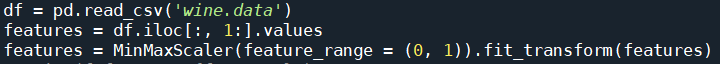
實作SOM(kohonen network)，分類wine data。

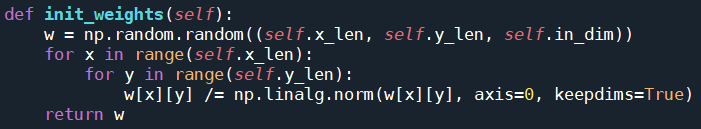
有關訓練資料提取：

需將特徵值正規化至[0, 1]區間，訓練過程中比較不會有問題。

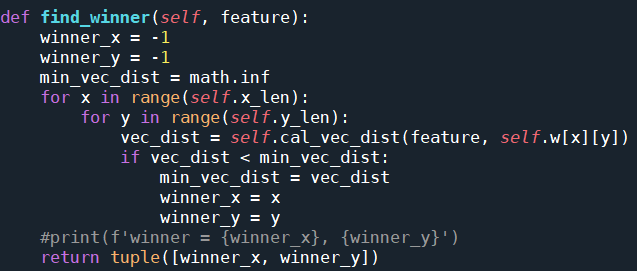


有關實作一個SOM類別：

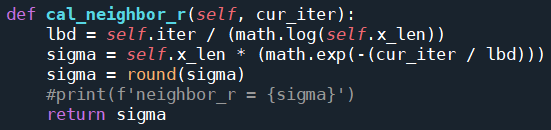
1. 初始化：
   1. 對各輸出神經元，隨機給值初始化權重向量(各長度為輸入特徵維度)。  
      以wine.data為例，有13個特徵，因此各輸出神經元權重長度為13。



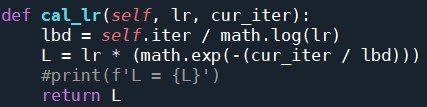
1. 訓練：  
   設定初始learning rate、iteration，在每一iteration中，針對每筆訓練資料：
   1. 找出winner(權重向量與輸入向量距離最近的輸出神經元)。



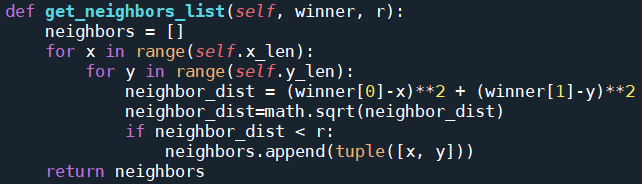
* 1. 根據目前iteration，算出鄰居半徑。



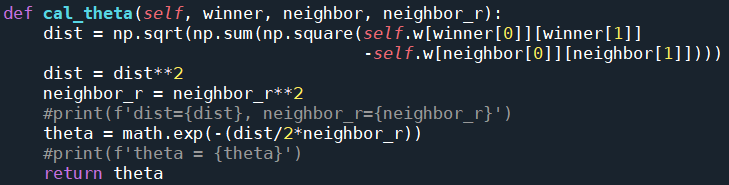
* 1. 根據目前iteration，算出當前learning rate。



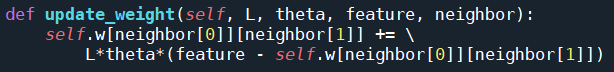
* 1. 以winner為中心，利用(b)算出的半徑，得到所有鄰居。



* 1. 針對所有鄰居(要更新權重的輸出神經元)：
     1. 算出其theta。

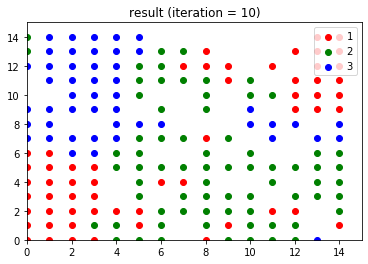


* + 1. 利用(c)的L和(i)的theta，更新該輸出神經元的權重。

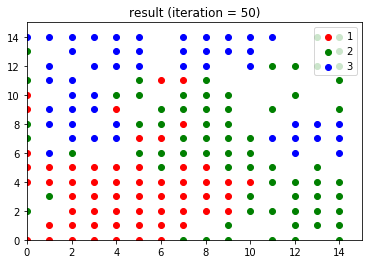


有關結果：

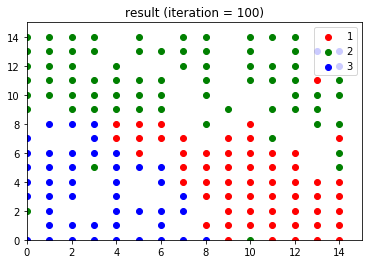
1. 訓練iteration = 10



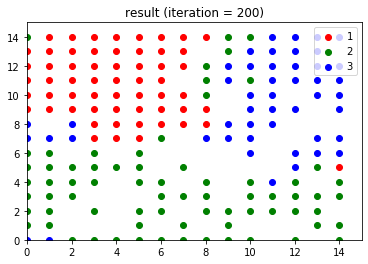
1. 訓練iteration = 50



1. 訓練iteration = 100



1. 訓練iteration = 200



小結：

一開始沒有對輸入特徵做標準化，theta或L經exp算出來都是0，導致權重更新等於沒更新，後來對輸入特徵做標準化才解決此問題。

最後在畫圖時，沒有對輸出神經元監控，導致很多筆資料對應的輸出神經元是相同的，圖畫出來只有很少的點，幾乎都重疊，後來有對輸出神經元做監控，已歸屬於某筆輸出資料的神經元不再被考慮，才解決此問題。

訓練的時間有點久，應為輸入特徵向量長度和輸出神經元數量所致。

訓練結果隨著iteration次數越多而越準確，大約iteration = 100以後都差不多。