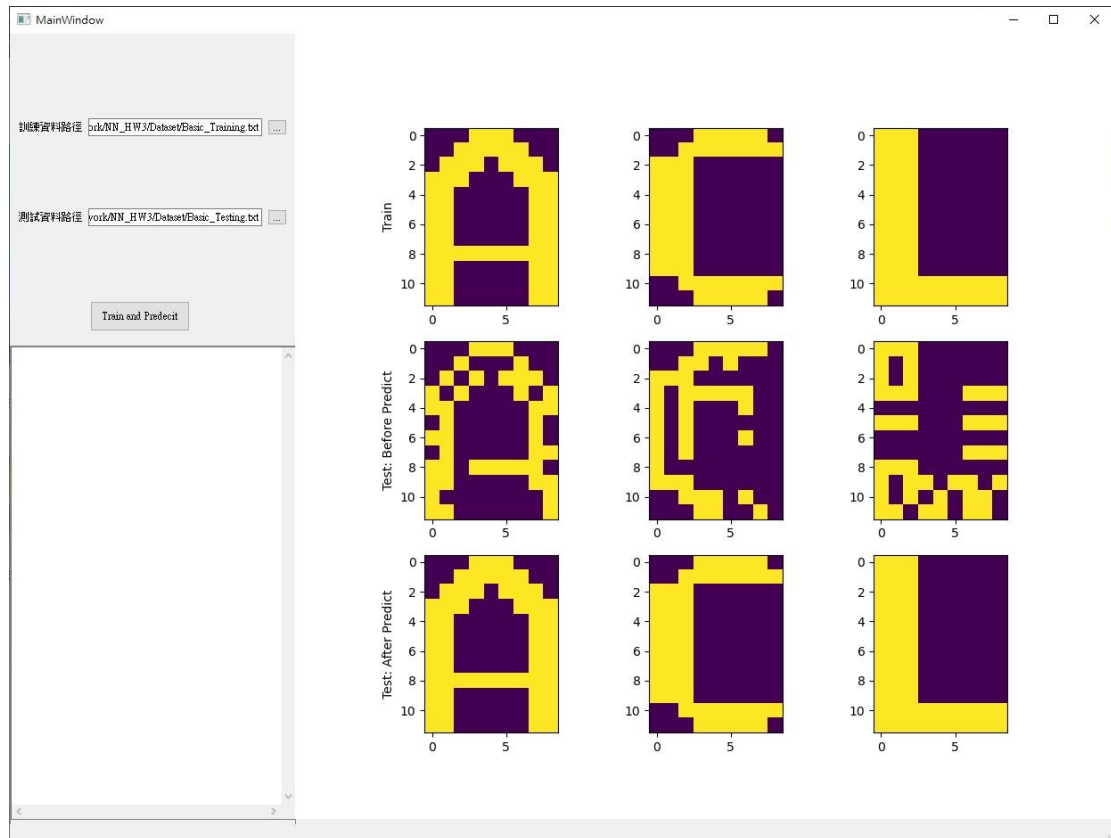


# 類神經作業三—Hopfield

111522094 資工碩一 涂建名

## 一、程式執行說明



選擇訓練資料和測試資料路徑後，按 **Train and Predict button** 就會跑出右邊這張圖，第一行顯示的是訓練資料、第二行顯示的是測試資料、第三行顯示的是從測試資料聯想出來的結果

## 二、程式碼簡介

### 1. Hopfield.py

Hopfield 網路是實作下面的演算法而得，本次預測是採用同步的方式來預測

### 5.3 Hopfield 網路 (2)

一、網路學習:假設有  $N$  筆輸入向量  $\mathbf{x}_i = [x_{i1}, \dots, x_{ip}]^T, i = 1, \dots, N$   
要用自聯想的方式儲存至離散 Hopfield 網路上

$$w_{ji} = \frac{1}{p} \sum_{k=1}^N x_{ki} x_{kj}$$

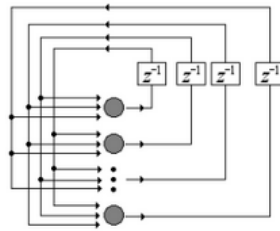
$$\theta_j = 0, j = 1, \dots, p$$

- 或者

$$\theta_j = \sum_{i=1}^p w_{ji}, i = 1, \dots, p$$

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1p} \\ \vdots & \ddots & \vdots \\ w_{p1} & \cdots & w_{pp} \end{bmatrix}$$

$\rightarrow$  不想正回授  $\rightarrow$  量越来越小  
 自己通过自建神经网络  
 $E = \left( \frac{1}{2} \right) \left( (1+1) + \left( \frac{1}{2} \right) (1+1) + \dots \right)$   
 $= \begin{bmatrix} N & & \\ & N & \\ & & N \end{bmatrix}$  但有希望训练  
 $\Rightarrow \begin{bmatrix} N & & \\ & N & \\ & & N \end{bmatrix} - N I$



W is a symmetric matrix  
∴ 配・配

圖5.2：離散的 Hopfield 網路。

### 5.3 Hopfield 網路 (3)

## 二、網路回想

- 假若此時有一輸入  $x$  進入此網路，我們將此時的輸入視做網路的初始輸出  $x(0)$ ，緊接著，每個類神經元的後續輸出是由下式計算

$$x_j(n+1) = \text{sgn}\left(\sum_{i=1}^p w_{ji} x_i(n) - \theta_j\right) = \text{sgn}(u_j(n) - \theta_j)$$

目标神经元的

$$= \begin{cases} 1 & \text{若 } u_j(n) > \theta_j \\ x_j(n) & \text{若 } u_j(n) = \theta_j \\ -1 & \text{若 } u_j(n) < \theta_j \end{cases}$$

- 離散 Hopfield 網路是用非同步 (asynchronization) 的方式來變更類神經元的輸出，整個聯想過程可以用下列之鏈狀關係來描述：

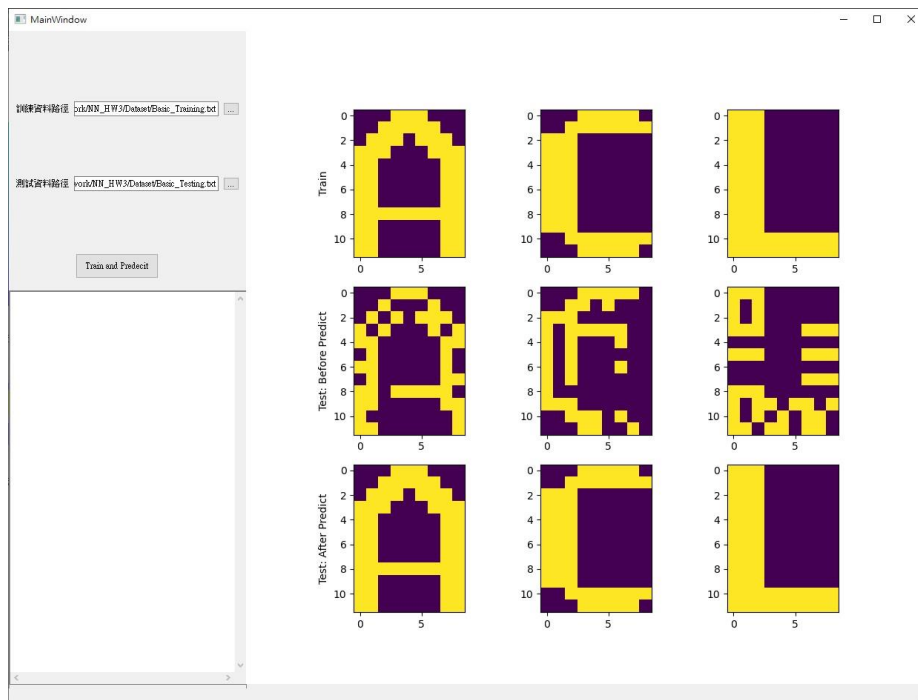
$$\underline{x}(0) \rightarrow \underline{x}(1) \rightarrow \underline{x}(2) \rightarrow \cdots \rightarrow \underline{x}(k) \rightarrow \underline{x}(k+1) \rightarrow \cdots$$

- 如果我們用同步 (synchronization) 的方式來變更網路輸出，結果會有些不同

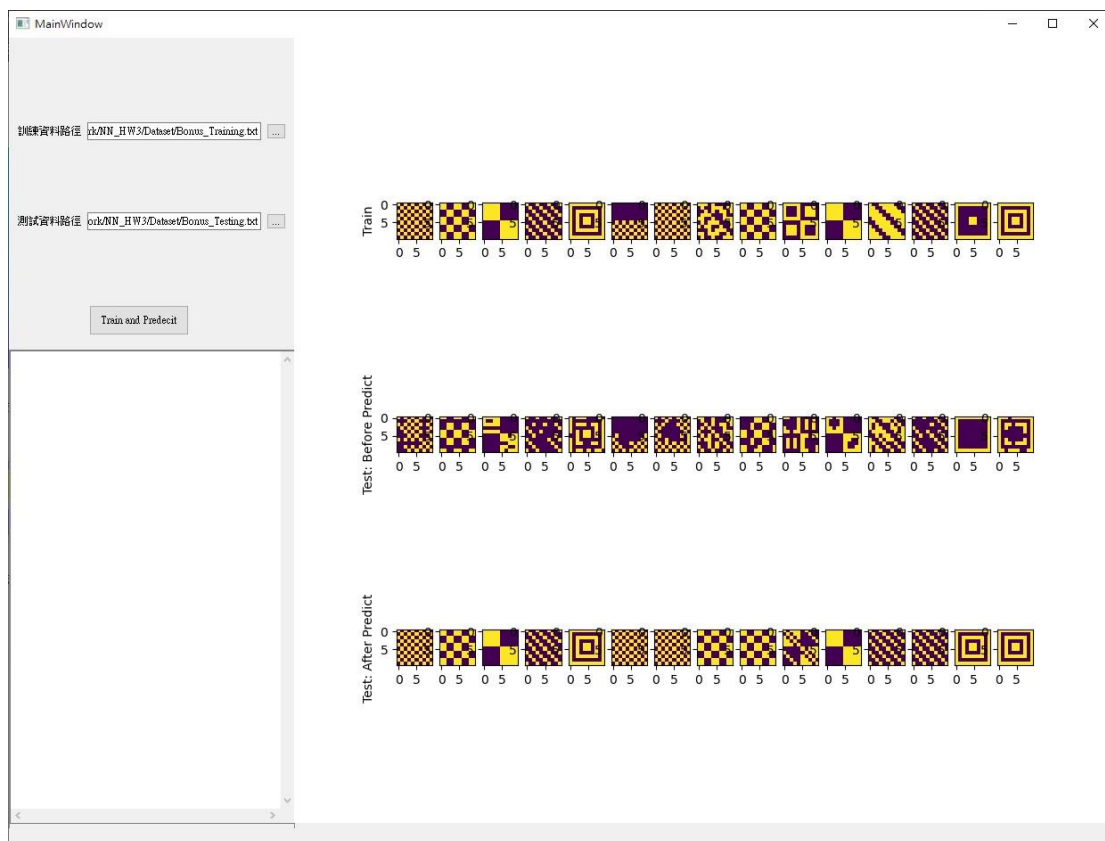
2. `Windows.py`：實作 GUI 的程式
3. `result_visible.py`：印出右邊的結果
4. `main.py`：將 GUI、`result_visible.py`、`Hopfield.py` 三支程式串起來的主程式

### 三、實驗結果

#### 基本題



#### 進階題



#### 四、實驗結果與分析:

**Hopfield** 對於基本題可以準確的偵測出來，但對於複雜的圖形可能就只能推論出來部分正確，推測也許用非同步可以得出比較好的結果?