復旦大學

作业题目：孤立词语音识别系统

院　　系：计算机科学技术学院

专　　业：计算机科学与技术

姓　　名：陈琳

学　　号：16307130343

指导教师：薛向阳

2019 年 6 月 24 日

目录

1 背景及任务介绍

语音识别（speech recognition）技术，也被称为自动语音识别，语音识别是以语音为研究对象，通过语音信号处理和模式识别让机器自动识别和理解人类口述的语言。语音识别技术所涉及的领域包括：信号处理、模式识别、概率论和信息论、发声机理和听觉机理、人工智能等等。

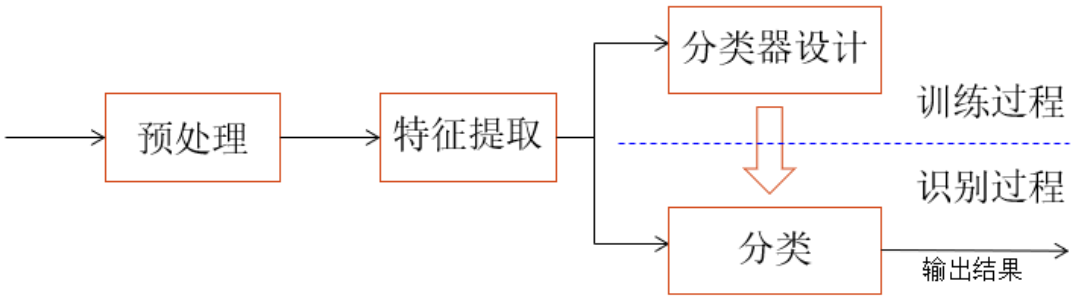
目前具有代表性的语音识别方法主要有动态时间规整技术(DTW)、隐马尔可夫模型(HMM)、矢量量化(VQ)、人工神经网络(ANN)、支持向量机(SVM)等方法。

动态时间规整算法基于动态规划的思想，将语音测试信号和语音参考模板进行比较，按照某种距离测度得出两者之间的相似度，选择最佳路径进行规整，解决了发音长短不一的模板匹配问题。隐马尔可夫模型是由马尔可夫链演变来的，是基于参数模型的统计识别方法。其识别过程中运用待识别语音序列与 HMM 参数之间的似然概率达到最大值所对应的最佳状态序列作为识别输出。矢量量化是一种重要的信号压缩方法，主要适用于小词汇量、孤立词的语音识别中。其过程是将语音的若干个特征组成一个矢量，把矢量空间分成若干个小区域，每个小区域寻找一个代表矢量，量化时落入小区域的矢量就用这个代表矢量代替。对孤立词识别任务，为每一个词训练建立一个码书，M 个孤立词就可以获得 M 个码书。在识别时，一个未知的语音向量序列，分别用 M 个码书进行量化，可以计算出 M 个平均失真。取最小平均失真的相应码书对应的孤立词即为识别结果。人工神经网络是 20 世纪 80 年代末期提出的一种新的语音识别方法。其方法是模拟人脑思维机制的工程模型，具有自适应性、并行性、鲁棒性、容错性和学习特性，其强大的分类能力和输入—输出映射能力在语音识别中都很有吸引力。支持向量机是应用统计学理论的一种新的学习机模型，采用结构风险最小化原理，有效克服了传统经验风险最小化方法的缺点。兼顾训练误差和泛化能力，在解决小样本、非线性及高维模式识别方面有许多优越的性能，已经被广泛地应用到模式识别领域。

在本课程项目中，我们需要实现的是孤立词语音识别系统。针对 20 个关键词：数字 语音 语言 识别 中国 总工 北京 背景 上海 商行 复旦 饭店 Speech Speaker Signal Process Print Open Close Project，采集了所有参与课程的同学朗读每个词 20 遍的语音，以此为数据集来构建一个能正确识别这 20 个关键词的孤立词识别系统。

2 识别算法

语音识别系统本质上是一种模式识别系统，我的系统基本结构如下图所示：



接下来我将按照上图的流程来介绍我的孤立词语音识别系统。

2.1 预处理

先解压所有压缩包，然后使用深度优先搜索来遍历所有文件夹，根据文件的命名(学号-第几个词-第几次录音.wav)，通过正则表达式匹配把符合要求的文件提取出来，按照 speech_data/学号/文件名.wav 的格式统一存储。为了效率的考虑，我使用了多线程来完成这一任务。

针对异常数据，我采用随机替换的方式，用同一个同学在同一个词下的另一个语音文件进行替换。这样就可以缓解数据缺失带来的样本不均衡的问题。

2.2 特征提取

2.2.1 归一化

由于声音在采集时强度不一致，先对每一段音频数据进行了归一化处理。

2.2.2 预加重

预加重的是为了消除发声过程中声带和嘴唇的效应，来补偿语音信号受到发音系统所压抑的高频部分。我们听到的声音，其实是类似经过一个低通滤波器，信号做了高频衰减，故语音信号的的高频成分被压抑了。加一个高频增强的高通滤波器，就可以补偿这个高频衰减。预加重通常采用一个一阶的线性高通滤波器

$$\mathrm{y(t)} = x(t) - \alpha x(t-1)$$

来实现，其中系数α介于 0.9 和 1.0 之间。在代码实现上，只需要对数组进行加权差分即可。

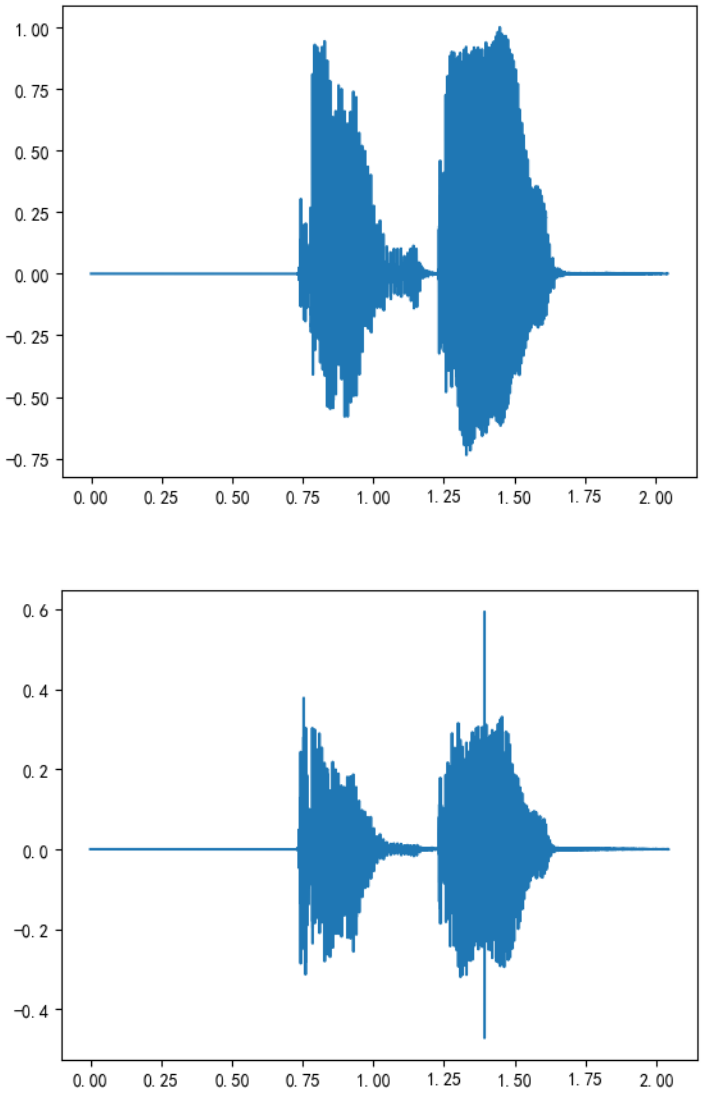


图 1：归一化但未预加重（上）与归一化且预加重（下）

2.2.3 分帧

分帧就是将语音分割成段进行处理，每一段称为一“帧”。

在一般情况下，语音信号是一种典型的非平稳信号，但是我们可以认为说话人的语音变换并不是突然的，因此可假定语音信号在短时间（10-30ms）内是平稳的；故而在对语音信号进行分析时，我们可以将语音信号以 10-30ms 的间隔截取成一段一段进行分析。为了保证帧与帧之间的平滑过渡，我们设置帧移为 0 - 1/2 帧长，即相邻两帧之间会有 0 - 1/2 帧长的重叠部分。

分帧的实现是简单的，只需要使用 python 支持的数组切片（slice）方法即可方便地取出一帧对应的采样点。对于 16KHz 采样的语音信号，要使帧长在 10-30ms 之间，我将每帧的长度设置为 512 个采样点，帧移设置为 256 个采样点。

2.2.4 加窗

对于每一帧数据，由于我们在分帧时发生了非周期截断，在起始和结束时会出现不连续的情况，导致频谱在整个频带内发生了拖尾现象，称为泄漏。泄漏后的频谱的幅值更小，频谱拖尾更严重。当截断后的信号不为周期信号时，就会发生泄漏。而现实世界中，在做 FFT 分析时，很难保证截断的信号为周期信号，因此，泄漏不可避免。[1]

为此我们应该强调数据帧的中间部分，对边缘的数据进行弱化，以增加每帧数据左端和右端的连续性，使边界上的不连续现象弱化（参见图 2）。使信号呈现出周期信号的部分特征。

我使用汉明窗：

$$w(n) = 0.54 - 0.46\cos\left(2\pi\frac{n}{N-1}\right)，\ 0 \le n \le N-1$$

作为窗函数。要注意到的是，加窗只能减少泄露，不能消除泄露。

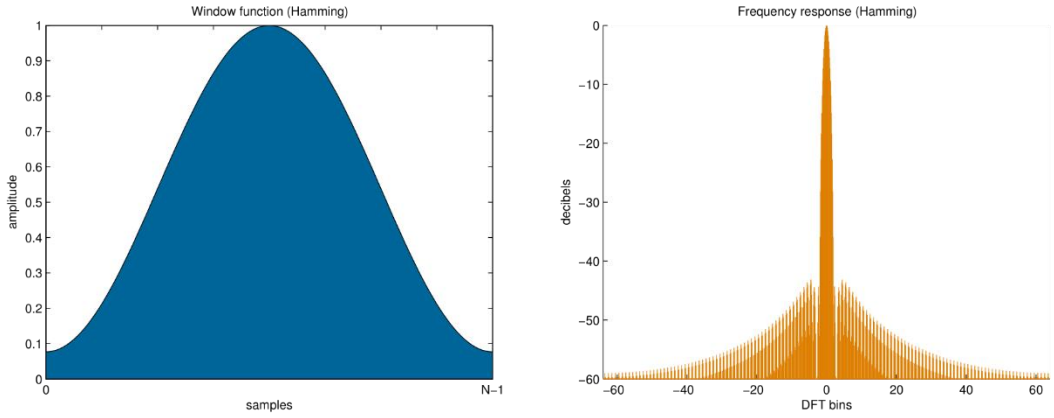


图 2：汉明窗函数

加窗其实就是将一帧内的采样数据与窗函数相乘，用y_t表示原始数据，用y_t'表示加窗后的数据，用 w[t]表示窗函数，则三者的关系为：

$$y_t' = y_t w[t]$$

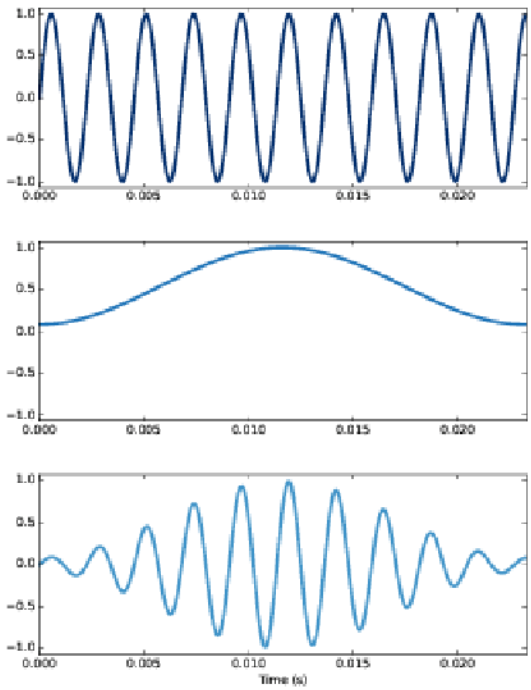


图 2：对数据使用窗函数的效果

2.2.5 端点检测

在录音过程中，难以避免背景噪声的存在，为此可以使用端点检测将真正有用的

语音信号分离出来。具体可以使用信号的短时能量和短时过零率。

用S_w表示加窗后的信号，用E_0表示能量，短时能量公式为：

$$E_0 = \sum_{n=0}^{N-1} S_w^2(n)$$

（假设窗的起点为 n = 0）

短时能量是刻画信号强度的一个重要指标，包含语音的信号强度往往会高于背景噪声的信号强度。通过设置一个阈值E_k，对于 $E_0 > E_k$的帧，判定其包含语音信号。但是，如果仅仅采用短时能量作为端点检测的指标，将很容易将清音信号与背景噪声一起忽略，导致语音不完整。为此引入短时过零率。

用S_w表示加窗后的信号，用Z_0表示过零率，短时过零率公式为：

$$Z_0 = \frac{1}{2}\sum_{n=0}^{N-1} |Sgn(S_w(n)) - Sgn(S_w(n-1))|$$

（假设窗的起点为 n = 0）

短时过零率就是帧内信号改变符号的次数。

因为无法检测出来的清音一般出现在单词的开始或者结束，所以我们采用以短时能量为主，短时过零率为辅的检测策略。设置一个阈值E_k，找到 $E_0 > E_k$的第一个点和最后一个点，设为N_1和N_2。N_1和N_2之间是语音段，但精确的起点和终点还得在N_1之前和N_2之后查找。设置另一个阈值Z_k，在N_1之前找，找到第一个大于这个阈值的点作为最终确定的起点，在N_2之后找，找到第一个大于阈值的点作为最终确定的终点。

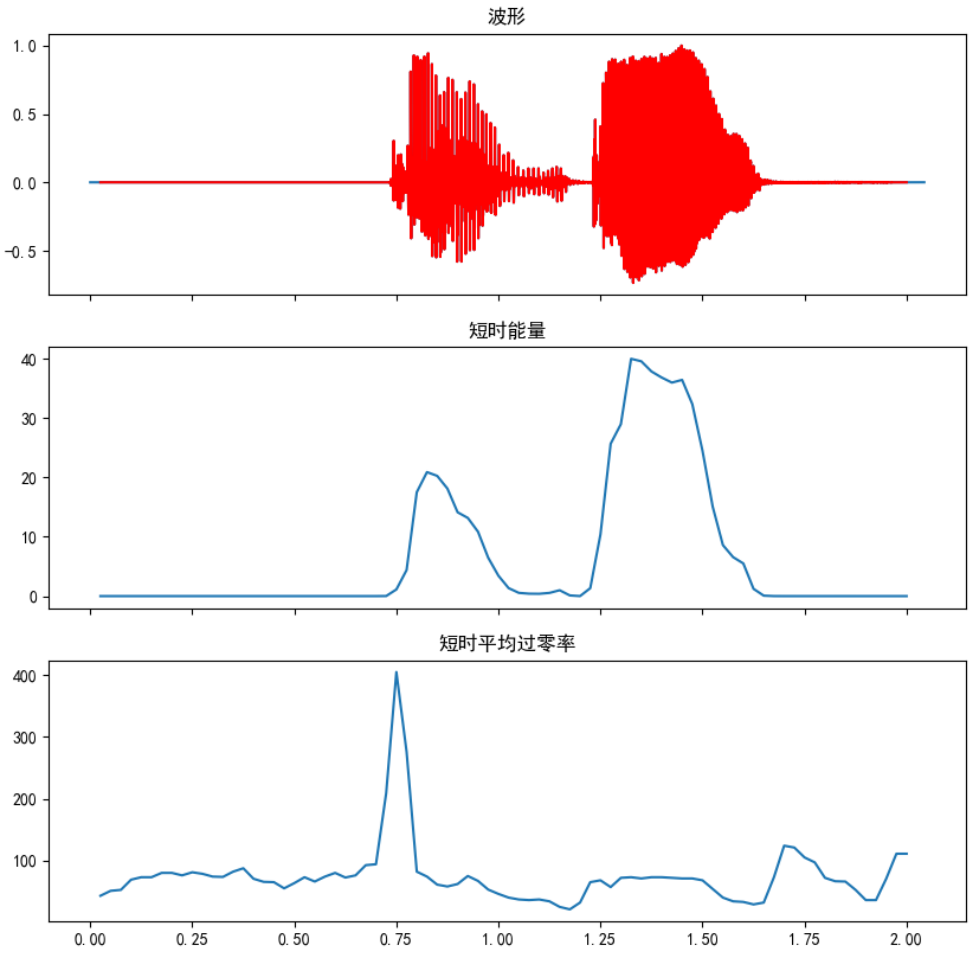


图 3：对“总工”的某个录音绘图

从图中可以看出，在短时能量第一个峰的起点左侧，还有一个短时过零率的峰，可以看出结合短时能量与短时过零率进行端点检测的必要性。

2.2.6 快速傅里叶变换

为了得到每一帧的频域特征，需要对每一帧进行傅里叶变换，这种分帧后进行的傅里叶变换又称为短时傅里叶变换，可以提供时间域与频率域的联合分布信息，以便描述出信号频率随时间的变化。

$$s(m,n) = x(n)\, w\left(n - m\frac{N}{2}\right) \quad 加窗$$

$$S(m,\omega) = DTFT\left\{x(n)\, w\left(n - m\frac{N}{2}\right)\right\} \quad 变换$$

在快速傅里叶变换中，按照下标的奇偶性将待变换的信号序列拆成两半，递归地进行计算，再从递归计算出的两个子序列的结果中得到整个序列的离散傅里叶变换。该算法时间复杂度为 $\mathrm{T(n)} = 2T\left(\frac{n}{2}\right) + \theta(n)$。根据主定理，该算法的时间复杂度为O(nlogn)。

在傅里叶变换完成后对结果进行平方，得到能量谱。

2.2.7 梅尔频率域特征

人耳的听觉对频率有选择性，听到的声音高低与声音频率不成线性关系，而是与声音频率的对数成近似线性关系：

$$\mathrm{B(f)} = 2595\ \lg(1 + \frac{f}{700})$$

因此人的听觉不能有效分辨所有频率分量，即所谓的屏蔽效应，只有当两个频率分量相差一定带宽时，人耳才能区分它们。为此我们需要构造一个梅尔滤波器组，使每一个滤波器的中心频率在梅尔频率域中呈线性分布，每一个滤波器的中心频率计算如下 ：

$$\mathrm{f(m)} = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\right)$$

N：窗口宽度；F_s：采样频率；M 是滤波器的数量

f_h和f_l为滤波器截止频率的上下限，一般取f/2和 0

先在梅尔频率域中求出滤波器中心，然后再转换为普通频率域（赫兹）。

然后利用下面的公式：

$$H_m(k) = \begin{cases} 0 & (k < f(m-1)) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & (f(m-1) \le k \le f(m)) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & (f(m) < k \le f(m+1)) \\ 0 & (k > f(m+1) \end{cases}$$

于是得到 M 个三角形滤波器。将前面得到的能量谱进行滤波后取对数得到相应频带的对数功率谱。

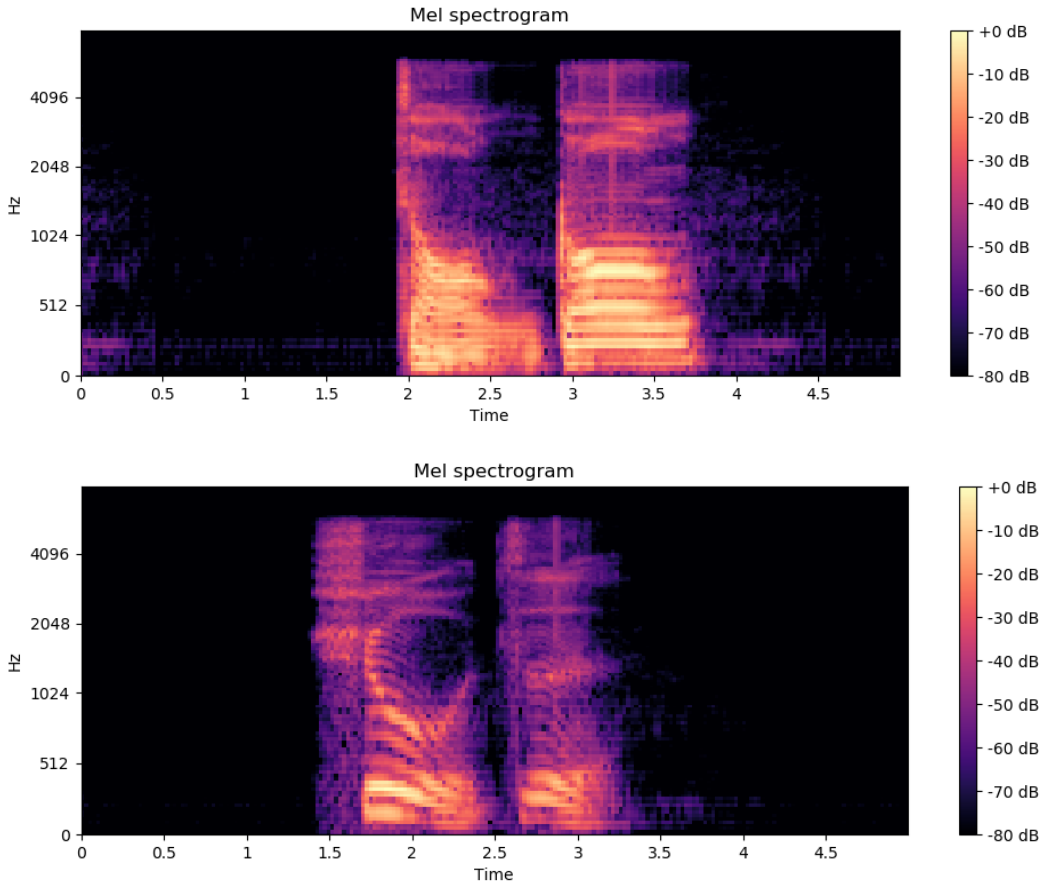


图 4：同一个人不同词语对应的谱图

传统上的语音识别算法会继续使用反离散余弦变换求出 MFCC 系数。但由于 CNN（卷积神经网络）在处理图像上展现了强大的能力，使得音频信号的频谱图特征的使用愈加广泛，甚至比 MFCC 使用的更多。

至此完成了特征的提取，接下来需要根据特征进行分类。

2.3 分类器设计

由于对神经网络的了解并不深入，我使用了他人搭建好的 VGG 神经网络，代码链接在参考文献中，这里不做赘述。

3 实验设置及结果

在上文中已描述过对数据的预处理与特征提取，得到处理过的数据后，首先需要对数据集进行划分，选择一部分数据作为测试集。由于数据量有限，对数据进行了交叉验证的划分，采用 10 折交叉的数据划分方式。利用 pytorch 提供的 Dataset 和 Dataloader 模块，编写了数据加载器。

利用 pytorch 提供的自动反向传播，用交叉熵损失函数作为优化目标，对模型进行了训练。绘制出训练的损失曲线如下图：

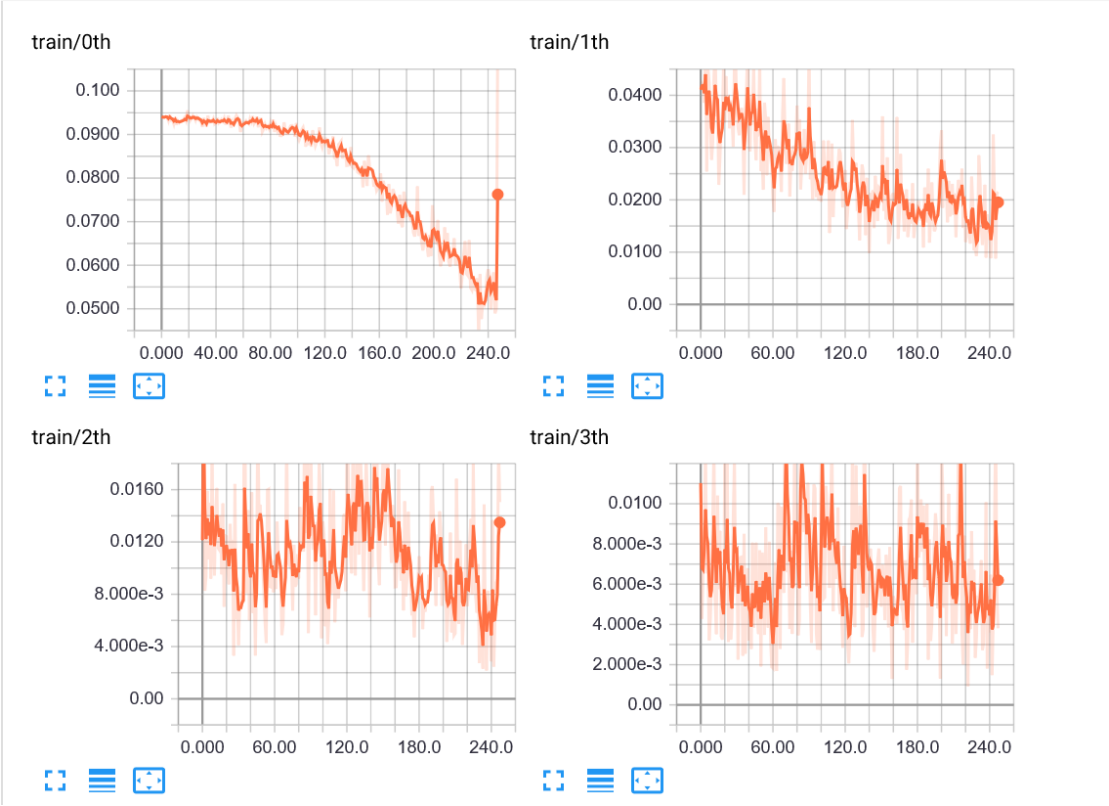


图 5：前几轮迭代的损失曲线

共进行了 9 次迭代，正确率达到 97.5%。

4 识别系统界面

前端基于 web 技术，通过浏览器获取用户麦克风进行录音，录音完成后通过 jQuery 和 Ajax 进行语音是上传和获取识别结果，利用了 vue.js 框架。后端则

使用了 flask 框架编写，当其接收到语音信号后，将载入训练好的模型参数进行识别，将识别结果返回前端。

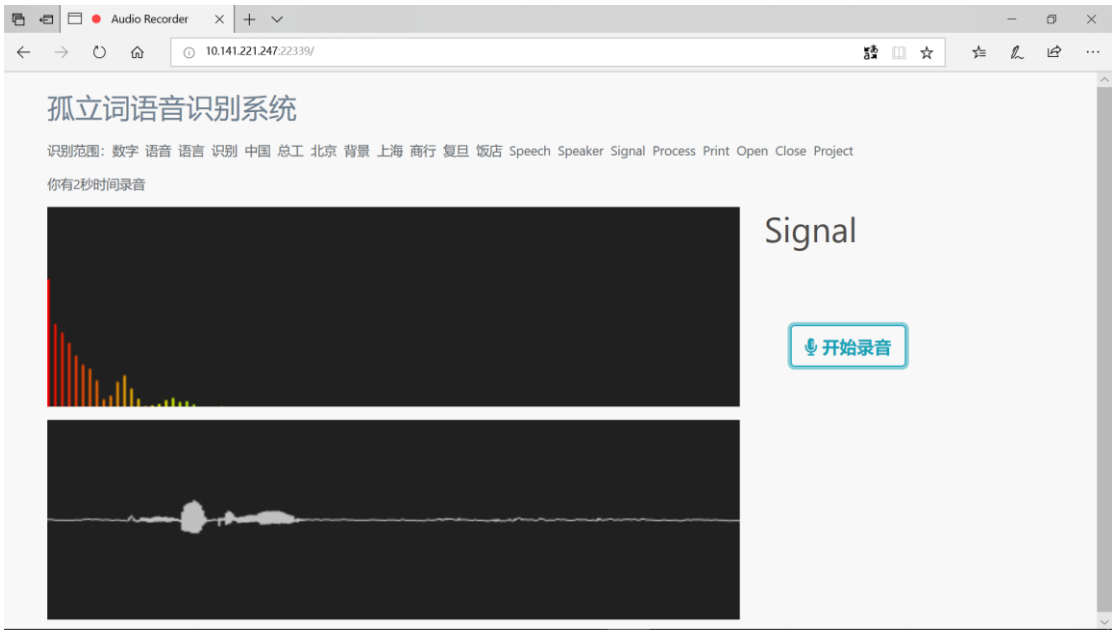


图 6：识别系统界面

5 结论

在本项目中，我实现了一个孤立词语音识别系统，利用梅尔频率域下的时频分析，以及卷积神经网络，得到了一个识别效果较好的系统。在这一过程中，我对数字信号处理的很多算法细节加深了理解，并将其运用到实践中，受益匪浅。

非常感谢薛向阳老师一个学期以来的教导，在这门课上我学到了好几门学科交叉的知识，您对我的问题的耐心解答，每堂课之前对前面知识的回顾，课上对同学们反应的关注，都令我十分感动。

6 参考文献

1. 《什么是泄漏》谭祥军
https://mp.weixin.qq.com/s?__biz=MzI5NTMOMTQwNA==&mid=2247484164&idx=1&sn=fdaf2164306a9ca4166c2aa8713cacc5&scene=21#wechat_redirect

2. MFCC 过程理解 jinmingz
https://blog.csdn.net/zjm750617105/article/details/51690364/

3. 语音识别的第一步 MFCC 特征提取代码（Python）断桥残雪
http://www.itkeyword.com/doc/862134162547590x963/python-MFCC

4. vgg 模型代码
https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py

5.数字信号处理课件