

[NTU ESOE] 110 Data Mining Final Project Deadline:

Student ID : R09631007 Name : 吳乙澤 Department : 生物機電工程學系

ProjectType : Computer Vision Classification_ChineseMNIST

1. 請詳述你對資料作的前處理，以及調整參數的過程邏輯，可以參考 InAnalysis 上 data preview 以及 model preview 的結果來描述，必須超過 350 字。

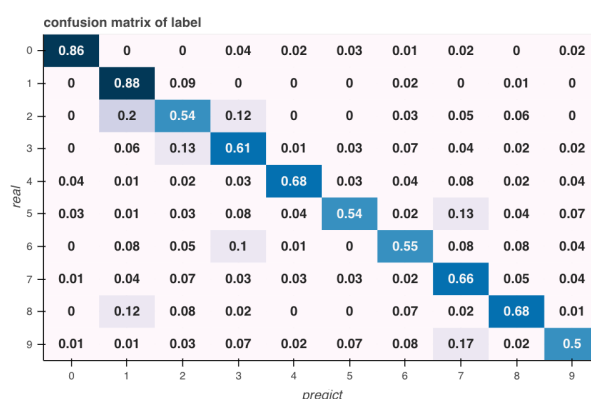
本期末專題沒有用到任何前處理，這是因為該專題目標是針對中文手寫數字零到九，共十個類別做影像辨識。因此，Inanalysis2 平台所提供的前處理功能，如 Outlier 濾除和 Normalize 正規化，是不適用在本次任務的。

在訓練模型前，我利用自己寫的 del_10upData.ipynb、create_Label.ipynb 以及 cut_trainAndTest.Ipynb 三個程式，將原始擁有 15000 張數的中文手寫數字資料集，刪到剩下 10000 張後，以 6:4 的比例，切分出 6000 張的訓練集並壓縮成 ChineseMNIST_Train.zip，以及切出 4000 張的測試集並壓縮成 ChineseMNIST_Test.zip。

訓練模型和調整參數的部分，我先以系統預設的 One_Layer_CNN 演算法，訓練中文手寫數字資料集，看看出來的準確率為何，以當作基準來測試我自己的演算法。由於 One_Layer_CNN 只有一層，所以訓練出來的結果不是很理想。圖一 (a) F-Score 最低還接近 0.5，類似於丟銅板的機率，圖二 (b) 混淆矩陣為真的比率，最低也是 0.5 左右。

Label	Precision	Recall	FScore	Support
0	0.8985	0.8571	0.8773	609
1	0.6083	0.878	0.7187	582
2	0.5186	0.5351	0.5267	598
3	0.5471	0.6143	0.5788	586
4	0.8402	0.6827	0.7533	624
5	0.7339	0.5435	0.6245	609
6	0.5971	0.5456	0.5702	603
7	0.527	0.6623	0.5869	604
8	0.6824	0.6801	0.6813	594
9	0.6689	0.5025	0.5739	591

(a)



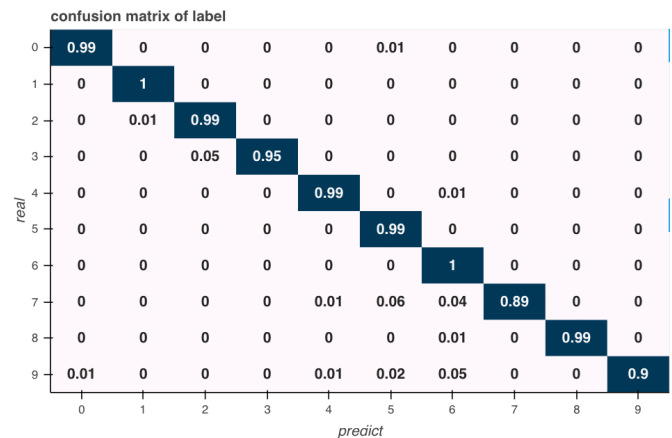
(b)

圖一、(a) 使用 One_Layer_CNN 預設參數訓練出來的模型準確度 (b) 混淆矩陣

為了改善準確率，我改而使用自己寫的 r09631007_CNN 演算法來訓練模型。該演算法具有四層卷積層，兩層全連接層。全連接層中的第一層可以自定義神經元數目，使調整參數有更多的空間。我以 r09631007_CNN 預設的參數訓練模型，此時的 epochs 數為 50，訓練出來的結果就有很不錯的準確度。圖二 (a) F-Score 皆在 0.94 以上，圖二 (b) 混淆矩陣為真的比率也都在 0.89 以上。

Label	Precision	Recall	FScore	Support
0	0.9886	0.9934	0.991	609
1	0.9949	1.0	0.9974	582
2	0.9473	0.9916	0.969	598
3	0.9893	0.9488	0.9686	586
4	0.9856	0.9904	0.988	624
5	0.9153	0.9934	0.9528	609
6	0.905	0.995	0.9479	603
7	0.9963	0.8907	0.9406	604
8	1.0	0.9899	0.9949	594
9	0.9963	0.9036	0.9476	591

(a)



(b)

圖二、(a) 使用 r09631007_CNN 預設參數訓練出來的模型準確度 (b) 混淆矩陣

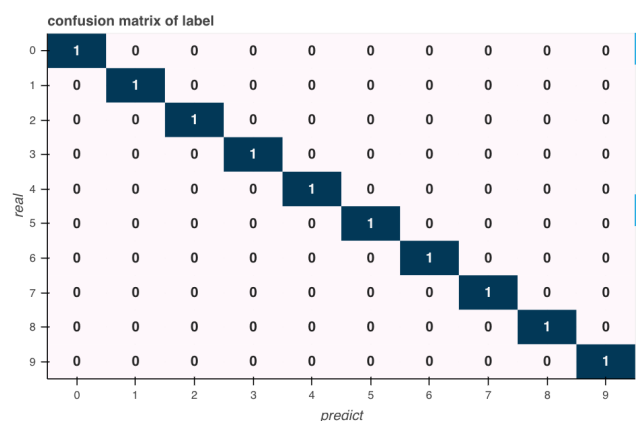
但是我不太滿意以上的準確率，單純的數字辨識準確度應能拉到接近於百分之百。因此，我有再去額外調整參數，使 **epochs** 增大到 150，並增加全連接層的神經元個數，以期訓練結果的 **F-Score** 值逼近於一。

2. 最終的訓練以及測試結果為何？請有邏輯的解釋你的結果。

在使用 150 個 **epochs**，256 個神經元的全連接層進行訓練後，訓練結果如下所示。圖三 (a) **F-Score** 值都接近於 1，圖三 (b) 混淆矩陣為真的比率通通微 1，表示訓練出來的準確率將近百分之百。

Label	Precision	Recall	FScore	Support
0	1.0	1.0	1.0	609
1	1.0	1.0	1.0	582
2	1.0	1.0	1.0	598
3	1.0	1.0	1.0	586
4	1.0	1.0	1.0	624
5	0.9984	1.0	0.9992	609
6	1.0	1.0	1.0	603
7	1.0	1.0	1.0	604
8	1.0	0.9983	0.9992	594
9	1.0	1.0	1.0	591

(a)



(b)

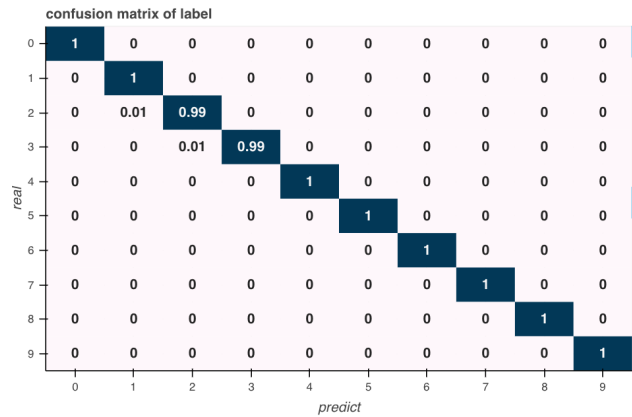
圖三、(a) 調整 r09631007_CNN 參數所訓練出來的模型準確度 (b) 混淆矩陣

光看訓練結果不準，因此我將以上演算法應用到測試集中，得到下方的準確率。圖四 (a) **F-Score** 值都接近於 1，圖四(b) 混淆矩陣為真的比率最低為 0.99，和訓練結果類似，表

示該模型沒有 **Overfitting** 的問題，測試集出來的準確率將近百分之百，有實際應用的可能性。

Label	Precision	Recall	FScore	Support
0	1.0	0.9974	0.9987	391
1	0.9858	1.0	0.9929	418
2	0.9875	0.9851	0.9863	402
3	1.0	0.9879	0.9939	414
4	1.0	0.9973	0.9987	376
5	0.9974	1.0	0.9987	391
6	1.0	0.9975	0.9987	397
7	1.0	1.0	1.0	396
8	1.0	1.0	1.0	406
9	0.9951	1.0	0.9976	409

(a)



(b)

圖四、(a) 調整 r09631007_CNN 參數所訓練出來的模型準確度 (b) 混淆矩陣

3. 報告影片連結（請將報告影片連結附在這邊，並確保影片觀看權限開啟）

影片連結為：<https://www.youtube.com/watch?v=OibadYiMk3A>