1. **项目问题描述**

**问题内容**：制造一个能初步解决语音识别问题的一个简单模型

**意义**：与机器进行语音交流，让机器明白你说什么，这是人们长期以来梦寐以求的事情。中国物联网校企联盟形象得把语音识别比作为“机器的听觉系统”。语音识别技术就是让机器通过识别和理解过程把语音信号转变为相应的文本或命令的高技术。语音识别技术主要包括特征提取技术、模式匹配准则及模型训练技术三个方面。

**目标：**我的内容将以特征提取，即挖掘声波中的隐含信息为主要目标，对声音中的音素（phoneme）进行分类

1. **形式化方法**

训练模型使用的语音数据只有英文，所以将所有的英文发音分为39类音素(phoneme)，即给定一个发音，目标是输出它对应的音素种类，或者给定一句话，输出这句话所包含的全部音素，通过音素的分类初步解决语音识别的问题，举例如下所示。

Machine Learning → M AH SH IH N L ER N IH NG

Machine Learning的发音可以拆解成10个音素，这些音素代表了这个词的发音特征

1. **数据集**

**数据类型以及预处理方法**：原始数据集是人类说话的发音，使用声学信号的特征提取方法MFCC，对原始声音进行特征提取和分割，目标是被分割后的每个数据段恰好对应一个音素，因此数据集的形式为

train\_11.npy 是一系列数据段

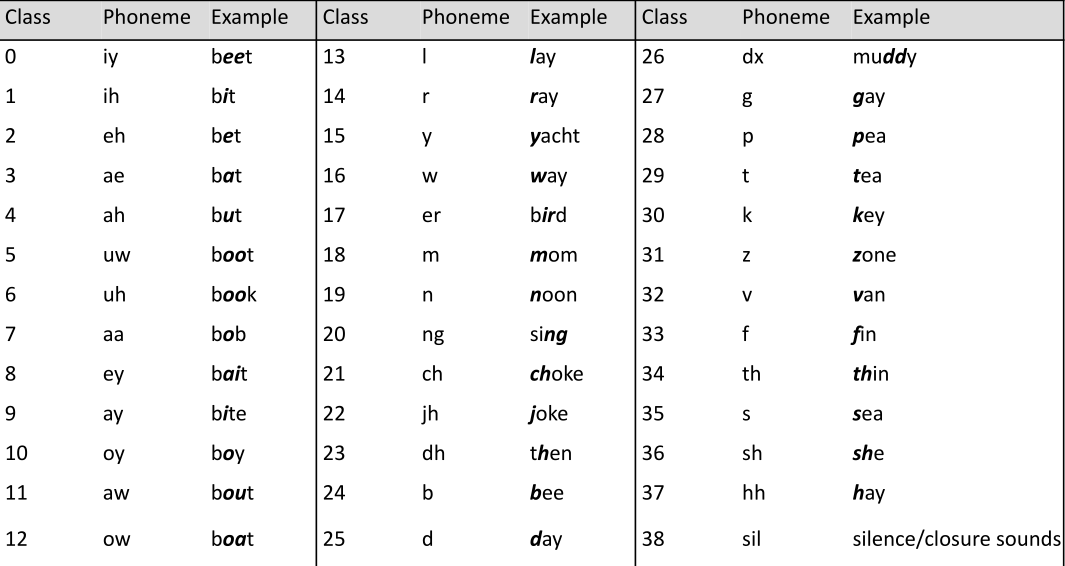
train\_label\_11.npy 是这每个数据段对应的音素标签

**数据来源**：声音数据来自TIMIT Acoustic-Phonetic连续语音语料库，由其他团队对TIMIT数据进行筛选并使用MFCC分割后的数据被发布在Google Colab上，网址如下：

<https://drive.google.com/uc?id=1HPkcmQmFGu-3OknddKIa5dNDsR05lIQR>

**注**：因为音素一共有39类，所以该任务可以是看作由train\_11.npy中的数据段分类到其对应39类音素的多分类任务，给每个音素一个对应标签0—38.

如图所示（图片来源于数据集注释）：



1. **算法实现**

**方法一：神经网络方法**

使用一个四层的神经网络，其中最后一层一共39个神经元，对应39个类别，因此，每一个类别对应到39个神经元，都可以用独热向量的表示方法，比如类别0就可以表示成（1，0，0…共38个0…0，0），同理类别1为（0，1，0，0…共37个0…0，0）

之后就可以用训练神经网络的方法寻找一个函数f，使得f（数据段）与类别向量间的loss最小，其中loss函数可以使用交叉熵，此前先使用Softmax进行归一化处理，loss的optimization使用最简单的梯度下降法，具体实现细节参见代码文档，代码及其注释。

**方法二：支持向量机**

可以使用OVA（OVR）的方法构建39个二分类器，即第0类和剩余38个，第1类和剩余38个，第2类和剩余38个……，使用投票的方法，其中正例置信度最高的一类作为最终预测的分类，具体实现细节参见代码文档，代码及其注释。

1. **评估方法**

将数据集分为训练集和测试集，因为原始任务可能没有查准率的查全率的权重要求，因此我可以直接将其中测试集里的分类正确率作为算法的评估方法

1. **代码运行环境**

Python 3.9.7

Pytorch 1.9.0+cu111

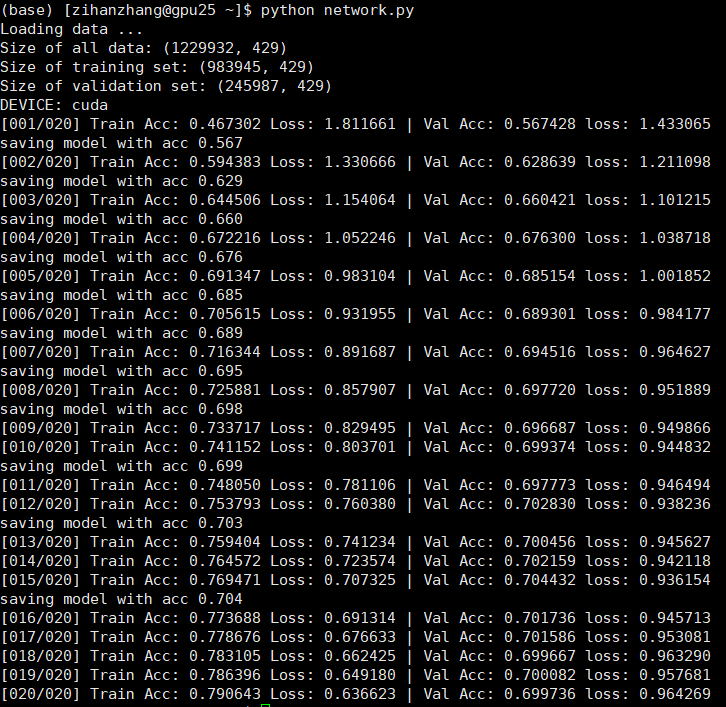
GTX3060Laptop NVIDIA CUDA 11.4.94 driver

其他引用的包详见代码

1. **结果**

因为这是一个39分类的多分类任务，随机选择的正确率为**2.6%**，以此为baseline评测算法表现

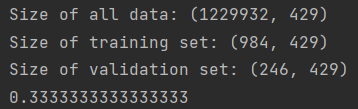
1. **神经网络方法：**如图所示，所有epoch中最好的模型正确率为**70.4%**



1. **支持向量机方法：**

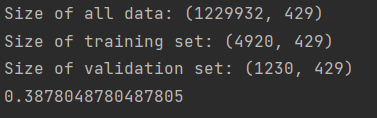
不同于神经网络的计算方式，因为984000 x 429 x 39维数的支持向量机需要耗费巨大的算力，为了简化计算，表达概念，我对数据集进行了裁剪，同时，为了考虑到数据集的平衡性，我采用了均匀采样的办法，对原始数据集进行采样，因为对特征降维（429维）需要额外的计算，而且本来的数据集条目数量就非常多，所以我没有削减数据维度，而是减少了数据条目

使用1000：1的均匀采样，用984 x 429 x 39的数据来计算精确度：



得到了仅仅**33.3%**的准确率，有可能是数据集削减量太大

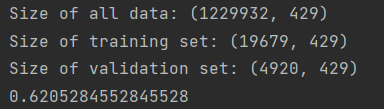
为了提高准确率再次实验使用200：1的均匀采样，用4920 x 429 x 39的数据来计算准确率，最后得到了**38.8%**的准确率



由此可以得出结论，线性核的SVM不适用于此类任务，我又换用了其他核函数，得到了如下结果

|  |  |  |
| --- | --- | --- |
|  | 1000：1抽样 | 200：1抽样 |
| 高斯核 | 46.3% | **53.4%（表现最好）** |
| Sigmoid核 | 34.9% | 33.5% |
| 多项式核 | 37.4% | 51.4% |

因此，高斯核可能更加适合音素识别任务，我又换用了50：1的抽样，此时的SVM和4层的神经网络在算力相同时，所需时间接近，得到了**62.05%**的正确率，因此多SVM投票的方式在音素识别的任务上表现较神经网络略差一些。



1. **备注**

1.项目文件包含数据集，由于数据集是npy文件，所以压缩后只有300M左右，但解压后4G多

2.关于项目文件的描述：

<datamining> 包含了代码和数据集，该文件夹是一个整个的pycharm项目文件，其中的python文件当然也可以直接打开运行

<network.py>是使用了神经网络方法的代码

<model.ckpt>是神经网络训练出来的效果最好的模型

<svm.py>是使用了支持向量方法的代码

此外，由于这是一个多分类任务，因此数据集包含特征和标签两部分

<train\_label\_11.npy>是数据集的标签，取值范围是0到38之间的整数数字

<train\_11.npy>是特征部分，每一条数据包含429个特征