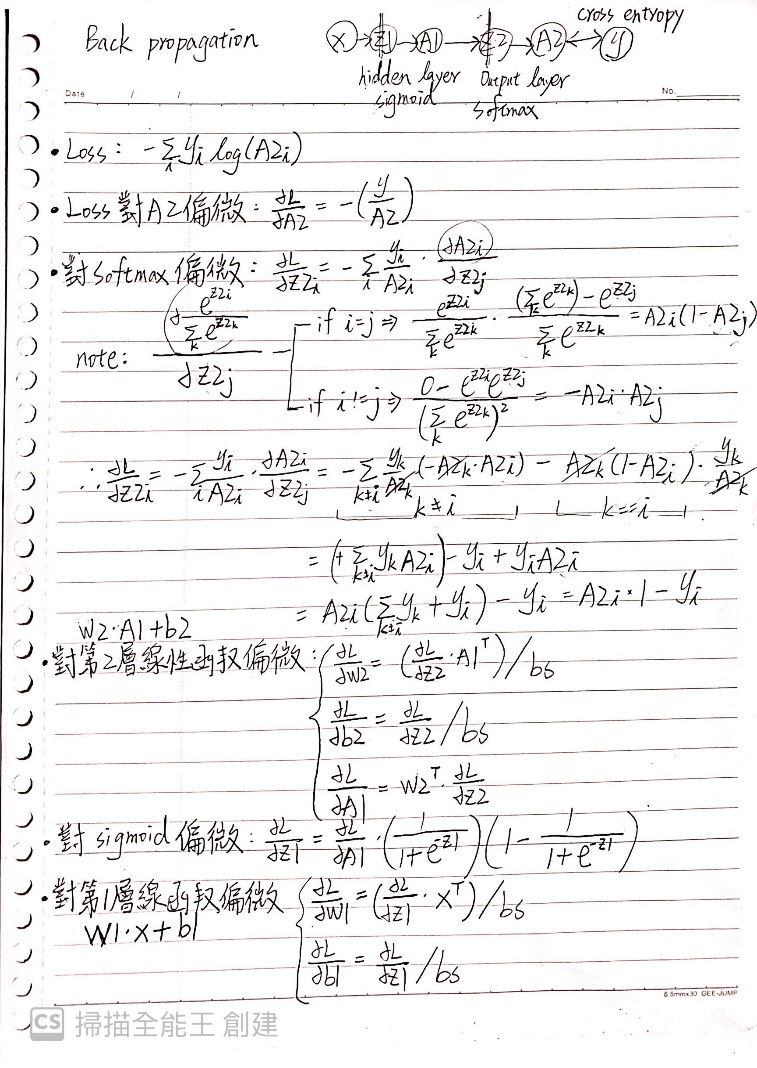
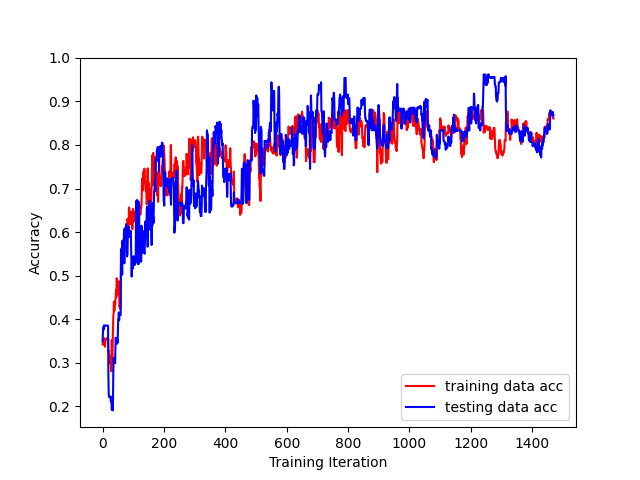
* How to run the code

有兩個py檔，分別是two\_layer\_NN.py與three\_layer\_NN.py，執行方式為在終端機輸入python3 two\_layer\_NN.py與python3 three\_layer\_NN.py即可。

* two\_layer\_NN.py程式碼說明
  + 此為two-layer neural network，分別有一層input layer，一層hidden layer(activation function為sigmoid)與一層output layer(activation function為softmax)，input layer為2個維度(用PCA降維至2維)與一個bias，hidden layer為35個neuron與一個bias，output layer為3個neuron。Learning rate設為0.02。因為題目要求SGD，每次更新參數的方式是從1470個training data中隨機不重複挑一個，先做forward passing得到output(3種水果的機率，總共三個數字，總和為1)，再用此output與ground true計算多類別cross entropy，並以cross entropy當作loss function做偏微分與chain rule得到各個參數的梯度(Backpropagation)並更新參數。參數總共更新1470次(讓model看過每一個training data)。weight於更新前的初始值為mean=0 & variance=1的gaussian隨機數，而bia於更新前的初始值為0。此model在testing data上的accuracy=86.75%。
  + 下圖為二層Backpropagation的公式推導

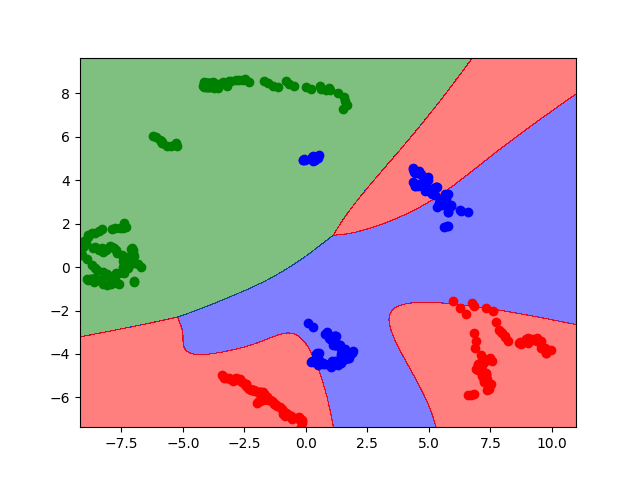


* + 程式執行時會產出兩張圖，第一張圖如下



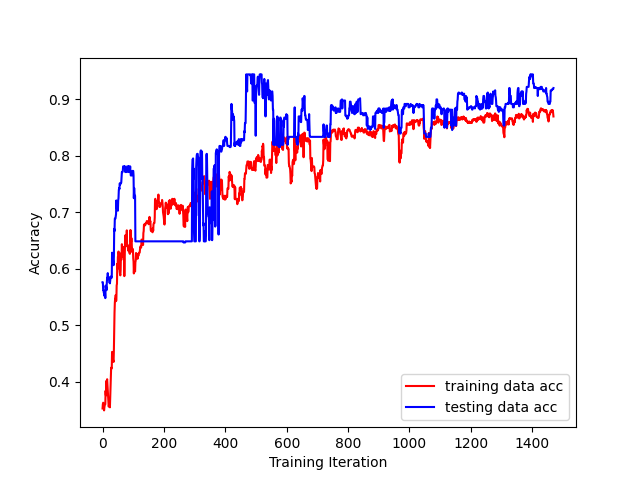
我記錄下參數每次更新時對training dataset與testing dataset的accuracy變化，可發現隨著model更新次數越多，看過的training data也越多，training dataset與testing dataset的accuracy都有上升的趨勢。

* + 第二張圖如下(decision regions)



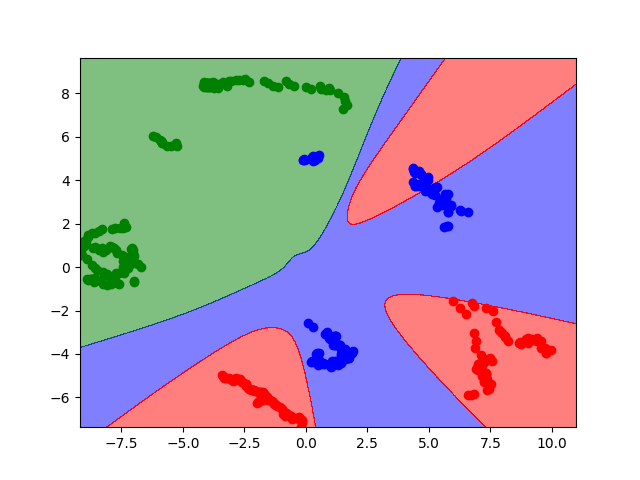
圖中的橫軸與縱軸分別是testing dataset做完PCA後得到了兩個維度，色塊區域是各種2維feature組合下餵給model的predict結果(3種顏色代表三種水果)，而綠藍紅點是testing dataset ground true，有些位置像是藍點點在紅色色塊就代表model預測錯誤，另外像是綠點點在綠色色塊代表model預測正確。

* three\_layer\_NN.py
  + 此為three-layer neural network，與two-layer neural network的設置上幾乎相同，差別在多一層hidden layer，而該層hidden layer的activation function是sigmoid，第一層hidden layer的neuron數為70，第二層hidden layer的neuron數為50，並且learning rate設為0.005。此model在testing data上的accuracy= 91.97%。
* 程式執行時會產出兩張圖，第一張圖如下



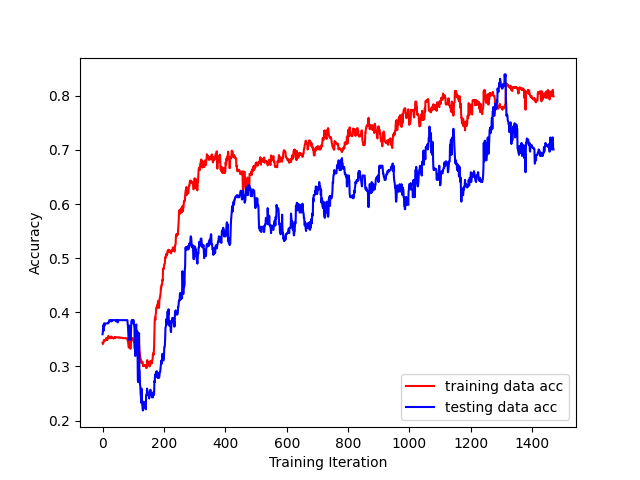
Training data acc比testing data acc還低，這種情況是不常見的，訓練模型通常是相反的狀況，並且兩者acc落差懸殊而有overfitting的情形發生。個人推測testing data之acc較高可能的原因是testing data的辨識難度較training data簡單很多，並且兩組dataset有相似的distribution，故model除了能在training data上學到對testing data有用的概念，且因為testing data更簡單，所以能在testing data上有更好的表現。

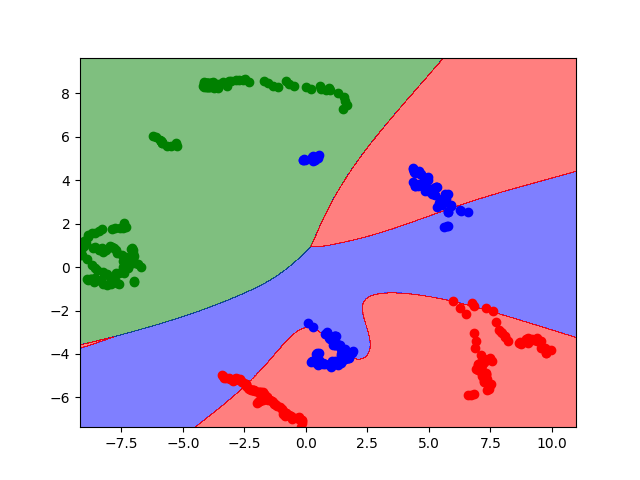
* + 第二張圖如下(decision regions)



* two-layer neural network & three-layer neural network比較
  + three-layer neural network是在two-layer neural network的基礎上多加一層hidden layer，故參數量較多，所以model size較大，可以處理較複雜的task，並且在testing data上的表現有機會更好，而上文提到兩個model的testing data acc，的確驗證了這件事。
* Different settings on two-layer neural network

我將原本two\_layer\_NN.py內的learning\_rate從0.02設成0.005，model在testing data上的acc變成70.08%，兩張結果圖如下:

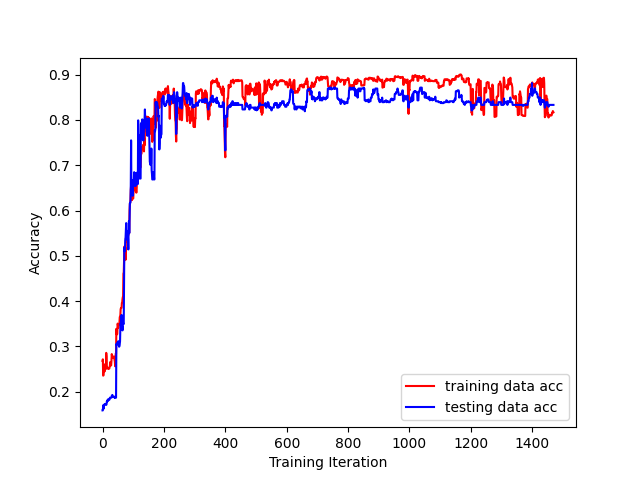


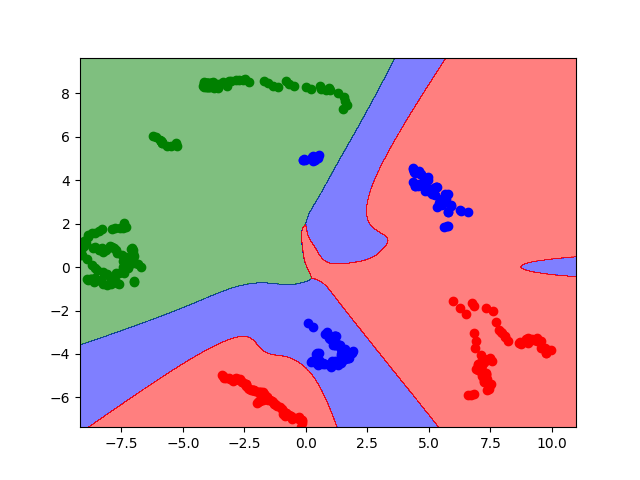


折線圖的尾端可看到acc都還在攀升的階段，代表model還沒達到收斂，意味著learning\_rate設成0.005太小了，應該調大learning\_rate或是增加參數更新次數。

* Different settings on three-layer neural network

我將原本three\_layer\_NN.py內的兩個hidden layers之neuron數都調大至100，參數量變得更多。照理說model可以有更好的performance，但是model在testing data上的acc變成83.33%，結果圖如下:





Model size變大可以處理更複雜的task或是有更好的表現，但是也不是絕對，要看task難易度與training data的量來決定，若在task過於簡單或是training data量不夠多的情況下提升model size會造成反效果。上面的結果即可證明此事。