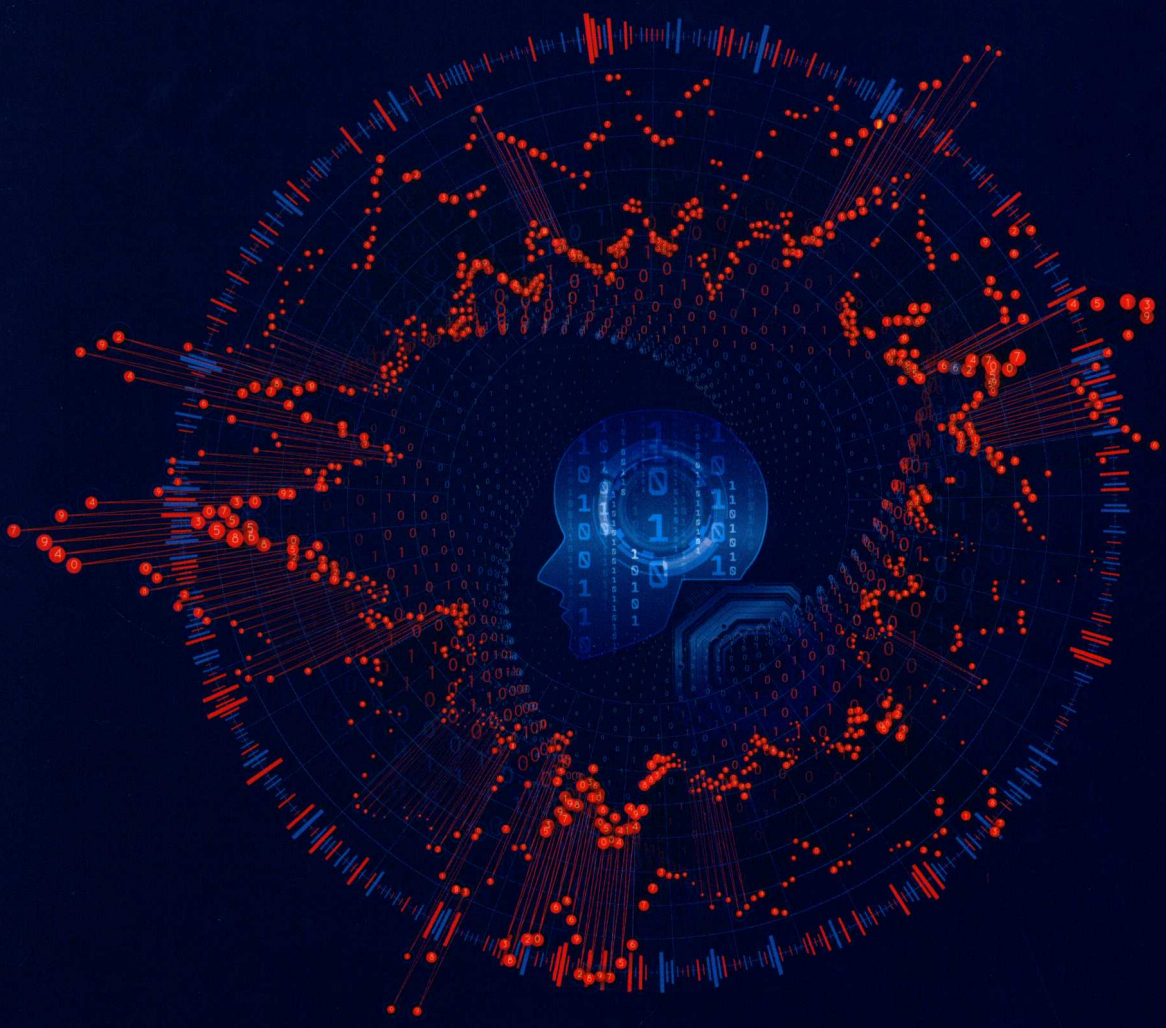

深度学习

语音识别技术实践

- 业内流行的Kaldi语音识别技术实践
- 猎兔搜索技术团队语音识别技术总结
- 引领语音识别技术升级

柳若边 编著

清華大學出版社

深度学习

语音识别技术实践

柳若边　编著

清華大學出版社
北京

内容简介

语音识别已经逐渐进入人们的日常生活。语音识别技术是涉及语言、计算机、数学等领域的交叉学科。本书介绍了包括 C#、Perl、Python、Java 在内的多种编程语言实践，开源语音识别工具包 Kaldi 的使用与代码分析，深度学习的开发环境搭建，卷积神经网络，以及语音识别中常见的语言模型——*N* 元模型和依存模型等，让读者快速了解语音识别基础，掌握开发语音识别程序的算法。

本书从语音识别的基础开始讲起，并辅以翔实的案例，既适合需要具体实现语音识别的程序员使用，也适合有一定机器学习或语音识别基础的学生、研究者或从业者阅读。

图书在版编目（CIP）数据

深度学习：语音识别技术实践 / 柳若边编著. —北京：清华大学出版社，2019
ISBN 978-7-302-51692-7

Ⅰ. ①深…　Ⅱ. ①柳…　Ⅲ. ①机器学习－语音识别－研究　Ⅳ. ①TP181

中国版本图书馆 CIP 数据核字（2018）第 265420 号

责任编辑： 张　敏
封面设计： 杨玉兰
责任校对： 胡伟民
责任印制： 杨　艳

出版发行： 清华大学出版社
网　　址：http：//www.tup.com.cn, http：//www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
印 装 者： 清华大学印刷厂
经　　销： 全国新华书店
开　　本： 186mm×240mm　　**印　　张：** 18.25　　**字　　数：** 355 千字
版　　次： 2019 年 4 月第 1 版　　**印　　次：** 2019 年 4 月第 1 次印刷
定　　价： 89.00 元

产品编号： 078565-01

前　言

作为人工智能技术的重要组成部分，语音识别旨在研究计算机如何听懂人的讲话。来源于人工神经网络的深度学习促进了语音识别技术的发展。本书从使用开源的语音识别构建系统 Kaldi 开始讲起，引导读者亲自实现语音识别系统，使用了 C#、Perl、Python、Java 等多种编程工具。第 1 章介绍语音识别的基本原理和 Kaldi 的基本使用方法，以及使用 Kaldi 开发语音识别系统应用到的 Linux shell 脚本基础；第 2 章介绍使用 C#开发语音识别系统；第 3 章介绍 Perl 语言开发基础；第 4 章介绍开发语音识别系统所需要的 Python 基础；第 5 章介绍使用 Java 开发语音识别系统；第 6 章介绍傅里叶变换、MFCC 特征等常用的语音信号处理方法；第 7 章介绍基本的神经网络和深度学习方法及训练神经网络的反向传播方法；第 8 章介绍语音识别解码阶段用到的语言模型，以及语言模型工具包——KenLM。

本书适合需要具体实现语音识别的程序员使用，对机器学习等相关领域的研究人员也有一定的参考价值。猎兔搜索技术团队已经开发出以本书为基础的专门培训课程和商业软件。

本书由柳若边编著，罗刚、沙芸、张子宪、许想娇、石天盈、张继红、罗庭亮、王全军、刘宇、张天津也参与了本书的部分编创工作。本书相关的参考软件和代码在读者 QQ 群（378025857）的附件中可以找到。Kaldi 及其底层依赖的软件，其复杂程度已经超越了一个人所能掌握的程度。此外，一些具体的细节也可以在读者 QQ 群讨

论。在此，感谢早期合著者、合作伙伴、员工、学员、读者的支持，他们为本书的编创提供了良好的工作基础。技术的融合与创新永无止境，就如同在玻璃容器中水培植物一样，这是一个持久的工作。

编著者

2018 年 12 月

目 录

第 1 章 语音识别技术

语音识别技术，也被称为自动语音识别（Automatic Speech Recognition，ASR），它是一门交叉学科，与人们的生活和学习密切相关。其目标是将说话者的词汇内容转换为计算机可读的输入按键、二进制编码或字符序列等。例如，打银行的客服电话，可以直接和银行系统对话，而不是普通的“请按 1”等把人当成机器的询问。在通信中，可以把对方的语音留言转换成文字，还可以根据识别出的文字识别语义，这样可以让机器和人交流。再如，儿童识别图片后，可以说出这个图中是老虎还是大象，系统使用语音识别技术判断孩子回答是否正确，对于不正确的，系统自动给出提示。

做好开放式语音识别不容易，可以辅助人工输入字幕，类似于语音输入法。

1.1 总体结构

语音识别可以看成是广义上的标注问题。给定声学输出 $A_{1,\mathrm{T}}$（由一个声学事件的序列组成 $a_1,\cdots,a_{\mathrm{T}}$），需要找到单词序列 $W_{1,\mathrm{R}}$ 的最大化概率：

$$\arg\max_{\omega} P\left(W_{1,\mathrm{R}} \mid A_{1,\mathrm{T}}\right)$$

根据贝叶斯公式重写上述公式，并删除在通过比较大小找最大值的过程中没有意义的分母，把问题转换成计算：

$$\arg\max_{\omega} P\left(A_{1,\mathrm{T}} \mid W_{1,\mathrm{R}}\right) P\left(W_{1,\mathrm{R}}\right)$$

这里将 $P(A_{1,T} \mid W_{1,R})$称为声学模型，而将 $P(W_{1,R})$称为语言模型。语言识别结构如图 1-1 所示。

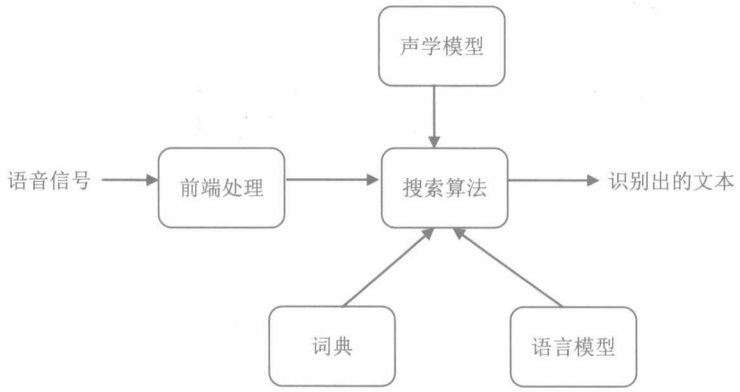

图 1-1　语音识别结构

人类获得信息的 80% 都来自图像。图像信息具有传递速度快、信息量大等一系列特点，因此图像信息得到了广泛的应用。但语音识别在车载系统、智能音响等领域也有非常关键的应用。

为了能开发出有效的语音识别系统，2009 年 Kaldi 在约翰·霍普金斯大学诞生了。Kaldi 不是一款语音识别系统，而是一款建立语音识别系统的系统。Kaldi 使用运行于 Linux 操作系统的 C++、Perl、Python、Bash 等多种语言开发。接下来介绍需要用到的 Linux 基础知识。

1.2　Linux 基础

Linux 是围绕 Linux 内核构建的免费和开源软件操作系统系列。通常，Linux 以桌面和服务器使用的称为 Linux 发行版的形式打包。Linux 有一些常用的发行版 CentOS 和 Ubuntu 等版本。

和 CentOS 不同，Ubuntu 操作系统上没有管理员知道 root 密码（root 密码是随机生成的），而 root 权限是通过操作 sudo 命令授予的。

有些语音识别系统运行在 Linux 服务器中，为了远程登录 Linux 服务器，用户可以先在 Windows 下安装 Chrome 浏览器，然后可以通过网址 http://sshy.us/登录 Linux 服务器。在界面中输入 IP 地址、用户名和密码，如果是用 root 账户登录，则终端提示符为“#”；否则，终端提示符为“$”。

查看 Ubuntu 操作系统版本号：

```
$cat /etc/issue
Ubuntu 18.04 LTS \n \l
```

或者：

```
$lsb_release -r
Release:        18.04
```

获取 Ubuntu 的代号：

```
$lsb_release -c
Codename:       bionic
```

ls 命令用于列出当前目录下的文件；history 命令用于显示历史。有的命令比较长，为了实现快速输入，可以用 Tab 键补全命令；也可以用上箭头选择最近运行过的命令再次执行。

连接到远程 Linux 服务器可以使用支持 SSH 协议的终端仿真程序 SecureCRT。因为它可以保存登录密码，所以应用起来比较方便。除了 SecureCRT，还可以使用开源软件 PuTTY（http://www.chiark.greenend.org.uk/~sgtatham/putty），以及可以保存登录密码的 PuTTY Connection Manager。在终端启动的进程断开连接后会停止运行。为了让进程继续运行，可以使用 nohup 命令。

如果需要安装软件，可以下载对应的 RPM 安装包，然后使用 RPM 安装。但操作系统对应的 RPM 安装包找起来往往比较麻烦——一个软件包可能依赖其他的软件包；为了安装一个软件，可能需要下载其他的多个它所依赖的软件包。

为了简化安装操作步骤，可以使用黄狗升级管理器（Yellow dog Updater, Modified），

一般简称 YUM。YUM 会自动计算出程序之间的相互关联性，并且计算出完成软件包的安装需要哪些步骤。这样在安装软件时，不会再被那些关联性问题所困扰。

YUM 软件包管理器会自动从网络下载并安装软件。YUM 有点类似 360 软件管家，但是不会有商业倾向的推销软件。例如安装支持 wget 和 lrzsz 命令的软件，执行以下命令行：

```
#yum install wget
#yum install lrzsz
```

Windows 格式文本文件的换行符为\r\n，而 Linux 文件的换行符为\n。dos2unix 命令是将 Windows 格式文件转换为 Linux 格式的实用命令，其实就是将文件中的\r\n 转换为\n。

开发语音识别系统的过程中，可能会用到大量的数据文件。例如需要在 Linux 操作系统上维护同一文件的两份或多份副本，除了保存多份单独的物理文件副本以外，还可以采用保存一份物理文件副本和多个虚拟副本的方法。这种虚拟的副本就称为链接。链接是目录中指向文件真实位置的占位符。在 Linux 中有两种不同类型的文件链接——符号链接和硬链接。其中，符号链接是一个实实在在的文件，它指向存放在虚拟目录结构中某个地方的另一个文件。这两个通过符号链接在一起的文件，彼此的内容并不相同。

对于过大的文件，可以使用 wget 命令在后台下载。

```
#wget -bc <path>
```

这里的参数 b 表示在后台运行；参数 c 表示支持断点续传。

1.3 安装 Micro 编辑器

为了方便在服务器端开发 Python\Perl 的相关应用，可以采用 Micro（https://github.com/zyedidia/micro）这样的终端文本编辑器。如果有 Snap 安装工具软件，可以使用 Snap

安装 Micro：

```
#snap install micro --classic
```

如果没有 Snap 安装工具软件，也可以直接安装 Micro 的预编译版本：

```
#wget  https://github.com/zyedidia/micro/releases/download/nightly/micro-1.3.4-67-linux64.tar.gz
#tar -xf ./micro-1.3.4-67-linux64.tar.gz
```

编辑/etc/profile 文件，增加 micro 所在的路径/home/soft/ micro-1.3.4-67 到 PATH 环境变量下。

```
#./micro /etc/profile
export PATH=/home/soft/micro-1.3.4-67:$PATH
```

可以使用它编辑配置文件：

```
#./micro  run.pl
```

输入以下命令行：

```
die "run.pl: Hello Error";
```

这里的 die 表示终止脚本运行，并显示出 die 后面双引号中的内容。

保存文件后，按 Ctrl+Q 组合键退出。

1.4 安装 Kaldi

一般在 Linux 操作系统下运行 Kaldi，下面讲解在 CentOS 下安装 Kaldi。

首先安装 Git。

```
#yum install git
```

然后下载 Kaldi。

```
#git clone https://github.com/kaldi-asr/kaldi.git kaldi --origin upstream
```

可以参考下载文件中的说明安装。在源码的根目录下有一个 INSTALL 文件，其中描

述了安装步骤。在 tools/下查看 INSTALL 安装指令，然后在 src/下查看 INSTALL 安装指令。到/tools/extras 目录下，运行 check_dependencies.sh 脚本，检查安装过程中所依赖的工具是否存在，运行的结果会提示安装依赖软件的命令。在./tool 目录下编译源代码，然后在./src 目录下编译。

在./tool 目录下只需输入 make 命令就可以编译，输入 make -j 4 命令可以用多核并行处理的方式加快速度。

切换到./src 目录下，运行如下命令：

```
./configure
make depend
make -j 4
```

为了方便以后使用，可以把环境打包：

```
tar -czf kaldiLinux.tar.gz ./kaldi
```

egs 目录下保存着一些指定数据集上的训练步骤（shell 脚本）及测试的结果。最简单的是 yesno 例子。

1.5 yesno 例子

首先运行以下这个例子。

```
#cd ./egs/yesno/s5
#./run.sh
```

经过一段时间的训练和测试，可以看到以下运行结果：

```
%WER 0.00 [ 0 / 232, 0 ins, 0 del, 0 sub ] exp/mono0a/decode_test_yesno/wer_10
```

这里的 WER（Word Error Rate）是字错误率，是一个衡量语音识别系统准确程度的度量。其计算公式为 $WER=(I+D+S)/N$，其中 I 代表被插入的单词个数；D 代表被删除的单词个数；S 代表被替换的单词个数。也就是说，把识别出来的结果中，多认的、少认的和认错的全都加起来，再除以总单词数。这个数值当然是越低越好。这里的 *WER*

为 0.00，说明全部识别正确。

数据集有 62 个.wav 文件，采样频率为 8kHz。所有音频文件由 Kaldi 项目的匿名男性贡献者记录，并包含在项目中用于测试目的。把它们放在 wave_yesno 目录中，但数据集也可以在 http://openslr.org/resources/1/waves_yesno.tar.gz 中找到。在每个文件中，这个人说 8 个字，每个单词都是“ken”或“lo”（希伯来语中的“是”和“否”），因此每个文件都是 8 个“是”或“否”的随机序列。以下文件名称用单词序列表示，1 代表“是”，0 代表“否”。

```
waves_yesno/1_0_1_1_1_0_1_0.wav
waves_yesno/0_1_1_0_0_1_1_0.wav
...
```

1.5.1 数据准备

将 62 个波形文件分为两半：31 个用于训练，其余用于测试。创建数据目录，在其中创建两个子目录 train_yesno 和 test_yesno。使用一个命名为 data_prep.py 的 Python 脚本生成必要的输入文件。读取 wave_yesno 中的文件列表。生成两个列表，一个存储以 0 开头的文件名称，另一个存储以 1 开头的名称，忽略其余的文件。

对于每个数据集（训练集和测试集）都需要生成代表原始数据的文件——音频文件和讲稿文件。

1. 讲稿文件

对于讲稿文件，每行一句话，语法格式为：

```
<utt_id> <transcript>
```

例如： 0_0_1_1_1_1_0_0 NO NO YES YES YES YES NO NO

这里使用没有扩展名的文件名作为 utt_ids。虽然录音语言是希伯来语，但是这里使用英语单词 YES 和 NO 代替，以免使问题复杂化。

2. wav.scp

对于唯一 ID 的索引文件，语法格式为：