AI CUP 2023 春季賽
多模態病理嗓音分類競賽報告

隊伍：TEAM_3373
隊員：陳漢棠（隊長）
Private leaderboard：0.587315 / Rank 14

壹、環境

- 作業系統：因為是使用Colab來訓練模型與撰寫程式碼，所以有時候是用桌面型電腦的Windows 10，有時候是用筆電的MacOS 13。
- 語言：使用的語言為Python 3
- 套件：一些基本的套件包含Os：讀取資料、Copy：用來複製物件，這樣才可以避免函式修改傳進來的參數。、Numpy：主流的機器學習資料操作程式庫。、Pandas：提供符合直覺的資料操作。還有用來讀取音檔的Scipy、Libros a：用來對音檔進行MFCCs，用來架設神經網路的Tensorflow.Keras，最後就是傳統的機器學習模型Sklearn：提供了諸如分離訓練驗證資料、混肴矩陣、支援向量機等等非常方便的功能。

並且這次所使用的程式碼是從主辦方所提供的參考用程式碼改編而成。

```
import os
from google.colab import drive

import copy
import numpy
import numpy as np
import pandas as pd
import librosa

import tensorflow as tf
from tensorflow.keras.layers import Activation, BatchNormalization, Dense, LayerNormalization
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras import datasets, layers, models

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, recall_score
from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn import svm
from sklearn.metrics import classification_report, accuracy_score
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression, SGDClassifier

import scipy.io.wavfile
from scipy.fftpack import dct
```

圖一、程式所使用的函式庫

貳、演算方法與模型架構

這次的訓練資料分為兩個部分，聲音與有二十六種特徵的病歷資料。所以針對不同形式的資料需要用到不同模型。並且為了最佳化預測結果，我採用Intermediate fusion。首先分別訓練出數值資料與聲音資料的模型，再訓練一個結合了數值特徵與聲音特徵的模型，最後結合這三個模型所預測出的答案作為輸入，再訓練一個模型去決定最後的分類。

對於病例報告的二十六種特徵，因為只有一千個訓練資料，對於深度學習是稍嫌不足的，所以我選擇了一些傳統的機器學習模型如：Logistic Regression classifier、SVM、Gaussian Naive Bayes、Linear models with stochastic gradient descent。這些傳統的機器學習模型在少量的訓練模型中有更佳的準確率。為了有更全面的比較，最後還是與一個200 神經元 、5 層的全連接神經網路進行比較，選擇出一個召回率最高的模型。

針對音檔資料，將其透過MFCCs處理完後，可以用CNN進行預測。這裡採用的是四層的Convolution layers：激活函數為Relu、Kernel size 皆為3*3。而filter 分別是32、64、96、192。經過卷積層後，將資料進行攤平，並連接到一個64個神經元，激活函數同為Relu的全連接層。最後再由Softmax函數輸出成五個種類。

Early fusion 的模型則是採了一般的DNN。600 神經元、5 層的全連接神經網路，激活函數為Relu，並且每一層都有0.1的 Drop out與Batch Normalization。最後經過Softmax輸出並預測。

最後的決策則是由三個召回率最高的模型所做的預測作為輸入資料，並訓練一個20 神經元 、3 層的全連接神經網路，並且每一層都有0.5的 Drop out與Batch Normalization。最後預測出的答案與將前面三個模型的預測直接加權平均的結果進行召回率的比較。選出最佳的結果。

參、創新性

整體的架構是閱讀了一些文獻後，自己設計的架構，因為我認為更全面的訓練有助於找出正確答案。所以不單單只參考病歷與音檔，更要訓練病歷與音檔一起參考的模型並且在這些模型所預測的答案中用再一層模型，或是用召回率的加權平均來找出最佳的答案。並且，我所採用的MFCCs並不是完整的從一開始的預強調到後來的餘弦轉換，而是跳過預加強並且只到帶通濾波器的步驟，這是因為我認為一開始預加強會破壞聲帶可能因為受損，所以在不同頻率會有不同的特徵。並且我認為最後的離散餘弦轉換會去除掉特徵之間的相關性，但是相關性對於神經網路可能有正面的幫助，所以就不去作餘弦轉換。神經網路的部分是從主辦方提供的參考程式碼修改而來，但其中的大小與深度都是自己測試而來。還有針對每個不同種類的權重也是自己調整出來的，訓練部分的超參數也是慢慢測試出來的。

肆、資料處理

先讀入病歷資料，並分成850筆訓練資料與150筆測試資料。對於輸入數據，把“ID”與”Disease category”這兩欄拿掉，再來就是針對剩下的每一項特徵作一般化，讓全部特徵都落在0到1之間。對於目標值，就是把”Disease category” 這一欄拿出來，並且做One-hot encoding成五個種類。對於音檔，我們依照讀入病歷資料的”ID”依序讀入檔案，並針對每個音檔做：

1. 音框化：因為隨著時間推移，頻率的變化會讓傅立葉轉換變得沒有意義。但是在極短時間內聲音頻率的變化非常小，所以我們可以對每一個短幀進行傅立葉轉換，並且透過重疊相連，取得平滑的頻率輪廓近似。我選擇讓每個音框的長度為25 ms 並且，音框間相距10 ms。
2. 加漢明窗：對每一幀乘上框函數，以增加音框左右的連續性。改善之後傅立葉轉換的誤差。
3. 傅立葉轉換：一般來說，聲音在時域上很難看出不同的特徵與訊號的特性，所以我們可以把身音訊號轉換成在頻域上的能量分佈，才能觀察出更多特徵。
4. 濾波組：40個依照美梅爾頻率平均分佈的三角濾波器，與傅立葉轉換後的頻譜進行濾波，就會得到每一幀在該濾波器的對應頻段的能量值。

對於病例特徵的模型，輸入資料就是26個特徵。對於音訊特徵的模型，因為每個音檔長度有所不同，所以會統一將音檔延長至3.5秒，少的補零多的截斷，並做上面提到的1~4操作。再來就是混合模型，我選擇把一秒的音檔（也是少得補零多的截斷）作以上操作，並且把得出的二維資料做攤平，並連接到26個病歷特徵後面。

伍、訓練方式

Models	Learning rate	Loss funci ton	Optimizer	batch size	Epochs
text	0.0001	categorica l_crossent ropy	Adam	256	10000
audio	0.0001	categorica l_crossent ropy	Adam	256	3000
fusion	0.001	categorica l_crossent ropy	Adam	256	10000
fianl	0.0001	categorica l_crossent ropy	Adam	256	3000

表一、各模型超參數

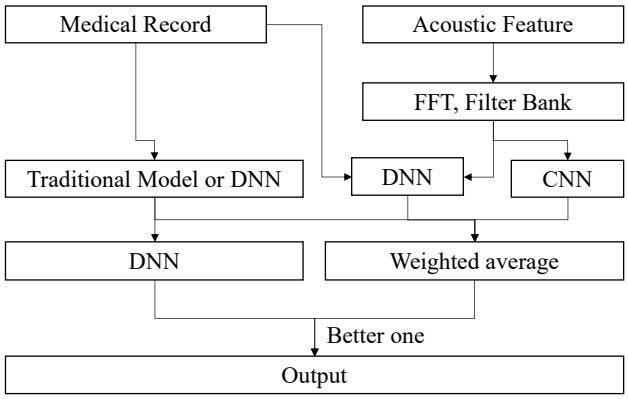
每個模型都會觀察Validation loss做Early stopping，並儲存Validation accuracy 最高的模型參數。並且除了上述的模型以外，以病歷特徵作為輸入的模型還有Sklearn提供的傳統機器學習模型：高斯貝氏分類器、支援向量機、邏輯回歸還有梯度下降的線性分類器。詳細的參數都可以在程式碼裡找到。值得注意的是，因為這次比賽所提供的資料各個種類之間的數量並不平均，所以在訓練的過程中可以對不同種類加入不同的權重，這樣能顯著地提高分數，因為本比賽的評分方式每個種類都一樣重要，資料比較少的種類甚至更為重要。我讓每個種類的權重與訓練及所提供的數量成反比，每個種類的權重為：嗓音誤用：1、聲帶閉合不全：2.436、聲帶麻痺：3.19、聲帶腫瘤：12.182、聲帶正常：16.75。

先分別訓練出病歷模型、聲音模型與融合模型，儲存表現最好，召回率最高的模型後，在由這三個模型做出預測。並把結果合併為一個矩陣，作為最後的模型的輸入，並把最終模型的輸出與三個模型的預測的加權平均做比較，繳交較為高分的答案。最後為了不要浪費任何資料以提高分數，所以在確定好優良的模型架構後，把一千筆資料都分成訓練集，在繳交預測出的測試集答案。

陸、分析與結論

Models	Validation Ratio	Public Score	Private Score
All Neural Network Models	0.01	0.485441	0.529318
All Neural Network Models	0.15	0.489767	0.518856
Logistic Regression	0.01	0.553982	0.587315
Logistic Regression	0.15	0.552543	-

- 可以看到，如果把資料都拿來訓練實際上獲得的分數確實有比較高。
- 上面兩列是依照Intermediate fusion的方式從病歷、音檔、混合，並在最後用加權平均的方式做出的預測。
- 下面兩列僅僅靠病歷的二十六個特徵作為輸入，用Sklearn的Logistic Regression訓練出來的結果。



圖二、完整架構圖

最後，最好的成績竟然是只靠著病歷報告所訓練出來的邏輯回歸模型所預測的答案，所以可以說這次的架構都白費了，連音檔都沒有用到。但是我覺得應該不是整體架構的問體，會導致這種結果，很可能是因為我訓練出的深度學習模型召回率都太低了，所以這次能榜上有名應該算是運氣好。可以說這次深度學習模型的參數都是我自己調出來的。並沒有找到什麼很好的文獻可以參考，像是總共要幾層、每層要多少神經元、要多少Dropout等等，都是我看召回率慢慢調整出來。所以可以說整體性能非常差，最後還是只有30%左右的召回率。這是我第一次參加機器學習的比賽，所以有很多要改進與調整的地方。主要需要改進的地方是對於深度學習架構的不熟悉，我相信參閱更多的文獻能找出訓練及大小以及複雜度與神經網路的架構之間的關係。還有訓練集不平均的問題，因為最後時間不夠，本來想加入Saarbruecken Voice Database 之中的正常聲音，看看召回率如何，我相信更平均的訓練集應該能訓練出更好的模型。

柒、程式碼

Github連結：https://github.com/tomchen86/ai-cup-2023-pathological-voice-classification

捌、使用的外部資源與參考文獻

1. fayek, H. m. (2016, April 21). Speech Processing for Machine Learning: Filter Banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What’ s In-Between. Haythamfayek. https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html