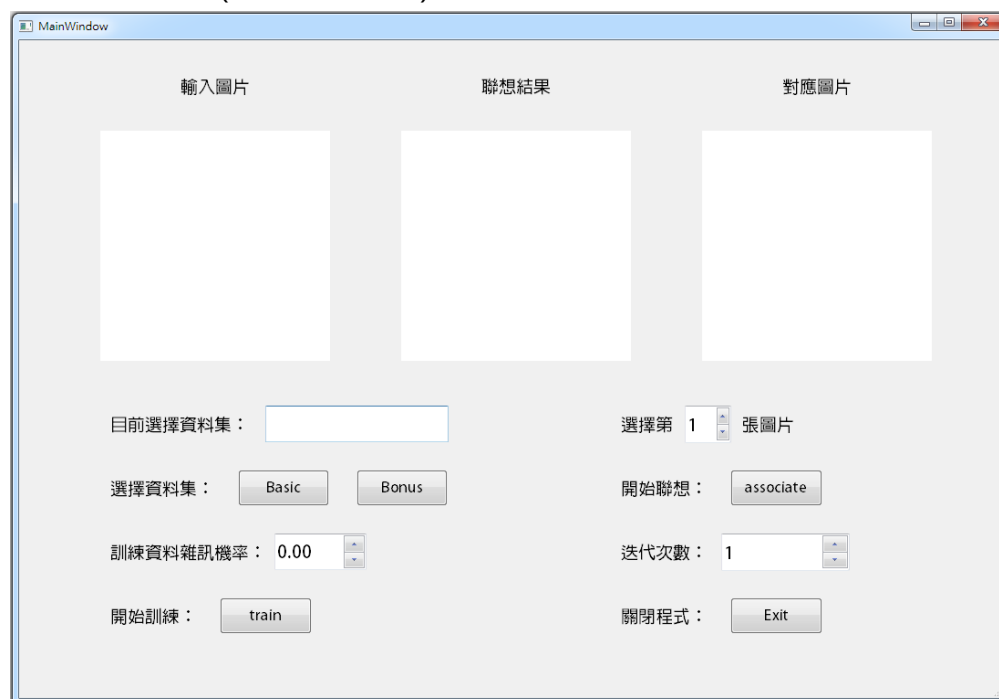
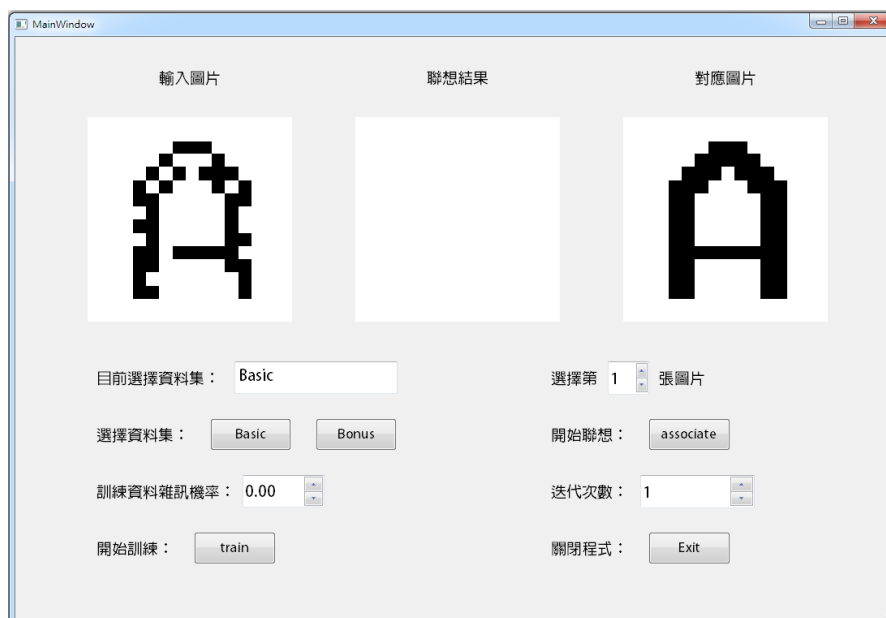


# 類神經網路作業三 - Hopfield

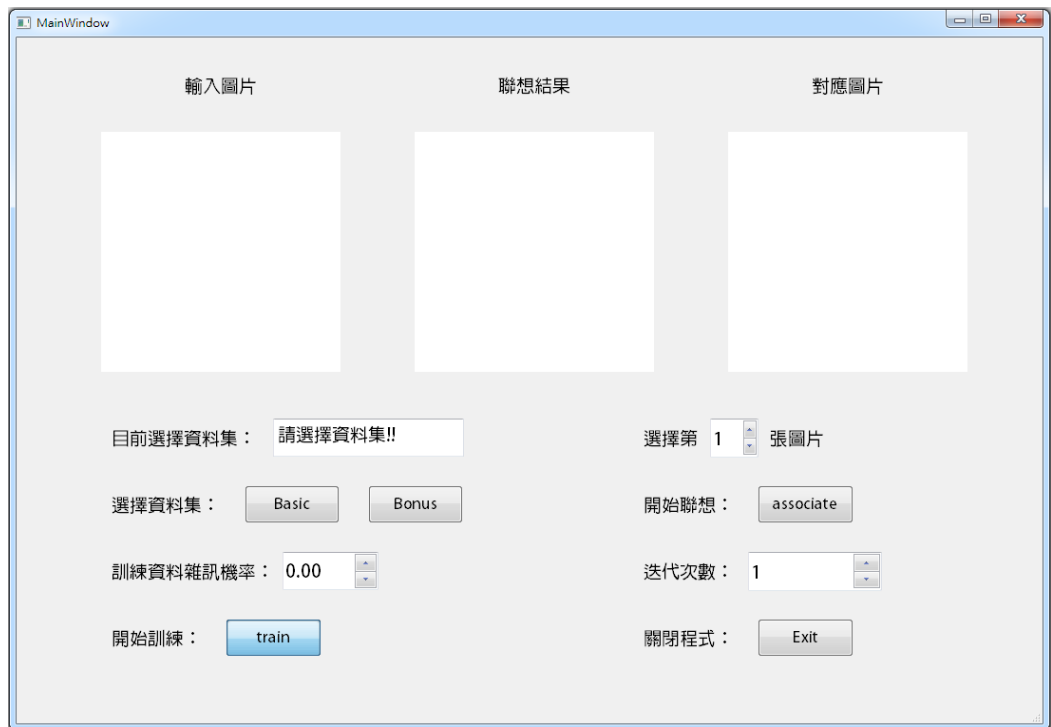
## 一、程式執行說明 (GUI 功能說明)



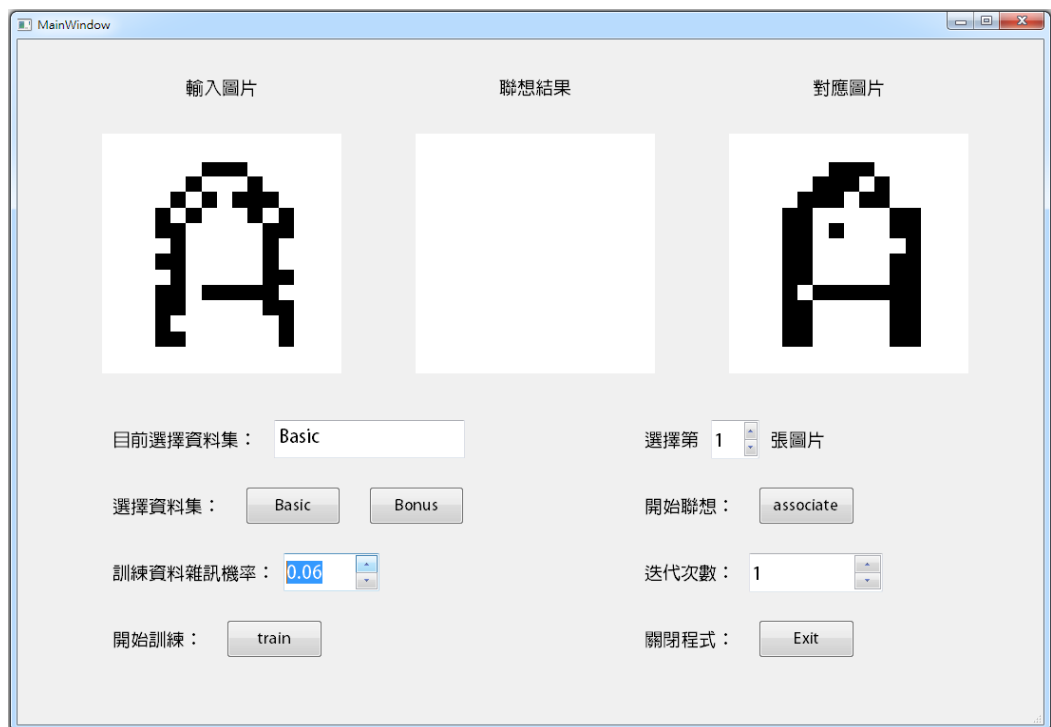
1. 進入畫面長這樣，這時要先選擇資料集，按下旁邊的 Basic 就會使用 Basic\_Training.txt 作為訓練資料，Basic\_Testing.txt 作為測試資料，並在上方的文字框中顯示目前選擇的是哪一種。需要注意的是，資料夾 Hopfield\_dataset 必須要與執行檔案位於同一層，否則程式無法找到檔案，會發生錯誤並自行關閉



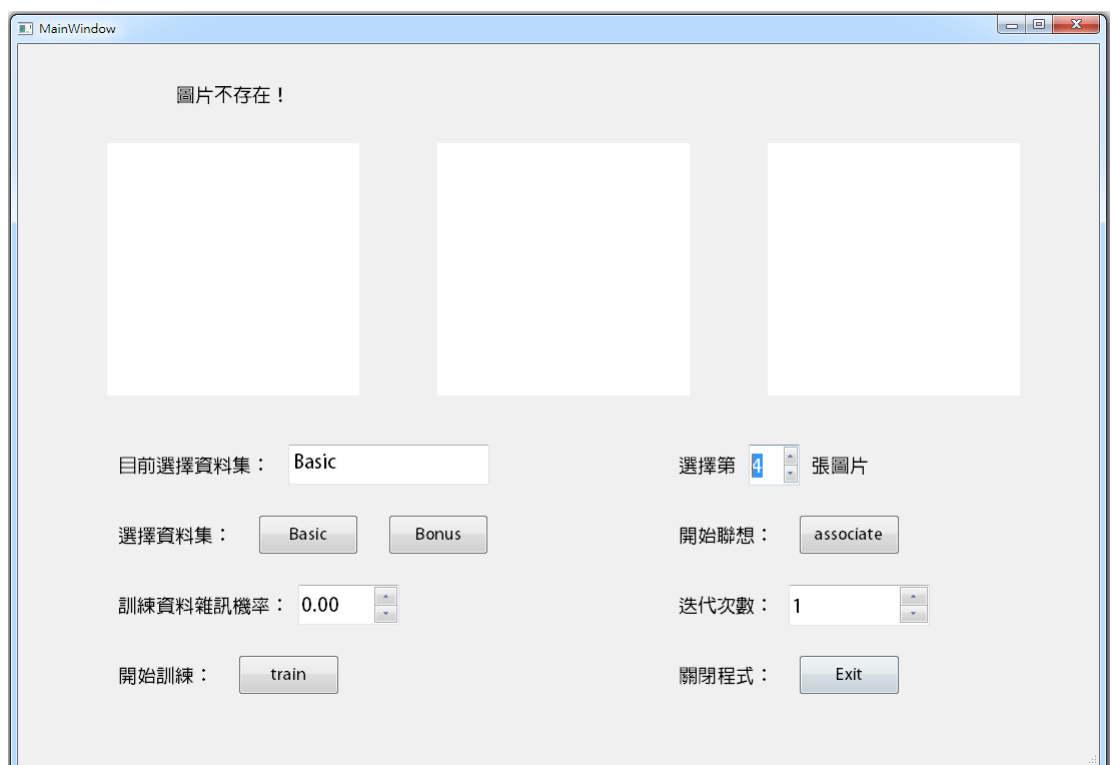
2. 若沒有選擇資料集，就更改訓練資料雜訊、按下開始訓練的 train 鍵等等的話，上方的文字框就會跳出「請選擇資料集！！」



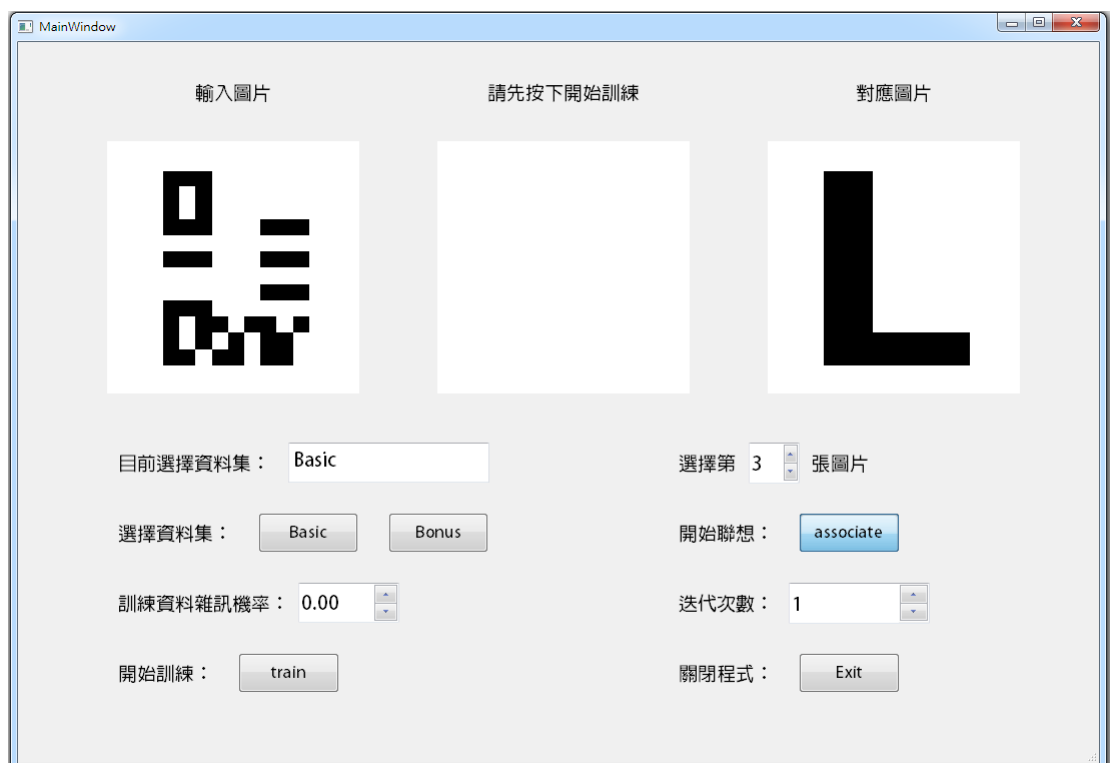
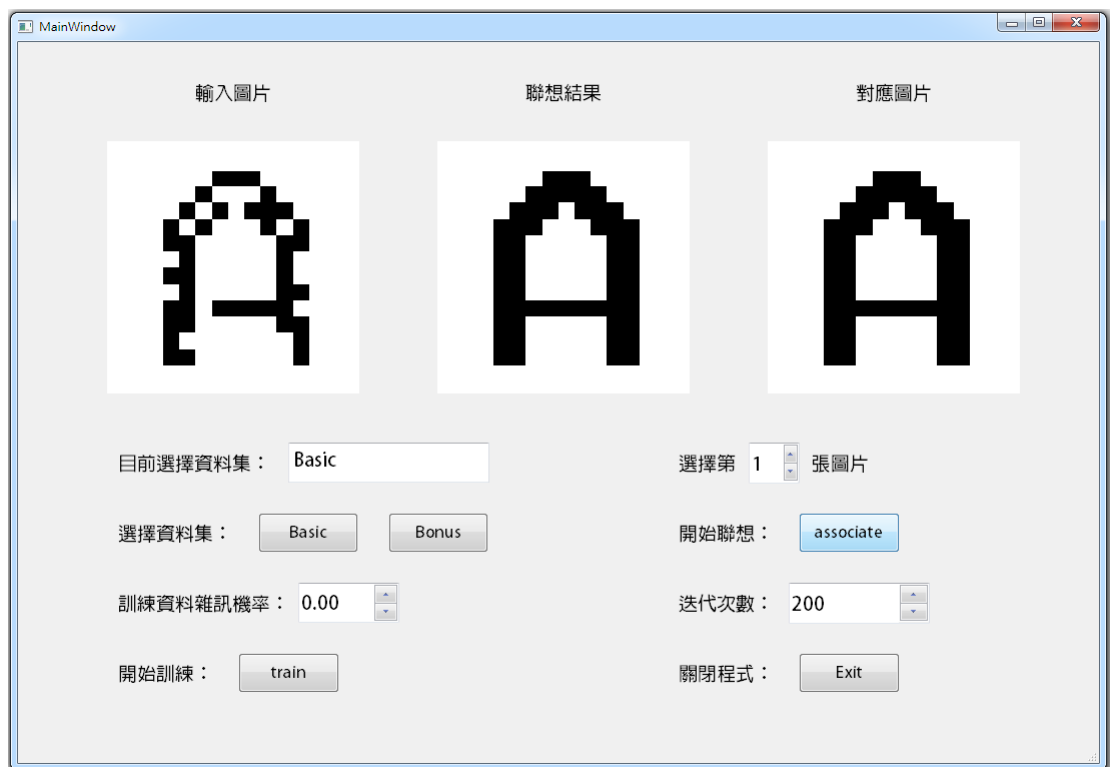
3. (加分項目) 訓練資料雜訊機率可以調整輸入的訓練資料，只要該值一更改，就會立刻將加入雜訊的訓練資料顯示在最右方的對應圖片中。雜訊機率值介於 0.00~1.00。



4. 可以用 spinbox 選擇圖片，當值更改的同時就會顯示新的圖片。若數字大於現有的訓練資料的張數，原本在左方的輸入圖片的文字會變成「圖片不存在！」

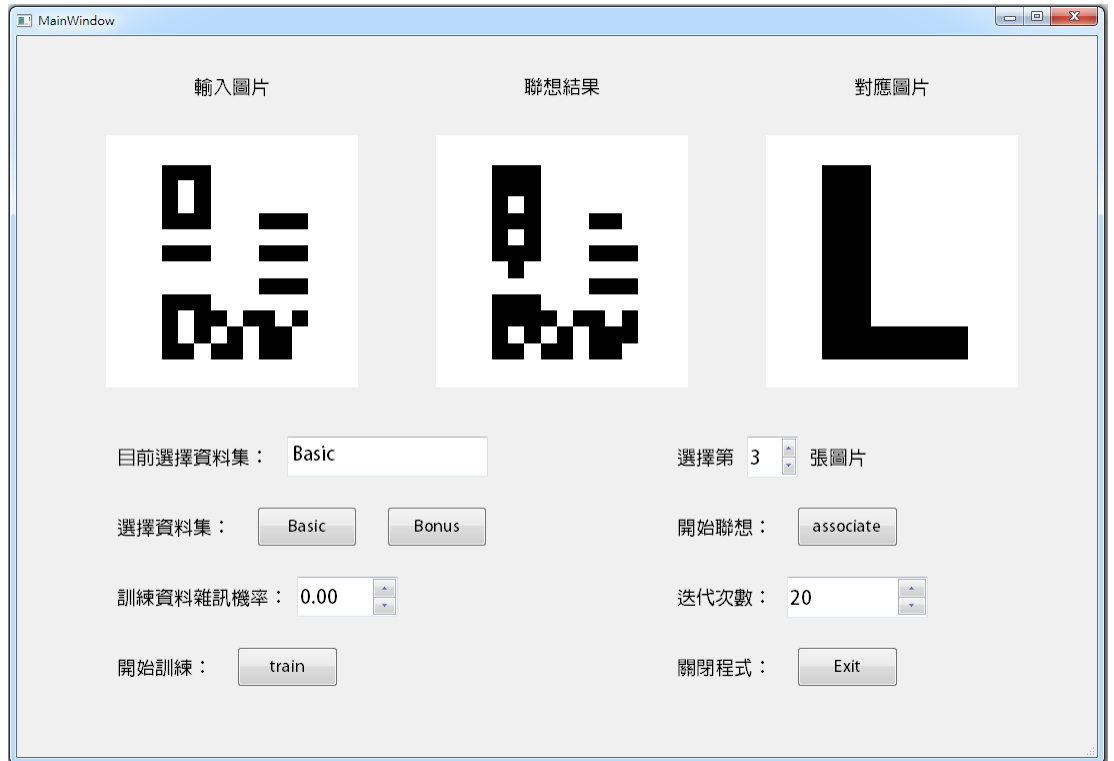


5. 按下 **associate** 鍵可以開始聯想，且結果會顯示在中間的聯想結果中。若選擇了資料集及圖片，卻沒有先按下 **train** 的話是無法聯想的，這種情況中間的聯想結果的文字會變成「請先按下開始訓練」提醒使用者。



6. 迭代次數就是指更改了幾次，一次會隨機取一個像素(方格)，所以若迭代次數很低，圖片就不會完全聯想完成。測試過了幾次，大概 200 次以上才會穩定。

以下是 20 迭代的結果，可以看到只有部分得像素被更改。



7. Exit 按鍵或是右上角 X 可以關閉程式

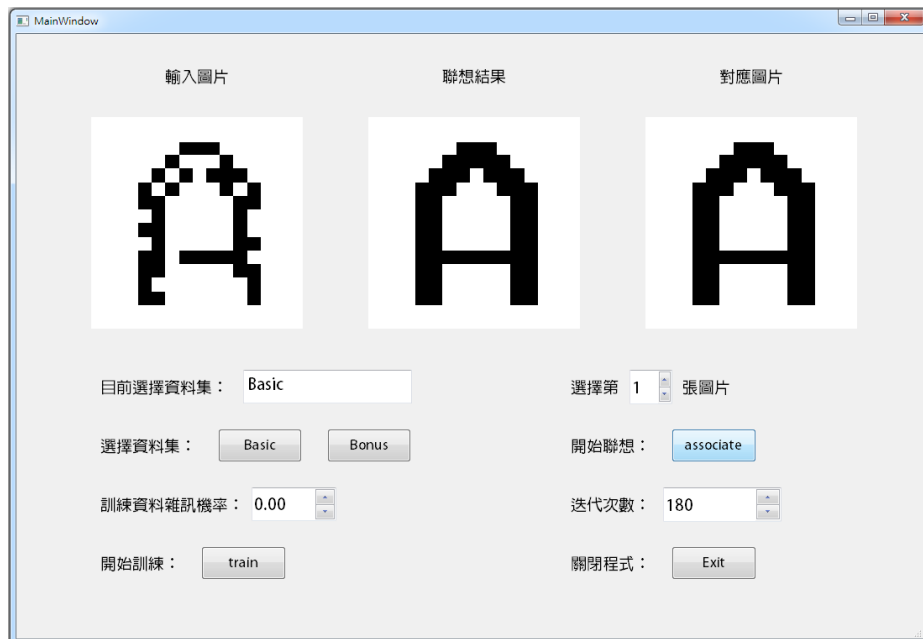
## 二、程式簡介 & 實作架構：使用 Hopfield

1. UI.py 為用 Qt designer 設計出來的介面轉成的程式碼
2. Mplwidget.py 則是將 matplotlib 嵌入介面所需的程式碼，裡面有 MplInput、MplOutput、MplAnswer，分別對應上方三個圖片的畫布。
3. Hopfield.py 則是主程式，包含介面的按鈕與程式碼的綁定、讀寫檔案、訓練網路、畫圖等等
  - A. \_\_init\_\_ 用於初始化及設定 UI
  - B. setup\_control 則是將 button 與功能連結在一起，當按下某個 button，或是 spinbox 發現值有更變時，則會做某項函式
  - C. open\_basic 和 open\_bonus 更改檔案路徑和文字框的顯示，並讀入檔案、顯示圖片(預設為第一張)。open\_basic 在按下 Basic 按鈕後執行，open\_bonus 在按下 Bonus 按鈕後執行。

- D. `check_data` 是檢查有沒有選取好資料集的函式，避免在未讀入檔時就執行其他動作造成程式錯誤並關閉。
- E. `load_train` 是讀入訓練資料的函式。(加分項目)會根據雜訊機率的值的不同，決定讀入的值要不要進行翻轉。具體實作是使用 `np.random.randint` 與雜訊機率( $\times 100$ )來作比較，若小於雜訊機率值，則翻轉。
- F. `load_test` 是讀入測試資料的函式。
- G. `train` 是網路學習的函式，即初始化 `w` 及 `theta` 的地方。
- H. (加分項目) `random_change` 是在雜訊機率更變時執行的函式，他會重新執行 `load_train` 讀入新的訓練資料，並顯示新的圖片
- I. `show_input` 是顯示現在選擇的圖片及其對應訓練資料的函式，利用 `matplotlib` 的 `imshow` 達成。如果圖片不存在則顯示空白
- J. `associate` 是網路聯想的函式，會先取得迭代次數的值，再來執行。每次迭代會隨機選擇一個像素，並決定是否要更改其值。迭代完成後會顯示聯想的結果。如果還沒有按過開始訓練則不會開始聯想。
- K. `exit` 是按下 Exit 按鈕後關閉程式的函式

### 三、實驗結果及其分析與討論：Basic

#### 1. 第一張：英文字母 A



迭代次數大概 180 就可以聯想成功，算是簡單的圖案，但因為更改得像素是隨機取的關係所以迭代次數設 180 按下去不一定會成功，要多按幾次，或是將迭代次數調高。

## 2. 第二張：英文字母 C



同上，算是簡單的圖案。

## 3. 第三張：英文字母 L



和字母 A 跟 C 不同，L 我設 180 次幾乎無法成功，這應該是因為測試資料要更改的像素點比較多，但是每次迭代不一定能選取到的關係。

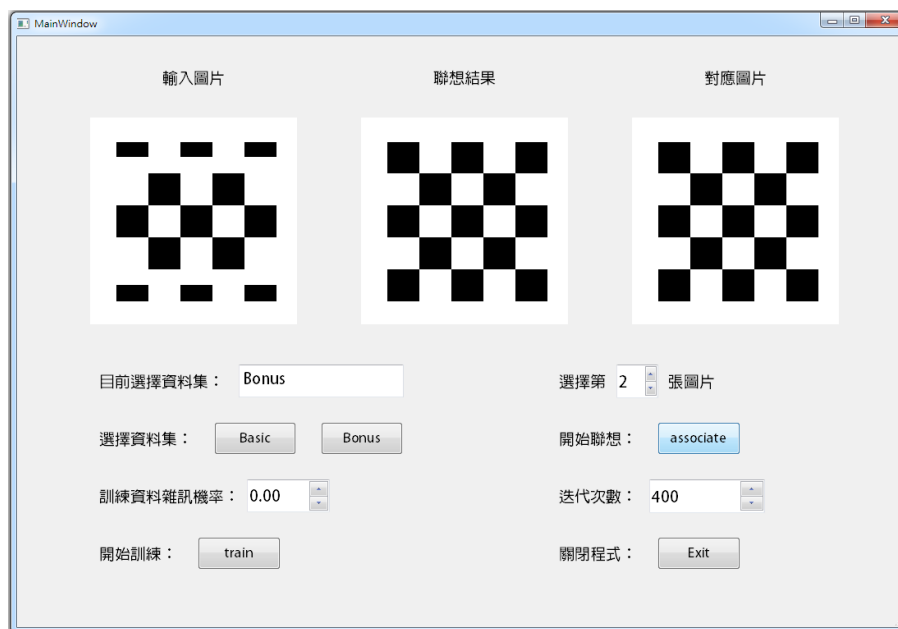
#### 四、（加分項目）實驗結果及其分析與討論：Bonus (部份成功)

由於 Bonus 圖片較複雜的關係，所以很多都回想失敗。我想這是因為訓練資料集的圖片太多，且相異不大，所以很容易互相干擾，而且每次選擇更改的像素順序會影響到結果。以下失敗的張數只會擷取圖片，不會放整個程式畫面。

##### 1. 第一張：失敗，迭代 300 次



##### 2. 第二張：成功，但機率不高，迭代 400 次

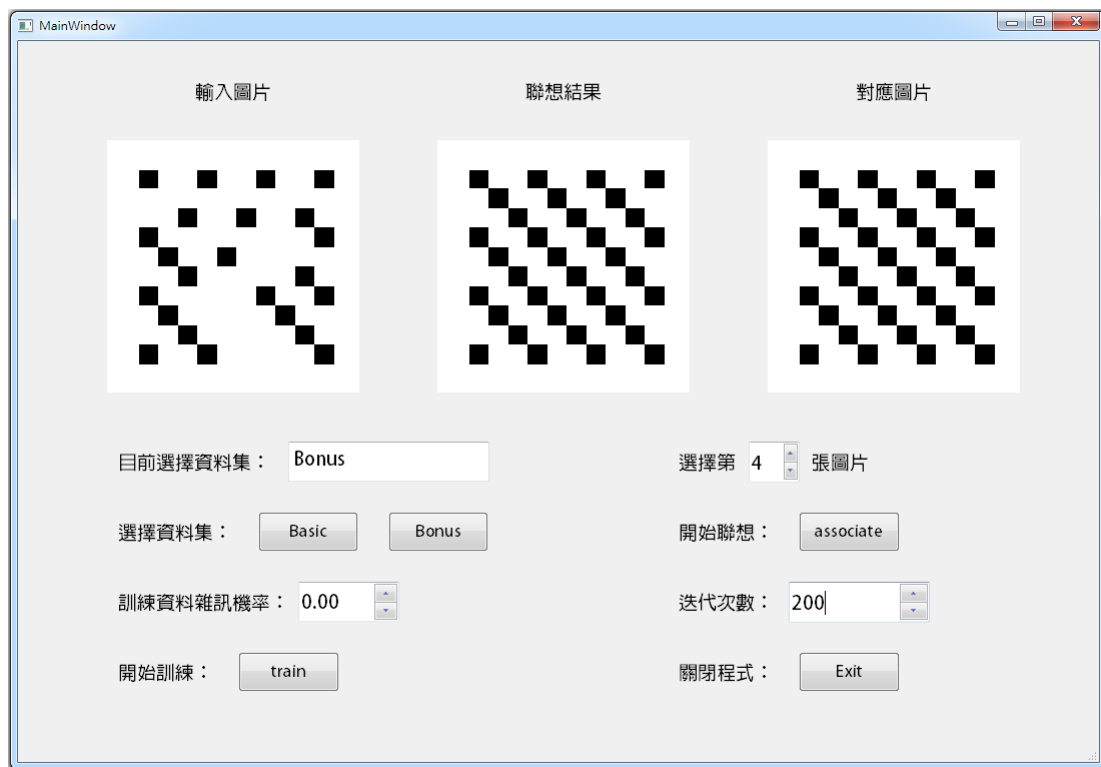


##### 3. 第三張：失敗，迭代 200 次，通常只有右下角那一塊能聯想的比較好





4. 第四張：成功，且穩定回想，迭代 200 次即可聯想成功，400 次幾乎一定能成功



5. 第五張：失敗，迭代 400 次



6. 第六張：失敗，迭代 200 次。



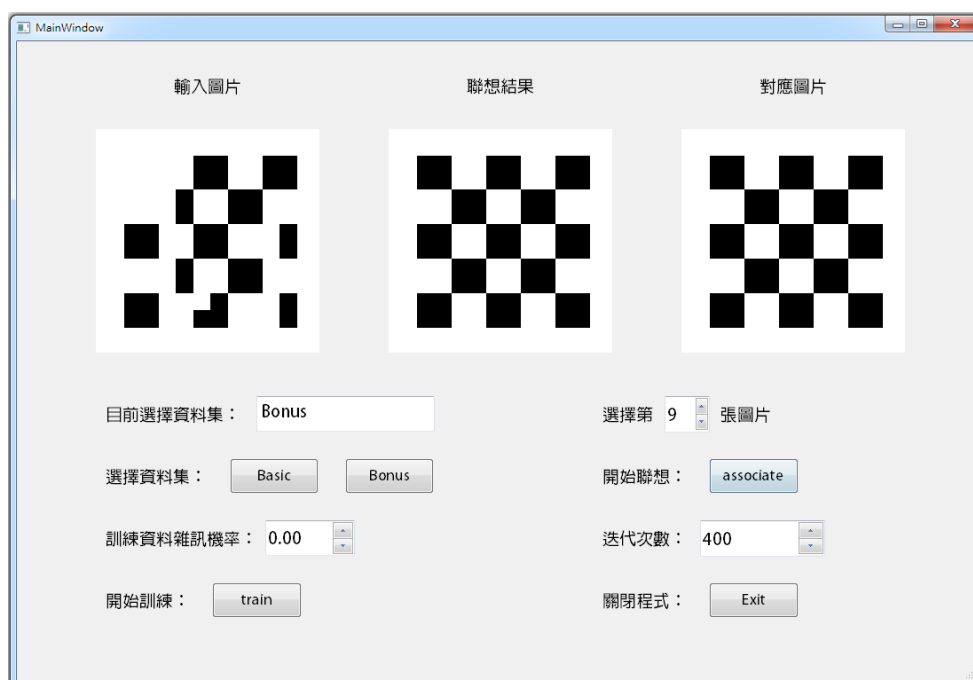
7. 第七張：失敗，迭代 300 次



8. 第八張：失敗，迭代 300 次



9. 第九張：成功，機率極低，迭代 400 次



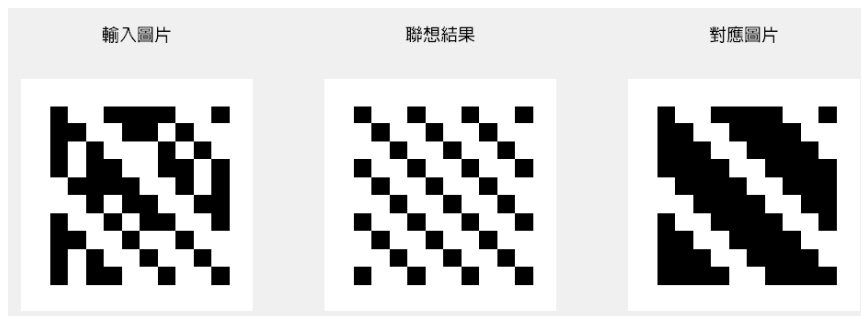
10. 第十張：失敗，迭代 400 次



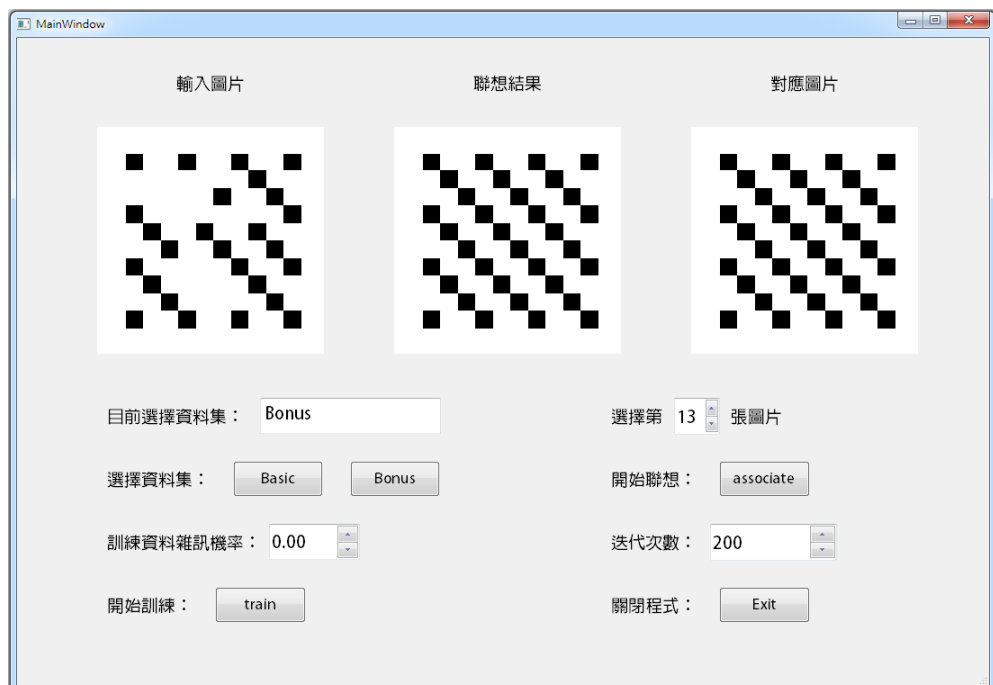
11. 第十一張：失敗，迭代 400 次



12. 第十二張：失敗，迭代 400 次



13. 第十三張：成功，迭代 200 次即可成功



14. 第十四張：失敗，迭代 400 次



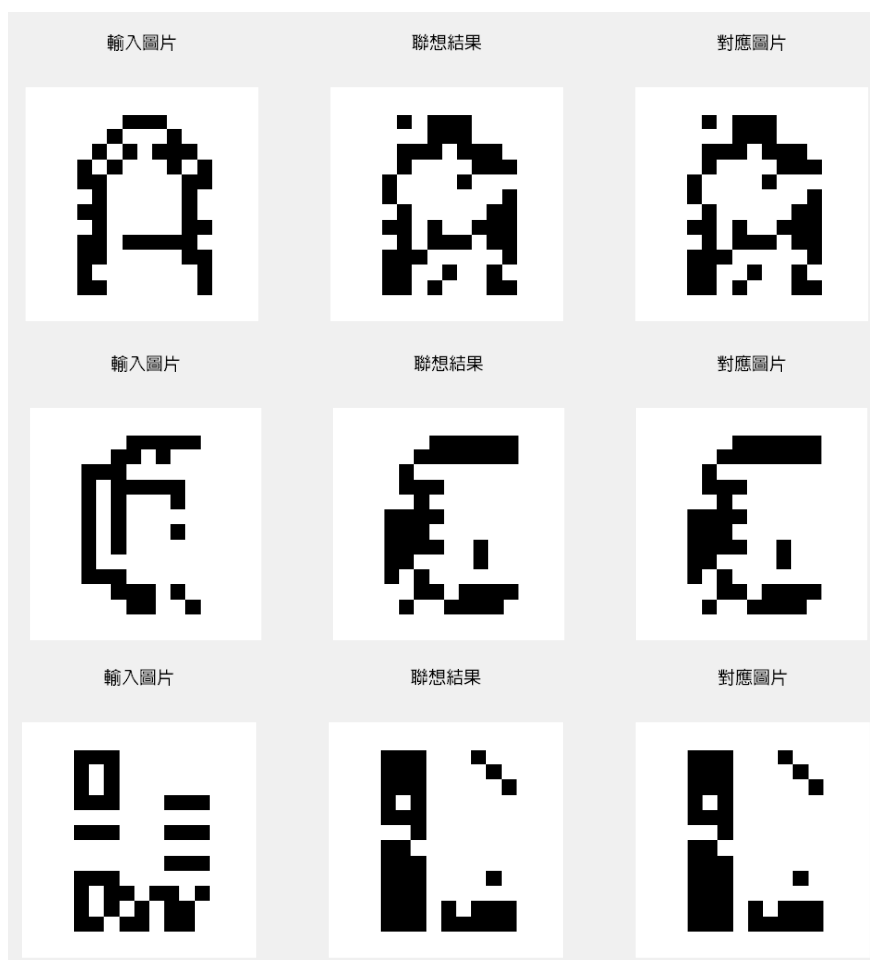
## 15. 第十五張：失敗，迭代 400 次



## 五、（加分項目）將訓練資料集加入雜訊之實驗結果

如果雜訊不幸加在三個資料會互相影響的地方，則有機會不會聯想成功，滿看運氣的。在三個都辨識成功的前提之下，雜訊機率最高大概能設到 0.30，再高很難辨識成功。

### 1. 雜訊機率設 0.15 的結果，以下迭代皆為 400 次，為方便觀看只截圖片



## 2. 雜訊機率設 0.30 的結果，迭代皆為 400 次



## 六、實作問題

1. 程式執行時，須把 Hopfield\_dataset 資料夾全部放在同一層中，且把訓練資料和測試資料放在裡面，不然會無法執行，這點其實滿不方便的，但想不到更好的寫法。
2. 原本有想要有逐步執行的功能，但將 matplotlib 的畫面更改 (draw 等等)設在迴圈內幾乎沒有效果，加上 timer 或 sleep 等相關函式也不會緩慢刷新畫面，只會讓程式計算變得很久、很卡，但是畫面只會呈現最終結果，可能要用 Qthread 來解決，不過因為對於 Pyqt 結合 thread 完全不熟悉所以只好放棄。