Neural Network Homework3 Report

R06922063 邱政凱

在這次的作業中，我們更改了原本只有 ASE(Associative Search Engine)的 Q-learning 的程式，使其在計算 reinforcement(reward) value 時能夠參考另一個稱為 ACE(Adaptive Critic Element)的元件，使得 reward 不再只是完全的 myopic 的只參考當前的環境，而是能夠以一個遞減的比例把離當前時間點越遠的比例乘上越低的權重加入 Q Value 的更新公式中。

這次作業我使用的是 Matlab。總共有 Q_Demo.m、get_action.m、get_box.m、failed_update.m、reset_cart.m 等檔案，Q_Demo.m 是主程式，用 matalb 開啟後執行即可使用。

實作面上我是基於老師提供的 code，並且修改成支援 ACE 的版本，並且改了一些程式使其不要在每個 trial 都重新一直把 Q-Mesh 和單擺的 figure 都畫出來(會使得程式沒效率)，而是只在程式結束執行(success > 50000 或 trial > 10000)時才一次把圖畫出來存檔。並且會繪製額外的摺線圖來讓我們可以比較 success iteration 隨著 trial 的增加跟著變化的情形。

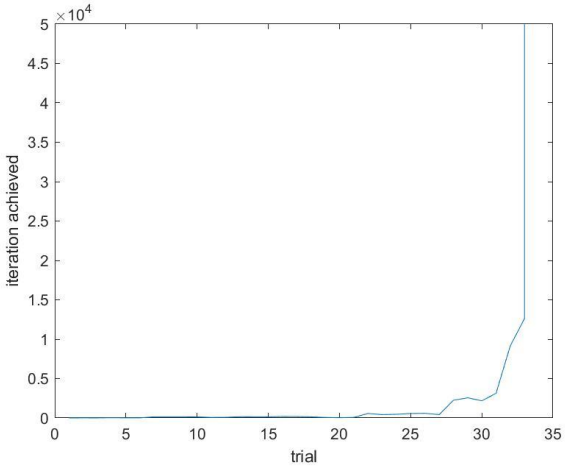
EXPERIMENTS & ANALYSIS:

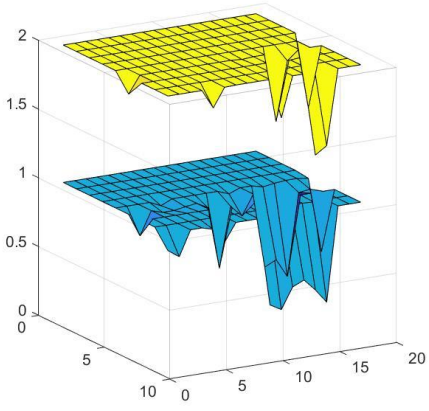
這個問題中兩個主要的參數: ALPHA 跟 GAMMA 我分別選擇了

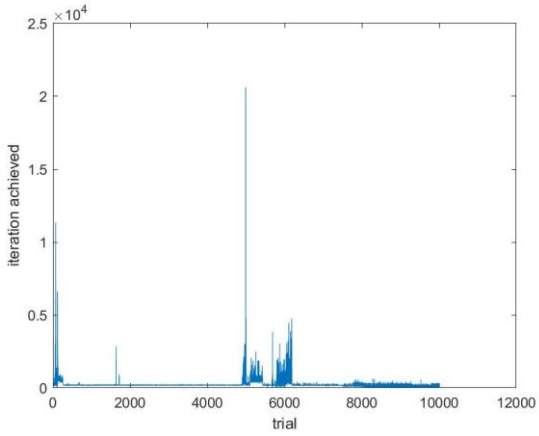
0.4 跟 0.8，這部分主要是參考老師的 DEMO 網站的表格並且自己做了一些實驗後選出來的。主要的考量是實驗後發現 Gamma 太低的話基本上 q value 參考其他時間點的 reward 的意義就不太大了，而 Alpha 太低的話 trial 學習的速度會太慢，可能數萬個 trial 都還無法達到理想的成功次數，太高的話又會導致參數學習不穩定，即時學到了較好的權重也會馬上跑掉，造成學習結果不易收斂。

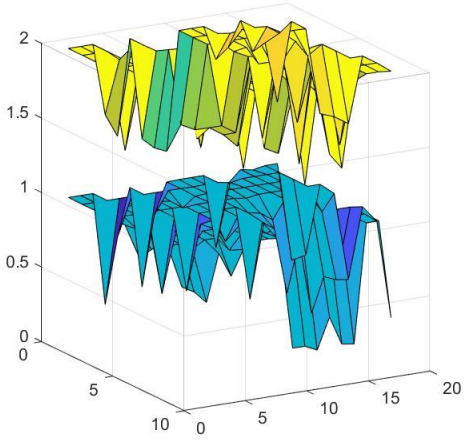
η γ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	∞	∞	∞	∞	∞	**870**	**198**	∞	∞
0.2	∞	∞	∞	∞	∞	**352**	**6831**	**53**	**541**
0.3	∞	∞	∞	∞	∞	∞	**155**	**231**	**3225**
0.4	∞	∞	∞	∞	∞	∞	∞	**283**	**1288**
0.5	∞	∞	∞	∞	∞	∞	**1383**	∞	**590**
0.6	∞	∞	∞	∞	∞	∞	∞	∞	∞
0.7	∞	∞	∞	∞	∞	∞	∞	**120**	∞
0.8	∞	∞	∞	∞	∞	∞	∞	∞	∞
0.9	∞	∞	∞	∞	∞	∞	∞	∞	∞

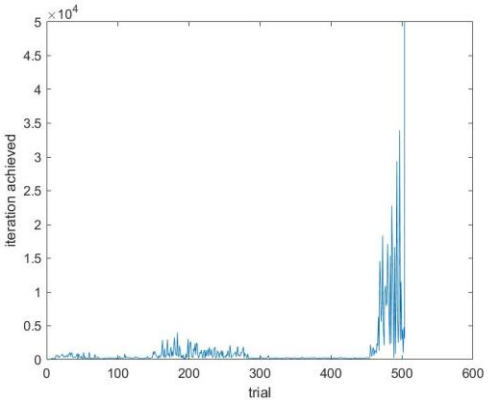
以下我會比較在修改前和修改後的 Q Learning 的成功次數與 trial number 的關係的折線圖來呈現加入 ACE 之後的效果改善。

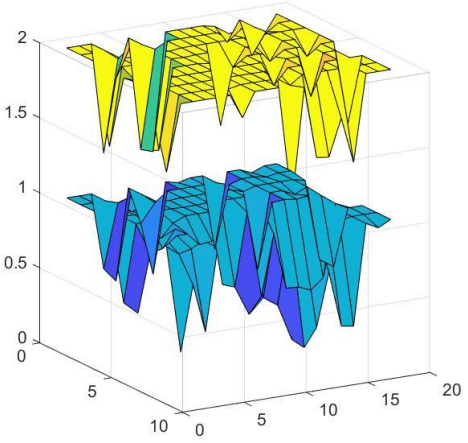


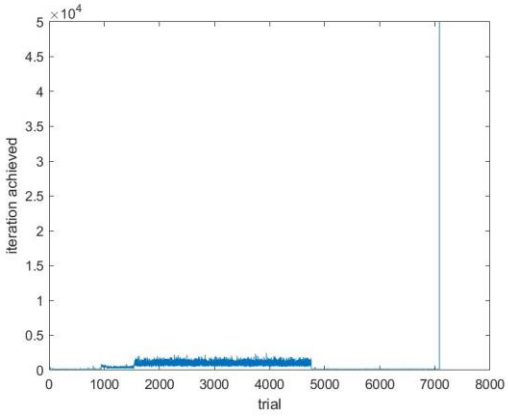


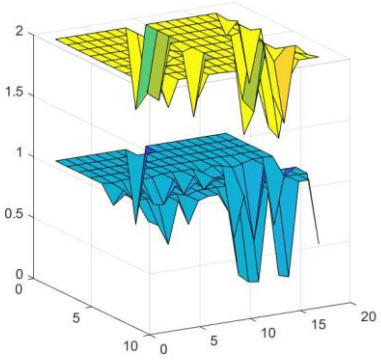

×10^4
iteration achieved
trial

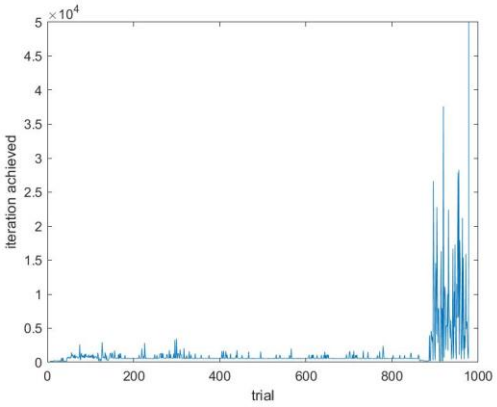


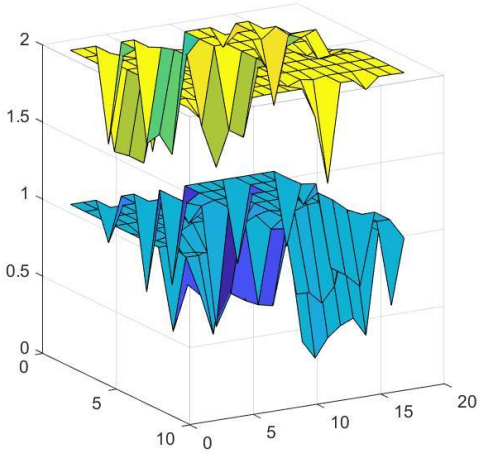

×10^4
iteration achieved
trial



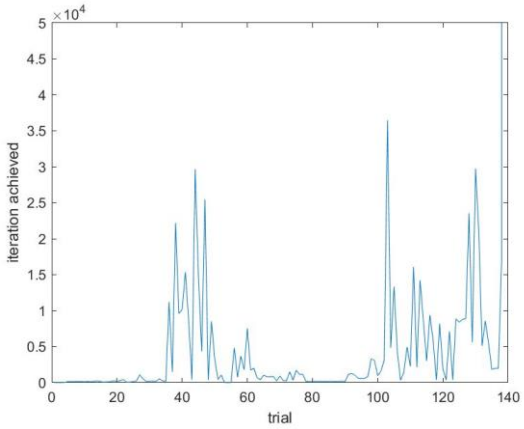

×10^4
iteration achieved
trial

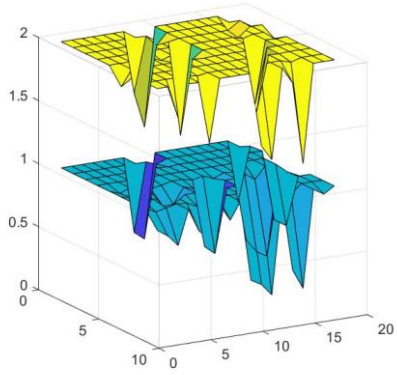


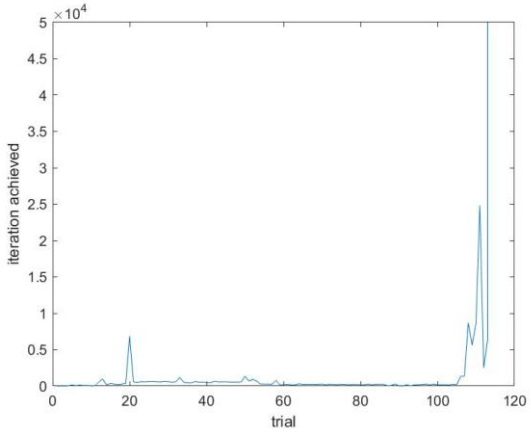


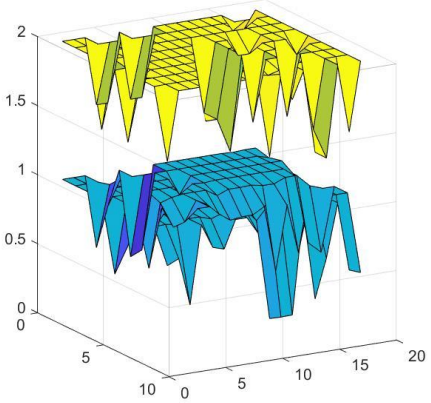


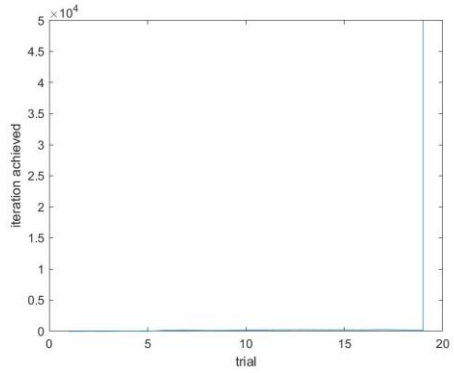
以上五個分別是修改”前”的 Q Learning 跑出來的結果(折線圖以及最終 Q Value 用 mesh 呈現的形式)

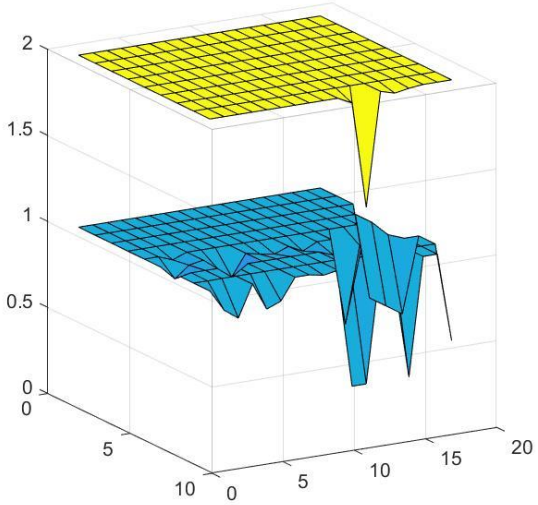


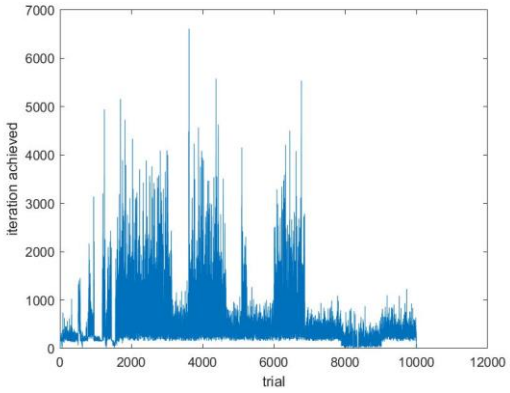


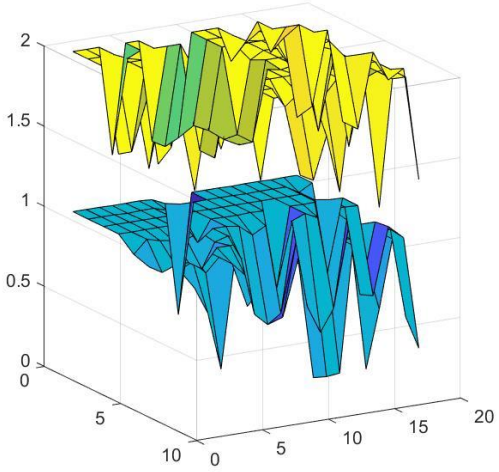


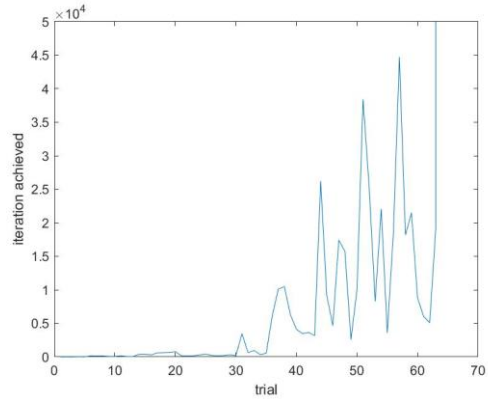


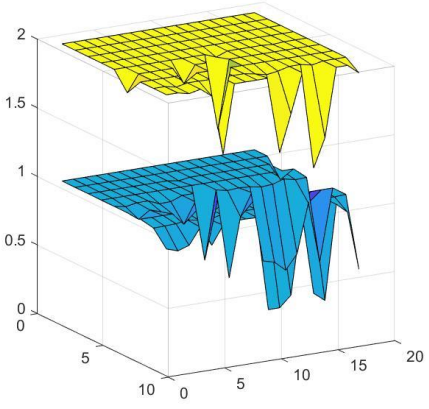












以上五個分別是修改”後”的 Q Learning 跑出來的結果(折線圖以及最終 Q Value 用 mesh 呈現的形式)

比較以上兩者的結果不難發現

(1) 平均而言，修改後的比修改前平均達到目標成功次數(50000)需要的 trial 次數較少，儘管我們仍然可以看見許多例外，像是修改前的第一個結果其實是比修改後除了第三個的結果都還要快達到目標的。

(2) 可以看到 Random 的狀況很重要，可以看到對於不論是修改前或是修改後的 Q Learning，都是有可能運氣不好在 10000 個 trial 之後仍舊達不到滿意的成功次數的(而其他次卻幾乎都在遠小於 10000 trial 之前就達到目標的成功次數)。

(3) 不論是修改前或修改後，都可以看到其實整個折線圖是經常在波動的，也就是說可能在某一個 trial 達到了滿高的成功次數，下一個 trial 可能又急遽下降。這部分也要歸因於 reset cart 使用了 random 的關係。不過雖然如此，但是如果 reset cart 裡面不使用 random 值得話經過測試其實是幾乎怎麼測都無法在 10000 個 trial 之內完成的。不過整體折線圖的大方向還是偏向於漸漸提升成功次數的。(修改前以及修改後的第五個圖對於這種傾向都表現得滿明顯的)。