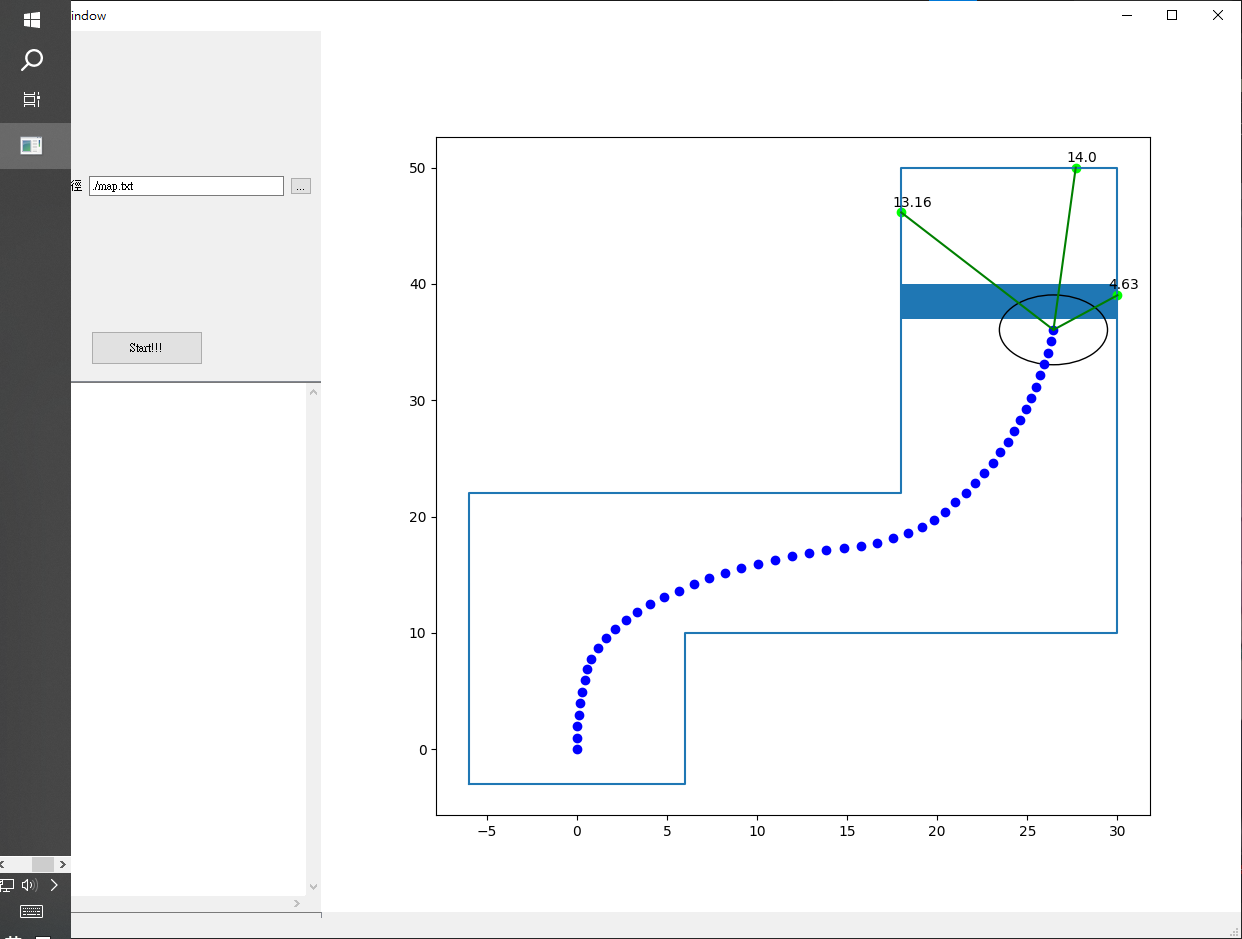
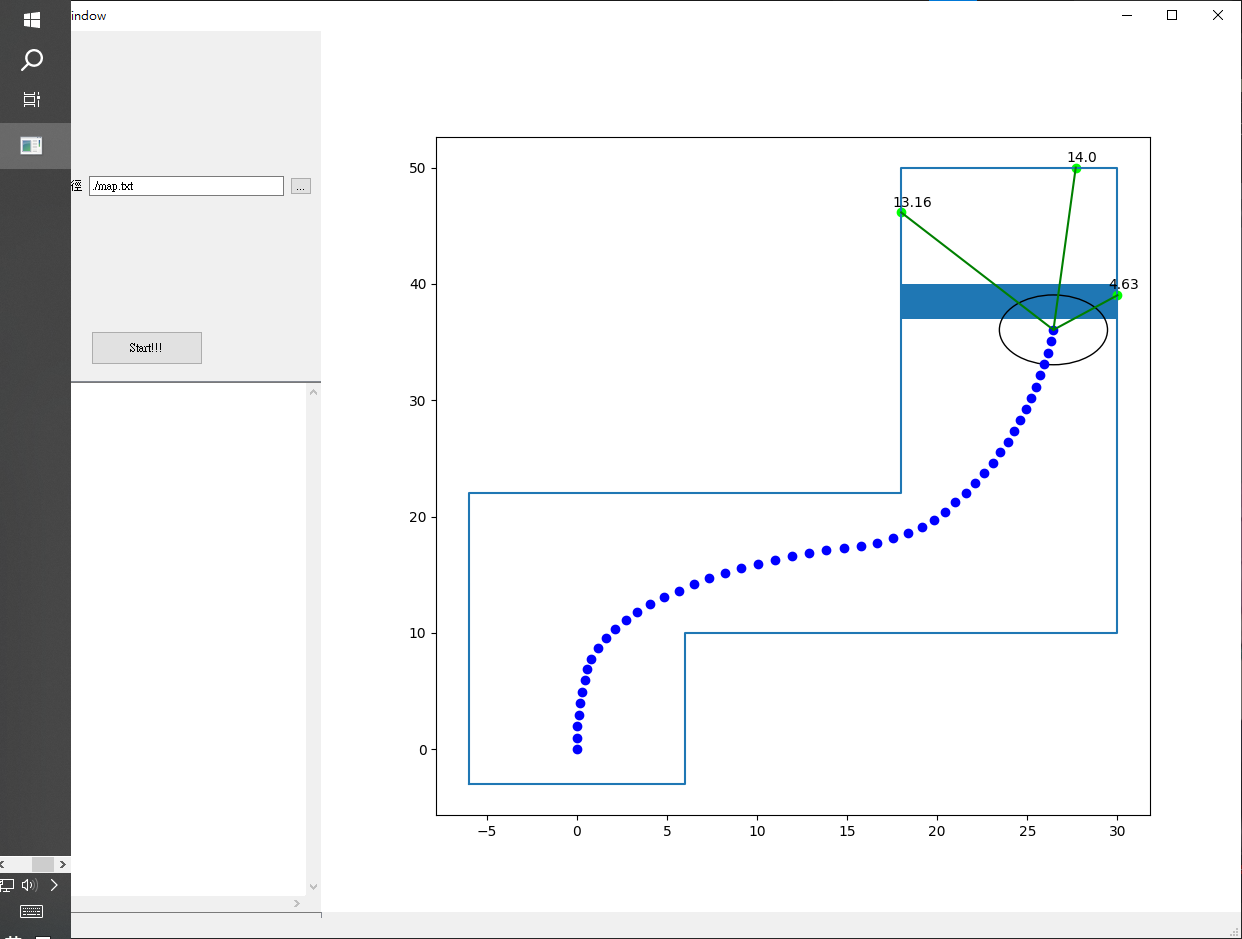
PSO

111522094 資工碩一 凃建名

1. 程式執行介面:



使用步驟:

1. 選擇地圖文件
2. 點選Start! 按鈕
3. 實驗結果:  
   

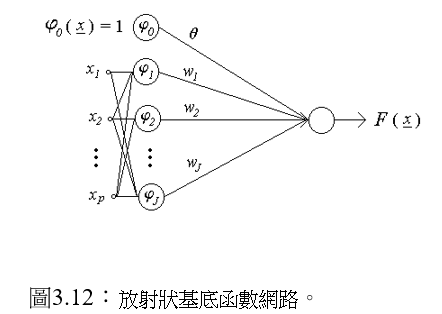
結果說明:

初始個體數 = 200，Vmax = 20，Vmin=-20 w= 0.8 Ψ1=0.2 Ψ2=0.2

結果: 在epoch 15~30時就可以跑到終點，費時1~2分鐘

透過使用PSO演算法train出適合的RBFN的參數值，確認自走車可順利到達終點，RBFN的輸入為前左右三個Sensor之距離，輸出則為方向盤之旋轉角度。

1. PSO實作細節:

網路採用RBFN，將RBFN所有可訓練的參數當作一條基因的內容。

一個個體 = [ weights, m\_list, std\_list ]

1. 初始個體的選擇:  
   weigths, std\_list均為0至1之間隨機選擇的隨機變數陣列

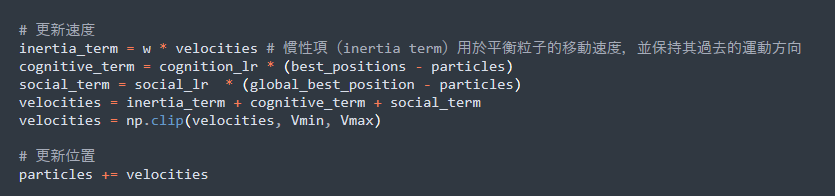
m 為 隨機選擇train dataset中的輸入參數

V(0) 根據常態分佈選擇Vmax和Vmin之間的數

1. 更新速度和位置：

每個粒子都根據其當前的速度和位置信息來更新自己的速度和位置。這是通過結合三個不同的項來實現的。首先是慣性項，它保持了粒子的運動慣性，使其在搜索空間中繼續移動。其次是認知項，它是根據粒子自身的最佳解來調整速度和位置，使其朝著更好的解移動。最後是社會項，它是根據整個群體中全域最佳解來調整速度和位置，以便粒子能夠利用群體的知識和經驗進一步優化解。

此外根據我的觀察由於個體中的元素最大為30左右因此訂Vmax和Vmin分別為 20 和 -20

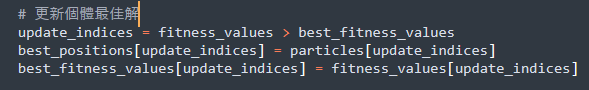


1. Fitness function

將基因實際放入一個RBFN中並讓車子使用此RBFN從起點開始跑，最後計算fitness\_value = -0.3 \* (使用步數) + 0.7 \* (最後停車的點與起點的距離)+bonus，如果車子到達終點則bonus = 100000，其餘bonus = 0，如果fitness value > 50000則最後算出來的fitness\_value還需減去 (使用步數) \* 10。

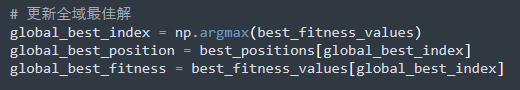
1. 更新個體最佳解：

每個粒子都記錄著自己曾經找到的最佳解，稱為個體最佳解。當一個粒子在每次迭代中找到一個更好的解時，它會更新自己的個體最佳解。這樣，每個粒子都保留著自己曾經找到的最好解，並根據這個最佳解進行調整，以便在下一次迭代中更好地搜索解空間。



1. 更新全域最佳解：

全域最佳解代表整個粒子群中找到的最佳解。每個粒子都保持著全域最佳解的信息，並在每次迭代中與自己的個體最佳解進行比較。如果某個粒子的個體最佳解優於當前的全域最佳解，則該粒子會更新全域最佳解。這樣，整個粒子群都可以共享最優解的信息，並朝著更好的解進行搜索，以達到全域最佳解的不斷改進。



1. 終止條件:

Epoch達到100 or 連20個epochs最好的fitness value都大於50000(最好的基因都能到達終點)

1. 分析:
2. 本次我遇到的第一個問題是在於收斂速度過慢，剛開始時我設個數為50的時候收斂緩慢常常在epoch100時仍無法達到終點，因此我將個體數擴增至100即可。
3. 第二個問題是Ψ的設定，原本我設定的Ψ為0.2 \* random數，但是後來發現他會導致越到後面速度越來越慢，但改為定值0.2後即可解決。