**《人工智能程序设计》课程设计报告**

**作品名称：塞壬的歌声**

**小组成员**

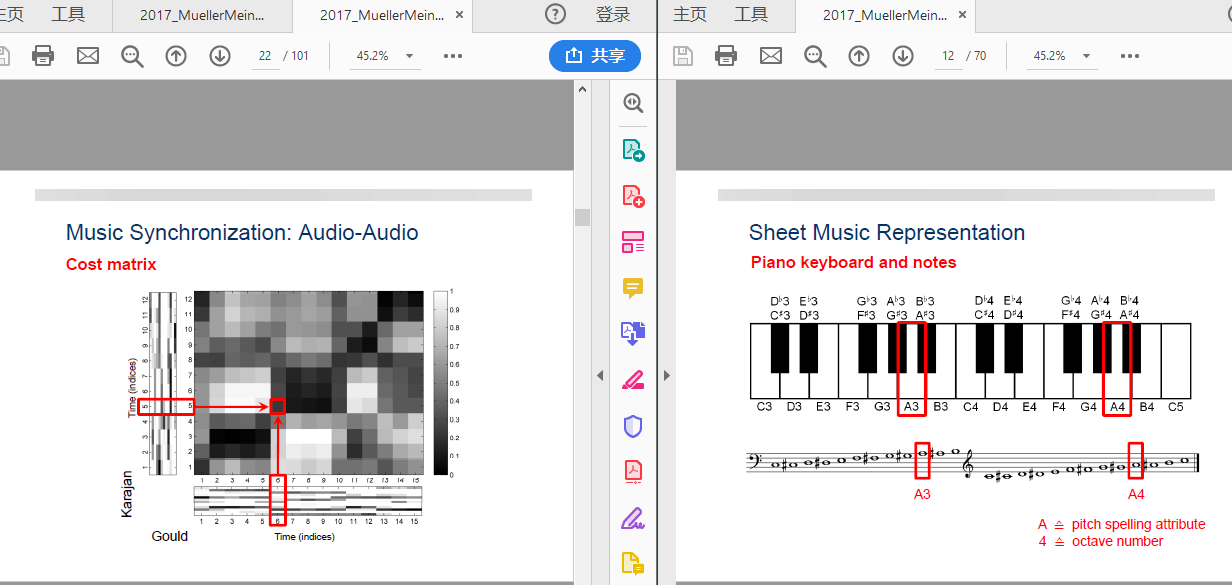
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学号** | **姓名** | **院系** | **手机** | **邮箱** | **在项目小组中承担的任务（简写）** |
| 181220010 | 丁豪 | AI | 15189780512 | 181220010@smail.nju.edu.cn | 选题、生成音频特征数据集算法、训练集音乐获取 |
| 181220013 | 淦海翔 | AI | 18170245542 | [2735314734@qq.com](mailto:2735314734@qq.com) | 探索分类算法、最终包装、测试集音乐选择 |
| 181220056 | 王宸旭 | AI | 18851128695 | 181220056@smail.nju.edu.cn | 研究音频特征文档、TensorFlow深度学习探索、尝试搭建GUI界面 |

**背景和问题**

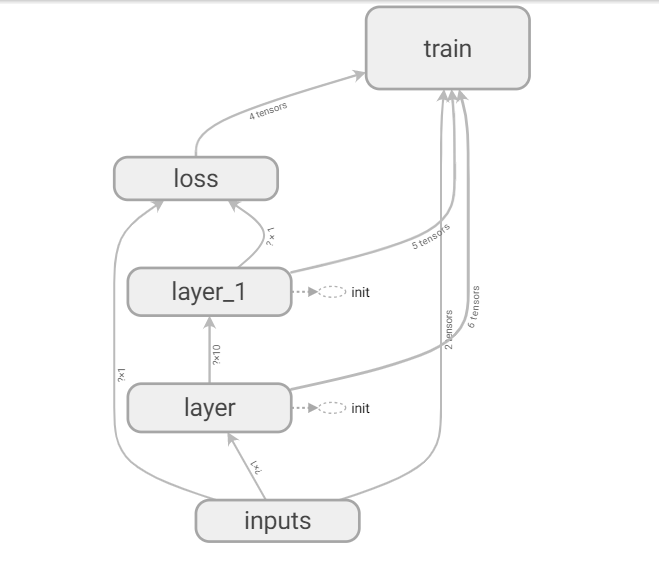
|  |
| --- |
| “听歌识曲”作为广大音乐软件用户耳熟能详的功能早已不再新鲜，当我们在某时某地听到一首陌生而动听的音乐，往往会拿出手机打开听歌识曲来抓住这瞬间的感动。然而，目前做音乐识别的大厂如Shazam、Facebook、Youtube、Netease163皆在使用**音乐指纹技术**，该算法需要依赖大量已经存储在云端的音乐指纹相关信息进行精确匹配。，这就导致一旦离线，现有的这种听歌识曲功能将绝无可能实现。  另一方面，当我们使用音乐软件时，一定体验过其中的“分类曲库”功能，顺着一层层索引向下，便能获得我们想要的音乐（有些类似于已经灭绝的索引试搜索引擎）。另一些软件虽然并不存在如此硬性的歌曲分类，却也可以通过搜索关键词来得到相应分类的网友自建歌单（如“网易云音乐”）。这些歌单的准确性之所以如此之高，一方面得益于最初创建者高超的音乐素养和细致认真的工匠精神，另一方面，音乐本身清晰的界限（如钢琴、小提琴）也十分重要。  基于以上背景与资源，我们设想：能否利用音乐本身存在的特征差异，将音乐分类的过程自动化，利用计算机强大的计算能力，快速准确地实现大量位置音频的分类功能。这种功能基于音乐本身的属性，一旦模型训练成型，将可以摆脱网络限制，直接在本地实现音乐分类过程。 |

**设计思路**

|  |
| --- |
| 为了实现功能，我们将程序细分为以下几个步骤：   1. 提取音频文件的特征值 2. 获得数据集 3. 对获得的特征值数据进行预处理 4. 根据处理后的数据，将音频文件进行分类 5. 根据分类结果，将文件移动至其种类对应的文件夹   以下将分别叙述我们在实践之前对于每一个步骤的想法。  1.提取特征  在思考如何对音频特征进行提取识别时，我们下载了德国Meinard Müller教授的Fundamentals of Music Processing课程课件学习相关理论。该课程的音频处理基本方法是对音频的各种特征图（如频率——时间图）搭建神经网络进行学习。由组员王宸旭初步模仿实现后，正确率可达到83%以上，可是我们面临三个问题：1、搭建神经网络时我们使用了Tensorflow包，无法完全手写复现核心代码。2、神经网络在对一首歌百万数量级的特征进行迭代处理时效率极低（受限于我们的学习方法与机器配置），不利于本地实现。3、基于该课程所做的理论实践要求操作人具有一定的乐理知识，本小组成员均缺乏足够的专业素养，在权衡时间后觉得现学乐理会拖沓整体进度。综上我们放弃该思路，转向探索其他方法。  上网查资料，发现python可用的音频处理库中有一些十分强大的音频处理库比如librosa、Pyaudio，利用这些库对wav或者mp3文件进行解析（使用内部的一些特殊处理函数），可以得到诸如一首音乐的“梅尔频率倒谱系数MFCC”、“频谱质心spectral centroid”、“过零率zero-crossing rate”等重要参数。对这些参数值建立歌曲对应列表，或许可以使用k-nn或者逻辑斯蒂模型进行学习。这样一来，我们的任务就转变成了对已经经过处理的数据列表的分析，而不再需要反复对原始音乐进行调包处理，大大降低了任务的时间复杂度。  2.获取数据集  首先需要在音乐软件中相应分类下下载足够数量的音乐文件，然后利用wave包和librosa包对每一首音乐进行解析，将各特征值作为列，建立每一首音乐与其对应特征值的dataFram，之后存储在csv文件中备用。  3.数据预处理  根据我们所学习的几种分类算法（如knn，逻辑斯蒂回归），我们期望得到每一条的数据应该是一个series，每一个index对应一种特征值，且这个值最好是一个确定的数。数据集则应该是由若干条这样的数据组成的dataframe，如果特征值的数量过大，最终可能会采取pca降维的算法提高计算效率。  此外，出于解决过拟合的考虑，在使用数据前加上标准化处理。  4.音频分类  我们最初想到的是利用本学习学习的knn分类算法和逻辑斯蒂回归算法一同对音频文件进行分类，最初的想法就是对两者的分类结果进行加权，从而达到综合判断的效果。但是线性加权对于两种判断方法并无意义。最终，我们决定再使用一种分类算法（如支持向量机），再对三者进行加权。  5.文件移动  为了将文件进行移动，只需对应每一个种类创建一个空的文件夹，再对每一个音频文件依据其类别进行移动。以上内容调用os库即可实现。 |



（Music Processing相关课程课件：核心思路是将音频识别转换为图像识别）



(初步实现的神经网络可视化)

**数据来源**

|  |
| --- |
| 音乐来源：网易云音乐网友自建歌单  相关特征使用方法：Khan学院及Youtube上的Music Processing视频介绍、文档。 |

**处理分析**

|  |
| --- |
| 1. 获取音乐   打开网易云，搜索想要的分类，点击歌单，依次选中歌单，点击全部下载。因为某些资源非会员无法下载，实际上需要寻找更多音乐。最终确定的的数据集是每一类500首左右。   1. 特征提取   如设计思路里所提到的，我们选择调用librosa包来提取音频文件的特征。为了提高分类的准确率，了解每一种特征对于分类起到的作用是十分重要的，于是我们尝试了解学习了相关理论知识，但受限于准备时间和该方面知识的难度（主要），不得不放弃了。最终结果是调用librosa中所有的特征值函数，多维数据按多列计。  根据最初的思路，我们希望librosa的每一个特征提取函数应该实现的效果是：传入一个音频文件，返回一个值。在实践的过程中我们发现，有许多特征函数返回的是一个ndarray，其每一列代表音频中的一小段时间，每一行代表一种子特征。这个问题即是我们在数据处理中遇到的最大问题。  我们尝试了四种方法。第一，经过搜索，我们了解到了dtw算法能够计算不同时长的两个音频之间的“距离”（类似于knn算法中的欧几里得距离），因此可以在knn算法中使用，解决这个问题。这就要求在存储数据集时，需要将一个ndarray存进csv文件的一个单元格中。由于我们未能成功实现ndarray的存入与读取，这种方法便失败了。第二，将各种子特征对时间取平均值和方差作为特征，并直接作为音频的一种特征。第三，将一种特征的子特征对时间取平均值和方差之后，再对所有子特征取平均值，作为音频的一种特征。第二、第三种方法的处理手段比较初级，我们又缺乏对于各种特征的了解，得到的结果应该并不太精确，但我们仍然予以尝试。第四，在第二种的数据基础上进行降维处理。具体采用哪一种方法，我们选择根据分类结果评判。   1. 数据可视化   使用sklearn中的SVD对数据进行降维，分别对降到2维和3维的数据绘图，绘图使用的是plotly库。  D:\OneDrive - smail.nju.edu.cn\desktop\5@~0C9DM6}2N(7Z3~OBOHFA.pngD:\OneDrive - smail.nju.edu.cn\desktop\0`@SJP5X0}8%BQTY`3A10LP.pngD:\OneDrive - smail.nju.edu.cn\desktop\`Y$[]}UNQ~[`RRR4EFR]$%1.png  从图像上我们不难看出，有一些分类之间存在明显的聚类和类间差异，而另外一些分类则相对不明显，甚至基本混作一团。   1. 音频分类   最初进行尝试的是knn算法。对于数据预处理中第三种和第四种方法，knn的判断准确率并不高，为了提高准确率，我们直接选择了使用第二种方法。这次knn的准确率达到65%，仍难以令人满意。但是第二种方法产生的数据的特征值有900种，这使得knn的计算量急剧增大，分类过程进行得过慢，再加上手写knn中忽略了对数据的正则化处理，我们开始考虑使用逻辑斯蒂算法。经过尝试我们发现逻辑斯蒂得到结果花的时间很少，且准确率将近80%。为了实现综合判断，还需要使用一种分类手段，于是我们选择尝试支持向量机。在网络上进行粗略的了解之后，我们调包实现了支持向量机，得到的结果十分不理想，这可能是由于数据的处理方法初级且没有考虑各种特种的实际意义。总之，由于支持向量机失败，综合判断的想法也不得不舍弃，最终分类的任务将由逻辑斯蒂回归承担。   1. 文件移动   文件移动方面较为简单，在网络上学习之后，调用os库便实现了功能 |

**结论**

|  |
| --- |
| 最终的全音乐测试成功率如下：  钢琴 识别率: 0.8928344958782498 吉他 识别率: 0.9319546364242829 小提琴 识别率: 0.7853107344632768 电音 识别率: 0.7658979734451432 萨克斯 识别率: 0.7342192691029901 vocal 识别率: 0.7056105610561056 rap 识别率: 0.8496868475991649 长笛 识别率: 0.7707792207792208  由于没有什么音频信号处理基础，使用简单粗暴的方法得到这样的正确率已经相当令人满意。不过数据集的选取主要是不同乐器，已经有在刻意拉大不同音乐之间的特征差异，对于正确率应该有较大帮助。下一步可以拓展的方向，将是更多更加细化的音乐流派与风格分类。  为了使我们的分类模型更加有实用意义，最终展示程序设计成了自动整理音乐文件夹。它的功能是传入待整理的文件夹地址，运行程序后将在其下新建曲风文件夹，并将每一首音乐转存到相应文件夹下。 |

**其他想说的（可选）**

|  |
| --- |
| 在分类问题中，我们使用了手写的KNN算法对处理好的数据进行加工，由于手写包与Sklearn包中的KNN算法相比缺少正则化以及一些数据处理过程，分类准确率与分类速度都不如包中的算法。在多次实验中我们尝试使用Logistic算法对进行分类，基础的Logistic只适用于二分类问题，若想进行本课题中的多分类，需要对算法进行改进得到Softmax算法，我们尝试复现Softmax算法，但因为没有系统概率论与数理统计而不得要领，遂放弃该尝试。组员淦海翔手写复现支持向量机算法进行分类，识别准确率不如KNN，最终决定使用KNN与Sklearn包中的Logistic算法进行分类。 |