分类号：TP311.5　　单位代码：10220

密　　级：

东北石油大学

Northeast Petroleum University

硕士研究生学位论文

论文题目：神经网络时序分类方法在语音识别中的应用研究

硕 士 生：王雨萌

指导教师：赵建民　副教授

学科专业：软件工程

研究方向：数字媒体技术

2019 年 5 月 29 日

Thesis for the Master degree in Engineering

Research on Connectionist Temporal Classification in Speech Recognition

Candidate: Wang Yumeng

Tutor: Zhao Jianmin

Specialty: Software Engineering

Date of oral examination: 29th May, 2019

University: Northeast Petroleum University

学位论文独创性声明

本人所呈交的学位论文是我在指导教师的指导下进行的研究工作及取得的研究成果。据我所知，除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：王雨萌　　日期：2019年5月29

学位论文使用授权声明

本人完全了解东北石油大学有关保留、使用学位论文的规定。学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文，可以公布论文的全部或部分内容。东北石油大学有权将本人的学位论文加入《中国优秀硕士学位论文全文数据库》、《中国博士学位论文全文数据库》和编入《中国知识资源总库》。保密的学位论文在解密后适用本规定。

学位论文作者签名：王雨萌

论文指导教师签名：

指导小组成员签名：

神经网络时序分类方法在语音识别中的应用研究

摘　要

随着人工智能领域研究的深入以及大数据语料的不断积累，语音识别技术得到突飞猛进的发展，神经网络开始大规模应用于语音识别技术，端到端语音识别成为近年来人工智能研究的热点课题。然而，由于特定说话人含有不同发音特性、不同语种发音特性不同等原因，导致端到端语音识别模型对中文的识别准确率偏低。基于以上背景，本文结合中文语言模型对现下主流端到端语音识别框架的结构进行研究和改进，以提高端到端语音识别框架对于汉语的识别准确率和效率。

首先，本文设计基于传统隐马尔科夫-混合高斯声学模型结合词典和 N-gram 语言模型的建模方法作为本文的基线实验。在对隐马尔科夫-混合高斯模型的研究中，对语音信号易于受到上下文内容影响的问题，设计利用上下文相关的三音素声学模型，通过考虑每一音素相邻的前后音素，进而提升模型性能。同时，对语音易受到说话人发音特性影响的问题，设计融合说话人自适应技术的隐马尔科夫-混合高斯模型的建模方式，以提高本文基线实验的识别准确率。

其次，本文针对端到端框架对中文识别准确率较低的问题，提出结合语言模型的非完全端到端框架的语音识别方法，将非完全端到端框架应用于神经网络时序分类方法对语音识别的研究中。针对 LSTM-CTC 端到端模型计算复杂度高，训练速度耗时过长的问题，本文提出了一种改进的映射长短期记忆时序网络，用以优化模型的训练速率。同时针对语音特征的长时依赖性并不只有正向传播的特点，在端到端模型中设计采用双向映射长短期记忆时序网络，替代固有的单向长短期记忆时序网络，通过对语音特征进行双向提取，从而提高模型的准确率。

最后，选取希尔贝壳公司的 AISHELL 语音数据库对本文所提出的方法进行实验验证，针对实验过程中双向神经网络训练易产生过拟合的问题，将语音数据库通过速度扰动技术进行扩充、实验。最终实验结果显示，模型的准确率和速率较基线实验结果都得到显著提升。

关键字：隐马尔科夫混合高斯模型；链接时序分类；映射长短期记忆时序网络；双向神经网络；速度扰动

Research on Connectionist Temporal Classification in Speech Recognition

ABSTRACT

With the development of research in artificial intelligence and the continuing accumulation of big data corpus, speech recognition has rapidly developed now. Neural network has been extensively applied to speech recognition technology, end-to-end speech recognition has recently become a hot topic in artificial intelligence research. However, due to the complexity of its real application scenarios and speaker pronunciation characteristics, the end-to-end speech recognition model for Chinese gets relatively low accuracy. Aiming at the above problems, we take the Chinese pronunciation characteristics into consideration to optimize and improve the current mainstream end-to-end speech recognition model structure, which is aimed to improve the recognition performance and training efficiency of the end-to-end speech recognition framework for Chinese.

Firstly, we design a baseline experiment based on method which combines Hidden Markov Model (HMM)-Gaussian Mixture Model (GMM) acoustic model, lexicon and N-gram language model. In the study of the GMM-HMM model, aiming at the problem of susceptibility of speech signals to context, we consider the front and back phoneme of current phoneme while building tri-phone acoustic model. Considering the influence of speaking style of different speakers, we adopt speaker adaptation technologies in GMM-HMM modeling to increase the recognition accuracy of baseline experiment.

Then, aimed at the low accuracy of end-to-end framework applied to Chinese, we use incomplete end-to-end structure and apply this structure to speech recognition research of neural network time series classification method. In our research, because the LSTM-CTC end-to-end model have drawbacks, such as high computational complexity and long training time, we propose an improved model, i.e. Projection Long Short-term Memory (PLSTM) to speed up the model training. Because the long-term dependence of speech is not only in forward direction, in this work we use bidirectional Long Short-term Memory (Bi-LSTM) instead of LSTM or RNN combined with Connectionist Temporal Classification (CTC), which can help improve the accuracy.

Finally, We started our experiment on the speech database of AISHELL,we use speed-perturbed training data to avoid overfitting while training Bi-LSTM. In the final experiment results, compared with the baseline experimental results ,the accuracy and the speed of the model are all significantly improved.

Keywords: Hidden Markov and Gaussian Mixture Model, Connectionist Temporal Classification, Prejected Long Short-Term Memory Network, Bidirectional Neural Network, Speed Perturbation .

创新点摘要

本文创新点如下：

1. 针对长短时序记忆网络计算复杂度高，训练和推理效率低的问题，提出了一种增加低维映射层的改进长短时序记忆网络，并将其与链接时序分类算法相结合进行端到端训练，提升模型计算效率与训练速度。

2. 针对单纯端到端模型对汉语识别准确率较低的问题，提出非完全端到端语音识别框架，将端到端语音识别框架与中文语言模型结合，通过对输出标签列表进行二次打分的方式，提高识别准确率。

目 录

第一章 绪 论

1.1 研究背景与意义

语音是人类和机器交互方式中较为便捷、自然的一种方式[1,2]。语音识别以人类语言为研究对象，通过对语音信号的处理和模式识别使移动终端设备自动理解人类语言，并使机器借助模式识别将语音信号转化为对应文本的一种技术。

近年来苹果公司的 Siri 语音助手、亚马逊公司的 Echo 智能音箱等产品相继问世[3]。小米、百度、京东等互联网公司也争相推出了人工智能音箱[4]。搜狗公司在 2018 年 3 月推出了实时旅行语音翻译宝和采访翻译笔，多家汽车厂商也推出了各类车载语音助手，让机器会“听”已成为融入人类生活的一项不可或缺的技术。

上面所提及的语音助手、智能音箱、翻译宝等产品，其核心就是语音识别技术，作为人工智能所涵盖的重要技术之一，语音识别技术一直以来都是学术界的研究重点。传统的语音识别技术，通常是基于隐马尔科夫和混合高斯分布构建声学模型，将训练完成的声学模型结合语言模型和词典进行识别实验[5]。此模型虽然具有较高的识别性能，但其模型相对复杂，训练效率较低。随着计算机硬件性能的飞速发展，越来越强大的计算能力促使神经网络技术不断进步，建模能力更强、训练和识别效率更高的网络模型被大量开发。混合高斯分布模型与隐马尔科夫模型相结合的传统固有模式，逐渐被深度神经网络所取代，并由此构成语音识别的混合系统模式。然而在这种训练方法中，深度神经网络还只是这套复杂系统中很小的一个组成部分，仅被用来对语音帧做分类，神经网络输出的概率分布被作为隐马尔科夫模型的输出概率模型。同时，采用混合系统中还存在两方面问题，一是深度神经网络强大的建模能力没有被完全利用，二是由于训练神经网络的目标函数跟语音识别性能指标函数（通常是句子或者词的准确率）不一致，从而可能导致神经网络对语音帧的分类准确率很高，但识别系统对语音的整体识别准确率却很低[6]。另外，基于隐马尔科夫模型的识别技术需要大量的专家知识，例如对三音素（triphone）进行建模时会出现参数量过大的问题。为了减少参数量，隐马尔科夫模型通常需要对其状态进行绑定，而状态绑定需要借助决策树做聚类，决策树的构建需要语言学及音素学的专业知识[7]。

虽然深度神经网络建模方式存在一定的问题，但这种建模方式也给语音识别技术带来了巨大的变革，并促成了端到端语音识别技术的出现。端到端模型是一种基于神经网络的训练方式，由于其简单的框架结构逐渐成为语音识别领域的研究热点。端到端模型在训练时，以声学特征作为输入，以音素或者词的分类准确率作为目标，直接对深度神经网络进行训练，减少了传统识别模型的中间环节和过程，并由此解决了由于中间过程目标和整体目标不一致而导致的模型整体性能下降的问题。另外，端到端模型的训练不再需要大量的专业知识，所以在工业界，无论大公司还是初创型公司都可以方便地训练语音识别模型，并将其运用到产品中。

因此，研究端到端的语音识别技术并将其运用到中文语音识别，一方面会对语音识别领域的技术进步有一定的贡献，另一方面，很大程度上降低了研发语音智能产品的成本，为相关公司带来巨大的价值回报。

1.2 国内外研究现状

1.2.1 国外研究现状

国外对语音识别技术的研究从上个世纪就已经开始，1976 年，Reddy 在卡耐基梅隆大学 CMU（Carnegie Mellon University）领导一个小组开始研究语音识别技术，这是人类最早开始研究机器语音识别技术的团队之一。在之后的几十年时间里，Reddy 和他的同事们创造了多个基于语音的系统，如语音控制的机器人、大词汇连续语音识别、说话人无关的语音识别系统以及词汇不限的语音听写等[8]。其中，Hearsay-I 系统是最早的连续语音识别系统之一，在 Hearsay 系统中，引入了束搜索（beam search）这一技术概念，这一技术是当时应用最为广泛的搜索和匹配技术。

1987 年 Reddy 开发的 Sphinx-I 是第一个说话人无关的语音识别系统。得益于参数绑定技术，5 年后 Sphinx-II 系统问世，Sphinx-II 系统让混合高斯分布模型和隐马尔科夫模型的训练变得高效，该系统在 1992 年举行的语音识别竞赛中获得最高的识别准确率。同一时期，由卡耐基梅隆大学、剑桥大学和约翰霍普金斯大学开发的 Sphinx、HTK（Hidden markov model Tool Kit）和 Kaldi 等语音识别工具，至今仍然是学术界和工业界研究和开发语音识别技术或系统的有效工具，应用十分广泛[9]。

在 2010 年之前，隐马尔科夫模型和混合高斯分布模型相结合的技术，是语音识别领域最先进的技术，主导着语音识别这一领域，直到现在工业界仍在大量使用此技术。但随着此模型技术的普及，研究者们也发现该模型存在着一些问题，最为典型的就是混合高斯分布模型对数据分布的建模能力相对低效。此问题被发现的同时，学者们对深度学习技术的探索也在不断深入，建模能力更强大的深度神经网络模型被提出。神经网络从最初的前馈神经网络逐渐演变出时延神经网络、卷积神经网络以及递归神经网路（包括长短时记忆神经网络、门限单元神经网络等）等复杂高效的网络模型[10-11]。

之后，Hinton 等研究者将深度神经网络引入到语音识别技术中，用神经网络直接替代了混合高斯分布模型[12]。这种将隐马尔科夫模型和深度神经网络相结合的模型被称之为混合系统（Hybrid System）。在混合系统中，深度神经网络不仅可以对音素声学特征的分布进行建模，而且还可以自动学习到强大的、有区分性的声学特征，语音识别技术也因此在领域研究上有了更大的进步[13]。

2016 年 5 月，IBM 宣布其开发了采用 HMM 解码和神经网络语言模型的英语会话语音识别系统，并创造出 6.9%的词错率新纪录。该系统的声学模型部分主要包括 RNN、DEEP-CNN（Deep-Convolutional Neural Network）以及 LSTM 三种神经网络。

谷歌公司在人工智能领域一直处于领先地位，从近年来谷歌在各大会议上发表的论文来看，谷歌正在尝试在端到端模型上使用多种神经网络融合的方法进行实验。如2017年的ICASSP会议中，谷歌展示了网络中网络NIN（Network-in-Network）、批规范化BN（Batch Normalization）以及LSTM融合结构的语音识别系统[14]。

1.2.2 国内研究现状

国内虽然对语音识别技术的研究起步较晚，但近年来的发展却十分迅速。1991 年开始，国家 863 计划的语音专家们开始每年举行语音识别的系统测试，自 2009 年开始，微软中国研究院的语音识别专家们开始与研究者 Hinton 的合作[15]。

2010 年左右，借助数据存储设备的迅速发展，存储海量数据成为现实，图形处理器（GPU）在矩阵运算中也表现出了超高的性能，神经网络的发展得到了强有力的硬件支持。于是很多国内大公司开始大规模投入到人工智能领域的研究。微软中国发布了基于神经网络的语音识别系统，该系统推翻了语音识别现有的框架结构，其结构为采用神经网络对高维特征进行模型训练[16-17]。虽然此时国内对于神经网络应用于语音识别技术研究有了一定的进展，但技术水平还远没达到投入商用的水平。所以虽然基于隐马尔科夫模型的混合系统存在一定的缺陷，但其依然是当时国内主流的语音识别技术。

2011 年辛顿、俞栋等科学家专注于深度神经网络应用于语音识别的研究，先后在小词汇量连续语音识别和大词汇量连续语音识别实验中取得了重大突破。自此基于 GMM-HMM 的识别框架不再主导语音识别领域，大多数研究人员开始投入到基于 DNN-HMM 语音识别系统的研究中，掀起了国内神经网络在语音识别领域的研究热潮。神经网络如何应用于语音识别技术和神经网络如何提升语音识别的准确率已成为现下人工智能领域的研究热点。

2012 年国内百度公司上线了一款基于端到端的训练模型的语音搜索系统，此模型丢弃了复杂冗余的语言先验知识，直接从声学特征或语音波形图中训练深度神经网络，完成从语音到文本的识别。2013 年百度公司再一次在其年会上提出将成立深度学习研究院 IDL（Institute Deep Learning），重点研究深度学习的各个应用领域。近年来，百度在语音识别技术上的研究一直在不断更新，2016 年百度公司宣布了其基于大数据和深度卷积神经网络 Deep-CNN 训练至少上万小时的语音产品。

2016 年，科大讯飞相继提出了两种基于神经网络的语音识别框架，分别采用前馈型记忆网络 FSMN（Feed-forworard Sequential Memory Network）和深度全序列卷积神经网络 DFCNN（Deep Fully Convolutional Neural Network），通过采用直接对整句语音信号建模的方式，更好的解决了语音的长时依赖性问题。

综上，如何利用神经网络表达语音的长时依赖性，以及如何训练端到端语音识别模型现今已成为行业领域的研究重点和热点。

第二章 基于中文语言模型的非完全端到端语音识别框架

语音识别技术发展至今已有近百年历史，在不同发展阶段，语音识别均形成了不同的识别框架。本章首先研究自语音识别技术发展以来的几种主流识别框架，然后分析语音信号预处理以及模型解码全过程的基本原理，最后本章通过对比分析不同框架的优缺点，设计了一种基于中文语言模型的非完全端到端语音识别框架。

2.1 传统语音识别框架结构分析

传统的语音识别框架结构是完全基于机器学习模式识别方法的，识别过程为首先对语音数据进行特征提取，然后解码器结合声学模型、词典和语言模型将语音识别为文字序列[29]。传统语音识别框架结构如图 2.1 所示。

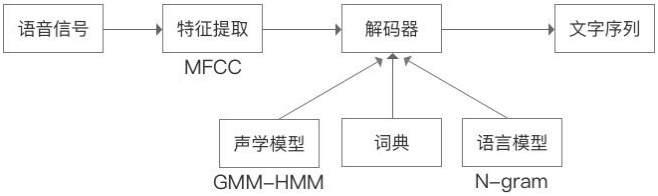

图 2.1 传统语音识别框架结构图

1．特征提取

人类在发出声音时产生的波形之所以不同，是因为声音经过声带、舌头、牙齿不同的变换形态过滤而形成。这三个部位中的任意一个在发声时稍有变动，那么声音的谱图就会有很大差异化的改变。因此在进行语音识别实验时，无法将语音波形进行直接比较来判别语音的内容。

这促使人们在进行实验前，首先需要将语音进行特征提取[30]。采用传统的语音识别框架时，人为设计的语音特征包含很多类别，如梅尔频率倒谱系数 MFCC（Mel Filter Cesptral Coefficient）、感知线性预测 PLP（perceptual linear prediction）、滤波器组 Fbank（FilterBank）等[31-33]。（本文第三章基于自适应技术的 GMM-HMM 算法模型实验所采用的语音特征为 MFCC，第四章基于神经网络时序分类算法模型所采用的语音特征为 Fbank）

本文以梅尔频率倒谱系数为例，对语音特征的提取过程开展研究。MFCC 特征提取过程中，首先对语音信号进行预处理，用以增强波形图中的高频部分。然后，将语音分为等长的语音段并对每帧语音进行加窗操作，接下来将处理好的语音波形进行傅里叶变换，得到能量谱图并对能量谱图通过梅尔滤波器。最后，将滤波器组的输出取对数再进行离散余弦变换。MFCC 特征提取流程如下图 2.2 所示。

（1）语音预处理

语音的预处理是首先将语音信号进行预加重，即对语音的高频部分进行增强处理，以利于在后续转换频域谱图时进行声道参数分析和频谱分析。预加重是将语音通过一节数字滤波器，滤波器为6db/倍程，预加重表达式如式2-1所示。

$$H(Z) = 1 - uz^{-1} \quad (2\text{-}1)$$

其中，u为滤波器系数，取值为u ∈ （0.9，1.0），如此预加重后，信号中语音的高频部分被提升，有利于后续特征参数的分析。

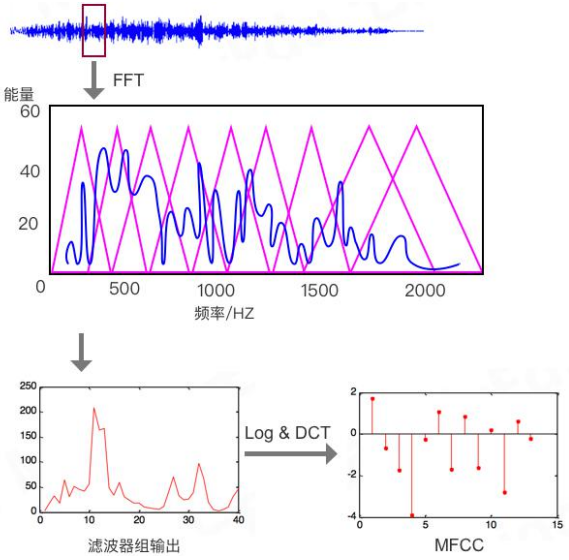

图 2.2 MFCC 特征提取过程图

（2）分帧加窗

由于语音信号是持续变化的，为了简化分析对语音进行预处理后需要对其进行分帧处理。分帧操作是将不定长的大段音频切分成固定长度的小段。通常每帧语音的长度需满足包含语音波形的两到三个周期，且满足每一帧的长度都在一个音素的范围内。故一帧语音的长度通常选取在 20~50ms 之间，同时为了避免帧与帧之间的特性过大，在分帧时会将每两帧间会将相邻两帧间重叠一部分，即所谓的帧移。

由于在后续需要进行的傅里叶变换步骤中，易出现频谱混叠的问题，所以在分帧后，会对每一帧语音进行加窗操作，以加强每帧端点的连续性。本文采用 Hanming 窗来对语音帧进行加窗操作。

（3）傅里叶变换与计算能量谱

由于语音波形在时域上几乎没有描述能力，所以在分帧后需将每一帧语音都进行傅里叶变换 FFT（Fast Fourier Tranformation），傅里叶变换的意义在于将语音信号在时域上的波形图转化为频域谱图[34]。波形图与频谱图对比如下图 2.3 所示。

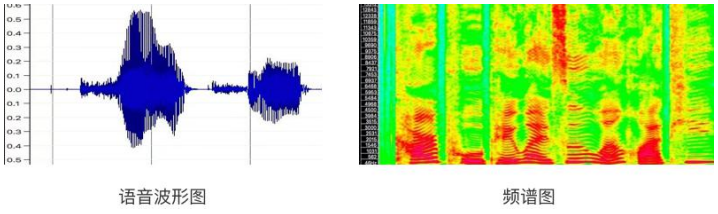

图 2.3 波形图与频谱图对比图

傅里叶变换的过程如下，首先定义傅里叶变换的输出值为 X(K)，离散傅里叶变换的表达式如式 2-2 所示。

$$\mathrm{X(k)} = \sum_{j=1}^{N} x(j) w_N^{(j-1)(k-1)} \tag{2-2}$$

其中，$x(j)$为输入的时域信号，N 为傅里叶变换点数。

得到频谱后需要计算每帧的能量谱。原因是当人类在接收到语音信号后，耳蜗会根据语音信号的频率震动来构成不同的点，从而刺激不同的听觉神经。计算能量谱正是依据人耳这种听觉特性，模拟耳蜗以获取每帧语音中的频率构成。能量谱的计算相对简单，将频谱的幅度取平方即可，表达式即为$X_{(k)}^2$。

（4）梅尔滤波

在能量谱中，包含很多跟内容识别无关的信息，所以需要对能量谱进行滤波，得到滤波带能量。因此使用梅耶尔滤波器对能量谱进行干扰信息的过滤。通常滤波器个数为 20-40 组。假定滤波器数量为 Q，滤波器的输出为 H(k)。

（5）对滤波器组输出取对数

此操作同样是模拟了人耳听觉系统对声音信号的处理方式，对于人类听觉系统来说，声音强度和声音自身的响度之间近似呈对数关系。则第 q 个滤波器组输出的对数能量如式 2-3 所示。

$$S_{(q)} = \ln\{\sum_{k=0}^{N-1} |X_{(k)}|^2 H_q(k)\} \tag{2-3}$$

（6）离散余弦变换 DCT（Discrete Cosine Transorm）

由于相邻滤波带之间是有重叠的，所以在对滤波器组输出进行离散余弦变换后，各维信号之间的相关性就会被消除，信号也都会映射到低维空间内。如此可以方便对 GMM 模型的对角协方差矩阵分布进行建模。

2．声学模型

在传统的语音识别框架中，声学模型采用 GMM-HMM 进行建模。声学模型的作用是把一系列 MFCC 特征识别成对应的隐马尔科夫模型的状态序列。这个过程中有两个概率需要学习，一是把当前时刻的一帧 MFCC 特征向量识别为当前时刻状态的概率，即 GMM 模型参数中的均值向量和协方差矩阵。二是上一时刻的状态转化为当前时刻状态的概率，即状态转移概率。

从一条语音特征向量转化一条音素串，本质上来讲是一个序列转化为另一个序列的过程，所以理论上来说路径条数会有指数级种，所以在声学模型解码过程中会采用 Viterbi 算法。Viterbi 算法的原理为每一帧只取概率最高的状态作为识别结果（本节所涉及的算法会在第三章中详细阐述）。GMM-HMM 模型结构如下图 2.4 所示。

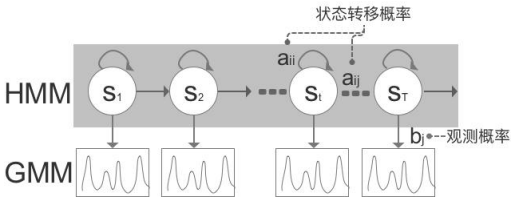

图 2.4 GMM-HMM 模型结构图

GMM-HMM 声学模型的训练过程为，HMM 利用每条训练数据和其对应的音素串，通过 Viterbi 算法来不断迭代更新 GMM 中每个状态的均值向量和协方差矩阵以及 HMM 中的状态转移概率，直到模型收敛。

3．词典

词典的作用是将 GMM-HMM 模型生成的音素串，对应的解码成汉字。图 2.5 展示了本文数据的部分词典。

```
<SPOKEN_NOISE> sil
一事无成 ii i2 sh ix4 uu u2 ch eng2
一劳永逸 ii i4 l ao2 ii iong3 ii i4
一呼百应 ii i4 h u1 b ai3 ii ing4
万丈高楼平地起 uu uan4 zh ang4 g ao1 l ou2 p ing2 d i4 q i3
七零八碎 q i1 l ing2 b a1 s ui4
万事大吉 uu uan4 sh ix4 d a4 j i2
三个臭皮匠赛过一个诸葛亮 s an1 g e4 ch ou4 p i2 j iang5 s ai4 g uo4 ii i1 g e4 zh u1 g e3 l
iang4
三人行则必有我师 s an1 r en2 x ing2 z e2 b i4 ii iu3 uu uo3 sh ix1
上半部分 sh ang4 b an4 b u4 f en4
不安好心 b u4 aa an1 h ao3 x in1
```

图 2.5 部分词典展示图

4．语言模型

每一门语言除了有自己的发音特点外，更重要的是语言中包含语义，所以如果直接把声学模型的识别结果单独输出表示为文字序列，那么结果往往因为训练目标与整体目标不一致而不尽如人意。故语言模型的作用就是把声学模型结合词典识别出的各个单词，纠正为正确的句子。

传统语音识别框架中的语言模型，通常采用 N-gram 语言模型。N-gram 语言模型将语音数据对应的文字串，以条件概率为理论基础构建为一个马尔科夫模型（Markov Model）。N-gram 语言模型的理论为只考虑之前单词中最近的若干个词。（N-gram 语言模型的理论会在第三章详细阐述）

2.2 神经网络语音识别框架结构分析

2.2.1 Tandem 框架结构分析

Tandem 结构是最早将神经网络应用于语音识别实验的一种框架结构，如下图 2.6 所示。

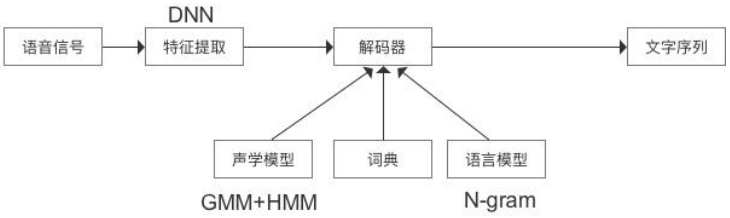

图 2.6 Tandem 识别框架结构图

在早期应用神经网络进行语音识别实验时，神经网络仅仅用在了语音识别的前端，即特征提取这一步骤。Tandem 结构中，利用神经网络提取语言特征，网络的瓶颈层特征替代了原本人工设计提取的语音特征。

Tandem 结构虽然将神经网络应用于语音识别，但还并未将其真正应用到语音识别框架的后端主体。Tandem 结构中 DNN 的瓶颈特征提取过程如下图 2.7 所示。

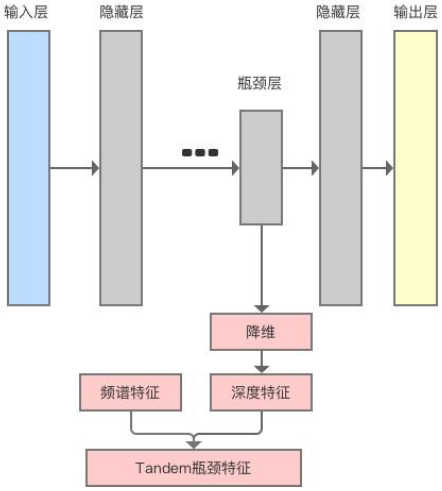

图 2.7 瓶颈特征提取图

神经网络提取特征原理简述如下，使用 triphone 状态标签作为训练目标对网络进行训练，训练时提取出录音中每一帧语音的深度特征。深度特征的维数通常与频谱特征维数保持一致。这些深度特征利用瓶颈层，反过来用于训练后端分类器，如 GMM-UBM 或 PLDA。将对应于给定语音帧的深度特征与频谱特征相结合，形成瓶颈特征。

图 2.7 中的特征提取过程总结为：

输入层：滤波器组输出的语谱图的包络结构。

隐藏层：首先对网络进行前向训练，并对瓶颈层进行降维处理，获取的 Deep Feature 结合 PLP 特征。

输出层：输出上下文有关音素的概率分布。

在 Tandem 结构框架中，虽然利用神经网络可以不再使用人为设计的语音特征，但音素 DNN 训练的状态校准却依然需要使用 GMM 模型生成，即说每一帧属于哪个上下文有关音素需要首先训练一遍传统的 GMM-HMM 模型[35]。

2.2.2 Hybrid 框架结构分析

Hybrid 结构是利用 DNN 代替原来的混合高斯模型来求每一帧属于各个音素标签的概率，然后用 HMM 结合 Viterbi 算法解码出音素序列。Hybrid 结构如图 2.8 所示。

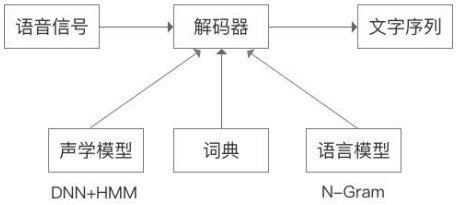

图 2.8 Hybrid 识别框架结构图

图2.8中，Hybrid结构最大的改进为不再需要特征提取，而声学模型的建模过程也从GMM-HMM模型优化为DNN-HMM模型[36]。在本章2.1节已经介绍过，ASR的传统模型应用GMM来拟合每段语音中的特征向量分布，同时利用GMM模型作为HMM模型中隐状态序列输出观测序列的概率分布。

由于GMM模型中参数的估计采用了EM算法（第三章中会详细阐述），导致模型在最大化目标序列的同时，其他信息序列（信息序列来自于最大似然输出的n-bestlist或者lattice）的概率可能也会被最大化。所以一些区分式训练方法，例如最大化互信息MMI（Maximum Mutual Information）、最小化分类误差MCE（Minimum Classification Error）、最小化音素误差MPE（Minimum Phone Error）等，开始逐渐成为了语音识别框架中的基本功能[37-39]。然而这些区分性训练的性能却受限于GMM模型天然的发散性分布，DNN-HMM声学模型结构如下图2.9所示。

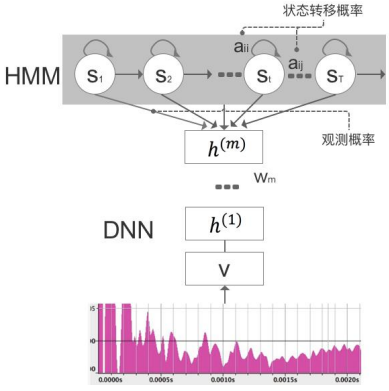

图 2.9 DNN-HMM 声学模型结构图

DNN-HMM这种声学模型的基础是使用强制对齐来获取训练神经网络的帧级标记。DNN-HMM与传统的GMM-HMM声学模型的关键区别在于两点。一是无需再进行人为设计语音特征导致特征信息丢失；二是DNN-HMM可以完成对建模单元的优化，例如，DNN-HMM在构建基于上下文信息的triphone模型时，DNN可以通过调节参数共享来训练上下文音素标签。在Hybrid结构中GMM模型除了需要在训练DNN时提供每一帧与上下文相关音素的对应关系外，声学模型中已经完全摒弃了GMM模型。

2.3 非完全端到端框架结构设计

2.3.1 端到端语音识别框架分析

应用神经网络进行声学建模时，无论使用的是普通的前馈神经网络还是具有记忆性的循环神经网络，最后的输出本质上都是多类判别器。当输入一帧带有上下文信息的语音信号时，神经网络的输出为判别该帧语音属于哪个上下文有关音素的标签。这与基于HMM 的识别模型不同，HMM 在训练时为网络提供每个音素的起止时间，在解码时HMM 为网络提供每一帧属于哪个状态，以及各个状态间的转移概率。但是基于 HMM 建模的语音识别模型，无法保证信息的长时依赖性，这主要是由于 HMM 的马尔科夫性，HMM 的每个状态的转移概率都只跟前一个时刻状态有关。而端到端的语音识别模型恰好解决了 CE 准则（基于 HMM 模型的语音帧对齐标准）的“硬对齐”要求，在端到端语音识别模型中，模型的对齐方式遵从 CTC 准则的“软对齐”，即不再需要知道每一帧语音具体属于哪一个上下文有关音素。端到端语音识别模型如下图 2.10 所示。

图 2.10 端到端语音识别框架结构图

图 2.10 中，端到端的语音识别框架抛弃了传统的 HMM 模型，极大的简化了语音识别框架的复杂度。对于 CTC 准则来说，建模单元的选择也相对灵活，无论是音素、音节或者单词都可以作为 CTC 准则的建模单元。

2.3.2 非完全端到端框架结构设计

1．传统框架的缺点

虽然传统框架是语音识别的一种有效方法，但传统框架仍有一些缺点，具体为：

（1）框架的结构复杂。这不仅导致不同模型间相互独立，难以进行整体优化，而且在传统框架中将音素转换为单词这一步骤，需要结合发音词典，这样对于解码出的单词序列相当有局限性。例如，在模型预测时出现了词典中并未出现过的单词拼写规则，那么网络是无法输出正确答案的。

（2）语音特征是人为设计，极易丢失特征中的重要信息。在传统识别框架时期，语音特征的提取是人依据语音学知识设计特征提取的方法。在应对不同语种的不同语音时，人为设计的特征经常会丢失一些重要特征。

（3）语音帧与音素间的硬性对齐要求。在传统框架中，语音帧与音素间必须一一对应。在实验中，第一轮迭代往往会将语音帧平均分配，在以后的迭代中模型自学习语音帧与音素间的对齐参数。显然在迭代的过程中只能将参数控制在误差范围内，并不能完全对应。

2．神经网络框架的缺点

无论是Hybrid框架还是Tandem结构，都没有将神经网络强大的模型拟合能力完全展现。并且在实际训练时，模型都需要首先训练一遍传统框架模型，以获取神经网络与每帧语音间的对应关系。

3．端到端框架的缺点

虽然端到端的语音识别框架中音素不再需要结合词典进行解码，并且网络会记忆单词拼写的规则，在迭代中输出概率最大的序列，使框架不受词典的局限性的同时去除了传统框架一些不合实际的假设，但是在实际训练中也存在一些问题，具体为：

（1）端到端语音识别需要大量训练数据。

对于神经网络来讲，本身就需要大量的训练数据来保证模型的预测效果，而端到端的网络模型通常都采用较深层次的网络结构，所以对训练数据量的要求则更高。其次直观来讲，只有更多的训练数据才足以支撑语音识别的多个复杂流程。

（2）端到端语音识别难以利用纯文本训练语言模型。

由于传统的语音识别框架，声学模型与语言模型是分开训练的，而语言模型又通常选择 Bi-gram 语言模型，那么语言模型完全可以利用纯文本数据进行训练。而端到端的识别框架并不具有单独的语言模型，所以语言模型必须采用文本数据结合对应的声音数据进行训练。

（3）端到端语音识别对于包含不同发音特点的不同语种泛化能力差。

不同语种有不同的发音构词特点，很多语种（如西班牙语）是音形文字所听即所写，并且发音构词没有声调变化，语音与文字序列的映射关系也是单一映射。而中文的构字规则为象形文字，且语调变化丰富，发音与构字规则间也没有映射标准。如此导致在使用端到端模型进行中文语音识别时，模型效果差。所以往往使用端到端模型进行中文语音识别实验时，模型准确率并不理想。

4．非完全端到端框架设计

基于以上分析，本文针对端到端模型对中文语音识别的缺陷，设计了融合中文语言模型的语音识别框架，命名为非完全端到端（Incomplete-end-to-end）语音识别框架，框架结构如下图 2.11 所示。

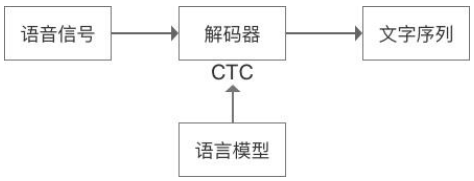

图 2.11 非完全端到端语音识别框架结构图

非完全端到端的语音识别框架主要采用外接中文语言模型的方法。在模型解码时，结合语言模型中的先验知识，对输出的文字标签进行二次打分。如此即保留了端到端框架不受词典的局限、Lable to Lable 输出以及建模单元灵活等优点，同时又解决了端到端语音识别难以用纯文本训练语言模型以及对中文一类文字识别率低的缺陷，关于本框架结合语言模型对输出标签二次打分的详细算法原理会在第四章中阐述。

2.4 本章小结

本章首先分析了自语音识别发展以来几种主流的语音识别框架，其中包括传统语音识别框架以及融合了神经网络语音识别框架，在分析几种语音识别框架的同时，对每一种框架的具体模型结构和识别方法也给出详细的阐述。最后，针对端到端模型对中文识别率较低的问题，本章结合现有端到端框架与汉语发音特点，设计了非完全端到端语音识别框架，并对其解码方法和框架优势进行了阐述。

第三章 基于自适应技术的 GMM-HMM 算法模型

隐马混合高斯（GMM-HMM）模型作为最成熟的语音识别模型被延用至今，隐马尔科夫模型对于时序信息的建模能力在神经网络出现以前是最为强大的，并且直到现在很多工程化的语音识别模型仍然使用 GMM-HMM 模型。本章分析 GMM-HMM 模型对语音数据建模的基本建模原理和解码算法，基于 GMM-HMM 模型的实验结果将被作为本文方法的基线模型与基于神经网络时序分类的模型结果进行对比分析。本章设计融合了说话人自适应技术的隐马混合高斯模型以提高基线模型的准确度。

3.1 隐马尔科夫模型

3.1.1 马尔科夫模型

马尔科夫模型（Markov Model）是一种基于时间推移，产生两状态间随机转移过程的序列模型[40]。本章实验中所使用的 N-gram 语言模型即为马尔科夫模型。马尔科夫模型的建立是为了搜索序列随时间变化的模式，马尔科夫模型利用时间步骤、状态以及人为假设，建立了一个可以产生模式的过程模型。一个标准的马尔科夫过程有两个元素，分别是初始向量和状态转移矩阵。特殊说明的是，虽然马尔科夫假设是基于时序的模型，但其状态转移概率却是不随时间变化的。

马尔科夫模型的公式表示如下。首先模型定义 q_t 表示模型在 t 时刻的状态值，则 t 时刻的状态取值 S_j 的概率取决于前 t-1 个时间，表达式如式 3-1 所示。

$$p(S_j) = p(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k \ldots) \tag{3-1}$$

而马尔科夫模型假定 t 时刻的状态只与其前一时刻 t-1 相关，则模型即构成了一个离散的一阶马尔科夫链，本章实验的 Bi-gram 语言模型使用的就是一阶的马尔科夫链，表达式如式 3-2 所示。

$$p(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k \ldots) = q_t = p（S_j | q_{t-1} = S_i) \tag{3-2}$$

记状态间的转移概率为a_{ij}，则a_{ij}需满足以下两个条件，如式 3-3 所示。

$$\begin{cases} a_{ij} \geq 0 \\ \sum_{j=1} a_{ij} = 1 \end{cases} \tag{3-3}$$

定义模型初始概率$\pi_i = P(q_1 = S_i)$，则马尔科夫模型的状态序列$S_1, S_2, \ldots S_t$概率如式 3-4 所示。

$$P(S_1, S_2, \ldots S_t) = \pi_{s1} \prod_{t=1}^{t-1} a_{S_t S_{t-1}} \tag{3-4}$$

3.1.2 隐马尔科夫模型

第二章中已经阐述过应用 GMM-HMM 构建声学模型的基本思路，以下对算法的具体推导过程进行展开。

隐马尔科夫模型 HMM（Hidden Markov Model）是双层的马尔科夫随机过程[40]。同样 HMM 也是关于时序的统计概率模型，HMM 由一条随时间改变的隐状态序列，和每个隐状态对应生成的观测值串联的观测序列组成，隐马尔科夫网络结构如下图 3.1 所示。HMM 是基于时序的模型，每个状态可看作为一个时刻。HMM 模型的本质是一个序列分类器（Sequence Classifier），序列分类器的作用把一个某长度的序列识别成另一个长度的序列，所以 HMM 模型在 ASR、NLP 以及模式识别等领域都表现出了优异的性能。

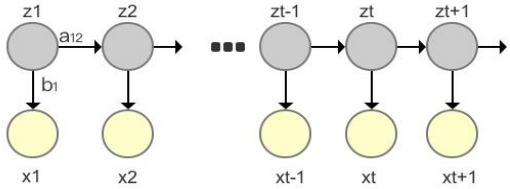

图 3.1 隐马尔科夫模型网络图

图 3.1 表示了一条完整结构的隐马尔可夫链，其中 X 表示观测序列，Z 表示状态序列，A 表示状态间转移概率矩阵，B 表示观测概率矩阵[41]。初始隐马尔科夫模型由三个概率分布确定，除上述的状态转移矩阵 A 和观测概率矩阵 B 外，还包含初始状态概率矩阵π。隐马尔科夫模型表示如式 3-5 所示。

$$\lambda = （A,B,\pi）\tag{3-5}$$

其中状态转移概率矩阵 A，需满足以下基本约束如式 3-6、式 3-7、式 3-8 所示。

$$a_{ij} = p\left(q_{t+1} = S_j \middle| q_t = S_i\right)\ i,j \in [1,N]\tag{3-6}$$

$$a_{ij} \geq 0\tag{3-7}$$

$$\sum_{j=1}^{N} a_{ij} = 1\tag{3-8}$$

观测状态矩阵 B 同样要满足上述基本约束，这里就不再赘述。

3.1.3 隐马尔科夫三个基本问题

HMM 模型中，有三个基本问题，分别为概率计算问题、解码问题和训练问题。针对不同的问题，HMM 分别需要不同的算法进行求解，下面分别阐述。

1．概率计算问题

概率计算问题也称为求值问题，是在已知HMM基本模型参数λ的条件下，求解观测序列O$\{o_1,o_2,...,o_T\}$生成的后验概率。求值问题可以用来评价一个给定的HMM模型λ与一

条给定的观测序列O之间的匹配度，在实际应用中可以利用求值问题来解决模式识别等问题。

（1）直接求解法

首先求解给定λ =（A,B,π）下，观测序列 O 的后验概率 $p(O|\lambda)$，定义模型的状态序列为 Q$\{q_1,q_2,...q_T\}$，此问题最直接的方法是将每个时刻的观测概率b_j与状态转移概率a_{ij}连乘再累加得出结果。表达式如式 3-9 所示。

$$p(O|\lambda)=\sum_{Q}p(O,Q|\lambda)=\sum_{Q}p(Q|\lambda)p(O|Q,\lambda) \tag{3-9}$$

将上式分别计算，则有给定 HMM 模型λ条件下，状态序列 Q 生成的概率表达式如式 3-10 所示。

$$p(Q|\lambda)=\pi_{q_1}a_{q_1q_2}a_{q_2q_3}...a_{q_{T-1}q_T} \tag{3-10}$$

在同时给定 HMM 模型λ和状态序列 Q 的条件下，观测序列 O 生成的概率表达式如式 3-11 所示。

$$p(O|Q,\lambda)=b_{q_1}b_{q_2}...b_{q_T} \tag{3-11}$$

将式观测序列 O 生成的概率表达式和状态序列 Q 生成的概率表达式带回到观测序列 O 的后验概率 $p(O|\lambda)$中，得出观测序列 O 的生成概率 $p(O|\lambda)$。如式 3-12 所示。

$$p(O|\lambda)=\sum_{Q}\pi_{q_1}a_{q_1q_2}b_{q_1}a_{q_2q_3}b_{q_2}...a_{q_{T-1}q_T}b_{q_T} \tag{3-12}$$

以上直接求解的方法虽然可以将观测序列 O 的生成概率求出，但存在一个问题，对于 HMM 模型λ来说，假定λ有包含 N 个隐含状态，时间长度为 T，那么此方法计算的时间复杂度就为N^T，这样来看一条呈指数级增长的搜索路径计算量是相当大的。解决此问题的方法为下一部分讲述的前向-后向算法。

（2）前向-后向算法

前向-后向算法的基本思想，是模型每经过一个时刻，都计算一步该时刻的状态概率，在每一步递推的过程中求解出状态为q_T时，观测序列 O$\{o_1,o_2,...,o_T\}$的生成概率。

1）前向算法

首先利用前向算法计算，定义前向变量$\alpha_t(i)$。$\alpha_t(i)$表示在 t 时刻，部分观测序列为O$\{o_1,o_2,...,o_T\}$并且状态为S_i的概率。表达式如式 3-13 所示。

$$\alpha_t(i)=P(o_1o_2...o_T,q_T=S_i|\lambda) \tag{3-13}$$

然后通过递推计算前向变量$\alpha_t(i)$，以及观测序列的概率 $p(O|\lambda)$如式 3-14 所示。

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)\,a_{ij}\right]b_{jo_{t+1}} \tag{3-14}$$

下图 3.2 直观的表示了前向算法从 t 时刻递推到 t+1 时刻计算前向概率的过程。

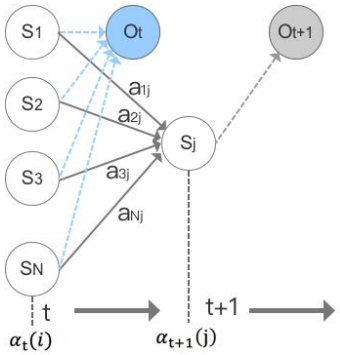

图 3.2 前向算法递推过程图

总结来说，前向算法包含三个步骤：

a．模型初始化，即给定初始的状态概率$\alpha_1(i) = \pi_i b_{io_1}$。

b．通过循环计算，递推每一时刻的前向概率$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_{jo_{t+1}}$

c．将每一步结果累加，得出最终结果 $p(\mathrm{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

前向算法在每计算一步$\alpha_t(i)$时，都只考虑从 t-1 时刻的所有 N 个状态转移得来的概率和自己本身输出的观测概率，故时间复杂度为N^2T，相较于直接计算，前向算法大大缩减了时间复杂度。

2）后向算法

后向算法在算法思路上与前向算法并无二致，只是区别于算法的搜索路径方向，后向算法的递推方向如图 3.3 所示。

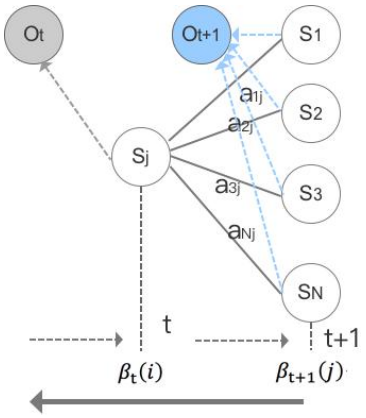

图 3.3 后向算法递推方向图

前向算法的搜索路径为从第一时刻起始递推到最后时刻，然后向算法的路径归纳方向为$\beta_t(x)$,$\beta_{t-1}(x)$,...$\beta_1(x)$。

首先定义前向变量$\beta_t(i)$，$\beta_t(i)$表示从 t 时刻到 t+1 时刻模型由状态S_i转移到S_j，并S_j输出O_{t+1},在 t+1 时刻，模型输出部分观测序列 O$\{o_1,o_2,...,o_T\}$，并且状态为S_j的概率，表达式如式 3-15 所示。

$$\beta_{\mathrm{T}}(i) = p(o_{\mathrm{t+1}}o_{t+2}...o_T,q_T = S_i|\lambda) \tag{3-15}$$

对于后向算法的计算过程同样总结为三步：

a．模型初始化，$\beta_{\mathrm{T}}(i) = 1$

b．通过循环计算，递推每一时刻的后向概率$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_{jO_{t+1}}\beta_{t+1}(j)$

c．将每一步结果累加，得出最终结果 $p(\mathrm{O}|\lambda) = \sum_{i=1}^{N} \pi_i b_{iO_1}\beta_1(i)$

2．解码问题

解码问题也称为预测问题，解码问题的本质是搜索一条最优状态序列可以最好的解释观测序列。即已知 HMM 模型λ＝（π，A，B）以及观测序列 O$\{o_1,o_2,...,o_T\}$，计算在给定观测序列条件下，最大状态序列的条件概率 p(S|O，λ)。解码问题在语音识别应用的非常广泛，在声学模型求解每帧语音对应的音素状态时，应用的就是 HMM 的解码问题。在结合声学模型和语言模型生成文字序列时，同样应用了 HMM 的解码问题。

（1）近似算法

同求值问题一样，解码问题同样可以根据直接计算来求得最优的隐状态序列。基本原理为，在每一个时刻 t 都搜索概率最大的状态值q_t，以此来得到一条状态序列$S^*\{S_1^*,S_2^*,...S_T^*\}$。

首先定义在给定模型λ条件下，t 时刻处于状态S_i的概率为$\gamma_t(i)$，则$\gamma_t(i)$的表达式如式 3-16 所示。

$$\gamma_t(i) = p(q_t = S_i|\mathrm{O},\lambda) = \frac{p(q_t = S_i,\mathrm{O}|\lambda)}{p(O|\lambda)} \tag{3-16}$$

将第一个求值问题中的前后向概率代入上式，则重写$\gamma_t(i)$式 3-16 如式 3-17 所示。

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\,\beta_t(i)} \tag{3-17}$$

虽然直接应用近似算法可以求解出一条每个节点都概率最大的路径，但这种优化每一个节点而不是优化整条序列的解码方式，在实际应用中，这种解码算法的实用性非常差。因此解决此类问题最常应用的是下一节提到的 Viterbi 算法。

（2）Viterbi 算法

Viterbi 算法的基本思想，是基于序列去搜索最优的状态序列，首先定义一个 Viterbi 变量$\delta_t(i)$, $\delta_t(i)$表示在 t 时刻状态为 i，并且输出观测序列 O$\{o_1,o_2,...,o_T\}$的所有路径中，概率的最大值。表达式如式 3-18 所示。

$$\delta_t(i) = \max_{q_1...q_{t-1}} p(q_1,q_2,...q_t = S_i|o_1,o_2,...,o_t,\lambda) \tag{3-18}$$

与前向-后向算法一样，Viterbi 算法同样需要在求出模型初值后，进行逐个时刻递推，具体为：

a．给定模型初值，$\delta_1(i) = \pi_i b_{io_1}$

b．将状态递推至下一时刻，$\delta_t(i) = \max_{1 \le j \le N}(\delta_t(j)a_{ji})b_{io_{t+1}}$

c．得出最终结果$P^* = \max_{1 \le j \le N}\delta_T(i)$

3．训练问题

训练问题又称学习问题，训练问题是根据可观测的观测序列 O$\{o_1,o_2,...,o_T\}$，估计最优 HMM 模型参数λ =（π，A，B）的问题，即求概率 p(O|λ)。训练问题本质上来讲，就是根据部分样本估算整体模型参数的问题，那么很自然的应用最大似然估计 MLE（Maximum Likelihood Estimation），由于 HMM 中的状态序列在模型中为未知的隐变量，所以训练问题则需要应用含有隐变量的最大似然估计，即 EM 算法，此算法会在下一小节 3.2 中阐述，本节就不再详细推导。

3.2 混合高斯模型参数估计

混合高斯分布模型 GMM（Gaussian Mixture Model）是一种对随机变量的分布情况进行描述的模型，有着十分广泛的应用[42]。中心极限定理指出，大量随机变量的分布会趋于高斯分布。换言之，当随机变量的数量非常大时，可以使用多个高斯分布模型来对其分布进行拟合。在语音识别中采用混合高斯分布对语音帧在高维空间中的分布进行建模。下图 3.4 为具有两个高斯分量的混合高斯模型三维模型图。

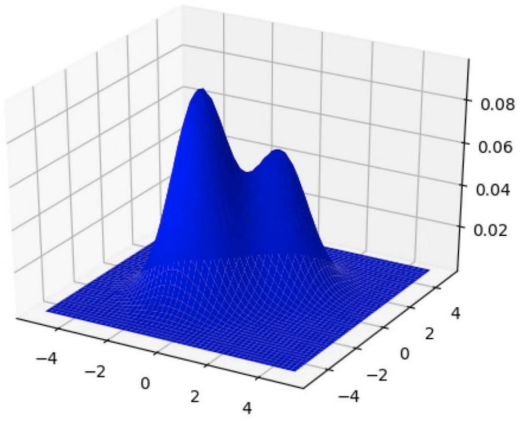

图 3.4 二元混合高斯模型三维展示图

1．期望最大化算法

混合高斯分布模型的参数所使用的算法为期望最大化算法 EM（Expectation Maximization）。概率模型的参数估计通常是用极大似然估计法，而期望最大化算法可以看作是对极大似然估计法的延伸，用于解决在模型的参数估计中存在隐含变量时极大似然估计法无法得到解析解的难题。

（1）琴生不等式

期望最大化算法使用琴生不等式（Jensen's Inequality），琴生不等式函数图像如下图 3.5 所示。

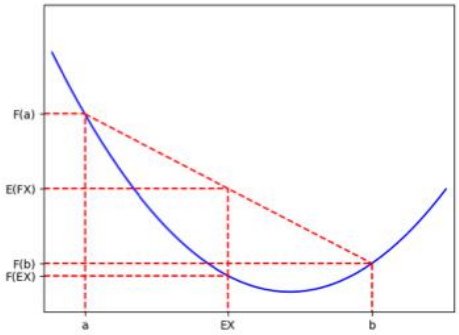

图 3.5 琴生不等式函数图

琴生不等式表达式如式 3-19 所示。

$$\mathrm{E(F(X))} \geq \mathrm{F(EX)} \tag{3-19}$$

其中：

$$\mathrm{F(r} * x_1 + (1-\mathrm{r}) * x_2) \leq \mathrm{r} * \mathrm{F}(x_1) + (1-\mathrm{r}) * \mathrm{F}(x_2) \tag{3-20}$$

其中 F 是关于变量 x 的函数 $\mathrm{x} \in \mathrm{L}$，$x_1 < x_2 \in L$,r 为任意实数，F 被称为 L 上的凹函数。假设 X 是一个随机变量，F 是一个凹函数。

（2）期望最大化算法

假如使用模型 $p(x,z)$对观测序列$\{x_1,x_2,x_3 \ldots x_n\}$（假定各个观测样本相互独立）进行拟合，在估计参数时，构建似然函数如式 3-21 所示。

$$\ell(\theta) = \sum_{i=1}^{m} \log p(x_i;\theta) = \sum_{i=1}^{m} \log \sum_{z} p(x_i,z;\theta) \tag{3-21}$$

因为似然函数中包含了隐含随机变量，所以难以直接对似然函数求极值。如果隐含变量（每个观测样本所属的类别 z）已知的话，那么似然函数的极值就会比较容易求得。期望最大化算法基于上述思路，通过逐步迭代来寻找似然函数的极大值。

首先定义Q_i是关于 z 的一个分布（$\sum_z Q_i(z) = 1, Q_i(z) \geq 0$），将其代入似然函数如式 3-22 所示。

$$\sum_{i} \log \sum_{z} p\left(x_i, z^{(i)};\theta\right)$$

$$= \sum_i \log \sum_{z^{(i)}} Q_i(z^{(i)}) \frac{p(x_i, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

$$\geq \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x_i, z^{(i)}; \theta)}{Q_i(z^{(i)})} \quad (3\text{-}22)$$

其中上式引入概率分布Q_i，最后一步应用琴生不等式。（log(x)是凸函数，所以变量的期望的函数值大于或等于变量函数值的期望）对于任意上述Q_i分布函数，$\sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x_i, z^{(i)}; \theta)}{Q_i(z^{(i)})}$即为似然函数$\ell(\theta)$的下界。

在期望最大化算法中，将$Q_i(z^{(i)})$定义为在给定模型参数θ的情况下，第i个观测值是由z^i生成的后验概率，表达式如式 3-23 所示。

$$Q_i(z^{(i)}) = \frac{p(x_i, z^{(i)}; \theta)}{\sum_z p(x_i, z; \theta)} = \frac{p(x_i, z^{(i)}; \theta)}{p(x_i; \theta)} = p(z^{(i)} | x_i; \theta) \quad (3\text{-}23)$$

期望最大化算法是个迭代算法，每次迭代有两个步骤，E步（E-step）和M步（M-step）。在 E 步，基于上一轮估计出的参数（或者对于第一轮迭代，参数是初始化得到的），计算$Q_i(z^{(i)})$；在 M 步，将计算得到的$Q_i(z^{(i)})$代入式 3-22，并求出使其最大化的参数；如此迭代，直至收敛。总结算法如下所示。

期望最大化算法：

do {

$Q_i(z^{(i)}) = p(z^{(i)} | x_i; \theta)$; (E-step)

$\theta := \arg\max \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x_i, z^{(i)}; \theta)}{Q_i(z^{(i)})}$; (M-step)

} while (not converged)

在实际训练过程中，通常会比较相邻两次迭代的似然函数值，如果似然函数值的增长小于阈值，就会认为算法达到收敛状态而终止迭代。

（3）混合高斯分布模型的参数估计

混合高斯分布模型表达式如式 3-24 所示。

$$p(x) = \sum_j w_j * N(x; \mu_j, \Sigma_j) \quad (3\text{-}24)$$

其中($N(x; \mu_j, \Sigma_j)$为高斯分布模型)。对混合高斯分布模型的参数估计时，首先构建似然函数如式 3-25 所示。

$$\ell(w,\mu,\Sigma)=\sum_{i=1}^{m}\log p(x_i;w,\mu,\Sigma)$$

$$=\sum_{i=1}^{m}\log\sum_{z}p(x_i,z;w,\mu,\Sigma) \tag{3-25}$$

似然函数中存在隐含变量（高斯分量 z），因此运用期望最大化算法。

1）在 E 步，直接求样本x_i由第 j 个高斯分量生成的后验概率如式 3-26 所示。

$$w_j^i=Q_i(z^{(i)}=j)=p(z^{(i)}=j|x_i;w,\mu,\Sigma)。 \tag{3-26}$$

2）在 M 步，最大化似然函数如式 3-27 所示。

$$\ell(w,\mu,\Sigma)=\sum_{i=1}^{m}\sum_{z^i}Q_i(z^i)\log\frac{p(x_i,z^i;w,\mu,\Sigma)}{Q_i(z^i)}$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{k}Q_i(z^i=j)\log\frac{p(x_i,z^i=j;\mu,\Sigma)p\left(z^{(i)}=j\middle|w\right)}{Q_i(z^i=j)}$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{k}w_j^i\log\frac{p(x_i,z^i=j;\mu,\Sigma)w_j}{w_j^i} \tag{3-27}$$

通过求似然函数对w_j、μ_j和Σ_j的偏导数，可以得到式 3-28、式 3-29、式 3-30。

$$w_j:=\frac{1}{m}\sum_{i=1}^{m}w_j^i \tag{3-28}$$

$$\mu_j:=\frac{\sum_{i=1}^{m}w_j^i x_i}{\sum_{i=1}^{m}w_j^i} \tag{3-29}$$

$$\Sigma_j:=\frac{\sum_{i=1}^{m}w_j^i(x_i-\mu_j)(x_i-\mu_j)^T}{\sum_{i=1}^{m}w_j^i} \tag{3-30}$$

在混合高斯分布的参数估计时，通常可以通过设定最大迭代次数，即达到该最大迭代次数就终止参数更新。

3.3 基于自适应技术的 GMM-HMM 模型

3.3.1 最大似然线性回归自适应算法

最大似然线性回归方法的过程为，在说话人无关模型训练完成后，将已训练的高斯混合模型参数矩阵融合说话人特性参数矩阵进行线性变换，后依据 MLE 估计出新的高斯混合模型参数矩阵。新的参数矩阵可以由 SI（Speaker Independent）模型的参数矩阵线性表示。SI 模型中的参数矩阵记为 u，如式 3-31 所示。

$$\mathrm{u}=[u_1,u_2,...u_n]^T \tag{3-31}$$

自适应模型中的参数矩阵记为$\hat{u}$。则自适应变换后的表达式如式 3-32 所示。

$$\hat{u} = Wu + b \tag{3-32}$$

其中，b 表示偏移矩阵，W 为线性变换矩阵。将参数矩阵记为$\hat{U}$，则$\hat{U}$ = [b,W]。

利用最大似然线性回归方法对非特定人声学模型进行训练，假定说话人数为 i，利用自适应前说话人无关模型的参数矩阵为λ，训练目标是更新权值矩阵$\hat{U}$，使得模型的输出概率最大。表达式如式 3-33 所示。

$$\left(\hat{U}，\hat{\lambda}\right) = \text{argmax}\prod_{i=1}^{I}\prod_{k=\beta_{(i)}} P(O_{(t)}|\lambda,U_i) \tag{3-33}$$

其中，上式中$\beta_{(i)}$表示第 i 人声音数据的集合，重估后的变换矩阵$\hat{U}$（高斯混合模型中的均值和方差），就是自适应后新的声学模型参数。

3.3.2 自适应技术优势

对语音识别技术来说，说话人的语音数据由于方言、性别、语速、发音习惯以及心情状态等音素，不同场景和不同说话人发出同一语音的频域谱图特性也不尽相同，这使得训练模型的识别性能大大降低。解决这一问题若使用子空间法（i-vector）进行实验虽然在识别效率上有所提升，但需要大量说话人相关 SD（Speaker Dependent）数据以提取特定说话人的声学特征[43]。故本文设计采用最大似然线性回归 MLLR（Maximum Likelihood Linear Regression）的自适应训练 SAT（Speaker Adaptive Training）方法来调整说话人无关模型的参数，以此提升非特定说话人模型的识别准确率。

3.4 本章小结

本章首先从马尔科夫模型引例详细阐述了隐马尔科夫模型的模型结构、三个基本问题以及对应的解码算法。然后从琴生不等式、EM 算法开始推导了混合高斯模型的参数估计算法。最后为提升基线模型的准确率，本文设计了结合自适应技术的隐马混合高斯模型，并对其原理和优势进行了阐述。

第四章 基于 Bi-PLSTM 的链接时序分类算法模型

链接时序分类算法是一种通过优化神经网络损失函数，提升神经网络训练和解码性能的方法。本章首先研究传统循环神经网络以及其变种长短期记忆时序网络的网络结构和信息传播过程，接下来针对长短期记忆时序网络的缺陷，对其内部单元结构进行优化改进。然后研究链接时序分类方法的训练与解码过程，为了进一步提升链接时序分类算法的识别性能，本章设计利用双向网络结构替代传统单向长短期记忆时序网络与链接时序分类算法相结合，进而优化模型准确率。

4.1 循环神经网络

1．循环神经网络基本结构

循环神经网络 RNN（Recurrent Neural Network）是一类用于处理序列数据的神经网络。大数据背景下很多序列型的数据，如文本、语音以及视频等，往往都在时序上具有关联性。若使用第一章阐述的前馈神经网络处理序列任务，由于前馈神经网络不具有记忆性，则会导致之前时刻的输出无法传递到后面的时刻，序列间的关联性不能被处理。而 RNN 与前馈神经网络的仅计算当前时刻网络的输入值不同。RNN 相较于一般的全连接神经网络增加了记忆单元，使神经网络在处理时序信号时拥有有限的短暂记忆能力[44]。

RNN 网络通过使用循环连接，使信息可以在网络神经元中循环一段时间，此特性来源于 RNN 的特殊网络结构。RNN 网络结构具有输入层 X、上下文层 S（即隐藏层）和输出层 Y，循环神经网络的结构如下图 4.1 所示。

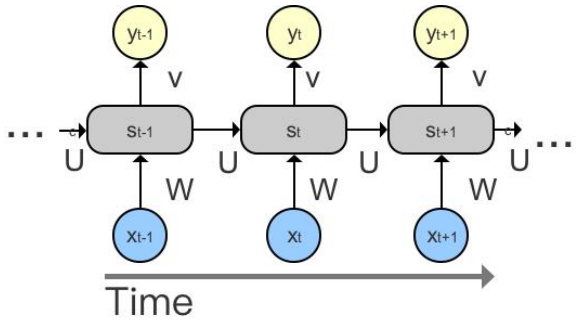

图 4.1 循环神经网络结构图

RNN 网络中，信号通过正向传输，在时刻 t 网络的输入为 x(t)，输出为 y(t)，s(t)为隐藏层的状态。s(t)的输入等于当前状态 x（t）的输入与 s（t-1）的输出之和。

2．循环神经网络的传播过程

循环神经网络在神经元中添加了记忆单元，故 RNN 的信息传播过程与普通的前馈神经网络有所不同，在前向传播过程中，如图 4.1，每一个隐藏层神经元的输入都包含前一时刻的输出，以及当前时刻的输入。

（1）信号的正向传播过程

定义输入序列 X，标准序列 D，输出序列 Y，上下文层输出为h_t，上下文层输入为s_t。则网络的前向传播过程如下:

定义输入层到上下文层权重矩阵为 W，则有隐藏层输入向量如式 4-1 所示。

$$s_t = Uh_{t-1} + Wx_t \tag{4-1}$$

定义上下文层激活函数为 $f(z)$，则有上下文层输出向量如式 4-2 所示。

$$h_t = f(s_t) \tag{4-2}$$

定义上下文层到输出层的权重矩阵为 V，则输出层输入向量如式 4-3 所示。

$$z_t = Vh_t \tag{4-3}$$

g（z）为输出层激活函数，选用 softmax 函数，则网络输出如式 4-4 所示。

$$y_t = softmax(z_t) \tag{4-4}$$

如在处理大量数据时，初始化 s(0)可以设置一个较小的值。这样在下一个时间步长中，s(t+1)就只相当于复制了 s(t)的信息。把输入向量 x(t)在时间 t 的输出，同时传入下一层的隐藏层。

（2）误差的反向传播过程

RNN 的误差后向传播过程采用基于时间反向传播算法 BPTT（Backpropagation though time）[45]。BPTT 与前馈神经网络的反向传播算法区别在于，将损失在时间维度上进行了两次累加。

由于输入的时间序列在每个时刻都有输出，所以网络对于输入向量网络的损失就为每一时刻的损失之和。定义损失函数为 L，则 L 表达式如式 4-5 所示。

$$\mathrm{L} = \sum_{t=1}^{T} L_t \tag{4-5}$$

网络损失函数选取交叉熵损失函数，则有：

$$L_t = -\, d_t log^{y_t} \tag{4-6}$$

首先对输出层权重 V 进行梯度优化，显然此步骤与普通的前馈层神经网络的参数优化并无二致，表达式如式 4-7 所示。

$$\frac{\partial L_t}{\partial v_{ij}} = (d_t - y_t)_i (h_t)_j \tag{4-7}$$

将每一时刻的损失累加得到整个网络的损失，表达如式 4-8 所示。

$$\frac{\partial L}{\partial V} = \sum_{t=1}^{T} (y_t - d_t) \otimes h_t \tag{4-8}$$

对隐藏层权重 U 进行梯度优化，由于 RNN 的记忆性导致隐藏层权值 U 不止与 t 时刻有关，而是与之前所有时刻的隐状态都有关联，所以，要将优化的梯度在时间维度上进行累加，表达式如式 4-9 所示。

$$\frac{\partial \mathrm{L}_t}{\partial U} = \sum_{k=1}^{t} \frac{\partial L_t}{\partial s_k} \cdot \frac{\partial s_k}{\partial U} \tag{4-9}$$

将上一式进一步在s_{k+1}处根据链式求导法则展开后，带入隐藏层对其输入向量的导数值可进一步得出式 4-10。

$$\frac{\partial L}{\partial U}=\sum_{t=1}^{T}\sum_{k=1}^{t}\frac{\partial L_t}{\partial s_k}\otimes h_{k-1} \tag{4-10}$$

输入层权值矩阵梯度优化，与隐藏层方法类似，结构表达式如式 4-11 所示。

$$\frac{\partial L}{\partial W}=\sum_{t=1}^{T}\sum_{k=1}^{t}\frac{\partial L_t}{\partial s_k}\otimes x_k \tag{4-11}$$

N-gram 语言模型中第 n 个单词只计算 n-1 时刻的条件概率，通过 RNN 构建的语言模型与 N-gram 语言模型不同。RNN 网络的最主要特点是记忆单元对当前时刻以前的序列信息保有记忆性。虽然 N-gram 语言模型在语音识别任务中也表现出了优异的性能，但这种建模方式，在语法特点上也暴露了一些弊端。原因主要是语音识别是对全局处理和时间依赖都要求非常高的序列任务，然而在建立语言模型时只考虑前一个单词的条件概率，确实会丢失很多有效信息。而 RNN 的模型结构，使某一时刻网络的输出除了与当前时刻的输入相关之外，还与之前某一时刻或某几个时刻的输出相关，这样的模型结构在语法表达上更加符合人类的语言特点。（例如，在语言模型解码过程中，如果在句子的开头出现了“因为”，那么在句中出现“所以”的概率就会增加。）

3．循环神经网络的优点

循环神经网络的优点除了可以保留序列信息的长时依赖性外，其优点还有 RNN 的网络结构可变性强，适合处理多种序列任务。RNN 的网络结构灵活，输入与输出的对应关系大体分为以下四种方式，如下图 4.2 所示。

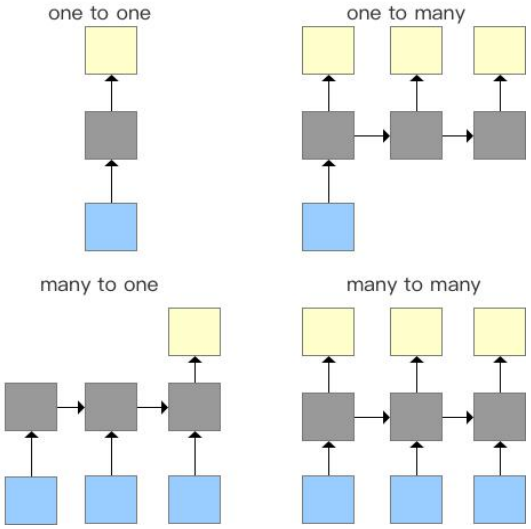

图 4.2 RNN 的可变性结构图

（1）单一输入对应单一输出，此种模型结构多用于处理单一结构标签的问题。

（2）单一输入对应多个输出，此种模型结构多用于图片的对象识别，即输入为图片，输出为一串文字序列。

（3）多个输入对应单一输出，此种模型结构多用于二分类问题，如情感分析，文本分类等。

（4）多个输入对应多个输出，此种模型结构多用于机器翻译，如语言间的文本转换。

正由于 RNN 的网络结构灵活，对于输入与输出的单元没有严格的要求，所以在语音识别实验的时候，输入和输出的数据都可以是不定长的。

4．循环神经网络的缺陷

在神经网络中，隐藏层的深度反映出训练数据的量，大量的数据则需要更多的隐藏层。而在网络梯度下降过程中，随着隐藏层的加深、计算向量也同时增加，数据维数就会呈指数级增长。RNN网络的输出又是同时受到当前时刻的输入和之前时刻网络状态的影响，所以如果出现长期依赖关系，那么网络就极易产生梯度消失和梯度爆炸问题。

在函数梯度较大时，当多个权重矩阵相乘，参数更新后的数值超出了取值的有效范围时，就会产生梯度爆炸[46]。而在函数梯度较小，参数更新缓慢，则会产生梯度消失[47]。梯度消失与梯度爆炸函数示意图如下图4.3所示。

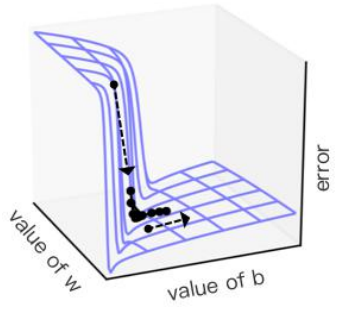

图 4.3 梯度消失与梯度爆炸示意图

对于梯度爆炸问题，在解决时可以直接采用梯度截断的方法，来给梯度设置阈值。而梯度消失的解决办法则稍微复杂。梯度对于深层网络的第一层特别关键，如果第一层梯度值为0，则网络根本不具备调整的方向，就没有办法调整权重值进而达到最小化损失函数的目的。

虽然深层前馈神经网络在传播过程中也会出现梯度消失现象，但前馈神经网络产生梯度消失的原因是由于激活函数 Sigmoid 函数的图像两端梯度趋近于 0，致使网络在参数更新时出现梯度消失现象，解决方法可以在神经网络训练时将激活函数更换为 ReLU 函数。

RNN 产生梯度消失这一现象的根本原因则是网络的参数共享。而解决这一现象的根本方法就是优化模型结构。即在本章 4.2 节中所要提到的单元长短记忆 LSTM（Longoing Short-term Memory），LSTM 可以防止梯度消失的主要原因是 LSTM 的曲线是连续的，所以 LSTM 特别适合反向传播和梯度下降中所涉及的偏微积分计算。同时，LSTM 可以调整网络权重，根据训练的梯度来保留或删除信息，以此

来转换和控制其存储数据的流入和流出，最重要的是 LSTM 可以长时间的保留重要的错误信息，使得梯度相对陡峭，训练时间相对较短，这就解决了梯度消失问题，并提高了 LSTM 网络的准确性，下节来详细阐述 LSTM 的网络结构。

4.2 双向映射长短期记忆时序网络

4.2.1 长短期记忆时序网络

1．长短期记忆时序网络基本结构

长短期记忆时序网络 LSTM（Longoing Short-term Memory），长短期记忆时序网络是循环神经网络的一种[48]。但 LSTM 与 RNN 不同的是 LSTM 的记忆单元具有衡量信息的价值的能力。相比于普通 RNN 对信息无选择的记忆性，LSTM 在每个时刻都将无用的信息遗忘，并输出真正有价值的信息。LSTM 的模型单元内结构如下图 4.4 所示。

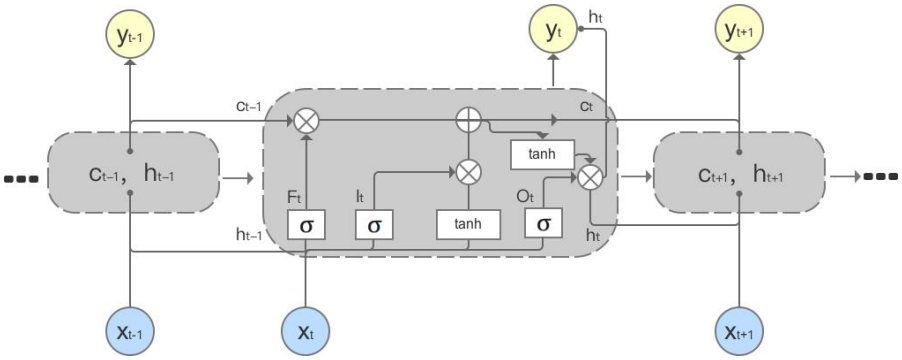

图 4.4 LSTM 单元内结构图

图 4.4 中，LSTM 的网络的状态包含两种，分别是内部状态和外部状态。LSTM 的外部状态同普通的循环神经网络相同，都是当前时刻 t 隐藏层的输出同时作为当前时刻 t 隐藏层的输出和 t+1 时刻隐藏层的输入。而 LSTM 的内部结构则是跟前馈神经网络和普通的 RNN 都有所不同。

2．LSTM 神经元的内部结构

LSTM 的内部状态采用了带有输入门（Input Gate）、遗忘门（Forget Gate）和输出门（Output Gate）的记忆块作为隐含层的神经元，LSTM 之所以能够对信息产生记忆长期依赖的主要原因就是添加了输入门和遗忘门[49]。对于 LSTM 网络首先定义参数下标 c 为隐藏层激活函数输出节点的参数，i 为输入门参数的下标，o 输出门参数的下标，f 遗忘门参数的下标。

（1）输入门

LSTM 中信息前向的传播过程与普通 RNN 传播方式相同，即隐藏层信息通过非线性激活函数 tanh，表达式如式 4-12 所示。

$$\widehat{c_t} = \tanh\left(\mathrm{U}_C h_{t-1} + W_c x_t + b_c\right) \tag{4-12}$$

其中c_t表示激活函数 tanh 节点的输出，U 为隐藏层输入门间的权值矩阵，W 为输入层输入门间的权值矩阵。

LSTM 的输入门决定了 t_i 时刻网络的状态保存到当前 t_i 时刻内部状态的信息量。输入门表达式如式 4-13 所示。

$$i_t = \sigma(U_i h_{t-1} + W_i x_t + b_i) \quad (4\text{-}13)$$

其中，σ为 sigmoid 激活函数，因此输入门根据(0，1)输出值来判定是否输入，如果i_t值趋向于 0 的话，那么c_t就只有极少量的信息会保留在内部状态中，相反的，如果i_t值趋近于 1，那么c_t中被保存的信息就同比增多。最后 LSTM 将c_t与i_t相乘以此来更新内部状态。

（2）遗忘门

在当前 t_i 时刻，遗忘门决定过去 t_{i-1} 时刻状态信息的丢弃量。遗忘门表达式如式 4-14 所示。

$$f_t = \sigma(U_f h_{t-1} + W_f x_t + b_f) \quad (4\text{-}14)$$

其中，U_f上一时刻输出到遗忘门的权值矩阵，W_f是当前时刻输入到遗忘门的权值矩阵。遗忘门的非线性函数依然选用 sigmoid 函数，其原理与输入门的选择有用信息保留的原理相同，当f_t输出值越趋近于 0，被遗忘的信息越多，反之同理。

（3）输出门

输出门决定当前 t_i 时刻的内部状态输出多少信息给神经元的外部状态。输出门表达式如式 4-15 所示。

$$O_t=\sigma（U_o h_{t-1} + W_o x_t + b_o） \quad (4\text{-}15)$$

输出门的非线性函数同样选择 sigmoid 函数。其对信息的筛选原理与输入门和遗忘门相同，当O_t的值越趋近于 1，则当前时刻 t_i 的内部状态 ct 就会有更多的信息输出给 t_i 的外部状态 h_t。

LSTM 相较基于隐马尔可夫模型的最大优点在于，HMM 对于上下文相关信息的建模能力是有限的，而 LSTM 可以有选择性的学习长时依赖的语言。LSTM 相较普通 RNN 的优点在于它能解决 RNN 在训练网络时由反向传播引起的梯度消失问题，因此能学习更长时序上的依赖关系。

长短时记忆单元神经网络虽然通过优化神经元内部结构的方式，有效的解决了普通循环神经网络在训练时出现的梯度消失问题，但传统的 LSTM 仍然存在一个很大的缺陷：因为在训练时隐含层中每前一时刻的输出值都会传递给下一时刻同层的神经节点，因此 LSTM 的内部记忆单元同层神经节点之间传递的向量维度较高，信息冗余较多，计算复杂度较高。如此导致模型在训练时速率低下，训练和识别的效率较低。下节将针对以上问题，对 LSTM 进行优化改进。

4.2.2 映射长短期记忆时序网络

为提高长短时记忆单元神经网络的训练效率，本文提出对 LSTM 的内部单元进行优化改进。优化方法为在 LSTM 的内部记忆块输出前添加两个包含较少神经节点的低维映射层p^1和p^2。其中映射层p^1将输出向量降维后传递给下一时刻的同层神经节点；映射层p^2与p^1将信息拼接完整输出当前时刻记忆单元的状态并传递至当前时刻 t 的输出节点y_t。本文将改进的长短期记忆时序网络命名为映射长短期记忆时序网络 PLSTM（Prejected Long Short-Term Memory Network）。映射长短期记忆时序网络的网络结构如下图 4.5 所示。

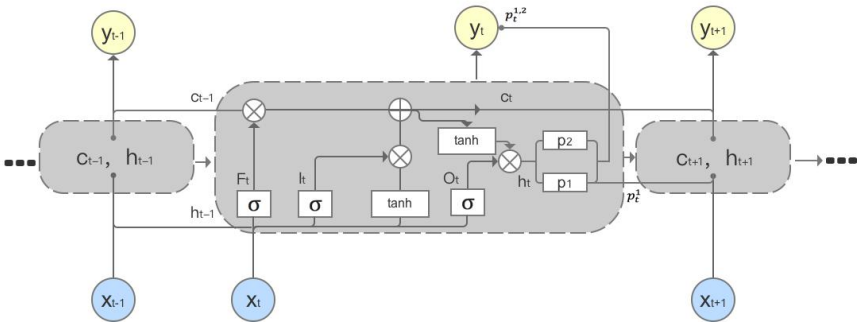

图 4.5 映射长短期记忆时序网络结构图

定义 PLSTM 的参数设置与 LSTM 相同，PLSTM 内部状态中每一非线性变换的作用都与 LSTM 相同，本节就不再赘述。PLSTM 与 LSTM 不同的是每一时刻来自上一时刻隐层神经元的信息都被进行了降维处理。则当前 t 时刻 PLSTM 信息传输的表达式如下：

隐藏层信息通过非线性激活函数 tanh 的单元状态如式 4-16 所示。

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c x_t + U_c p_{t-1}^1 + b_c) \tag{4-16}$$

输入门表达式如式 4-17 所示。

$$i_t = \sigma\left(x_t + U_i p_{t-1}^1 + b_i\right) \tag{4-17}$$

遗忘门表达式如式 4-18 所示。

$$\mathrm{f}_t = \sigma(W_f x_t + U_f p_{t-1}^1 + b_f) \tag{4-18}$$

输出门表达式如式 4-19 所示。

$$o_t = \sigma(W_o x_t + U_o p_{t-1}^1 + b_o) \tag{4-19}$$

隐藏层输出表达式如 4-20 所示。

$$h_t = o_t \odot tanh(c_t) \tag{4-20}$$

p^1层的将隐藏层输出做了低维分割，表达式如式 4-21 所示。

$$p_t^1 = \sigma(W_{p_1} h_t) \tag{4-21}$$

p^2层原理与的p^1层相同，表达式如式 4-22 所示。

$$p_t^2 = \sigma(W_{p_2} h_t) \tag{4-22}$$

t 时刻网络的输出表达式如如式 4-23 所示。

$$y_t = \phi(W_y(p_t^1, p_t^2) + b_y) \tag{4-23}$$

PLSTM 的网络结构具有 LSTM 的所有特性，即可以解决 BP 神经网络对信息没有记忆性的缺陷问题又能解决 RNN 的梯度消失问题。PLSTM 将内部单元优化后又解决了隐层神经元信息传递冗余的问题。但单向的神经网络结构对于时间状态间的影响，却只能停留于当前时刻t_i捕获之前时刻$t_1, t_2, \ldots, t_{i-1}$传入的信息。如果输出序列的预测依赖于$t_i$时刻之后的状态或者整个序列，那么很显然单向 PLSTM 只能学习t_i时刻之前的信息，在性能上并不能实现双向信息依赖，针对以上问题，本文设计采用双向映射长短期记忆时序网络，来处理本文的语音识别任务。下节来阐述可以双向提取序列特征的双向映射长短期记忆时序网络 Bi- PLSTM（Bidirectional Prejected Longoing Short-term Memory）。

4.2.3 双向映射长短期记忆时序网络

双向映射长短期记忆时序网络Bi-PLSTM（Bidirectional Prejected Longoing Short-term Memory），Bi-PLSTM同单向的PLSTM一样，包含内外两种状态结构，单元内结构与单向传播时相同，而外部结构的信息是从时间序列的起点和终点双向传播的网络状态结构[50]。网络的输出单元$O^{(t_i)}$能够同时依赖于$t_1, t_2, \ldots, t_{i-1}$和$t_{i+1}, t_{i+2}, \ldots, t_n$进行计算。Bi-PLSTM模型的外部结构如图4.6所示。

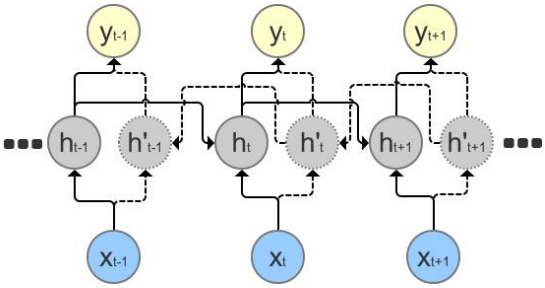

图 4.6 Bi-PLSTM 模型的外部结构图

在进行语音识别实验时，CTC 实现了状态间的“软对齐”和解码识别，双向网络结构则可以从正反两个方向同时提取语音特征，在保证了语音数据的长时依赖性的同时，解决了特征在不同方向的权值共享问题。所以本文采用 Bi-PLSTM 替代的单向网络结构（如单向 RNN、LSTM 等）与链接时序分类算法相结合的方法开展实验。

4.3 链接时序分类算法

链接时序分类算法 CTC（Connectionist temporal classification）是一种对神经网络损失函数的优化方法[51]。CTC 的优点是直接把一串输入的语音 label 直接映射为另一串文字 label。基于 CTC 的建模方法本质上讲就是序列分类问题，网络每个节点的输出都选择一条概率最大生成路径的过程，所以利用 CTC 结合神经网络进行语音识别建模的输入输出关系往往是多对一的关系，如下图 4.7 所示。

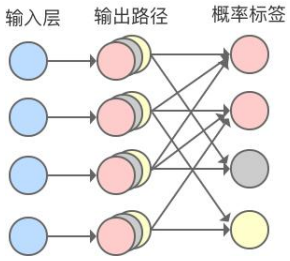

图 4.7 CTC 语音识别输入输出关系图

图 4.7 中，CTC 训练神经网络的的目标是通过输入序列 X 直接得出输出序列 Y，在神经网络的训练过程中会得出输出序列的概率分布 $p(l|x)$），直观上来讲只需选择概率最大的输出序列作为识别结果即可，表达式如式 4-24 所示。

$$O(x) = argmax\, p(l|x) \tag{4-24}$$

如第三章所说，在传统的 GMM-HMM 语音识别框架中优化是逐帧进行的，但对于语音这种针对序列训练的数据，其实每帧甚至每一个模块的优化结果都不是模型训练的最终目标。而 CTC 训练最大的优点就是针对序列数据的训练，网络优化的是 label 的损失，对于单点的损失并不会进行过多计算。

同时 CTC 在神经网络输出的标签序列中插入了 Blank 标签，Blank 的作用主要为以下两点。

1．Blank 会在模型连续多帧输出同一音素时将其合并，这使得声学模型摆脱了逐帧对齐的硬性标准。

2．Blank 可插入在语音端点和两个连续相同的音素中间（即语音卡顿），用以消歧。

例如，识别一段语音序列 ABC，那么输出层的生成路径可以包含多种输出序列，如下图 4.8 所示。

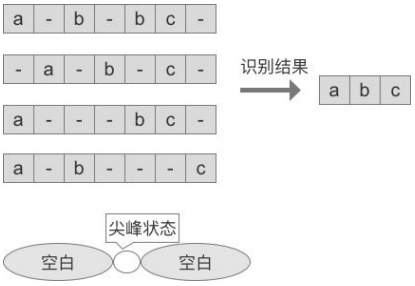

图 4.8 多种生成路径图

图 4.8 中，虽然每一个节点对每个标签的最大后验概率都不同，但是由于 CTC 在解码时会将 Blank 标签去除，将相同音素合并。所以序列任何一种可能的展开，都是这个序列的实例。如此在 CTC 解码时，每一帧对齐方式的重要性就已不再重要。

CTC 对神经网络的优化主要体现在两个问题上，分别是训练和解码。

4.4 基于 Bi-PLSTM 的 CTC 训练过程

1．CTC 相关符号定义

给定一个 Bi-PLSTM 进行 CTC 推导训练，首先定义 L 为所有输出标签集合，CTC 扩展至 $A' = A \cup \{blank\}$。在给定历史条件下 t 时刻输出标签 k 的概率表示如式 4-25 所示。

$$y_k^t = p(O_t = k|x_0,x_1,...x_t) \tag{4-25}$$

假定在给定输入序列 x 条件下，t 时刻的输出概率独立（此处假设上下文相关信息已由 PLSTM 的外部结构处理）定义A'^T为由 A'构成的长度为 T 的输出序列集合，则一条路径$\pi \in A'^T$的条件概率表达式如式 4-26 所示。

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t \tag{4-26}$$

定义从路径π到 label 序列 l 的映射关系 F:$A'^T \to A^{\leq T}$，该映射关系将π中连续和相同的标签只保留一个，然后去除 Blank 标签。则标签序列 $l \in A^{\leq T}$的概率需要将所有属于 l 的路径概率进行累加，表达式如式 4-27 所示。

$$p(l|x) = \sum_{\pi \epsilon F^{-1}(l)} p(\pi|x) \tag{4-27}$$

$p(l|x)$可以利用 HMM 中的前后向算法进行求解，由于网络已经假设在给定输入序列 x 条件下，t 时刻的输出概率独立，所以不需要考虑状态间的转移概率。前后向算法计算如下：

定义α（t，u）为第 t 时刻输出扩展标签 u 的概率，则 label 序列的后验概率如式 4-28 所示。

$$p(l|x) = \alpha(\mathrm{T},\mathrm{U}') + \alpha(\mathrm{T},\mathrm{U}' - 1) \tag{4-28}$$

前向概率计算如下，首先进行参数初始化如式 4-29 所示。

$$\alpha(1，1) = y_b^1；\ \alpha（1，2）= y_{l_1}^1；\ \alpha(1，\mathrm{u}) = 0\ (\forall \mathrm{u} > 2) \tag{4-29}$$

然后进行递推计算如式 4-30、式 4-31 所示。

$$\alpha(\mathrm{t}，\mathrm{u}) = y_{l'_u}^t \sum_{i=f(u)}^{u} \alpha(t-1,i) \tag{4-30}$$

$$\text{其中 } f(u)=\begin{cases} u-1 & if\ \ l'_u = blank\ \ or\ \ l'_{u-2} = l' \\ u-2 & otherwise \end{cases} \tag{4-31}$$

后向算法的过程与前向算法类似，只是初始化从最后一个时刻的输出概率进行，然后递推回第一个时刻。首先定义后向概率β(t,u)。

首先进行参数初始化，如式 4-32 所示。

$$\beta(T,U') = \beta\left(T,U'-1\right) = 1,\beta(T,u) = 0,\forall u < U'-1 \tag{4-32}$$

然后将每一时刻的输出向前递推，如式 4-33、4-34 所示。

$$\beta(t,u) = \sum_{i=u}^{g(u')} \beta(t+1,i) y_{l'_u}^{t+1} \tag{4-33}$$

$$\text{其中 } g(u)=\begin{cases} u+1 & if\ l'_u = blank\ or\ l'_{u+2} = l'_u \\ u+2 & otherwise \end{cases} \tag{4-34}$$

通过计算序列的前向概率和后向概率可知，t 时刻通过标签 u 的概率如式 4-35 所示。

$$\sum_{\pi \epsilon X(t,u)} p(\pi|x) = \alpha(t,u)\beta(t,u) \tag{4-35}$$

如图 4.9 表示前后向算法的推导过程。

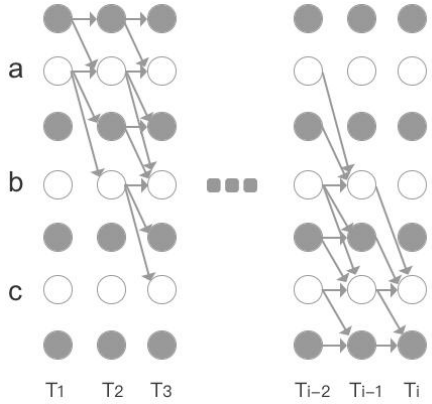

图 4.9 前后向算法推导图

2．CTC 目标优化及参数更新

CTC的损失函数定义为训练集S上label序列的负对数概率，则每个样本输出的损失函数L(l,z)如式4-36所示。

$$L(l,z) = -\ln p(l|x) \tag{4-36}$$

由于每条输出序列间假设条件独立，训练集S上的总输出为$\prod_{(x,l)} p(l|x)$，对CTC的损失函数进行对数的数学变换后得到整个训练集的损失函数L_S表达式如式4-37所示。

$$L_S = -\sum_{(X,L)\in S} \ln p(l|x) \tag{4-37}$$

CTC 损失函数的优化同样采用链式求导法则进行梯度下降的方法。损失函数 L 对到网络输出层参数y_k^t进行求导，表达式如式 4-38 所示。

$$\frac{\partial L}{\partial y_k^t} = -\frac{\partial \ln p(l|x)}{\partial y_k^t} = -\frac{1}{p(l|x)}\frac{\partial p(l|x)}{\partial y_k^t} \tag{4-38}$$

然后将序列不同位置可能出现相同的标签去重，定义集合B(l,k) = {u:l'_u = k}，B表示标签序列z中标签为k的所有位置集合。代入前后向概率，则对y_k^t的梯度优化如式4-39所示。

$$\frac{\partial\, \alpha\beta}{\partial y_k^t} = \begin{cases} \dfrac{\alpha(\mathrm{t,u})\beta(t,u)}{y_k^t} & if\ k\ occurs\ in\ z \\ 0 & otherwise \end{cases} \tag{4-39}$$

按照如上优化过程，使用BPTT逐层逐帧更新网络的模型参数即可使模型收敛。

4.5 基于 Bi-PLSTM 的 CTC 解码原理

4.5.1 不结合语言模型解码

对于不结合语言模型的解码过程，本文采用 Best path Decoding 算法，Best path Decoding 算法是取每一帧的最大概率标签生成的序列，并假设概率最大路径的最优序列为全局最优序列，Best path Decoding 解码算法将最优表达式如下。

首先计算网络最优输出序列，表达式如式 4-40 所示。

$$l = argmax_l\ p(l|x) = argmax_l \sum_{\pi:F(\pi)=l} p(\pi|x) \tag{4-40}$$

然后用后验概率最大的路径序列π对应的 label 序列，近似为后验概率最大的全局最优序列，定义 $l \approx \mathrm{F}(\pi^*)$，则 l 的表达式如式 4-41 所示。

$$l \approx \hat{l} = F(argmax_\pi p(\pi|x)) \tag{4-41}$$

这一近似使用得分最大的路径概率来近似label序列所有可能路径的累加，这种不带语言模型的解码在理论上可以得到最优解，但在实际任务中，尤其是中文语音识别任务中，CTC解码模型往往由于中文的发音与拼写特点导致解码效果差。所以本文采用端到端语音识别模型外接语言模型的解码方式。以下重点介绍CTC模型在引入语言模型条件下的Viterbi解码过程。

4.5.2 结合语言模型解码

1．带语言模型解码

定义在结合语言模型解码时，解码给模型增加的限制信息记作 G，在加权有限转换器格式 WFST（Weighted Finite State Transducer）框架下，当给定节点跳转到新 label 的概率 $p = p(l_\mathrm{i}|S)$时，解码目标函数 l 表达式如式 4-42 所示。

$$l = \mathrm{argmax}_l\, p(l|x,G) \tag{4-42}$$

由条件概率公式可得式 4-43。

$$p(l|x,G)=\frac{p(l|x)p(l|G)p(x)}{p(\mathrm{x}|\mathrm{G})p(l)} \tag{4-43}$$

p 对于相同输入序列的所有竞争路径，$p(x)$为常量可忽略。另外假设输入序列 X 和限制信息 G 相互独立，即认为输入序列 X 与所结合的语言模型无关，则上式可直接忽略$p(x|G)$，基于以上假设重写上式为式 4-44。

$$p(l|x,\mathrm{G})=\frac{p(l|x)p(l|G)}{p(l)} \tag{4-44}$$

根据上式将目标函数进行优化，表达式如式4-45。

$$l=\mathrm{argmax}_l\, p(l|x,G)=argmax_l\frac{p(l|x)p(l|G)}{p(l)} \tag{4-45}$$

依据4.5.1小节不结合语言模型解码原理的阐述过程，将后验概率最大的路径序列π对应的label序列，近似为后验概率最大的全局最优序列，表达式如式4-46所示。

$$l\approx\hat{l}=F(argmax_\pi\frac{p(\pi|x)p(F(\pi)|G)}{p(\pi)}) \tag{4-46}$$

以上阐述了CTC训练结合了语言模型的解码过程。

2．CTC 的区分性训练

结合语言模型解码的过程中，解码时优化目标为 $l=\mathrm{argmax}_l\, p(l|x,G)$，而实际 CTC 的训练目标为不结合语言模型解码的优化目标 $l=argmax_l\ p(l|x)$，如此就出现训练目标与解码目标不匹配的问题。故本文对 CTC 模型结合语言模型解码的过程，引入了 Hybrid 结构中的区分性训练来解决此问题。

区分性训练的目的是使得已有模型在解码条件下，正确的标签路径序列的得分相对其他竞争序列占优。本文采用最大化互信息MMI（Maximum Mutual Information）的训练准则来进行CTC模型的区分式训练。

首先训练优化的目标是正确序列的后验概率，表达式如式 4-47 所示。

$$\mathrm{F}_{MMI}=\sum_r logp(s_r|x_r) \tag{4-47}$$

其中，s表示可能的识别结果序列，即竞争序列，s_r为数据r的标注序列。

但是上式所表示的优化目标与识别词错率没有直接关联，故本文利用基于最小化贝叶斯风险框架的优化准则来优化改进目标函数，优化后表达式如式4-48所示。

$$F_{MBR}=\sum_r\sum_s p(s\,|\,x_r)A(s,s_r) \tag{4-48}$$

其中，$A(s,s_r)$为识别的结果序列s相对目标序列s_r的正确率度量，本文采用MBR优化准则，即优化目标为正确率度量$A(s,s_r)$的期望，则重写上式为式4-49。

$$F_{MMI}=\sum_u log\frac{\sum_{s_r}p(s_r|x,G)^m}{\sum_s p(s|x,G)^m} \tag{4-49}$$

其中，m为后验概率规整因子。定义Bi-PLSTM网络的输出层输出为y_k^{ut}，其中，t表示序列r的第t帧特征，k表示神经网络的第k个节点的输出。则开始迭代网络权值，目标函数对输出层的求导公式表示如式4-50所示。

$$\frac{\partial F_{MMI}}{\partial y_k^{ut}} = \frac{\partial log \frac{\sum_{s_r} p(s_r|x,G)^m}{\sum_s p(s|x,G)^m}}{\partial y_k^{ut}} \tag{4-50}$$

按照如上优化过程，逐层逐帧更新网络的模型参数即可使模型收敛。由于基于CTC准则的一个建模单元有限状态机的词图是一条弧线，所以不需要像基于HMM模型的三音素状态那样对弧内部使用前后向算法，基于CTC的模型只要将词图弧上的得分累加即可。另外CTC模型输出标签的得分分布呈“尖峰”状，并且尖峰位置不稳定，所以在训练时应及时应用迭代出的最新模型对词图的得分进行重估。

4.6 本章小结

本章首先阐释了传统 RNN，LSTM 的网络结构特点与算法推导过程，然后针对 LSTM 的缺陷，提出了一种命名为 PLSTM 的改进 LSTM。同时，设计了利用 Bi-PLSTM 代替传统 RNN 和 LSTM 与 CTC 算法相结合的方法进行实验。接着阐述了 CTC 结合神经网络在语音识别中的建模原理。最后详细介绍了 CTC 的训练和结合语言模型的解码过程。

第五章 实验设计及结果分析

5.1 实验数据集介绍

1．实验数据

本实验所用数据是由北京希尔贝壳科技有限公司开源的 AISHELL 语音数据库。该数据库中，语音包含高保真麦克风、Android 手机、iOS 手机三种不同设备的录音数据，共计 178 小时。数据内容涉及工业生产、智能家居、无人驾驶等领域。

音频数据的基本信息如下表 5-1 所示。

表 5-1 音频基本信息表

采样率	采样精度	每条数据时长	是否包含口音	说话人数
16000HZ	16bit	4s	是	400+

本实验中，语音数据集划分为训练集、验证集和测试集三部分，划分的语音条数和时长统计的结果如下表 5-2 所示。

表 5-2 实验数据及其划分表

集合	语音条（条）	时长 （小时）
训练集	120098	150.85
验证集	14326	18.09
测试集	7176	10.03
总计	141600	178.97

5.2 基于自适应技术的 GMM-HMM 算法模型实验结果及分析

1．系统环境及实验工具

（1）基于 GMM-HMM 模型的语音识别实验机器配置如下表 5-3 所示。

表 5-3 实验机器配置表

操作系统	CPU 处理器	内存	GPU
Mac OS	8 核	16G	--

（2）实验工具

本实验的实验工具为开源的语音识别工具 Kaldi，由约翰霍普金斯大学 Daniel Povey 教授开发，目前已成为语音识别领域使用最为广泛的工具之一。Kaldi 在最初的开发阶段，仅使用 C++代码实现了语音识别的经典算法模型 GMM-HMM，提供数据处理、特征提取、声学、语言学模型训练以及解码等功能。

随着近几年语音识别技术不断的发展完善，Kaldi 现今除支持传统的 GMM-HMM 模型训练外，还支持多种类型的神经网络模型训练，并在性能和效率上都取得了不断的进

步。另外，Kaldi 工具同时可以应用于进行说话人识别、语种识别、语音关键词检索等领域[52-53]。

2．实验设计

（1）特征提取

特征是连接多样化数据与机器学习的桥梁，所以特征提取是机器学习模型训练前的必要步骤。提取有效的语音特征对于模型的识别准确率有着非常重要的影响。本实验采用语音识别领域应用广泛的梅尔倒谱系数 MFCC 作为传统模型语音识别模型的声学特征，由于在第二章中已经详细介绍过 MFCC，本节就不再赘述。

提取时特征基本参数如下表 5-4 所示。

表 5-4 特征提取基本参数表

帧长	帧移	梅尔滤波器
25ms	10ms	39 组

如下图 5.1 所示，为本文部分实验数据提取的 MFCC 特征。

```
58.44347 10.69868 25.14065 12.9943 -6.539965 12.20891 13.52931 4.445545 -26.27521 25.67029 -17.82642 -0.5804935 0.1862898 -0.7773106 -0.1881143 -0.001365513
55.77683 13.49476 28.05092 16.64765 1.883205 5.436808 6.039011 -5.468349 -16.98991 8.661319 -16.56565 -6.093733 0.6730661 -0.795765 -0.1808114 0.006693959
54.29537 14.61319 28.46668 14.61801 0.6286907 -2.72744 7.376565 -4.854383 -23.5196 1.071599 -10.70267 -5.169558 1.938683 -0.6973411 -0.1844628 -0.06929535
63.48044 9.859857 1.852678 -10.42035 23.13544 4.97028 6.306522 -22.34554 -10.34233 7.902347 6.581785 -4.245382 -9.664541 -0.5175321 -0.2063715 -0.02554393
68.36927 -6.15163 6.635927 -2.206221 20.5499 0.7715244 -8.341982 -34.76506 -9.178999 12.34775 11.6677 -4.70747 -10.47931 -0.4339105 -0.2282801 -0.1256785
64.22118 -3.19558 0.3969069 2.381348 19.25712 1.937845 -18.18467 -37.98494 -10.00995 18.08969 13.62382 -3.783295 -7.699478 -0.1250977 -0.2647945 -0.0002141595
62.73971 12.09672 1.852678 0.284173 7.191232 -0.6280613 -22.08073 -42.58476 -22.83069 12.88988 9.711578 2.355407 -0.4951963 -0.1653664 -0.2465373 -0.02669528
63.33229 16.57045 3.30845 -1.157634 -5.464666 1.471315 -12.85321 -45.34465 -18.98419 12.13091 5.799337 3.156108 3.886828 -0.3344948 -0.2465373 0.05044538
63.18415 22.44222 3.30845 -4.041248 -4.927017 0.07173157 -7.93187 -39.82487 -24.8974 21.88 22.23075 -0.5804935 -6.04444 -0.5583231 -0.2282801 0.06886703
```

图 5.1 本文部分数据 MFCC 截图

（2）声学模型的训练

声学模型是用于表示语音的声学特征与音素之间的映射关系的模型。提取语音的声学特征后，要对每个音素建立并训练声学模型。

语言学研究表明，每个音素（对应中文的声母和韵母）的发音不光受这个音素本身影响，还会受其相邻的前后音素影响。单独的音素称之为单音素（例如 b，p，i，都是单音素），将考虑其直接相邻的前后各一个音素这样上下文的音素称之为三音素。（比如 b_i+p，f_ai+m 为三音素）

为了提高基于 GMM-HMM 模型基线试验的准确率，以验证本文基于神经网络时序分类方法的有效性。本文先后训练了单音素模型和三音素模型，并运用了说话人自适应技术来优化模型的识别准确率。最后，将每个模型在测试集上分别做识别并计算词错率进行比较。

1）单音素声学模型训练

单音素声学模型训练流程图如下图 5.2 所示。单音素声学模型训练具体流程如下：

a．首先生成整个模型的拓扑结构（topo），指定每个音素（建模单元）的 HMM 状态数及其 id 等；生成训练图（即整个句子对应的 HMM 串）；

b．根据 topo 来生成初始模型 0.mdl 和决策树 tree（单音素训练不需要决策树算法，但在三音素训练时决策树用以共享 HMM 状态。根据三音素的判别条件，借助决策树获取该三音素的聚类状态）；

c．将训练语音帧往 HMM 状态上做平均化对齐，统计对齐后的统计量；

d．根据统计结果，在 0.mdl 基础上更新模型参数，得到迭代后的模型 1.mdl；

e．利用一轮迭代后的模型 1.mdl 对训练语音做对齐；

f．统计对齐后的统计量；

g．根据统计结果，更新模型参数，继续迭代模型；

h．判断是否已经到达最大迭代步骤，如果到达则训练结束，否则返回 e 继续迭代。

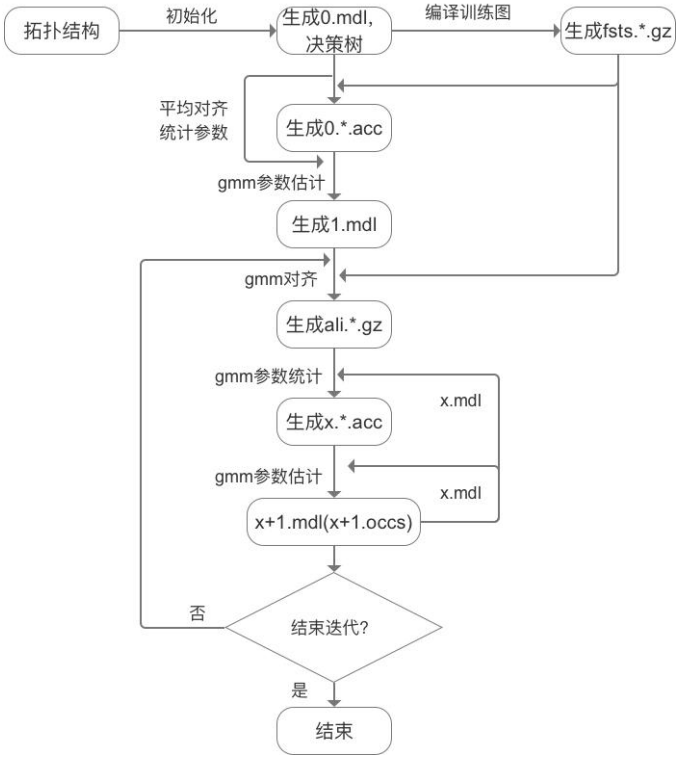

图 5.2 Kaldi 单音素声学模型训练流程图

2）三音素声学模型训练

三音素模型的训练流程和单音素的训练流程基本类似，区别在于三音素声学模型添加了动态特征（包含一阶差分和二阶差分）、LDA 降维、自适应等，其训练流程此处不再赘述。

（3）语言学模型训练

本实验中的语言模型借助斯坦福大学的语言模型训练工具 srilm 来统计得到 N-gram 语言模型。语言模型的训练样本为本文语音数据 aishell 中的对应文本，共计 120098 个文本语句。

1）语言模型训练流程及结果

统计过程中，词典中没有出现过的词用“<SPOKEN_NOISE>”表示。

本实验将 N-gram 阶数设置为 3，即统计 trigram 语言模型。统计结果显示，unigram 数目 137076，bigram 数目 438252 和 trigram 数目 100860。N-gram 语言模型统计结果举例如下表 5-5 所示。

表 5-5 N-gram 语言模型统计结果举例表

N-gram 语言模型统计结果举例（数字表示对数概率）						
Uni-gram 结果举例			Bi-gram 结果举例		Tri-gram 结果举例	
李	电饭煲	电扇	开发 商人	开发 项	楼市 持续 呈现	今天 发布 通知
-3.243	-6.480	-6.189	-2.690	-1.992	-0.3294	-2.202

语言模型的可视化为加权有限转换器格式 WFST，如下图 5.3 所示。

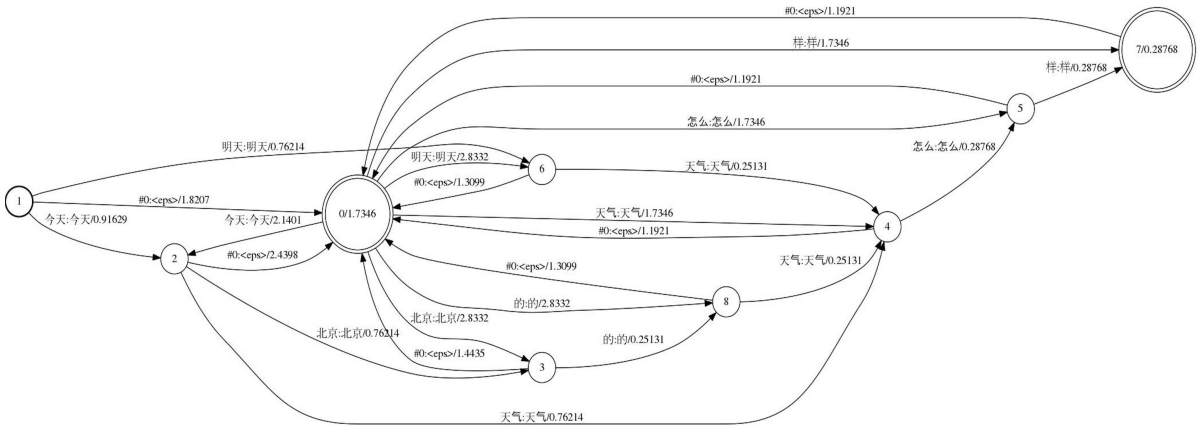

图 5.3 语言模型的 WFST 表示示例图

图 5.3 中，同心圆表示终止状态，状态 1 是初始状态。两个状态之间的有向边上的信息格式是“输入：输出/权重”，分别表示从一个状态转换到另一个状态可以接收的输入内容以及输出内容和权重。

3．实验结果及分析

（1）实验组设计

本文共设计了 6 组 GMM-HMM 模型实验。

1）monophone model：建模单元为单音素，每个单音素有一个 GMM-HMM 模型。模型的参数最初经过随机初始化得到，设置迭代次数为 39 次，得到最终的单音素模型；

2）triphone model1：建模单元为三音素。在语音特征中加入了动态特征，利用上一步训练得到的单音素模型来对训练语音做帧到 HMM 状态的对齐，并训练决策树；

3）triphone model2：建模单元是三音素。基本流程与步骤 2）类似，但这次的训练是基于上一步的三音素模型，而不是基于单音素模型；

4）triphone + LDA：LDA 目的是对特征进行降维，将原来的高维特征映射为更利于聚类的低维特征；

5）triphone + SAT1：训练数据中有多个说话人，每个说话人的发音方式存在差异。利用说话人自适应技术，减小因说话人发音方式不同而带来的模型差异；

6）triphone + SAT2：基本流程与步骤 5）相同，但此次实验增加了 GMM 模型的高斯参数，使得本次实验的模型规模更大，参数量更多。

（2）语音识别的衡量指标

语音识别的衡量指标为词错率（Word Error Rate，WER），WER 表达式如式 5-1 所示。

$$\mathrm{WER} = 100\% * \frac{Subs + Dels + Ins}{word\ in\ correct\ sentence} \tag{5-1}$$

其中，Subs（Subsitutions）表示错词需进行替换，Dels（Deletions）表示漏词需进行插入，Ins（Insertions）表示多词需进行删除。

中文的语音识别的衡量指标为字错率 CER（Character Error Rate）。其计算方式与 WER 相同，都是通过计算识别的文本结果和正确的标注之间的插入、删除、替代三种错误的总体比例得到的。

（3）实验结果

基于 GMM-HMM 模型的实验结果如下表 5-6 所示。

表 5-6 GMM-HMM 模型实验结果表

models	CER
monophone model	35.45%
triphone model1	19.98%
triphone model2	19.86%
triphone + LDA	19.50%
triphone + SAT1	13.83%
triphone + SAT2	**12.12%**

如上表格显示：

1）单音素模型的识别准确率较低，CER 为 35.45%，并不足以为本文实验方法作基线理论支撑。

2）对于连续的语音，音素的发音受到其前后音素（上下文）的影响。训练三音素模型实验结果为 19.86%，模型准确率明显优于单音素模型。

3）将高维特征进行降维处理，剔除高维特征中的冗余信息，减小冗余信息对识别的干扰，此实验 CER 降为 19.50%。

4）使用说话人自适应技术，削弱不同说话人的口音差异。提升实验结果，采用自适应技术后 CER 降为 12.12%。

5.3 基于 Bi-PLSTM 的链接时序分类算法模型实验结果及分析

1．系统环境及实验工具

本节基于 CTC 的模型训练同样采用 AISHELL 数据集进行。由于 CTC 结合双向映射长短期时序记忆网络训练会出现大量矩阵的操作，比如矩阵相乘、相加等，计算量非常大，普通 CPU 的计算效率远远达不到神经网络的训练要求。所以此实验本文选择在计算性能更好的 GPU 环境中进行。

基于 CTC 模型的语音识别实验机器配置如下表 5-7 所示。

表 5-7 CTC 模型实验机器配置表

操作系统	CPU 处理器	内存	GPU
Ubuntu	16 核	64G RAM	Titan X(pascal)

（1）EESEN 语音识别工具

EESEN 是一款用 C++实现了 LSTM-CTC 核心算法的工具，由卡内基梅隆大学的苗亚杰博士开发[54]。该工具基于 Kaldi，主要借助 Kaldi 完成数据准备（如语音文件的索引文件、句子及其说话人的映射文件、标注文本等）、特征提取、词典和语言模型构建以及任务分配等。

相比于传统的语音识别工具，EESEN 抛弃了以下几个模块：

1）隐马尔科夫模型和混合高斯分布模型。

因为 EESEN 用于端到端语音识别技术，所以不需要基于隐马尔科夫模型-混合高斯分布模型来为神经网络的训练提前生成语音帧到隐马尔科夫模型状态的映射（即每一帧的对齐）。

2）决策树和用于音素分类的问题集。

因为三音素存在状态共享（绑定）的现象，共享通过应用决策树算法对状态所对应的输出模型进行聚类。在决策树的每个非叶子节点都对应到一个问题，比如“这个音素是不是鼻音？”、“这个音素的前一个音素是不是元音？”等，这些问题的答案都为“是”或“非”，经过对这些问题的回答，一个三音素最终得到它对应的隐马尔科夫模型的状态，决策树模型的质量也会影响到最终语音识别的准确性。

同时 EESEN 实现了单机多卡的训练方式，因此为了提高训练效率，本实验选择用 3 块 GUP 卡并行来加速训练。

2．实验设计

（1）特征提取

在基于 GMM-HMM 的模型训练中，本文所使用声学特征为 MFCC，而在基于 CTC 的模型训练中，本文选择特征为 Fbanks。原因为在 GMM 的训练中，高斯分布的协方差矩阵使用的是对角协方差矩阵，这要求特征各维之间是独立的，而 MFCC 在计算过程中

做了离散余弦变换以解除特征各维的相关性。所以在基于 GMM-HMM 的实验中选择 MFCC 作为语音特征。

对于神经网络而言，没有要求输入的特征各维之间需要不相关。Fbanks 特征和 MFCC 特征类似，但提取过程只经过分帧、傅里叶变换、梅尔滤波以及取对数几个操作。且据实验表明，在神经网络中使用 Fbanks 特征模型性能优于 MFCC 特征。因此，本实验选择 Fbanks 作为神经网络的输入。提取时特征基本参数如下表 5-8 所示。

表 5-8 特征提取基本参数表

帧长	帧移	梅尔滤波器
25ms	10ms	40 组

（2）基于 CTC 的模型训练

基于 CTC 实验的模型结构如图 5.4 所示。

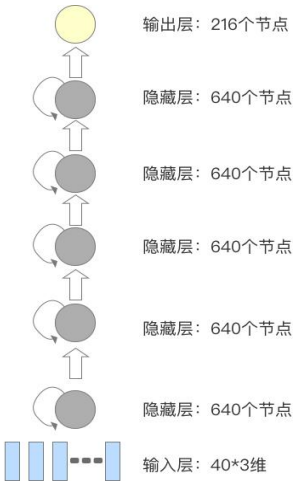

图 5.4 声学模型网络结构图

通过图 5.4 可以看出，本实验模型的网络结构：

输入层：网络的输入是 40 维的 Fbanks 特征及其一阶、二阶差分，共 120 维；

隐藏层：5 个隐含层采用 Bi-PLSTM 结构，前向后向分别包含 320 个 PLSTM 神经元，隐含层采用 tanh 激活函数；

输出层：输出层采用 softmax 激活函数对最后一个隐含层的输出做仿射变换，输出层 216 个节点，每个节点的输出表示模型对该帧的输入在各个音素（包含<blk>）上的概率。

本实验基本参数设置如下表 5-9 所示。

表 5-9 实验基本参数表

初始学习率	迭代轮数
0.00004	25

表 5-9 中，初始学习率设置为 0.00004，在训练过程中统计训练集和验证集的识别准确率，当验证集上的准确率上升低于 0.5 则学习率降半。为防止过拟合，本实验在训练过程中采用早停策略（Early Stopping），当模型在验证集上的准确率提升小于 0.1 时，就终止训练。训练的 batch size 设为 10，即 10 条语音并行处理。

（3）语言模型训练

本次实验和基于 HMM 的实验使用相同词典，实验将词典中的音素（韵母，并且带声调）抽取出来，并加上<blk>音素和一些消歧符，构成神经网络的输出集合，将其编译成 WFST 格式，得到 T.fst。T.fst 的部分内容可视化结果如图 5.5 所示。

在图 5.5 的 T.fst 中，每个音素都有对应的路径，而且在每条非自循环的路径上，输入和输出都是同一个音素。语言模型的生成和基于 HMM 的实验是相同的，这里不再赘述。

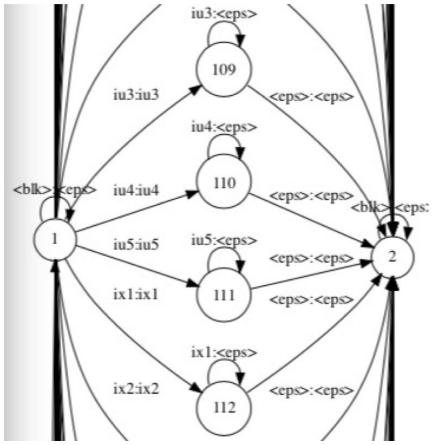

图 5.5 T.fst 可视化图

3．实验结果及分析

（1）实验设计

因为本文采用的实验方法基于 Bi-PLSTM 结合 CTC 的模型进行训练，输入的数据特征越少越易产生过拟合的现象，数据越多神经网络模型的效果发挥的越好。故本文在原训练数据集 150 小时语音数据的基础上，设计结合了速度扰动技术。将实验数据的语音扩充至原语音的 1.1 倍速和 0.9 倍速语音数据。如此本实验在并未新增语音数据的前提下，扩充 3 倍的训练数据量，使此模型在实际应用中，如果出现缺少数据的情况下同样可以将模型充分训练。

（2）实验结果

本文分别在原数据集（150 小时）和结合扰动的数据集（450 小时）上进行训练。

1）模型准确率对比

训练过程中模型在训练集和验证集上的音素识别准确率如图 5.6 所示。

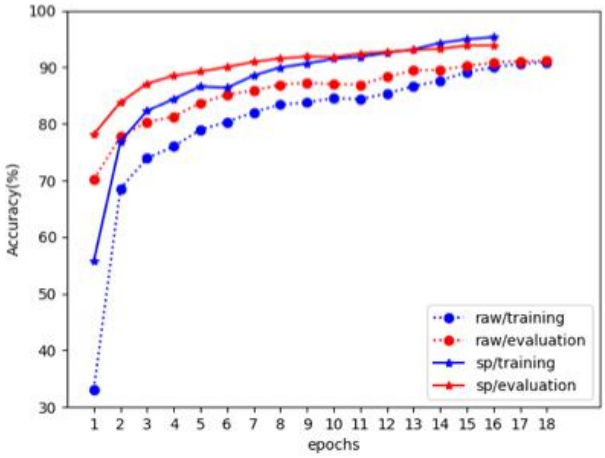

图 5.6 音素识别准确率对比图

图 5.6 中，raw/training 和 raw/evaluation 分别表示用原 150 小时数据集训练模型过程中在训练集和验证集上的识别准确率；sp/training 和 sp/evaluation 分别表示添加速度扰动后训练的模型在训练集和验证集上的识别准确率。

原数据集上的训练（对应虚线圆点的红蓝两条曲线），易得出在初始化的第一轮训练中，训练集上的识别准确率较低。经过这一轮训练后，在训练数据集上的识别准确率迅速提升为超过 60%，在验证集上识别准确率达也达到 70%，训练效果初见成效。在接下来的迭代训练中，识别准确率逐步提升，但是提升率呈现下降趋势，直到第 18 轮迭代，比上一轮的准确率提升小于设置的阈值，训练终止。

在经过速度扰动的训练集上进行训练（对应实线星号的红蓝两条曲线），其训练过程中模型表现也呈类似情况。但模型在训练到第 16 轮时由于在验证集上的准确率上升小于阈值而训练终止。

基于 CTC 的模型和基于 HMM 的模型性能对比如表 5-10 所示。

表 5-10 Bi-PLSTM-CTC 模型和 GMM-HMM 模型的识别性能对比表

模型	训练集	词错率
GMM-HMM	原数据集（150 小时）	12.12%
Bi-LSTM-CTC	原数据集（150 小时）	9.93%
	速度扰动数据（150*3）	8.54%
Bi-PLSTM-CTC	原数据集（150 小时）	**9.88%**
	速度扰动数据（150*3）	**8.52%**

在原数据集训练的 Bi-PLSTM-CTC 模型，已大幅降低了测试集上的词错率。原数据集只有 150 小时，基于 CTC 的模型就已取得较高的准确率，证明了本文方法模型能力的强大。基于 CTC 的端到端模型直接将减小模型对音素的识别误差作为目标，免去 HMM 的决策树等中间模型的训练，也减小这些模型训练带来的误差。在进一步通过结合速度扰动技术来扩充训练数据后，模型词错率再一次降低。因为端到端的训练方法对

训练数据的要求比较高，所以即使做了速度扰动，本实验目前使用的训练数据量还是相对较少，所以本文方法的模型潜力还没有完全被发挥。

2）训练速率实验结果对比

本实验设定迭代次数相同、实验环境相同以及训练数据集（扰动数据集）相同的情况下，分别训练 LSTM 和 PLSTM 模型，统计训练速度如表 5-11 所示：

表 5-11 PLSTM 与 LSTM 训练速度对比表

模型	训练时长（小时）	迭代次数（次）	平均训练速度（小时/次）
LSTM	173 小时	16	10.81
PLSTM	159 小时	16	9.94

表 5-11 中可以看出，在相同数据集上，迭代轮次数同为 16 轮的情况下，LSTM 训练 450 小时的训练数据耗时 173 小时，PLSTM 仅耗时 159 小时。改进后的 PLSTM 较传统 LSTM 缩短相对 8.09%的速率提升。

通过对比分析实验结果，在原数据上本文基于 PLSTM-CTC 的训练方法比基于 GMM-HMM 的训练方法，错词率相对下降 18.50%；在扰动数据集上，PLSTM-CTC 相比于 GMM-HMM 词错率下降 29.70%。同时优化后的 PLSTM 在保证识别准确率的前提下，获得相对 LSTM 网络 8.09%的速率提升，实验结果表明本文在训练速率和识别准确率两个维度都取得了显著的效果。

5.4 本章小结

本章分别介绍了基于自适应技术的 GMM-HMM 算法和基于 Bi-PLSTM 的链接时序分类方法实验的相关内容，包括实验环境、实验工具，以及实验数据和训练的整体流程。最后通过对比实验，验证了本文提出的方法在训练速率和模型准确率上都有明显的提高。

结 论

本文以改进的传统语音识别模型为基线，设计基于神经网络时序分类的算法模型用于语音识别，本文的主要研究工作如下：

1．为验证本文所提方法的有效性，设计了一组基线实验，该实验采用融合说话人自适应技术的隐马高斯模型，用以完成基于隐马尔科夫模型对语音时序信息进行建模、利用高斯混合模型拟合状态转移矩阵的观测概率、基于自适应技术的重估高斯混合模型的参数矩阵等工作。基线实验结果表明，基于单音素的建模方式在经过一轮迭代后模型的词错率为 35.45%；增加上下文音素聚类后，模型词错率降低至 19.50%；将模型融合说话人自适应技术，削弱不同人的口音差异，模型词错率降低至 12.12%。

2.本文提出了利用改进的映射长短时序记忆网络 PLSTM 与链接时序分类算法 CTC 结合的端到端训练方式，用以完成对比分析多种循环神经网络的优缺点。根据对比结果选取长短期序列网络进行序列训练，并针对长短期序列网络的缺陷对其进行模型结构和信息传播两个维度的优化改进，设计并使用双向映射长短时序记忆网络替代单向网络开展训练任务，同时，本文研究了链接时序分类算法对神经网络的优化方式。经实验验证，本文提出的基于 CTC-Bi-PLSTM 的模型在识别准确率和训练速率上都取得了较明显的提升。

3．对本文提出的融合中文语言模型的非完全端到端框架进行原理阐述与实验分析。主要完成框架的原理设计、端到端框架结合语言模型解码的原理过程与算法分析、数据特征提取以及中文语言模型的构建等工作。经实验验证，本文提出的基于中文语言模型的非完全端到端框架在中文语音识别中具有较高的准确率。

4．本文针对采用双向神经网络训练时易产生过拟合的问题，引入速度扰动的方法，通过设计原始数据集扰动速率，在实验中对生成的扰动数据进行特征提取并将数据输入网络，用以辅助模型训练。实验结果表明，在原数据上本文基于 CTC-Bi-PLSTM 的非完全端到端训练方法，错词率下降为 9.88%，相对于基线实验有 18.50%的提升；结合速度扰动技术后，本文方法在相对于 CTC-Bi-LSTM 实验 8.09%速率提升的同时，词错率降至 8.52%；相对于 GMM-HMM 的模型实验有 29.70%的提升，通过实验验证了本文方法的高效性。

然而，在研究过程中发现，利用基于 CTC-Bi-PLSTM 的非完全端到端模型训练时，还存在某些问题有待研究，如：本文方法是在播音级语音数据上进行的实验，并未验证远场语音数据使用本文方法的有效性，为了让基于 Bi-PLSTM 的算法模型得到更充分的训练，可以对数据作进一步扩充，比如加噪音、加混响等，这将留在以后的工作中进行探讨，这也是本人今后研究与学习的重点。

参考文献

[1] A Zeyer, K Irie, R Schluter, and H Ney, “Improved train- ¨ ing of end-to-end attention models for speech recognition,” CoRR, vol. abs/1805.0, 2018.

[2] Gamliel O , Shallom I D . Perceptual Time Varying Linear Prediction model for speech applications[C]// IEEE International Conference on Acoustics. IEEE, 2009.

[3] 戴银云,易华,余涛.基于广义傅里叶变换的线性卷积算法（英文）[J].工程数学学报,2019,36(01):106-114.

[4] M. Paralic, R. Jarina. Iterative Unsupervised GMM Training for Speaker Indexing[J].Radioengineering, 2007, 16(3).

[5] D. Yu L. Deng and G. E. Dahl "Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition " in NIPS workshop on Deep Learning and Unsupervised Feature Learning 2010.

[6] S. Kapadia, V. Valtchev, and S. J. Young, “MMI training for continuous phoneme recognition on the TIMIT database,” in Proc. ICASSP, 1993, vol. 2, pp. 491–494.

[7] B. H. Juang, W. Chou, and C. H. Lee, “Minimum classification error rate methods for speech recognition,” IEEE Trans. Speech Audio Process., vol. 5, no. 3, pp. 257–265, May 1997.

[8] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri, “Discriminative training for large vocabulary speech recognition using minimum classification error,” IEEE Trans. Speech Audio Process., vol. 15, no. 1, pp. 203–223, Jan. 2007.

[9] D. Povey and P. Woodland, “Minimum phone error and i-smoothing for improved discriminative training,” in Proc. ICASSP, 2002, pp. 105–108.

[10] Tohidypour, Seyyedsalehi, Roshandel, et al. Speech recognition using three channel redundant wavelet filterbank[C]// International Conference on Industrial Mechatronics & Automation. 2010.

[11] N. Morgan and H. Bourlard "Continuous speech recognition using multilayer perceptrons with hidden markov models " in IEEE International Conference on Acoustics Speech and Signal Processing 1990.

[12] G.L. Licea-Haquet,E.P. Velásquez-U 冀瑞国.神经网络在语音识别中的应用[J].电子技术与软件工程,2019(03):249.

[13] 邢安昊,黎塔,颜永红.利用二重打分方法的激活词语音识别[J].声学技术,2013,32(S1):211-212.

[14] 陈志刚,刘权.人工智能技术在语音交互领域的探索与应用[J].信息技术与标准化,2019(Z1):16-20.

[15]王颢毅.基于人工智能技术的智能音箱发展现状与趋势探究[J].通讯世界,2018,25(12):225-226.

[16]贺玲玲,周元.基于改进 MFCC 的异常声音识别算法[J].重庆工商大学学报(自然科学版),2012,29(02):52-57.

[17]邵娜,李晓坤,刘磊,陈虹旭,郑永亮,杨磊.基于深度学习的语音识别方法研究[J].智能计算机与应用,2019,9(02):135-142.

[18]王霖,张琴,柳秀山.基于聚类算法的模糊语言识别系统的研究[J].中国新通信,2019,21(01):39-41.

[19]W Chan, N Jaitly, Q Le, and O Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in Proceedings of ICASSP, 2016.

[20]马树文.深度学习在语音情感识别中的应用与分析[J].科技传播,2019,11(04):147-148+155.

[21]杨洋,汪毓铎.基于改进卷积神经网络算法的语音识别[J].应用声学,2018,37(06):940-946.

[22]白海莉.情感语音合成技术或对声纹鉴定准确性产生影响[J].科技创新与应用,2018(36):24+26.

[23]Hinton, g., et al. deep neural networks for acoustic modeling in sr. IEEE Signal Processing 29, 11 (2012).

[24]Huiyong Wang. Multi-Level Adaptive Network for Accented Mandarin Speech Recognition[A]. IEEE Beijing Section.Proceedings of 2014 4th IEEE International Conference on Information Science and Technology[C].IEEE Beijing Section:,2014:4.

[25]Zhang, Y., Chan ,W., Jaitly, N.:VERY DEEP CONVOLUTIONAL NETWORKS FOR END-TO-END SPEECH RECOGNITION.In:Proc. International Conference on Acoustics, Speech and signal Processing(ICASSP 2017)

[26]Huiyong Wang. Multi-Level Adaptive Network for Accented Mandarin Speech Recognition[A]. IEEE Beijing Section.Proceedings of 2014 4th IEEE International Conference on Information Science and Technology[C].IEEE Beijing Section:,2014:4.

[27]张湘莉兰,骆志刚,李明.Merge-Weighted Dynamic Time Warping for Speech Recognition[J]. Journal of Computer Science and Technology,2014,29(06):1072-1082.

[28]yan, Z., huo, Q., and Xu, J. a scalable approach to using dnn-derived features in gmm-hmm based acoustic modeling for LVcsr. in Proceedings of Interspeech (2013).

[29]周晓兰.普通话水平测试系统中语音识别和语音评测技术研究[J].中外企业家,2016(29):265-266.

[30]Toth L . Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition[C]// IEEE International Conference on Acoustics. IEEE, 2014.

[31]胡石,章毅,陈芳等.基于 HMM 模型语音识别系统中声学模型的建立[J].通讯世界,2017(08):233-234.

[32]H. Hermansky and S. Sharma "Temporal patterns (TRAPS) in ASR of noisy speech " in IEEE International Conference on Acoustics Speech and Signal Processing 1999.

[33]Jaitly, Navdeep and Hinton, Geoffrey E. Learning a bet- ter representation of speech soundwaves using restricted boltzmann machines. In ICASSP, pp. 5884–5887, 2011.

[34]W. Fisher G. Doddington and K. Goodie-Marshall "The DARPA speech recognition research database: Specifications and status " in DARPA workshop on Speech Recognition 1986.

[35]G. E. Dahl D. Yu L. Deng and A. Acero "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition " IEEE Transactions on Audio Speech and Language Processing vol. 20 pp. 30-42 2012.

[36]马婉婕,孙虎元,孙立娟等.基于神经网络集成的手写识别系统[J].计算机应用与软件,2009,26(08):5-7+44.

[37]Molina D, Liang J L J, Harley R , et al. Comparison of TDNN and RNN performances for neuro-identification on small to medium-sized power systems[C]// IEEE, 2011.

[38]Alexander Waibel, Tashiyuki Hanazawa, Geoffrey Hinton, Kiyohito Shikano, Kevin J. Lang, Phoneme Recognition Using Time-Delay Neural Networks, IEEE Transactions on Acoustics, Speech, and Signal Processing, Volume 37, No. 3, pp. 328. - 339 March 1989.

[39]赵建民,王雨萌.基于 BP 神经网络学习率优化的研究[J].微型电脑应用,2018,34(08):89-92.

[40]Sithara A,Abraham Thomas,Dominic Mathew. Study of MFCC and IHC Feature Extraction Methods With Probabilistic Acoustic Models for Speaker Biometric Applications[J]. Procedia Computer Science,2018,143.

[41]阙大顺,赵永安,文先林等.基于 DHMM 和 VQ 的关键词识别系统研究[J].武汉理工大学学报,2011,33(02):140-143+152.

[42]pegui,T. Holtgraves,M. Giordano. Speech act recognition in Spanish speakers[J]. Journal of Pragmatics,2019,141.

[43]张晶晶,黄浩,胡英等.口语理解中改进循环神经网络的应用[J/OL].计算机工程与应用:1-8[2019-03-26].

[44]H Sak, M Shannon, K Rao, and F Beaufays, “Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping,” in Proceedings of Interspeech, 2017.

[45]曾国荪.改善神经网络反向传播算法的训练时间[J].小型微型计算机系统,1996(11):69-72.

[46]余玲飞,刘强.基于深度循环网络的声纹识别方法研究及应用[J].计算机应用研究,2019,36(01):153-158.

[47]杨丽,吴雨茜,王俊丽,刘义理.循环神经网络研究综述[J].计算机应用,2018,38(S2):1-6+26.

[48]Chenjie Sang,Massimo Di Pierro. Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network[J]. The Journal of Finance and Data Science,2018.

[49]艾虎,李菲.基于改进的长短期记忆神经网络方言辨识模型[J].科学技术与工程,2019,19(02):163-169.

[50]任勉,甘刚.基于双向 LSTM 模型的文本情感分类[J].计算机工程与设计,2018,39(07):2064-2068.

[51]Woellmer M , Eyben F , Schuller B , et al. Spoken term detection with Connectionist Temporal Classification: A novel hybrid CTC-DBN decoder[C]// Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on. IEEE, 2010.

[52]E Battenberg, J Chen, R Child, A Coates, Y Li, H Liu, S Satheesh, A Sriram, and Z Zhu, “Exploring neural transducers for end-to-end speech recognition,” in Proceedings of ASRU, 2018.

[53]P. Schwarz P. Matejka and J. Cernocky "Hierarchical structures of neural networks for phoneme recognition " in IEEE International Conference on Acoustics Speech and Signal Processing 2006.

[55]Y Miao, M Gowayyed, and F Metze, “EESEN: Endto-end speech recognition using deep RNN models and WFST-based decoding,” in Proceedings of ASRU, 2016.

发表文章目录

1、 Yumeng Wang. Continuous Speech Recognition Model Based on CTC Technology[A]. Wuhan Zhicheng Times Cultural Development Co., Ltd.Proceedings of 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018)[C].Wuhan Zhicheng Times Cultural Development Co., Ltd:2018:4.

Abstract： In end to end speech recognition， the linguistic knowledge such as pronunciation 1exicon is not essential。 and therefore the performance of the ASR systems based on CTC is weaker than that of the baseline． Aiming at this problem， a strategy combining the existing linguistic knowledge and the acoustic modeling based on CTC is proposed. and the tri—phone is taken as the basic units in acoustic modeling. Thus，the sparse problem of the modeling unit is effectively solved． and the discrimination and robustness of the CTC model are improved substantially.

Keywords: Connectionist Temporical Classification;Speech Recognition; End to End

2、赵建民，王雨萌 基于 BP 神经网络学习率优化的研究 [J] .微型电脑应用,2018,34(08):89-92.

摘 要：现今社会人工智能技术快速迭代发展，其应用也越发广泛，包括搜索、数学优化、逻辑推演等工具都应用了人工智能技术。神经网络作为人工智能的重要方法正在被不断地深入研究，而 BP 神经网络是经典的神经网络之一，在语音分析、图像识别、数字水印、计算机视觉等应用领域都取得了显著的效果。在对 BP 神经网络进行训练时，学习率的设置是众多参数中至关重要的一项。学习率选取不当将直接导致模型收敛速度慢、模型易越过全局极小值点等问题。因此本文针对 BP 神经网络中的学习率选取开展研究，将传统的固定学习率优化为变化学习率，从而有效提高 BP 神经网络模型的收敛速度以及精确度。

关键字：BP 神经网络；固定学习率；变化学习率

3、赵建民，王雨萌，王梅 基于深度学习的中文语音合成技术研究[J].计算机技术与发展（已录用，待发表）

摘 要：语音合成技术在近些年得到较快发展，但是在相似性和自然度等方面还是存在一定的问题，例如合成出的语音韵律差且机械音明显，使得其无法满足实际应用对语音质量的要求。将深度学习技术应用于语音合成，目前已经可以表现出和基于隐马尔科夫模型以及基于语音拼接的语音合成技术相当的合成性能。由于神经网络拥有强大的建模能力，而且有更好的灵活性、易控制性，所以深度学习技术在提升语音合成性能方面有很大的潜力和研究价值。其中，递归神经网络（RNN）、长短时记忆神经网络（LSTM）以及门阀递归单元神经网络（GRNN）是几项典型的深度学习模型，普遍被应用于语音识别、机器翻译时序建模方面。本文首先通过多组对比实验，展示这些深度学习技术在语音合成方面的性能。然后针对深度学习模型都存在复杂度高，训练效率低的问题。本

文提出了对 GRU 模型进行优化改进，该模型对标准 GRU 的记忆单元的输出值做降维，然后传递给下一时刻的神经单元从而降低了计算量。本文命名其为 PGRU 模型。通过本文实验，本文验证了 PGRU 在不影响模型建模能力的前提下，模型的训练速度与优化前相比相对提升了 10%，证明了本文方法的有效性。

关键词：深度神经网络；语音合成；长短时记忆；门阀递归单元

致　谢

回首过去的三年，时光荏苒，白驹过隙。不知不觉，硕士三年之期将满，这三年来我在老师们的指导与同学们的帮助下，顺利完成的硕士论文的全部工作。所以在我临毕业之际，我想对你们表达我内心无限的感谢！

首先，我要感谢我的指导教师赵建民老师，赵建民老师从我硕士入学的第一天起，就在学习和生活中给予了我极大地关心与帮助。在我硕士选题期间，赵建民老师尊重并支持我的个人意愿，让我可以在自己感兴趣的研究方向专心学习；在我课题研究期间，赵建民老师不厌其烦的给予我指导、教学，帮助我克服重重难关；在我撰写论文期间，赵建民老师对我的写作思路、论文框架以及论文内容反复推敲，层层把关，耐心的指导我顺利完成了硕士研究生毕业论文。

再者，我要感谢实验室其他老师们，包括袁文翠老师、孙丽娜老师、李井辉老师等，感谢黑龙江工程学院计算机科学与技术学院杨茹教授的精心指导。老师们待人随和、治学严谨的态度深深影响着我。

再者，我要感谢操海兵同学，操海兵同学在我对语音识别技术的入门阶段就陪伴我一起学习，并在我学习遇到困难时给予我严厉严谨的辅导。同时，在我因为模型实验结果准确率过低而一筹莫展的时候给予我技术上的帮助。

再者，我要感谢我在搜狗公司的同事们，他们在我进行基于神经网络分类方法实验的初期，曾耐心为我解答在链接时序分类方法的不明之处，并在实验创新设计上给予我很多专业的建议。

再者，我要感谢我的同学、室友、师兄师姐们，三年的同窗学习，我们结下了深厚的友谊，感谢你们的陪伴，让我的研究生生活多姿多彩。

再者，我要感谢我的父母，父母养育我成人，并在我人生遇到岔路不知如何抉择时，一直对我的决定支持和鼓励，让我可以心无旁骛的将精力都投入到学习中。

最后，我要感谢所有答辩组的老师，是老师们专业的意见和认真负责的态度，才能让我及时发现论文中的不当之处，让我完善毕业论文的撰写。

由衷感谢以上每位为我付出过的人！

艰苦创业 严谨治学

严谨朴实 勤奋创新

招生办：0459-6503721

培养办：0459-6504792

学位办：0459-6503938

学校网址：http://www.nepu.edu.cn

1.3 相关技术研究

1.3.1 语音识别基础原理

语音识别是通过分别训练声学模型 AM（Acoustic Model）和语言模型 LM（Language Model）并结合发音词典，使得给定一段语音的声学特征输入解码器后找到一个词串，使得文字串的后验概率最大化的过程[18-23]。语音识别基本原理的流程如下图 1.1 所示。

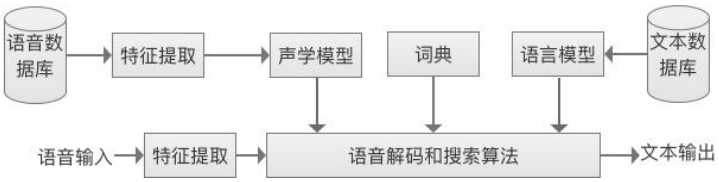

图 1.1 语音识别基本原理图

由图 1.1 可得，首先定义，O 为波形经过分帧处理后得到的声学特征向量，表达为式 1-1 所示。

$$\mathrm{O} = o_1,o_2,...o_t \tag{1-1}$$

定义 W 为识别后输出的文字序列，表达为式 1-2 所示。

$$\mathrm{W} = w_1,w_2,...w_n \tag{1-2}$$

根据语音识别的目的，将输入的语音信号识别为对应的文字序列，则语音识别原理如式 1-3 所示。

$$W_{max} = argmax_w P(W|O) \tag{1-3}$$

根据贝叶斯公式推导可得式 1-4。

$$W_{max} = \frac{argmax_w P(O|\mathrm{W})P(W)}{P(O)} \tag{1-4}$$

对于识别结果 W 来讲，输入 O 是不变的，所以可将式 1-4 简化为式 1-5。

$$W_{max} = argmax_w P(O|\mathrm{W})P(W) \tag{1-5}$$

其中，$P(O|W)$为给定文字观测序列条件下产生声音的概率，即通常所说的声学模型；$P(W)$为文字序列的先验概率，即通常所说的语言模型。由以上公式同样可以看出，语音识别就是将声学模型和语言模型的识别结果进行结合，从而达到将语音转变为文字的目的。

1.3.2 神经网络基本结构

1．神经元基本结构

神经网络是一种模拟人脑神经网络以期能够实现类人工智能的机器学习技术。神经网络的训练过程为：首先给定一组输入 $\mathrm{x}[x_1...x_n]$，通过训练算法使网络学习一组连接权重 $\mathrm{w}[w_1...w_n]$，然后计算它们的输出 $\mathrm{y} = \sigma(w_1x_1 + w_2x_2... + w_nx_n)$。

多个神经元通过线性连接就可以组成一个最简单的神经网络。每一层网络的输出将作为下一层网络的输入[24]。神经元示意如下图 1.2 所示。

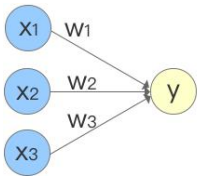

图 1.2 神经元示意图

2．前馈神经网络

前馈神经网络（Feedforward Neural Network），又称多层感知机或多层感知器。前馈神经网络的网络结构由输入层，隐藏层和输出层组成，其中隐藏层数量越多，模型拟合效果越强。其结构如下图 1.3 所示。

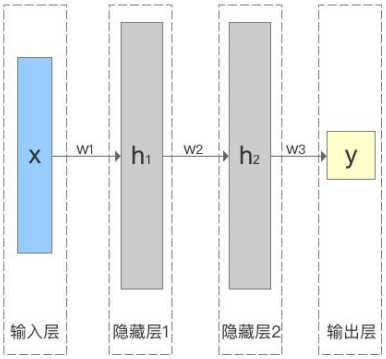

图 1.3 前馈神经网络结构图

在前馈神经网络中，所有信号全连接，信号由正向传导，误差由反向传播。前馈神经网络通过定义 $y = \sigma_3(w_3\sigma_2(w_2\sigma_1(w_1x + b_1) + b_2) + b_3)$在模型学习 w 的过程中，实现输入到输出的映射。

前馈神经网络在语音识别中可用来处理上下文信息。在语音识别特征提取时，当提取长时包含当前帧和前后帧之间的依赖关系的声学特征后，可将包含上下文信息的特征直接输入前馈神经网络，实现对声学单元的建模。

3．时延神经网络

时延神经网络 TDNN（Time-Delay Neural Network）是多层前馈神经网络的一种，同时也是本文方法循环神经网络的基础雏形[25]。TDNN 与前馈神经网络的不同在于，对于时序信号，TDNN 的每个单元都具有到下面单元的输出连接，同时也具有该单元相同单元的连接作为延时输出，该过程形象的模拟了序列的时间轨迹[26]。TDNN 可以对具有移位不变性的模式进行分类并且可以在网络的每一层对模型包含上下文信息的单元进行分类。其模型结构如下图 1.4 所示。

图 1.4 中第一个隐藏层的节点只与输入层的 3 帧语音特征有连接，随着隐藏层层次增加，它所处理的时序长度越来越长，所以层次越深的隐藏层就有能力学习更宽的时序信息。

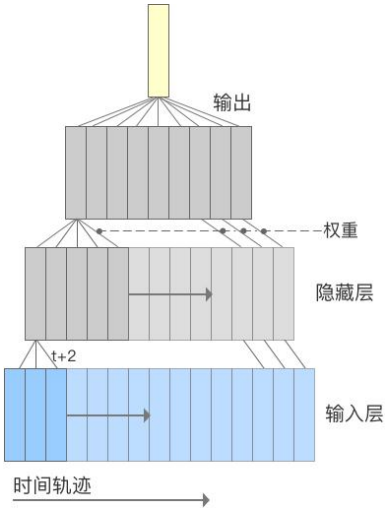

图 1.4 时延神经网络结构图

TDNN 在建立声学模型时，根据短时声学特征来学习信号的时序动态性。图 1.4 中，第一个隐藏层对所有上下文信息（整体的神经网络的输入）做仿射变换[27]。同样，更高层的隐含层对前一层的所有输出做仿射变换，构成了全连接的网络结构。而普通的前馈神经网络只在一个比较窄的时序上下文上做变换。另外，TDNN 在对声音进行分类时，具有位移不变的分类特性，移位不变分类意味着分类器在分类之前不需要显式分割。所以不必确定声音的起点和终点。

1.3.3 神经网络的传播过程

最基础的神经网络为前馈神经网络，前馈神经网络最典型的则为含有单一隐藏层的 BP 神经网络。BP 神经网络具有非线性映射能力强，容错能力强，泛化能力强等优点。标准的 BP 神经网络模型包含三层，分别是输入层、隐藏层和输出层[28]。BP 神经网络中信息前向传播和误差反向传播的过程如下。

1．信号的正向传播输入过程

定义 BP 神经网络如下：

输入层 X 含有神经元 n 个，隐藏层 H 含有神经元 m 个，输出层 Y 含有神经元 l 个，则网络输出如式 1-6 所示。

$$Y_k=f(\sum_{j=0}^{m} w_{jk}h_j) \qquad (k=1,2,\ldots l) \qquad (1\text{-}6)$$

其中，w_{jk}为隐藏层到输出层的权值矩阵，h_j为隐藏层输出，f（x）为非线性激活函数。

2．误差的反向传播过程

定义预期的输出为 D，损失函数选用均方误差函数，则网络损失如式 1-7 所示。

$$E=\frac{1}{2}\sum_{k=1}^{l} d_k-y_k \qquad (1\text{-}7)$$

定义输入层到输出层的权重矩阵为 V，则误差由输出层反向传播回输入层的信息如式 1-8 所示。

$$\mathrm{E}=\frac{1}{2}\sum_{k=1}^{l}(d_k-f(\sum_{j}^{m}w_{ij}f(\sum_{i=0}^{n}v_{ij}y_i)))^2 \tag{1-8}$$

利用梯度下降对误差进行优化，定义输入层到隐藏层的网络结构为net_j，隐藏层到输出层网络为net_k，则有输出层权值优化如式 1-9 所示。

$$\frac{\partial E}{\partial V}=\frac{\partial E}{\partial net_j}\cdot\frac{\partial net_j}{\partial v_{ij}}=-(\sum_{k=1}^{l}(d_k-y_k)f^{''}(net_k)w_{jk}f^{''}(net_j)x_i) \tag{1-9}$$

隐藏层权值优化如式 1-10 所示。

$$\frac{\partial E}{\partial \mathrm{w}_{jk}}=\frac{\partial E}{\partial net_j}\cdot\frac{\partial net_j}{\partial w_{jk}}=-(d_k-y_k).f^{''}(h_j).h_j \tag{1-10}$$

因为 TDNN 也是前馈神经网络的一种，其传播过程也是通过信息的前馈和误差的后项传播来实现模型参数的学习的，故此节就不再对 TDNN 的信息传播过程进行赘述。

1.4 本文研究主题及章节安排

1.4.1 本文研究主题

本文的研究主题是基于神经网络时序分类在端到端语音识别技术中的研究。本文将链接时序分类算法 CTC（Connectionist Temporal Classification）与改进的双向映射长短期时序网络（Bi-PLSTM-CTC）相结合的非完全端到端模型在中文语音识别技术的应用研究。将神经网络时序分类方法应用于语音识别技术中，不仅可以简化模型的框架结构、增强模型的可优化性，同时因为端到端模型有效降低了传统模型中对每帧语音与状态间硬性对齐的要求，所以端到端模型可以显著的提升语音识别的准确率。

为了更好地实现神经网络时序分类方法在语音识别中的应用研究，本文主要以传统语音识别的模型和算法为基线，通过对比实验，研究神经网络时序分类方法在语音识别中的有效性。最后，本文针对双向神经网络模型在训练时要求大量数据的特点，在实验时结合速度扰动技术，保证数据高保真的前提下，扩充实验数据量，提升模型的速率与准确率。

1.4.2 本文章节安排

本文内容安排如下：

第一章：绪论。

本章首先阐述了本文的研究背景和选题意义，接下来介绍了语音识别技术的国内外发展现状，然后对本文涉及的相关技术基础做了简要的阐述，最后给出本文的研究主题以及论文具体的章节安排。

第二章：基于中文语言模型的非完全端到端语音识别框架。

本章首先分析了从语音识别发展以来几种主流的结构框架，并分别对每种框架结构识别流程进行了详细分析，其中包括特征提取、声学模型、语言模型和词典等。然后本章通过对比分析不同框架结构的优缺点，设计了一种结合语言模型的非完全端到端模型框架。本章是全文的理论基础。

第三章：基于自适应技术的 GMM-HMM 算法模型。

本章首先针对传统声学模型 GMM-HMM 模型中的基本算法进行分析，其中包括隐马尔科夫模型原理及其算法过程、混合高斯模型参数估计及其算法过程。然后针对传统模型对于非特定说话人易产生语义混淆的问题，设计了结合自适应技术的隐马混合高斯模型，并对自适应技术算法原理进行了阐述。本章实验结果将做为本文的基线实验，证明本文方法的有效性。

第四章：基于 Bi-PLSTM 的链接时序分类算法模型。

本章首先介绍了循环神经网络及其变种长短期记忆时序网络的相关基础理论，分析了传统 RNN、LSTM 在处理时序信息的性能优劣。针对 LSTM 计算复杂度较高的问题，提出了一种改进的 PLSTM，同时针对单向网络无法获取反向时序特征的问题，提出了利用双向网络结构替代单向 RNN 与 CTC 算法结合的训练方法。最后本章对链接时序分类的建模方法以及如何对融合语言模型的解码进行了详细的分析。

第五章：实验设计及结果分析。

本章首先对基于 GMM-HMM 模型进行了实验验证了，其中包含单音素建模、三音素建模两种建模方式，得到了本文的初步基线实验结果，接下来，通过对基于说话人自适应技术的 GMM-HMM 模型进行实验，取得了本文最终的基线实验结果。然后，本章对融合了语言模型的 Bi-PLSTM-CTC 模型进行了实验验证，并设计添加速度扰动技术优化实验结果。最后，通过对比分析传统模型与本文所提方法的实验结果，验证了本文方法的有效性。