第 5 章 语音增强原理与技术

【内容导读】语音增强旨在去除环境干扰音以增强语音信号的质量，按照参与录音的麦克风数量可以分为单通道语音增强和麦克风阵列语音增强。近年来，随着深度学习在语音领域广泛应用，增强的方法也逐渐由信号处理方法转向神经网络方法。本章在系统详细回顾语音增强技术发展的同时，重点分析了基于深度学习的语音增强和麦克风阵列语音增强方法，并对其未来发展进行展望。在本章结尾提供语音增强的实验样例。

5.1 语音增强概述

在现实语音交互场景中，目标语音信号并不是单独存在的，往往伴随着环境噪声的干扰，例如图 5-1 中所示的背景噪声、混响等干扰。这些环境噪声干扰使得目标语音信号严重受损，进而影响语音应用的正常交互和使用。因而，语音增强技术作为语音信号处理领域的核心问题，对语音识别、说话人识别、语音对话等一系列重要任务都具有非常重要的研究意义和应用价值[1]。尤其在近些年，随着智能设备和便携式计算设备的爆炸式发展，语音已经成为人类接入智能计算设备和平台的最重要入口之一。基于此，面对日常生活中典型和常见的复杂应用环境，如何设计一个有效且易用的语音增强系统将是语音交互研究的重中之重。

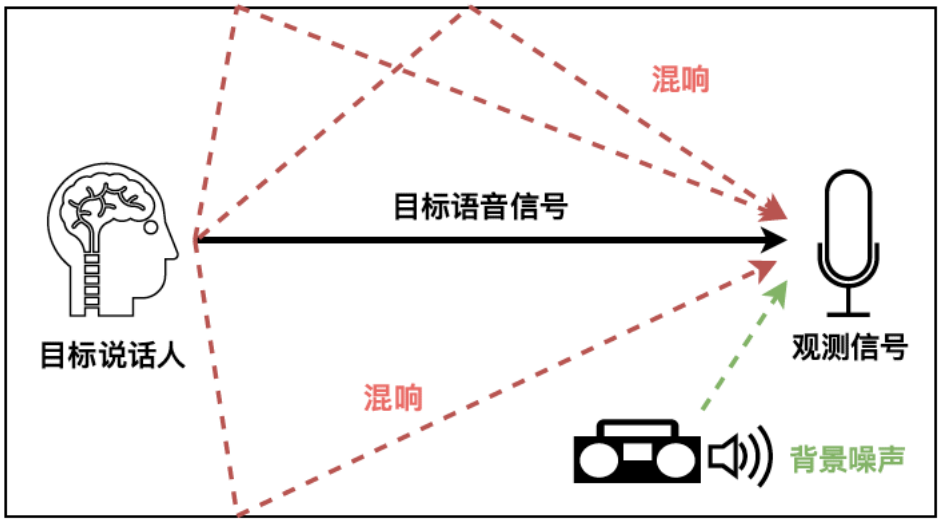


图 5-1 单说话人的现实复杂声学场景

在本章节中，我们将会介绍语音增强是个什么样的任务，以及语音增强期望得到什么样的目标信号。语音增强可以分为两种，一种是提升带噪信号的人耳听觉感受，而另一个则更偏向于为语音任务提供更好的特征。本章节的语音增强是为提升带噪信号的人耳听觉感受。

语音增强的目标是通过去除环境噪声干扰来提升受损语音的质量以及可懂度。可懂度也称为语言清晰度（speech intelligibility），听者能听懂通过一定传声系统传递的语音信号的百分率。语音质量和可懂度都是语音信号的诸多特性之一，二者并不等同[2]。因而，语音增强的具体目标往往与具体语音应用有关，不同的语音产品侧重的目标不太一样。例如，在空对

地通讯等过程中，驾驶舱里面极大的噪声会对飞行员的声音形成严重的干扰，此时我们就需要语音增强技术来预处理语音提升地面接收端接收的语音质量和可懂度。这种类似的军用通讯设备和系统，通常对于语音可懂度的要求要高于语音质量的要求。在电话会议或者视频会议系统中，往往一个终端的噪声会被传送到其他所有的接收终端上，如果此时这个终端所在的房间存在严重的混响的话，电话会议或者视频会议的效果会特别糟糕。再比如，对于那些需要助听设备或者人工耳蜗设备辅助的听障人士来说，环境噪声干扰过大将会使得交流非常困难，因而语音增强算法可以将带噪语音信号被放大之前进行预处理，从而在一定程度上抑制噪声。这两个例子就对语音质量提了更高的要求。

在理想情况下，我们希望语音增强技术既能改善语音质量，又能提升语音可懂度。然而在实际应用产品中，语音增强技术在减少环境噪声的同时，也会引入语音的失真，进而损伤了语音的可懂度，因此大多数语音增强算法只是改善了语音的质量[3]。由此可知，语音增强技术的关键挑战是如何设计一个高效易用的算法，在不引入明显信号失真的前提下，对环境噪声干扰进行有效抑制。

语音增强技术的具体解决方案与很多因素密切相关，包括具体的应用场景、噪声源或者干扰源的特性、噪声与干净目标语音之间的关系、麦克风的数量等。首先我们了解下噪声源的特性和区别，噪声源可以是平稳的，即不随时间而改变，例如空调风扇冰箱等背景噪声。噪声也可以是非平稳的，例如在餐馆里的背景噪声，包含多个说话人的声音夹杂着各种设备等噪声。这种餐馆里的噪声时域和频域特性会随着说话人之间的交谈内容的改变而随时改变。显而易见，非平稳噪声的语音增强技术难度远远超过平稳噪声的增强技术难度。同时，噪声源对于干净的目标信号源而言，噪声可以是加性噪声，也可以是卷积噪声，例如在密闭房间中产生的混响就是典型的卷积噪声。这种房间混响的噪声不同于加性噪声简单的线形叠加，它往往取决于房间本身的声学特性，包括房间的体积大小和吸声量强度。体积大且吸声量弱的房间，混响时间长，体积小且吸声量强的房间，混响时间短。混响时间通常用 T60 来描述，其定义为声源停止发生后，声压级别减少 60 分贝所需要的时间即为混响时间。混响时间过短，声音发干，枯燥无味不亲切自然，混响时间过长，会使声音含混不清，这都将严重损坏原始干净语音的质量。在学术研究上，房间混响通常是由干净的语音信号和房间冲击响应信号通过卷积操作产生，因此称卷积噪声。如图 5-2 所示，房间冲击响应信号通常可以分为三个部分，分别是直接路径响应、早期混响和晚期混响。早期混响一般是 30-50 毫秒，晚期混响室 100-1000 毫秒。有关学术研究表明，早期混响成分有助于提高语音的可懂度，因而诸多语音增强技术致力于对晚期混响的抑制和消除[4]。

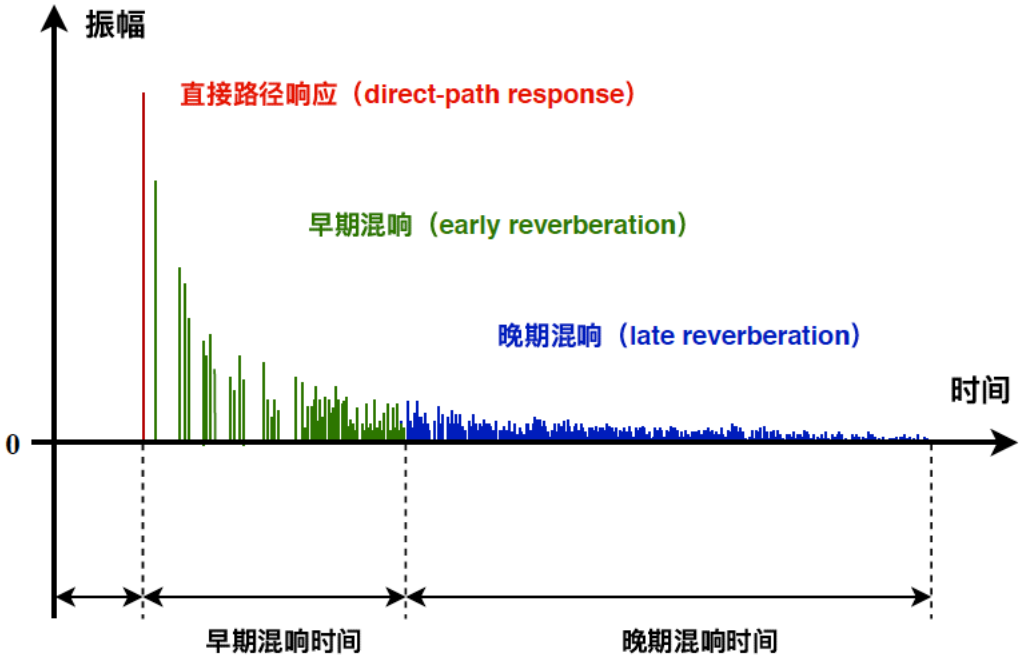


图 5-2. 房间冲击响应示意图

麦克风的数量也和语音增强的效果紧密相关。根据麦克风的数量，语音增强技术可以分为单通道语音增强技术和多通道语音增强技术。多通道语音增强相比于单通道语音增强，不仅拥有时域和频域的语音信息，还能获取麦克风阵列和目标声源头之间的空间信息。一般来说，麦克风的数量越多，越容易抑制噪声增强语音，从而最后语音增强的效果会越好。因而，在实际语音交互产品中，通过麦克风阵列来实现多通道语音增强提升语音质量可懂度的方式越来越普遍，特别是在智能机器人和智能音响上的应用。本书语音增强技术章节也将分为单通道语音增强和多通道语音增强技术进行详细阐述。

衡量语音增强算法的性能，往往是通过对增强后语音的语音质量和可懂度的评估来总体判断。语音质量的评价可以通过主观听音测试和客观音质测量。主观评价是通过一组试听者比较原始语音和处理后的语音，并按照预先设置好的等级来对音质进行评分。客观评价则是对原始语音和处理后的语音信号进行统计对比，量化的数值通常能反映两者之间的相似度，从而体现音质的差别。可懂度与音质不同，可懂度是非主观的，他的指标是指说话人所说的“内容”，例如所说的词语的含义或者内容。这个可以很容易地评估，即通过试听者试听语音材料，让他们辨识所听到的单词，通过计算辨识正确的单词或者因素的数量就可以得到可懂度。由于主观评价需要邀请试听者试听来评价，比较费时费力，因而近几年的研究中使用客观指标来评估语音质量和可懂度的越来越多。表 5-1. 常用语音质量和可懂度的评价指标表 5-1 总结了几种常用的客观评价指标。

表 5-1. 常用语音质量和可懂度的评价指标

软件	指标
ITU-T (2001)	感知语音质量评估（PESQ）[5]
Taal et al. (2011)	短时客观可懂度（STOI）[6]
Loizou (2007)	信噪比（Segmental SNR）
	对数似然比（Log-likelihood ratio）
	倒谱距离（Cepstral distance）
BSS Eval (Vincent et al. 2006)	信号失真比（SDR）[7]
	信号干扰比（SIR）[7]
	信号伪影比（SAR）[7]
Falk et al. (2010)	语音混响调制能量比（SRMR）[8]

5.2 单通道语音增强

单通道语音增强通常分为基于传统信号处理的语音增强方法和基于深度学习的语音增强方法。本小节中，我们将会介绍几种典型的基于传统信号处理的语音增强方法和基于深度学习的语音增强方法。

5.2.1 基于传统信号处理的语音增强方法

先前的研究中针对加性噪声的抑制提出了很多算法，例如谱减法、维纳滤波器、卡尔曼滤波器等等。这些算法对于含噪语音的加性噪声能够很好地抑制，本小节将主要介绍两种常用的语音增强算法，谱减法和维纳滤波器。

5.2.1.1 谱减法

谱减法是一种用于降低语音中加性噪声频谱效应的噪声抑制方法。描述这种算法变化的论文比任何其他算法都多。它基于一个简单的原则: 假设加性噪声，我们可以通过从有噪声的语音频谱中减去噪声频谱的估计来得到干净信号频谱的估计。当信号不存在时，可以估计和更新噪声谱。假设噪声是平稳的或缓慢变化的过程，并且噪声谱在更新周期之间没有显著变化。增强信号是通过计算估计信号频谱的反离散傅里叶变换得到的。该算法计算简单，因为它只涉及一个正傅里叶变换和一个反傅里叶变换。

简单的减法处理是有代价的。减法过程需要谨慎地进行，以避免任何语音失真。如果减得太多，那么一些语音信息可能会被删除，而如果减得太少，那么很多干扰噪声仍然存在。许多方法被提出来缓解，在某些情况下消除，大部分语音失真的谱减法过程。对含噪语音进行简单的减法处理是有风险的，如果减去的内容过多，可能会导致部分语音信息的损失，而如果减去的内容过少，语音中仍然存在残留的干扰噪声。因此减法过程需要谨慎地进行，避免造成语音失真。为了缓解谱减法过程中的语音失真，很多改进方法被提出。本章介绍谱减法的基本算法。

在现实声学场景中，由于噪声和混响的存在麦克风拾取到的语音信号如下:

$$y(n) = x(n) * h(n) + v(n) \tag{5. 1}$$

公式(5. 1)中， $x(n)$ 是干净语音信号，$h(n)$和$v(n)$分别为房间脉冲响应（Room Impulse Response, RIR）和加性噪声。在封闭空间中，语音信号会经过墙壁等物体的多次反射被麦克风拾取到，从而导致混响的出现。除了混响，加性噪声（背景噪声）也会被麦克风拾取到，比如机车鸣笛、风扇转动发出的声音等。混响和加性噪声都会对语音信号带来很大的影响。本章节只考虑加性噪声，因此我们将语音信号用以下数学模型表示：

$$y(n) = x(n) + v(n) \tag{5. 2}$$

对两边进行离散傅里叶变换得到

$$Y(\omega) = X(\omega) + V(\omega) \tag{5. 3}$$

我们可以用极坐标形式表达$Y(\omega)$如下：

$$Y(\omega) = |Y(\omega)|e^{j\theta_y(\omega)} \tag{5. 4}$$

其中$|Y(\omega)|$是幅度谱，$\theta_y(\omega)$是含噪语音的相位。

噪声频谱$V(\omega)$也可以由幅度和相位来表示：$V(\omega) = |V(\omega)|e^{j\theta_v(\omega)}$。噪声的幅度谱$|V(\omega)|$未知，但可以使用它的非语音活动期间（例如，在静音或语音停顿期间）的平均值计算得到估计值$\left|\hat{V}(\omega)\right|$。同样,噪声的相位谱$\theta_v(\omega)$可以用含噪语音的相位谱$\theta_y(\omega)$来进行粗略表示。假设干净语音$x(n)$和噪声$v(n)$ 是不相关的，

$$|Y(\omega)|^2 = |X(\omega)|^2 + |V(\omega)|^2$$

谱减法的大致思路如下：带噪语音的能量谱减去噪声的能量谱得到干净语音能量谱的估计值（因此得名“谱减法”），从而得到语音振幅谱的估计值$\left|\hat{X}(\omega)\right|$以及干净语谱的估计值$\hat{X}(\omega)$，

$$\left|\hat{X}(\omega)\right| = (|Y(\omega)|^2 - \left|\hat{V}(\omega)\right|^2)^{1/2} \tag{5. 5}$$

$$\hat{X}(\omega) = \left|\hat{X}(\omega)\right| e^{j\theta_y(\omega)}$$

干净语音估计的相位谱可以用带噪语音的相位谱$\theta_y(\omega)$来进行粗略表示，这是由于在一定程度上，相位不会严重影响语音质量。增强语音信号可以通过简单的求$\hat{X}(\omega)$的傅里叶逆变换来获得。　　图 5-3 为谱减法的简易计算流程图。

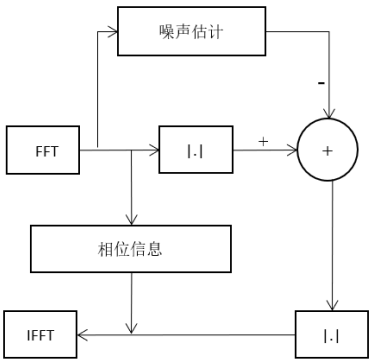


图 5-3. 谱减法流程图

从含噪语音的能量谱减去噪声语音的能量谱估计值可能产生负的能量谱值。为了避免这一现象，以及对能量谱或者振幅谱进行一般化操作，广义谱减法的公式如下所示：

$$|\hat{X}(\omega)|^{2n} = max(|Y(\omega)|^{2n} - \alpha \cdot |\bar{V}(\omega)|^{2n}, \beta \cdot |Y(\omega)|^{2n}) \quad (5.6)$$

其中α代表噪声过估计因子，是依赖于信噪比并按照先验知识进行定义的；β代表频谱下限系数，用于防止产生负的频谱结果；n代表任意指数参数，当n的取值为 1 时表示对能量谱进行操作，取值为 0.5 时表示对幅度谱进行操作。

5.2.1.2 维纳滤波器

谱减法算法很大程度上是基于直觉和启发式的原则，更具体地说，这些算法利用了噪声是加性的这一事实，人们可以通过简单地从有噪声的语音频谱中减去噪声频谱来获得干净语音信号的频谱估计。增强信号的频谱并不是最佳的去噪方法。现在我们把注意力转向基于维纳滤波的语音增强方法，它通过优化数学上可处理的误差准则和均方误差来获得增强信号。

考虑到图 5-4 中给出的统计过滤问题，输入信号经过一个线性时不变系统产生一个输出信号$y(n)$。我们需要设计一个系统，使其的输出信号尽可能的得到所需信号$d(n)$。这可以通过计算估计误差$e(n)$来实现，并使其尽可能小。使估计误差最小化的最优滤波器称为维纳滤波器，它以数学家诺伯特维纳命名，他首先提出并解决了连续域的这个滤波问题。

应该注意的是，滤波器的一个限制条件是它是线性的，因此很容易对其分析处理。原则上，滤波器可以是有限脉冲响应(FIR)或无限脉冲响应(IIR)，但通常使用 FIR 滤波器，这是因为(1)它们本身是稳定的，(2)其结果是线性的，并且计算起来容易。假设有 FIR 系统(参见图 5-5)，我们有

$$\hat{d}(n) = \sum_{k=0}^{M-1} h_k y(n-k), n = 0,1,2,\ldots \quad (5.7)$$

式中$\{h_k\}$为 FIR 滤波器系数，M 是系数的个数。接下来,我们需要计算滤波器系数，因此需要估计误差，也就是说使得$d(n) - \hat{d}(n)$最小。估计误差的均方通常用作最小化的准则，

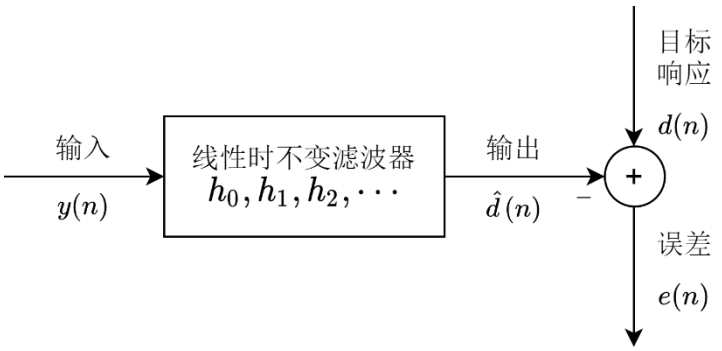


图 5-4. 统计滤波问题的流程

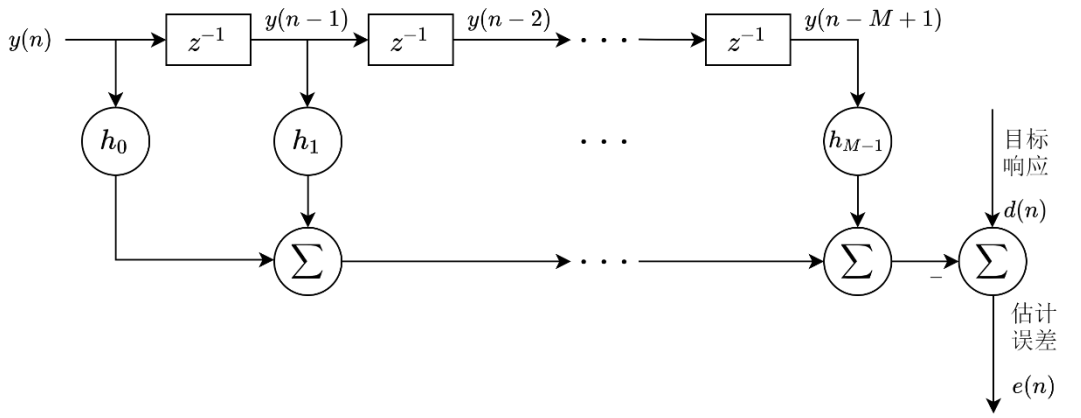


图 5-5. 利用 FIR 滤波器的最优滤波问题的框图

可以在时域或频域上推导出最优滤波器系数, 这里我们将对时域上的维纳滤波器进行详细的介绍:

首先我们假设输入信号 y(n)和期望信号 d(n)是联合广义随机过程的实现，估计误差 e(n)可以计算为:

$$e(n) = d(n) - \hat{d}(n) = d(n) - \mathbf{h}^{T}\mathbf{y} \tag{5. 8}$$

其中$\mathbf{h}^T = [h_0, h_1, h_2, \dots, h_{M-1}]$是滤波系数矩阵, $\mathbf{y}^T = [y_n, y_{n-1}, y_{n-2}, \dots, y_{n-M+1}]$是输入向量它包含输入的过去 M 个采样点。为了找到最优滤波系数，我们将 e(n)的均方值最小化，即使得 e $[e^2(n)]$最小，其中 e[·]为期望算子。均方误差为:

$$\begin{aligned} J = E[e^2(n)] &= E(d(n) - \mathbf{h}^{\mathrm{T}}\mathbf{y})^2 \\ &= E[d^2(n)] - 2\boldsymbol{h}^T E[\boldsymbol{y}d(n)] + \mathbf{h}^T E[\mathbf{y}\mathbf{y}^T]\boldsymbol{h} \\ &= E[d^2(n)] - 2\mathbf{h}^T\mathbf{r}_{yd}^{-} + \mathbf{h}^T\mathbf{R}_{yy}\mathbf{h} \end{aligned} \tag{5. 9}$$

其中$\mathbf{r}_{yd}^{-} \triangleq E[\mathbf{y}d(n)] = E[(y(n), y(n-1), \dots, y(n-M+1))d(n)]$是输入和期望信号之间的互相关矩阵(Mx1)，$\mathbf{R}_{\mathrm{yy}} = E[\mathbf{y}\mathbf{y}^T]$是输入信号的自相关矩阵。$\mathbf{r}_{\mathrm{yd}}^{-}$的上标用来表示互相关向量的第 m 个元素实际上也可以表示为$\mathbf{r}_{\mathrm{yd}}(-m)$。从公式(5. 9)，我们推导出均方误差$J$是系数 **h** 的二次函数，因此有一个单一的最小值。例如，如果滤波器只有两个系数 $\mathbf{h} = [h_0, h_1]$，那么误差曲面的形状将是一个碗形，只有一个最小值。

为了使损失函数J达到最小值，我们要求梯度向量的所有元素使其都为零，即

$$\frac{\partial J}{\partial h_k} = 0 = 2E[e(n)\frac{\partial e(n)}{\partial h_k}], k = 0,1, \dots, M-1 \tag{5. 10}$$

$$\frac{\partial J}{\partial h_k} = -2E[e(n)y(n-k)] = 0, k = 0,1, \dots, M-1 \tag{5. 11}$$

上式给出代价函数 J 达到其最小值的充要条件。估计误差 e(n)需要与输入信号 *y(n)* 正交[e(x·z) = 0 时两个随机变量 x 和 z 正交]。这一陈述构成了众所周知的最优线性滤波

的线性原理。

接下来，我们继续推导最优滤波器系数。利用矩阵导数和向量导数的性质，我们计算了 J 对向量 **h** 的导数

$$\frac{\partial J}{\partial \mathbf{h}} = -2\mathbf{r}_{\mathrm{yd}}^{-} + 2\mathbf{h}^{T}\mathbf{R}_{\boldsymbol{yy}} = 0 \tag{5. 12}$$

求解方程的 **h**，得到最优滤波系数 **h***:

$$\boldsymbol{R}_{\mathrm{yy}}\boldsymbol{h}^{*} = \boldsymbol{r}_{\mathrm{yd}}^{-} \tag{5. 13}$$

我们可以将上式进一步表示为

$$\sum \mathbf{h}_{k}\mathbf{r}_{\mathrm{yy}}(m-k) = \mathbf{r}_{yd}(-m), m = 0,1,\ldots,M-1 \tag{5. 14}$$

可见，通过求解 M 个未知数方程组$\{h_k\}$，可以得到滤波器系数h_k。求出方程中的**h**，我们得到

$$\boldsymbol{h}^{*} = \boldsymbol{R}_{\mathrm{yy}}^{-1}\boldsymbol{r}_{\mathrm{yd}}^{-} \tag{5. 15}$$

公式(5. 15)中的前一个解称为 Wiener-Hopf 解，也可以用矩阵形式表示为

$$\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ \vdots \\ h_{M-1} \end{bmatrix} = \tag{5. 16}$$

$$\begin{bmatrix} r_{yy}(0) & r_{yy}(1) & r_{yy}(2) & \cdots & r_{yy}(M-1) \\ r_{yy}(1) & r_{yy}(0) & r_{yy}(1) & \cdots & r_{yy}(M-2) \\ r_{yy}(2) & r_{yy}(1) & r_{yy}(0) & \cdots & r_{yy}(M-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{yy}(M-1) & r_{yy}(M-2) & \cdots & r_{yy}(1) & r_{yy}(0) \end{bmatrix}^{-1} \begin{bmatrix} r_{yd}(0) \\ r_{yd}(-1) \\ r_{yd}(-2) \\ \vdots \\ r_{yd}(-M+1) \end{bmatrix}$$

由于假设输入信号是广义平稳的，因此自相关矩阵$\boldsymbol{R}_{yy}$是对称的，Toeplitz(沿对角线的值相同)。为了$\boldsymbol{R}_{\boldsymbol{yy}}$的逆矩阵，可以使用有效的数值技术，如 Levinson-Durbin 算法。

为了求均值平方误差 J 的最小值，我们将公式(5. 15)代入公式(5. 9)中的最优滤波算子，得到

$$J_{min} = E[d^2(n)] - 2\left(\boldsymbol{R}_{yy}^{-1}r_{yd}^{-1}\right)^T r_{yd}^{-1} + \left(\boldsymbol{R}_{yy}^{-1}r_{yy}^{-1}\right)^T \boldsymbol{R}_{yy}\boldsymbol{R}_{yy}^{-1}r_{dd}^{-} \tag{5. 17}$$

$$= E[d^2(n)] - 2\left(r_{yd}^{-}\right)^T \boldsymbol{R}_{yy}^{-T}r_{yd}^{-} + \left(r_{yd}^{-}\right)^T \boldsymbol{R}_{yy}^{-T}r_{yd}^{-}$$

$$= E[d^2(n)] - \left(r_{yd}^{-}\right)^T \boldsymbol{R}_{yy}^{-T}r_{yd}^{-}$$

$$= E[d^2(n)] - \left(r_{yd}^{-}\right)^T \boldsymbol{h}^*$$

在前面的推导中，我们利用了$\boldsymbol{R}_{yy}$是对称的这一原理，即：$\boldsymbol{R}_{\mathrm{yy}}^{-\mathrm{T}} = \boldsymbol{R}_{yy}^{-1}$。

在语音增强的应用中，图 5-4 中的输入信号$y(n)$是含噪语音信号。

图 5-4 中的目标信号$d(n)$就是想要得到的干净语音信号$x(n)$，维纳滤波器的目标因此变成了估计干净语音信号$x(n)$。

我们可以使用公式（5. 7）推导出相应的时域上的维纳滤波器，为了评估时域上的维纳滤波器，我们第一步需要计算$\boldsymbol{R}_{\mathrm{yy}}$，向量方程可以由公式(5. 18)进行表示

$$\boldsymbol{R}_{yy} = \boldsymbol{E}[\boldsymbol{y}\boldsymbol{y}^{\boldsymbol{T}}] = \boldsymbol{E}[(\boldsymbol{x}+\boldsymbol{n})(\boldsymbol{x}+\boldsymbol{n})^{\boldsymbol{T}}] \tag{5. 18}$$

$$= \boldsymbol{E}[\mathrm{x}\mathrm{x}^{\boldsymbol{T}}] + \boldsymbol{E}[\mathrm{n}\mathrm{n}^{\boldsymbol{T}}] + \boldsymbol{E}[\mathrm{x}\mathrm{n}^{\boldsymbol{T}}] + \boldsymbol{E}[\mathrm{n}\mathrm{x}^{\boldsymbol{T}}]$$

$$= \boldsymbol{R}_{\mathrm{xx}} + \boldsymbol{R}_{\boldsymbol{nn}}$$

倒数第二个方程中的最后两个期望为零，因为假设信号和噪声不相关且均值为零。由于假设语音信号和噪声是无关的，等式(5. 13)中的互相关矩阵$\boldsymbol{r}_{\boldsymbol{yd}}^{-}$和$\boldsymbol{r}_{\mathbf{xx}}$相等。因此，时域中的维纳滤波器可以进一步表示成

$$\boldsymbol{h}^* = (\boldsymbol{R}_{\boldsymbol{xx}} + \boldsymbol{R}_{\boldsymbol{nn}})^{-\boldsymbol{1}}\boldsymbol{r}_{\boldsymbol{xx}} \tag{5. 19}$$

维纳滤波器h^*是干净语音信号$\boldsymbol{x}(\boldsymbol{n})$的自回归函数，因此它是不能实现的。我们可以对公式（5. 20）进行进一步优化

$$\boldsymbol{h}^* = \left[\frac{\mathbf{1}}{\mathrm{SNR}} + \widehat{\boldsymbol{R}}_{\mathrm{nn}}^{-\mathbf{1}}\widehat{\boldsymbol{R}}_{\boldsymbol{xx}}\right]^{-\mathbf{1}} \widehat{\boldsymbol{R}}_{\mathrm{nn}}^{-\mathbf{1}}\widehat{\boldsymbol{R}}_{\boldsymbol{xx}}\boldsymbol{u}_{\mathbf{1}} \tag{5. 20}$$

$$\mathrm{SNR} = \frac{\boldsymbol{E}\{\boldsymbol{x}^{\mathbf{2}}(\boldsymbol{n})\}}{\boldsymbol{E}\{\boldsymbol{n}^{\mathbf{2}}(\boldsymbol{n})\}} = \frac{\boldsymbol{\sigma}_{\boldsymbol{x}}^{\mathbf{2}}}{\boldsymbol{\sigma}_{\boldsymbol{n}}^{\mathbf{2}}} \tag{5. 21}$$

SNR 是信噪比的缩写，$\boldsymbol{I}$是$(M \times M)$的单位矩阵，$\boldsymbol{u}_{\boldsymbol{1}}^{\boldsymbol{T}} = [1,0,\cdots,0]$是形状为$(1 \times M)$的单位矩阵，$\widehat{\boldsymbol{R}}_{\boldsymbol{xx}} \triangleq \boldsymbol{R}_{\boldsymbol{xx}}/\sigma_x^2$，$\widehat{\boldsymbol{R}}_{\boldsymbol{nn}} \triangleq \boldsymbol{R}_{\boldsymbol{nn}}/\sigma_n^2$。由式（5. 21），我们可以写出对于大、小信噪比值时的维纳滤波器的渐近关系:

$$\lim_{SNR\to\infty} \boldsymbol{h}^* = \boldsymbol{u_1} \tag{5. 22}$$

$$\lim_{SNR\to 0} \boldsymbol{h}^* = \boldsymbol{0} \tag{5. 23}$$

5.2.2 基于深度学习的语音增强

随着深度学习模型和方法的不断发展，基于深度学习的语音增强方法突显出了很强的降噪能力。基于深度学习的语音增强方法依赖深层神经网络具有很强的非线性建模能力，在处理非稳态噪声时，往往能够得到更好的降噪性能，并减少由于传统语音增强方法对于信号的假设，导致的一些信号扭曲或引入的音乐噪声[1]。

在本小节中，我们首先介绍在频域中基于深度学习方法的语音增强，通过介绍学习目标及几个典型的深度学习模型来进一步展开介绍语音增强。随后，我们介绍在时域中基于深度学习方法的语音增强。最后，我们介绍目前语音增强结合其他任务的应用。

5.2.2.1 频域的语音增强

(a) 频域语音增强的整体流程

在现实中，只考虑加性噪声的情况下，语音信号可以表示为如下：

$$y(n) = x(n) + v(n) \tag{5. 24}$$

基于深度学习的语音增强，在信号处理的过程中，可以简单分成两类：一类是在频域，对语音信号特征进行增强；第二类则是在时域，直接对语音波形进行增强。

在频域进行语音增强时，首先需要将时域的语音信号转换成频域特征，常用的方法是短时傅里叶变换（Short Time Fourier Transform, STFT）[9]。并将复数域的短时傅里叶变换结果分别取模和相位，则将信号表示成两部分：语谱图的振幅和相位信息。

语谱图[10]的振幅是语音增强中最常用的一种特征，由于噪声的存在，干净语谱图的振幅和带噪语谱图的振幅往往相差很大。

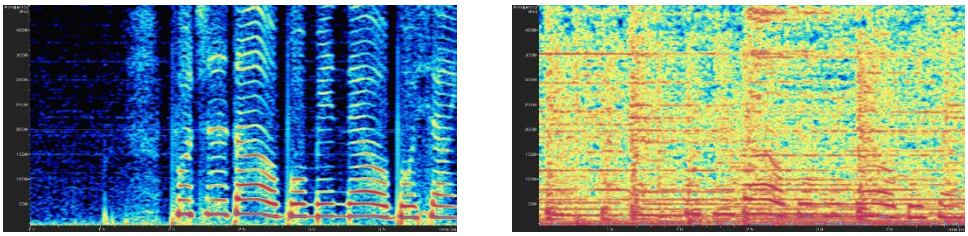


图 5-6.干净和带噪语谱图振幅的例子：横坐标代表时间，单位秒(s);纵坐标代表频率，单位赫兹(Hz)。左图是干净语谱图振幅，右图是带噪语谱图振幅。

语谱图是三维表示的，横轴是时间，纵轴是频率，而第三维表示幅度。幅度用亮色如红色表示高，用深色表示低。这也是可以将语谱图以二维图形呈现的原因。利用语谱图可以查看指定频率端的能量分布。

除了语谱图的振幅外，相位信息对语音增强效果也有很大的影响。目前已经存在一些网

络和技术来处理相位信息。但限于篇幅原因，本节主要关注利用深度学习方法增强语谱图的振幅，有关相位信息的说明涉及较少。频域语音增强的整体框架如下：

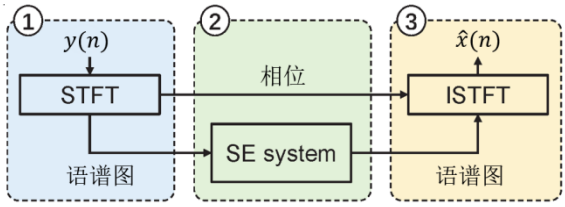


图 5-7. 频域语音增强的流程图：①特征提取；②特征增强；③波形还原。

频域语音增强分为三部分：①特征提取，利用短时傅里叶等变换，将时频信号转换为频域特征；②利用深度学习技术，增强语谱图的振幅；③将增强后的语谱图振幅，和带噪的相位信息，利用逆向短时傅里叶变换（Inverse Short Time Fourier Transform, ISTFT）还原成时域波形信号。

时域语音增强相较于频域语音增强，不需要进行特征提取，其特征便是时域波形信号。相较于频域语音增强，处理信号的思路更直观，而且，由于不需要处理相位信息等棘手的问题，受到越来越多的关注。

(b) 频域语音增强的学习目标

利用深度学习的语音增强是一个有监督任务，需要利用深层神经网络学习到适合的学习目标，从而达到语音增强的效果。在本节中，我们介绍了几个在基于深度学习的语音增强中常用的学习目标。其中主要分为两类：基于 masking 的目标(本小节的(1)-(5))和基于 mapping 的目标(本小节(6))。基于 masking 的目标描述了干净语音与其背景干扰的时频关系，而基于 mapping 的目标对应于干净语音的频谱表示。

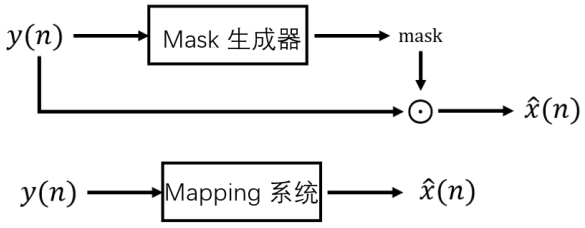


图 5-8. masking 目标和 mapping 目标的对比

(1) 理想二值掩蔽（Ideal Binary Mask, IBM）[11]

第一个训练目标是理想二值掩码，其灵感来自于听觉掩蔽现象和听觉场景分析中的排他分配原则。IBM 的定义是基于二维时频表示的噪声信号，如耳蜗图或语谱图：

$$IBM = \begin{cases} 1, & if\ SNR(t,f) > threshold \\ 0, & otherwise \end{cases} \tag{5.25}$$

式(5.25)中，t 和 f 分别表示时间和频率。当该时频单元（T-F bin）的信噪比（Signal-to-Noise Ratio, SNR）大于阈值（threshold）时，理想二值掩蔽被设置为 1；否则为 0。IBM 掩蔽技术显著提高了正常听力和听力受损的听众在噪声中的言语清晰度。IBM 将每个 T-F bin 标记为目标主导或干扰主导。因此，IBM 评估可以自然地视为一个监督分类问题。IBM 评估中一个常用的成本函数是交叉熵（Cross entropy, CE）。

(2) 目标二值掩蔽（Target Binary Mask, TBM）[11]

与 IBM 一样，目标二进制掩码使用二进制标签对所有 T-F 单元进行分类。与 IBM 不同的是，TBM 通过比较每个 T-F 单元的目标语音能量与一个固定的干扰:语音形状的噪声，这是一个与所有语音信号的平均值相对应的平稳信号，来获得标签。目标二值掩蔽也能显著提高噪声中的语音清晰度，TBM 已被用作训练目标。

(3) 理想比值掩蔽（Ideal Ratio Mask, IRM）[11]

理想比率掩码可以看作是 IBM 的软版本，而不是每个 T-F 单元上的硬标签：

$$IRM = \left(\frac{S(t,f)^2}{S(t,f)^2 + N(t,f)^2}\right)^{\beta} \tag{5. 26}$$

式(5. 26)中，$S(t,f)^2$和$N(t,f)^2$分别表示 T-F bin 的语音和噪声能量。可调参数 β 通常选择 0.5。在假设$S(t,f)^2$和$N(t,f)^2$不相关的情况下，有平方根的 IRM 保持每个 T-F bin 的语音能量。这一假设适用于加性噪声，但不适用于房间混响情况下的卷积干扰(不过，延迟混响可以合理地认为是不相关干扰)。式(5. 26)中的 IRM 类似于经典的维纳滤波器，它是功率谱中目标语音的最优估计。均方误差（Mean squared error, MSE）通常用作 IRM 估计时的损失函数。

(4) 相位敏感掩蔽（Phase-Sensitive Mask, PSM）[11]

相位敏感掩蔽引入了相位信息：

$$PSM(t,f) = \frac{|S(t,f)|}{|Y(t,f)|} cos\,\theta \tag{5. 27}$$

在 θ 表示在 T-F bin 的干净语音和带噪语音之间的相位差别。在 PSM 中包含相位差会导致更高的信噪比。

(5) 复值理想比值掩蔽（Complex Ideal Ratio Mask, cIRM）[11]

复值理想比值掩蔽是复数域上的一种理想掩蔽。与上述掩蔽不同的是，它可以从带噪的语音中直接重构出干净的语音：

$$X = cIRM * Y \tag{5. 28}$$

X和Y分别表示干净语音和带噪语音的 STFT 表示。$*$ 表示复数乘法。求解掩蔽分量，得到如下定义：

$$cIRM = \frac{Y_r S_r + Y_i S_i}{Y_r^2 + Y_i^2} + i\frac{Y_r S_i - Y_i S_r}{Y_r^2 + Y_i^2} \tag{5. 29}$$

式(5. 29)中，Y_r 和 Y_i 分别表示噪声语音 STFT 的实数分量和虚数分量，S_r 和 S_i 分别表示干净语音 STFT 的实数分量和虚数分量。虚数单位用“i”表示。因此，cIRM 有一个实数分量和虚数分量，可以在实数域中分别估计。

(6) 直接映射（Directly Mapping, DM）[11]

除了利用网络预测一个掩蔽外，可以直接利用网络来对语谱图的振幅进行增强。神经网络的输入输出都是语谱图的振幅。

$$\hat{\boldsymbol{x}}(\boldsymbol{n}) = f_\theta(\boldsymbol{y}(\boldsymbol{n})) \tag{5. 30}$$

其中f_θ是代表神经网络的函数。

有时，对语谱图的振幅采取取对数操作，这样能更符合人耳的听觉系统。

(c) 频域语音增强的模型

基于深度学习的频域语音增强模型，往往都是利用深层神经网络（Deep neural networks, DNNs）[12]的非线性映射能力，通过对网络输入特征，得到掩蔽或者增强后的特征。如无特殊强调，下列模型的输入都是语谱图振幅特征。目前基于深度学习的语音增强往往有如下特征：

(1) 输入采用拼帧处理

无论利用网络来预测掩蔽还是采用直接映射得到语谱图振幅，网络的输入往往都是采用拼帧处理：DNNs 能捕获到沿着时间轴上的声学内容信息（将带噪语音信号的多帧作为输入），并且沿着频率轴，将多帧连接成一个输入特征向量。拼帧处理能够为网络提供上下文信息，这往往有助于网络得到更好的增强特征。拼帧特征作为输入的一个例子如图 5-9 所示：

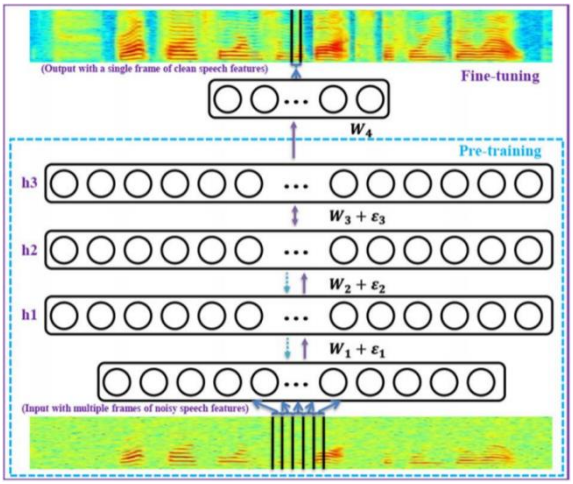


图 5-9. 拼帧处理作为输入[12]

(2) 训练用丰富噪声数据

神经网络往往能够对已经学习到的数据类型有较好的性能，而那些没有被学习到的数据类型往往性能较差。为了提高泛化能力，在 DNN 训练集的设计中需要加入更多种类的噪声类型，以进一步提高模型对于可见及不可见噪声类型，特别是非平稳噪声分量的降噪能力。图 5-10 显示了 104 种噪声的语谱图，每一种语谱图都存在不同的数据分布，利用存在多种数据分布的数据，能够让网络学习到更具有普适性和鲁棒性的增强效果。

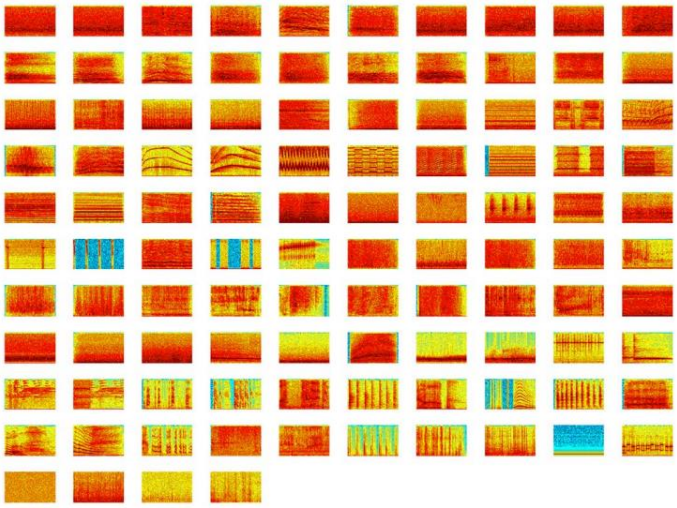


图 5-10. 以 104 种噪声的声谱图作为噪声信号，合成带噪声的语音训练样本[12]。

类似于使用多种噪声数据，采用多种的 SNR[13]也可以提高模型的普适性和鲁棒性。

(3) 结合其他特征进行多目标学习

在基于深度神经网络的语音增强中，通过引入一个辅助结构来学习次要的连续特征，如梅尔频率倒谱系数（Mel-frequency cepstral coefficients, MFCCs）[14]和分类信息，如理想二进制掩码（IBM），并将其集成到原始的 DNN 架构中，以联合优化所有参数。这种联合估计方案增加了直接预测语谱图振幅所没有的额外约束，并有可能提高主要目标的学习能力：

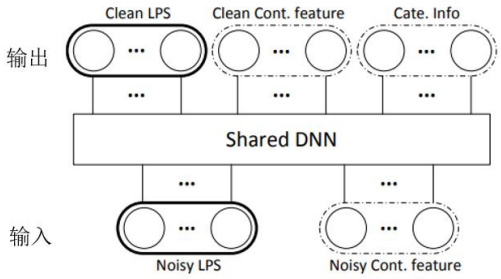


图 5-11. 基于多目标学习的语音增强示例

在目标函数中添加来自 MFCC 或 IBM 等特性的约束，可以获得更准确的语谱图振幅估计。多目标学习（Multi-target learning, MTL）[1]能够在多个特征有互补性时，利用这种互补性，得到更好的性能。

(4) 神经网络结构的不断改进

神经网络结构能够极大的影响语音增强的效果。多个受限玻尔兹曼机（Restricted Boltzmann machines, RBMs）[16]堆叠，通过逐层预训练，整体调优的训练方式，在较早基于深度学习的语音增强模型受到广泛的应用。

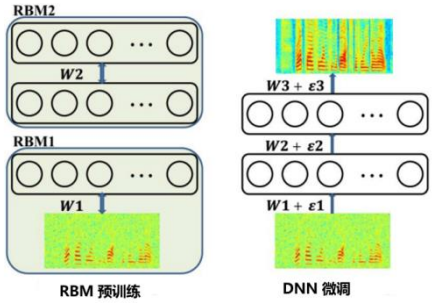


图 5-12. 基于受限玻尔兹曼机堆叠，逐层预训练并整体调优的语音增强示例

随着深度学习的不断发展，除了全连接神经网络（Fully connected neural network, FNN）具有更深的网络结构外。一些新的网络架构也被应用在语音增强任务中。卷积神经网络（Convolutional Neural Networks, CNNs）[17]便是其中一种。卷积神经网络通过引入卷积核来获取多尺度特征，并且，卷积神经网络相较于全连接神经网络，更容易将层级结构加深，这有助于模型获得更好的性能。下图是一个基于卷积神经网络的语音增强：

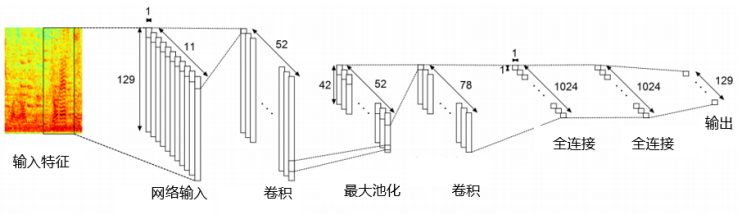


图 5-13. 基于卷积神经网络的语音增强示例

卷积神经网络还有一个比较大的优势是能够权值共享。这极大的减少了模型的权重数目，使得卷积神经网络相较于全连接神经网络，拥有更少的权重。如果将全连接层全部替换成卷积层的话，那么此时的卷积神经网络叫做全卷积神经网络（Fully Convolutional Neural Network, FCN）[17]。全卷积神经网络在语音增强的一个应用如下图所示：

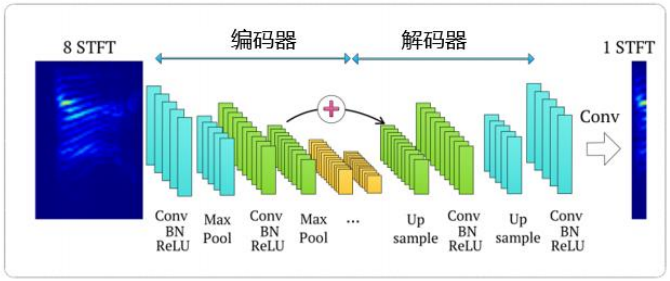


图 5-14.基于全卷积神经网络的语音增强示例

卷积神经网络也可以像全连接神经网络一样，利用拼帧的方式来获取上下文信息。但是这仅限于相邻帧之间的信息扩充，还无法处理长时声学信息的建模。循环神经网络（Recurrent neural networks, RNNs）[1]的提出便是为了解决这个问题。循环神经网络通过在前一帧和当前帧之间使用循环结构来捕获长期的上下文信息，从而更好地进行预测，从而缓解了这一问

题。由于循环神经网络算法的弊端在于，随着时间的流逝，网络层数的增多，会产生梯度消失或梯度爆炸等问题。长短时记忆循环网络（Long short-term memory, LSTM）[1]是循环神经网络中的一种，设计初衷是希望能够解决神经网络中的长期依赖问题，也在语音增强任务中受到了广泛的应用。长短时记忆神经网络的神经元如下图所示：

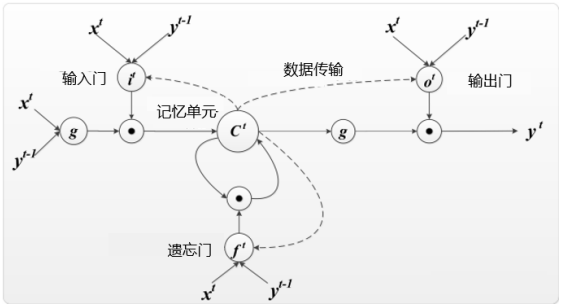


图 5-15. 长短时记忆网络神经元

长短时记忆循环网络的记忆单元具有遗忘门、输入门和输出门，正是这些门控结构的存在，使得长短时记忆循环网络的记忆单元拥有长短时记忆机制。卷积神经网络和循环神经网络的结合也往往可以得到更好的效果。卷积神经网络和长短时记忆网络神经元结合的语音增强示例如下图所示。

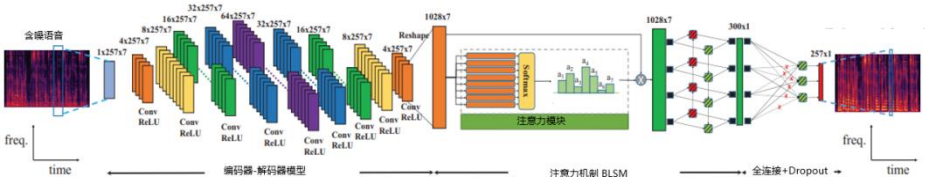


图 5-16. 卷积神经网络和长短时记忆网络神经元结合的语音增强示例

图 5-16 中应用了注意力机制（Attention Mechanism）[18]来对特征进行建模，得到相关性信息。在其他领域的一些新机制，也可以应用在语音增强领域并起到了正向作用。

生成对抗神经网络（Generative Adversarial Networks, GAN）[19]的提出能够进一步提升语音增强的性能。生成对抗神经网络包含两部分：生成器（Generator, G）和判别器（Discriminator, D）。生成器是用来生成增强后的语谱图振幅，而判别器则是用来判断生成器生成的语谱图振幅是否具有较优的性能。判别器的输出是一个 0 到 1 之间的值，值越接近 1，则表示生成器的效果越好；反之，则越差。对抗生成神经网络的语音增强示例如下图所示：

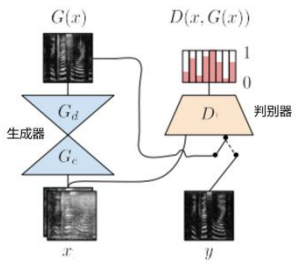


图 5-17. 对抗生成神经网络的语音增强示例

需要强调一点的是，对抗生成神经网络中的生成器和判别器，可以用卷积神经网络、循环神经网络等来实现。

(5) 额外声学信息[20]

在嘈杂的环境中，如果听话者很好地掌握了这门语言，他/她就可以自动地恢复丢失的语音信号。也就是说，有了“语言模型”的内在知识，听者可以有效地抑制噪声干扰，检索目标语音信号。因此，熟悉口语的潜在语言内容有助于在嘈杂的环境中增强语音。一个抽象的符号顺序建模方法被纳入到语音增强框架中。这种符号顺序建模可以看作是学习声学无噪声语音映射函数的一种“语言约束”。利用矢量量化变分自编码算法，将声信号的符号序列以离散形式表示。所获得的符号能够从语音信号中捕获高级音素类内容。一个向语言模型中加入声学信息的语音增强示例如下图所示:

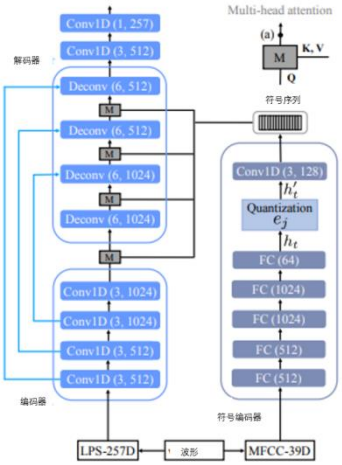


图 5-18. 向语言模型中加入声学信息的语音增强示例[20]

(6) 多任务学习

考虑将语音信息加入到增强系统中并进行互增强，也逐渐出现。在语音转换领域，通过语音后验图（PPGs）使用语音信息已经取得了重大进展。利用噪声的 PPG 来增强语音是一个不错的互增强尝试。由于 PPG 预测与语音增强相互促进，在语音增强系统中，引入了 PPG 预测器，并在训练系统时，将语音增强模块与 PPG 模块迭代训练。一个该方法的示例如下图所示:

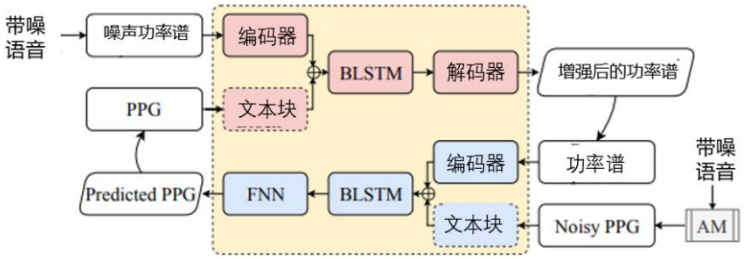


图 5-19. 语音增强与 PPG 预测互增强的语音增强示例

目前，基于深度学习的频域语音增强系统呈现的趋势如上面几点所示。增强的思路百花齐放，未来也会有更多的思路来处理语音，得到更好的语音增强效果。

(7) 相位的处理

近年来，除振幅预测外，相位预测也成为关注的焦点。相位-谐波感知的深度神经网络，通过一个双流（振幅流和相位流）网络，使两个流间能相互通信。通过这样的网络，能够处理相位，而不是单纯使用带噪的相位信息。

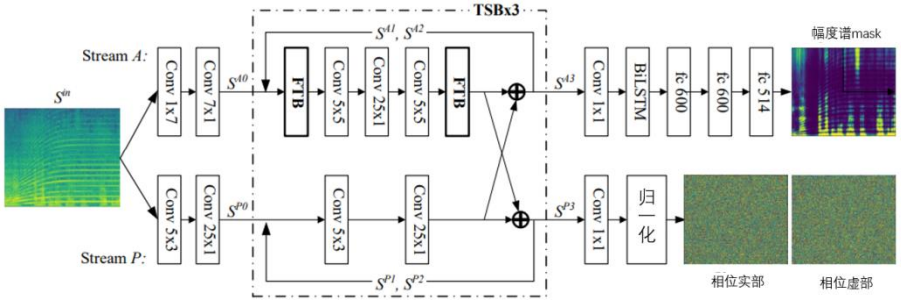


图 5-20. 相位和振幅增强的语音增强示例[22]

5.2.2.2 时域的语音增强

在频域的语音增强，往往受限于相位信息的影响，很难进一步提升性能。时域语音增强模型因为不需要提取特征，便可以利用深层神经网络，从时域信号得到增强后的时域信号，所以也被称为“端到端”语音增强模型。而随着深层神经网络模型在语音领域的应用，更多适合于在时域上对语音增强的模型被提出：

在较早的时期，利用全连接神经网络对时域语音信号进行增强时，随着全连接层的数量增加时，高频分量丢失的问题变得非常重要。这意味着，隐藏的全连通层实际上也有建模波形的困难。其原因可能是保持采样点之间在时域的关系以表示某一频率分量是至关重要的。而全连通层所映射的特征是抽象的，不保留原有特征的空间排列。换句话说，全连通层破坏了特征之间的相关性，使得生成波形变得困难。

此外，DNN 产生的高频分量也受数据输入方式的影响。通常情况下，波形是通过在有噪声的语音中滑动输入窗口来呈现给 DNN 的。

所以得益于 CNN 的进一步发展，相较于全连接神经网络，CNN 缺少高频分量的问题相对较小，因为 CNN 包含较少的全连接层。这些特性，使得在进行时域语音增强时，往往选择采用 CNN 而非全连接神经网络，全卷积神经网络（Fully convolutional neural network, FCN）便是其中一种[1]：

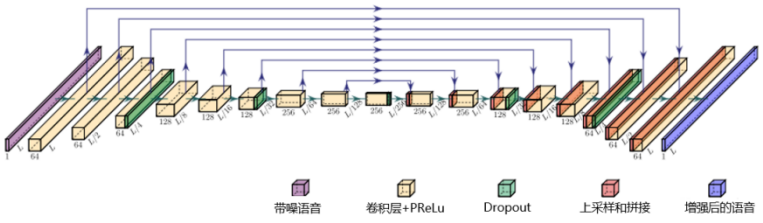


图 5-21. 基于全卷积神经网络的端到端语音增强示例。

在时域上进行语音增强，性能受到损失函数极大的影响。常见的损失函数有：时域的均方误差（Time-Domain Mean Square Error）、时域的平均绝对误差（Time-Domain Mean Absolute Error）、短时波幅均方误差（Short-Time Spectral Amplitude Mean Square Error）、短时目标清晰度（Short-Time Objective Intelligibility）、语音质量评价的感知度量（Perceptual Metric for Speech Quality Evaluation）。此外，时域语音增强也可以引入频域的损失，来对语音增强效果进行约束[23]:

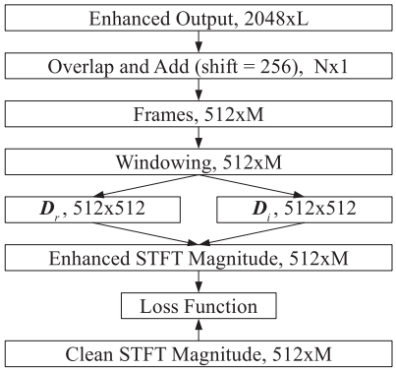


图 5-22. 时域语音增强引入频域损失函数的语音增强示例。

5.2.2.3 语音增强结合其他语音任务的应用

语音增强除了更符合人耳听觉外，还可以结合其他语音任务来使用。比较常见的有自动语音识别（Automatic Speech Recognition, ASR）、说话人识别（Speaker Recognition）等任务。

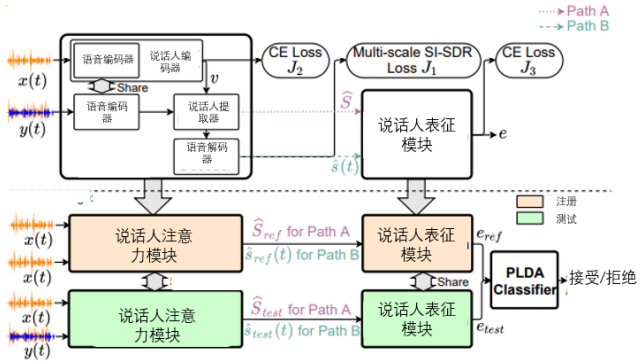


图 5- 23. 语音增强和说话人识别结合图示[40]

说话人识别是用于识别特定的说话人的语音信号，根据不同的说话人的情况，只针对特定的说话人进行识别。将说话人识别和语音增强相结合，可以用于解决混杂人声的去噪问题。

图 5- 23 为语音增强和说话人识别结合的任务。此模型主要分为三个部分，说话人注意模块，说话人表征模块，说话人验证模块。其中的说话人注意模块就是一个典型的时域语音增强模块，从混合的信号中提取出目标的语音信号，此网络是通过多目标学习的方式来优化的。其语音表征模块则是从增强得到的语音信号中提取适用于后端说话人验证的语音特征。最后得到的特征输入到说话人验证模块，验证提取的说话人是否为合法身份。

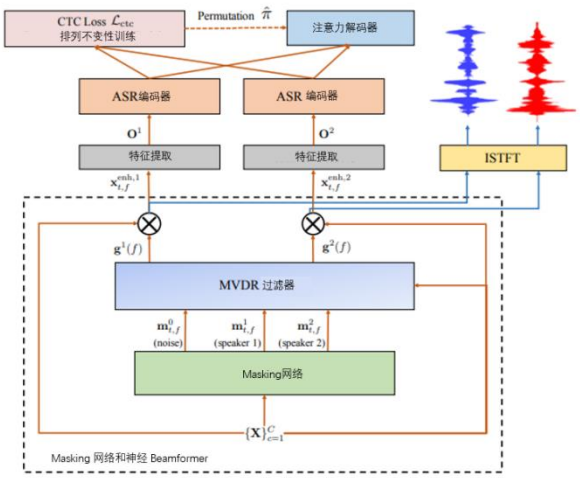


图 5- 24. 语音增强和语音识别结合实例[41]

语音识别指将语音转化为文字的过程，在强噪声，多干扰的环境下，对语音识别的准确率有很大的挑战。因此前端语音增强降噪的过程就显得很重要。

图 5- 24 代表和语音识别结合的任务。此图为端到端的多通道语音识别流程图。此模型主要分为两个部分，语音增强网络和语音识别网络。其中，语音增强部分，先通过基于掩蔽的网络（Masking Network）估计掩蔽值（Mask），然后通过基于掩蔽的 MVDR 波束方法得到增强后的语音特征。其后端的语音识别网络，则将前端增强得到的语音特征，进一步提取语音识别相关的特征，用基于连接时序分类准则（CTC）和基于注意力（Attention）的联合损失函数进行网络优化。整个模型是端到端模型，增强网络和语音识别网络是联合优化的。

近年来，随着语音合成（Text-to-Speech, TTS）任务的广泛研究，语音增强任务和语音合成任务的结合也变得越来越重要。

虽然使用干净的语音构建的语音仍然是首选，但是得到很多标注的干净数据费时费力。但是，如果从带噪的语音构建的文本到语音的语音质量会受到损害。在训练前使用语音增强来增强语音数据已被证明可以提高语音合成的质量。有两种方法可以达到增强的效果：

训练一个递归神经网络，从有噪声的语音提取的声学特征映射到描述干净语音的特征，然后利用增强后的数据训练 TTS 声学模型。

按照传统的语音增强方法，我们只用从幅度谱中提取的梅尔谱系数训练一个神经网络。将增强的梅尔谱特征与从噪声语音中提取的相位相结合，重构出波形，然后利用波形提取声学特征训练 TTS 系统。

两种方式增强模型的流程图如下图所示：

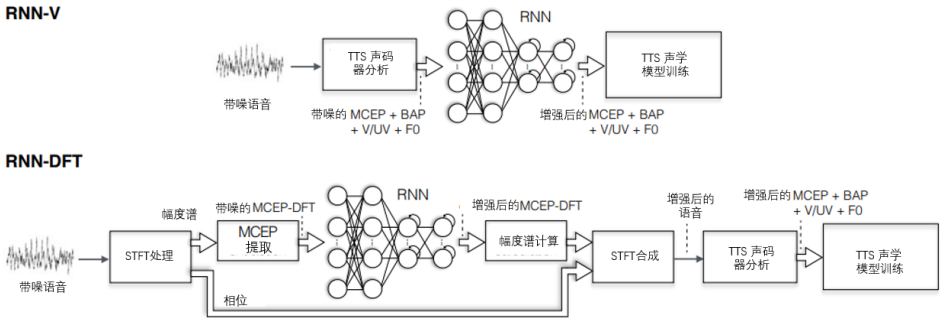


图 5-25. 结合语音合成的语音增强的示例[24]

此外，语音增强也可以应用在低资源下语音合成时，合成效果较差的情况下。将合成的语音，通过语音增强系统，得到更高质量的语音信号。

5.3 基于麦克风阵列的语音增强

前面所述的语音增强方法，多是针对单通道语音。而对于多通道语音(即使用麦克风阵列录制的音频，图 5-26)，与之对应的增强方法中，较为出名的便是波束形成(Beamforming)技术。基于麦克风阵列的语音增强有其特有的优势，主要在于考虑了声源的位置信息和通道音频之间的信息差距，故而对于具有方向性的噪声有较好的抑制作用。基于麦克风阵列的语音增强在实际应用中大多针对方向性的干扰语音。

波束形成技术成熟较早，在早期该技术主要应用于雷达等定位系统当中。波束形成的基本思想是对各个麦克风单元接收到的语音信号进行加权求和，该加权系数决定了语音增强的效果。通过调整加权系数，达到对目标声源方向的增强以及对其他干扰方向的抑制的效果，使得整个麦克风阵列对目标方向形成聚焦。根据加权系数是否能够随着接收信号的统计特性进行自我调整，可以将波束形成方法划分为固定波束形成(Fixed Beamforming)和自适应波束形成(Adaptive Beamforming)[1]。固定波束形成指的是一旦设计好滤波器后，加权系数就不再改变。该方法的优点是计算简单，容易实现；缺点也较为明显，主要表现在环境适应性差，对噪声不具有自适应性。而在自适应波束形成方法中，加权系数具有自我调节的能力，故而实际应用范围非常广泛。下面将按照固定波束形成和自适应波束形成两个类型，分别详细介绍波束形成技术的具体实施方案。

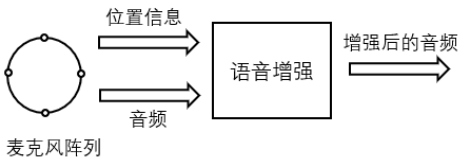


图 5-26. 基于麦克风阵列的语音增强

5.3.1 固定波束形成

在正式介绍波束形成技术之前，先了解一下麦克风阵列的相关知识。麦克风阵列指的是

由 2 个以上的麦克风按照一定的几何形状排列成的一个整体系统。根据麦克风阵列的拓扑形状，可以将麦克风阵列大致分为一维阵列，二维阵列和三维立体阵列。一维阵列最常见的是均匀线性阵列(见图 5-27 ，其中 L 为阵列长度，d 为相邻麦克风间距)；二维阵列最常见的是均匀圆形阵列(见图 5-26，其中 R 为圆形麦克风阵列的半径，d 为实际的相邻麦克风直线间距)[1]。实际工程中使用最多的便是上述两种阵列。研究表明，阵列的拓扑结构对语音增强的性能会产生很大的影响。

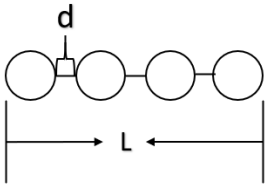


图 5-27. 均匀线性麦克风阵列

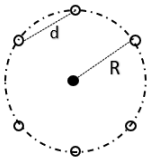


图 5-28. 均匀圆形麦克风阵列

了解了麦克风阵列的拓扑结构知识后，还需要知道声源方向信息的表达方式。在本文中统一描述为方向公式。对于一个在三维空间中传播的平面波而言，可以统一表达为：

$$a(r,k,t) = Acos(k \cdot r - \omega t + \varphi) \tag{5. 31}$$

其中$\boldsymbol{r}$表示三维空间位置，$\boldsymbol{k}$为波矢量，表示平面波的传播方向以及波形随距离变化快慢的程度。t表示时间，$\omega = 2\pi f$为角频率，A表示振幅，φ为初始相位。为了形式简便，可以将上式变成复数形式：

$$a(r,k,t) = Re\{Ae^{j\varphi}e^{jk\cdot r}e^{-j\omega t}\} \tag{5. 32}$$

若令$B = Ae^{j\varphi}$，则

$$a(r,k,t) = Re\{Be^{jk\cdot r}e^{-j\omega t}\} \tag{5. 33}$$

假设麦克风阵列由M个麦克风组成，每个麦克风各有其空间位置$\vec{r_i}$，t时刻整个麦克风阵列的输出总和表示为：

$$b(k) = \sum_{i=1}^{M} w_i B_i\, e^{jk\cdot r} \tag{5. 34}$$

其中，w_i是第i个麦克风阵元的加权系数，B_i是语音信号在第i个麦克风处的强度，并且上述表达式省略了$e^{-j\omega t}$。对上述表达式取模，即可得到麦克风阵列的方向表达式。该表达式的物理意义是麦克风阵列的输出强度对于波达方向(DOA: Direction Of Arrival)以及波长的依赖关系。

下面将详细介绍经典的固定波束形成方法，主要为延迟-求和波束形成(DSB: Delay and Sum Beamforming)[1]。延迟-求和波束形成方法是最早的应用方法，该方法十分具有代表性。除此之外，子阵列波束形成(SAB: Sub-Array Beamforming)[1]是对延迟-求和波束形成方法的改进，也会简单科普。

5.3.1.1 延迟-求和波束形成

延迟-求和波束形成方法是最简单的多通道语音增强波束形成方法。最早由美国学者

Flanagan 于 1985 年提出。DSB 方法首先对各个麦克风接收到的信号进行时间延迟补偿，由于声源方向和麦克风阵列之间存在一个接收角度，故而每个麦克风接收到信号的时间是不完全相同的。通过时间补偿，使得各个麦克风信号在时间上对齐，然后进行加权求和，得到增强的输出信号。该算法实现简单，算法复杂度不高，具有一定的去噪效果。但是需要数量较多的麦克风才能达到预期的噪音抑制效果，并且对于环境和噪音的变化适应性较差。

下面是关于延迟-求和波束形成方法的系统流程。现假设麦克风阵列接收到的信号表示为：

$$x_i(n) = \alpha_i * s(n - \tau_i) + v_i(n) \qquad i = 1,2, \ldots, \tag{5. 35}$$

其中 M 表示麦克风的个数，$x_i(n)$、α_i、$s_i(n)$和$v_i(n)$分别表示接收到的音频信号，声波传播衰减系数(小于 1)，声源发出的音频信号以及加性噪音。其中*操作表示卷积。τ_i表示时延。

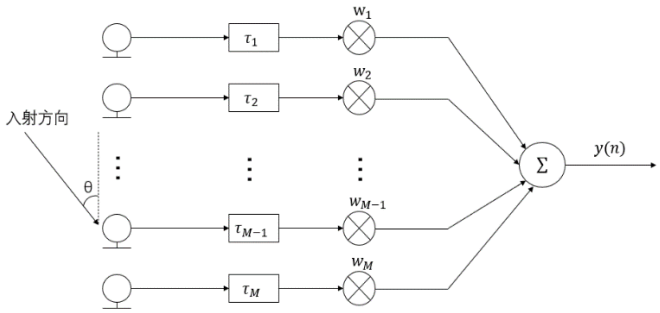


图 5-29.　DSB 算法流程

由 DSB 算法流程图可知，经过该算法增强后的输出为：

$$y(n) = \sum_{i=1}^{M} w_i x_i(n + \tau_i) \tag{5. 36}$$

其中τ_i表示第i 个麦克风信号的时间延迟，w_i是波束形成算法对第i 个麦克风的加权系数，由于该算法为固定系数，故而每个系数的值都固定相同。

下面针对线性均匀阵列和圆形阵列，详细描述 DSB 的算法流程。

均匀线性阵列

假设麦克风间距绝对均匀，间距为 d，声源的入射角度为θ，并且声源和麦克风阵列处于同一高度，即不存在俯仰角，只有平面方向角。

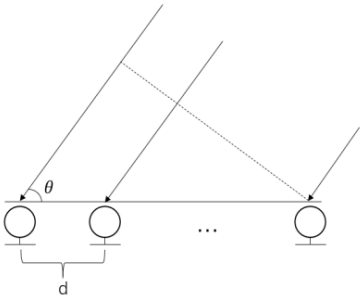


图 5-30.　DSB 之均匀线性阵列

在已知平面波的方向角和俯仰角的前提下，假设分别为θ和ξ，则波矢量的表达公式为：

$$k = \frac{2\pi}{\lambda}(cos\xi cos\theta, cos\xi sin\theta, sin\xi) \tag{5. 37}$$

当麦克风阵列和说话人位于同一平面的前提下，ξ为0°，则将上式代入取模的方向公式，有：

$$b(k) = B\left|\sum_{i=1}^{M} w_i\, e^{j\frac{2\pi}{\lambda}(i-1)dcos\theta}\right| \tag{5. 38}$$

此处以第一个麦克风为参考，麦克风的相对位置信息可以表示为$r_i = (i-1)d$，而每个麦克风处的强度是一样的，统一表示为B。

假设期望的声源方向为α，将加权系数w_i设为与阵元位置成正比的相位延迟，则w_i的表达式可以写成：

$$w_i = e^{-j\frac{2\pi}{\lambda}(i-1)dcos\alpha} \tag{5. 39}$$

带入上式则变成，

$$b(k) = B\left|\sum_{i=1}^{M} e^{j\frac{2\pi}{\lambda}(i-1)d(cos\theta-cos\alpha)}\right| \tag{5. 40}$$

该公式的含义是，当声源的真实方向与预期的方向相同时，加权系数对信号的延迟补偿恰好等于信号在预期方向上由方向角带来的相位延迟。此时信号得到增强。现在回到式(5. 36),此时只要将期望方向上的加权系数带入该式，即为 DSB 算法增强后的输出：

$$y(n) = \sum_{i=1}^{M} e^{-j\frac{2\pi}{\lambda}(i-1)dcos\alpha} x_i(n) \tag{5. 41}$$

均匀圆形阵列

不同于均匀线性阵列，均匀圆形阵列是存在俯仰角的。假设一个圆形均匀阵列水平置于空间中，波达方向 DOA 与该阵列的空间关系如图 5-31 所示，此外，将 DOA 垂直投影到该阵列上，其平面图如图 5-30 所示。

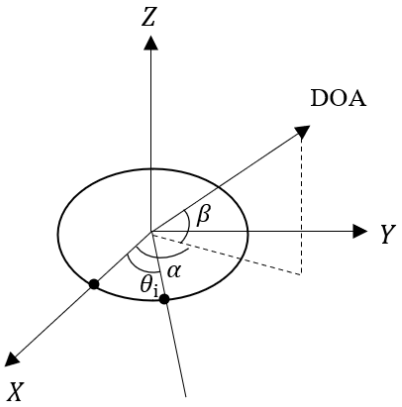


图 5-31. 均匀圆形阵列三维图

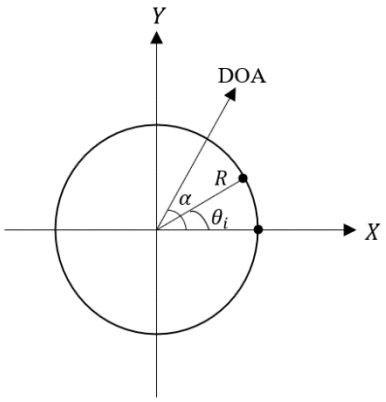


图 5-32. 均匀圆形阵列二维图

圆形阵列的半径为R，相邻麦克风的夹角为θ_i，波达方向的方位角为α，俯仰角为β。第i个麦克风的位置可以表示为：

$$\boldsymbol{r_i} = \begin{bmatrix} Rcos\theta_i \\ Rsin\theta_i \\ 0 \end{bmatrix} \tag{5.42}$$

相邻麦克风的夹角为θ_i可由半径和麦克风个数推出。所以波矢量和位置的乘积相比线性阵列发生了很大的变化，由于波矢量的维度和位置信息的维度满足矩阵相乘规则，能够得到一个 1*1 维的元素，该元素同时满足三角公式，故而最终化简为：

$$k \cdot \boldsymbol{r_i} = \frac{2\pi R cos\beta}{\lambda} cos\ (\alpha - \theta_i) \tag{5.43}$$

将该式带入取模的方向公式，并再次假设每个麦克风强度相同，皆为B：

$$b(k) = B \left| \sum_{i=1}^{M} w_i\, e^{j\frac{2\pi R cos\beta}{\lambda} cos\ (\alpha - \theta_i)} \right| \tag{5.44}$$

该方法在俯仰角的计算上精度较低，故而一般不会用 DSB 组合圆形阵列去探求俯仰角。再次的，设期望的方向角为γ，俯仰角仍然为β(常设为 0)。类比线性阵列，加权系数可以表示为：

$$w_i = e^{-j\frac{2\pi R cos\beta}{\lambda} cos\ (\gamma - \theta_i)} \tag{5.45}$$

类比线性阵列，可以将上式带入方向公式，从而将加权系数具体化。再次回到式(5.36)，此时只要将期望方向上的加权系数带入该式，即为 DSB 算法增强后的输出：

$$y(n) = \sum_{i=1}^{M} e^{-j\frac{2\pi R cos\beta}{\lambda} cos\ (\gamma - \theta_i)} x_i(n) \tag{5.46}$$

以上讨论了 DSB 对于两种阵列的计算方式。DSB 产生的波束方向图能够显示目标语音的具体方向，方向图中最大辐射波束叫做主瓣，主瓣旁边的小波束叫做旁瓣。方向图通常都有多个瓣，其中辐射强度最大的瓣称为主瓣，其余的瓣称为副瓣或旁瓣。通常主瓣所指的方

向即为声源方向。图 5-33 为使用自适应波束形成方法(MVDR)和固定波束形成方法产生的方向图，其中横坐标代表角度，纵坐标代表平均功率(计算方法将在下节介绍) 。这里主瓣越尖锐，表明方向计算的越准确。在同样的音频中，使用 MVDR 方法得到的主瓣尖锐程度要好于 DSB 方法

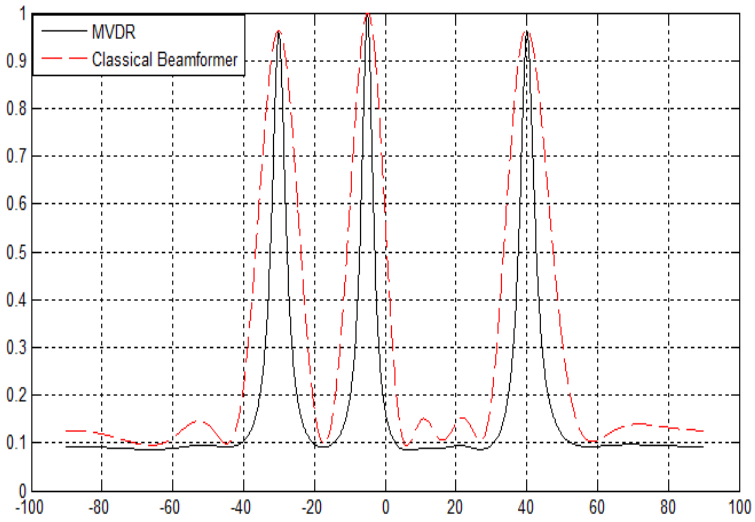


图 5-33. 方向图的主瓣与旁瓣

5.3.1.2 子阵列波束形成

子阵列波束形成是对延迟-求和波束形成的补充和发展，我们对于该部分的介绍偏向科普与说明。将不再进行详细的流程阐释以及公式科普，详细说明请参考[38]。下面将简述该算法相比于原始算法的改进之处。

DSB 形成的方向图具有一些特性。例如随着音频频率的变化，主瓣有着不同的表现。通常来讲频率低时主瓣将变宽，频率高时主瓣将变窄。较窄的主瓣对于方向信息有着精确的反馈，而较宽的主瓣不是我们所希望得到的。针对该缺点，有学者提出了子阵列波束形成方法(SAB)。该方法将信号从频谱上划分出多个区间，每个区间使用不同的麦克风阵列，区别主要是间距不同，从而每个区间都能获得近似的主瓣。最后将各个阵列的信号滤波后累加求和，获得输出信号。这样做的好处是尽可能抵消低频信号的主瓣变宽所带来的影响。由于不可能无限制的使用麦克风阵列，整个系统的麦克风采用重叠的方式，即每个麦克风可能从属于多个麦克风子阵列。

除了子阵列波束形成算法，类似的固定波束形成算法还有滤波求和波束形成算法(FSB: Filter-Sum Beamforming)[1]等，他们都是对固定波束形形成算法的探索。随着波束形成应用范围的不断扩大，固定波束形成的方法已经无法满足大部分应用需求；与此同时，得益于加权系数能够随着信号的特性进行自我调整，自适应波束形成受到越来越广泛的关注。

5.3.2 自适应波束形成

相较于固定波束形成，自适应波束形成具有较强的加权系数自我调节的特性。目前常见的几种自适应方法有最小方差无畸变响应(MVDR: Minimum Variance Distortionless Response)、线性约束最小方差(LCVM: Linear Constrained Minimum Variance)以及广义旁瓣抵消法(GSC: Generalized Sidelobe Canceller)等[1]。目前自适应波束形成已经被广泛地应用于麦克风阵列语音增强，下面将详细介绍上述几种常见方法的原理。

5.3.2.1 MVDR 波束形成

MVDR 波束形成是基于最大信干噪比(Signal-to-interference-plus-noise，SINR)准则的自适应波束形成算法。该算法可以自适应的使麦克风阵列输出在期望的方向上功率最小同时信干噪比最大，以抑制噪音[25]。MVDR 在军事领域有着比较广泛的应用，例如常被用于水声无线通信技术，可以实现水面舰艇和潜艇之间的通信等等。MVDR 算法采用了自适应波束形成中常用的采样矩阵求逆算法，该算法在信干噪比下具有较快的收敛速度。对于式(5. 35)，表示为多通道形式：

$$x(n) = \boldsymbol{\alpha} s(n - \tau) + v(n) \tag{5. 47}$$

对上式进行傅里叶变换，将时域信号转为频域信号：

$$X(\omega) = \boldsymbol{\alpha} S(\omega) + V(\omega) \tag{5. 48}$$

其中$\boldsymbol{\alpha}$表示阵列的方向矢量，反应了麦克风阵列对方向的敏感程度，该矢量是有具体值的，该参数的值为：

$$\boldsymbol{\alpha} = [\alpha_1 e^{-j\omega\tau_1}, \alpha_2 e^{-j\omega\tau_2}, \dots, \alpha_M e^{-j\omega\tau_M}] \tag{5. 49}$$

其中τ指时延，α_i是和第 i 个通道相关的传播衰减系数。有时候会将$\boldsymbol{\alpha X}(\omega)$使用$X(\omega)$统一表达，在波束形成加权系数已知的情况下，增强后的信号可以表示为：

$$\boldsymbol{Y(\omega)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{M}} \boldsymbol{W_i(\omega) X_i(\omega)} = \boldsymbol{W^H X} \tag{5. 50}$$

其中$W_i(\omega)$为第i个麦克风在频率ω的加权系数。而 MVDR 波束形成的关键则是计算信号的功率谱密度矩阵(PSD: Power Spectrum Density)，整体输出信号的 PSD 矩阵为：

$$\boldsymbol{\phi} = E(\boldsymbol{YY^H}) = \boldsymbol{W^H} E(\boldsymbol{XX^H}) \boldsymbol{W} \tag{5. 51}$$

其中$\boldsymbol{Y}$和$\boldsymbol{X}$分别表示麦克风阵列的增强信号和原始输出信号。$\boldsymbol{W}$表示加权系数，$\boldsymbol{\phi}$表示 PSD 矩阵，$\boldsymbol{H}$表示共轭转置。在继续之前，先了解一下功率谱密度的计算。

对于频域的一个信号$\boldsymbol{S}(\omega)$，我们一般将式(5. 52)称为信号$\boldsymbol{S}(\omega)$的 PSD，这里假设噪声和信号之间不相关：

$$\boldsymbol{\phi_{SS}} = \boldsymbol{E[S(\omega) S(\omega)]} \tag{5. 52}$$

故而对于接收信号、原始信号和噪音，三者之间满足：

$$\boldsymbol{\phi_{XX}} = \boldsymbol{\phi_{SS}} + \boldsymbol{\phi_{VV}} \tag{5. 53}$$

输入和输出信号的 PSD 矩阵决定着信号的信干噪比。MVDR 方法就是使得输出信号的功率最小，来获得最优加权系数的估计。而输出功率谱由上式决定，在最优化过程中要避免使得加权系数变为 0，即保证信号在期望的方向上没有失真：

$$\boldsymbol{W^H \alpha = 1} \tag{5. 54}$$

其中$\boldsymbol{\alpha}$即阵列的方向矢量，在该约束条件下求解最优问题，即在上式的情况下，求式(5. 51)中加权系数最小：

$$\boldsymbol{min_w W^H \phi_{XX} W \quad s.t.\ W^H \alpha = 1} \tag{5.55}$$

这样，经过求解约束优化问题，可以得到 MVDR 波束形成的自适应加权系数为：

$$\boldsymbol{W_{MVDR} = \frac{\phi_{VV}^{-1}\alpha}{\alpha^H \phi_{VV}^{-1}\alpha}} \tag{5.56}$$

其中$\boldsymbol{V}$表示噪音。在实际的阵列处理中，我们无法得到统计意义上理想的 PSD 矩阵，因此只能使用最大似然来代替。

在实际的阵列处理中，噪音等干扰成分往往不能从阵列输出中分离出来，这在一定程度上限制了算法的应用。在期望信号与噪声加干扰完全不相干时，用包含期望信号的 PSD 矩阵进行估计所得的权系数与理想情况下的最优权系数相同。所以在实际使用时，常常用包含期望信号的 PSD 矩阵$\boldsymbol{\phi_{XX}}$来代替。从 MVDR 的参数矢量表达式中我们可以看出，该参数可以根据噪音等干扰的 PSD 矩阵的变化而变化，因而 MVDR 算法可以自适应的使麦克风阵列输出在期望方向上的 SINR 最大，达到最佳效果。

而经过一系列的推导，前文提到的平均功率我们现在也可以给出其具体的计算方法了：

$$\mathbf{P} = \frac{1}{\boldsymbol{\alpha^H \Phi_{XX} \alpha}} \tag{5.57}$$

当麦克风阵列中阵元数下降，或者高信噪比的环境下，期望信号与噪音和干扰往往存在明显的相干性，这在很大程度上影响了 MVDR 算法的性能。在散射噪声场中，高频部分噪声相关性弱，低频部分噪声相关性强，MVDR 在不同的频率段往往会发生效果区别较大的情况。

5.3.2.2 LCMV 波束形成

LCMV 波束形成算法可以实时地对信号进行处理，同时又可以抑制噪声干扰。该算法也可以对麦克风阵列地加权滤波系数不断进行迭代更新，使得噪声输出功率最小的同时保持对期望方向上的频响不变。本质上该算法类似于 MVDR，是约束条件下的最小均方算法(LMS: Least Mean Square)。该方法需要预先给定一个固定的频响。

对于输出信号，每个麦克风在完成延迟补偿后的波形是一样的，对于M个麦克风，假设每个麦克风置于一个K阶的滤波器下，则整个系统的加权系数共有KM个。这些系数的选择应当满足每个麦克风的滤波效果。换句话说，需要满足K个线性约束条件，这K个约束将所求信号的功率输出保持一个定量值；而剩余$KM-K$个系数，要满足输出的总功率最小。该系统见图 5-34[25].

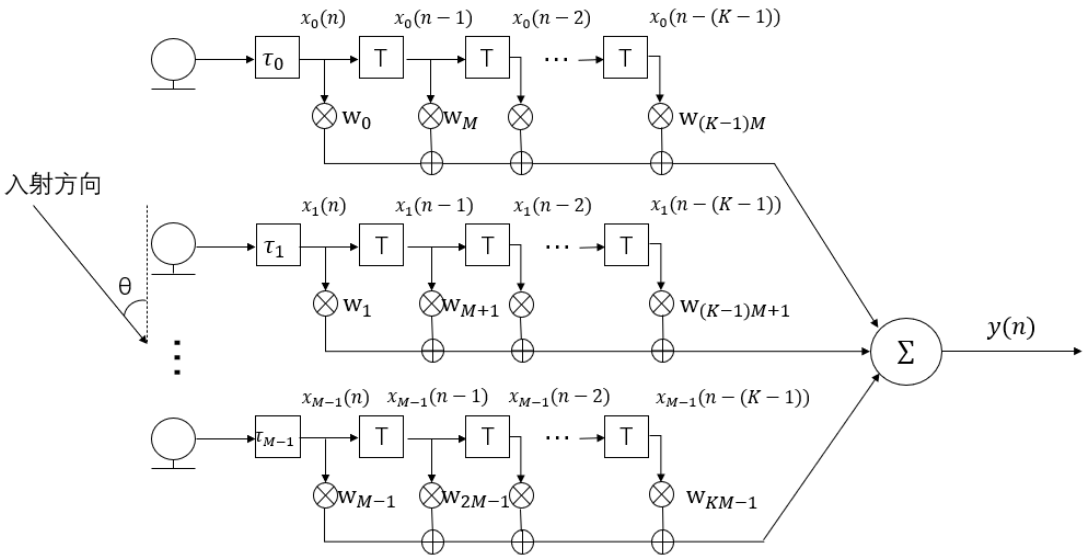


图 5-34.　LCMV 波束形成算法图

按照图示理论，我们可以将麦克风接收信号的公式重新表达，如下所示：

$$\boldsymbol{x(n)} = [x_0(n), x_1(n), \dots, x_{M-1}(n-(K-1))]^{\boldsymbol{T}} \tag{5.58}$$

$$\boldsymbol{s(n)} = [s(n), \dots, s(n), \dots, s\big(n-(K-1)\big), \dots, s\big(n-(K-1)\big)]^{\boldsymbol{T}} \tag{5.59}$$

$$\boldsymbol{v(n)} = [v_0(n), v_1(n), \dots, v_{M-1}(n-(K-1))]^{\boldsymbol{T}} \tag{5.60}$$

这里做一些假设，假设所求信号和噪音干扰的均值皆为 0，期望方向上的信号与其他方向上的噪音干扰互不相关，即二者的期望值为 0：

$$\boldsymbol{E[s(n)v^T(n)] = 0} \tag{5.61}$$

最后输出信号可以表示为：

$$\boldsymbol{y(n)} = \boldsymbol{w^T x(n)} = \boldsymbol{x^T(n)w} \tag{5.62}$$

其中w表示滤波系数矢量，共有KM个，组成了系数一维向量。由于噪音和信号不相关，将式(5. 62)带入求解输出功率公式中，输出功率可以化简为：

$$\boldsymbol{E[y^2(n)] = E[w^T x(n) x^T(n) w]} = \boldsymbol{w^T R_{xx}\, w} \tag{5.63}$$

其中，$\boldsymbol{x(n)x^T(n)}$的计算结果正是接收信号x的互相关矩阵$\boldsymbol{R_{xx}}$。约束条件中要求麦克风阵列中延时相同的权系数之和满足事先给定的滤波系数：

$$\boldsymbol{c_k^T w} = \boldsymbol{f_k}, \quad \boldsymbol{k = 0, 1, \dots, K-1} \tag{5.64}$$

向量$\boldsymbol{c}$是KM维度的，对应下标为k时，此时第k组初始化为m个 1，其他位置对应皆为 0。将向量$\boldsymbol{c}$组成约束矩阵：

$$\boldsymbol{C_{MK\times K} = [c_0, c_1, \dots, c_{K-1}]} \tag{5.65}$$

而滤波系数的初始值也将组成一个向量：

$$\boldsymbol{F} = [\boldsymbol{f_0}, \boldsymbol{f_1}, \cdots, \boldsymbol{f_{K-1}}]^{\boldsymbol{T}} \tag{5. 66}$$

故而，式(5. 64)可以表示为向量形式，将该约束条件重新表示为：

$$\boldsymbol{C^T w} = \boldsymbol{F} \tag{5. 67}$$

对于无失真约束条件，我们一般将滤波系数初始化为f_0为 1，其余的滤波系数为 0。该方式为最常见的初始化方式。

麦克风阵列所期望的声源方位上的频响由K个约束条件所确定，故而输出信号的功率也是确定的。为了减少其他方向的噪音和干扰，就要求全部输出功率取得最小值。即目标是使得式(5. 63)得到最小值。这样，问题彻底变成 LMS 带约束条件的优化问题。整体可以表示为：

$$\boldsymbol{min_w w^T R_{xx}\, w} \quad \boldsymbol{s.t.\, C^T w = F} \tag{5. 68}$$

该优化问题可以根据拉格朗日乘子法来求解，该方法可以将含约束条件的优化问题转化为无约束条件的优化问题。在此我们不再详细说明求解的具体过程。感兴趣的同学可以自行查阅相关资料进行求解。

根据拉格朗日乘子法，可以得到 LCMV 的最佳滤波系数矩阵为：

$$\boldsymbol{w_{LCMV}} = \boldsymbol{R_{xx}^{-1}[C^T R_{xx}^{-1} C]^{-1} F} \tag{5. 69}$$

在 LCMV 的实际应用中由于不可避免的存在麦克风阵列位置误差、阵元之间的相位误差及方向误差等，常常对性能产生较大的影响。

5.3.2.3 GSC 波束形成

GSC 波束形成相比前述方法，其流程较为复杂。GSC 是从 LCMV 过渡而来，是更一般的情况，LCMV 可以看作是 GSC 的一个特例。作为一种通用模型，该算法主要由图 5-35 三部分组成，分别是固定波束形成部分(延迟相加波束形成单元)、自适应噪声抵消部分(ANC: Adaptive Noise Canceller)以及在自适应噪声抵消前增加了一个信号阻塞矩阵(BM: Block Matrix)[25]。该方法提出的目的便是为了改进 ANC 部分，以减少信号泄露。通过引入一个主通道和辅助噪声通道，将约束条件从系统中分离。

固定波束形成部分只让特定方向的信号通过，但是肯定会残留一部分其他非期望方向的噪声信号，而阻塞矩阵的作用是阻止特定方向的信号通过，让其他方向的信号通过。那么通过自适应地调整滤波参数的权重，使得阻塞矩阵的输出近似于固定波束形成输出中残留的噪音部分，就可以得到较为纯净的语音信号估计。GSC 结构将约束求解转化为无约束问题[26]。

固定波束形成单元的输出为：

$$\boldsymbol{y_f(n)} = \boldsymbol{w_f^T x(n)} \tag{5. 70}$$

这里w为滤波系数矢量，是由M个加权系数组成。简单起见，设该矢量满足：

$$\boldsymbol{w_f^T \cdot 1 = 1} \tag{5. 71}$$

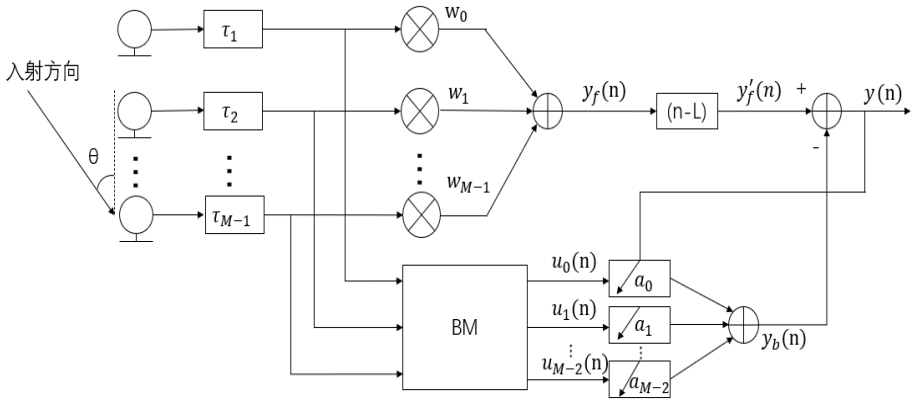


图 5-35.　GSC 波束形成算法图[26]

为了补偿 BM 矩阵和自适应滤波的时间，使用延迟单元衔接在$y_f(n)$后，此时$y'_f(n) = y_f(n-L)$。在 BM 部分，其输出只含噪声，经过该模块后，得到的信号为：

$$\boldsymbol{u(n)} = \boldsymbol{W_b^T x(n)} \tag{5. 72}$$

对于该加权矩阵的特性，第m行的加权系数满足：

$$\boldsymbol{w_b^T \cdot 1 = 0} \tag{5. 73}$$

而$\boldsymbol{w_b}$是线性独立的，所以$\boldsymbol{u}(n)$中至多含有$M-1$个元素。之后经过自适应滤波器组后，得到输出信号$y_b(n)$。$y_b(n)$是$\boldsymbol{u}(n)$经过系数矩阵$\boldsymbol{a}$后得到的。最后经过相减，系统最后的输出为：

$$\boldsymbol{y(n)} = \boldsymbol{y'_f(n)} - \boldsymbol{y_b(n)} \tag{5. 74}$$

故而寻找滤波系数$\boldsymbol{a_i}(n)$使得系统输出的噪声功率最小。在该模式下，问题变为无约束求解。最后，该系数自适应迭代公式为：

$$\boldsymbol{a_i(n+1)} = \boldsymbol{a_i(n)} + \boldsymbol{\mu y(n) u_i(n)} \tag{5. 75}$$

该算法创造性的引入 BM 矩阵对噪音的加权系数进行求解，不过当非平稳的噪音与目标信号方向接近时，该算法性能收到很大的影响。上述章节分别对固定波束形成和自适应波束形成相关算法的过程进行了较为完整的描述。除了上述两类方法外，比较常见的还有后置滤波算法，该类算法对所有的噪音(相干和非相干)都有一定的抑制能力，这里就不再详细介绍。另外，随着深度学习的广泛应用，众多学者们也把目光转移到波束形成和 DNN 的结合上，所得出的增强结果相比传统方法更优秀。感兴趣的同学可以自行查找相关书籍或论文[39]，相信你们会取得不错的收获。

5.4 语音增强技术的展望

近年来，随着智能设备广泛进入日常生活方方面面，复杂声学场景下的语音交互应用成为关注的热点，语音增强技术也逐渐成为现实智能设备和平台的必备环节和关键入口。受益于大数据与深度学习技术的迅猛发展，语音增强技术已逐渐从原来的基于规则和信号处理的

方法逐渐过渡到数据驱动的深度学习方法。截至目前，尽管语音增强技术突飞猛进，在某些公开数据集上取得了不错的成果，但是相比于人类处理复杂声学场景的表现，依旧有很大的差距。可以预见，如何设计一个高效、有效并且易用的语音增强系统势必仍然是未来非常受瞩目的一个方向。针对复杂声学场景下的语音信号处理的机制和计算模型，目前尚有很多值得探索的问题和方向，以下具体探讨几点语音增强技术的展望。

5.4.1 语音增强和语音分离的结合

在本章语音增强技术的内容中，主要侧重介绍了单个说话人的语音增强技术，然而在现实日常生活中，目标说话人的语音信号，除了夹杂着背景噪声、混响等环境噪声外，可能存在其他无关说话人的声音。在这种真实场景下，目标语音信号综合了所有可能的干扰，其提取难度可想而知，这也就是著名的“鸡尾酒会问题”。如图 5-36 所示，除了目标语音信号（黑色实线），其他的所有信号都是干扰信号，这个难度就比单说话人时候的声学场景复杂许多。显然，现实日常生活中的多说话人声学场景更加复杂，也就说明更难去除干扰噪声，提升语音的质量和可懂度。因此，此时单纯地用语音增强技术来解决问题就会比较困难。目前可行有效的方案是通过结合语音分离技术，来分离出干净的目标语音信号提升语音质量和可懂度，便于提升后续的语音识别等语音应用的识别精度。

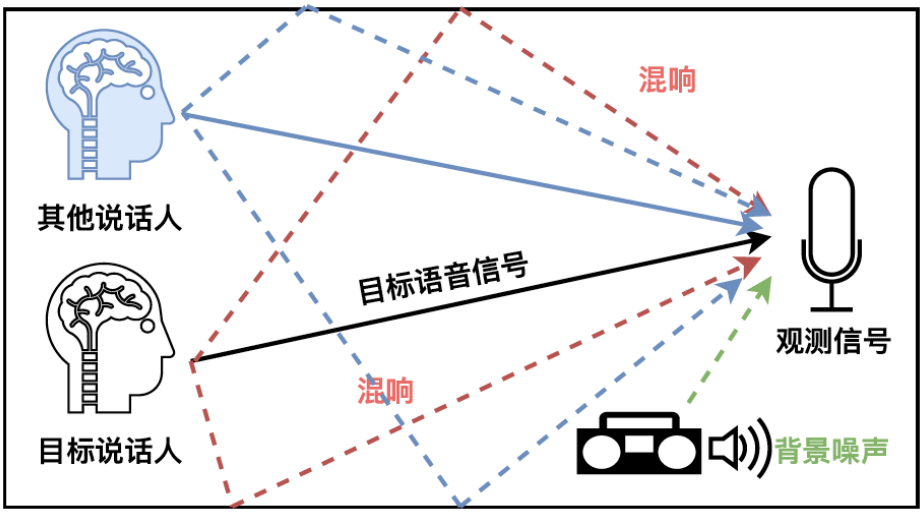


图 5-36. 多说话人的现实复杂声学场景（鸡尾酒会问题）

针对如此复杂的声学场景，解决方案的思路一般是：预增强去噪-语音分离-后增强去混响。假设此声学场景下有M个说话人（包含目标说话人），每个人说话产生的语音信号为$x_i(n)\ (i=1,\dots,M)$，每个人在房间中的房间冲击响应函数是$h_i(n)\ (i=1,\dots,M)$。同时，我们将x_1作为目标说话人产生的目标语音信号，背景噪声假设为符号v，最终收录语音的设备是单个麦克风，则此麦克风最后收集到的观测信号可以定义为：

$$\boldsymbol{y(n)}=\sum_{\boldsymbol{i}}^{\boldsymbol{M}}(x_i(n)*h_i(n))+\boldsymbol{v(n)} \tag{5.76}$$

因而，这个复杂声学场景下的语音增强问题，就转变为已知观测信号 y 求解目标语音信号x_1数学问题。从数学公式中，我们就能很好地理解如何解决这个问题。首先，我们可以通过语音增强技术预先处理比较简单的加性噪声$v(n)$，然后我们可以通过语音分离（或说话人

分离）来提取目标说话人的部分$x_1 * h_i$，最后再次训练一个语音增强模型去去除混响干扰，获得期望的目标说话人语音信号x_i。因而，我们会发现通过结合语音增强技术和语音分离技术能较好地解决现实的复杂声学场景的降噪问题。

语音分离的目标是从干扰信号中分离出目标语音信号。从定义上分析，语音增强也可以通过语音分离的方式处理，即将背景噪声和混响等作为干扰源，然后分离出目标语音信号。下面就单通道语音分离技术做简单叙述。

现阶段，单通道语音分离技术主要分为频域和时域的单通道语音分离方法。如表 5-2 所示，单通道语音方法主要有三部分组成，分别是编码器、分离器和解码器。混合语音信号通过编码、分离，最后解码重构目标语音信号。基于频域的单通道语音分离方法通常是通过短时傅里叶变换（STFT）和短时逆傅立叶变换（ISTFT）作为编码器和解码器，常使用双向长短时记忆网络（BLSTM）来做分离器，损失函数多使用最小均方误差。典型的频域单通道语音分离方法有深度聚类模型（DPCL）[1]、置换不变模型（PIT）[1]、计算听觉场景分析模型（CASA）和说话人提取模型（VoiceFilter[1]和 SpEx[1]）。基于时域的单通道语音分离方法则常用 1 维卷积操作来作为编码器和解码器，常用卷积网络（CNN）和双向长短时记忆网络（BLSTM）来做分离器，实际使用中用卷积网络较多。此类方法尝试用信噪比（Si-SNR）来作为目标函数进行训练，典型的方法主要是基于 TasNet 方法的分离方法，包括 BLSTM-TasNet、Conv-TasNet 和 DPRNN-TasNet 模型。在公开数据集 WSJ0-2mix 数据集上，基于时域的分离方法的性能要远优于基于频域的分离方法。

表 5-2. 单通道语音分离方法的通用方案

方法类别	输入特征	编码器/解码器	分离器	损失函数
频域方法	时频谱特征	STFT/iSTFT	BLSTM	最小均方误差（MSE）
时域方法	时域波形	1 维卷积	CNN/ BLSTM	信噪比（Si-SNR）

5.4.2 语音增强和脑科学研究的结合

语音增强技术的研究启发于人类在复杂听觉环境下的一种听觉选择能力。在受到其他说话人或者噪声干扰的情况下，人类总能很容易地将注意力集中于目标声音并忽略其他无关的背景声音，从而准确理解目标说话人所表达的意图。其中听觉注意机制起到了重要作用，有相关实验研究发现，人类不可能听到或者记住两个同时发生的语音信号。然而，如图 5-37 所示，人类却可以精准地从被混合的复杂语音中选择出其注意到的语音信号，以及同时忽略掉其他语音或者噪音等背景音。因而，研究人类注意并理解目标语音信号这一过程背后的逻辑，能很好地启发计算模型过滤出目标语音信号，建模出高效易用的智能机器。人类的听觉系统是一个高度非线性的系统，神经回路中神经元之间的连接非常复杂，神经元对刺激采用多种编码方式，主要有频率编码，时间编码和群体编码这三种方式。声音中富有丰富的时空结构，而听觉系统对这些时空结构是高度敏感的。然而，现阶段的基于深度学习的语音增强技术，对语音的编码方式较为单一，无法充分挖掘利用语音中的时空结构信息。同时，目前在类脑计算中应用较广的脉冲神经网络性能还远不如常用的人工神经网络，存在较大的性能差距。

因而，如何结合脑科学研究有效编码语音并高效过滤目标语音将是未来值得探究的热点问题。

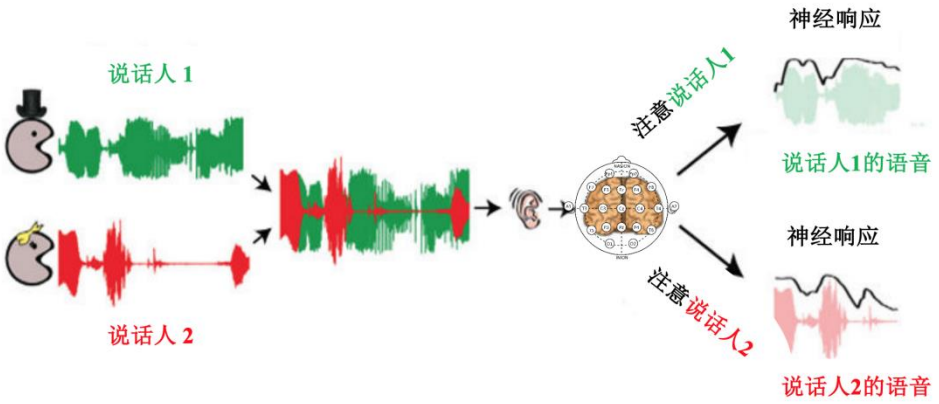


图 5-37. 听觉系统的注意力选择机制

听觉系统处理外界刺激一般可以分为自下而上的刺激驱动的过程和自上而下的任务驱动的过程。自下而上的处理过程是指从输入的刺激进行处理，继而完成相应的任务。自上而下的处理过程是指在高层的抽象概念或信息的指引下完成特定的任务，其过程通常涉及长期记忆和学习机制。现阶段，少许研究尝试结合自上而下的任务驱动机制来提升语音增强或语音分离技术。例如，图 5-38 所示 [27]，这类方法常用皮层脑电信号（ECoG）或者是目标说话人的语音信息作为先验信息，进而辅助语音增强或语音分离网络更好地提取目标语音信号，从而抑制噪声提升语音质量和可懂度。然后，皮层脑电信号的采集需要将电极插入到受试者的皮层，数据采集成本较高。因而，如何使用成本较低的脑电信号（例如 EEG 信号）来重构和增强目标语音信号将会是未来的研究热点。此项研究的发展也将帮助构音障碍者或聋哑患者带来言语表达的新世界。

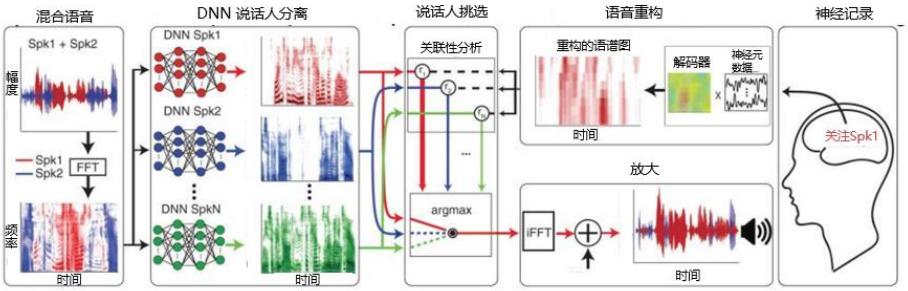


图 5-38. 多说话人环境下注意力选择的神经解码[32]

5.4.3 多模态语音增强技术

仅仅使用听觉模态的信息，往往存在难以区分目标语音和噪声类似的情况，比如同性别说话人的声音等。近几年，基于多感知整合等理论，语音增强技术开始将视觉模态信息整合到语音信息中，在一定程度上解决了存在类似噪声干扰带来的问题。视听结合的多模态语音

增强方案利用听觉信息和视觉信息在时间上的高度相关性，可以进行自监督学习，从而可以做到无序标记数据。尽管现阶段尚未知晓多感知整合是人脑处理时的哪个阶段，但是可以确定确实是存在的。因此，未来随着数据采集技术的不断提升，多模态语音增强技术也将带来新的突破，必将带来满意的增强效果。

目前比较经典的是谷歌团队在 2018 年发表的《Looking to Listen at the Cocktail Party》这个工作 [28]。如图 5-39 所示 [28]，论文提出了一种基于深度学习的音频——视频结合的模型，将单个目标语音信号从背景噪声、其他人声等混合声音中分离出来。提出的模型通过计算生成视频，增强其中目标说话人的语音，同时抑制其他说话人的声音。此方法用在具有单个音频轨道的普通视频上，用户需要做的就是在视频中选出他们想要听到的说话者的面部，或者结合语境用算法选出这样的人。此方法用途广泛，从视频中的语音增强和识别、视频会议，到改进助听器，不一而足，尤其适用于有多个说话人的情景。这项技术的独特之处在于结合了输入视频的听觉和视觉信号来分离语音。直观地讲，人嘴部的运动应当与这个人说话时产生的声音相关联，这反过来又可以帮助识别音频的哪些部分对应于这个人。视觉信号不仅可以在混合语音的情况下显著提高语音分离质量（与仅仅使用音频的语音分离相比，与我们的论文得出的结论相同），但是重要的是，它还能够将分离的干净语音轨道与视频中的可见说话者相关联。此方法可以作为预处理程序应用于语音识别和自动视频字幕添加。处理语音重叠的说话者对于自动字幕添加系统来说很有挑战性，将音频分离为不同的来源可以帮助生成更加准确、易读的字幕。

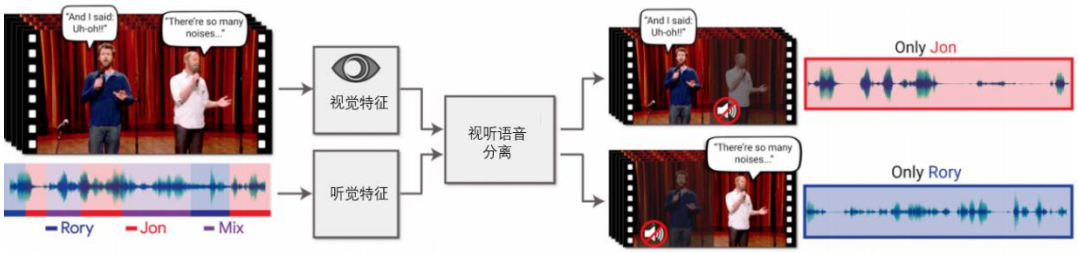


图 5-39. 基于深度学习的音频——视频结合的模型

5.4.4 实时在线语音增强技术

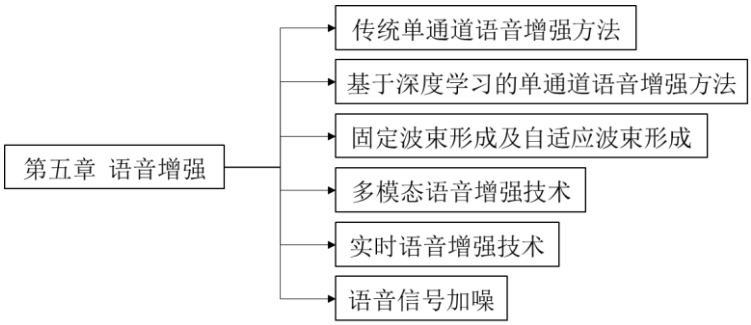
实时语音增强的研究就是针对这种实际的强噪声环境，希望通过算法在软硬件系统平台上的实时实现，尽可能地降低背景噪声，提高语音的质量，从而达到语音增强的效果。针对现实的语音交互应用，实时性是重中之重的考虑因素。在一些特殊场合的应用中如人工耳蜗、军事通讯设备及车载通信系统等等，这些应用和设备对语音增强系统的实时性要求更加严格。然而，现阶段的语音增强方案，特别是基于深度学习的语音增强技术方案，存在模型复杂度高、计算资源要求高等缺点，实时性问题也同样面临挑战。因此，实时性作为现实语音交互系统的产品落地的前提之一，同时也是语音增强技术研究的实用价值体现之一，如何设计实现复杂度低、资源利用高的高可用实时语音增强系统是未来的研究重点。

5.5 小结

本章介绍语音增强技术。语音增强在许多噪声环境下的语音应用中都被当作一个语音信号前端处理技术而被应用。基于传统信号处理的语音增强方法由于存在数学假设，且在处理非稳态噪声时会有极差的效果，而渐渐被基于深度学习的语音增强所替代。基于深度学习的语音增强方法可以简单分为频域语音增强模型和时域语音增强模型。近些年，语音增强也会结合其他语音任务实现彼此增强的效果。在单通道语音增强方法的基础上，我们介绍了多通道语音增强的方法。相比于单通道语音增强方法，多通道语音增强方法会将多个麦克风接受信号的时延等信息考虑进来，从而提升语音增强的效果。我们详细介绍了延迟-求和波束形成和子阵列波束形成的固定波束形成，以及 MVDR、LCMV、GSC 波束形成的自适应波束形成方法。语音增强和语音分离的结合、语音增强和脑科学研究的结合、多模态语音增强技术以及实时在线语音增强技术被当作语音增强技术的展望。最后，我们给出了相关实验。我们 A. 介绍怎样向语音信号中加入加性噪声和混响；B. 基于 MATLAB 的传统语音增强方法的实现；C. 基于 PYTHON 等编程语言的基于深度神经网络的语音增强方法实践；D. 基于 Wave-U-Net 的语音增强实践。

到目前为止，虽然语音增强技术已经取得了很大的进步，但是仍有很多不足，例如如何使用真实数据来训练得到一个较好的语音增强系统。仍需要很多研究者继续设计相关实验，不断推动语音增强的发展。

本章的知识点：



5.6 语音增强实践

5.6.1 加噪声、混响

混响和噪声开源数据库：http://www.openslr.org/28/

数据库说明：

A. 混响：

●Real RIRs：真实 RIRs 的集合由三个数据库组成：RWCP 声音场景数据库，REVERB 挑战数据库和亚琛脉冲响应数据库。总共有 325 个真实 RIRs。

●Simulated RIRs：对房间参数和房间中的接收器位置进行采样，然后根据不同的扬声器位置随机生成许多 RIRs。房间参数包括房间尺寸（宽度 w，长度 l 和高度 h）和吸收系数 1。

根据房间的宽度和长度取样的范围将 Simulated RIRs 分为三组：

①Ssmall（小型房间）：范围为 1m 至 10m。

②Smed （中号房间）：范围为 10m 至 30m。

③Slarge（大型房间）：范围为 30m 至 50m。

在这三组中，均对房间高度进行均匀采样，范围为 2m 到 5m；吸收系数从[0.2，0.8]均匀采样。在每组中，根据扬声器和接收器的位置，对 200 个房间进行采样，并在每个房间中采样 100 个 RIRs。

B. 噪声：

数据库中使用各向同性和点源噪声。真实 RIRs 数据库中可用的各向同性噪声与关联的 RIR 一起使用。点源噪声是从 MUSAN 语料库的 Freesound 部分采样的。语料库包含 843 个噪声记录，并且每个噪声记录都被手动分类为前景噪声或背景噪声。在指定的时间将前景噪声添加到混响的语音信号中，而通过重复将背景噪声扩展为覆盖整个语音记录。点源噪声的相加率决定了每次语音记录添加的点源噪声的总数。

加噪声、混响开源代码：https://github.com/linan2/add_reverb2

具体操作步骤：

●步骤一：下载并准备数据

●步骤二：运行 run.sh

5.6.2 语音增强 MATLAB 算法实践

开源代码：

https://github.com/lyapple2008/SpeechEnhancement/tree/ff56135baa205d37a67715afc14f77820434721a/Book_SpeechEnhancement_MATLAB/statistical_based

实验配置：matlab2016a

文件说明:

表 5-3 MATLAB 算法文件名、介绍以及参考文献

文件名	算法	文献
wiener_iter.m	基于全极点语音产生模型的迭代维纳算法	[29]
wiener_as.m	基于先验信噪比估计的维纳算法	[30]
wiener_wt.m	基于小波阈值多窗谱的维纳算法	[31]
mt_mask.m	基于心理声学的算法	[32]
audnoise.m	可听（Audible）噪声抑制算法	[33]
mmse.m	采用或未采用语音存在不确定度的 MMSE 算法	[34]
logmmse.m	对数 MMSE 算法	[35]
logmmse_SPU.m	结合语音存在不确定度的对数 MMSE 算法	[36]
stsa_weuclid.m	基于加权欧式失真测度的贝叶斯估计器	[37]
stsa_wcosh.m	基于加权 Cosh 失真测度的贝叶斯估计器	[37]
stsa_wlr.m	基于加权似然比失真测度的贝叶斯估计器	[37]
stsa_mis.m	基于修正 Itakura-Saito 失真测度的贝叶斯估计器	[37]
sp02_train_sn5.wav sp04_babble_sn10.wav sp06_babble_sn5.wav sp09_babble_sn10.wav	带噪语音文件	

具体操作步骤（例如：运行谱减算法）：

●步骤一：将代码下载到本地硬盘（例如 D 盘）。

●步骤二：运行 MATLAB。

●步骤三：通过在 MATLAB 中键入以下命令切换到包含谱减算法的文件夹：

→cd d:\speech_enhancement_MATLAB\spectra_subtractive

●步骤四：选择一个算法并执行。

例如，使用对数 MMSE 算法增强带噪文件 sp02_train_sn5.wav，键入 MATLAB 命令：logmmse(sp02_train_sn5.wav,out_sp02.wav)，其中 out_sp02.wav 为增强后的信号文件。

其余算法命令：

>> wiener_iter(infile.wav,outfile.wav,NumberOfIterations)

where 'NumberOfIterations' is the number of iterations involved in iterative Wiener filtering.

>> wiener_as(infile.wav,outfile.wav)

>> wiener_wt(infile.wav,outfile.wav)

>> mt_mask(infile.wav,outfile.wav)

>> audnoise(infile.wav,outfile.wav)

Runs 2 iterations (iter_num=2) of the algorithm.

>> mmse(infile.wav,outfile.wav,SPU)

where SPU=1 - includes speech presence uncertainty

SPU=0 - does not includes speech presence uncertainty

>> logmmse_SPU(infile.wav,outfile.wav,option)

where option=

1 - hard decision (Soon et al)

2 - soft decision (Soon et al.)

3 - Malah et al.(1999)

4 - Cohen (2002)

>> stsa_weuclid(infile.wav,outfile.wav,p)

where p>-2

>> stsa_wcosh(infile.wav,outfile.wav,p)

where p>-1

>> stsa_wlr(infile.wav,outfile.wav);

>> stsa_mis(infile.wav,outfile.wav);

5.6.3 基于深度神经网络进行语音增强实践

开源代码（pytorch 版本）:

https://github.com/yongxuUSTC/sednn/tree/master/mixture2clean_dnn_pytorch

环境配置：pytorch 1.0.0.

文件说明:

表 5-4 文件名及相应说明

文件名	说明
tmp01.py	基础 DNN 网络模型
tmp01b.py	DNN + mse on log scaled
tmp01c.py	DNN + L1 loss on magnitude + wiener filter
tmp01d.py	tmp01c.py + Batch Normalization
tmp01e.py	tmp01c.py + 学习率为 1e-5
tmp01f.py	tmp01c.py + 2048 个隐藏单元
tmp02.py	DenseNet 网络模型
tmp02b.py	DenseNet + 6 layers
tmp03.py	tmp01c.py + 信噪比为 5db,0db,-5db
tmp03b.py	tmp01c.py + 信噪比为[-10, 10]db
tmp04.py	tmp01c.py + loss on samples
prepare_data.py	数据处理
runme.sh	运行文件
evaluate.py	计算增强语音的 PESQ

具体操作步骤：

●步骤一：使用 mini data 运行 ./runme.sh

●步骤二：下载并准备数据

●步骤三：在./runme.sh 文件中修改 MINIDATA=0，并修改 WORKSPACE，TR_SPEECH_DIR，TR_NOISE_DIR，TE_SPEECH_DIR，TE_NOISE_DIR 的路径。

●步骤四：在自己的数据集上运行./runme.sh。

●步骤五：在 inference step 中可以添加--visualize，实现可视化，绘制混合，干净和增强后的语音对数频谱图。

5.6.4 基于 Wave-U-Net 的语音增强实践

开源代码（pytorch 版本）:

https://github.com/haoxiangsnr/Wave-U-Net-for-Speech-Enhancement

环境配置：python 3.6.5 ， Pytorch 1.2.0

具体操作步骤:

●步骤一：配置环境

①确保将 CUDA 的/bin 目录添加到 PATH 环境变量中；通过附加 LD_LIBRARY_PATH 环境变量来安装 CUDA 附带的 CUPTI

export PATH="/usr/local/cuda-10.0/bin:$PATH"

export LD_LIBRARY_PATH="/usr/local/cuda-10.0/lib64:$LD_LIBRARY_PATH"

②安装 Anaconda，以清华镜像源和 python 3.6.5 为例

Wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-5.2.0-Linux-x86_64.sh

chmod a+x Anaconda3-5.2.0-Linux-x86_64.sh

③创建一个虚拟环境

conda create -n wave-u-net python=3

conda activate wave-u-net

④安装依赖环境

conda install pytorch torchvision cudatoolkit=10.0 -c pytorch

conda install tensorflow-gpu

conda install matplotlib

pip install tqdm librosa pystoi pesq

git clone https://github.com/haoxiangsnr/Wave-U-Net-for-Speech-Enhancement.git

● 步骤二：使用 train.py 训练模型

python train.py [-h] -C CONFIG [-R]

参数说明：-h，显示帮助信息

-C，--config，指定训练所需的配置文件

-R，--resume，从最后保存的模型的检查点继续训练

● 步骤三：使用 enhancement.py 增强带噪的语音

python enhancement.py [-h] -C CONFIG [-D DEVICE] -O OUTPUT_DIR -M MODEL_CHECKPOINT_PATH

参数说明：-h，-help，显示帮助信息

-C，-config，指定模型，增强的数据集和用于增强语音的自定义参数。

-D，--device，增强使用的 GPU 索引，-1 表示使用 CPU

-O，--output_dir，指定将增强语音存储的位置，您需要确保该目录预先存在

-M，--model_checkpoint_path，模型检查点的路径，检查点文件的扩展名为.tar 或.pth

● 步骤四：在训练过程中，可以使用 tensorboard 启动静态前端服务器以可视化相关目录中的日志数据

tensorboard --logdir config["root_dir"]/<config_filename>/

注：训练期间生成的所有日志信息都将存储在 config [“ root_dir”] / <config_filename> / 目录中。 假设用于训练的配置文件为 config / train / sample_16366.json，sample_16366.json 中 root_dir 参数的值为/ home / UNet /。 然后，在当前实验训练过程中生成的日志将存储在 / home / UNet / sample_16366 /目录中。

参考文献

[1] Ya-Ting H, Jing S, Jia-Ming X, et al. Research advances and perspectives on the cocktail party problem and related auditory models[J]. Acta Automatica Sinica, 2019, 45(2): 234-251.

[2] Preminger J E, Tasell D J V. Quantifying the relation between speech quality and speech intelligibility[J]. Journal of Speech, Language, and Hearing Research, 1995, 38(3): 714-725.

[3] Kim G, Lu Y, Hu Y, et al. An algorithm that improves speech intelligibility in noise for normal-hearing listeners[J]. The Journal of the Acoustical Society of America, 2009, 126(3): 1486-1494.

[4] Speech dereverberation[M]. Springer Science & Business Media, 2010.

[5] Rix A W, Beerends J G, Hollier M P, et al. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs[C]//2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221). IEEE, 2001, 2: 749-752.

[6] Taal C H, Hendriks R C, Heusdens R, et al. An algorithm for intelligibility prediction of time–frequency weighted noisy speech[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2011, 19(7): 2125-2136.

[7] Vincent E, Gribonval R, Févotte C. Performance measurement in blind audio source separation[J]. IEEE transactions on audio, speech, and language processing, 2006, 14(4): 1462-1469.

[8] Hu Y, Loizou P C. Evaluation of objective quality measures for speech enhancement[J]. IEEE Transactions on audio, speech, and language processing, 2007, 16(1): 229-238.

[9] Grin D, Jae Lim. Signal estimation from modified short-time Fourier transform [C]. In ICASSP ’83. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1983: 804–807.

[10] Greenberg S, Kingsbury B E D. The modulation spectrogram: in pursuit of an invariant representation of speech [C]. In 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997: 1647–1650 vol.3.

[11] Wang D, Chen J. Supervised Speech Separation Based on Deep Learning: An Overview [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2018, 26 (10): 1702–1726.

[12] Xu Y, Du J, Dai L, et al. An Experimental Study on Speech Enhancement Based on Deep Neural Networks [J]. IEEE Signal Processing Letters, 2014, 21 (1): 65–68.

[13] Cohen I. Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator [J]. IEEE Signal Processing Letters, 2002, 9 (4): 113–116.

[14] Mathur A, Saxena V, Singh S K. Understanding sarcasm in speech using melfrequency cepstral coecent [C]. In 2017 7th International Conference on Cloud Computing, Data Science Engineering - Confluence, 2017: 728–732.

[15] Xu Y, Du J, Huang Z, et al. Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement[J]. arXiv preprint arXiv:1703.07172, 2017.

[16] Sailor H B, Patil H A. Filterbank learning using Convolutional Restricted Boltzmann Machine for speech recognition [C]. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016: 5895–5899.

[17] Park S R, Lee J. A fully convolutional neural network for speech enhancement [J]. arXiv preprint arXiv:1609.07132, 2016.

[18] Ge M, Wang L, Li N, et al. Environment-dependent attention-driven recurrent convolutional neural network for robust speech enhancement [C]. In Proc. 20th Annu. Conference of the International Speech Communication Association, 2019: 3153–3157.

[19] Pascual S, Bonafonte A, Serr`a J. SEGAN: Speech Enhancement Generative Adversarial Network [C]. In Proc. 18th Annu. Conference of the International Speech Communication Association, 2017: 3642–3646.

[20] Lu, Y. J., Liao, C. F., Lu, X., Hung, J. W., & Tsao, Y. (2020). Incorporating broad phonetic information for speech enhancement. arXiv preprint arXiv:2008.07618.

[21] Z. Du, M. Lei, J. Han, et al. Pan: Phoneme-Aware Network for Monaural Speech

Enhancement[C]//ICASSP 2020 IEEE ICASSP. IEEE, 2020: 6634-6638.
[22] Yin, C. Luo, Z. Xiong, et al. PHASEN: A Phase-and-Harmonics-Aware Speech Enhancement Network[C]//AAAI. 2020: 9458-9465.
[23] Pandey, D. Wang. A New Framework for Supervised Speech Enhancement in the Time Domain[C]//Interspeech. 2018: 1136-1140.
[24] Valentini-Botinhao C, Wang X, Takaki S, et al. Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech[C]//SSW. 2016: 146-152.
[25] Adel, Hidri , et al. "Beamforming Techniques for Multichannel audio Signal Separation." International Journal of Digital Content Technology and its Applications 6.20(2012).
[26] Griffiths, Lloyd J, and C. W. Jim. "An alternative approach to linearly constrained adaptive beamforming." IEEE Trans Antennas & Propag 30.1(1982):27-34.
[27] O'Sullivan, James, et al. "Neural decoding of attentional selection in multi-speaker environments without access to clean sources." Journal of neural engineering 14.5 (2017): 056001.
[28] Ephrat, Ariel, et al. "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation." arXiv preprint arXiv:1804.03619 (2018).
[29] Lim, J. and Oppenheim, A. V. (1978). All-pole modeling of degraded speech. IEEE Trans. Acoust. , Speech, Signal Proc., ASSP-26(3), 197-210.
[30] Scalart, P. and Filho, J. (1996). Speech enhancement based on a priori signal to noise estimation. Proc. IEEE Int. Conf. Acoust. , Speech, Signal Processing, 629-632.
[31] Hu, Y. and Loizou, P. (2004). Speech enhancement based on wavelet thresholding the multitaper spectrum. IEEE Trans. on Speech and Audio Processing, 12(1), 59-67.
[32] Hu, Y. and Loizou, P. (2004). Incorporating a psychoacoustical model in frequency domain speech enhancement. IEEE Signal Processing Letters, 11(2), 270-273.
[33] Tsoukalas, D. E., Mourjopoulos, J. N., and Kokkinakis, G. (1997). Speech enhancement based on audible noise suppression. IEEE Trans. on Speech and Audio Processing, 5(6), 497-514.
[34] Ephraim, Y. and Malah, D. (1984). Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. IEEE Trans. Acoust.,Speech, Signal Process., ASSP-32(6), 1109-1121.
[35] Ephraim, Y. and Malah, D. (1985). Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE Trans. Acoust., Speech, Signal Process., ASSP-23(2), 443-445.
[36] Cohen, I. (2002). Optimal speech enhancement under signal presence uncertainty using log-spectra amplitude estimator. IEEE Signal Processing Letters, 9(4), 113-116.
[37] Loizou, P. (2005). Speech enhancement based on perceptually motivated Bayesian estimators of the speech magnitude spectrum. IEEE Trans. on Speech and Audio Processing, 13(5), 857-869.
[38] 王文昌. 子阵级波束形成技术研究[D]. 电子科技大学, 2010.
[39] 张禄. GSC 自适应波束形成的后置滤波算法研究[D]. 哈尔滨工业大学.
[40] Xu C, Rao W, Wu J, et al. Target Speaker Verification with Selective Auditory Attention for Single and Multi-talker Speech[J]. arXiv preprint arXiv:2103.16269, 2021.
[41] Chang X, Zhang W, Qian Y, et al. MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition[C]//2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019: 237-244.