

# 類神經網路作業二

學號：111526009 姓名：薛竣祐

## 一. 程式執行說明

程式提供兩執行檔，一為exe執行檔，一為Unix執行檔，兩執行檔結果相同，擇一執行。確認training資料及軌道資料在同一執行目錄下後，執行後會呈現此畫面，可為Model設定Feature數量及設定關注的核心數量。

Feature Count：4D為選擇train4dAll資料；6D為選擇train6dAll資料。

Focus Count：輸入關注核心數量，4D預設為800，6D預設為1000，可自由在0以上整數選擇，數值越大執行時間需越久。

完成設定後，請點擊Run按鈕，等待執行完畢後會在下圖顯示執行結果，可重複執行來呈現不同結果，並可以使用Reset All按鈕來清除所有結果。



## 二. 程式碼簡介

Source Code為Python，分為四個檔案。

run.py：程式進入點及介面的呈現。

model.py：RBF模型，包含訓練及預測的函式。

simple\_playground：修改自助教提供之程式，將model應用於playground，並修改部分錯誤程式及簡化邏輯。

simple\_geometry.py：助教提供之Point2D及Line2D程式。

主要使用到的套件有：numpy、matplotlib、tkinter。

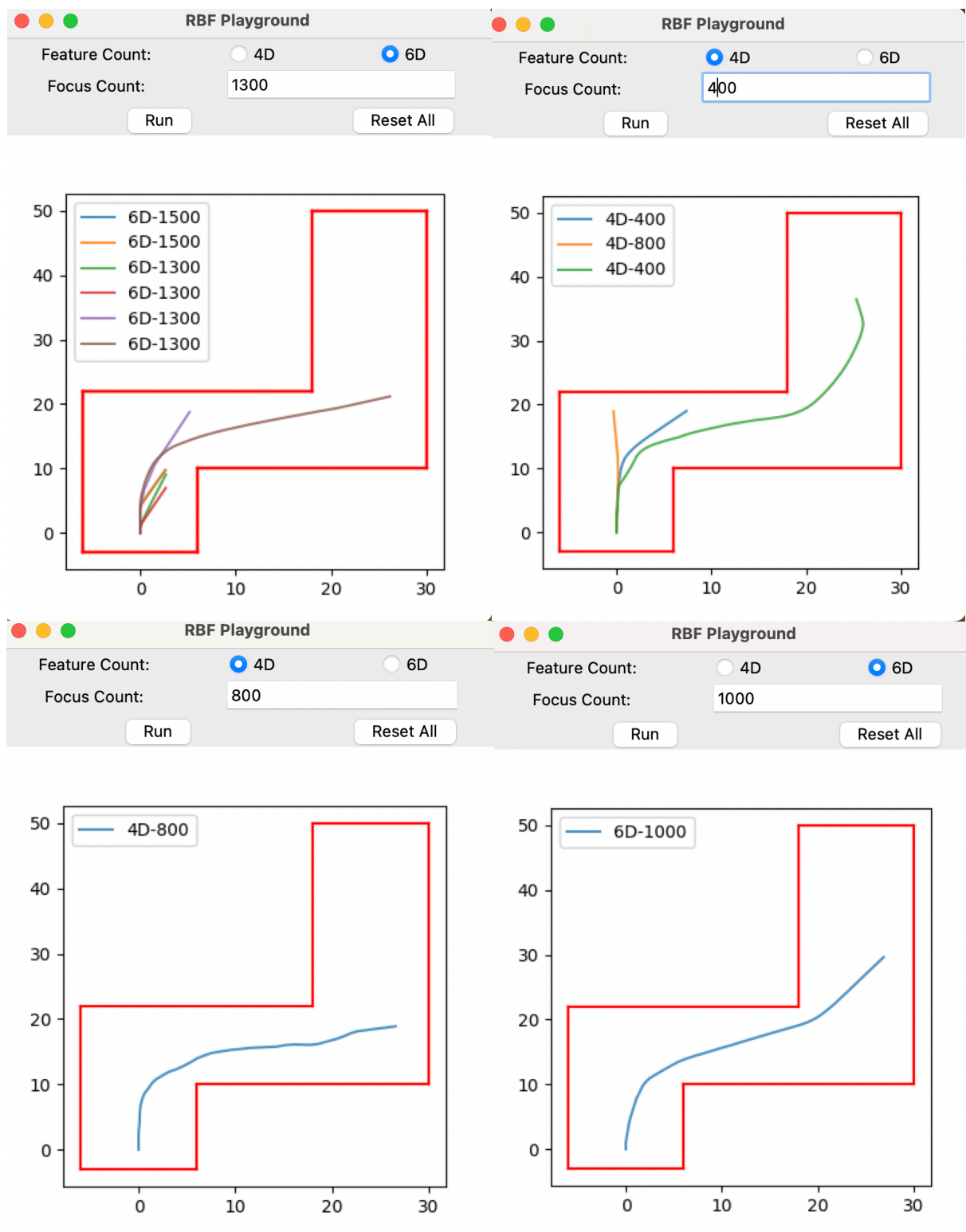
GUI使用tkinter製作畫面，設計了一個int input、一個radio input、一個執行按鈕、一個重置按鈕。並使用mainloop function來等待畫面關閉，並設計多個callback函式來設定各按鈕的行為。

其中run按鈕會根據設定值執行model訓練，訓練完後於playground模擬自走車執行過程，並將過程紀錄下來，並使用FigureCanvasTkAgg來將matplotlib結果顯示於tkinter，最後更新畫面來顯示圖表。

### 三. 實驗結果

因執行結果有一定的隨機性，每次執行結果並不相同，根據多次實驗，普遍兩種訓練資料都只能走第二個轉角處，並有機率走完路徑或於第一轉角處停下。

以下四張圖展示在不同feature及focus count狀況下的各種結果，大多數狀況無法完成路線。較意外的是6D資料並不會比4D資料結果還好，甚至平均而言效果更差。



---

## 四. 實驗結果分析與討論

以結果來說模型實驗下來的效果不佳，但也有可能是因為模型的設計有問題，並且以目前模型設計的狀況來說6D的效果較4D還差。通常最容易遇到的狀況是太早轉彎，為了彌補這一個現象，我也做了幾個嘗試。第一種是強制第一次朝向正前方前進，並當下的轉彎幅度作為下次的轉彎方向。第二種是紀錄每次的轉彎方向，並把最後N次轉彎幅度平滑化。第三種是將輸入資料做min-max正規化，可以在simple\_playground裡找到相關的func，但這幾種的效果都不佳，因此最後並沒有採用。為此可以猜測最大的失敗原因比較有可能是因為模型本身設計的不好，未來可以朝這方面做改進。