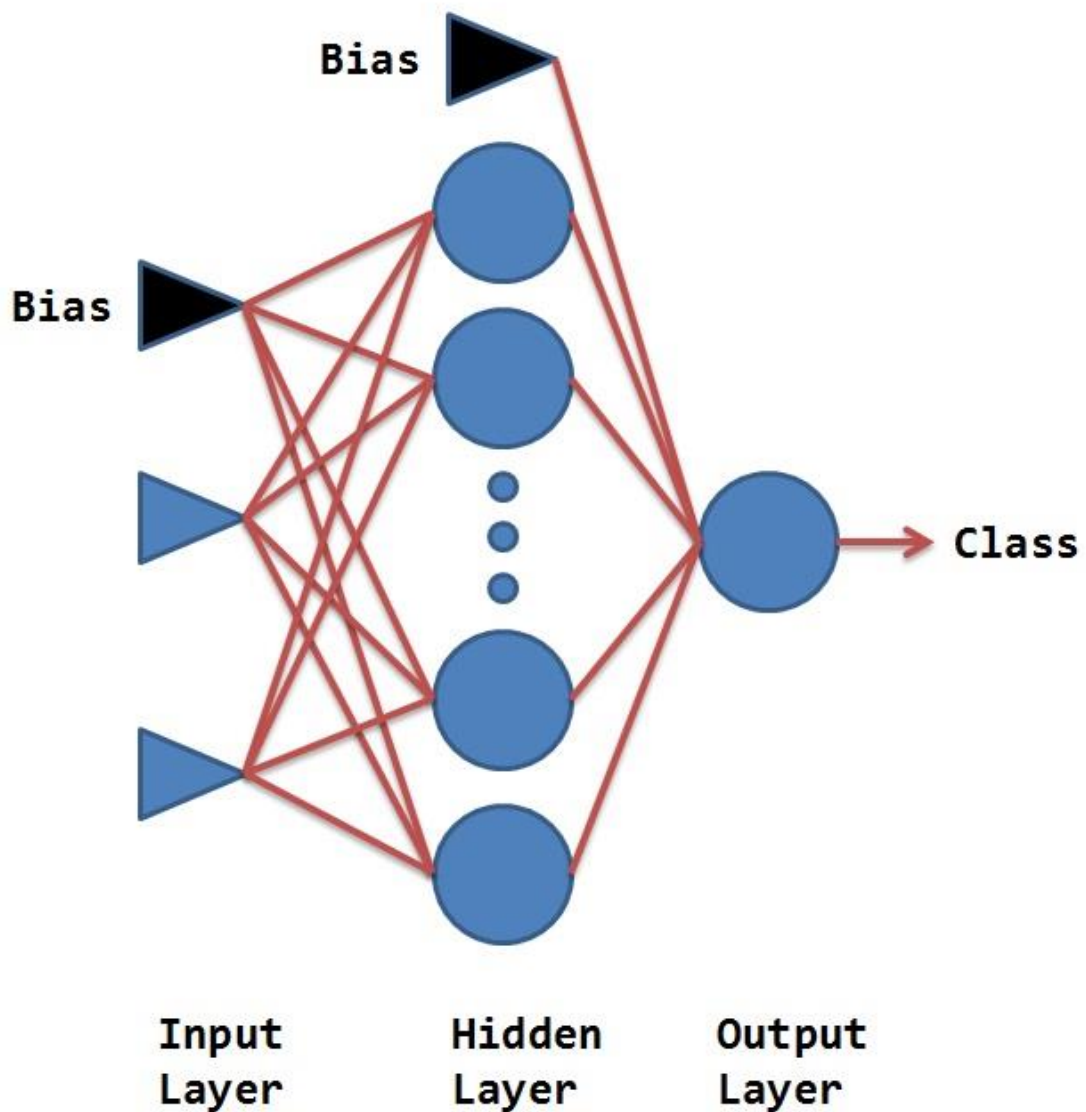


## Computational Intelligence and Applications HW01

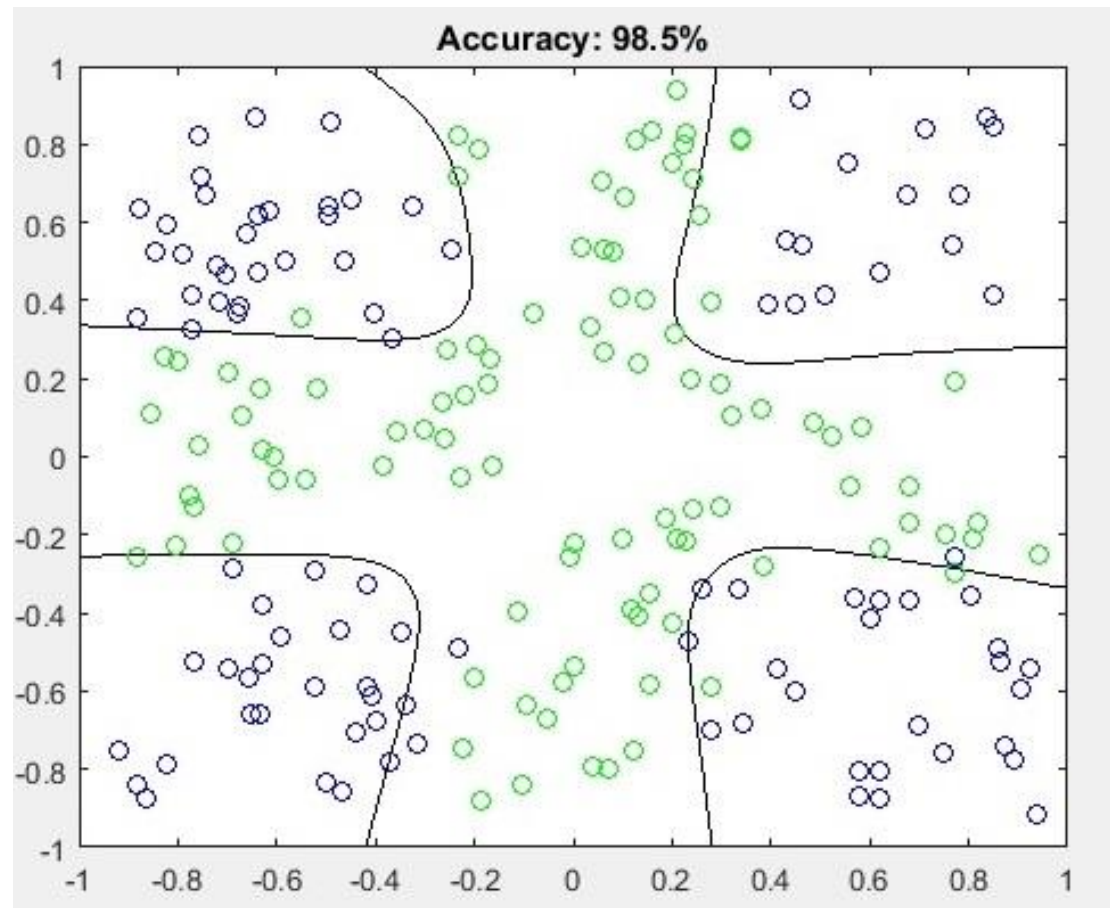
- Methods

- MLP架構

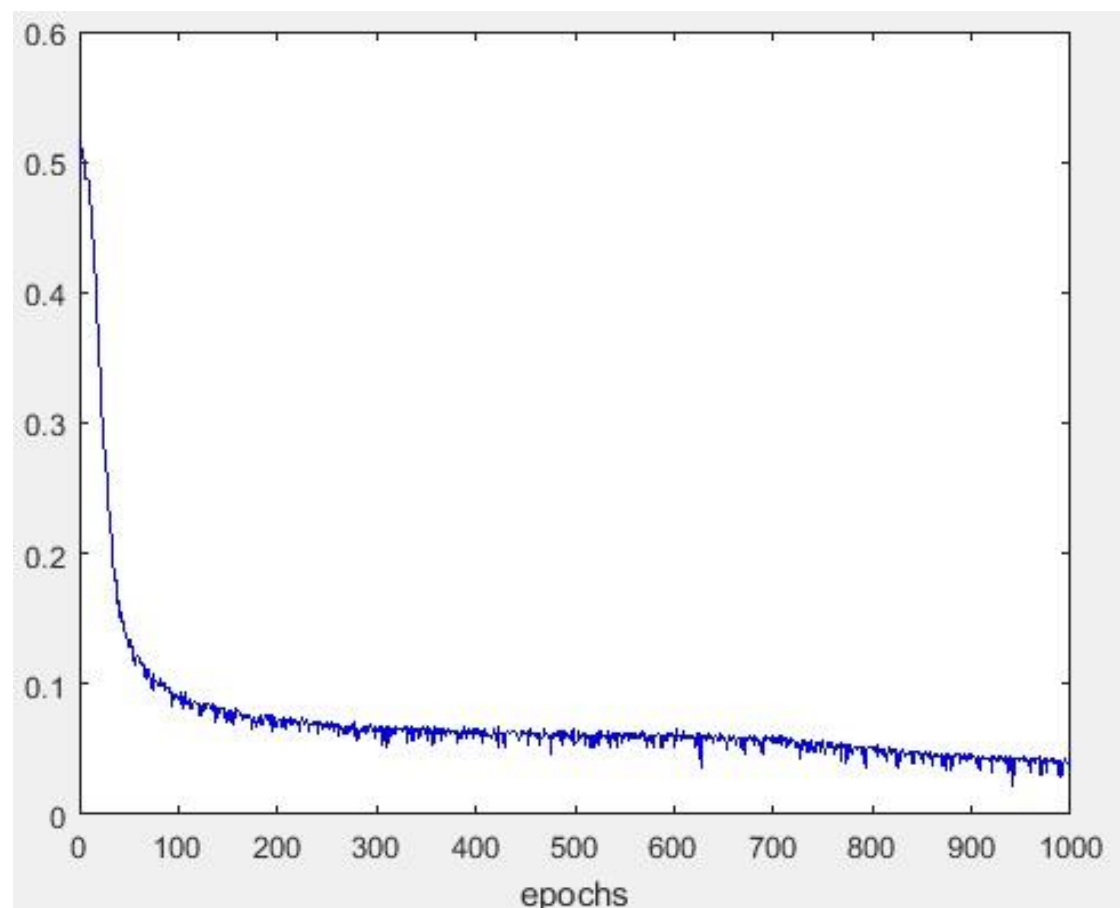
- ◆ 讀取附檔 cross200.txt作為訓練資料。
    - ◆ 輸入有兩個 feature加上一個 bias。
    - ◆ 擁有一層 hidden layer並含有10個 hidden neurons。
    - ◆ output layer含有1個 output neuron。
    - ◆ 各 neurons的 weight經過 random來初始化。
    - ◆ Activation function  $f(v) = \tanh(av)$ 。



- 前饋階段
  - ◆ 每一層的輸入與該層 weights 做矩陣運算後，經過 Activation Function 來計算出該層輸出，並前饋到下一層，。
- 倒傳遞階段
  - ◆ 利用下一層的 error 跟 weights 與學習率運算後得出該層的修正值。
- Experiments & results.
  - 訓練1000輪，每一輪都將200個訓練資料打散後選出195來做為輸入。



- 上圖中，線將點分類為兩群，實驗中準確率在97~98.5%之間徘徊，而 hidden neurons 的數量在三以下時會使得分類效果不佳，大於等於四之後則呈現差不多的效果。



- 隨著 epochs(訓練輪數)的增長，整體的誤差值不斷下降，小範圍內的波動應該是來自於每一次選取的訓練資料不同所帶來的小量差異性。

#### ● Analysis

- Hidden neurons 的數量過少時，會無法很好的處理過於複雜的分類，但無限制的提升 Hidden neurons 的數量，卻不一定會帶來準確率上的提升，只會平白增加運算時間，要找出適當的 Hidden neurons 數量，需要花上一些時間來調整。
- 初始化 weights 時，由於使用隨機賦值得方式，有時候會導致準確率較低，不過在這次的訓練資料中，準確率的高低值並不至於相差太多，應該是因為訓練資料中的分類難度並不高。
- 準確率卡在最高 98.5%，應該是由於我一次的訓練資料都會經過隨機篩選後再輸入，導致訓練結果並不會完全貼合訓練資料。