CSLT TECHNICAL REPORT-20180002 [Wednesday 26th December, 2018]

A Research of Automatic Speech Recognition

Wu Jiayao

Correspondence: wujiayao@cslt.org
Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

本文作为对语音识别的研究性报告，在第一部分介绍了语音识别的前沿科技。第二部分介绍了语音识别的基本原理和不同的语音识别系统结构。第三部分介绍了kaldi中thchs30的脚本流程，最后一部分展示了实验结果。

Keywords: 语音识别; 神经网络

1 国内外前沿科技

如今，一些商业语音识别系统已经取得了很好的识别效果，国外有苹果公司的Siri，微软的Cortana等虚拟助手，亚马逊的Alexa等家居助手；国内有科大讯飞、百度语音、阿里语音等。在近场等情况下，一些任务如语音搜索（voice search, VS）、短信听写（SMS dictation, SMD）等词错误率已经可以媲美人类水平。在一定程度上可以认为近场单人语音识别问题已经被解决，但其实语音识别还存在有如下问题：
(a)远场麦克风语音识别。
(b)高噪音环境下的语音识别。
(c)带口音的语音识别。
(d)不流利的自然语音，变速或者带有情绪的语音识别。
当语音识别系统处于上述场景时，稳定性急剧下降。针这些语音识别问题，本文做了一些针对性的学术进展调研。

1.1 GAN去混响技术

1.1.1 概念

当语音交互场景从进场变成远场，房间混响成为一个影响语音识别性能的关键问题。声音由直达声、早期反射和晚期混响组成。在很多语音产品如智能家居语音助手的交互环境中，接收到的语音信号除了原始发声源信息还有无法避免的混响，导致识别性能下降。

1.1.2 相关理论

理论上，包含混响的语音信号可以看做干净语音和脉冲响应的卷积。从这个角度出发，可以利用深度学习神经网络的学习能力做回归任务，学习一个从带混响的语音输入到干净语音输出映射函数，从而达到过滤信号、去混响的目的。 解决思路是可以通过干净语音构造很多的混响语音数据，来训练这样一个映射网络。

1.1.3 相关算法

在研究[1]中用目前在学界热度很高的生成对抗网络(Generative Adversarial Network, GAN)解决混响问题。GAN目前已经拓展出很多网络变体，但基本的结构一般由生成器和判别器两个网络组件构成。生成器输入带混响语音信号特征，输出生成语音信号特征。生成器生成的语音信号特征和对应的干净语音信号特征共同

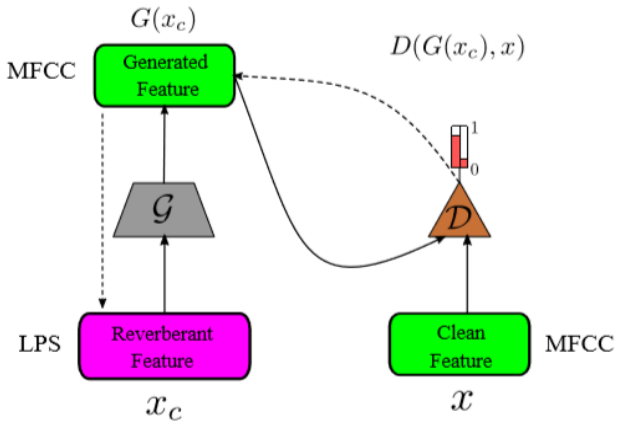

Figure 1: GAN去混响模型

作为输入给判别器，通过多次迭代博弈，达到生成干净语音信号特征的目的，也即赋予生成器去混响的能力。 具体实现中，由于LSTM较强的时序建模能力，生成器网络用LSTM网络效果最佳。其次，随着网络层数加深，加入残差网络可以进一步提高效果。而网络训练过程中，用同一个Mini-batch的数据去更新生成器G和判别器D能提升网络性能。

如图1所示，为GAN去混响的具体模型结构示意图。最终在同一数据集下，去混响GAN相较于去混响DNN网络得到14%-19%的相对字错误率下降。

1.2 说话人自适应解决方言口音问题

1.2.1 概念

在语音识别具体场景中，不同语种由于小范围地域差异也会出现各种方言变体，对语音识别的准确率造成了很大的影响，如何解决方言问题和口音问题是语音识别的一个重要任务。

1.2.2 相关理论

近年，采用说话人自适应（Speaker Adaptation，SA）技术能有效解决该类问题。在一个已经训练好的初始系统上，用一定的新说话人语音数据来提高系统对新说话人的建模精度。说话人自适应的具体方法大致分为三类：线性变换法，子空间方法和保守训练法。线性变换方法包括线性输入网络（linear input network，LIN)，线性隐层网络（linear hidden network ，LHN)和线性输出网络（linear output network ，LON)等。子空间方法旨在找到一个描述说话人特性的子空间，然后构建自适应的网络权值，例如i-vector方法。保守训练法通过在自适应准则上加一个正则项来得到。

1.2.3 相关算法

在研究[2]中，对说话人自适应的三种方法进行了性能比较。

1 第一种方法是（linear input network, LIN），属于线性变换类别。在原始网络的模型基础上在输入端加入线性变化层，把不同人的语音输入变成通用特征，原始网络参数不做任何变化。也就是将说话人相关的特征从通过线性变换变换到另一个与说话人无关的能与DNN匹配的特征向量

2 第二种方法是直接用新的说话人语音数据调节神经网络模型参数。为了避免过拟合问题，采用（Kullback-Leibler divergence，KLD）准则在模型自适应过程中来做一个约束，使得适应后模型的后验概率分布与说话人无关神经网络模型的后验分布越接近越好。这种方法需要为每位说话人训练一个单独的网络模型。

3 第三种方法是(learning hidden unit contribution, LHUC)，通过定义一组说话人相关的参数来调节网络模型参数。

实验结果表明，KLD能取得最好的效果，但是上述方法相较于传统模型都有识别错误率的下降。

1.3 端到端模型

1.3.1 概念

传统自动语音识别系统（automatic speech recognition, ASR）由声学模型（acoustic model, AM）、词典（Lexicon）和语言模型（language model, LM）组成。不同组件往往单独训练，且训练数据集不同。 端到端模型（Sequence to sequence model）还原语音识别序列到序列的问题本质，将传统的ASR系统独立训练的声学模型、词典和语言模型 （AM，Lexicon，LM）合并成单个神经网络，成为当下语音识别领域研究热点。

1.3.2 相关理论

（一）基于注意力机制的Seq2Seq框架

近日，谷歌开发了Sequence-To-Sequence[3]端到端语音识别框架。新系统建立在 Listen-Attend-Spell（LAS）[4]端到端结构基础之上。如图2所示，LAS 系统由三个模块组成，分别是听者编码器组件（listener encoder component），参与者组件（attender）和speller（即解码器）。Encoder将输入特征序列映射到 一个高级特征表示序列——h^{enc}，通过attention注意力机制学习输出单元和特征序列的对齐关系，而speller将输出单元构成完整的一句话。有趣的是，encoder 和speller 分别承担了和传统语音识别系统中AM和LM类似的工作。LAS 模型的所有组件都是统一训练，实现了输入语音数据特征、输出句子的端到端语音识别任务。

在新的系统上，谷歌提出了结构上的提升和优化训练过程的方法。在结构上，提出了更长的子字单元（word pieces）的优化方法，达到提高解码效率和性能提升的效果；同时还改进了注意力机制，如图3所示，从单头结构变为多头注意力（multi-head attention, MHA）结构。在训练方法上，提出了包括最小词错率训练法（minimum word error rate training）；采用预定采样（scheduled sampling）的方法训练解码器；用标签平滑正则化（Label Smoothing, LS）的方法降低模型过拟合度，同步训练（synchronous training）的方法获得更快的收敛速度和更好的模型质量。正是这些结构化和训练方法优化提升使新模型取得了相对于传统模型的性能提升。

（二）CTC（Connectionist Temporal Classification）框架

在传统的语音识别系统中，对语音模型训练之前，需要将文本和语音进行严格的对齐操作，耗费人力和时间。CTC框架如图4所示，不再进行逐帧判别，采用CTC损失函数训练，基于序列训练准则，可以进行端到端训练，而在系统结构中，往往外接神经网络语言模型获得更好的识别结果。

需要说明的是，目前端到端系统相对传统语音识别系统来说只是差距的缩小，在很多场景下识别性能相对混合系统来说还有差距，因此还有很多研究空间。

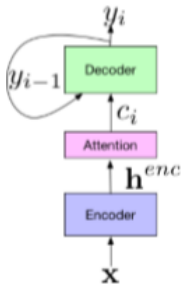

Figure 2: LAS端到端结构

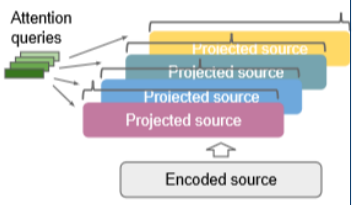

Figure 3: MHA结构

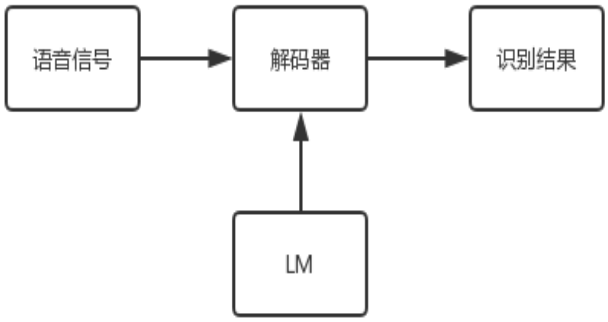

Figure 4: CTC框架结构图

1.4 多任务学习和迁移学习

在深度神经网络（Deep neural network, DNN）中，每个隐藏层都是输入DNN的原始数据的一种新特征表示，较高层次的表征比较低层次的表征更为抽象，而这些特征的表示可以通过多任务（multi-task）和迁移学习（transfer learning）等技术共享和迁移到相关的任务，即通过共享隐层的DNN架构来实现。

1.4.1 概念

多任务学习（Multitask learning, MTL）是一种通过联合训练同时学习多个相关任务从而提高模型泛化能力的机器学习方法。相关不意味相似，而意味着在一定的抽象层次上能共享一部分特征表示。

迁移学习是指将在一个任务上学到的知识迁移到其他任务上。与多任务学习不同的是：多任务学习目的是提升所有或一个主要任务的性能，迁移学习强调的是通过迁移在相似但不同的任务上获得的知识来提升目标任务的性能。

1.4.2 多任务学习结构

如图5所示为一种同时训练语音识别（ASR）系统和说话人识别（SRE）多任务学习[5]的框架示意图。

说话人任务和语音识别任务在任务上具有相关性，因为都是处理语音信号特征；但是数据集不具有相似性，因为语音识别主要是对语音内容进行识别，标注特

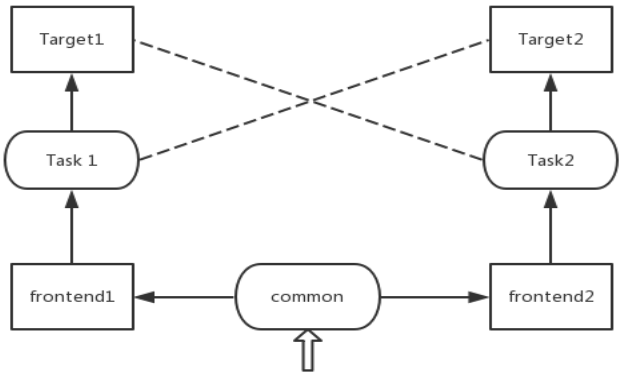

Figure 5: 多任务学习结构

征是语音的语义，在特征提取的时候需要降低说话人的影响，而说话人识别的任务就是区分说话人的不同，这两个任务在一定程度上是互相抵抗的，但是通过多任务学习对不同任务的泛化能力，论文[5]表明通过将两个任务的神经网络进行特定方法的连接，能使两种任务的神经网络性能都有所提升。

1.4.3 迁移学习结构

如图6所示为一种跨语音识别神经网络结构图。

这种结构就是共享隐层的多语言神经网络结构。输入层和隐层被所有语言共享，但是输出层不被共享，而是每种语言有自己的softmax层来估计后验状态。

2 语音识别系统简介

如图7所示为目前较为常见的语音识别的基本框架流程图。

而语音识别的基本公式为$\overset{argmax}{W}\mathrm{P}(W \mid X) = \overset{argmax}{W} \mathrm{P}(X \mid W)\, P(W)$。 其中$P(X \mid W)$为声学模型，$P(W)$为语言模型。

在语音识别的发展历史上，声学模型一直被一个浅层的隐马尔科夫-混合高斯模型（HMM-GMM）所统治，虽然达到了一定程度上的识别率，但是效果仍然不理想，和人类语音识别能力相距甚远。直到2010年后，随着计算机计算能力的大幅提升，神经网络重新回到人们的视线中。研究人员将神经网络应用到语音识别领域，语音识别终于取得了突破性的进展，短短几年，训练出来的神经网络模型在某些数据集上的表现已经可以媲美人类水平，并已成功在智能家居、智能手机等多种实用场景落地。

纵观神经网络的发展史，取得较大突破的网络结构都具有以下特点：

(1)对语音信号的时序信息有较强的建模能力

(2)能添加语音信号的长时依赖特性，从而提高模型性能

(3)易于训练，结构较为简单

语言模型目前最为流行的还是N-gram模型，但是随着神经网络的兴起，人们也开始尝试用神经网络对语言模型进行建模。

2.1 声学模型(Acoustic model)

目前深度神经网络中较具代表性的网络结构主要有三大类：全连接前馈神经网络（full-connected neural networks, FNN）,卷积神经网络（convolutional neu-

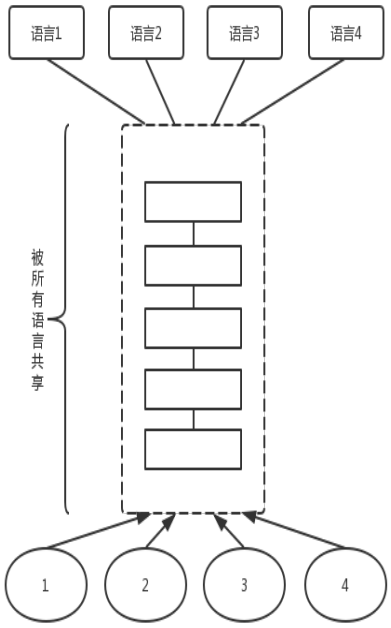

Figure 6: 跨语言识别神经网络图

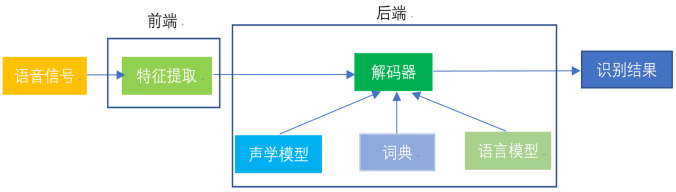

Figure 7: 语音识别流程图

ral network, CNN）以及循环神经网络（Recurrent neural network, RNN）等。特别的网络结构有空间变换网络（Spatial Transformer network ）/ Highway Network(Grid lstm) / Recursive structure / External Memory / Batch Normalization / Sequence-to-sequence/ Attention等。

而一个神经网络的训练主要包括三个步骤：

(1)定义网络结构。一个复杂的神经网络由简单神经元组成，神经元通过不同的权重和偏置产生连接，并且连接方式也有所不同。总的来说，数据信息通过输入层、隐藏层到输出层的顺序层层传递，网络连接方式需要根据不同的需求进行自定义。一个神经网路通过神经元的连接方式和超参数来诠释。

(2)定义损失函数（cost function）来评判网络参数的好坏，而损失函数的定义根据不同的任务也有所不同。

(3)训练方法的确定。常见的训练方法有反向传播算法（backpropagation, BP）、沿时反向传播算法（BPTT）等。

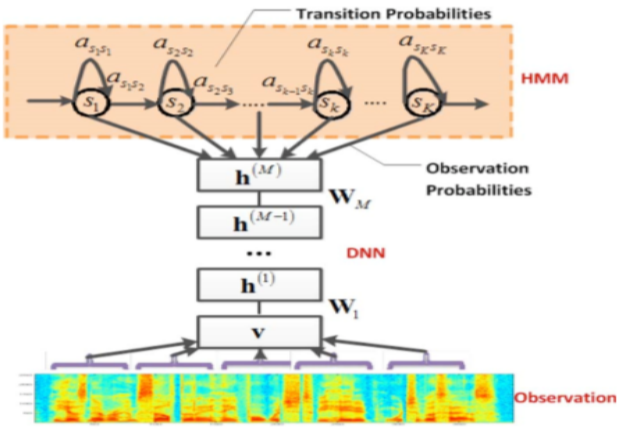

Figure 8: DNN-HMM结构图

本节总结的目前较为流行的声学模型主要包括以下四大类：深度神经网络-隐马尔科夫模型混合系统（DNN-HMM），时延神经网络（Time delay neural network ,TDNN ）, 前馈序列记忆神经网络（Feedforward sequential memory networks，FSMN）以及循环神经网络（recurrent neural network , RNN）。 在信号处理学科中，有两种滤波器，分别叫做 IIR 和 FIR（Infinite Impulse Response Filter vs. Finite Impulse Response Filter），它们和两种神经网络相对应。IIR 就相当于 RNN 模型，FIR 就相当于 CNN 模型，在卷积了足够多层之后，CNN就能利用足够远的信息（类似 RNN）。就好像在很多场景下，FIR 滤波器是可以近似 IIR 滤波器的。

对不同声学模型的介绍主要遵循以下几点：

(1)模型结构的阐述。首先将其结合最基本的FNN、CNN以及RNN从模型结构和具体的公式表达两方面进行阐明。

(2)重要的改进。其次分析这些声学模型针对语音建模的要素在相应的基础神经网络上所做的改进以及改进的思路和原因。

(3)模型训练方法。然后对相应的模型训练方法进行说明。

(4)优缺点比较。最后将几种典型的声学模型的优点和缺点进行对比。

2.1.1 深度神经网络-隐马尔可夫模型混合系统*(DNN-HMM)*

（一）模型结构

论文[6]中俞栋老师等人关于CD-DNN-HMM模型的研究工作可以算是语音识别领域从HMM-GMM模型结构到神经网络模型发展的转折点。 DNN不能直接为语音信号建模。因为语音数字信号是时序连续信号，而DNN需要固定大小的输入，但是DNN的强分类能力对不同的音素有很好的鉴别效果。因此需要找到一种方法来处理信号长度变化的问题。而如图8所示的DNN-HMM混合系统可以在实际问题中广泛应用。

在这个框架中，HMM用来描述语音信号的动态变化，即对语音的序列特性进行建模，观察特征的概率则通过DNN来估计，相较于传统的HMM-GMM结构的区别是：原先GMM提供P\{特征 | 状态\}，而现在变为DNN提供P\{状态 | 输入\}。为添加上下文信息，DNN对所有聚类后的状态（聚类后的三音素状态）的似然度进

行建模。在给定声学观察特征下，用DNN的每个输出节点来估计连续密度HMM某个状态的后验概率。但是在训练过程中，还是需要GMM+HMM 系统提供对齐方式。 对于所有的状态$s \in [1, S]$，只训练一个完整的DNN来估计状态的后验概率$p(q_t = s \mid X_t)$。典型的DNN输入不是单一的一帧，而是一个$2\varpi + 1\,(9 - 13)$帧大小的窗口特征$X_t = [O_{max}(0, t - \varpi)...O_t...O_{min(T,t+\varpi)}]$，使得相邻帧的信息被有效地利用，部分缓和了传统的HMM无法满足观察值独立性假设的问题。

（二）特点总结

模型结构的优点如下：

(1)充分利用了DNN的内在鉴别属性

(2)训练过程可以使用维特比算法，解码通常非常高效

(3)来自CD-DNN-HMM的DNN的后验概率代替了传统GMM-HMM中的混合高斯模型，其他都保持不变，结构改动较小。

模型结构的缺点如下：

(1)HMM对上下文的建模能力有限，因为HMM的马尔可夫性导致当前状态只和上一时刻的状态有关。

(2)性能提升还未接近人类水平。

2.1.2 时延神经网络*(Time Delay Neural Network,TDNN)*

（一）模型结构

用于语音信号处理的时延神经网络（TDNN）其实是卷积神经网络（Convolutional neural networks, CNN）的前身。因此在此处将TDNN和CNN结构类比。其结构特点与传统的神经网络类似，包含输入层、隐层和输出层，且通过权重和偏置一一连接。但是为了对语音信号中的动态时域信息进行建模，TDNN做出了一些改进，加入了上下文信息，即隐含层的特征不仅与当前时刻的输入有关，而且还与未来时刻的输入有关。 通过这种精妙的结构设计，TDNN能用于对来自短期语音特征（即MFCC）的长期时间依赖性进行建模[7]。

(1)标准的TDNN（Time delay neural network ，TDNN）

如图9所示为语音包kaldi中使用的标准TDNN结构[8]，初始变换从较为狭窄的时间宽度上学习，而较深层的是从更广泛的时间上下文中处理，即随着隐层层数的变化，TDNN每一层都以不同的时间分辨率进行，较高层具有学习更宽时间的能力。

时延神经网络训练过程中采用子采样（sub-sampling）的方法减少训练过程中的计算量。

在典型的TDNN中，在所有时间步骤计算隐藏激活。然而，在相邻时间步长计算的激活的输入上下文之间存在大的重叠。在相邻激活相关的假设下，可以对它们进行子采样。在网络的隐藏层中，通常拼接不超过两帧。而且层数越高，拼接的帧距相隔越远。Sub-sampling的方法能让训练时间提升5倍。Sub-sampling的另一个优点是模型尺寸的减小。

（二）和CNN模型结构类比分析

TDNN作为CNN的前身，和CNN有相同的结构设计思想。设计滤波器的参数，其中滤波器参数在时间维度上共享，从而达到精简模型结构，并且包含上下文信息的目的。

（三）训练方法总结

(1)标准的TDNN（Time delay neural network ，TDNN）

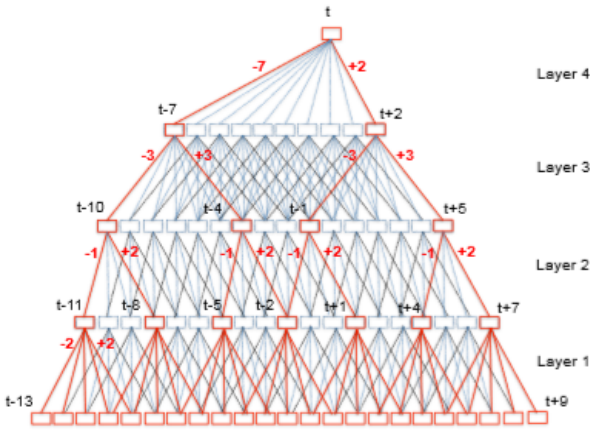

Figure 9: Kaldi中TDNN结构图

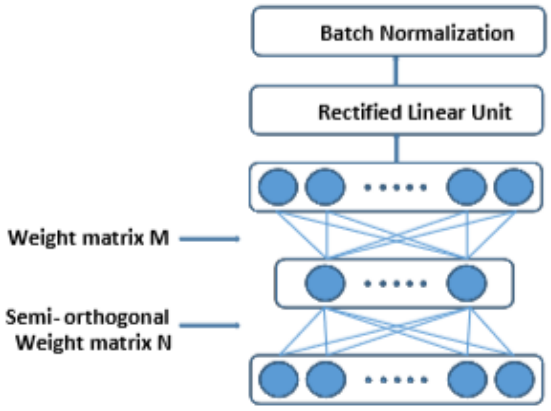

Figure 10: TDNN-F结构图

训练方法使用贪婪的分层监督训练，预处理随机梯度下降更新参数，学习率采用指数下降法，开始时学习率要比较大，加快学习速度；到后面学习率要小增加训练精度。其中训练准则采用sMBR序列鉴别性训练。

(2)分解的TDNN（factorized TDNN, TDNN-F）

在研究[9]中提出一种有效的方法来训练具有参数矩阵的网络，所述参数矩阵表示为两个或更多个较小矩阵的乘积，其中除了一个因素之外的所有因子都被约束为半正交，如图10所示。将此方法应用于TDNN系统，即为分解的TDNN（TDNN-F），并应用其他一些改进，例如跳转连接（skip connection）和跨时间共享的丢失掩码方法，通过这些优化方法 TDNN-F模型通常会取得更好的识别结果，同时解码速度更快。

（四）模型特点分析

TDNN声学模型在建模和训练上有以下特点：

(1)训练时间相对较短。虽然循环神经网络（RNN）在语音信号上表现出了较强的建模能力，但是因为其序列特性，训练难度大且非常耗时。相较而言，时延神经网络（TDNN）本质上是前馈神经网络，因此容易训练，耗时较短。

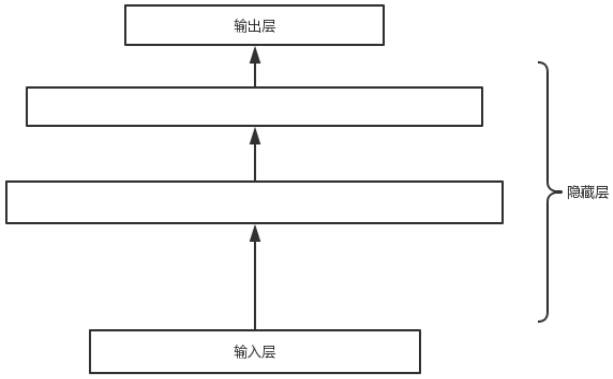

Figure 11: FNN结构图

(2)网络是多层的，每层对特征有较强的抽象能力。
(3)有能力表达语音特征在时间上的关系。
(4)具有时间不变性。
(5)学习过程中不要求对所学的标记进行精确的时间定位。
(6)通过共享权值，方便学习。

2.1.3 前馈序列记忆神经网络*(FSMN)*

FSMN声学模型结构是张仕良在博士期间从事ASR研究的创新成果。随后加入阿里巴巴语音算法组后，与同事们一同对FSMN结构进行了一系列改良，最终的LFR-DFSMN，已经成功落地工业级应用，取得识别率和解码速度的大幅度提升，刷新了英文开源数据集Librisspeech的世界当前最好结果。

（一）模型结构特点

FSMN[10]其本质是一个前馈全连接神经网络（FNN）。前馈全连接神经网络顾名思义，不同层的所有神经元两两之间都有连接关系。前馈全连接神经网络是一种单向结构，每层包含若干神经元，同层神经元没有连接，信息沿着一个方向层层传递。数据从输入层输入，经过多层隐层，最后从输出层输出。具体的模型结构如图11所示。

(1)标准前馈序列记忆神经网络（Feedforward sequential memory networks，FSMN）

在前馈全连接神经网络的基础上，通过在隐层旁边添加一些记忆模块（memory block）来对周边的上下文信息进行建模，从而使得模型可以对时序信号的长时相关性进行建模。记忆模块采用如图12所示的抽头延迟结构将当前时刻以及之前N个时刻的隐层输出通过一组系数编码得到一个固定的表达。

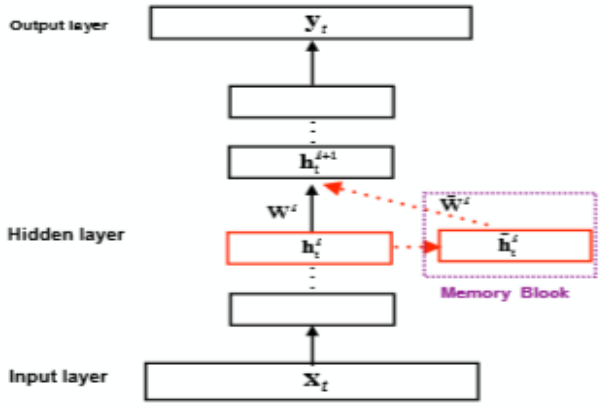

Figure 12: FSMN结构图

输入序列：$X = \{x_1, ..., x_T\}, x_t \in R^{D \times 1}$

第l隐层输出序列：$H^l = \{h_1^l, ..., h_T^l\}, h_t^l \in R^{D_l \times 1}$

记忆模块：$\tilde{h}_t^l = \sum_{i=0}^{N} a_i^l \bigodot h_{t-i}^l$

第l+1隐层输出序列：$h_t^{l+1} = f\left(W^l h_t^l + \tilde{W}^l \tilde{h}_t^l + b^l\right)$

因为FSMN本质是前馈神经网络，所以训练方法可以使用误差反向传播算法（error backpropagation, BP）算法来学习，同时采用基于小批量训练的mini-batch的随机梯度下降（Stoachstic gradient decent, SGD）。在语音识别任务中，如果在早期使用64到256个样本大小，而在后期换用1024到8096的样本大小，可以学习到一个更好的模型。

(2) 简洁的FSMN（compact FSMN, cFSMN）

在研究[11]中结合矩阵低秩分解（Low-rank matrix factorization）的思路，提出了改进的FSMN结构，称之为简洁的FSMN（compact FSMN, cFSMN），如图13 所示，是一个第l隐层包含记忆模块的cFSMN结构框图。 对于cFSMN，通过在网络的隐层后添加一个低维的线性投影层，并且将记忆模块添加在这些线性投影层上。cFSMN对记忆模块的编码公式也进行了调整，通过将当前时刻的输出显式地添加到记忆模块的表达中，从而只需要将记忆模块的表达作为下一层的输入。这样可以有效地减少模型的参数量，加快网络的训练。

$$\tilde{P}_t^l = P_t^l + \sum_{i=0}^{N} a_i^l \bigodot P_{t-i}^l$$

$$\tilde{P}_t^l = P_t^l + \sum_{i=0}^{N_1} a_i^l \bigodot P_{t-i}^l + \sum_{i=0}^{N_2} c_j^l \bigodot P_{t+j}^l$$

其中$P_t^l = V^l h_t^l + b^l$代表线性投影层第l层的线性输出，第l+1层的输出则为$h_t^{l+1} = f\left(U_l \tilde{P}_t^l + b^{l+1}\right)$。

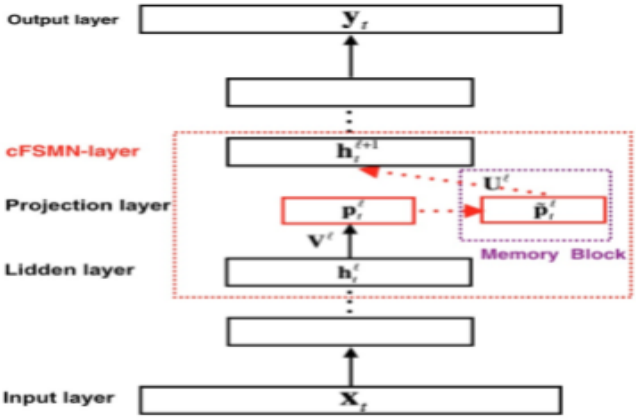

Figure 13: cFSMN结构图

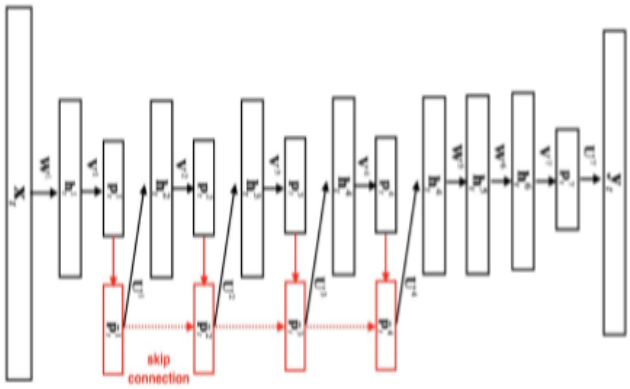

Figure 14: DFSMN结构图

其和标准FSMN的本质区别就是记忆模块和线性投影层模块用了相同的权重矩阵，而标准FSMN采用了不同的权重矩阵，从而大量减少了模型参数，减少了训练时间。 训练方法和标准FSMN一样也是采用基于小批量训练随机梯度下降的后向传播算法，且当训练准则是基于序列鉴别式训练的MMI时能取得更好的性能结果。

(3)深层的FSMN（Deep FSMN, DFSMN）

进一步地，通过在cFSMN的记忆模块之间添加跳转连接（skip connection），从而使得低层记忆模块的输出会被直接累加到高层记忆模块里。这样在训练过程中，高层记忆模块的梯度会直接赋值给低层的记忆模块，从而可以克服由于网络的深度造成的梯度消失问题，使得可以稳定地训练深层网络。 如图14所示为DFSMN[12]的结构示意图。左为输入层，右为输出层，红色方框为记忆模块。对记忆模块的表达也进行了修改，通过借鉴扩张（dilation）卷积的思路，在记忆

模块中引入一些步幅（stride）因子，具体的计算公式如下：

$$\tilde{P}_t^l = H\left(\tilde{p}_t^{l-1}\right) + p_t^l + \sum_{i=0}^{N_1^l} a_i^l \bigodot p_{t-s_{1*i}}^l + \sum_{j=1}^{N_2^l} c_j^l \bigodot p_{t+s_{2*j}}^l$$

相比于之前的cFSMN，提出的DFSMN优势在于，通过跳转连接可以训练很深的网络。对于原来的cFSMN，由于每个隐层已经通过矩阵的低秩分解拆分成了两层的结构，这样对于一个包含4层cFSMN层以及两个DNN层的网络，总共包含的层数将达到13层，从而采用更多的cFSMN层，会使得层数更多而使得训练出现梯度消失问题，导致训练的不稳定性。而提出的DFSMN通过跳转连接避免了深层网络的梯度消失问题，使得训练深层的网络变得稳定。需要说明的是，这里的跳转连接不仅可以加到相邻层之间，也可以加到不相邻层之间。跳转连接本身可以是线性变换，也可以是非线性变换。具体的实验可以实现训练包含数十层的DFSMN网络，并且相比于cFSMN可以获得显著的性能提升。

DFSMN的另外一个改进采用低帧率（Low Frame Rate, LFR）方式，输入不再是每帧语音的声学特征，而是通过将相邻时刻的语音帧进行绑定作为输入，去预测这些语音帧的目标输出得到的一个平均输出目标，这样可以极大地提升语音识别系统声学得分的计算以及解码的效率。（二）和FNN模型结构比较分析

无论是标准的FSMN，还是改进的cFSMN以及到最终的DFSMN，虽然结构略有不同，但设计思想和主题框架大致一样，区别主要在模型的深度和模型的参数设计上。它们在语音识别领域相较于传统的前馈神经网络之所以能取得性能上的巨大提升，主要原因总结为以下两点：

(1)通过添加记忆模块从而可以添加上下文信息，较好地对语音长时依赖信息进行建模。

(2)在此基础上通过参数的设计对模型进行优化和精简，从而达到较优的训练性能。

（三）训练方法总结

FSMN本质是前馈神经网络，所以训练方法使用误差反向传播算法（error backpropagation, BP）算法来学习，同时采用基于小批量训练的mini-batch的随机梯度下降（Stoachstic gradient decent, SGD）方法。

（四）模型特点分析

(1)相较于BLSTM，FSMN获得了显著的性能提升。

(2)训练速度相较于BLSTM也更快。

2.1.4 循环神经网络*(RNN)*

循环神经网络中神经元的一些连接组成了一个有向环，有向环使得在循环神经网络中出现了内部状态或带记忆的结构，赋予了循环神经网络建模动态时序的能力。在语音识别中，即可以建立一个和输入句子长度一样层数的深度模型。

（一）模型结构特点

简单的单隐层RNN的结构特征如图15所示。

其实，RNN的本质特征就是将同样的结构特征反复使用，即所谓的权重共享，从而达到输入是序列，且模型参数不会过于复杂的效果。为便于理解，表示成下图所示的带有输入输出特征的网络结构。其中$h^0, h^1, h^2...$是具有相同维度的向量，$x^1, x^2, x^3...$是具有相同维度的向量，$y^1, y^2, y^3...$是具有相同维度的向量。F的计算过程为$h^1, y^1 = f\left(h^0, x^1\right)$，即输入两个向量，输出两个向量，向量的维度有相应的对应关系。在此基础上，f的定义就可根据具体需要进行具体的设计。

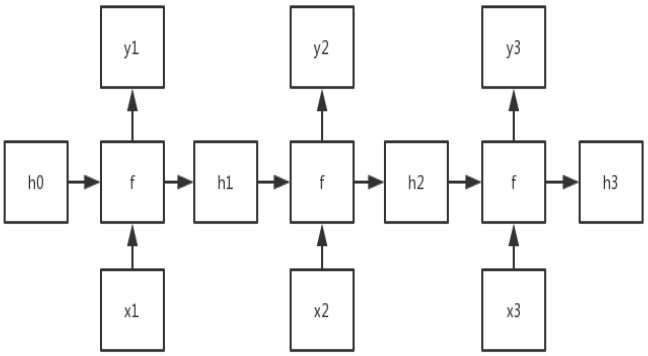

Figure 15: 单隐层RNN结构

具体的公式表达式为：

$$h_t = f\left(W_{xh}x_t + W_{hh}h_{t-1}\right)$$
$$y_t = g\left(W_{hy}h_t\right)$$

可以将RNN的核心结构特点总结成如下几点：
(a)无论输入序列的长度有多少，总是共用相同的函数f。从这个角度来说RNN相较于FNN的好处是能精简模型的参数。因为无论序列的长度如何变化，f不变，参数不会增多。虽然FNN的输入也可以是序列，但是序列增多的时候，即输入层的维度增大，相应的参数就会增多。参数一旦剧增，就会导致过拟合的现象产生。
(b)F的输入包括此刻的输入和上一时刻的输出，因此可以对语音信号的长时依赖特点进行建模。
RNN的训练方法是基础沿时反向传播（BPTT）方法。它用于学习循环神经网络随时间展开网络的权重矩阵和通过时间顺序回传错误信号。这是前馈网络的经典反向传播算法的一个扩展，其中对同一训练帧t时刻的多个堆积隐层，被替换成T个跨越时间的相同单一隐层t=1,2,⋯,T。BPTT由于帧之间的依赖关系而收敛得更慢，而且因为梯度的爆发和消失问题在音频样本层（代替了帧级别层）的随机化，更可能收敛到一个不好的局部最优解。

(1)长短时记忆单元（long short-term memory，LSTM）循环神经网络

为了解决RNN训练过程中的梯度消失和梯度爆炸的问题，一种成为“长短时记忆单元”（LSTM）的结构被引入到RNN中。这种变种成功解决了传统RNN所不能克服的基本问题。 模型结构的具体改变如图16所示（相同的颜色方框代表相同维度的向量）

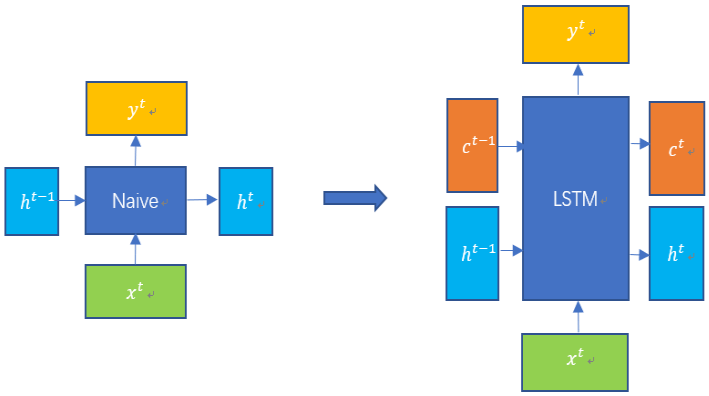

Figure 16: LSTM结构

$$
\begin{aligned}
i_t &= \sigma\left(W^{(xi)}x_t + W^{(hi)}h_{t-1} + W^{(ci)}c_{t-1} + b^{(i)}\right)\\
f_t &= \sigma\left(W^{(xf)}x_t + W^{(hf)}h_{t-1} + W^{(cf)}c_{t-1} + b^{(f)}\right)\\
c_t &= f_t \cdot c_{t-1} + i_t \cdot \left(W^{(xc)}x_t + W^{(hc)}h_{t-1} + b^{(c)}\right)\\
o_t &= \sigma\left(W^{(xo)}x_t + W^{(ho)}h_{t-1} + W^{(co)}c_{t-1} + b^{(0)}\right)\\
h_t &= o_t \cdot tanh\left(c_t\right)
\end{aligned}
$$

其中W^{ci}是对角矩阵 训练方法：异步随机梯度下降（ASGD）算法和截断的沿时反向传播（BPTT）算法

（二）训练方法总结

训练方法采用异步随机梯度下降（ASGD）算法和截断的沿时反向传播（BPTT）算法。传统RNN上做BPTT时梯度会随着两个时间的间隔增加或指数减少，所谓梯度爆炸或梯度消失，只有使用启发式规则或者一些约束优化方法才能有效地学习参数。但LSTM单元能解决这个问题的原因是当梯度从输出层被反向传播到隐层时，LSTM可以对梯度进行记忆，从而使得LSTM的训练变得有效。

（三）特点总结

循环神经网络模型有以下优缺点：

(a)由于循环神经网络对语音信号动态建模的能力以及对时间长时依赖性的表达能力，让它在性能上有较好的表现。

(b)模型通常比较庞大，训练解码时间长。

(c)由于输出的依赖性，不能很好地进行实时的语音识别。

2.2 语言模型(Language model)

在语音识别中，声学模型和语言模型配合使用来进行结果的输出。其中语言模型的作用是估测一句话出现的概率。例如两个不同的序列recognize speech和wreck a nice speech有相同的发音，判断是哪个序列就需要用到语言模型。一般通过混淆度（Perplexity，PP）来评价语言模型性能。

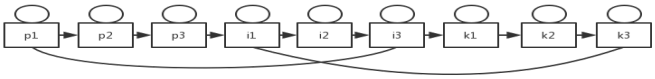

Figure 17: 语言HMM结构图

2.2.1 N-gram

如果将一个序列作为一个整体来进行数据采集的话很可能会出现数据稀疏的情况。所以最常用的做法是将句子拆分,$N-gram$方法就是用马尔可夫链的结构，原理是每个词只和前$n-1$个词有关，常用的形式是bigram和trigram。例如bigram每个词和前一个词有关，一句话出现的概率为$P=(W_1,W_2,W_3,...,W_n)=P(W_1 \mid START)P(W_2 \mid W_1)...P\left(W_n \mid W_n-1\right)$，而其中的每一小部分都通过统计文本的方法采集。 当声学模型是HMM模型时，因为都是马尔可夫链的结构，所以语言模型可以和单词的声学模型复合得到一门语言的HMM。具体步骤为训练的音素HMM按词典拼接成单词HMM，单词HMM与语言模型复合成语言HMM。如图17为单词pick的复合模型。

N-gram的优点是容易训练和使用，但是这种方法几乎很难估计准确，因为数据不可能统计完全，当N很大的时候，又容易出现数据稀疏的现象。 为解决数据集小导致的稀疏问题，提出了一些数据平滑优化方法。

(a)当某个词在数据集中依赖频率出现为0时，统计时将0改为一个较小的概率。

(b)词向量法。构造历史单词和当前单词的词向量，将历史单词和当前单词做点积，优化全局概率来训练单词向量。然后用训练好的单词向量点积值来替换稀疏值。

2.2.2 神经网络语言模型

神经网络在声学模型中的运用使得语音识别的性能有了大幅度的提升，于是人们开始尝试将神经网络运用到语言模型中。 一句话出现的概率还是由$P=(W_1,W_2,W_3,...,W_n)=P(W_1 \mid START)P(W_2 \mid W_1)...P\left(W_n \mid W_n-1\right)$表达。但是与N-gram不同的是，$P\left(W_n \mid W_n-1\right)$由神经网络训练得到。 基础的神经网络语言模型如图18所示： 而由于RNN建模长时相关信息的能力，所以使用最广泛的还是RNNLM语言模型结构。 结构图如图18所示。 RNNLM虽然性能有所提升，但是实际应用的时候因为节点数多，所以存在占空间、训练和测试的计算量都很大的问题，对训练数据也比较敏感。在较多论文里，都对RNNLM的训练优化做了相关研究。在论文[13]中采用抽样训练方法加快训练速度；在论文[14]中采用快速边界适应（fast marginal adaptation，FMA）的框架结构，将来自RNNLM的概率乘以每个单词的特定因子，并重新归一化。

3 Thchs30简介

3.1 数据集简介

如表1所示，为THCHS30数据集[15]的具体构成。

3.2 流程分析

3.2.1 数据预处理和特征提取

（一）对data/train, test生成text，wav.scp，utt2spk，spk2utt

命令行：local/thchs-30_data_prep.sh

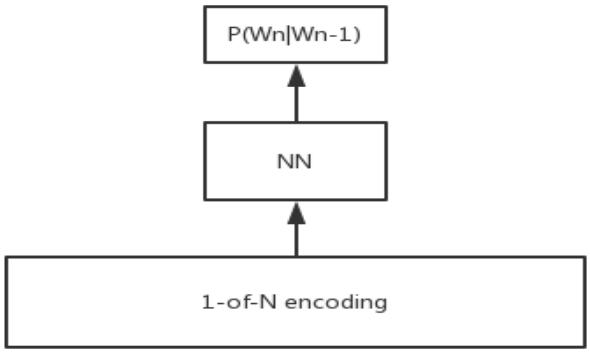

Figure 18: 神经网络语言模型

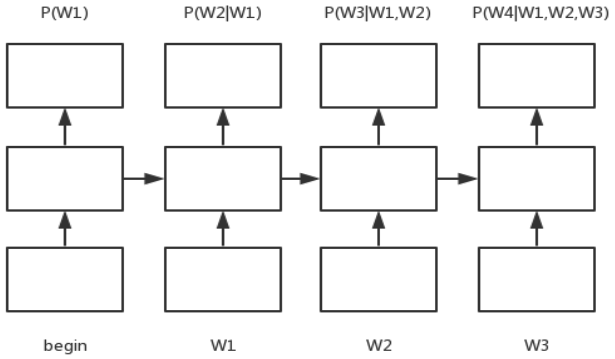

Figure 19: 神经网络语言模型

读取语料库中的 train, test 文件夹下的.wav文件和.trn文件。利用wav文件的名字和所在路径生成wav.scp文件，利用wav.trn文件生成word.txt 。utt2spk(句子到说话人)和spk2utt(说话人到句子)里的内容都是两列相同的wav 文件名。

（二）读取语音数据库词典文件内容

1. 建立data/dict文件夹

2. 将语音数据库中的相关文件

（extra_questions.txt，nonsilence_phones.txt，optional_silence.txt，silence_phones.txt）拷贝到data/dict目录下

3. 查找字典文件不包含< s >和< $/s$ >字符的行输入到data/dict目录下

Table 1

数据集	说话人数	男性	女性	年龄	句子数	时长
训练集	30	8	22	20-55	10893	27.23
测试集	10	1	9	19-50	2496	6.24

的lexicon.txt中
（三）生成训练过程所需的语言模型文件
1. 建立data/lang文件夹
2. Utils/prepare_lang.sh命令构建字典L.fst文件
3. 建立data/graph文件夹
4. 将语音数据中的语言模型word.3gram.lm解压到data/graph文件夹下
5. Utils/format_lm.sh命令生成G.fst文件并检查是否包含空环，方便和L.fst一起作用。
附：FST(Finite State Transducer)有限状态转换机
FST允许声学信息(HMM集合)、发音模型、语言模型和识别语法统一的表示。以便在遍历搜索网络期间，可以减少计算量。 L.fst和G.fst是语言模型和字典模型的另一种表现形式。
（四）特征提取
1. 对train和test提取MFCC特征储存在文件夹data/mfcc里
2. 对train和test提取fbank特征储存在文件夹data/fbank里
3. 对mfcc继续做CMVN（倒谱均值方差归一化）
4. 对fbank继续做CMVN（倒谱均值方差归一化）
附：CMVN
受不同麦克风及音频通道的影响,会导致相同音素的特征差别比较大，通过CMVN可以得到均值为0，方差为1的标准特征。

*3.2.2 GMM-HMM*模型训练
（一）单音素模型训练mono
1. steps/train_mono.sh 用来训练单音素模型，利用data/mfcc/train的训练数据和data/lang里的语言模型，主要输出为在exp/mono文件夹里的final.mdl和tree。训练的核心流程就是用EM算法迭代对齐-统计算GMM与HMM信息-更新参数，其中迭代40次。
gmm-init-mono 通过少量的数据快速得到一个初始化的HMM-GMM模型
compile-train-graphs 生成句子fst，然后生成音素级别的fst
align-equal-compiled 对训练数据进行初始均匀对齐
gmm-acc-stats-ali 对齐后对数据进行训练
2. local/thchs-30_decode.sh用来解码和测试部分，用刚刚训练得到的模型来对测试数据集进行解码并计算准确率和似然度等信息。
3. steps/align_si.sh 使用训练好的模型对数据进行强制对齐，方便继续使用。输出结果在exp/mono_ali里。
（二）三音素模型训练tri1
1. steps/train_deltas.sh就是三音素模型的训练部分，利用data/mfcc/train的训练数据和data/lang的语言模型以及exp/mono_ali里的对齐做三音素训练。该训练和单音素模型的主要区别是状态绑定部分。训练方法也是EM算法，输出模型保存在exp/tri1里。
2. local/thchs-30_decode.sh是解码测试部分，和单音素的解码测试是一样的，只是少了 - mono选项
3. steps/align_si.sh利用训练得到的三音素模型来做强制对齐。代码和单音素一样，区别是输入模型的变化，输出结果保存在exp/tri1_ali里。
（三）线性判别分析（Linear Discriminant Analysis， LDA）tri2b
1. step/train_lda_mllt.sh 利用data/mfcc/train的训练数据和data/lang的语言

Table 2

nnet1	DNN	%WER 23.66 [19195 / 81139, 392 ins, 641 del, 18162 sub]
	DNN_MPE_it1	%WER 23.42 [18999 / 81139, 381 ins, 626 del, 17992 sub]
	DNN_MPE_it2	%WER 23.40 [18984 / 81139, 389 ins, 614 del, 17981 sub]
	DNN_MPE_it3	%WER 23.31 [18914 / 81139, 381 ins, 600 del, 17933 sub]
nnet3	TDNN	%WER 23.28 [18892 / 81139, 336 ins, 776 del, 17780 sub]
	TDNN_MMI	%WER 23.11 [18751 / 81139, 366 ins, 702 del, 17683 sub]

模型以及exp/tri1_ali三音素训练里的数据对齐做LDA，获得新模型保存在exp/tri2b里。LDA用来做特征调整并训练新模型

2. local/thchs-30_decode.sh是解码测试部分,同上

3. steps/align_si.sh 训练好模型后根据模型对数据进行对齐, 输出结果保存在exp/tri2b_ali里。

（四）说话人自适应训练（Speaker Adaptive Training，SAT）tri3b

1. steps/train_sat.sh 利用data/mfcc/train的训练数据和data/lang的语言模型以及exp/tri2b_aliLDA训练里的数据进行SAT训练得到新模型。新模型储存在exp/tri3b里。说话人自适应技术是利用特定说话人数据对说话人无关(Speaker Independent, SI)的脚本进行改进，目的是得到说话人自适应(speaker Adapted, SA)的脚本来提升识别性能。

2. local/thchs-30_decode.sh 是解码测试部分，同上

3. steps/align_fmllr.sh 训练好模型后根据模型对数据进行对齐, 输出结果保存在exp/tri3b_ali里。

（五）quick训练 tri4b

1. steps/train_quick.sh 利用data/mfcc/train的训练数据和data/lang的语言模型以及exp/tri3b_ali 里的数据进行进行quick训练得到新模型。新模型储存在exp/tri4b里。

2. local/thchs-30_decode.sh 是解码测试部分，同上

3. steps/align_fmllr.sh 采用quick训练得到的模型对数据进行对齐，输出结果保存在exp/tri4b_ali里。

3.2.3 深度神经网络训练

使用的框架为nnet3的TDNN模型

（一）训练TDNN模型

1. local/nnet3/run_tdnn.sh 利用data/fbank/train滤波器组提取的特征和exp/tri4b_ali里的数据进行进行tdnn模型的训练。tdnn模型储存在exp/nnet3/tdnn里。

2. steps/nnet3/decode.sh 为解码测试部分

（二）MMI准则训练

1. local/nnet3/run_tdnn_discriminative.sh 用exp/nnet3/tdnn和data/fbank/train的训练数据做MMI准则模型训练

2. steps/nnet3/decode.sh 为解码测试部分

4 Experiment

利用thchs30数据集训练nnet1中的HMM-DNN声学模型结构和nnet3中的TDNN模型，结果如表2所示。 由训练结果可以看出，使用TDNN声学模型相较于HMM-DNN声学模型性能有所提升。

References

1. Ke Wang, Junbo Zhang, Sining Sun, Yujun Wang, Fei Xiang, and Lei Xie, "Investigating generative adversarial networks based speech dereverberation for robust speech recognition," 2018.
2. Ke Wang, Junbo Zhang, Yujun Wang, and Lei Xie, "Empirical evaluation of speaker adaptation on dnn based acoustic model," 2018.
3. Chung Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, and Katya Gonina, "State-of-the-art speech recognition with sequence-to-sequence models," 2017.
4. William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell," *Computer Science*, 2015.
5. Zhiyuan Tang, Lantian Li, Dong Wang, and Ravi Chander Vipperla, "Collaborative joint training with multi-task recurrent model for speech and speaker recognition," *IEEE/ACM Transactions on Audio Speech & Language Processing*, vol. PP, no. 99, pp. 1–1, 2017.
6. George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2012.
7. Alexander Waibel Member Ieee, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano Member Ieee, and Kevin J. Lang, "Phoneme recognition using time-delay neural networks," *Readings in Speech Recognition*, vol. 1, no. 2, pp. 393–404, 1990.
8. Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
9. Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohamadi, and Sanjeev Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," *choice*, vol. 1000, pp. 1.
10. Shiliang Zhang, Cong Liu, Hui Jiang, Si Wei, Lirong Dai, and Yu Hu, "Feedforward sequential memory networks: A new structure to learn long-term dependency," *arXiv preprint arXiv:1512.08301*, 2015.
11. Shiliang Zhang, Hui Jiang, Shifu Xiong, Si Wei, and Li-Rong Dai, "Compact feedforward sequential memory networks for large vocabulary continuous speech recognition.," in *INTERSPEECH*, 2016, pp. 3389–3393.
12. Shiliang Zhang, Ming Lei, Zhijie Yan, and Lirong Dai, "Deep-fsmn for large vocabulary continuous speech recognition," *arXiv preprint arXiv:1803.05030*, 2018.
13. Hainan Xu, Ke Li, Yiming Wang, Jian Wang, Shiyin Kang, Xie Chen, Daniel Povey, and Sanjeev Khudanpur, "Neural network language modeling with letter-based features and importance sampling," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on. IEEE*, 2018.
14. Ke Li, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Recurrent neural network language model adaptation for conversational speech recognition," *INTERSPEECH, Hyderabad*, pp. 1–5, 2018.
15. Dong Wang and Xuewei Zhang, "Thchs-30: A free chinese speech corpus," *arXiv preprint arXiv:1512.01882*, 2015.