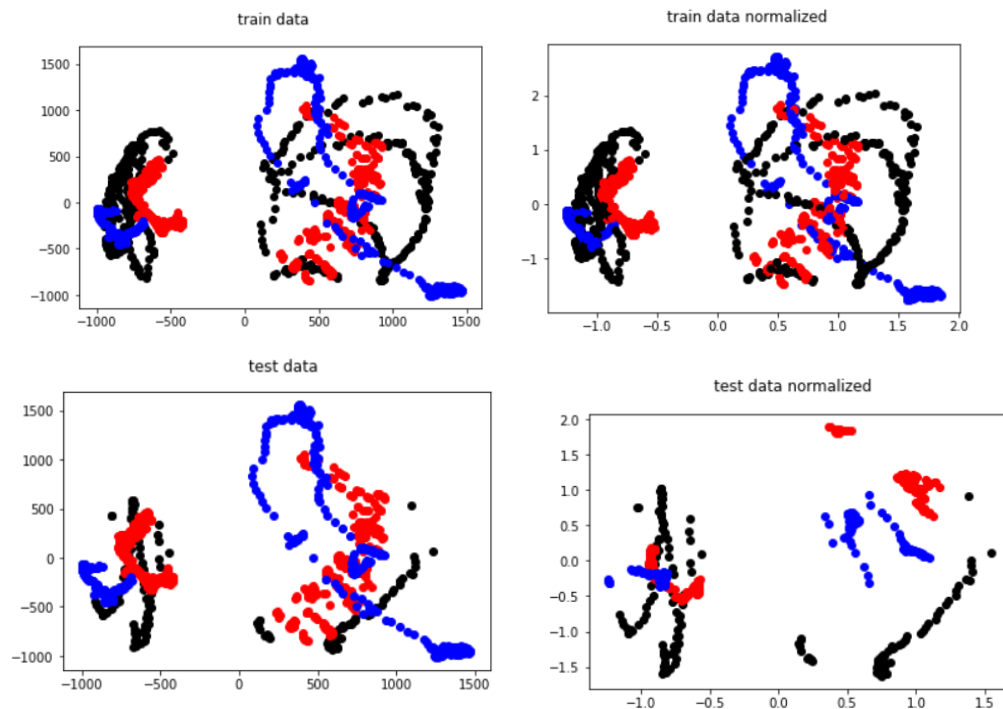


設計方法

一開始先將圖片讀進來後調整形狀成一維向量，之後利用 PCA 抽取特徵為兩個主要特徵，並加入一個值為 1 的向量做 bias。之後再將所有資料進行標準化，然後利用 SGD 算法進行訓練，初始化參數使用 xavier，激活函數皆使用 sigmoid，最後輸出函數使用 softmax。

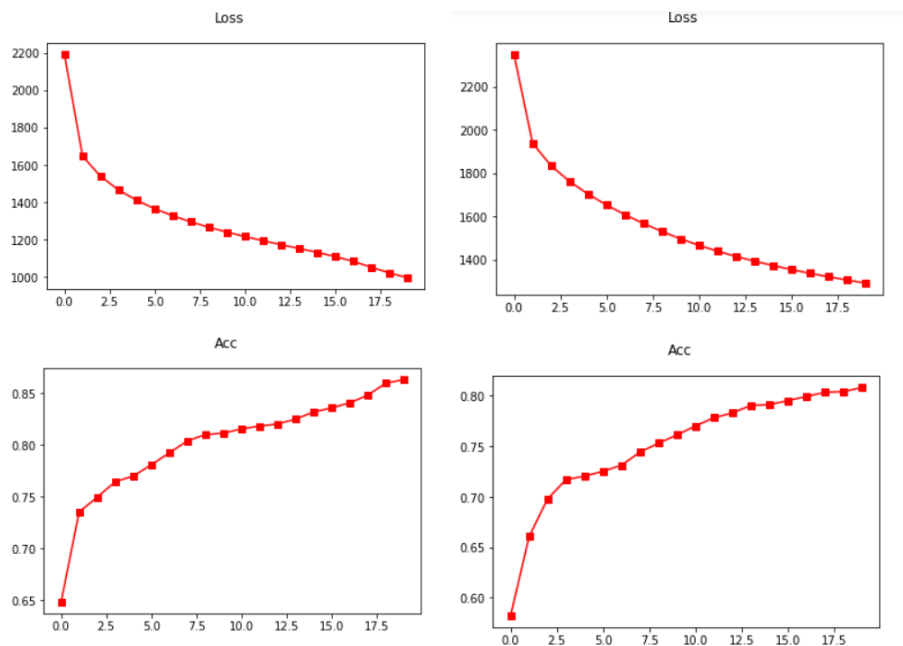
數據討論



上圖是將抽取過後的特徵進行繪圖的結果，同時有進行標準化和未標準化的結果，測試資料的標準化使用的平均值和標準差是從訓練資料取得的。可以觀測到三類圖片大致可以分為兩群，而且重複性極高，但進行標準化後可發現測試資料的重複性變低，也可以從模型訓練結果觀測到成功率提升。

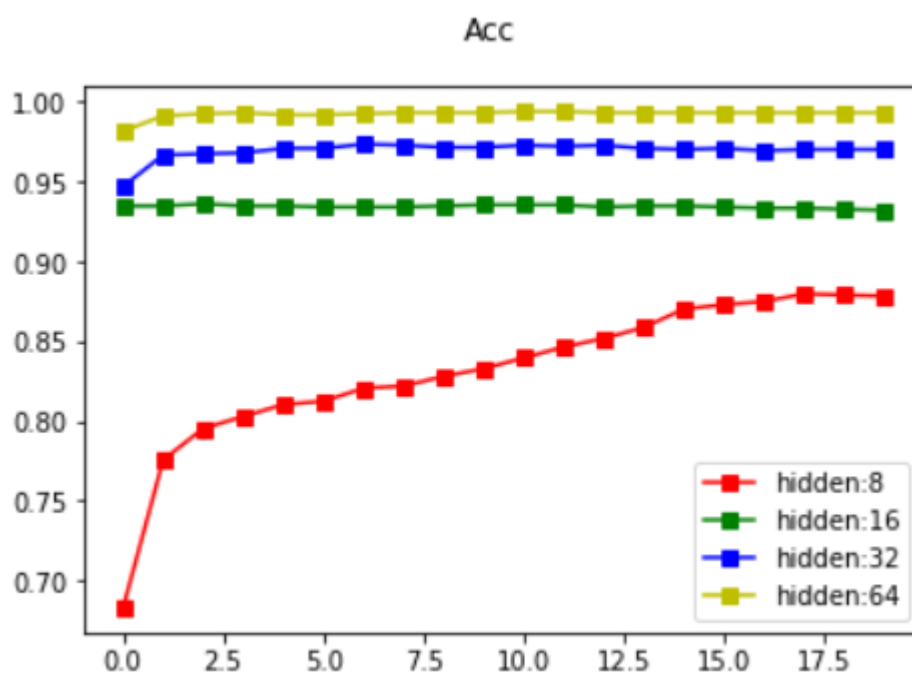
Two-layer vs three-layer

下面左圖是 three-layer 的訓練資料預測結果，右圖是 two-layer 的訓練資料預測結果，都是訓練 epoch:10、batch_size:1、learning_rate=0.01，而最終 three-layer 在測試資料的準確度為 60%，而 two-layer 則是 56%，而如果將 epoch 調高到 50 在 two-layer 可以得到 73% 的準確度。從中可發現 three-layer 在相同的狀況訓練結果比 two-layer 好，但花費的時間卻超過一倍，除此之外一開始的參數初始化也是非常重要的，常常會有一樣的 model 跑兩次在測試結果的正確率差超過 20%。

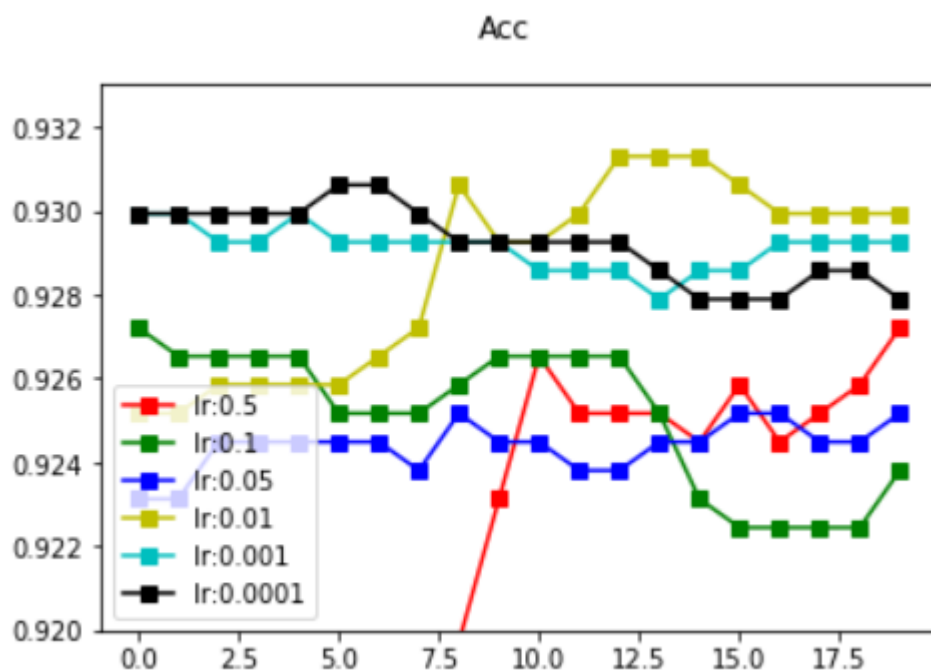


各項參數調整

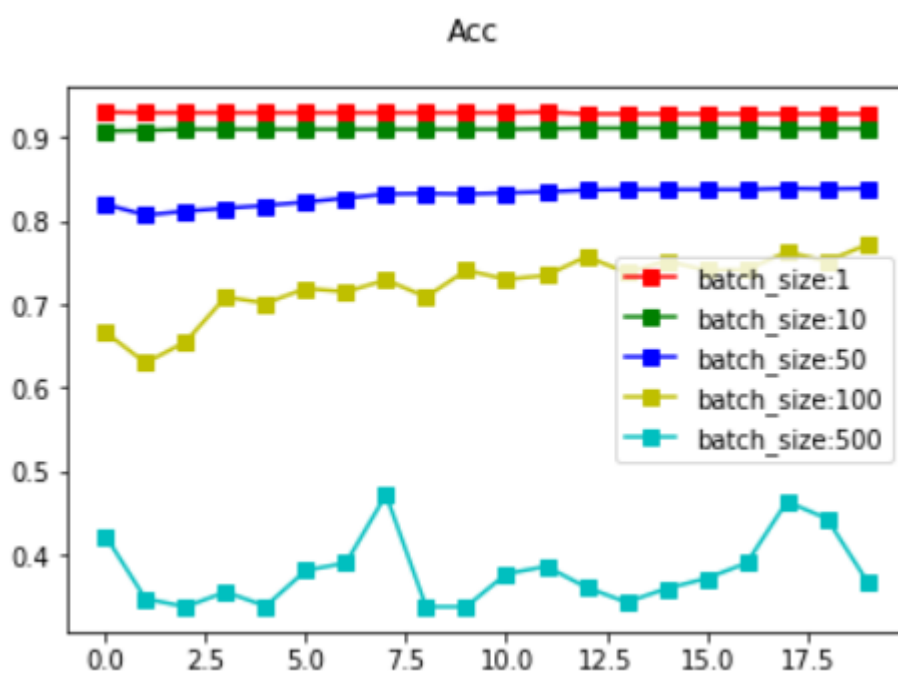
接下來的測試已 two-layer 為基準，各項參數分別為 hidden layer:32、batch_size:1、learning rate=0.1。



上圖為 two-layer 模型在不同 hidden layer 數量下的訓練結果，而最終測試資料 8、16、32、64 層分別是 45.5%、59.6%、60.2%、61.6%，可以發現層數越深結果越好的趨勢。



上圖為模型在各種 learning rate 的訓練結果，從測試資料的結果來看 0.001 和 0.0001 都可以到達 62.4 而大於 0.01 後雖然降低到 61 或 60 但還是差不多的，因此判斷 learning rate 大概設定為 0.001 是比較合適的。



上圖為模型在各種 batch_size 的訓練結果，可以發現 batch_size 越大訓練成果會越糟糕，而在測試資料的結果來看 batch_size 在 1 和 10 都可以獲得 63% 的準確度，但當變成 500 時卻下降到 33%，跟沒有訓練一樣。