双周报告之语音识别技术总结与实施

吴嘉瑶 @ CSLT

1.语音识别国内外前沿科技

如今，一些商业系统的语音识别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。

对于这些任务，当前最好系统的词错误率往往在 20%左右。针对这些语音识别问题，本文做了一些针对性的学术进展调研。

1.1 GAN 去混响技术

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。

1.1.1 概念

在语音交互过程中，发音者声音除了直达声到达对方的耳朵，还有各种各样的反射面产生的反射，共同叠加传到对方的耳朵中。声音是由直达声、早期反射和晚期混响构成的。声音的传输和传播，从发声声源传出来，会在房间驻留相当长的一段时间，混响对语音识别性能有严重的影响。

1.1.2 相关理论

可以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无混响干净语音输出之间的一个映射。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]中用目前非常火热的生成对抗网络(Generative Adversarial Network, GAN)解决去混响问题。生成对抗网络一般由两个网络组成：一个生成器，一个判别器。根据具体任务，通过生成器和判别器的多次迭代博弈，达到生成较好数据的目的。首先，和其他网络如 DNN 和 CNN 相比，生成器网络用 LSTM 网络效果最优，因为它本身有很强的时序建模能力，而混响和时间非常相关。如果网络层数比较深，则加入残差网络可以进一步提升效果。在网络训练过程中，用同一个 Mini-batch 的数据去更新两个网络（G 和 D）对网络的性能提升是至关重要的。

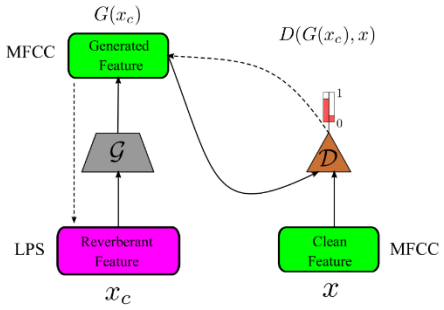


图 1.1 GAN 去混响模型

如图 1.1 所示，为 GAN 去混响的具体模型结构示意图。最终语音识别数据集结果表明，GAN 能够比单纯 DNN 去混响获得到 14-19%相对字错误率的下降。最终在多条件训练(multi-condition training，MCT)的场景下，进一步将字错误率从 16%降到 13%。

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，现有通用的语音识别声学模型，往往是通过不同口音人群数据的覆盖，来缓解这一问题。但是这终究是一个“平均模型”，不可能在每个人身上获取到最佳的语音识别性能，这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 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 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。

- 第一种方法是（linear input network, LIN），属于线性变换类别。在传统语音级别大网络声学模型前提下可以加一个线性变换网络，把不同人的语音输入变成某种通用特征，原始大网络参数不做任何变化。也就是将说话人相关的特征从$v^0 \in R^{N_0 \times 1}$通过线性变换$v_{LIN}^0 = W^{LIN}v^0 + b^{LIN}$变换到另一个与说话人无关的能与 DNN 匹配的特征向量$v_{LIN}^0 \in R^{N_0 \times 1}$。
- 第二种方法是(learning hidden unit contribution, LHUC)，为每个人学习一组个性化参数，用于调节大网络声学模型参数的幅度。
- 第三种方法是用每个人的数据去直接更新大网络声学模型参数，即一人一个网络。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，使得适应后的模型的后验概率分布与说话人无关的大网络模型上的后验分布越接近越好。

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（Sequence to Sequence Model）

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）、发音模型（pronunciation model, PM）和语言模型（language model, LM）组成，所有这些模块都会经过独立训练，同时通常是由手动设计的，各个组件会在不同的数据集上进行训练。AM 提取声学特征并预测一系列子字单元（subword unit），通常是语境依赖或语境独立的音素。然后，手动设计的词典（PM）将声学模型生成的音素序列映射到单词上。最后，LM 为单词序列分配概率。独立地训练各个组件会产生额外的复杂性，最终得到的性能低于联合训练所有的组件。

端到端模型（Sequence to sequence model）将传统的 ASR 系统中各自独立训练的声学模型、发音模型和语言模型（AM，PM，LM）合并成单个神经网络，在自动语音识别（ASR）领域中获得了越来越广泛的关注。

1.3.2 理论基础

（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。LAS 架构由三个组件组成，如图 1.2 所示。输入端是听者编码器组件（listener encoder component），和标准的声学模型（AM）相似，输入语音信号用时频表示，然后使用一系列的神经网络层将输入映射到一个高级特征表示——$\boldsymbol{h}^{enc}$。听者编码器的输出被传递到第二个组件——参与者组件（attender），其使用 $\boldsymbol{h}^{enc}$ 学习输入特征 x 和预测子字单元之间的对齐方式，其中每个子字通常是一个字素或字片（word piece）。最后，注意力模块（attention module）的输出被传递给第三个组件——speller（即解码器），speller 和 LM 相似，可以生成一系列假设词的概率分布。LAS 模型的所有组件都是被当做一个单一端到端神经网络模型进行联合训练，这一点与传统系统的分开训练不同，同时也让训练过程变得更加简单。此外，由于 LAS 模型完全采用神经网络模型，所以它不需要手动设计额外的组件（例如，有限状态转换器、词库和文本标准化模块）。最后，与传统模型不同的是，训练端到端模型不需要来自单独训练系统生成的决策树或者时间

对准的引导程序，并且可以训练给定的文本副本（Text trans）对和相应的声学对。

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，做出了更长的子字单元（即字片，word pieces）的优化，该优化能精简解码步骤，获取性能提升；此外，还优化了传递给解码器的注意力向量，如图 1.3 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。

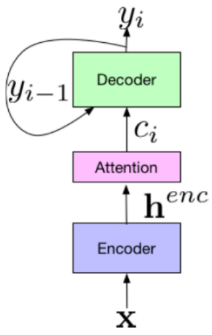


图 1.2 LAS 端到端结构

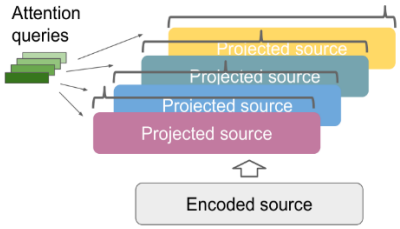


图 1.3 MHA 结构

（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 1.4 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。

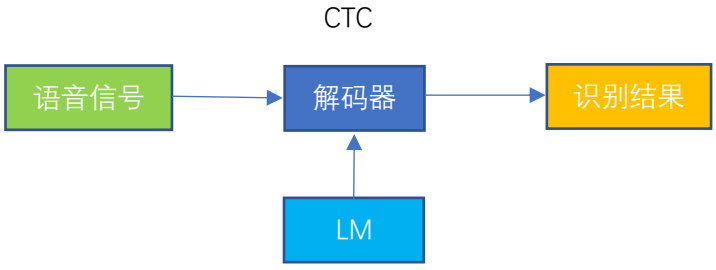


图 1.4 CTC 框架结构图

需要说明的是，目前端到端系统相对混合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。

1.4 多任务学习和迁移学习

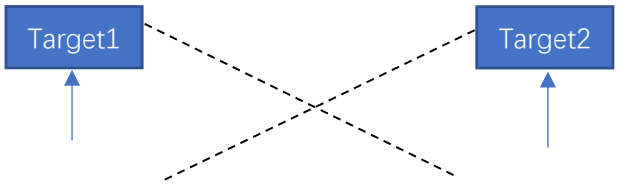
在深度神经网络（Deep neural network, DNN）中，每个隐藏层都是输入 DNN 的原始数据的一种新特征表示，较高层次的表征比较低层次的表征更为抽象，而这些特征的表示可以通过多任务（multi-task）和迁移学习（transfer learning）等技术共享和迁移到相关的任务，即通过共享隐层的 DNN 架构来实现。

1.4.1 概念

- 多任务学习（Multitask learning, MTL）是一种旨在通过联合学习多个相关的任务来提高模型泛化能力的机器学习技术。多任务学习成功的关键是任务之间应是相关的。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。如果任务相似，多任务学习通过有效地增加每个任务的训练量从而有助于在任务间迁移知识；如果任务是相关的，但并不相似，共同学习可以限制每个任务可能的函数空间，从而提高每个任务的泛化能力。多任务学习在训练数据集相比模型尺寸要小的时候最有用。
- 迁移学习通过利用从一个或多个相似的任务中学习到的知识，来快速并有效地为一个新任务开发一个有较好性能的系统。与多任务学习不同的是：多任务学习旨在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。迁移学习的实践意义是在标注数据受限的情况下学习到较好的神经网络。

1.4.2 多任务学习结构

如图 1.5 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。



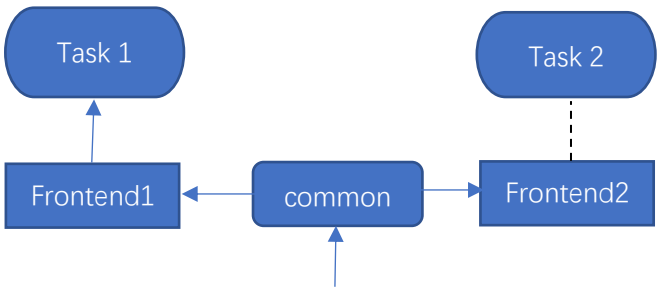


图 1.5 多任务学习结构图

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 1.6 所示为一种跨语音识别神经网络结构图。

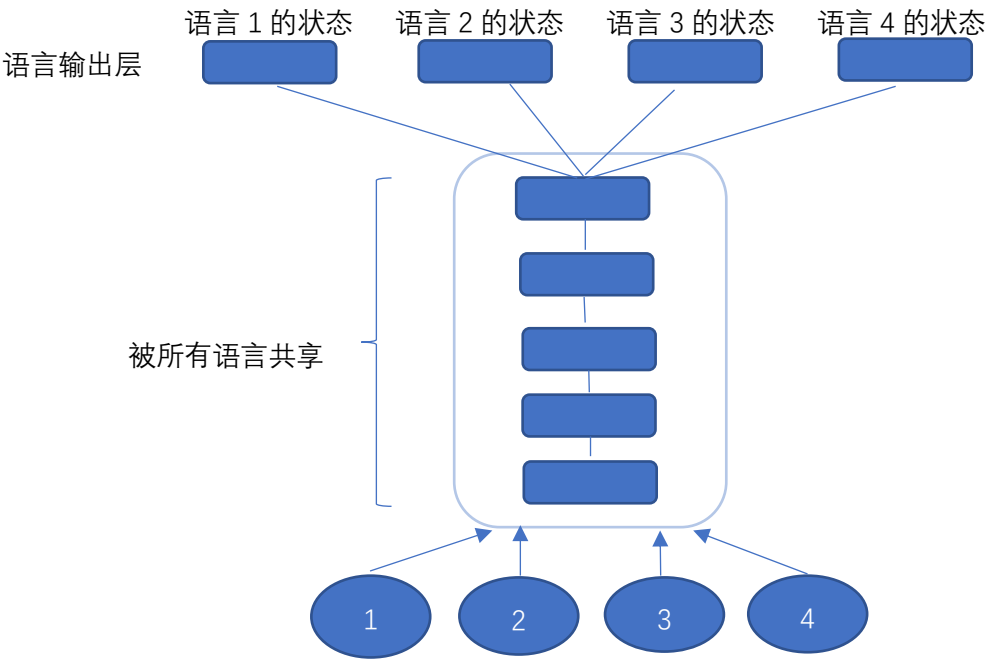


图 1.6 跨语言识别神经网络图

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.THCHS30

2.1 数据集简介

如表 2.1 所示，为 THCHS30 数据集[6]的具体构成。

表 2.1 thchs30 数据集结构

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

2.2 流程分析

2.2.1 数据预处理和特征提取

- **对 data/ {train, test} 生成 text，wav.scp，utt2spk，spk2utt**

  命令行：**local/thchs-30_data_prep.sh $thchs/data_thchs30**

  读取语料库中的{ train, test } (没有 dev)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 文件中的第 1 行和第 3 行生成 word.txt （没有 phone.txt)。由于此处没有说话人识别，因此对于 utt2spk(语段到说话人)和 spk2utt(说话人到语段)里的内容都是两列相同的 wav 文件名。

- **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. 用 cat 命令显示字典文件并用 grep 查找不包含<s>和</s>字符的行输入到 data/dict 目录下的 lexicon.txt 中

lexicon.txt	字典文件	Word phone1 phone2 … phoneN
lexiconp.txt	带概率的字典文件	Word pron-prob phone1 …phoneN
silence_phones.txt	静音音素	每一行代表一组相同的 base phone,但可能有不同的重音或者声
nonsilence_phones.txt	非静音音素	

		调的音素，eg: a a_1 a_2 a_3
optional_silence.txt	包含一个单独的音素	用来作为词典中默认的静音音素
extra_questions.txt	用来构建决策树的问题集	以是空的，包含多组相同的音素，每一组音素包含相同的重音或者声调；也有可能是一致表示非语音的静音/噪音音素。这可以用于增加自动生成问题的数量。

- **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，该文件不包含消歧符。
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。

- **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。

2.2.2 GMM-HMM 模型训练

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。

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。

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。

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。

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。

2.2.3 深度神经网络训练

使用的框架为 nnet3 的 TDNN 模型

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

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

3.声学模型和语言模型

如图 3.1 所示为目前较为常见的语音识别的基本框架流程图。

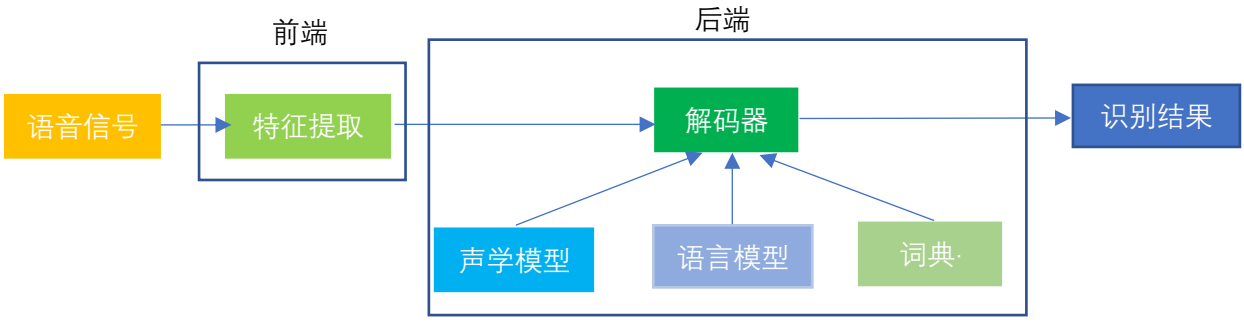


图 3.1 语音识别流程图

而语音识别的基本公式为$W^* = {}^{argmax}_{W} P(W|X) = {}^{argmax}_{W} P(X|W)P(W)$。其中$P(X|W)$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：

- 对语音信号的时序信息有较强的建模能力
- 能添加语音信号的长时依赖特性，从而提高模型性能
- 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。

3.1 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 neural 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。

- 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。

3.1.1 深度神经网络-隐马尔科夫模型混合系统（DNN-HMM）

（一）模型结构

论文[7]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 3.2 所示的 DNN-HMM 混合系统可以在实际问题中广泛应用。

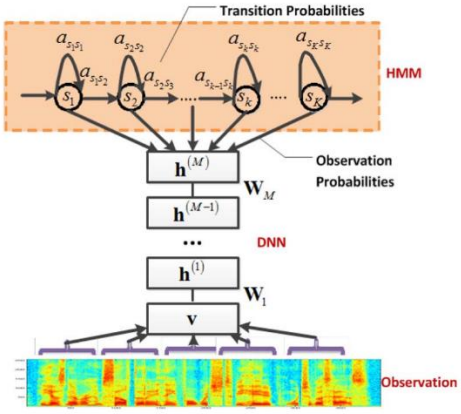


图 3.2 DNN-HMM 结构图

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$\mathrm{p}(q_t = s|X_t)$。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$X_t = [O_{\max(0,t-\varpi)} \dots O_t \dots O_{\min(T,t+\varpi)}]$，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:

- 充分利用了 DNN 的内在鉴别属性
- 训练过程可以使用维特比算法，解码通常非常高效
- 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:

- 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。
- 性能提升还未接近人类水平。

3.1.2 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，具体结构特点如图 3.3 所示。为方便观察将其等效为图 3.4，左图为一个滤波器结构，随着时间的推进对相邻帧的信号做特征提取。

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[8]。

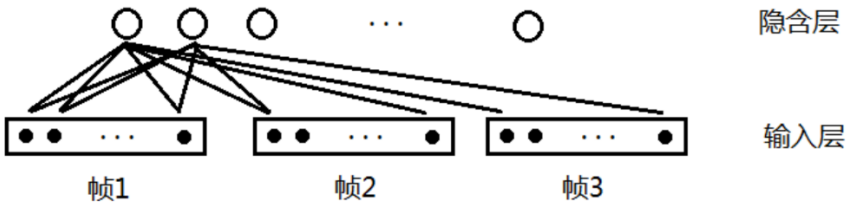


图 3.3 TDNN 结构图

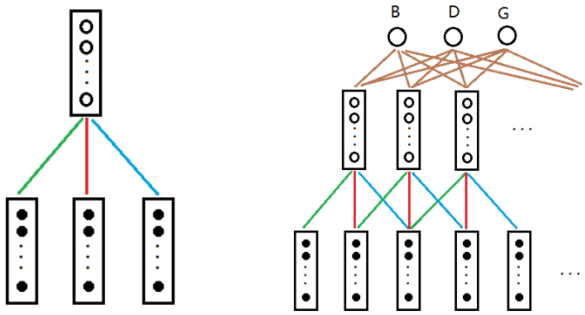


图 3.4 TDNN 结构化简图

标准的 TDNN（Time delay neural network ， TDNN）

如图 3.5 所示为语音包 kaldi 中使用的标准 TDNN 结构[9]，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。

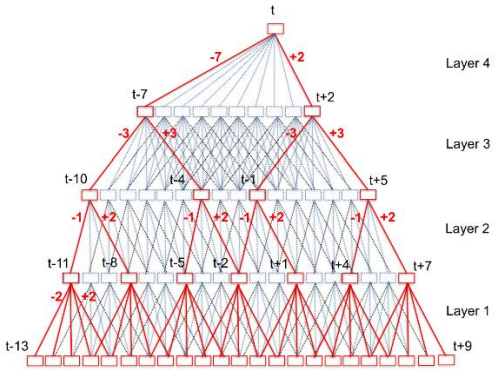


图 3.5 kaldi 中 TDNN 结构图

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 的模型结构类比分析

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

标准的 TDNN（Time delay neural network ， TDNN）

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。

分解的 TDNN（factorized TDNN, TDNN-F）

在研究[10]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 3.6 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。

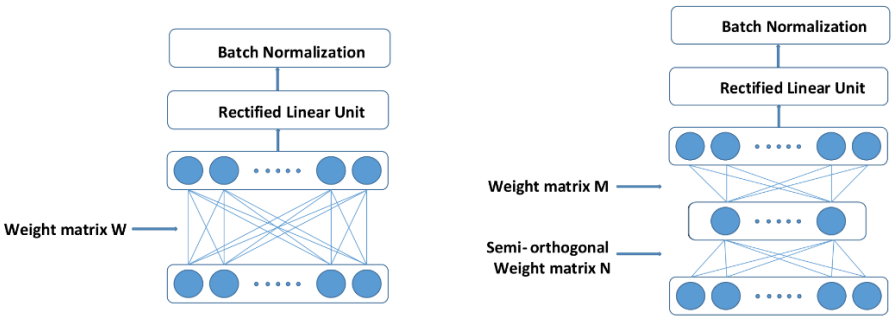


图 3.6 TDNN-F 结构图

（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。
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。

3.1.3 前馈序列记忆神经网络（Feedforward sequential memory networks，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[11]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 3.7 所示：

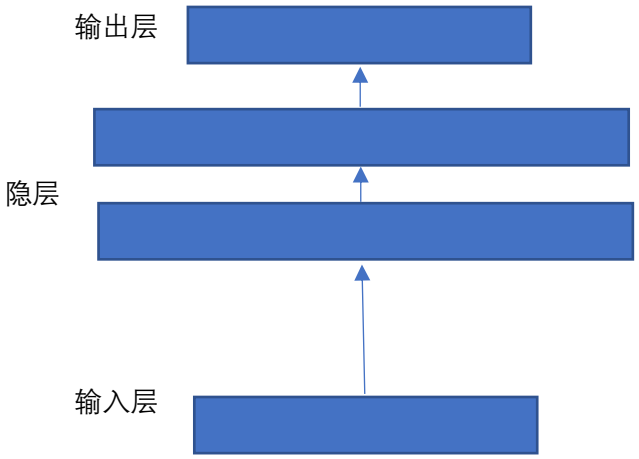


图 3.7 FNN 结构图

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 3.8 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。

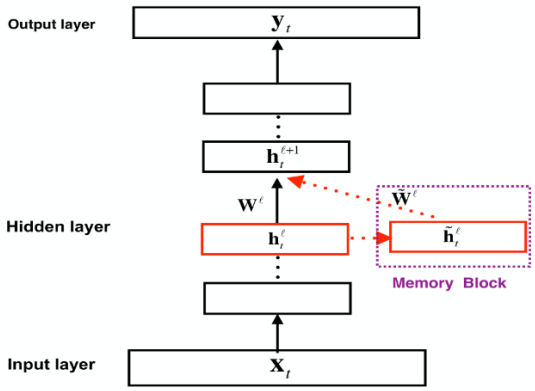


图 3.8 FSMN 结构图

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

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

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

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

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）。小批量训练从训练样本中抽出一小组数据并基于此估计梯度。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。

简洁的 FSMN（compact FSMN, cFSMN）

在研究[12]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 3.9 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。

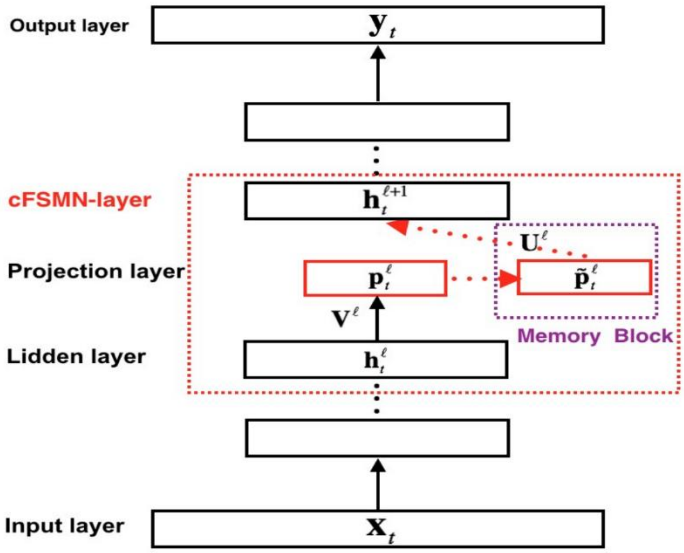


图 3.9 cFSMN 结构图

具体单向和双向的 cFSMN 记忆模块的公式表达分别如下：

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

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

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$h_t^{l+1} = f(U^l \tilde{P}_t^l + b^{l+1})$。

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。

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。

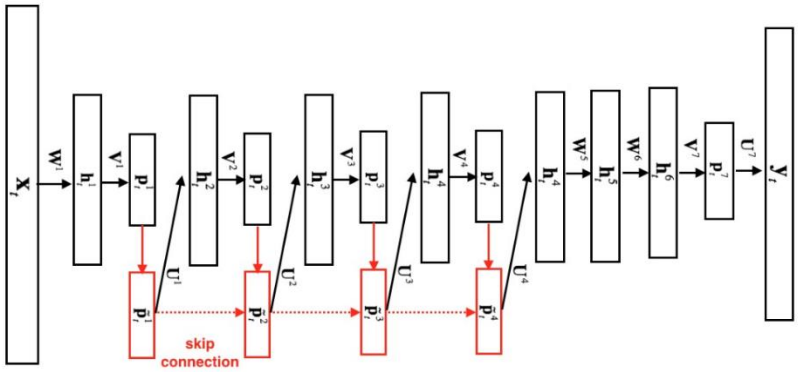


图 3.10 DFSMN 结构图

如图 3.10 所示为 DFSMN[13]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：

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

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）相对 FNN 的模型结构改进

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）特点总结

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。

3.1.4 循环神经网络（recurrent neural network, 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 3.11 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$h^1, y^1 = f(h^0, x^1)$，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。

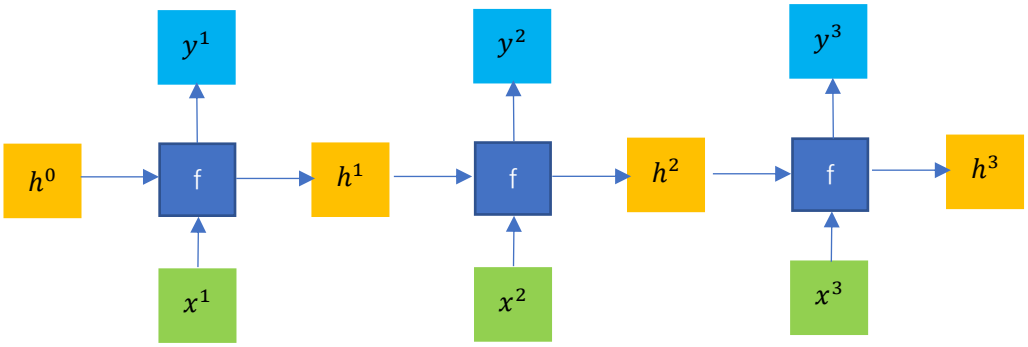


图 3.11 单隐层 RNN 结构图

具体的公式表达式为：

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1})$$

$$y_t = g(W_{hy}h_t)$$

可以将 RNN 的核心结构特点总结成如下几点：

1) 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。
2) 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$t = 1,2,\ldots,T$。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。

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下图所示（相同的颜色方框代表相同维度的向量）：

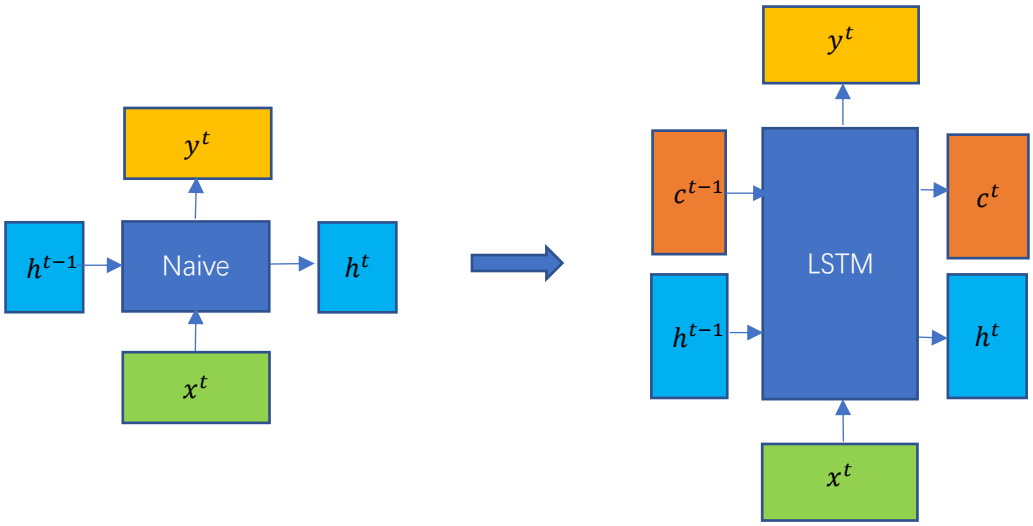


图 3.12 LSTM 结构图

LSTM 表达公式可以由以下的关于 t=1，2，…，T 的递推公式描述：

$i_t = \sigma\left(W^{(xi)}x_t + W^{(hi)}h_{t-1} + W^{(ci)}c_{t-1} + b^{(i)}\right)$——输入门

$f_t = \sigma\left(W^{(xf)}x_t + W^{(hf)}h_{t-1} + W^{(cf)}c_{t-1} + b^{(f)}\right)$——遗忘门

$c_t = f_t \cdot c_{t-1} + i_t \cdot \left(W^{(xc)}x_t + W^{(hc)}h_{t-1} + b^{(c)}\right)$——细胞状态

$o_t = \sigma\left(W^{(xo)}x_t + W^{(ho)}h_{t-1} + W^{(co)}c_{t-1} + b^{(o)}\right)$——输出门

$h_t = o_t \cdot tanh(c_t)$——隐藏层

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:

- 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。
- 模型通常比较庞大，训练解码时间长。
- 由于输出的依赖性，不能很好地进行实时的语音识别。

3.1.5 Kaldi 中 thchs30 声学模型

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，结果如下表所示。

nnet1	DNN	%WER **23.66** [19195 / 81139, 392 ins, 641 del, 18162 sub] exp/tri4b_dnn/decode_test_word/wer_8_0.0
	DNN_MPE_it1	%WER **23.42** [18999 / 81139, 381 ins, 626 del, 17992 sub] exp/tri4b_dnn_mpe/decode_test_word_it1/wer_8_0.0
	DNN_MPE_it2	%WER **23.40** [18984 / 81139, 389 ins, 614 del, 17981 sub] exp/tri4b_dnn_mpe/decode_test_word_it2/wer_8_0.0
	DNN_MPE_it3	%WER **23.31** [18914 / 81139, 381 ins, 600 del, 17933 sub] exp/tri4b_dnn_mpe/decode_test_word_it3/wer_8_0.0
nnet3	TDNN	%WER **23.28** [18892 / 81139, 336 ins, 776 del, 17780 sub] exp/nnet3/tdnn/decode_test_word/wer_8_0.0
	TDNN_MMI	%WER **23.11** [18751 / 81139, 366 ins, 702 del, 17683 sub] exp/nnet3/tdnn_mmi/decode_test_word/wer_8_0.0

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。

3.2 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。

3.2.1 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$\mathrm{P} = (W_1, W_2, W_3 \ldots W_n) = \mathrm{P}(W_1|START)\mathrm{P}(W_2|W_1) \ldots \mathrm{P}(W_n|W_{n-1})$，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。例如单词 pick 的复合模型为：

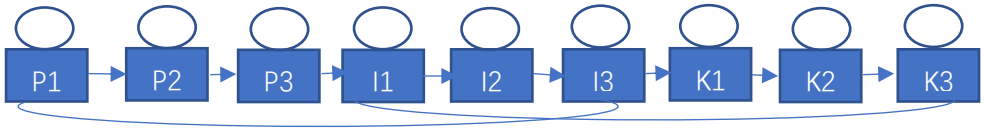


图 3.13 语言 HMM 结构图

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。

- 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。
- 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。

3.2.2 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 $\mathrm{P}=(W_1,W_2,W_3\ldots W_n)=\mathrm{P}(W_1|START)\mathrm{P}(W_2|W_1)\ldots\mathrm{P}(W_n|W_{n-1})$表达。但是与 N-gram 不同的是，$\mathrm{P}(W_n|W_{n-1})$由神经网络训练得到。

基础的神经网络语言模型如图 3.14 所示：

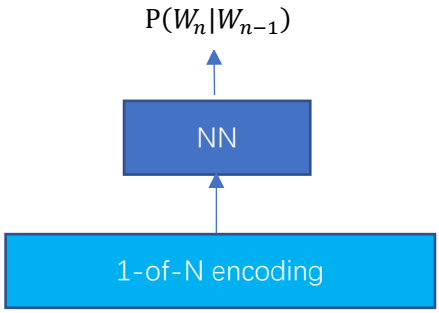
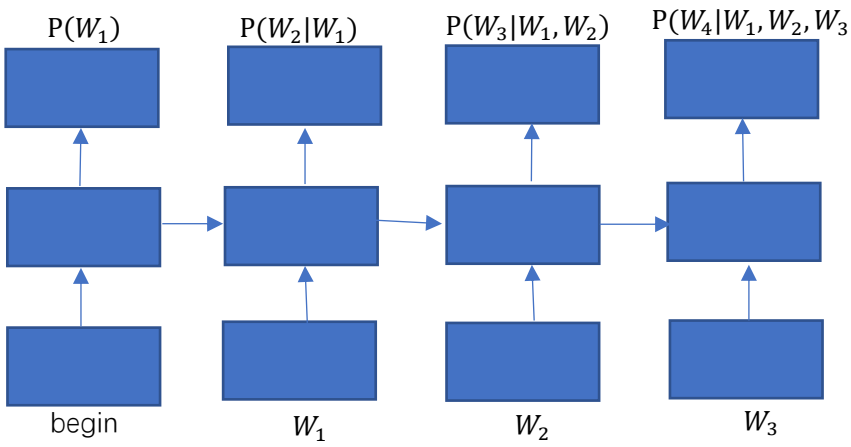


图 3.14 神经网络语言模型

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 3.15 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[14]中采用抽样训练方法加快训练速度；在论文[15]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。


$P(W_1)$
$P(W_2|W_1)$
$P(W_3|W_1,W_2)$
$P(W_4|W_1,W_2,W_3$
begin
W_1
W_2
W_3

4.改进应用

4.1 研究目的

(一)利用 GAN 做降噪

在 nnet1 中，baseline 采用 DAE 处理带噪声的语音，识别结果有所提升（如下表所示），但仍然不是很理想。在研究[1]中已有相关实验表明利用 GAN 做去混响能取得比较好的效果，去混响任务和降噪任务在一定程度上具有相似性，所以相似地，可以尝试利用 thchs30 的噪音数据实验 GAN 的降噪能力。

cafe	none	%WER **88.74** [72003 / 81139, 564 ins, 53609 del, 17830 sub] exp/tri4b_dnn_mpe/decode_word_0db/cafe/wer_8_0.0
	DAE	%WER **52.09** [42264 / 81139, 1063 ins, 3436 del, 37765 sub] exp/tri4b_dnn_dae/decode_word_0db/cafe/wer_11_0.0
car	none	%WER **27.46** [22280 / 81139, 521 ins, 704 del, 21055 sub] exp/tri4b_dnn_mpe/decode_word_0db/car/wer_9_0.0
	DAE	%WER **24.88** [20184 / 81139, 475 ins, 636 del, 19073 sub] exp/tri4b_dnn_dae/decode_word_0db/car/wer_9_0.0
white	none	%WER **98.21** [79685 / 81139, 28 ins, 55051 del, 24606 sub] exp/tri4b_dnn_mpe/decode_word_0db/white/wer_4_0.0
	DAE	%WER **66.34** [53828 / 81139, 833 ins, 5081 del, 47914 sub] exp/tri4b_dnn_dae/decode_word_0db/white/wer_15_0.0

（二）在已有的数据集上扩充得到一个可以训练模型的数据集

在进行语音识别研究或者深度学习研究的时候我发现，要想使模型的表达能力比较强，首先是要想出一个能够很好进行鉴别工作的模型，而第二步则是模型的训练即模型参数的确定。想要模型参数训练的比较好，就需要一个很大的数据集。而在现实生活中，获得较大的数据集需要庞大的人力物力资源，可否有一种方法能够在已有数据集的基础上再生成数据集，然后去训练神经网络呢？

4.2 研究思路

（一）GAN 降噪

目前生成对抗网络 GAN 在图像生成上已经取得了显著的成果。根据 GAN 的学习原理，可以采用如图 4.1 结构的 GAN，利用 GAN 学习一个从有噪语音到无噪语音的映

射，从而达到降噪的目的。

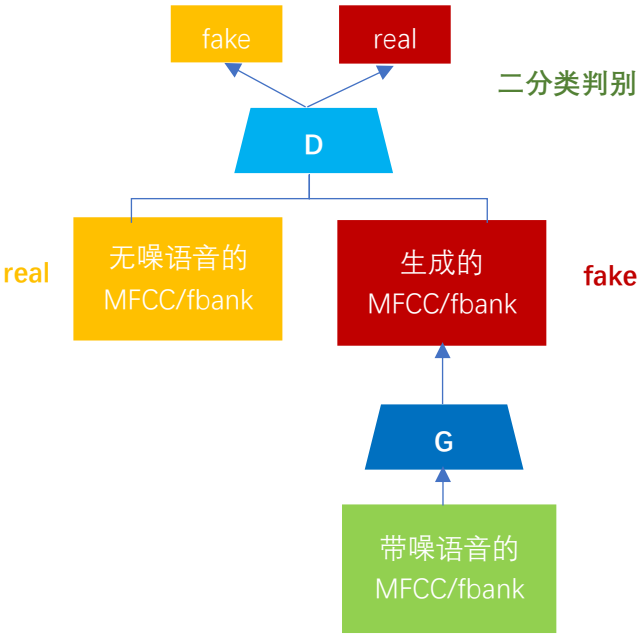


图 4.1 GAN 降噪结构图

即将带噪的语音提取特征，输入给生成器 G 得到处理的语音特征，将无噪语音的特征和生成的语音特征输入给判别器 D，让 D 分辨不出它们的区别，从而达到降噪的目的。

（二）扩充语音数据集

如图 4.2 所示为生成新语音的流程图，过程类似，相比于语音降噪的过程有输入输出的变化。

但是想要达到扩充语音数据集，即语音合成的目的，相对语音降噪来说复杂很多。因为要考虑语音输入给判别器和生成器合理的数据形式，如果仅是输入语音的特征向量，可能产生语音语义上的出入。 在论文[16][17]中有关于用 GAN 做语音合成的研究。

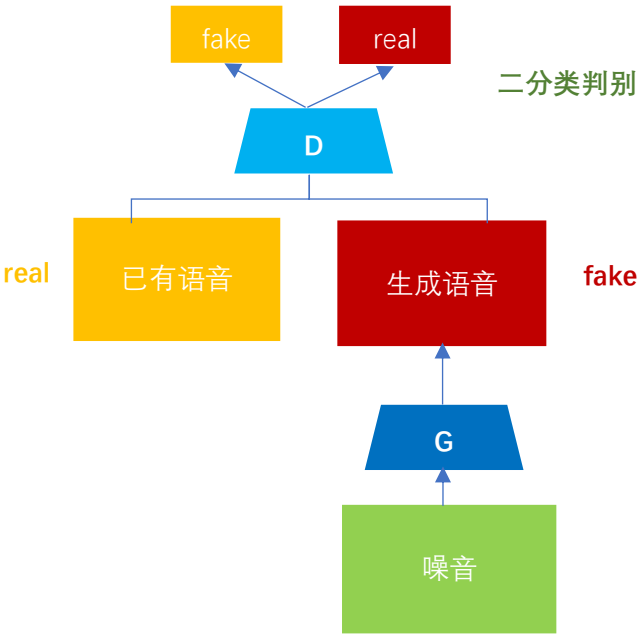


图 4.2 GAN 语音合成流程图

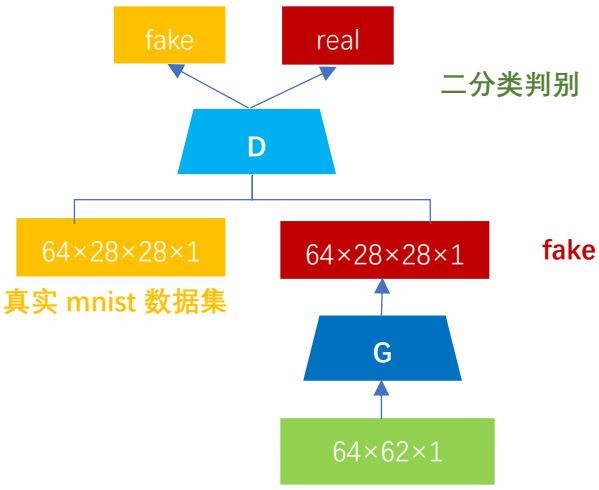
4.3 研究过程

1. 通过 mnist 数据集来学习基础 GAN 的原理和训练过程

   - 用于处理图像数据集的 GAN 网络

     GAN 网络由生成器（G）和判别器（D）组成。G 的输入是噪音，输出是图像。D 的输入是真实图像和生成图像，然后 D 做二分类，判断输入图像是否是真实图像。在优化 D 的同时也优化 G，让 G 能生成使 D 无法判别真伪的图像，从而达到生成效果较好的图像的目的。

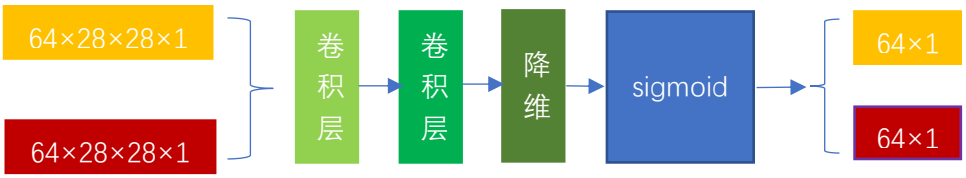
     所使用的 GAN 模型具体结构和图像输入输出的流程如下图所示：



➢ 实现过程（Tensorflow）

1. 搭建鉴别器 D 网络模型

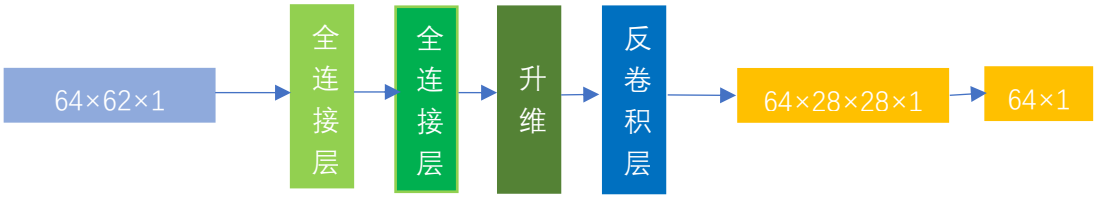
鉴别器的作用是让网络能够判别出真实的图片和合成的图片，即处理的是分类问题。具体的输入和输出以及网络结构如下图所示：



根据鉴别器的鉴别原理，若用 1 代表真实图片，0 代表非真图片，训练时鉴别器的损失函数应该让真实图片和 1，生成图片和 0 的交叉熵和越小越好。

2. 搭建生成器 G 网络模型

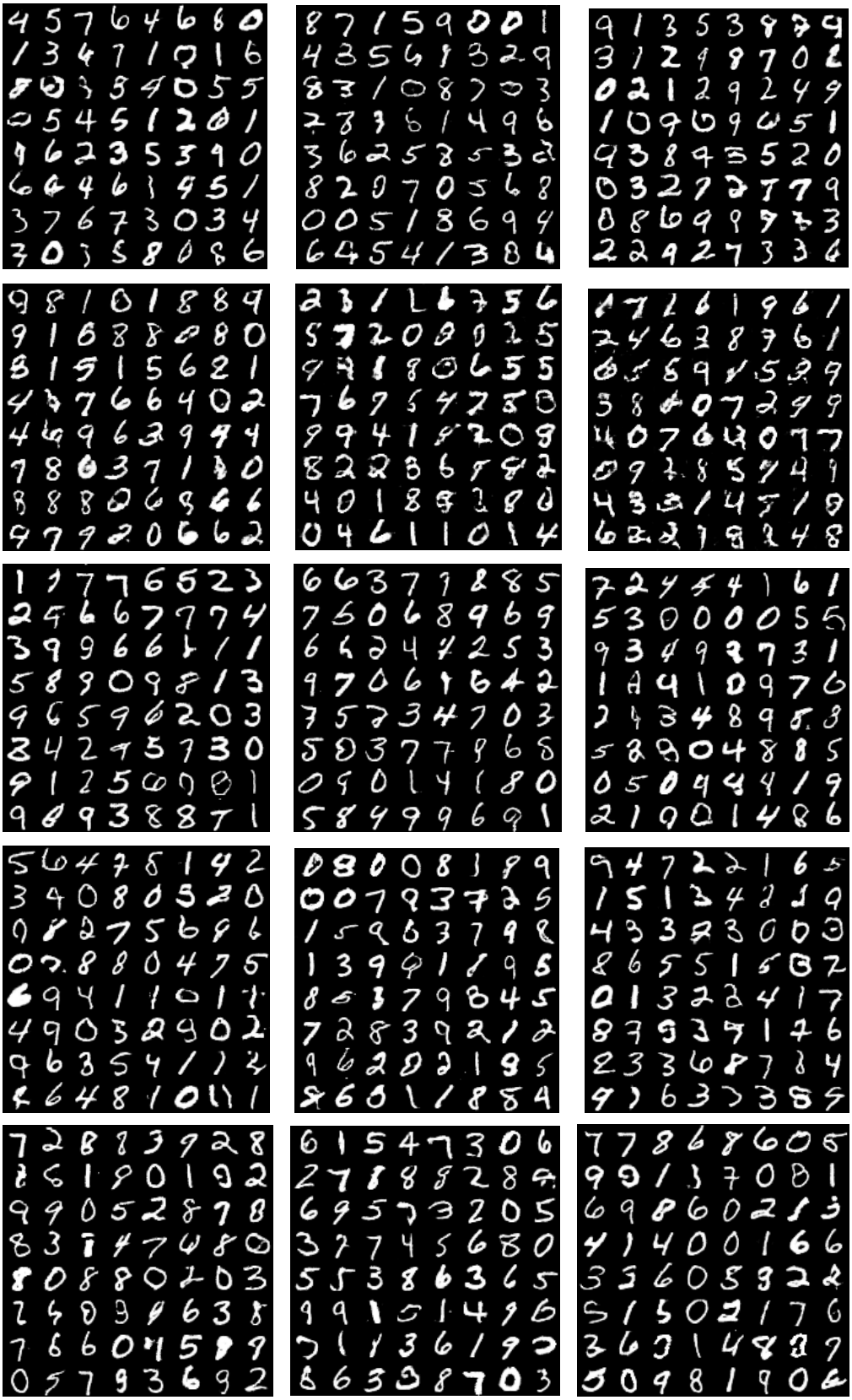
生成器的作用是输入噪音，生成图片，并让图片能“以假乱真”，具体的输入和输出以及网络结构如下图所示：

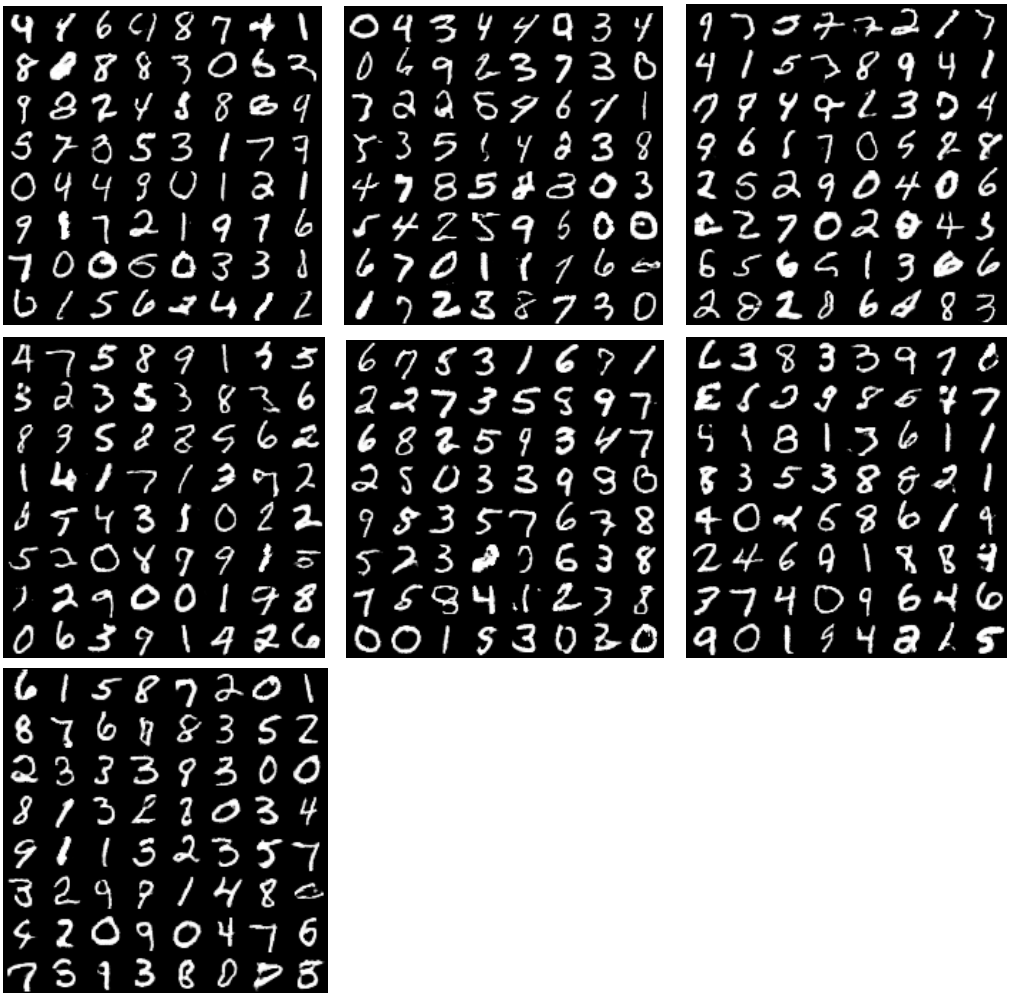


根据生成器的原理，应让生成图片和 1 的交叉熵越小越好。

➢ 实验结果

如图所示，为迭代 25 次，每次迭代后生成的图像数据集。可以看出，随着迭代次数的增加，生成的图片数据效果越来越好。





2. 将基础 GAN 运用在语音数据集上

   其实语音信号处理和图像处理有很多相似性和相关性，就比如在图像处理中取得较大性能提升的 CNN 神经网络就和运用于语音信号处理的 TDNN 神经网络有异曲同工之妙。但是图片数据和语音数据形式不同，导致存储方式和存储空间有所不同，这就需要在神经网络的结构设计上做不同的处理。

4.4 实施步骤

具体实现步骤如下：

- 生成噪音数据和无噪音数据的特征

  利用 kaldi 生成噪音数据和无噪音数据的 fbank 特征，并利用 copy-feats 操作将 ark 文件变成 txt 文件，便于读取和操作。

- 搭建生成器网络

根据研究[1]中的相关工作表明，FNN、CNN、LSTM 都可用来学习混响特征到无混响特征的映射，其中 LSTM 的效果最好。类似地，预期假设在降噪任务中，LSTM 也有可能取得最佳效果。在实验过程中，可以利用 Tensorflow 搭建不同的网络模型，比较效果。

- 搭建鉴别器网络

  鉴别器的工作原理和图像生成 GAN 的原理一样，只是输入数据维度的差别，也可以搭建不同的网络进行比较分析。

- 训练 GAN

  将带噪音的语音特征和不带噪音的语音特征一一对应输入给 GAN, 训练 GAN 模型。

- 验证 GAN 性能

  将生成器 G 生成的去噪特征输入给声学模型，声学模型可以基于 kaldi 中的声学模型获得。

参考文献

[1] Ke Wang, Junbo Zhang, Sining Sun, Yujun Wang, Fei Xiang, Lei Xie. Investigating Generative Adversarial Networks based Speech Dereverberation for Robust Speech Recognition. arXiv:1803.10132v2 [cs.SD] 17 Jun 2018

[2] Ke Wang, Junbo Zhang , Yujun Wang , Lei Xie. Empirical Evaluation of Speaker Adaptation on DNN based Acoustic Model . arXiv:1803.10146v2 [cs.SD] 17 Jun 2018

[3] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu. et al. State-of-the-art Speech Recognition with Sequence-to-Sequence Model. arXiv:1712.01769v6 [cs.CL] 23 Feb 2018

[4] William Chan. et al. Listen, Attend and Spell. arXiv:1508.01211v2 [cs.CL] 20 Aug 2015

[5] Zhiyuan Tang, Lantian Li, Dong Wang. Collaborative Joint Training With Multitask Recurrent Model for Speech and Speaker Recognition. IEEE/ACM Transaction on Audio, Speech, and Language Processing, VOL. 25, NO. 3, March 2017

[6] Dong Wang, Xuewei Zhang. THCHS-30 : A Free Chinese Speech Corpus. arXiv:1512.01882v2 [cs.CL] 10 Dec 2015

[7] George E. Dahl, Dong Yu. et al. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Transaction on Audio, Speech, and Language Processing, VOL. 20, NO. 1, January 2012

[8] Alexander Waibel. et al. Phoneme Recognition Using Time-delay Neural Network. IEEE Transaction on Acoustic, Speech, and Signal Processing, VOL. 37, NO. 3, March 1989

[9] Vijayaditya Peddinti , Daniel Povey , Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. INTERSPEECH 2015

[10] Daniel Povey , Gaofeng Cheng , Yiming Wang , Ke Li ,Hainan Xu , Mahsa Yarmohamadi , Sanjeev Khudanpur. Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. INTERSPEECH 2018

[11] S. Zhang, C. Liu, H. Jiang, S. Wei, L. Dai, and Y. Hu, “Feedforward sequential memory networks: A new structure to learn long-term dependency,” arXiv preprint arXiv:1512.08301, 2015.

[12] Shiliang Zhang , Hui Jiang , Shifu Xiong , Si Wei , Lirong Dai. Compact Feedforward Sequential Memory Networks for Large Vocabulary Continuous Speech Recognition. INTERSPEECH 2016

[13] Shiliang Zhang , Ming Lei , Zhijie Yan , Lirong Dai. Deep-FSMN For Large Vocabulary Continuous Speech Recognition. arXiv:1803.05030v1 [cs.NE] 4 Mar 2018

[14] Hainan Xu , Ke Li , Yiming Wang. et al. Neural Network Language Modeling with Letter-based Features and Importance Sampling. INTERSPEECH 2018

[15] Ke Li , Hainan Xu , Yiming Wang. et al. Recurrent Neural Network Language Model Adaptation for Conversational Speech Recognition. INTERSPEECH 2018

[16] Yuki Saito , Shinnosuke Takamichi , Hiroshi Saruwatari. Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing, VOL. 26, NO. 1, January 2018

[17] Shan Yang, Lei Xie, Xiao Chen. et al. Statistical Parametric Speech Synthesis Using Generative Adversarial Networks Under a Multi-task Learning Framework. arXiv:1707.01670v2 [cs.SD] 11 Jul 2017

相关工作

- 2018.7.24—— 列出工作大纲，进行时间规划
- 2018.7.25—— 阅读文献，总结声学模型
- 2018.7.26—— 阅读文献，总结语言模型
- 2018.7.27~2018.7.28—— 阅读文献，总结语音识别前沿科技
- 2018.7.29~2018.7.30—— 运行 thchs30，了解训练流程
- 2018.7.31—— 一周工作总结归纳，寻找创新点
- 2018.8.1——2018.8.2 利用基础的 GAN 结构跑 minist 数据集，了解 GAN 的模型结构和训练过程
- 2018.8.3——2018.8.6 尝试用 GAN 做语音降噪，完成学习总结