环境鲁棒性

2019 年 1 月 2 日

1 环境鲁棒性简介

我们常有这样的体验，本来好好的语音输入法，在办公室里基本不会有什么错误，但在大街上使用时就会感到性能明显下降；挺好的语音助手，平常百试不爽，但在公交车上问个问题经常答非所问。这主要是因为实际应用场景中的声学环境非常复杂，这些复杂场景不可能在训练时被全部覆盖，因此形成识别场景与模型的不匹配，导致系统性能的急剧下降。

总体而言，声学场景的复杂性主要可归结为三类：

1. **背景噪音**。实际应用场景中可能包含各种不同类型的噪音，如机器声、汽车引擎声、开门声、背景音乐声、其它人的谈话声等。这些噪声的混入会使语音信号发生显著变化，引起识别性能的下降。

2. **混响和回声**。在一个房间里，发音会在房间四壁反射，形成混响。房间越大，反射回来的声音相对原始声音延迟越久，产生的混响效果越明显。同样的混响效果也发生在电话通信中，在这一场景下，虽然没有象墙壁那样的直接反射，但一方的语音有可能通过对方的听筒和麦克风反射回说话者，形成混响。这种混响一般称为回声。混响和回声会显著降低声音的清晰度，严重影响识别性能。

3. **信道差异**。不同麦克风的物理特性可能有显著差别。如电容式麦克通过电容的充放电产生的电压变化代表声音，而动圈式麦克通过线圈在磁场中运动时产生的电流来代表声音信号。因此，对同一个声音信号，不同麦克风录制的声音会有显著区别。即使同一种麦克风，因信号处理方式的不同，也会得到不同的声音采样。这些差异包括增益设置、静音门限、频域补偿、编解码方式、压缩算法等。我们将上述录音设备、传输媒介等因素的影响称为信道差异。信道差异极大增加了语音识别系统的复杂性，当训练和识别的信道差异较大时，识别系统的性能将明显下降。

一个语音识别系统如果可以对抗实际应用场景的复杂性，在复杂场景下依然可以得到较好的识别性能，我们称之为为一个**环境鲁棒的识别系统**。为了提高识别系统的鲁棒性，研究者提出了各种方法，这些方法大体上可以分为两类：**前端信号处理**和**后端模型增强**。前端信号处理方法通过各种信号处理算法减小噪音、回声和信道对语音信号的影响，使之接近正常安静的语音；后端模型增强方法通过对模型做适当调整，使之更加适应实际场景的声学特性。一般来说，前端处理方法计算量小，灵活方便，但性能提升有限；后端模型增强方法计算量较大，需要的数据较多，但性能更好。下面我们将对这两种方法做简单介绍。

2 前端信号处理方法

前端信号处理方法通过对语音信号进行一系列变换，以去除信号中各种噪声和失真，恢复原始清晰语音。不同处理算法基于不同假设，产生的效果也不尽相同。总体来说，我们可以将环境影响分为加性噪声和卷积噪声两类，其中背景噪声可以认为是加性噪声，是在原有声音信号上叠加另一种信号，从而产生破坏；混响、回声和信道差异可以认为是一种卷积噪声，是在原有声音信号上的一种附加变换。我们将介绍对不同类噪声的不同处理算法。

2.1 语音增强方法

语音增强是一种时域或频谱域上的信号处理方法。历史上，语音增强的目的是为了提高语音相对人耳的可懂度，而不是语音识别性能的提高。仅管如此，在很多情况下，这种方法对语音识别依然有所帮助。

2.1.1 谱减法与加性噪声去除

谱减法（Spectral Substraction，SS）是一种常用的语音增强方法[1]。这一方法假设带噪语音的能量谱是由原始语音的能量谱和噪音的能量谱简单相加得到的，因此，如果可以估计出噪音的能量谱，即可将该能量谱从带噪语音中减去，从而恢复原始干净语音的能量谱。写成公式为：

$$|\hat{X}(f)|^2 = |Y(f)|^2 - |\hat{N}(f)|^2,$$

其中$Y(f)$和$\hat{N}(f)$分别为带噪语音和噪声的频谱，$\hat{X}(f)$为利用谱减法估计出的原始语音频谱。事实上，上述能量叠加假设忽略了原始语音与噪音之间的相关性，因此只能是一个近似估计。谱减法需要估计噪音信号的频谱，这可以通过

确定一些非语音帧（如句子的开始片段和结束片段），并对这些帧的能量谱进行平均。然而，基于这一平均能量得到的噪音估计未必能保证每一帧信号做谱减后都是正数，因此需要在应用时做适当调整。这些调整有可能会导致相邻频谱间变化过于剧烈，从而引入音乐噪音，需要做进一步平滑处理[2]。

2.1.2 回声消除

谱减法处理的是加性噪声。对于回声和混响这种卷积噪声，直接应用谱减法并不合适。为了减小回声和混响的影响，可能估计原始语音到接收端的传递函数，这一函数可用房间脉冲响应（RIR）来表示。基于这一脉冲响应，可以设计一个逆滤波器，使之与RIR的作用互相抵消，从而减弱回声和混响的影响[3, 4]。然而，估计RIR本身就是很困难的问题，而不精确的RIR会严重影响去回声的效果，还可能引入新的畸变。一些研究者发现带混响的语音在做LPC估计时，其残差往往具有更显著的高斯性。基于这一发现，可以直接设计逆滤波器，使生成语音的LPC残差更加非高斯化[5]。这种方法不需估计房间的RIR，防止了错误累积。另外一些研究者关注对高延迟混响的去除，这些混响对语音识别的影响最大。例如在大礼堂中，混响会持续很久，这些持久混响使得频谱结构发生显著改变。研究者设计了一种基于混响时间来估计RIR的方法。一般常用T_{60}作为混响时间。所谓T_{60}，是语音信号衰减60dB所需要的时间。基于T_{60}设计一个RIR模型，从而可以估计出延迟较大的混响，最后利用谱减法将这些混响的能量从带噪语音中去除。另一些研究者利用线性预测模型从历史观察信号中恢复出当前原始信号（注意当前观察信号是由历史信号经过延迟衰减并与当前原始信号叠加而成），从而将逆滤波器的设计问题转化为线性预测模型的参数估计问题[6]。上述这些方法都利用了混响和回声产生的物理机理，因而针对性较强。

2.1.3 麦克风阵列

前面所述的各种去噪和去回声方法都是基于单一麦克风，能利用的信息有限，面对复杂场景时很难得到较好的归一化效果。近年来，多麦克风设备开始普及（例如在几乎所有手机上，都装有两个以上的麦克风），使得我们得以利用语音信号传递过程中的更多空间和时间信息，从而极大提高了语音增强能力。最简单的方法是利用一个远端麦克风录制背景噪音，一个近端麦克风录制说话者语音，通过简单谱减法实现语音增强。更通用的解决方案是麦克风阵列技术[6, 7, 8]。

麦克风阵列（Microphone Array）是按一定几何结构组合在一起的一组麦克风。最常用的阵列包括线性阵列和环形阵列，如图1所示。一般来说，阵列中

的麦克风都是全指向的，即对各个方向的敏感度是一致的，但当这些全指向麦克风组合在一起的时候，就产生了强烈的指向性，从而实现方向选择、去噪、去混响等强大的功能。我们以一个线性阵列为例来对此进行说明。

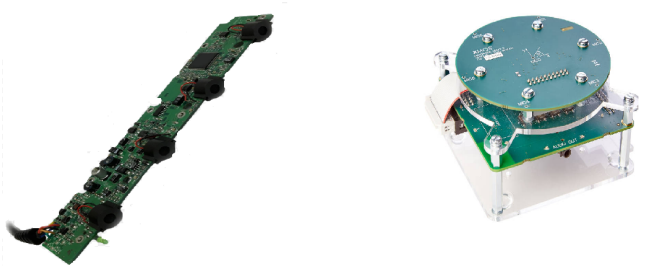

Figure 1: 线性和环形麦克风阵列。

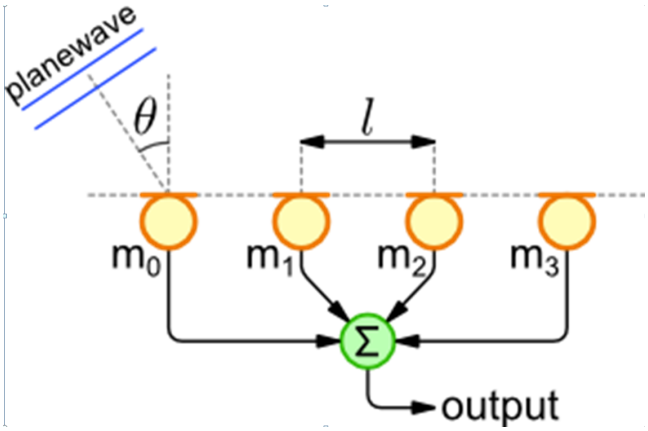

Figure 2: 线性麦克风阵列的声音采集和加和。

考虑如图2所示的四麦阵列，每两个麦克风之间的间隔为l，阵列的输出为四个麦克风输出的简单加和。对一个频率为f，由夹角θ入射的平面波，可以计算相临两个麦克风之间的信号延迟为:

$$\Delta t = \frac{lsin(\theta)}{c},$$

其中c为声速。由此可计算相临麦克风之间的相位差为$2\pi f\Delta t$。如果我们将最左边的麦克风接收到的信号记为$Ae^{j2\pi ft}$，则第i个麦克风的信号则为：$Ae^{j2\pi f(t+i\Delta t)}$。由此，可计算这四路麦克风输出的结果为：

$$\frac{1}{4}\sum_{i=0}^{3} Ae^{j2\pi f(t+i\Delta t)} = \frac{1}{4}\sum_{i=0}^{3} Ae^{j2\pi f(t+\frac{ilsin(\theta)}{c})}.$$

与单独一个麦克风相比，可知其输出的增益（以dB为单位）为：

$$20\log_{10}\frac{1}{4}\sum_{i=0}^{3} e^{j2\pi f\frac{ilsin(\theta)}{c}}$$

由上式可知，该增益是入射角θ的函数，如图3所示。为了更清晰地对比，图中同时给出了单一麦克风在不同方向上的增益函数。可见，阵列具有明显的方向选择性，只有在正前方的输入才有较好的增益，其它方向的输入都被抑制。这种指向性使得阵列可以选择特定方向的声音，抑制其它方向的声音，从而极大提高信噪比。同时，不同麦克风所接收的噪音是不相关的，这些不相关噪音在互相叠加时会互相抵消，因此可显著消除加性噪声的影响。

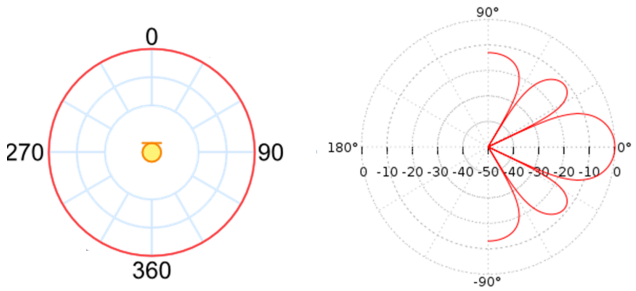

Figure 3: 单一麦克风（左）和线性麦克风阵列（右）在各方向上的增益。可见，麦克阵列具有明显的指向性。

对上述简单加和的阵列而言，其增益的指向性是固定的，即只有在正前面的增益最大。如果我们对每路麦克风的输出做适当延迟，再对延迟后的信号做加和，即可选择阵列的指向性。事实上，如果我们想对入射角为θ的方向做最大增益，只需对由该入射角引起的延迟Δt进入补偿即可，这一方法称为**延迟-加和**算法（Delay-Sum Algorithm），如图4所示。为提高延迟-加和算法的性能，研究者提出了各种改进方法，包括为每个麦克风引入增益参数，调节这些参数使之更适合语音识别任务等[9]。

2.2 特征域补偿方法

如前所述，语音增强方法的目的是增加语音的清晰度和可懂度，这一目标与语音识别有一定差距。对语音识别系统来说，特征本身的鲁棒性，或环境不

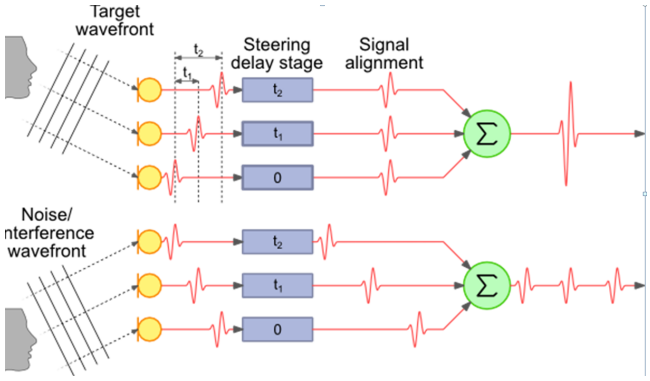

Figure 4: 线性麦克风阵列的延迟-加和算法。对目标语音方向，选择合理的延迟补偿，使得各路麦克风在做完补偿后的相位恰好一致，即可实现该方向的增益最大化。在这一设置下，其它方向的声音因相位失配，在输出信号中的增益减小。

变性更加重要。因此，在特征域上的补偿或正规化通常可起到更好的效果。我们将讨论三种方法：倒谱特征归一化（CMN & CVN）、向量泰勒展开（Vector Taylor Series, VTS）和SPLICE。

2.2.1 CMN & CVN

CMN和CVN是最常用的特征补偿方法，主要用来对卷积噪声进行消除。我们首先讨论CMN。我们已经知道，Fbank和MFCC是最常用的两种特征。这两种特征基于共同的前端处理：加窗、预加重、FFT变换、Mel频谱归整、频域能量加窗、log压缩。对一个在时域上卷积的信道噪声，通过上述过程可以实现分解。为清楚看到这一点，设原始信号为$x(t)$，带噪语音信号为$y(t)$，信道的卷积噪声为$h(t)$，则有：

$$y(t) = x(t) * h(t) \tag{1}$$

$$Y(w) = X(w)H(w) \tag{2}$$

$$log|Y(w)|^2 = log|X(w)|^2 + log|H(w)|^2 \tag{3}$$

由此可得:

$$\mathbf{y} = \mathbf{x} + \mathbf{h},$$

其中$\mathbf{x}$和$\mathbf{y}$分别为原始语音信号和带噪语音信号的Fbank特征，$\mathbf{h}$是和信道相关的卷积噪声。因此，如果我们可以估计出$\mathbf{h}$，即可估计出原始语音信号

的Fbank特征$\mathbf{x}$。在实际操作中，可以选择信号中的非语音片段来估计$\mathbf{h}$。进一步，如果我们假设$\mathbf{x}$是高斯的，也可以通过对整句语音信号取平均来得到，即：

$$\mathbf{h} = \boldsymbol{\mu}_{\mathbf{y}},$$

$$\hat{\mathbf{x}} = \mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}}.$$

MFCC特征是在Fbank特征基础上加入一个DCT变换，因为该变换是线性的，因此上述分解关系依然成立，即：

$$C\mathbf{y} = C\mathbf{x} + C\mathbf{h}$$

其中C是DCT的变换矩阵。由于MFCC是倒谱系数，上述方法称为**倒谱均值正规化**（Cepstra Mean Normalization, CMN）[10]。

形式上，CMN可以认为是对特征进行一阶归一化的方法。基于这一思路，可以设计一种二阶归一化方法，即对方差进行归一化，称为**倒谱方差正规化**（CVN）。实际应用中，CVN一般和CMN联合使用，称为CMVN，计算公式为：

$$\hat{\mathbf{x}} = \frac{\mathbf{y} - \boldsymbol{\mu}_{\mathbf{y}}}{\boldsymbol{\sigma}_{\mathbf{y}}},$$

其中的除法为按位除。和CMN不同，CVN并没有特别明确的物理背景，但在实际应用中通常会取得一定的性能提高。

CMN和CVN只对一阶和二阶量进行正规化，类似的思路可以扩展到对特征向量的分布进行正规化。一种方法是将特征向量的每一维都正规化到标准高斯分布，称为**特征的高斯化**[11]。高斯化通常采用统计方法，以直方图形式统计特征的实际分布，将其变换为累积概率分布，再将该分布映射到标准高斯分布的累积分布。高斯化对一些任务有一定效果，但在某些任务上的表现未必好于简单的CMN。

在实际系统中，为了保证实时性，需要设计一种在线CMN。在对一句话进行识别时，最初没有任何数据，这时采用缺省的CMN参数来对特征进行归一化；当数据逐渐积累后，对CMN参数逐渐求精，从而得到更好的归一化特征。这种在线估计可以理解为是一种高通滤波器（滤掉了固定不变的成份）。将正规化过程表述为一种滤波过程具有很大启发性，一些著名的去噪方法，如ARMA滤波和RASTA滤波[12]都遵循这一思路。

2.2.2 向量泰勒展开（VTS）

向量泰勒展开（VTS）是一种对加性噪声的建模方法。如在谱减法中所述，对于加性噪声，我们假设带噪语音的能量是原始语音和噪声语音的能量之和。

假设我们使用的是Fbank特征，这一关系可表示为：

$$e^{\mathbf{y}} = e^{\mathbf{x}} + e^{\mathbf{n}}.$$

做简单变换，有：

$$e^{\mathbf{y}} = e^{\mathbf{x}}(1 + e^{\mathbf{n}-\mathbf{x}}),$$

$$\mathbf{y} = \mathbf{x} + \ln(1 + \mathrm{e}^{\mathbf{n}-\mathbf{x}}).$$

如果记$\mathbf{r} = \mathbf{n} - \mathbf{x}$，且：

$$g(\mathbf{r}) = \ln(1 + \mathrm{e}^{\mathbf{r}}).$$

可得如下关系：

$$\mathbf{y} = \mathbf{x} + g(\mathbf{r}).$$

注意，$g(\mathbf{r})$是一个非线性函数。如果对这一非线性函数做一阶泰勒展开，即可得到带噪语音、原始语音和噪音之间的简单对应关系，从而由带噪语音推导出原始语音，这一方法称为VTS方法。一般假设原始语音具有混合高斯形式，噪声具有高斯形式。在这一假设下，可通过迭代求出在每一个高斯成分s下，$\mathbf{r}$的期望$\mu_s^{\mathbf{r}}$，并由此得到基于状态s的原始语音估计$\hat{\mathbf{x}}$如下[13]：

$$\hat{\mathbf{x}} = \mathbf{y} - \ln(\mathrm{e}^{\mu_{\mathrm{s}}^{\mathbf{r}}} + 1) + \mu_{\mathrm{s}}^{\mathbf{r}},$$

由上述推导过程可知，VTS的基本假设是噪声是加性的，因此语音和噪声之间的能量具有加和关系。基于这一基本假设，VTS推导出基于Fbank特征，带噪语音和原始语音之间的关系，并用泰勒展开对这一关系进行近似。值得说明的是，对倒谱特征，如MFCC，上述推导过程依然成立，只不过需要加入一个DCT变换。

2.2.3 SPLICE

SPLICE是另一种对特征进行建模的方法。和VTS不同，SPLICE并不假设噪音的加性，而是直接对原始语音和带噪语音的特征向量建立联合概率分布。为保证建模精确性，SPLICE采用GMM模型：

$$p(\mathbf{y}, \mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}|\mathbf{y}, k)p(\mathbf{y}, k)$$

其中$p(\mathbf{y}, k)$也是一个GMM:

$$p(\mathbf{y}, k) = p(\mathbf{y}|k)p(k).$$

SPLICE定义条件概率$p(\mathbf{x}|\mathbf{y}, k)$具有如下线性形式：

$$p(\mathbf{x}|\mathbf{y}, k) = N(\mathbf{x}; A_k\mathbf{y} + b_k, \Sigma_k),$$

则可由带噪语音估计出原始语音：

$$\hat{\mathbf{x}} = \sum_{k=1}^{K}(A_k\mathbf{y} + b_k)p(k|\mathbf{y}).$$

SPLICE模型中的$p(\mathbf{y}, k)$部分可通过对带噪语音的GMM建模实现，而条件概率$p(\mathbf{x}|\mathbf{y}, k)$中的参数$\{A_k, b_k\}$一般需要基于原始语音和相应的带噪语音数据（Stereo Data）进行训练。

2.3 基于DNN的特征映射

前面所述的大部分方法都假设了一个物理过程，基于该物理过程进行建模。这些方法具有较好的理论基础，需要的数据和计算量通常较小。然而，这些建模方法都或多或少引入了一些人为假设，这些假设在实际应用中可能无法满足，带来模型的不精确性。同时，对一些难以建模的场景（如传输过程中信道的即时改变），这些方法也很难凑效。近年来，深度神经网络（DNN）成为语音信号处理的强大工具。DNN的一个显著优势是可以近似任何映射函数，因此可以学习任何复杂的信号传递过程。我们可以利用这一能力，基于DNN将复杂环境中的语音信号或特征映射成安静环境下的信号或特征。研究表明，基于DNN的特征映射方法可取得非常好的效果[14, 15, 16]。

去噪自编码器（Denoising Auto Encoder, DAE）是一种常见的特征映射模型。和SPLICE一样，我们需要准备一份干净数据和一份相应的带噪数据，将带噪数据输入DAE，输出的目标是对应的干净数据。通过训练DAE的参数，即可学习到由带噪语音（或特征）还原出原始语音（或特征）的映射函数。图5给出一个利用DAE去除音乐噪音的例子[17]，可以看到，DAE可以有效恢复被音乐破坏的语音数据。

Kaldi的THCHS30 recipe提供了DAE训练流程样例，如图6所示。在这一流程中，对每一条训练语句，随机选出一种噪声（基于狄利克雷分布）以及噪声大小（基于高斯分布），将该噪声按选定的大小混入训练语句中，形成DAE的一个训练对。在训练时，对原始数据和加噪后的数据分别提取Fbank特征，DAE学习由带噪特征到原始特征的映射函数。

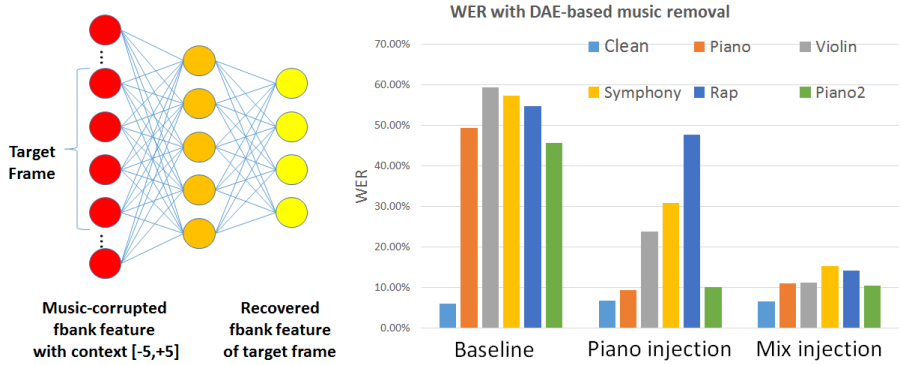

Figure 5: 基于DAE的音乐噪音消除。左图是DAE的结构，右图是实验结果。其中每一组直方图表示一个系统，每一组里的一个直方图代表测试数据中包括某种音乐的结果。从第一组结果可以看到，基于原始模型，在测试数据中加入音乐后性能显著下降；从第二组结果可以看到，即使只加入一种音乐做DAE训练，也可明显提高系统性能，即使对没见过的音乐也是如此；从第三组结果看到，当加入更多类型的音乐进行训练后，性能有了进一步提高。

3 后端模型增强方法

后端模型增强方法通过调整声学模型，使系统适应实际应用场景。依系统模型种类不同（如HMM-GMM 或DNN），模型增强的方式也不同。我们主要讨论三种模型增强方法：基于噪声模型的简单模型增强方法、模型自适应和带噪训练。

3.1 简单模型增强

对于一个HMM-GMM系统，如果我们假设噪音是加性的，则在2.2.2一节所讨论的对特征的补偿方法可以同样用于对模型参数的补偿。和特征补偿相比，这种在模型上的补偿更加灵活，性能通常也更好。

以Fbank特征为例，HMM-GMM系统假设分配到每个高斯成份上的语音帧（Fbank）是高斯分布的，同样，噪声也是高斯分布的。这一高斯分布映射到频域能量谱上，分别记为X和N，这两者也是随机变量，分布为对数高斯（即取对数后是高斯的）。引入加性噪声假设，得到带噪语音的能量谱为$Y = X + N$。一般来说，视X和N的相关性以及它们的相对强弱，Y的分布是不规则的。如果我们假设Y依然是对数高斯的，即可求出对应到Fbank域的高斯分布的参

```
#quick_ali
steps/align_fmllr.sh --nj $n --cmd "$train_cmd" data/mfcc/train data/lang exp/tri4b exp/tri4b_ali || exit 1;

#quick_ali_cv
steps/align_fmllr.sh --nj $n --cmd "$train_cmd" data/mfcc/dev data/lang exp/tri4b exp/tri4b_ali_cv || exit 1;

#train dnn model
local/nnet/run_dnn.sh --stage 0 --nj $n  exp/tri4b exp/tri4b_ali exp/tri4b_ali_cv || exit 1;

#train dae model
#python2.6 or above is required for noisy data generation.
#To speed up the process, pyximport for python is recommeded.
local/dae/run_dae.sh $thchs || exit 1;
~
~
~
```

```
#!/bin/bash
# Copyright 2016  Tsinghua University (Author: Dong Wang, Xuewei Zhang).  Apache 2.0.
#           2016  LeSpeech (Author: Xingyu Na). Apache 2.0

# Conducts experiments of dae-based denoisng

stage=-1
nj=8

. ./cmd.sh ## You'll want to change cmd.sh to something that will work on your system.
           ## This relates to the queue.

. ./path.sh ## Source the tools/utils (import the queue.pl)
. utils/parse_options.sh || exit 1;

thchs=$1

if [ $stage -le -1 ]; then
  echo "DAE: switching to per-utterance CMVN mode"
  for x in train dev test test_phone; do
    mv data/fbank/$x/cmvn.scp data/fbank/$x/cmvn.scp.per_spk
    mv data/fbank/$x/spk2utt data/fbank/$x/spk2utt.per_spk
    mv data/fbank/$x/utt2spk data/fbank/$x/utt2spk.per_spk
    awk '{print $1 " " $1}' data/fbank/$x/utt2spk.per_spk > data/fbank/$x/utt2spk
```

Figure 6: Kaldi THCHS30 recipe中提供的DAE训练和识别脚本。上图是run.sh中的调用程序，下图是local/dae/run_dae.sh脚本。

数，由此实现对原模型的修正。这一方法称为平行模型加和（Parallel Model Combination）[18]。

另一种方法是将特征域上的VTS补偿应用到模型增强，即不对特征进行修正，而是对模型参数进行改进，以更好描述加噪后的语音。这一方法同样用到加性噪声假设，但和PMC中的对数高斯近似不同，VTS基于泰勒展开对$\mathbf{y}$和$\mathbf{x}$之间的关系做近似[19]。

上述两种方法相对简单，但只能处理加性噪声，且只能应用于HMM-GMM系统，现在已经较少应用。

3.2 模型自适应

如果我们将实际应用环境看作是与训练不同的另一种声学场景，则可基于“说话人自适应”一节中所提到的领域自适应方法，利用应用场景的数据对模型进行更新。对HMM-GMM系统，一般采用MAP和MLLR两种方法；对DNN 系统，可在原模型基础上进行再训练，训练时选择较小的步长；或采

用知识迁移方法[20]，以原系统的输出作为约束，以减小过拟合的风险。当前对DNN模型最有效的自适应方法还是基于i-vector的条件学习方法。前面提到过，i-vector事实上是一种全信息向量，包括说话人、信道、语言、情绪等多种长时信息，因而可以充分覆盖噪声、混响、编码方式等环境因子的变化。实验表明，将i-vector作为一种辅助信息引入到DNN模型训练和识别过程中，可以非常有效地对抗环境影响。

3.3 多场景学习和数据增强

DNN的一个显著优势是可以进行**多场景进行学习**。在传统GMM-HMM系统中，虽然我们可以通过收集更多实际应用场景的数据来提高系统性能，但由于模型限制，当收集的数据具有较大差异性时，将导致音素的区分性下降。这意味着大量数据虽然可以提高对场景的覆盖能力，但对某一应用场景来说，多场景学习并不能达到单一场景建模的效果。DNN极大改变了这种状况。实验表明，DNN模型可以有效学习多场景下的数据，这些各异场景的数据不仅不会降低音素的区分性，反而会互相促进，在各种场景下都能得到同步提高[21]。这一结果具有重要意义，说明如果我们可以收集到足够多、对场景覆盖足够全的数据，那么一个DNN系统即可在所有场景下顺利工作。这事实上已经在原则上解决了环境鲁棒性的问题。某种程度上说，DNN的这种多场景学习能力是今天大规模商用语音识别系统的基础。

仅管如此，我们依然要考虑如何有效利用DNN的这种多场景学习能力。这是因为数据天然具有长尾效应：绝大部分数据可能是正常的，但对很多特别场景（如特别强的噪音，特别强的混响，很少用的编码方式等），数据通常是不足的。数据在场景上的分布不均衡事实上带来了另一种更深刻的环境鲁棒性问题。**数据增强**（Data Augmentation)[22]，或**带噪训练**（Noisy Training），是解决数据不均衡问题的有效方法。具体来说，数据增强方法对原始训练数据进行各种变换，以模拟不同场景下语音信号的变异情况。这些模拟包括在数据中随机加入不同类型的噪声，让数据通过随机生成的RIR，通过各种编解码器进行重构等。实验发现，数据增强方法可极大提高系统的鲁棒性，特别是提高非典型场景下的识别性能[23, 24]。

4 小结

我们简要介绍了提高识别系统鲁棒性的几种方法，这些方法可分为前端信号处理和后端模型增强两类。前端信号处理方法的目的是对不同环境下的语音信号或特征进行归一化，使之可以适应标准语音训练出的模型。最常用的前端

处理方法是CMN，这种方法简单、高效，且有明确物理意义，被广泛应用于各种商用识别系统。另一种前端处理方法是基于DAE的特征映射。归因于神经网络强大的函数学习能力，DAE可以实现对各种复杂声学场景的归一化。后端模型增强方法的基本思路是对模型进行改进，使之对目标场景有更好的识别效果。模型增强的主要方法有两种，一是对模型进行自适应，使其适应目标场景，二是多场景学习，提高模型的泛化能力。对DNN来说，多场景训练一般可取得较好的效果，利用数据增强方法可以进一步提高对非典型场景的覆盖。

References

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 27, no. 2, pp. 113–120, 1979.

[2] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'79.*, vol. 4. IEEE, 1979, pp. 208–211.

[3] B. W. Gillespie and L. E. Atlas, "Acoustic diversity for improved speech recognition in reverberant environments," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 1. IEEE, 2002, pp. I–557.

[4] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 36, no. 2, pp. 145–152, 1988.

[5] B. Yegnanarayana and P. S. Murthy, "Enhancement of reverberant speech using lp residual signal," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 3, pp. 267–281, 2000.

[6] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Blind speech dereverberation with multi-channel linear prediction based on short time fourier transform representation," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.* IEEE, 2008, pp. 85–88.

[7] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing.* Springer Science & Business Media, 2008.

[8] K. Kumatani, J. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 127–140, 2012.

[9] M. L. Seltzer, B. Raj, R. M. Stern *et al.*, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *IEEE Transactions on Audio, Speech and Language*, vol. 12, no. 5, 2004.

[10] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *the Journal of the Acoustical Society of America*, vol. 55, no. 6, pp. 1304–1312, 1974.

[11] A. De La Torre, A. M. Peinado, J. C. Segura, J. L. Pérez-Córdoba, M. C. Benítez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, 2005.

[12] H. Hermansky and N. Morgan, "Rasta processing of speech," *IEEE transactions on speech and audio processing*, vol. 2, no. 4, pp. 578–589, 1994.

[13] J.Droppo and A. Acero, "Environmental robustness," in *Springer handbook of speech processing*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. springer, 2007.

[14] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1759–1763.

[15] K. Han, Y. Wang, D. Wang, W. S. Woods, I. Merks, and T. Zhang, "Learning spectral mapping for speech dereverberation and denoising," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 6, pp. 982–992, 2015.

[16] B. Wu, K. Li, M. Yang, and C.-H. Lee, "A reverberation-time-aware approach to speech dereverberation based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 102–111, 2017.

[17] M. Zhao, D. Wang, Z. Zhang, and X. Zhang, "Music removal by convolutional denoising autoencoder in speech recognition," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*. IEEE, 2015, pp. 338–341.

[18] M. J. F. Gales, "Model-based techniques for noise robust speech recognition," Ph.D. dissertation, University of Cambridge Cambridge, 1995.

[19] P. J. Moreno, B. Raj, and R. M. Stern, "A vector taylor series approach for environment-independent speech recognition," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2. IEEE, 1996, pp. 733–736.

[20] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5900–5904.

[21] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks-studies on speech recognition tasks," *arXiv preprint arXiv:1301.3605*, 2013.

[22] S. Yin, C. Liu, Z. Zhang, Y. Lin, D. Wang, J. Tejedor, T. F. Zheng, and Y. Li, "Noisy training for deep neural networks in speech recognition," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2015, no. 1, p. 2, 2015.

[23] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," in *Interspeech 2017*, 2017.

[24] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5220–5224.