AIcar实验报告

苏吉雅 2016202116

**一、完成的模块：**

蓝牙电脑交互、GMM声纹识别、语音控制、代码整合

**1.1蓝牙电脑交互模块：**

调用的pyserial库进行蓝牙的交互

ser = serial.Serial("COM4", 9600, timeout=0.5)

蓝牙连接的是COM4虚拟端口，频率和arduino上设的相同时9600，如果0.5s没有响应就退出

if ser.isOpen():

print("succeed")

else:

print("fail")

如果蓝牙成功连接打开，打印succeed。

ser.write(d.encode())

给蓝牙发信号，d是发送的内容，.encode是对d转码，否则会报错。

ser.readline()

读一整行蓝牙发的数据，这里我读的是一串乱码，如果不是读一整行，小车运行会出问题，在原地不动，原因不太清楚。

之前因为蓝牙装反导致蓝牙烧坏等一系列问题导致我们在蓝牙和小车交互上有很大的问题，解决方案是又买了两个蓝牙，并更改读的模式，从一个字符改为一行。

电脑蓝牙还有一个问题，每发一次信号只能移动一下，但是手机蓝牙发一次信号可以一直移动，不知道原因。

**1.2声纹识别：GMM模型**

GMM即高斯混合模型，其基本思想是通过对训练数据建立概率模型，再将测试数据代入每一类的模型中计算出它属于这个类别的概率，最后，比较可得出概率最大的类别，即得预测值。这里是收集了三个人的对小车名字“畅畅”的十次音频，每个人建立了一个GMM模型，然后将测试音频放入这三个GMM之中得出属于这三个人的概率，概率最高的那个人作为预测值返回。

**GMM的搭建过程：**

（1）BIC算法：使用Bayesian Information Criterion算法确定最佳核数K（Gaussian的数目）

（2）EM算法：确定隐含变量和每个Gaussian的参数。

**a.参数初始化**

**样本属于某个隐含变量的概率P（h）：**采取平均的方法，使测试值属于每种类别的概率均为1/K

**均值μ：**在确定了核数K后，使用kmeans的方法聚K类，取质心用来初始化各个均值

协方差矩阵Σ：先用全局的协方差作为替代。

**b.Expectation step**

利用参数的初始化值或上一次迭代的模型参数来算出隐变量的后验概率，即隐变量的期望，作为隐变量的现估计值

**c.Maximization step**

将似然函数最大化，更新参数值

d.重复bc步，反复迭代直至收敛。这里判断收敛的标准是其对于参数的更新不再有明显提高，停止迭代阈值采用默认值tol=1e-3

**代码过程：**

所用库：wave、numpy、scipy、sklearn

1. wav音频数据读入
2. 加窗分帧、特征提取

对每个音频，我们通过加窗和分帧，能得到帧数不同，但每帧向量维度相同的MFCC特征数据。在GMM中，我们把每个数字的音频样本中分出来的帧看做独立，则所有帧的MFCC即可直接投入GMM中进行训练。而且，因为训练以帧为单位进行，也不需要考虑音频长度不同带来的问题。



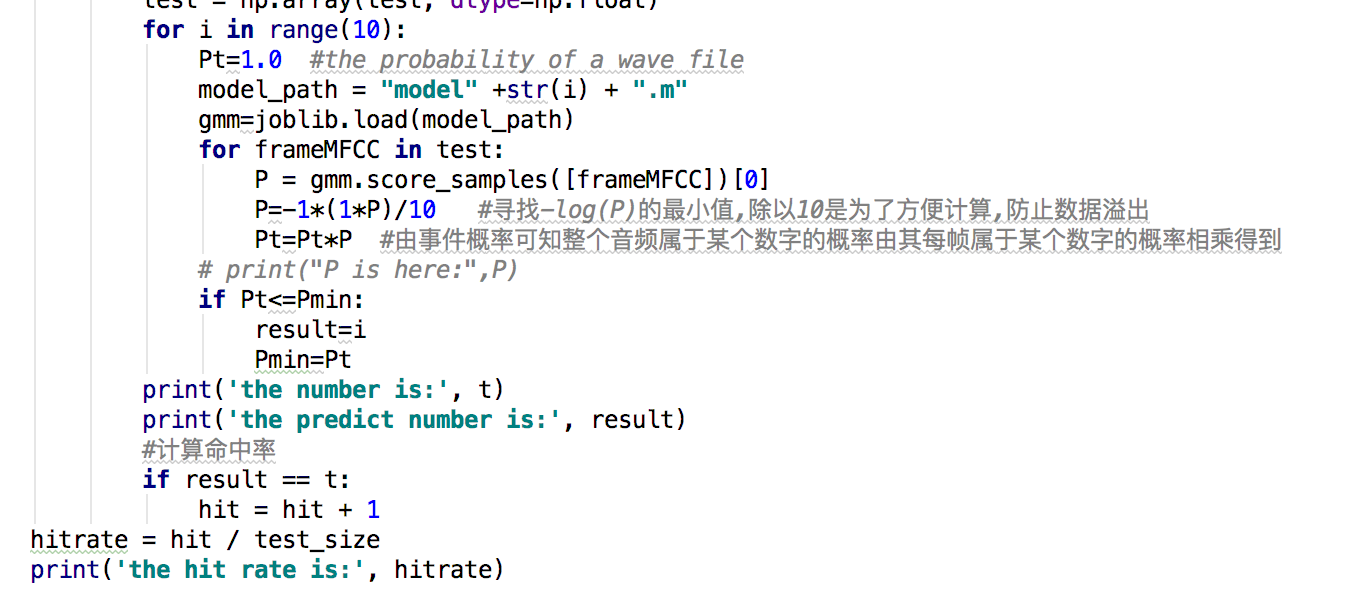
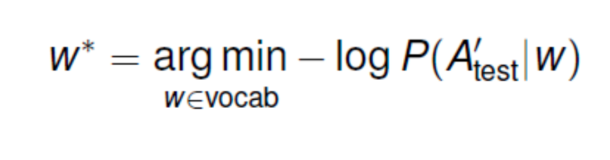
1. 对每个数字，直接调用sklearn库中的mixture.GaussianMixture库训练模型并保存，这里后期还需要调整主要参数，寻找最优值。

在这里，我使用了bic算法，计算不同K值下的惩罚，选择惩罚最小的K值，从而确定每个数字最优的高斯核数。



1. 对每个测试数据，分别调出十个GMM模型，计算出概率的log值的相反数，挑出概率最大时对应的数字，作为预测值返回。

经过以帧为单位的训练，我们得到的GMM模型是一个输入为测试帧，输出为测试帧属于该GMM模型对应说话人的概率。而对于一整个测试音频，它的每一帧属于某一数字的概率之积，就是这个音频属于某个数字的概率。注意，这里如果直接使用概率值，可能会出现由于概率值太小，贴近于0而出bug的问题，因此我们可以直接对概率取log再取相反数进行比较，为了防止溢出，可以再除以10。这些操作对最后的乘积影响不大。



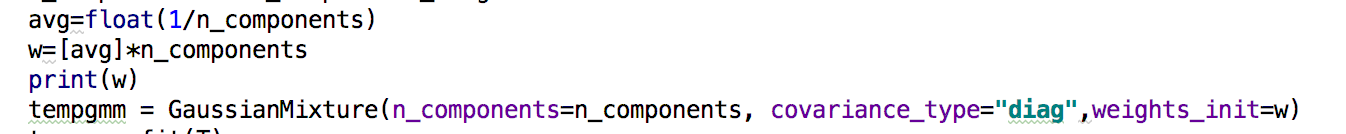
1. 测试、修正与调整

**关于 covariance\_type协方差阵类型**

上课时我们了解到，在ASR中为了避免参数过多开销过大，我们一般采用对角协方差阵，因此，我们选择 covariance\_type=diag。对于该参数，我也选用过full的尝试，比如，采用diag的命中率为65%，而采用full的命中率也为65%，多次试验可以发现开销过大且效果提高不大，遂放弃，确定选用对角协方差阵diag

**关于每个高斯的权重初始化**

我采取了平均的方法，使测试值属于每种类别的概率均为1/K，在gmm中通过设置参数weight\_init确定。但这种方法和默认的KMeans初始化方法相比似乎差距并不大。



**关于每个声纹gmm模型的高斯个数选择**

为选择出每个声纹gmm模型的最佳高斯个数，我在其中加入了BIC算法。BIC算法对不同的高斯个数选择所带来的惩罚进行了比较，选择了惩罚最小的作为该GMM模型选用的高斯核数。对于说话人，我都对1-50个核进行了遍历，发现bic算法显示的最优核数在18左右。然而，bic算法的最优似乎并不代表着算法最终的命中率最优。经过对数据集大小、对计算开销、bic最优结果、算法命中率的权衡，我将高斯核数目限制在了15以下。经试验发现，每个数字最优的高斯核数可能是不同的，但差异不会非常大。

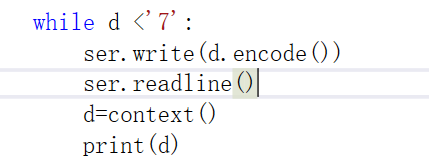


**1.3语音控制模块：**

语音控制是我根据百度语音识别API转的文字来实现的。和上面类似，我对list[0]逐字识别，当识别到‘前’或者‘直’返回1给小车，当识别出‘后’返回2，以此类推。



由于电脑控制蓝牙发消息时每次只能移动一小段，我这里采用while循环不断读取go,wav的文件，不断识别，不断发送信号给小车，从而达到连续移动的效果。当出现‘停’或者‘终’字样时退出循环，结束程序。

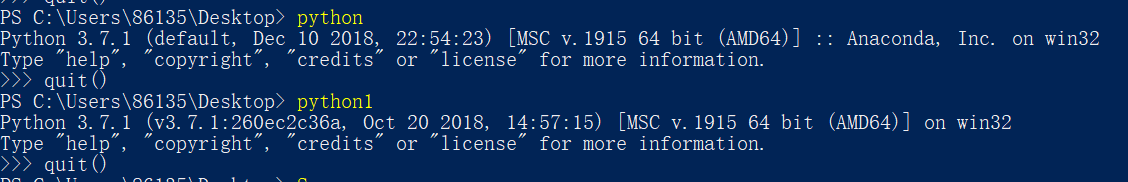


**1.4整合代码：**

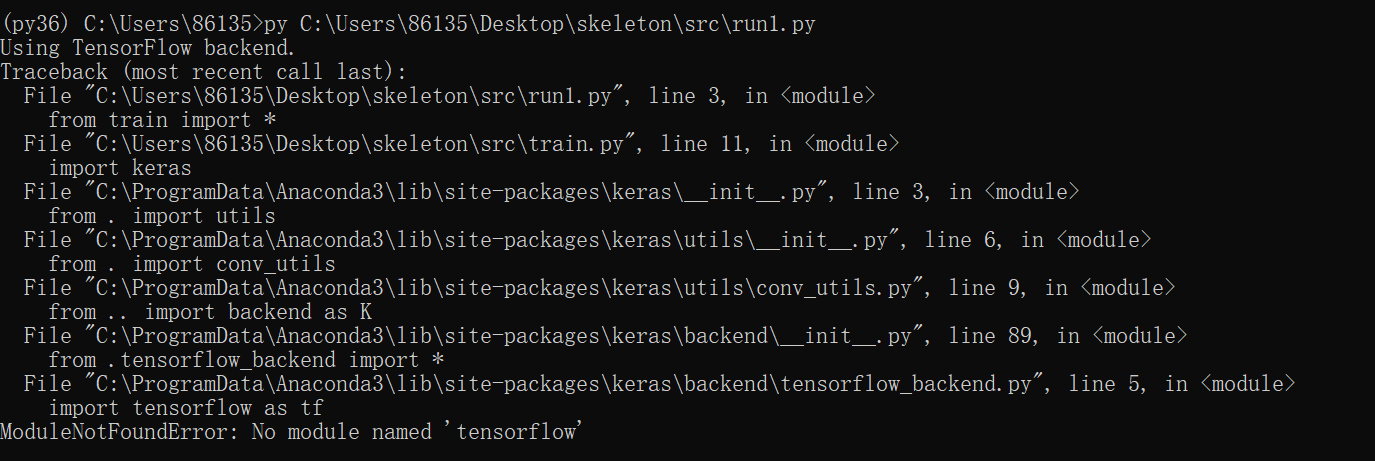
我们小组主要分为语音和图像两个模块，语音的算法比较简单，不需要TensorFlow环境，只要pip相应的包就可以，但是图像识别主要是在TensorFlow环境里运行。为了能够运行代码，我尝试安装对应的anaconda环境，安装TensorFlow和2.0.0版本的keras。

安装环境：

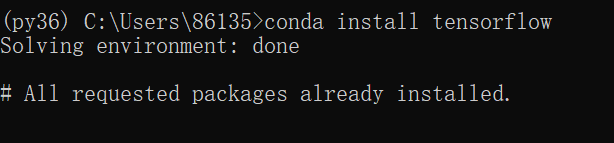
之前是在同学的Mac电脑上运行，在本机安装anaconda后，经过更改本地的设置：path路径，python.exe和pip.exe的重命名后，成功能区分出anaconda的python和电脑之前的python，因为之前的语音是直接在python上跑得。



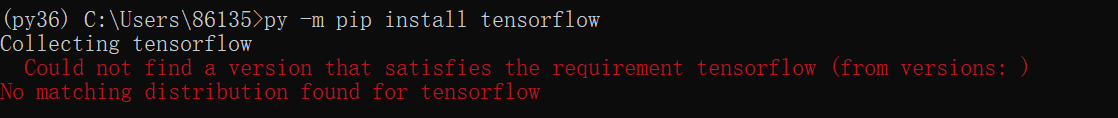
之后是按照同学要求安装好python3.6环境和TensorFlow后尝试安装2.0.0.版本报错最后安装了新版keras后，运行发现



可是我已经装过

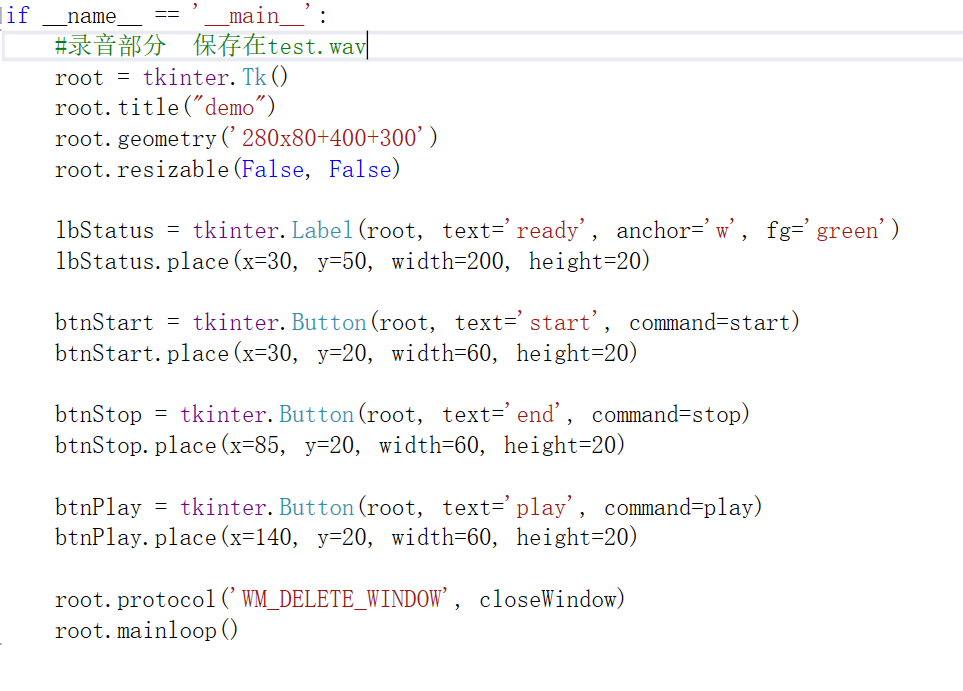


后来我发现它装到我以前的python里了，但是我用py -m pip install tensorflow又会报错

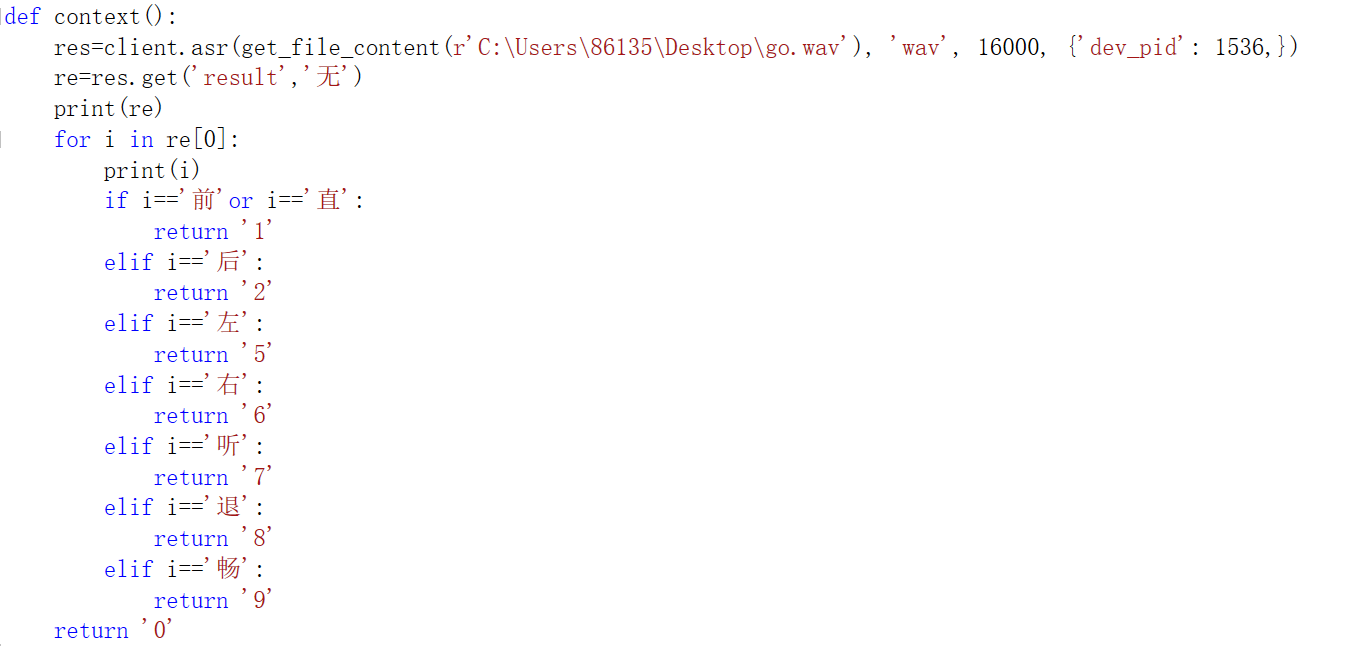


所以最后放弃了。

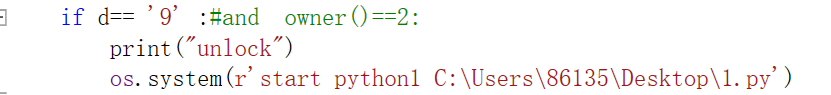
对于语音代码的整合，我利用的是os.system调用cmd，由于我有要开启非阻塞程序，在命令前增加了一个start，表示额外开启一个cmd运行py。语音代码的思路是有一个start主程序，它主要有实时录音、GMM声纹识别和语音识别功能。首先开启录音：



这是一个不断循环的录音模块，只有closeWindow能结束，GMM声纹识别和语音识别功能放在录音的record里，表示录一次音就执行一次，在录音模块里，录制完成后调用百度API识别语音，API会返回一个list，list[0]是识别出的文字，然后我进行逐字识别，当听到“畅”时返回9，听到“听”时返回7，其他部分暂不需要。



根据返回值来确定执行哪个模块，当返回9是，先进行声纹识别，我们目前有三个声纹，在这里我规定声纹2是主人，当声纹也返回2时，开启语音控制程序。



当context()函数返回7时，开启背景音识别的程序。

**二、课程收获**

这门课简要讲述当下主流的人工智能的一些算法以及如何调用相关算法的python包来实现它，比较有趣味的是要自己搭一个小车写代码来跑训练出来的代码。在这门课中，我了解了一些人工智能算法：卷积神经网络、博弈论、决策树等等。在小组分工里，我主要是扮演一个整合调试的角色，把小组成员在不同平台不同电脑系统的python程序在我的电脑上完美运行起来，所以在码代码方面我做的不是很深入，只做了粗浅的GMM算法。在完成大作业过程中我意识到端口的链接并没有我想象中的那么简单，蓝牙和电脑的端口链接我先是在mac尝试，发现macOS不提供虚拟串口，所以又转战win10，链接控制也出了很多问题，明明就几行代码却一直调不通。在电脑上装pyaudio时，因为我在win10下的是最新版本3.7导致很多包没有提供相应的pip接口，也调了好久，这告诉我以后不能安装太新的版本，基础设备还没有更上换新的速度呢。但让我最头痛的是anaconda和TensorFlow，这也是我现在还没有搞定的地方。我因为以前装了python，不想卸载，导致pip命令常常装错地方。比如说我在anaconda环境下用coda install装了TensorFlow，但却装在了我以前的python里，我想用py -m pip install又报：

Could not find a version that satisfies the requirement tensorflow (from versions: )

No matching distribution found for tensorflow

让人绝望。

最后我想提一个建议：希望老师能把上课时间调的稍微晚一点，有时候起不来床，导致上课迟到，对不起老师了。