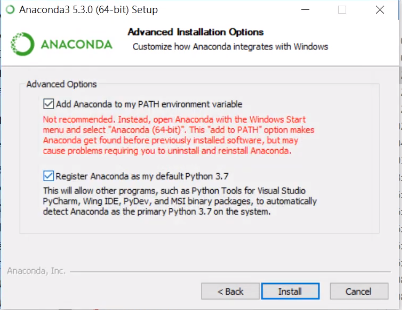
Windows 安裝 python(CPU版)

1.安裝Anaconda3.7版本<https://www.anaconda.com/distribution/>僅有以下畫面務必要做調整,調整如下圖,其餘按next即可。

2.安裝相關套件

2.1搜尋Anaconda Promot

2.2安裝套件(有遇到Y/N選擇,一律選Y)

輸入conda install tensorflow

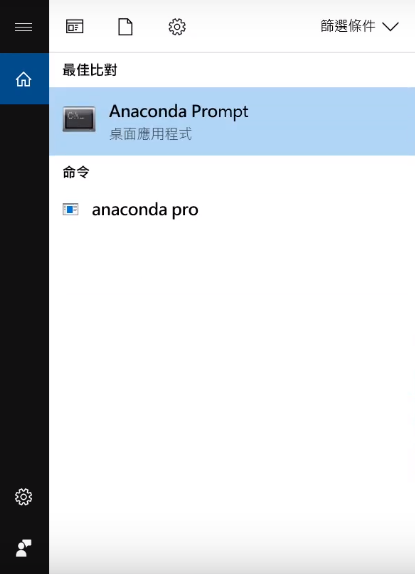
輸入conda install keras

輸入conda install pandas

輸入conda install matplotlib

輸入pip install librosa

輸入pip install opencv-python

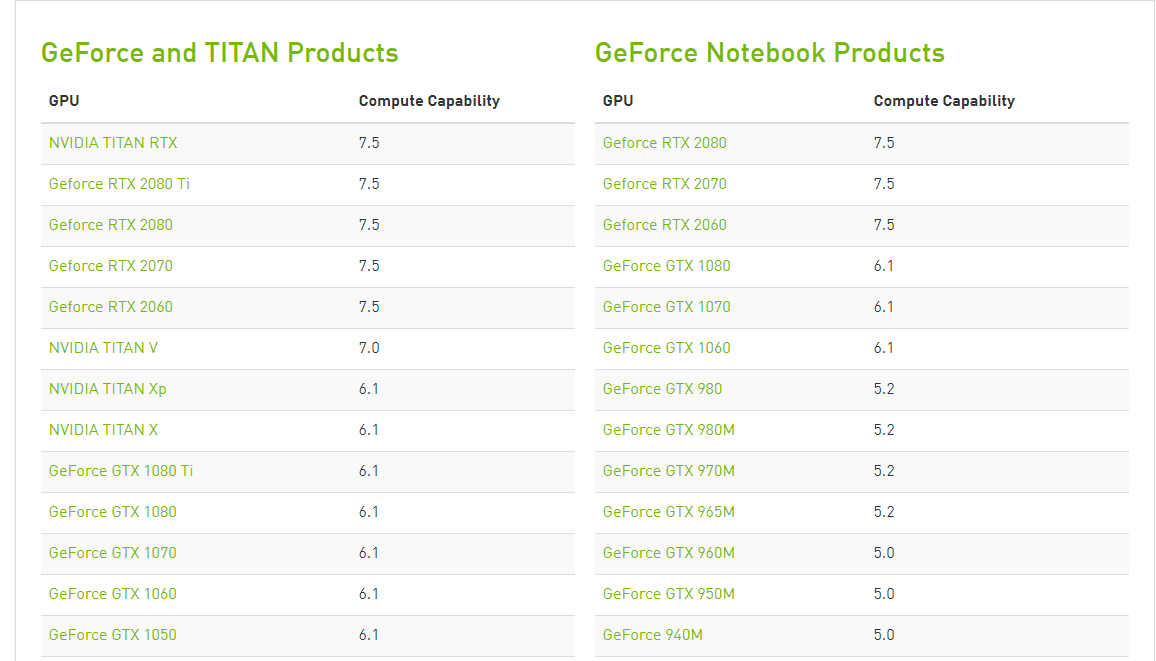
2.1

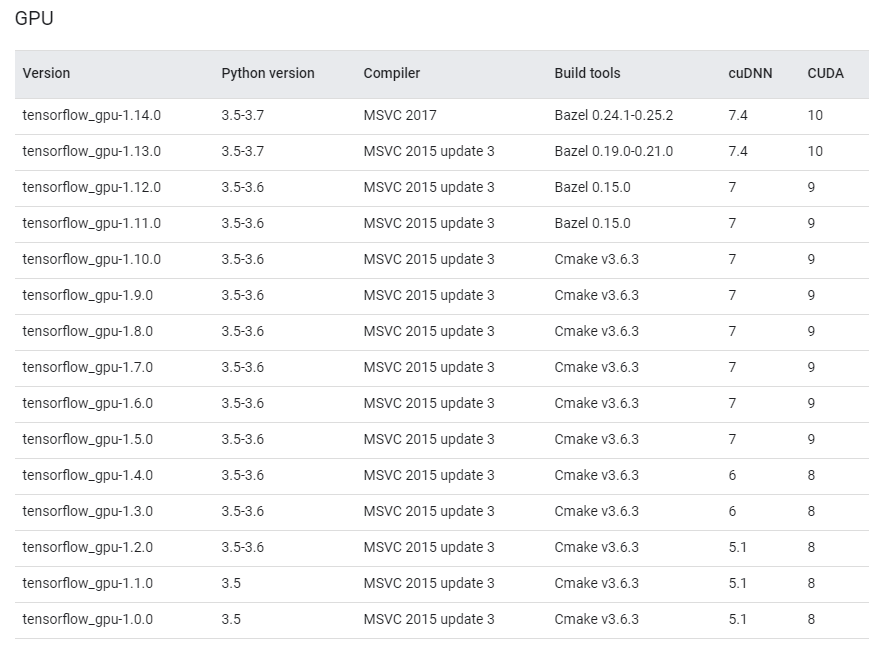
3.輸入jupyter notebook,則開啟jupyter notebook

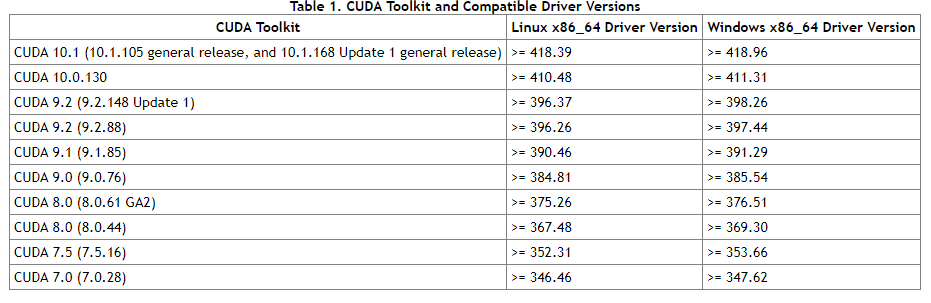
詳細可參考<https://www.youtube.com/watch?list=PLs94gvnxOkCvtnuVJ7tJyhO8KiJa2PtkV&time_continue=113&v=B-3kBy0gXlM&feature=emb_logo>

Windows 安裝 python(GPU版)

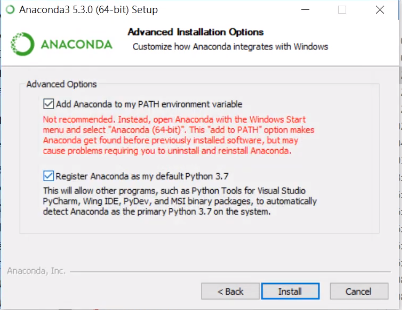
1. 請至Nvida官網查看你的GPU性能<https://developer.nvidia.com/cuda-gpus#compute>範例圖如下:



2.安裝GPU驅動程式Tensorflow版本與CUDA toolkit對應關圖如下

CUDA toolkit與GPU driver對應關係圖如下-若我今天選擇CUDA10.1,GU驅動就要升級至418.96版以上

接下來請至<https://www.nvidia.com.tw/Download/index.aspx?lang=tw>填入你的GPU型號、系統規格並按下搜尋,範例圖如下出現了431.36版，該版>418.96，OK，可以下載進行安裝

3.安裝Anaconda3.7版本<https://www.anaconda.com/distribution/>僅有以下畫面務必要做調整,調整如下圖,其餘按next即可。

4.安裝Tensorflow(GPU版本)-

4-1開啟CMD

4-2輸入conda install tensorflow-gpu=1.13(有遇到輸入Y/N一律輸入Y)

4-3安裝完成後，輸入指令conda list,看有無畫線的程式

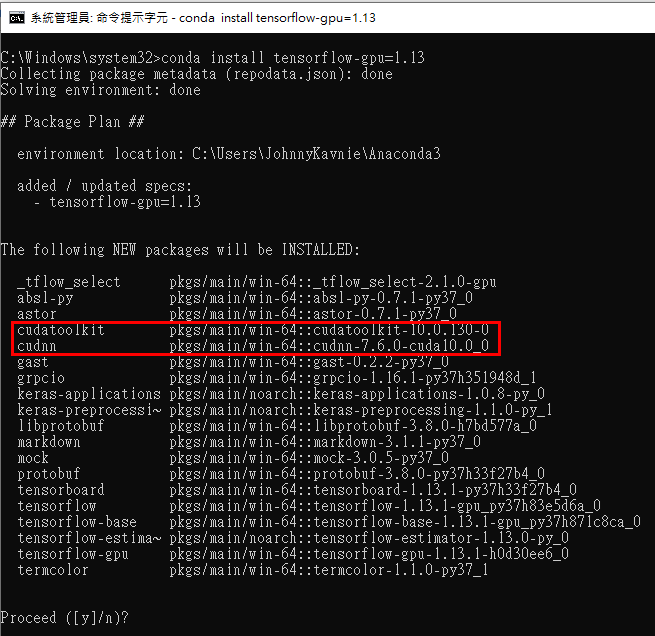
4-4測試安裝是否成功

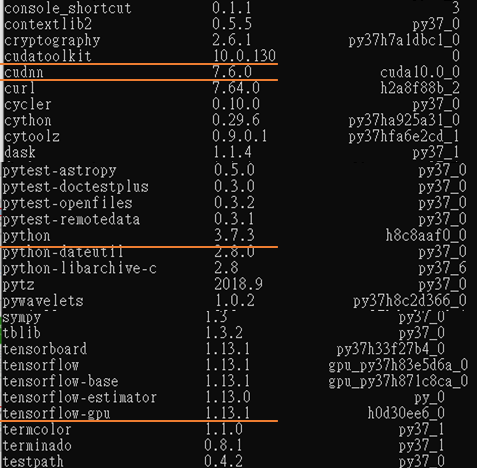
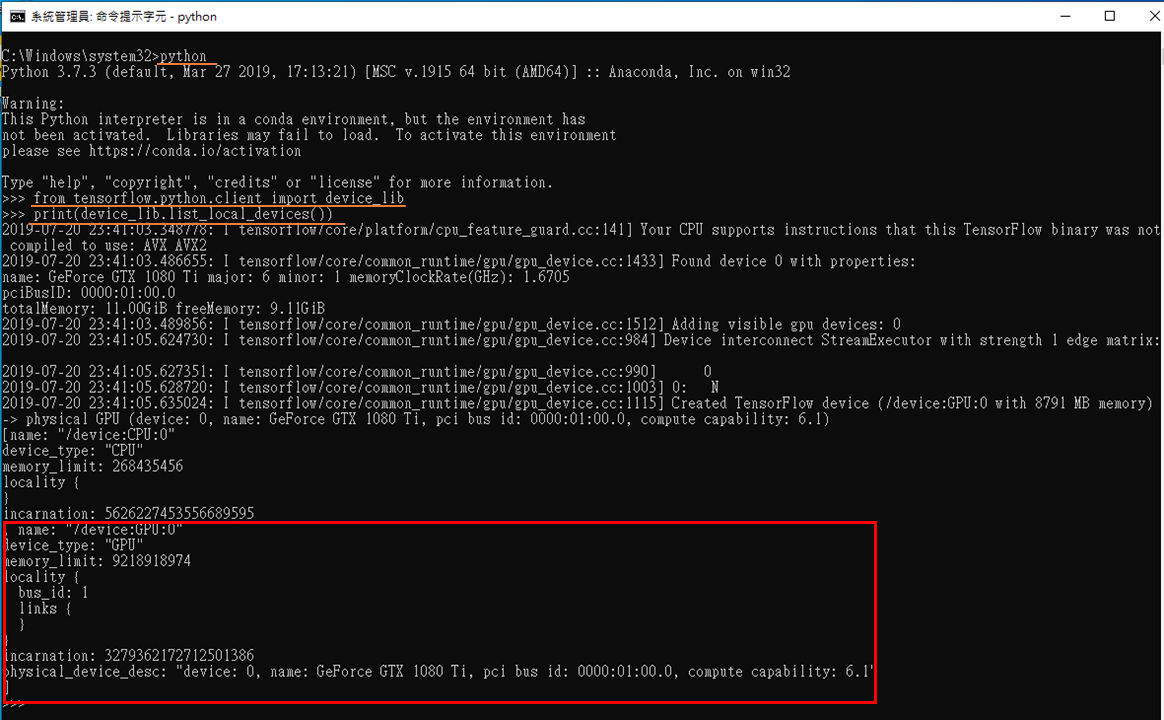
輸入python

輸入from tensorflow.python.client import device\_lib

輸入print(device\_lib.list\_local\_devices())

4-1

4-2

4-34-4

5.建立虛擬環境-進入命令提示字元

5-1輸入conda create --name tf\_1.9 python=3.6

(若是要安裝3.7版本將3.6改3.7即可)

5-2進入虛擬環境

輸入conda activate tf\_1.9(進入環境)

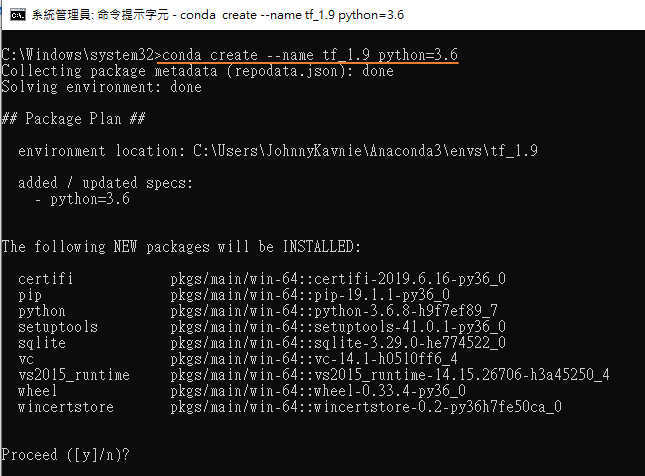
輸入conda install tensorflow-gpu=1.9(安裝)

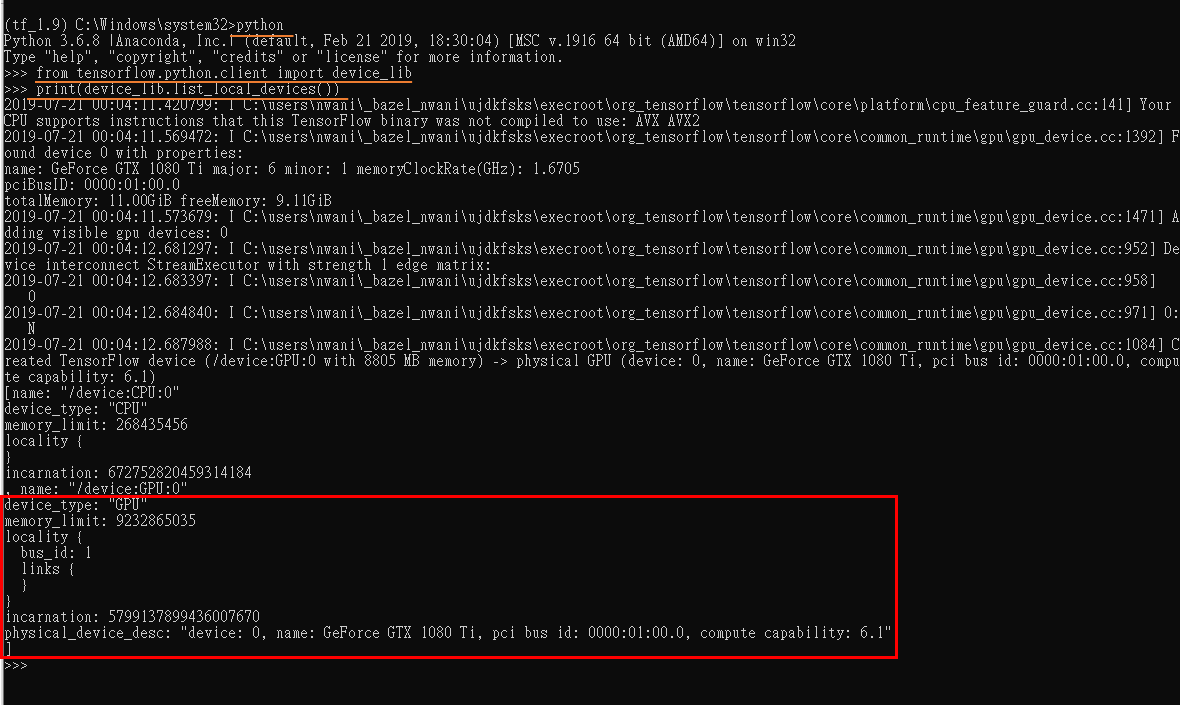
5-3測試安裝是否成功

輸入python

輸入from tensorflow.python.client import device\_lib

輸入print(device\_lib.list\_local\_devices())

5.1

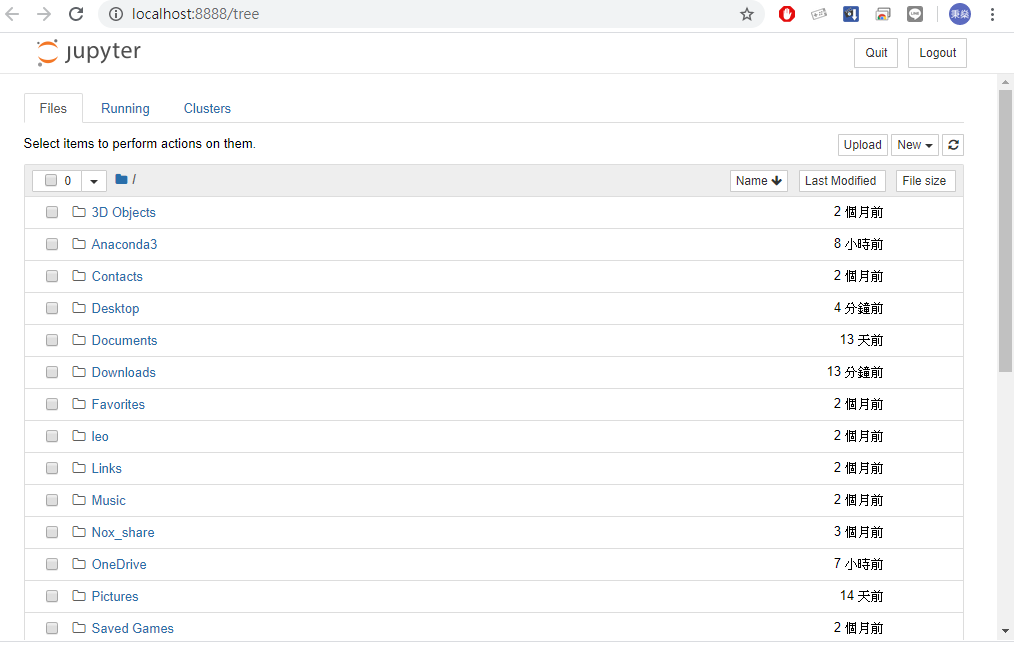
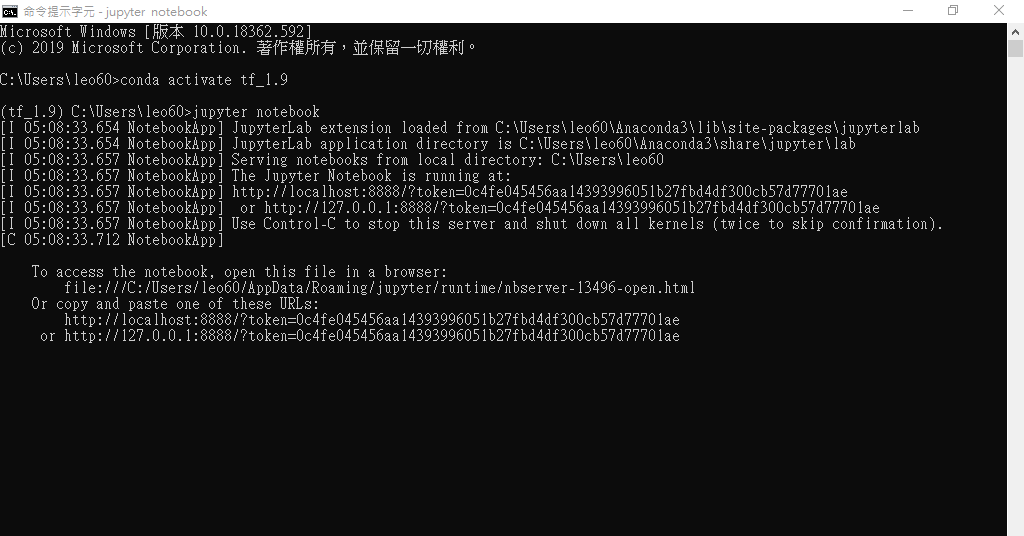
5-2,5-3(P.s我個人是有遇到測試是否安裝成功時,紅色框框沒有GPU,他說因為CUDA版本不符合,因此到<https://developer.nvidia.com/cuda-downloads>下載即可,再者因為安裝GPU版本,很吃CUDNN,CUDA,Tensorflow-gpu的互相匹配性,因此安裝若是出錯可以朝這方面去看)

6.開啟juputer notebook

6-1打開CMD

6-2輸入conda activate tf\_1.9(因為前面設的環境叫tf\_1.9)

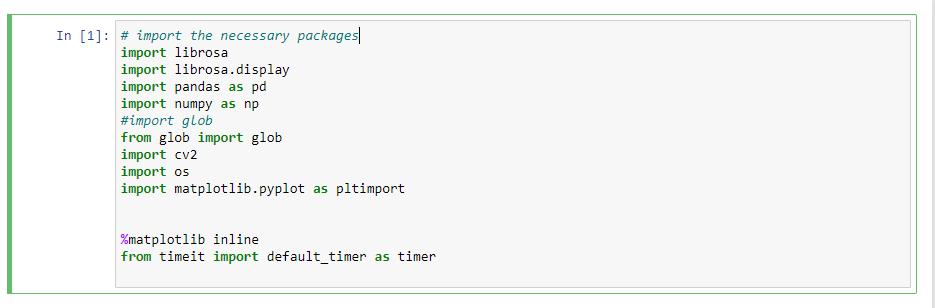
6-3輸入jupyter notebook



**步驟一:載入套件-一般來說不會有librosa 與CV2因此要進tf\_1.9環境裡安裝**

輸入pip install librosa

輸入 pip install opencv-python



導入套件,若是沒有安裝這些套件,程式則無法執行。

**步驟二:建立CSV檔-事前要先把所有資料分成訓練跟測試,約7:3,CSV做成下面圖片務必是隨機排列,因為隨機排列的成果會遠優於排列完整的情形。**

write\_to\_csv-定義函式名

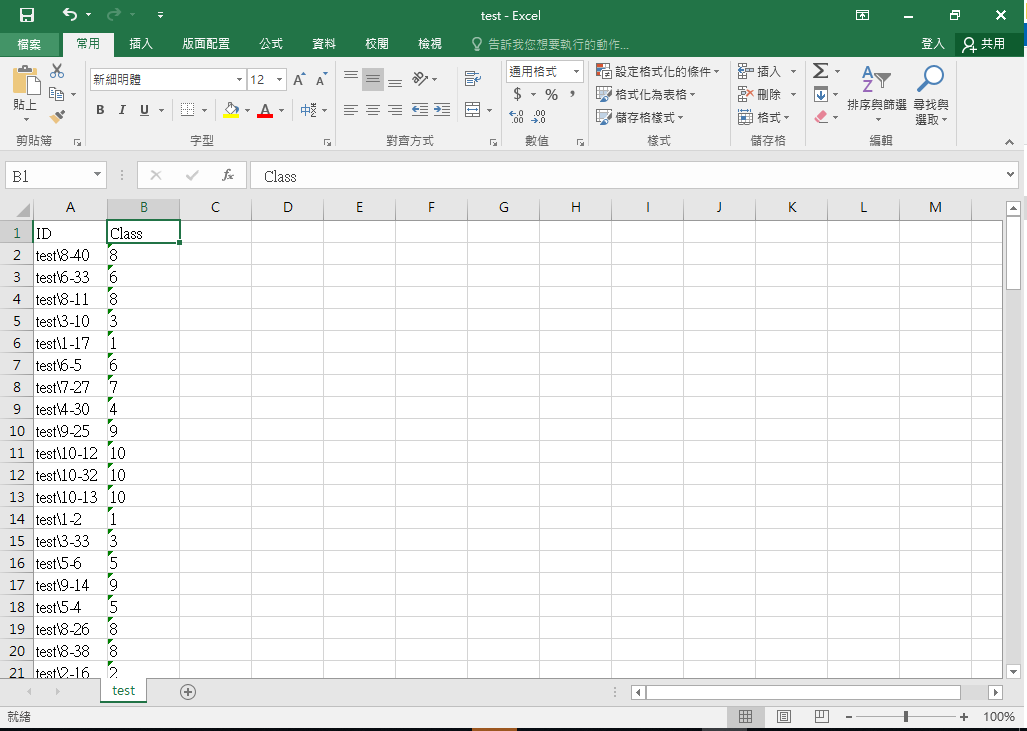
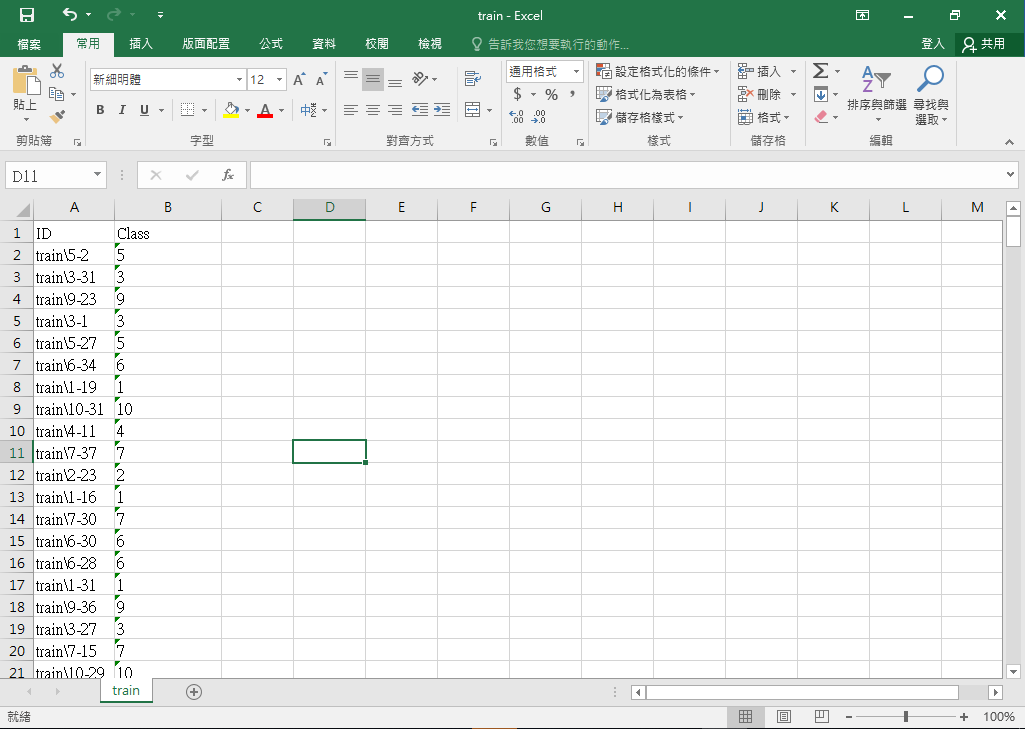
csvName, dataDir-變數名

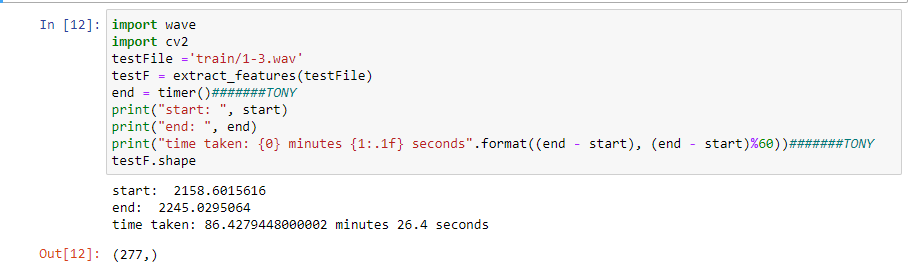
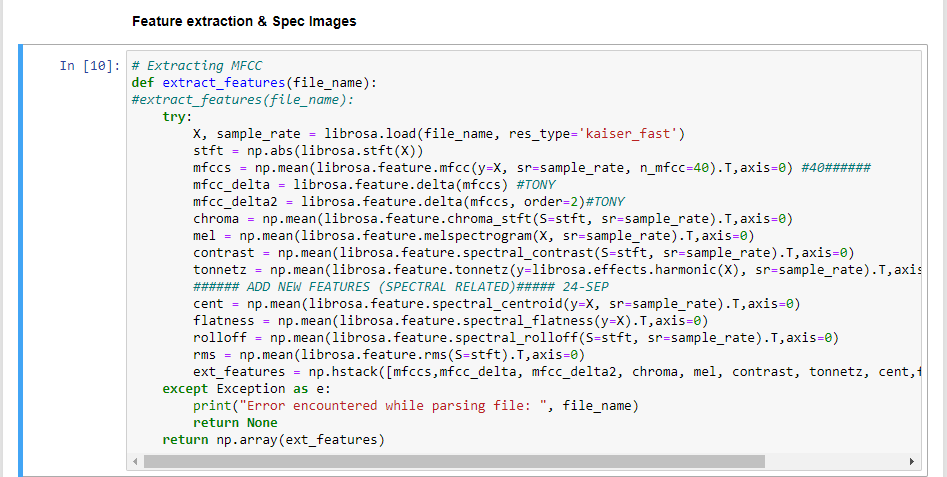
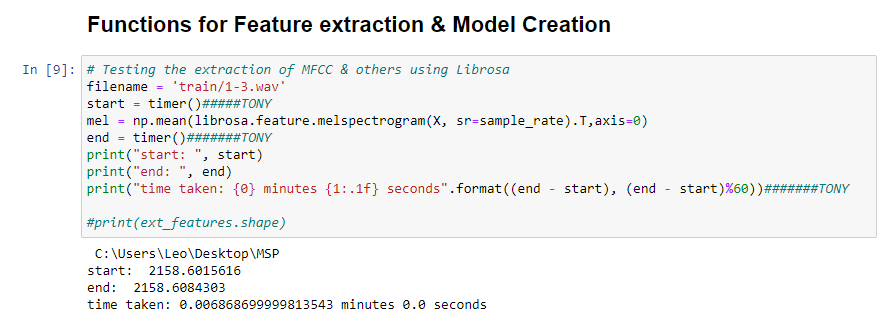
fileID-檔案名稱

classID-類別

split-分隔符號

glob("train/\*")-檔案位置



**步驟三:提取特徵值與創建模型-基本上只有filename要更動**

filename-檔案名稱

librosa.load(filename)-讀取資料

librosa.feature.melspectrogram-計算梅爾頻譜圖

mel-計算梅爾頻譜圖

chroma-根據波形或功率譜圖計算色譜圖。

Contrast-計算光譜對比度

tonnetz- Computes the tonal centroid features

cent-計算頻譜質心

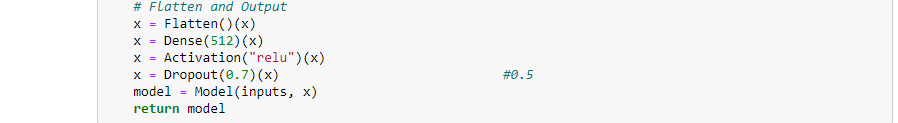
flatness-計算光譜平坦度

rolloff-計算滾降頻率

rms-根據音頻樣本或聲譜圖計算均方根

ext\_features-將所有特徵矩陣堆疊起來

shape-維

步驟四:創建MLP與CNN模組-可自行改參數輸出結果會有變化

model = Sequential()-建立線性堆疊模型

model.add(Dense(256, input\_shape=(277,)))-256為神經元個數,277為輸入維數

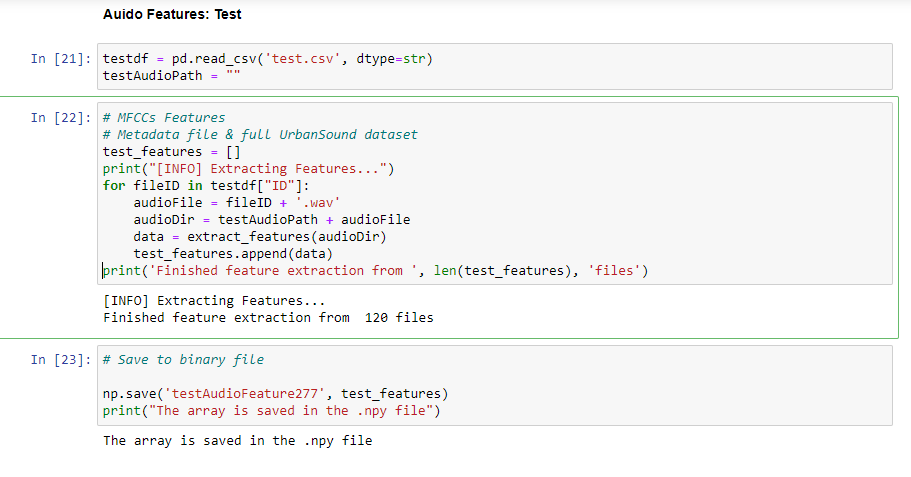
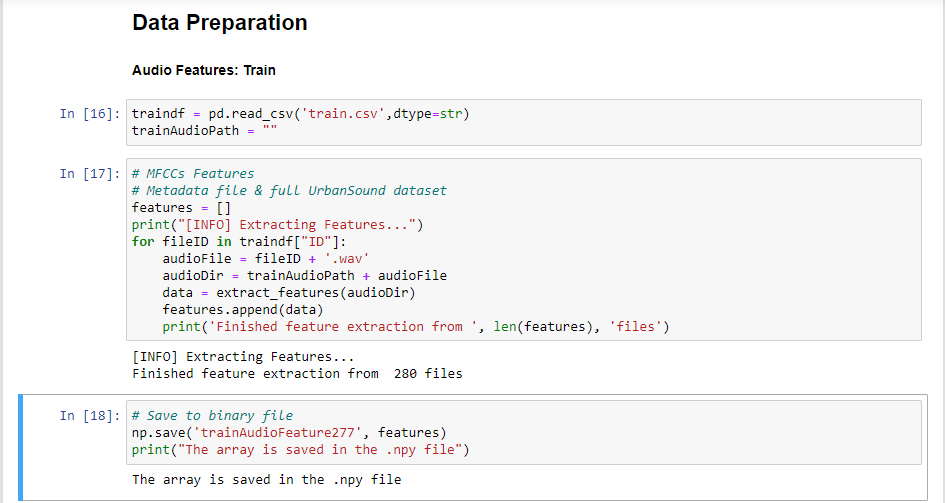
Activation('relu')-定義激活函數為relu

Dropout(0.5)-每次訓練迭代時,隨機在隱藏層中放棄50%神經元

Conv2D(32, (3, 3), padding="same")-32為濾鏡filter weight個數,(3,3)為濾鏡大小為3X3, padding="same”-設定讓捲積運算,產生的捲積影像大小不變

pool\_size=(2, 2)-尺寸縮小一半

Flatten()-將影像轉換成一維的向量

**步驟五:抓取特徵並且儲存成.npy檔**

traindf-讀取訓練CSV檔

testdf-讀取測試CSV檔

trainAudioPath-檔案位置

trainAudioFeature277-訓練特徵儲存之檔名

testAudioFeature277-測試特徵儲存之檔名

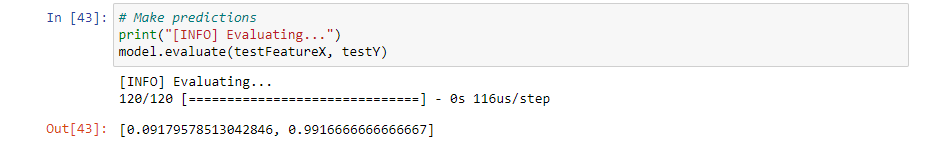
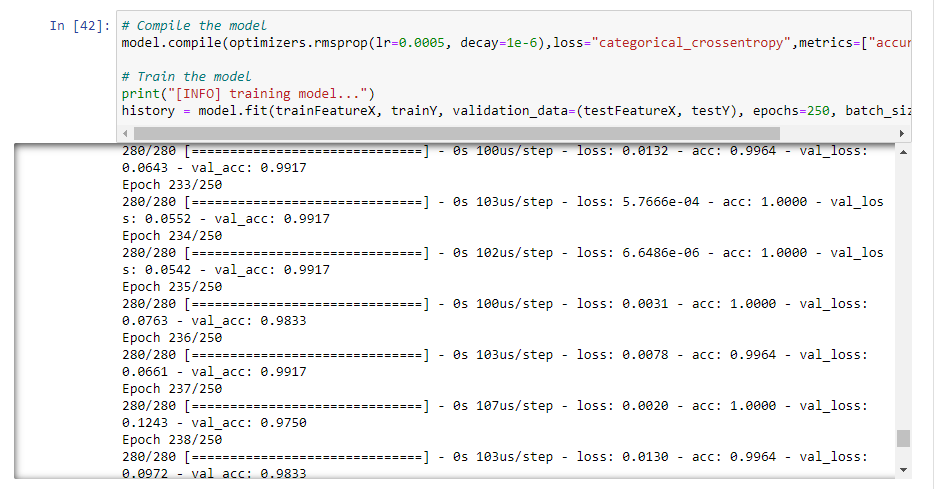
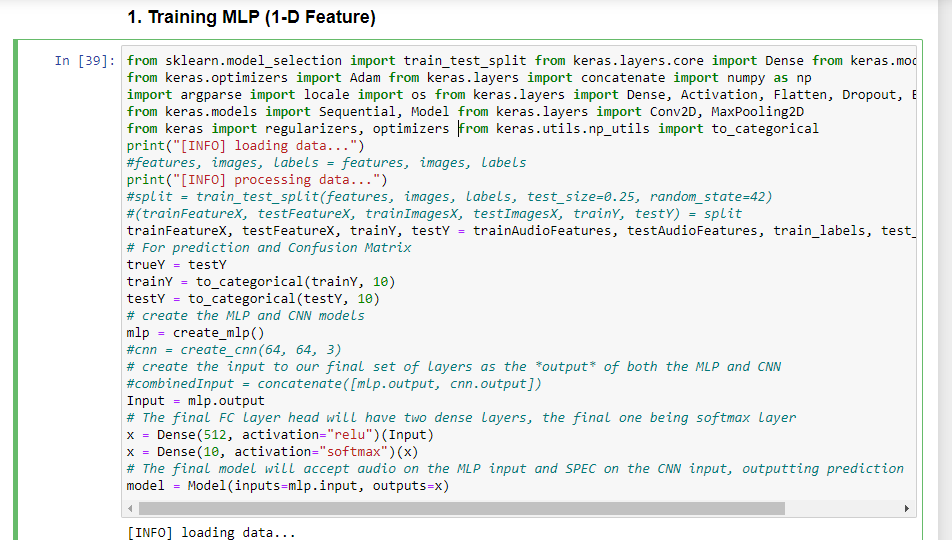
步驟六:讀取特徵與真實值

trainAudioFeatures = np.load('trainAudioFeature277.npy', allow\_pickle=True)-載入先前做好的測試特徵值

train\_labels = traindf["Class"]-訓練的label為,CSV檔案中class那行

test\_labels = testdf["Class"]-測試的label為,CSV檔案中class那行

le.transform-轉換成整數

**步驟七:**MLP訓練測試

trainY = to\_categorical(trainY, 10)-10為有10種可能分別是1-10號語者

testY = to\_categorical(testY, 10) -10為有10種可能分別是1-10號語者

mlp = create\_mlp()-呼叫先前的create\_mlp()

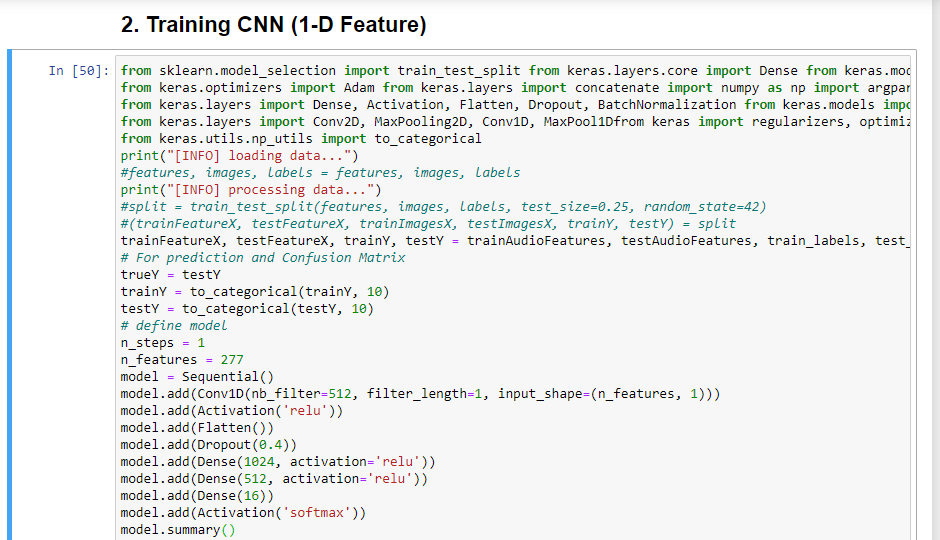
x = Dense(10, activation="softmax")(x)-使用softmax激活函數進行轉換,10為對應到結果有1-10共10位語者

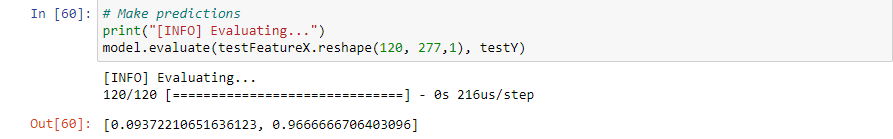
optimizers-設定訓練時的最優化方法

loss-設定損失函數

epochs=250-執行250次訓練週期

batch\_size=32-每一次32筆資料

**步驟八:CNN訓練測試**



trainY = to\_categorical(trainY, 10)-10為有10種可能分別是1-10號語者

testY = to\_categorical(testY, 10) -10為有10種可能分別是1-10號語者

model = Sequential()-建立線性堆疊模型

nb\_filter=512-filter 個數

filter\_length=1-filter 長度

input\_shape=(n\_features, 1)-輸入維度

Activation('relu')-定義激活函數為relu

Flatten()-將影像轉換成一維的向量

Dropout(0.4)-每次訓練迭代時,隨機在隱藏層中放棄40%神經元

Softmax-使用softmax激活函數進行轉換,10為對應到結果有1-10共10位語者

model.summary()-查看模型摘要