一、项目介绍

该项目是通过对已有音频的训练，构造出较好的神经网络模型，以实现对其他音频数据类别的精准判断。整体思路是读取音频数据后，先对音频文件进行一系列处理，再提取音频的梅尔频谱特征，并以此为基础数据构造并训练卷积神经网络模型，该模型可以对音频数据进行较为精准的分类。

二、项目结构

```
├─data                    # 数据集
├─Model                   # 保存的模型
├─AudioClassify.ipynb     # 主要程序，负责模型的训练
├─model_predict.py        # 供用户调用的语音分类程序
├─dataset.csv             # 数据集的路径、类别信息
├─test_set_result.csv     # 测试结果
└─requirements.txt        # 项目第三方库要求
```

如上图所示，负责数据读取、预处理和训练的程序位于文件`AudioClassify.ipynb`中，该文件是一个`jupyter notebook`文件，可以通过`jupyter notebook`或`jupyterlab`等程序打开，选择使用该文件，是为了使每一步的得到的结果都清晰的呈现。

用户测试程序为`model_predict.py`，该程序是一个`python`脚本文件，里面包含了一个预测类，可以加载训练好的模型，以实现对用户指定语音文件的种类进行精准判别，用户可以指定一个音频文件，也可以指定一个包含众多音频文件的文件夹。

三、使用的工具和库

开发语言：`python3`

开发工具：`VSCode`、`jupyterlab`

第三方库：`tensorflow`、`librosa`、`sklearn`等，详见`requirements.txt`文件。

三、项目的详细思路

1、音频读取

使用`python`第三方库`librosa`对音频数据进行读取，音频数据包含两个关键信息，分别是音频时间序列和采样率。为最大程度读取音频信息，读取时采用了音频本身的采样率`44.100KHz`。

2、音频标注

音频数据分类存放在以类名命名的文件夹中，故读取音频数据时利用文件夹名对音频进行标注，免去了人工标注的麻烦。为方便计算机处理，对音频类别进行了编号，共10类音频，采用0-9编号来标注类别。读取音频文件后，将文件路径和标注好的文件类别写入到`dataset.csv`文件中，为以后的使用提供方便。

3、音频预处理

- 读取音频数据时发现样本音频时长基本都是5s，有少数音频时长不是5s，故需对音频时长进行处理。对超过5s的音频进行裁剪，对少于5s的音频进行随机填充，使音频的时长都为5s。
- 通过`librosa`读取的音频时间序列是一个一维数组，反映音频频率和时间的关系，此为原始数据，包含较多无用信息，且维度较大，不能够作为判别音频种类的标准。查阅资料可知，`梅尔频谱`、`mfcc`、`gfcc`和`Fbank`等特征均能够反映音频的某些特性，可以作为音频分类的标准。考虑到提取难度和识别效果，最终选用`梅尔频谱`作为分类标准。音频`梅尔频谱`的提取，要经过傅里叶转换、分帧、加窗和离散余弦变化等一系列操作，但`librosa`库中有已实现的提取方法`melspectrogram`，使用该方法提取大大降低了处理难度。音频数据经过提取后，维度从几万维降至了128维，将会大大减少计算复杂度。
- 通过前几步的预处理，得到了时长固定为5s的音频数据，并获取到音频数据的种类标签和`梅尔频谱`，利用`pandas`库`DataFrame`类型的变量存储了这些数据，以便后续处理。

4、划分数据集

对数据进行一系列处理后得到了可以用于训练的数据，但不能够全部用于训练，还需保留部分数据用于模型的预测，以此评判模型的预测效果。使用`sklearn`库的`train_test_split`方法对数据集进行划分，划分后的训练集和测试集之比为3:1，数据共有390条，故训练集为292条，测试集为98条。

5、模型构建与训练

- 音频的`梅尔频谱`为一个128维的数组，为便于卷积运算，将其变形为一个16x8的矩阵。种类标签为十进制数字，为方便训练，使用`tensorflow`中的`to_categorical`方法将其转换为了`one-hot`编码。
- 使用`tensorflow`构建卷积神经网络模型，该模型包含了两个卷积层和两个池化层，其中卷积层用来提取特征，最大池化层用于选择数据。两个卷积层均使用了3x3的卷积核，不同的是，第一个卷积层使用了64个卷积核，第二个卷积层使用了128个卷积核。卷积层使用的激活函数为`tanh`,填充方式为`same`，池化层采用了最大池化方式。
- 为防止训练过程中的过拟合，还会在两次卷积池化之后舍去一些节点，舍去的节点占总结点的一成。该模型的输入是一条音频数据的梅尔频谱，输出是该条音频数据是十个类的概率。通过不断地训练，一步步调整模型内部的权值，以构建较为精确的模型。通过参数设置，使模型训练50轮，每一次均输出精确率和损失，通过观察，该模型对训练集和测试集的准确率均不断上升，损失均逐渐减少，模型符合预期。

6、模型的判断与预测

训练完成后，利用测试集对模型进行检验，准确率达到75.51%。

7、模型的保存

使用`tensorflow`自带的相关函数对训练好的模型进行保存，即将模型以文件的形式保存到磁盘上，当下次需要进行音频分类时，可以直接加载模型进行判断，免去了训练模型的麻烦。

8、设计测试类供其他人测试

创建一个`python`类，用于加载保存的模型，并向用户提供相关的接口，用户只需要指定音频数据的路径，便可以得到相关音频数据所属的类别。