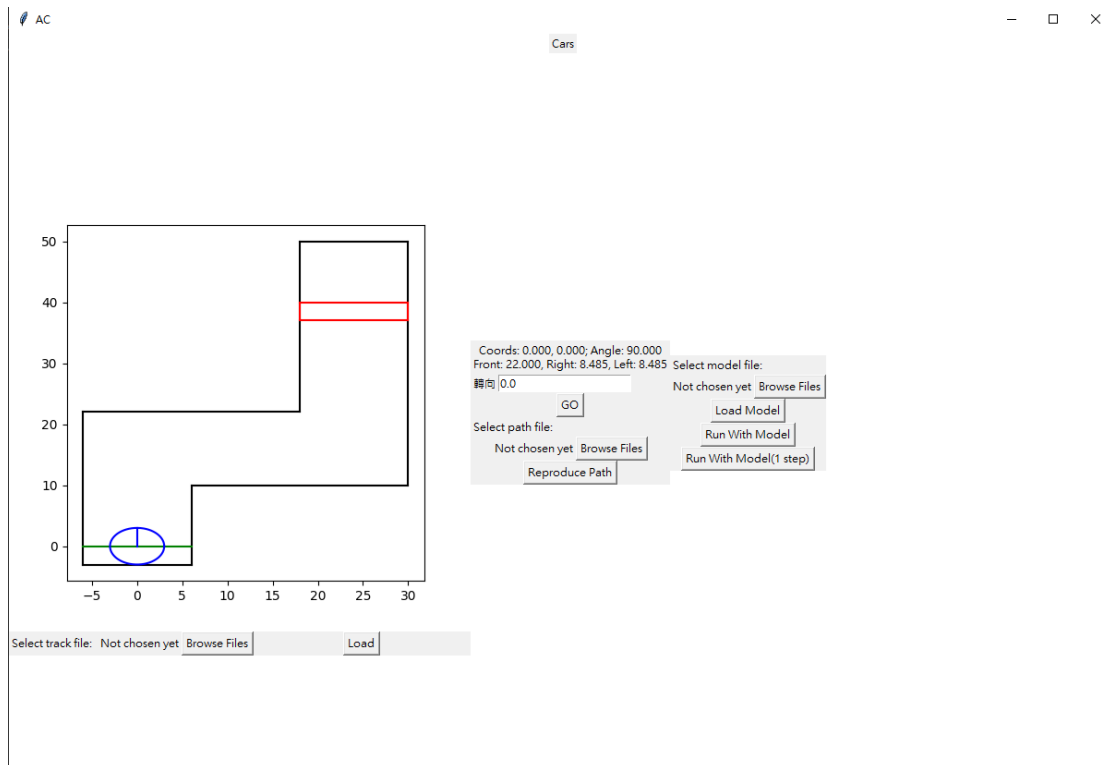


類神經網路-作業二

資工 3A 108502005 戴裕笙

程式介面說明簡介：



- 左側為地圖及車子，車子中的線是朝向的方向，下方可以選擇軌道的檔案並載入
- 中間為車子目前的資訊，包含：座標、角度、三個方位到牆壁的距離，下方可以輸入轉向並手動執行，若到達終點會將路徑輸出，再下面可以將路徑導入並重現(限 6D)
- 右側可以選擇存好的 model 資料並載入，可以選擇讓他直接跑到終點或是一步一步移動

程式碼說明：

- `simulator.py`

先有簡單的角度轉換 `function(rad<->degree)`

然後定義了 `Car` 這個 `class`，裡面包含了各種可以對車子進行操作的
`function`:

- `reset` 可以將車子隨機在起跑線上重置，或是指定要在哪個座標重置
- `run` 會根據傳入的 `theta` 對車子進行移動
- `distance` 可以求得車子各個方位到牆壁的距離，其中
`will_touch_line` 是判斷車子對該方向的延長線是否會接觸到特定的牆壁，`distance_to_line` 是算出上述的距離
- `reach` 用來判斷車子是否有碰到牆壁或終點，其中
`point_distance_line` 為車子中心到特定線段的距離
- `logging` 將目前的資訊儲存，分為 4D 及 6D 兩種版本
- `dumplog` 將儲存的資料分別輸出到兩個檔案

另外還分別有 `Dot` 和 `Line` 兩個 `class` 分別是座標點和線段

最後是 `load_map` 這個 `function` 可以從檔案載入地圖，會回傳 `car`,
`end`, `map`, `start`，分別是車子、終點、地圖、起點線

● MLP.py

先定義了模型中常用的公式，sigmoid，將期望值正規化及還原，還有算出瞬間均方誤差

接著是 neuron 這個 class，代表模型中的神經元，可以用決定好的 weight 或是隨機生成，activation 可以求出對於某個輸入而言，神經元內部的結果 v 和經過 sigmoid 函數的輸出 y

後面 MLP 這個 class 就是模型的本身，也有許多 function：

- load_data 是將訓練的資料讀入，並隨機分為訓練集與測試集
- load_model 若有訓練完成的模型可以直接匯入，若無則初始化一個
- train 會根據設定好的參數開始訓練，在每個 epoch 結束時都會進行 test 決定是否終止(有多個條件)
- Eavg 可以算出每筆資料的 E 的平均值
- dump_log 是將訓練完的模型輸出
- gety 是根據 x 回傳對應到的輸出(有將正規化的輸出還原為車子的轉向角度)

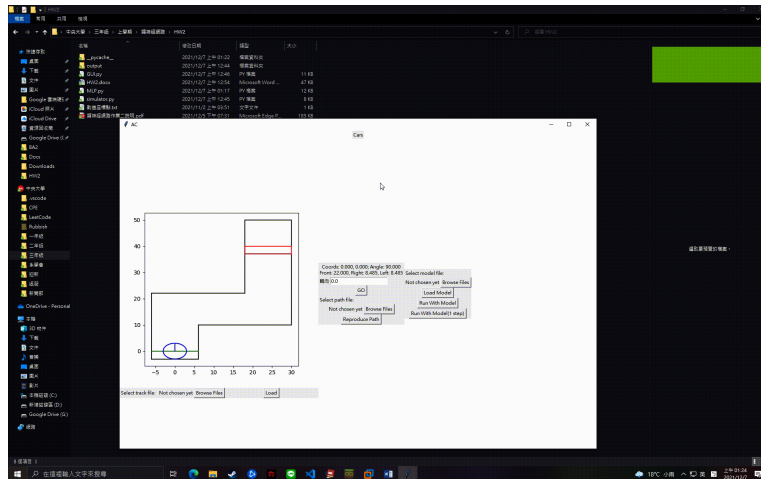
如果單獨執行 MLP.py 的話會直接開始訓練

- GUI .py

處理圖形化介面的部分，大多的 function 都是用來處理畫圖的部分，其

他可以讓車子移動的部分便使用了上面兩個檔案定義好的各種功能

實驗結果與分析：



(圖片有超連結可以點到影片檔案)

經過充分的訓練後，在絕大部分情況下車子都能不撞到牆壁且抵達終點