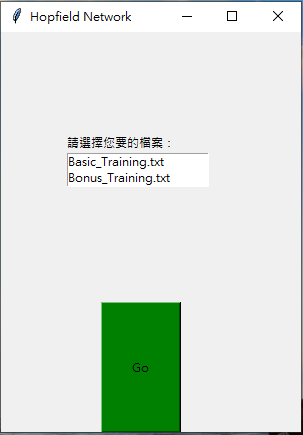
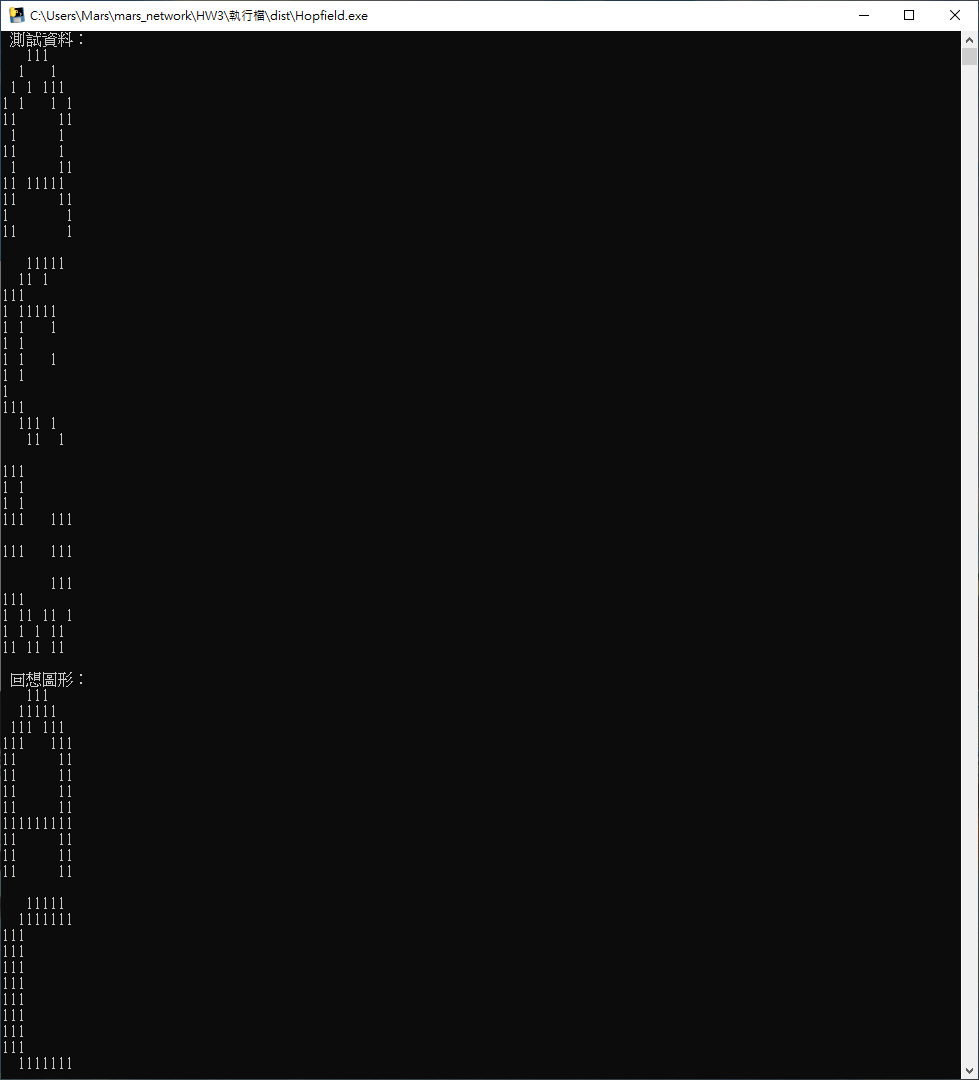
類神經網路—作業三(Hopfield)

電機3A 106501004 王孟洋

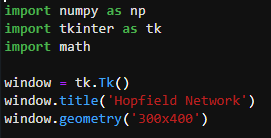
一、程式執行說明:

左下圖此為exe檔打開時會出現的圖形介面，使用者可以選取想要哪組資料(Basic\_Training以及Bonus\_Training)做回想，並且選取，點選以後按下Go的按鈕，在我們的視窗中就會跳出我們的測試資料和我們回想的資料做顯示(如右下圖所示)。

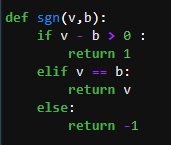


二、程式簡介:

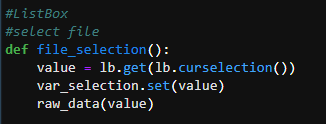
程式開頭，我們將我們會用到的套件import進來做使用，並且呼叫出我們的圖形介面視窗。



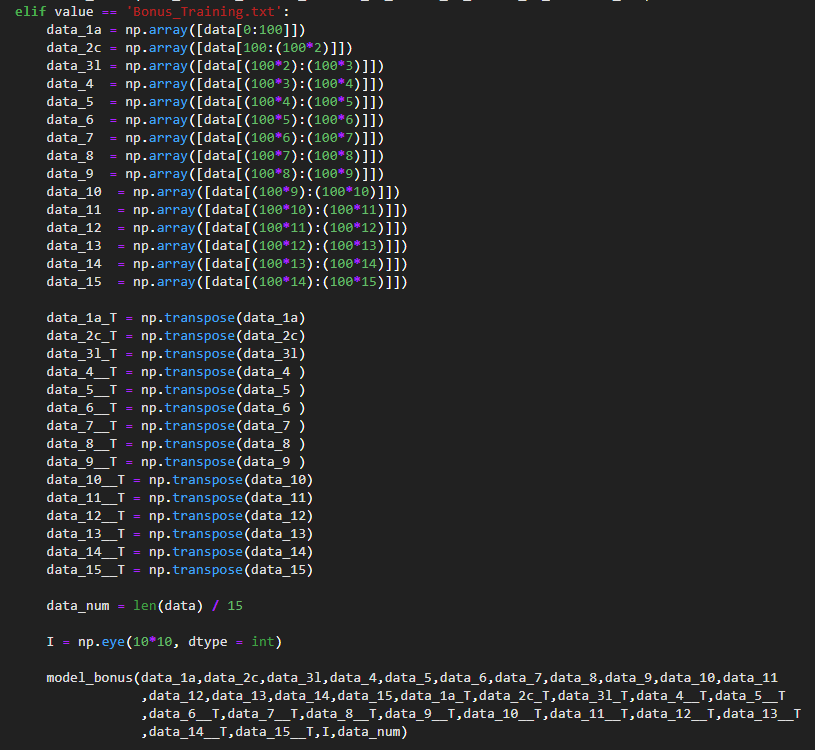
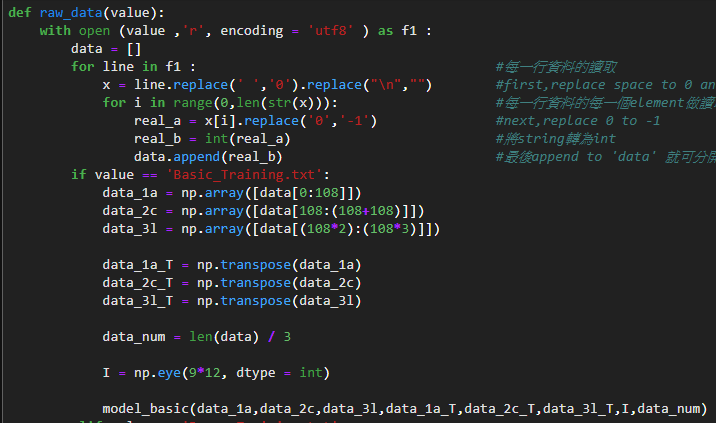
再來，我們定義我們訓練的函數，也就是V(鍵結值)減去b(閥值)，若>0則我們的資料會被更新成新的值。



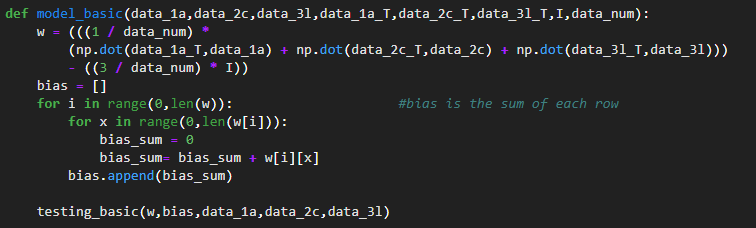
此處我們將圖形介面選檔案的選項用滑鼠點擊資料選取，並將所選取的資料集叫做value。



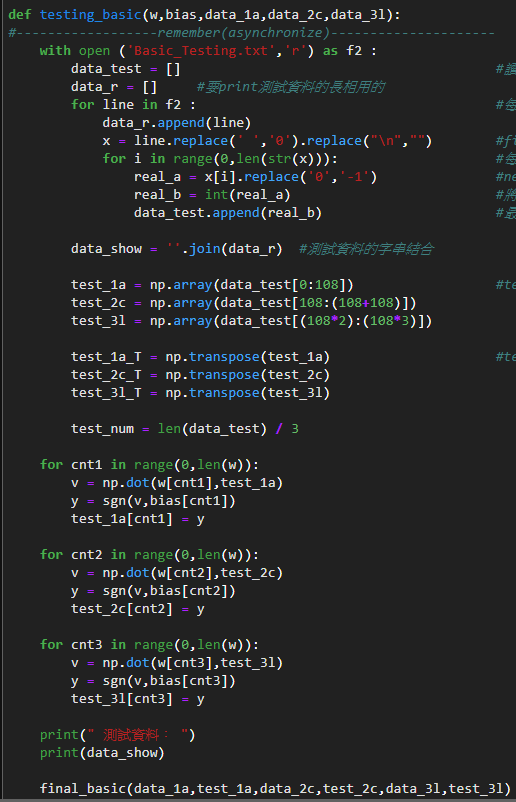
我們將所選取的資料value進入這個函數，我們在一行一行讀取時，先將空格處replace成0，將換行處去除掉。接著，我們在往下進入字串環節，我們再將字串中的0轉換成整數的-1這樣才能完整的做轉換，最後append進我們data的list中。並且判斷是Basic\_Training還是Bonus\_training的data，然後將我們的資料做每個圖型矩陣的切割(將每108或100時切割)。



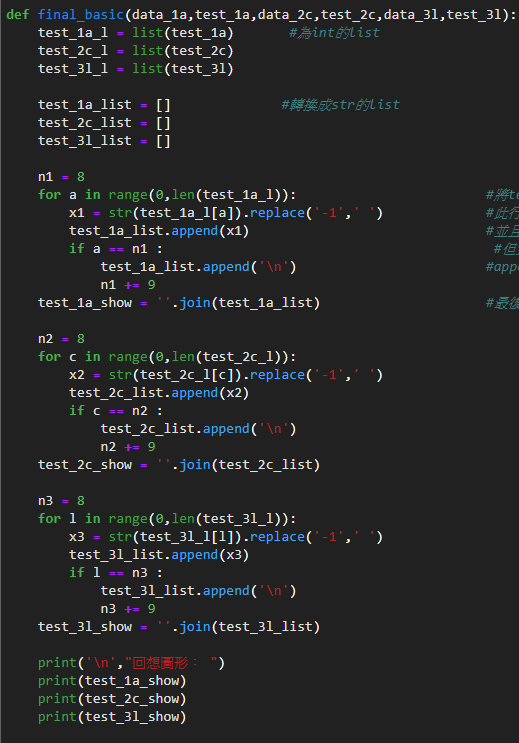
我們在這邊訓練我們Basic\_Training的模型，也就是將W(鍵結值)依照課本上的公式處理，並且將bias設定為sum of each row。



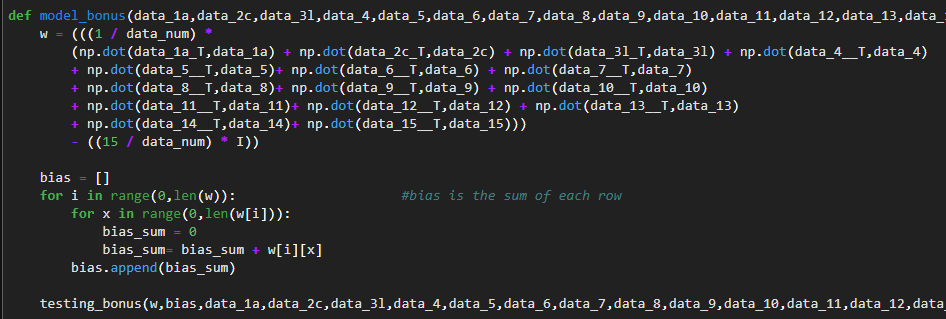
訓練完成後叫出們的測試資料，按照和訓練資料的模式整理我們的測試資料，並且在整理完後直接用for loop對每個一切割好圖型的測試資料做測試，v就是(each row的鍵結值x測試資料)，接著帶入我們前面所定義的函數sgn，最後將函數所算出來的值更新回我們的測試資料中。此處我採「非同步」處理。



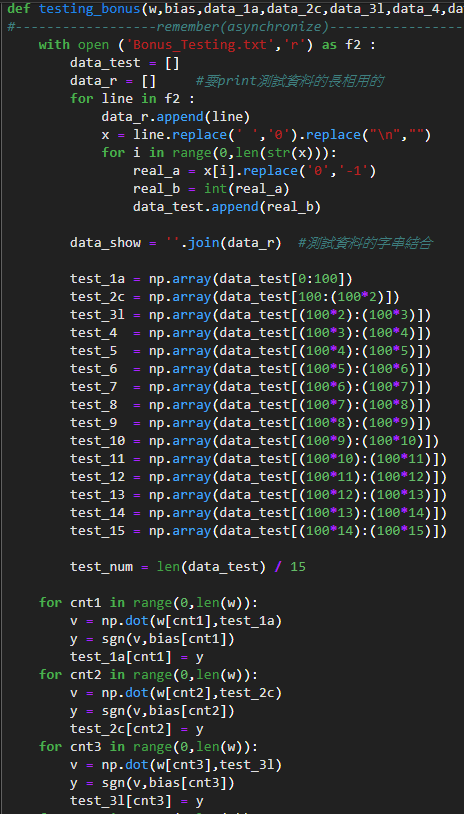
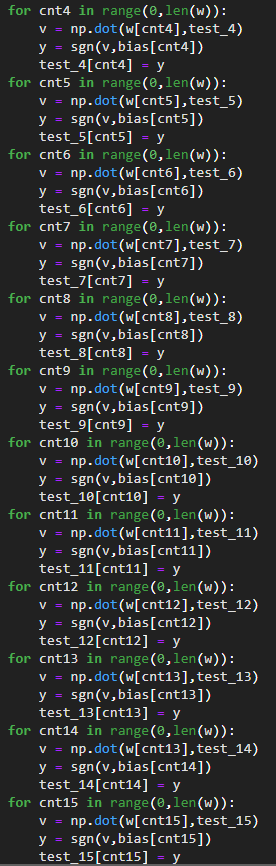
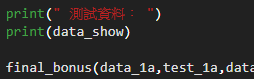
最後，將測試好的資料準備轉回圖型的樣子並且輸出，我先將測試的資料轉回list的形式，並且分別利用for looop將我們的資料轉回string並且將-1 replace成空白，再每9個字元就加入換行，這樣才能會是9\*12矩陣，最後利用.join來將每個圖型的每個字元合在一起。



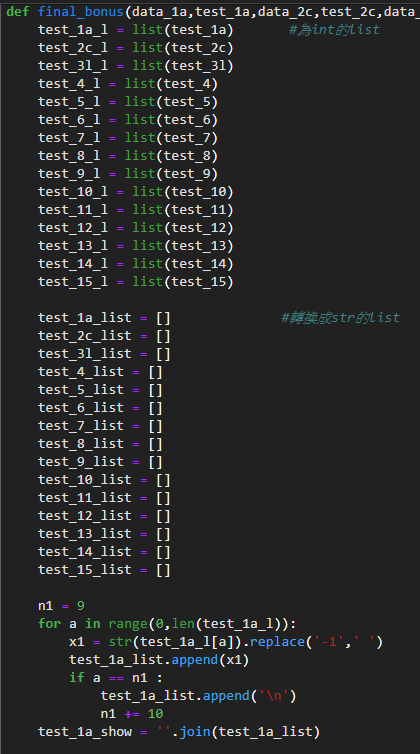
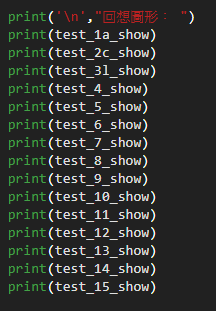
我們在後面做了Bonus資料的處理，此處則是Bonus\_Training的模型訓練，和Basic的資料處理一樣，bias也是利用each row的相加。



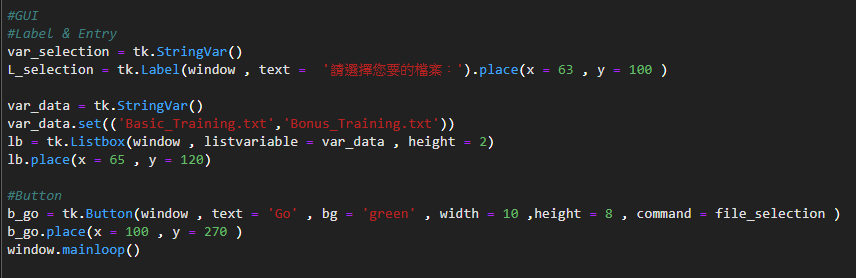
我們將Bonus的測試資料叫入，並且用同樣的方式做處理資料，切割成100\*100的15個圖型資料，最後的for loop一樣是讓每個值做非同步的更新，每個圖型的更新都是一個for loop，分別做運算。



此處是Bonus的最後圖型資料整理，我們將所做好運算的資料轉回list，並且replace -1 to space，且在每10個字元就加入換行，這樣才能是10\*10的矩陣圖型呈現，最後利用.join將每個字元串在一起最後輸出。



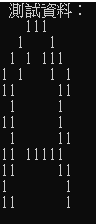
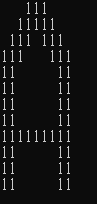
程式的最後，就是我們GUI介面。



三、實驗結果分析及討論:

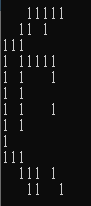
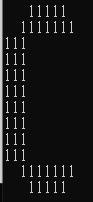
1.Basic\_Testing:

我們在第一個圖型中，能正確的回想圖型，原因可能是因為我們資料的缺失以及雜訊不夠多，因而我們能夠在不缺失大部分資料的情況下正確的回想。

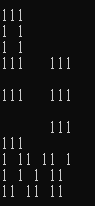
測試圖型 回想圖型

此組測試圖型也是，我們因為對於圖型缺失仍能夠看出大體能夠想出的資訊，因為我們的模型也能正確地去做回想無誤。

測試圖型 回想圖型

而此組資料我們利用肉眼時較難以分辨是何種字母，所擁有的雜訊較多，但經過模型訓練之後我們可以訓練出是字母L的圖案，並且也能正確地做回想。

測試圖型 回想圖型

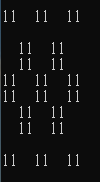
2.Bonus\_Testing:

此組測試資料因為所缺失以及雜訊的資訊不多，因此我們能做正確的回想。

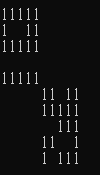
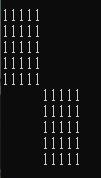
測試圖型 回想圖型

此組資料也是，我們缺失的資料只有一點，仍能清楚的看出是哪組圖型，因此我們的模型也能訓練出正確的回想圖型。

測試圖型 回想圖型

此組資料雜訊不多，所缺失的訊息也不多，大抵上仍能看出是何種圖型，因此訓練模型也能清楚地回想圖型。

測試圖型 回想圖型

此組資料中，測試圖型已經有較多的缺失，但整體能表示出的圖型仍然明顯，因此模型的訓練不太有太大的問題，所以能正確的回想出圖型。

測試圖型 回想圖型

此組圖型也是，我們經過訓練後，大多都能正確的回想出圖型。

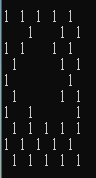
測試圖型 回想圖型

這組測試資料就未能正確的回想，雖然圖型下半部的回想是正確的，但上半部仍出現其他圖型的資料，我猜想因為是因為模型聯想錯誤，因為我們訓練的過程中，有類似的圖型(在下一組資料)，因此導致回想錯誤，出現下半部分正確，而上半部錯誤的情況。

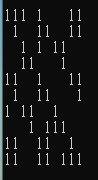
測試圖型 回想圖型

此組資料也未能正確回想，雖然能大致看原來的圖型長甚麼樣子，但仍然在中上部分，有一小個資料的回想錯誤，可能是因為我們bias的設定原因，因此我們在此處的更新無法正確。

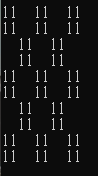
測試圖型 回想圖型

此組測試資料中，雖然我們有訓練資料做訓練模型的依據，仍能大抵上回想出大部分圖型，但仍然會在某些部分有微小的缺失，原因有可能是因為本身這個圖型過於複雜，和我訓練模型中其他的資料過於相似，因此在某些部分才因而回想錯誤。

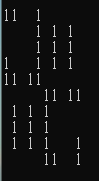
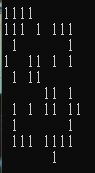
測試圖型 回想圖型

此組資料就因為明顯能辨別且缺失不多，因此我們能正確的回想出圖型。

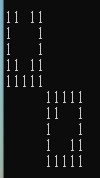
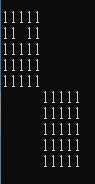
測試圖型 回想圖型

此組資料中，並未能回想出正確的圖型，連大致的樣子都沒能呈現，反而往另外一種圖型的樣子去做聯想，可能需要調變一下我的bias值，試看看若改成其他的值是否能正確回想，或者使用同步更新來做回想。

測試圖型 回想圖型

和前面的資料一樣，我們雖然圖型的大致可以回想，但是仍然會有小部分資料未能回想成功，就像此回想圖型一樣，有一小個資料點未被回想。

測試圖型 回想圖型

在此資料中，我們的測試資料因為和其他的訓練資料會過於相近，因此我們回想的圖型才會回想成其他的圖型。

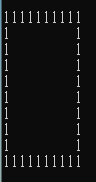
測試圖型 回想圖型

在此筆測試資料中，回想的圖型很成功，可能是因為雜訊較少，較能清楚的判別出回想的圖型應該試何者。

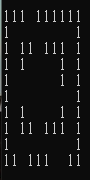
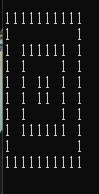
測試圖型 回想圖型

此組測試資料中，我們應該回想為中間還有4個資料點的圖型，但是直接形成了未曾見過的圖型，要改善的方法應該是改變一下我們的bias或著我們可以利用同步更新的方法試看看。

測試圖型 回想圖型

此組資料較為明顯能讓我們清楚知道該回想的圖型應該為何，而剛剛的測試資料會較為模糊，因此我們這組測試資料的回想圖型在回想後也成功。

測試圖型 回想圖型

四、心得

這次的作業一開始在資料的整理上花了很大的功夫，想著該怎麼樣才能像課本上那樣的用-1,1做運算，後來在字符字串的地方卡了很久才搞出來，還有也在思考要怎麼能在資料中自動地去切割9\*12的矩陣，才不用像是已經知道資料的維度然後才去做切割，讓自己有點像偷工減料，但也是思考很久以後沒有想到，所以還是直接用[:]去切割我們的矩陣來做運算，這次對老師上課講的一句話很有心得，就是「要先知道訓練資料是甚麼」，就像這次Hopfield的資料，因為是字串的關係，所以要轉換成整數，但之中又有很多的細節要先處理，所以我覺得在做網路架構的時候我應該要先去想好我的訓練資料的整理是最重要的，最後，也在這幾次的作業中學習到了很多有關類神經網路的實作，讓自己從一個甚麼都不懂的大學生有一點點的進步~

五、加分項目:

有做Bonus\_training、Bonus\_testing。

Demo影片。