以不斷的增加。