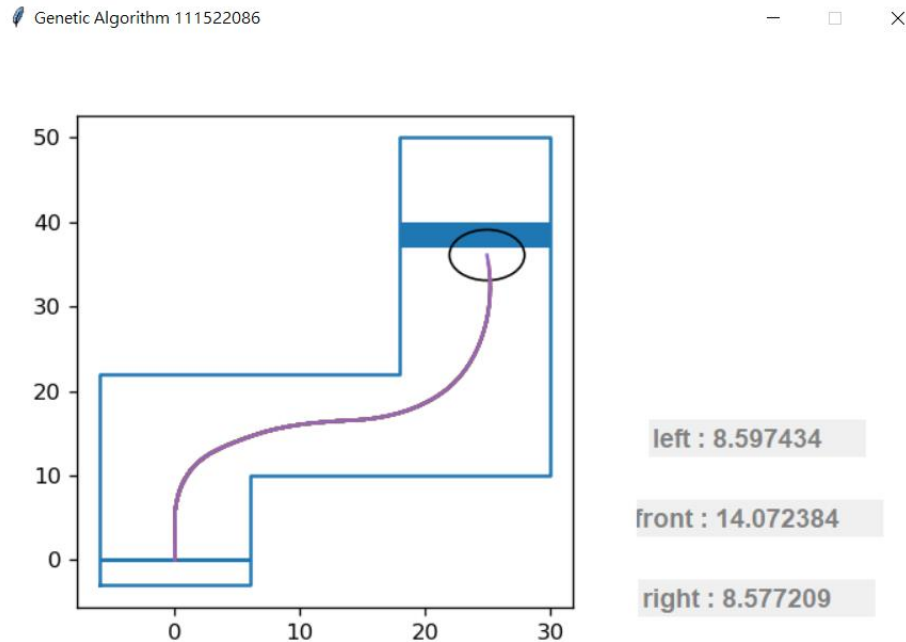


計算型智慧 HW2 – Genetic Algorithm

111522086 林思婷

一、 程式介面說明



藍色細線為軌道的牆壁，黑色圓圈為自走車，起點置於軌道座標(0,0)的位置，終點線位於藍色粗線位置處。程式執行後會自動以動畫顯示自走車每一步的位置並在介面右側顯示左/右前方 45 度角、正前方等三個測量距離 sensor 所測量到的距離。

此次作業共有六份 python 檔，分別為：

1. GeneParameter (基因演算法實作細節)
2. generateGP
3. gene (基因初始化、管理和設置，計算適應值)
4. RBF
5. geometry
6. playground

前 4 份檔案利用基因演算法訓練 RBFN 的網路參數

執行 2.檔案，讀進助教補充的 train4D.txt 檔進行訓練，訓練完成後得一組網路參數，實作的細節將在第三節中詳細說明。

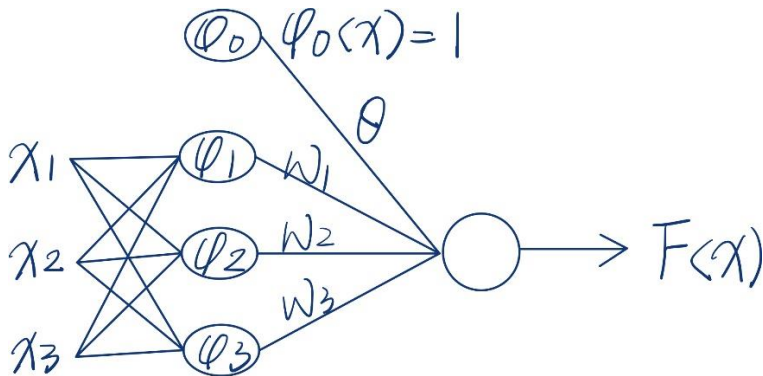
後 4 份檔案將 sensor 偵測所得距離輸入 RBFN 並輸出方向盤角度

二、實驗結果

如一、圖所示，在移動的過程中均能根據 sensor 回傳距離改變方向盤的角度，以避免觸碰軌道牆壁，順利抵達終點線。

三、基因演算法實作細節

此次作業採用實數型基因演算法，下圖為 RBFN



$$F(\underline{x}) = \sum_{j=1}^J w_j \varphi_j(\underline{x}) + \theta = \sum_{j=0}^J w_j \varphi_j(\underline{x})$$

此 RBFN 選用高斯型基底函數

$$\varphi_j(\underline{x}) = \exp\left(-\frac{\|\underline{x} - \underline{m}_j\|^2}{2\sigma_j^2}\right)$$

我們需將 RBFN 的所有可訓練的參數作為一條基因的內容

故此基因向量=1+3+3*3+3=16 維度的向量，如下所示：

$\theta, w_1, w_2, w_3, m_{11}, m_{12}, m_{13}, m_{21}, m_{22}, m_{23}, m_{31}, m_{32}, m_{33}, \sigma_1, \sigma_2, \sigma_3$

各向量初始值範圍：

$\theta, w_i : 0 \sim 1$

$m_{ij} : 0 \sim 30$

$\sigma_i : 10^{-6} \sim 1$ (分母不可為 0)

適應函數為：

$$E(n) = \frac{1}{2} \sum_{i=1}^N (y_n - F(\underline{x}_n))^2$$

N：作業 1 產生的 N 筆成功到達目的訓練資料

y_n ：表示訓練資料的方向盤期望輸出值

目標即利用基因演算法找出一組最佳基因向量(適應值越小越好)

以下詳述基因演算法實作細節

- PoolSize (族群數): 128
- MaxIteration: 20
- CrossoverRate: 0.5, RatioOfCrossover: 0.2
- MutationRate: 0.7, RatioOfMutation: 0.1

起初隨機產生初始族群

→計算各族群適應函數值

→找出最佳適應函數值，觀察是否滿足終止條件(是則終止否則往下)

→複製(採競爭式選擇，複製目前適應值前 10%優良基因，剩餘 90%隨機由 128 族群中挑選兩組比較其適應值)

→交配(兩兩一組共 64 組，每組隨機分配一機率值，若此機率值小於 CrossoverRate 則進行交配)，算式如下：

$$\begin{aligned}x'_1 &= x_1 + \sigma(x_1 \pm x_2) \\x'_2 &= x_2 - \sigma(x_1 \pm x_2)\end{aligned}\quad \sigma \in [-0.2, 0.2]$$

→突變(每個族群隨機分配一機率值，若此機率值小於 MutationRate 則進行突變)，算式如下(s 為 RatioOfMutation)：

$$x = x \pm s \times \text{random_noise}, \text{random_noise} \in [-1, 1]$$

→返回 step2.，迭代 20 次即一次的演化過程

每演化 1 次(=迭代 20 次)會記錄此族群中的最佳基因向量，下次演化時，會複製 1/4*128 個此基因至交配池中，而其他 3/4 依然隨機產生

四、分析

根據此次的設計經驗，20 次迭代為 1 次演化。

原本我使用自己訓練的資料，大概需演化 100 次始能得出好結果，後來加上助教給的補充資料(train4dAll.txt)後只需演化 5 次內即可成功抵達終點，猜測 data 的多寡會強烈影響到訓練網路參數的結果。

基因演算法也可增加迭代次數使網路參數的訓練只需一次即可，但是我認為目前的設計方便觀察得到的參數能不能讓自走車抵達終點，若結果不太好需要重新訓練時能有一個 checkpoint 能接著訓練。