

Neural Network Homework 2 Report

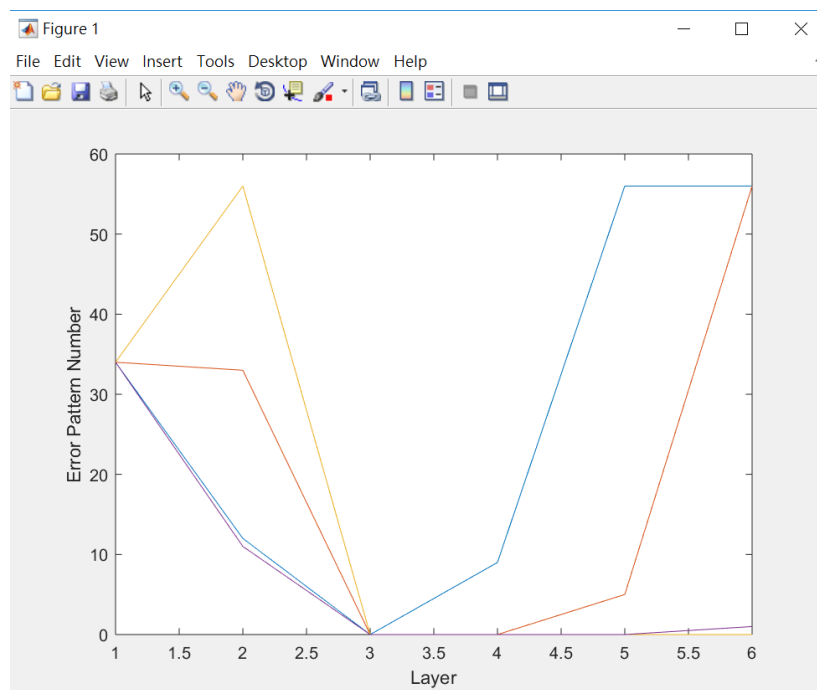
R06922063 邱政凱

這次的作業要求我們實作的是老師的 SOM Perceptron。這個演算法的重點來自於 SIR(Separable Internal Representation)的概念，藉由迭代減少一個描述相同 CLASS 的 pattern 中最長距離跟來自不同 class 的 pattern 中最短距離的 Energy Function，來對於 Network 中每一層 Layer 進行個別的優化，使其對於 input pattern 產出的 internal representation 能盡可能的把相同 class label 的 representation 聚集，並盡可能分開有不同 class label 的 representation。

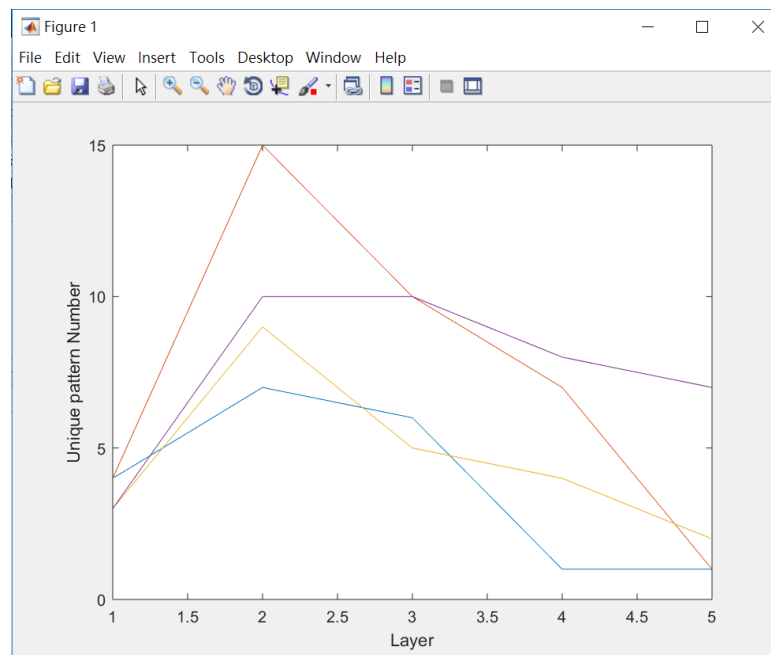
在實作方面我使用 Native C++，只有使用到 Standard Library 裡面的函式。基本上只要用 GNU g++ 直接編譯就可以(記得 specify version 為 C++11，i.e. `g++ -o hw2 r06922063_hw2 -m11`)。

執行方式基本上就是把名為"hw2pt.dat"和"hw2class.dat"兩個檔案放在跟執行檔同一個路徑下執行，會產生"output.txt"裡面會在 SOM perceptron 每增加新的一層時印出完成訓練的前一層的權重以及所有 pattern 在經過前一層之後的 Internal Representation，並且會輸出該層的 Error(我定義 Error 的方式為: 有多少屬於不同 class label 的 input pattern 在經過該層後出現了一樣的 Internal Representation)，同時，在每一個 Epoch 時都會輸出當前距離最短的相異 class pattern 的距離以及當前距離最長的相同 class 的 pattern 的距離。

1.



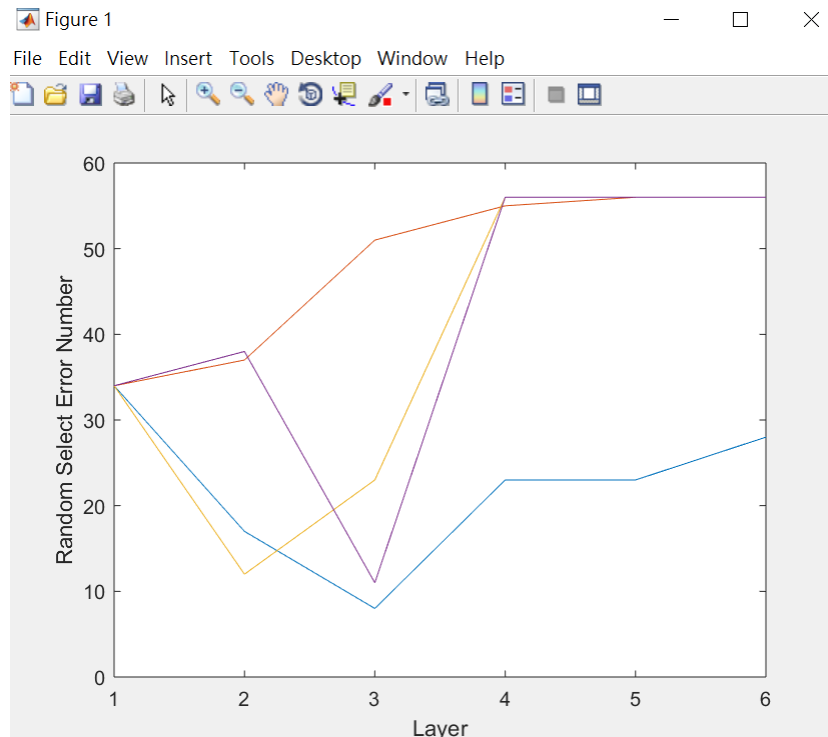
上圖是我以 Epoch=5000，attraction Rate = 0.001，repelling rate = 0.1 並以-1~1 的 Uniform distribution 去初始化網路的參數去進行 training 的結果。不同顏色的線代表不同次的實驗。可以發現其實大部分的狀況下，Error 從初始的 34 開始，都會穩定的在經過兩層 Layer 之後達到 0(也就是已經可以完美進行 classification)。然而同時也都可以發現，其實越多層的 layer 到後來其實 Error 會不減反升，這樣的現象我可以用另一張圖表畫出每一層中的 Unique Internal Representation Number 來解釋：



上圖表示在每一層時，對於所有的 pattern 在該層的 Unique 的 IR 數量，比對第一張圖不難發現，第一張除了紫色以外其他都在後面的 Layer error 急遽上升，而 Unique IR 下降，但紫色因為 Unique IR 並沒有下降太多，使的 Error 不至於急遽上升。所以我們可以對於 SOM Perceptron 的 Layer 增加時 Error 變大的現象解釋為因為幾乎所有的 PATTERN 都被擠到少少的 IR 中，導致 Class label 會有混淆的現象。

(第二題在下頁)

2.



接著我們照著第二題的要求試著在 SOM Perceptron 的每一層訓練時都把前 1/10 的 Epoch 用來當作 attraction 或 repelling 參考的 pattern 用隨機挑選的方式來進行。可以發現在某些初始值的狀況下還是有可能可以達成 SOM Perceptron 的目標，但是相比於全部都是挑選最近或最遠距離的 pattern 來當作訓練的標準，可以發現 Error 下降幅度變小，甚至有幾次 Error 從一開始就只有上升沒有下降。可以推測為就算只有 1/10 的次數是用 random selection，但是還是實際上打亂了原本的 pattern 的排列，也許把很多不同 class 的 pattern 都混在一起了，或是把相同 class 的 pattern 距離越推越開，這也不是我們樂見的。

Conclusion:

SOM Perceptron，或者說整個 SIR 的概念，提供了一個有別於傳統訓練 NN 用的 BP 途徑，直接從了解內部資料特性的角度切入，用可以被解釋數學特性的方式訓練 NN，卻也著實地可以處理一定程度的分類和辨識問題，提供了我不不少的啟發。