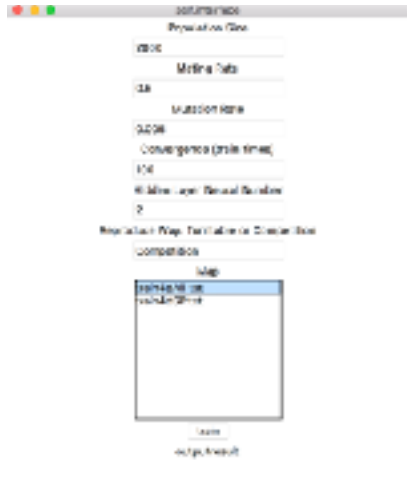


程式介面說明

在ubuntu中執行main檔案，會出現GUI。



選擇想要的數值，並選取檔案，按下訓練

程式碼說明

[graphic.py](#):利用pygame畫出GUI

def mainGraphic() 畫出GUI

def readFile(file) 讀取牆壁和終點

def showWinGraphic(gameDisplay) 顯示結束畫面

[environment.py](#):畫出車子、牆壁、偵測線

class Car(object) 畫出車子、偵測牆壁距離、呼叫模糊系統、改變方向盤角度

class Destination(object) 畫出終點、抵達終點判斷
class Edge(object) 畫出牆壁

[lineIntersectPoint.py](#):額外程式去計算偵測線和牆壁的交匯點

[RBFN.py](#):

class RBFN(object):

def get_steeringWheel(self, straight, right, left):取得偵測距離並利用RBFN計算應轉的角度

def _train(self, convergenceCondition):呼叫基因演算法的程式去訓練找到RBFN最佳數值

def _normalize_Train_Data(self, trainData):對訓練資料做正規化

[genetic_algorithm.py](#):

class Genetic_Algorithm(object):

def _create_gene_pool(self):創造基因庫

def calculate_Fitness_Function(self, trainData):計算適應函數，會使用到gaussian_basis_function和_fitness_Function的function

def reproduce(self):複製方法，有分成：Competition和Turntable

def mate(self): 交配，會先隨機化基因庫，讓基因隨機交配

def mutate(self):突變

def get_optimization_para(self, data_len, i):取得最佳基因

基因演算法說明

使用實數型基因：所有的基因值都位於0~1之間，維度看RBFN神經元個數

競爭式複製方法：先以適應函數值做排列，然後在取前1/2的好基因做複製

交配方式：對基因庫做隨機排列，並在可以交配時，以下列公式交配

$$\begin{aligned}\underline{x}'_1 &= \underline{x}_1 + \sigma(\underline{x}_1 - \underline{x}_2) \\ \underline{x}'_2 &= \underline{x}_2 - \sigma(\underline{x}_1 - \underline{x}_2)\end{aligned}$$

突變方式：對基因加入一些隨機雜訊

誤差值的變化：Competition可以穩定地收斂，但是Turntable因為在選擇基因時，還是有機會選取到壞的基因，所以基因數值可能會發散

訓練時間：Competition訓練時間大概1分鐘。

執行結果

以此數值訓練大概一分鐘可以讓車子過關

self.interface

Population Size

2000

Mating Rate

0.5

Mutation Rate

0.005

Convergence (train times)

100

Hidden Layer Neural Number

2

Reproduce Way: Turntable or Competition

Competition

Map

train4dAll.txt

