第5章　语音合成

5.1　简介

语音合成是一种将文字自动转换为语音信号的技术。语音合成技术涉及声学、语言学、自然语言理解、信号处理、模式识别等多个学科，是信息处理领域的一门前沿技术。随着计算机硬件水平的不断提高和机器学习技术的蓬勃发展，语音合成技术逐渐从最初的基于语音学规则的参数合成，发展成基于大语料库的拼接合成和基于统计参数的语音合成，到目前基于大语言模型的语音合成，合成语音的可懂度和自然度也取得了明显提升，在很多场景都取得了成功的应用，例如语音播报系统、有声读物、地图导航、信息查询系统等。可以说语音合成技术正在悄然改变生活，甚至将来会成为人们生活中不可或缺的一部分。

5.1.1　语音合成的内涵

语音研究的目的不只是“弥补听官之不足或方便文字之录入”，更重要的是揭示言语交际的机理，获取自然语音中的各种知识和信息，并为人类的信息交流服务。

一般来讲，实现计算机语音输出有两种方法：一是录音/重放，二是文字与语音转换。若采用第一种方法，首先要把模拟语音信号转换成数字序列，编码后，暂存于存储设备中（录音），需要时，再经解码，重建声音信号（重放）。录音/重放可获得高质量声音，并能保留特定人的音色，但所需的存储容量随发音时间线性增长，而且不能满足实时修改发音内容的需要。

第二种方法是基于语音合成技术的一种声音产生技术。它源于语音生成机理及可计算声学模型，主要功能是能将计算机中任意出现的文字，转换成自然流畅的语音输出，如图 5.1 所示。它的研究涉及语音学、人工智能、计算机科学、语言学、心理学等，推动了相关学科的进步和发展。

图 5.1　语音合成系统框架

目前，语音合成技术在很多领域都获得了成功的应用并取得了良好的社会效益，例如银行、医院、火车站、机场等人口密集区的信息播报系统、车载导航系统、自动应答呼叫中心等。另外，随着智能手机、个人数字助理等便携式设备的迅速普及，语音合成技术也逐渐向有声小说、电子娱乐、语音教学等领域渗透。可以说，语音合成技术正在影响着人们生活的方方面面。

5.1.2　语音合成的发展历史

在早期，“语音合成”是指用机器产生人工言语的技术。它可以通过力学的（机械的）、光学的或电子的手段产生类似人说话的声音。

最早的语音机器是由 von Kempelen 于 1780 年制造的。它完全是机械式的，通过风箱向簧片送气来模拟声带的振动。声道是用一段软的橡胶管模拟的谐振器，其形状由操作员的手来控制。操作者通过控制操作杆和开口，可以发出/a//o//u//p//l//m//r//n/等元音和辅音。在此之后的许多年中，很多人致力于这种机械式语音合成器的改进和完善。20 世纪 30 年代，Paget 的合成器已能说出像“Hello London，are you there?”之类简单的话。但是，所有这些机械式合成器合成的语音都和人说的自然语音相差甚远。

Homer Dudley 与他的同事 Bell 电话公司的工程师 Ricsz 和 Watking，于 1937 年研制了 Voder（Voice Demonstrator），使语音合成技术从机械模拟步入了电子模拟的新时代。Voder 不再拘泥于声带和声道的生理特征，而着眼这些器官的声学功能，并借助于电子技术模拟这些功能，达到合成语音的目的。1939 年，在美国纽约的国际博览会上，展出了 Bell 实验室 Dudley（1939）制作的电子式语音合成器 Voder。它是利用共振峰原理制作的早期的语音合成器。Voder 有一个像琴键一样的键盘，控制着十个带通滤波器，产生各种语音。一个训练有素的操作员，可以用 Voder 说一些简单的语句。

这些早期的合成器，功能都非常简单，称不上“合成系统”，没有实际的应用价值所以也谈不上性能评测。但是在这一时期，由于电话等通信技术的发展，人们对言语清晰度试验和可懂度理论的研究日趋完善。

早在 1908 年，Rayleigh 研究电话机时便注意到了言语清晰度问题。他为使电话机能分出 f 和 s 进行了不少试验。随后，1910 年，Campbell 提出了“电话可懂度”问题，并设计了测试音节表。当时使用的音节比较简单，都是 C-V（辅音元音）结构。

在这一时期，对语言学的研究也不断发展。美国的 Dewey 于 1923 年出版了英语语音相对出现频率的统计结果。1929 年，美国 Bell 实验室的 Fletcher 与 Steinberg 根据 Dewey 的统计数据，改进了清晰度测试音节表，编制出等难度、等出现率的 C-V-C 音节表，并对试验队员的选择和试验条件的控制进行了比较系统而广泛的研究，建立了比较完整的清晰度试验方法。

哈佛大学的 Egan 对清晰度测试方法进行了系统的研究，为言语清晰度测试方法标准化作出了重要贡献。他对测试条目的难度分布做了特别的研究，使清晰度的测试更为灵敏可靠。言语清晰度试验为合成语音的性能评测奠定了良好的基础。

语音合成技术的发展是和计算机技术、数字信号处理技术的发展分不开的。随着电子技术的发展，人们开始使用计算机、数字滤波器及各种电子设备进行语音合成的研究。G. Fant 在 1960 年所著 *Acoustic Theory of Speech Production* 一书中，系统地阐述了言语

产生的声学理论，从而使语音合成技术的发展迈出了关键的一大步，随之而来的是大批的基于该理论之上的串联或并联共振峰合成器的诞生。

Klatt(1987)总结了这一时期的语音合成技术发展过程。Klatt 从合成器的早期发展、按规则合成方案、实验室文语转换系统和商用文语转换系统四个层面上，沿着发音参数合成、共振峰合成和波形编码合成三条线索描述了英语语音合成技术的发展过程。其中典型的合成器有 Coker(1967)、Olive(1977)、Klatt(1980)、Holmes(1983)等。值得一提的是 Holmes(1973,1983)的串联共振峰合成器和 Klatt(1980)的串、并联共振峰合成器。只要精心调整合成参数，这两种合成器都能合成出非常自然的语音。后来，许多语音合成系统都是基于这两个模型的。

20 世纪 80 年代末，语音合成技术又有了很大的发展，特别是基音同步叠加方法(Moulines and Charpentier, 1990)的提出，使基于时域波形拼接方法合成的语音自然度大幅提高。基于 PSOLA 技术的汉语、法语、德语、英语、日语等语种的文语转换系统已研制成功，并公开发表。TD(Time Domain)PSOLA 算法只作用于时域波形，而 LP(Linear Prediction)-PSOLA 及 FD(Frequency Domain) PSOLA 是与 LPC 编码技术相结合的产物，使语音合成的质量又提高了一步。进入 90 年代，计算机的迅猛发展为基于大语料库的波形拼接语音合成系统提供了大量的技术保证。并成为语音合成的主流方法。

20 世纪末，语音信号统计建模算法的研究已经趋于成熟，人们对语音合成系统提出更高的要求。统计参数语音合成系统(Statistical Parametric Speech Synthesis, SPSS)已经成为新的主流算法，尤其以基于隐马尔可夫(Hidden Markov Model, HMM) 的语音合成最为成功，其相应的合成系统为基于 HMM 的语音合成系统(HMM based Speech synthesis System, HTS)。HTS 可以在不需人工干预的情况下，高效自动的搭建合成系统，由于统计的缘故，对发音人和发音风格的依赖较小，合成语音的语音风格和音色容易人为控制，并且合成系统的规模没有波形拼接的那么大。

进入 21 世纪，随着深度学习在语音识别领域取得了突破性的进展，深度神经网络(Deep Neural Network,DNN)在语音合成中也有了大量的应用，凌振华等在传统 HTS 合成框架的基础上，将深度置信网络(Deep Belief Network,DBN)作为语音参数后增强模型应用到语音合成中。利用 DBN 强有力的学习能力，在语音合成的后端实现在谱参数上的一个更加精细的调整，使得合成音质有了不少的改善；Heiga Zen 提出了基于 DNN 的统计参数合成方法，核心思想是直接通过深层神经网络来预测声学参数，避免了基于 HMM 的语音合成方法中由于决策树聚类导致的模型精度降低，从而提高了合成语音的音质。2014 年，LSTM 在语音技术中的使用掀起了新的热点。Frank K. Soong 等进一步将双向 LSTM 神经网络应用到语音合成中，大幅提高了合成语音的音质。

近年来，一些学者致力于端到端的语音合成模型的建模，并取得了性能上的巨大提升。2016 年，谷歌公司 Deepmind 研究团队提出了基于深度学习的 WaveNet 语音生成模型。该模型可以直接对原始语音数据进行建模，避免了声码器对语音进行参数化时导致的音质损失，在语音合成和语音生成任务中效果非常好。然而由于该模型是样本级自回归采样的本质(Sample-level Autoregressive Nature)，速度较慢。同时，它还需要对来自现有语音合成文本分析前端的语言特征进行调节，因此不是端到端的。最近开发的神经模型是百度公司提出的 Deep Voice 和支持多说话人的 Deep Voice 2，它通过相应的神经网络代替传统参数

语音合成流程中的每一个组件。但其中的每个组件都是独立训练出来的，因此也不是端到端的。2017 年 1 月，Bengio 等提出了一种端到端的用于语音合成的模型 Char2Wav，其有两个组成部分：一个读取器(reader)和一个神经声码器(nerual vocoder)。读取器用于构建文本(音素)到声码器声学特征之间的映射；神经声码器则根据声码器声学特征生成原始的声波样本。本质上讲，Char2Wav 是真正意义上的端到端的语音合成系统。2017 年 3 月，谷歌公司王雨轩等提出了一种新的端到端语音合成系统 Tacotron，该模型可接收字符的输入，输出相应的原始频谱图，然后将其提供给 Griffin-Lim 重建算法直接生成语音。在美式英语测试里的平均主观意见评分达到了 3.82 分。此外，由于 Tacotron 是在帧层面上生成语音，所以它比样本级自回归方式快得多。谷歌公司王雨轩等还进一步将 Tacotron 和 WaveNet 进行结合，在某些数据集上能够达到媲美人类说话的水平。随着 Neural Vocoders(如 WaveNet、WaveGlow、HiFi-GAN)的出现，合成语音的音质进一步接近人类发音的真实感和细腻度。

2021 年，扩散模型在语音合成任务中占据了重要地位。DiffWave 是扩散模型在语音合成中的典型应用，它使用扩散过程从随机噪声生成高保真音频，并展现出接近甚至超越 WaveNet 的音质表现。扩散模型的生成过程相比传统的序列模型具有更高的灵活性，且其并行化潜力使得其在大规模语音合成任务中具备更高的效率。2022 年，Codec 技术不仅用于压缩和解压缩音频信号，还能够用于构建高效的音频生成系统。例如，EnCodec 是 Meta 推出的一种基于神经网络的高效音频压缩框架。该系统不仅能够在较低的比特率下保持高音质，还可以用于实时语音合成和音频生成。2023 年，一系列基于语言模型的 TTS 框架被提出，如 VALL-E 和 SoundStorm 等，这些模型在减少语音失真、提高语音自然度和保持个性化声音特征方面都展现了强大的能力，特别是在少样本语音合成和多语言支持方面，表现十分突出。

5.2　文本分析

文本分析是语音合成系统的前端模块，它的主要任务是对输入的待合成文本进行分析理解，为后端合成器提供必要的信息，如拼音、停顿等。不同的后端合成器需要的信息也不尽相同。对于最简单的语音合成系统，可能文本分析只提供拼音信息就足够了，而对自然度要求较高的语音合成系统，文本分析需要给出更详尽的语言学和语音学信息。因此，文本分析实际上是一个人工智能系统，属于自然语言理解的范畴。

对于汉语语音合成系统，文本分析的处理流程通常包括文本预处理、文本规范化、自动分词、词性标注、多音字消歧、节奏预测等，如图 5.2 所示。一些新的系统还包括重音预测、方言处理、不同语气处理等模块。

输入语音合成系统的待合成文本，首先必须预处理成统一的格式后才能进行后期处理。文本预处理包括删除无效符号、断句、内码转换等。依次检查输入文本，将其中的无意义符号删除或设置特殊标记，之后处理时对这些标记过的符号不做任何处理。这样的检查对于后面的分析非常重要，因为无意义符号的存在会使后面的文本规范化、自动分词等处理出错。断句是指依据特征标点符号、句长统计等信息将输入的连续篇章分割成句子单元，它决定了合成的粒度。断句的处理通常忽略了单句之间的影响，认为一个单句可以作

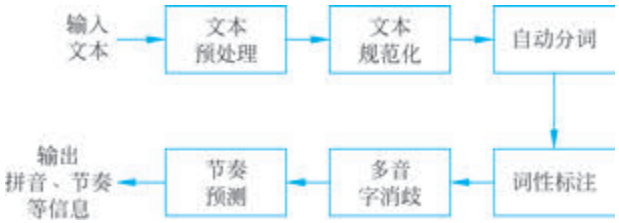

图 5.2　文本分析流程

为一个单独的发音单元来对待。内码转换则是将输入的各种内码统一转换成系统中使用的内码，以便合成系统支持多种内码的输入。

文本中除了普通汉字外，还经常出现各种日期、时间、货币、度量等特殊字符。文本规范化的任务就是将文本中的这些特殊字符识别出来，并转换为一种规范化的表达。例如，原始文本“同比增长 8%”规范化后为“同比增长[PER 百分之八]”，其中“百分之八”用方括号括起来，表示一个独立的分词单元，避免在后面的分词模块中与普通文本交错产生分词错误。“PER”表示括起来的部分是一个百分比类型。文本规范化是文本分析中的一个重要模块。特殊字符如果处理得不好，不仅会影响合成系统的自然度，还会导致系统完全读错某些字符串的读音，使合成的语句很难听懂或导致理解错误。

词是具有固定形式的并能独立运用的最小语义单元。汉语是一种词根语，与英语等西方语言不同，它采用连续的书写方式，词和词之间无自然界限，无词尾形式标记，无形态变化。这种“三无”现象使得人们在阅读时，大脑需要将语句切分成更小的语言单位——词。同样，计算机在理解和处理书面汉语之前，也必须先进行自动分词，因此分词模块是语音合成文本分析中一个非常重要的模块。

词性是自然语言理解中使用最为频繁的浅层语法信息。在接下来的多音字消歧和节奏预测模块中，上下文的词性均起着非常关键的作用。一个词语可能的词性往往多于一种，如“打”在“打酱油”中是动词，在“一打鸡蛋”中是量词。词性标注的难点就是确定这种兼类词的词性。汉语缺乏形态变化，词的使用非常灵活，可以充当句子的不同成分，因此汉语中词语兼类情况特别多，特别复杂。尽管兼类词在词典中的比例不是很高，但是在文本语料中出现概率却比较高。

字音转换的任务是将待合成的文字序列转换为对应的拼音序列，即告诉后端合成器应该读什么音。大多数情况下，字音转换可以通过在词典中搜索当前词，配以对应的拼音。然而，对于当前字(词)对应多种拼音的情况，简单的词典检索的方法就很难解决了。字音转换的关键就是解决这种一字(词)多音的问题。事实上，这种情况并不是汉语独有的，许多语言都存在多音字(词)现象，如英语中的“read”，只是汉语中更为常见，更为复杂。如“干”字在“干衣服”中读“gan1”，而在“干重活”中读“gan4”。汉语中常见的多音字有“为、长、重、中、行、种”等。除去多音字，汉语中还有少量多音词，如“教授(jiao4shou4 或 jiao1shou4)”“朝阳(chao2yang2 或 zhao1yang2)”等。

节奏有时又称为韵律结构，节奏组织的好坏极大地影响着合成语音的自然度和可懂度。节奏预测是语音合成系统中必不可少的一个环节，其后的停顿、时长的预测，以及基频曲线的生成都与节奏预测的结果密切相关。大多数的实用系统都是基于分词和词性标注

得到的词法信息来预测韵律节奏。不过也有一些系统为了更好地进行韵律节奏的分析，要求文本分析模块提供更多深层的语法信息，例如语法结构，这就需要在文本分析模块中集成句法分析器。目前实用句法分析器的准确率还不够高，英语中较为常用的分析器是Collins(1999)的基于中心词驱动的概率句法分析器(准确率达 90%)，汉语由于句子构成更灵活，所以比英文更难处理，还达不到英语句法分析的准确率，实用程度不高。

在后端的韵律模型中，停顿一般仅由节奏层级决定，基频和时长的预测通常还需要重音分布的信息。然而，由于重音与说话者想要表达的语义或者语用信息密切相关，已经有研究人员(朱维彬，2007；李雅，陶建华，2011)在此领域进行了探索，但目前大部分文本分析模块暂时还不具备自动预测重音的能力。

5.2.1　文本规范化

语音合成系统的输入文本经常不是规范的文字，它充满了数字、英文、公式及一些特殊符号。文本规范化是对输入文本进行分析，把输入文本中的数字、特殊符号等转换为规范的文本，为后续的文本分析进行准备。

在有些文献中，文本规范化的内容还包括文本格式的规整，如输入的文档格式有WORD 文档、WPS 文档或网页等，需要将这些各式各样的文档格式进行规整以进行后续处理。此外，文件中还有可能包含一些标记，如字体的标记(加粗或倾斜)等，这些对于语音合成通常并不需要，因此，在文本规范化的过程中，也需要去除这些无用的信息(任卫军，2003)。

文本规范化的困难在于，很多时候如果没有充分的先验知识，人也无法正确理解文本中符号的意思。目前主要采取的思路就是充分利用文本的上下文信息，归纳在特定环境下的各种处理策略，通过建立大量规则的方式对文本进行规范化处理。

文本中需要特殊处理的符号主要包括数字字符、英文字符和特殊符号等几种。例如，在 NANTC(North American News Text Corpora)中共有 121 464 个 NSW(Non-Standard Words)，分布如表 5.1 所示。

表 5.1　NSW 在 NANTC 中的分布

Major type	Minor type	%
Numeric	Number	26
	Year	7
	Ordinal	3
alphabetic	As word	30
	As letters	12
	As abbrev	2

文本规范化的方法有基于规则的和基于统计的两种方法。

1) 基于规则的方法

以上这些问题，总体看来，有一定的规律可循，可以采用人工加入规则的方法进行处理，当然，有限的规则远不能解决所有的杂乱无章的文本，但能够较好地解决以上罗列的问题，就可以说文本规范化工作已经大体可以接受。

规则可分为外部规则和内部规则。内部规则是嵌入系统中的，可读性和可维护性较差。外部规则可以采用一些标记语言来写，更简单的可以自己定义一套标记，可维护性和可扩充性都较好(陈志刚，2003)。

外部规则的构造也是形式多样。但一般地，每条规则都至少包括两部分：条件部分(C)和处理结果(R)。基本形式为：C--R。

举例如下。

2008 年北京召开奥运会。

这里面涉及的是数字年份的处理。可以定义规则如下：

[number](2|4)[Char](“年”)--[number](“year”)

这条规则表示，如果遇到 2 位或者 4 位的数字，并且数字后面紧接着汉字“年”，那么，这个数字代表的意思就是年份“year”，然后根据表示年份的数字变化规则进行转换。

同样地，可以写出更多的规则。如：

[char](“比赛|比分|结果”)[char](NULL) * [number](0)[char](“:”)[number](0)--[number](“score”)

这条规则表示如果遇到“比赛”“比分”和“结果”中的任意一个词时，其后如果接着一些不定长度的字符，“ * ”表示 0 次或者多次。“[number](0)”表示长度大于或等于 1 的数字。“[char](“:”)”表示字符“:”。在这样的情况下，数字表示比分。

这样的规则可以处理如下的文本：

两队的比分已经逐渐拉开，目前为 98 ∶ 76。

为了能够更细致地进行文本规范化，规则的数量总是越加越多，不可避免地会出现规则冲突的问题。如“15:20”，在没有上下文的情况下，很难确定这个数字是表示时间还是比分，因此，在上下文信息不够充分的时候，规则的冲突是不可避免的，因此需要一定的机制来进行规则的消歧。规则的消歧可以利用规则的使用顺序来简单解决，也可以借用概率的思路，采用具有最大概率的规则。一般来说，先确定特殊符号的读法，最后确定数字的读法。

2) 基于统计的方法

也有一些学者采用基于统计的方法进行文本规范化。主要的方法有加权的有限状态自动机方法、语言模型方法(Kneser and Ney，1995；Chen and Goodman，1998)、决策树模型、最大熵方法、信源信道模型等。

采用统计的方法做文本规范化的工作并不是太多，主要因为要处理的特殊符号与一般的统计语言学有着不同的规律，而且，用于统计训练的特殊符号的标注语料一般也较少，故文本规范化部分多采用基于规则的方法来做。

文本规范化是文本处理中的一个重要方面，影响着语音合成系统的可懂度。正确的文本规范化必须充分结合上下文信息。

5.2.2 分词和词性标注

生成自然、清晰的语音离不开对文本的理解，分词和词性标注是语音合成后续所有文本分析的基础步骤，也是自然语言处理中的一个基本问题。

1) 分词

汉语文本不像西方书面语言，汉语词与词之间没有任何空格之类的显示标志指示词的

边界。汉语自动分词的主要任务就是把没有明显分割标志的字符串，包括标点符号、数字、各种缩略语、标记、人名、地名、机构名等分割成相应的词串。

在进行分词前，首先必须明确“词”的概念。词是语言中最小的能独立运用的单位，但它的划分还是有很大的争议，主要表现在三方面：

(1) 词与语素之间的划分。

(2) 词与短语的划分。如蓝天、白云等这些字符串很难肯定地说应该切开还是作为一个单独的词。

(3) 对于“词”的认识，普通说话人的语感和语言学家的标准也有很大差异。有关专家调查表明，在母语为汉语的被试者之间，对汉语文本中出现的词语的认同率只有 70%左右，从计算的严格意义上说，自动分词是一个没有明确定义的问题(黄昌宁等，2003)。关于这个问题，1992 年公布的国家规范《信息处理用现代汉语分词规范》(GB/T 13715—1992)(刘开瑛，2000)给出了一些分词的参考标准。但是，对于具体的不同目的的分词系统，分词单位的选择仍然有一定的自由性。对于语音合成系统，分词主要是为了韵律模块和声学模块使用，并不需要太多的句法信息，因此，可能不需要将语素单独地列出，可以从韵律的角度来考虑词的单位。

中文分词的难点包含以下几点：

(1) 切分歧义：中文自动切分歧义在汉语文本中普遍存在。切分歧义主要包括交集型歧义、组合型歧义和混合型歧义，这些歧义字段中有些属于真歧义，有些属于伪歧义(刘颖，2002)。

(2) 交集型歧义：汉字串 ABC 中如果 AB、BC 都是词，此时汉字串 ABC 为交集型切分歧义(宗成庆，2008)。例如，汉字串“大学生”中“大学”“学生”都是词，因此，“大学生”这个汉字串包含交集型歧义字段。这种情况在汉语文本中非常普遍，类似的还有“研究生物”“中小学生”等。

(3) 组合型歧义：如果汉字串 AB 是一个词，同时满足 A 是一个词，B 是一个词，则汉字串 AB 为组合型歧义(宗成庆，2008)。如以下两个句子中“将来”有两种不同的切分：他/将/来北京作报告；他/将来/想成为一名科学家。

(4) 混合型歧义：同时包含交集型歧义和组合型歧义(宗成庆，2008)。例如以下三个句子：这样的/人/才能/经受住考验；这样的/人才/能/经受住考验；这样的/人/才/能/经受住考验。

(5) 真歧义：歧义字段在不同的语境中确实有多种切分形式。例如以下两个例子的“地面积”这一汉字串：这块/地/面积/还真不小；地面/积/了厚厚的雪。

(6) 伪歧义：歧义字段单独拿出来看有歧义，但在所有真实语境中，仅有一种切分形式。例如“挨批评”这一汉字串，在所有的语境中只能切分成“挨/批评”，不能被切分成“挨批/评”。

未登录词处理：互联网每天都会产生大量的新词，虽然一般的词典都覆盖了大部分的词，但还是有很多词没有收录进去，词典不可能穷尽式地收录进所有的词汇，将这些未收录进词典的词叫作未登录词。主要包括两大类：一类是新涌现的普通词汇或专业术语，如“超女”；另一类是人名、地名、机构名、译名等专有名词。时间词和数词也是未登录词的一种，但是相对前两种未登录词而言，时间词和数词的区分还是相对较容易。真实文本中，未登

录词对分词精度的影响超过了歧义切分(孙茂松,1995),未登录词总数的大约九成是专有名词,其余的为通用新词或专业术语(黄昌宁等,2003)。

对于以上两种不同的未登录词,处理方式也是不同的。对于第一种,大多采用基于大规模语料库的统计方法解决,包括基于 HMM 或语言模型的识别方法(Bikel et al.,1997; Sun et al.,2002; Zhang et al.,2003c)、基于最大熵模型的识别方法(Mikheev,1998; Borthwich,1999)、基于错误驱动的学习方法(Aberdeen,1995),以及决策树方法(Sekine et al.,1998; Collins,2002; Collins et al.,1999)等。对于第二种专有名词的识别,近年来也逐渐成为中文信息处理领域的一个热点问题,一般采用规则加统计的方法解决。

2) 自动分词方法

基本分词方法都是基于词表的方法,包括正向最大匹配法(Forward Maximum Matching Method, FMM)(刘源等,1994)、逆向最大匹配算法(Backward Maximum Matching Method,BMM)(梁南元,1987)、双向扫描法(揭春雨,1989)、N-最短路径方法(张华平等,2002)、逐词遍历法(刘颖,2002)等。

(1) 正向最大匹配算法:从左向右取代切分语句的 m 个字符作为匹配字段,m 为词典中最长词条包含的字符个数;查找词典并进行匹配。若匹配成功,则将这个匹配字段作为一个词切分出来。若匹配不成功,则将这个匹配字段的最后一个字去掉,剩下的字符串作为新的匹配字段,进行再次匹配。重复以上过程,直到分析完待切分汉字串为止。

(2) 逆向最大匹配算法:该算法是正向最大匹配算法的逆向思维(最大匹配的顺序不是从首字母开始,而是从末尾开始),匹配不成功,将匹配字段的最前一个字去掉。逆向最大匹配算法要优于正向最大匹配算法(朱巧明等,2005)。

(3) 双向扫描法:双向扫描法是将正向最大匹配法得到的分词结果和逆向最大匹配法得到的结果进行比较,从而决定正确的分词方法(Sun Benjamin.,1995)。

(4) 最短路径匹配算法:张华平等(2003)提出了旨在提高召回率并兼顾准确率的基于N-最短路径方法的汉语词语粗分模型。其基本思想是根据词典,找出字串中所有可能的词(也称全分词),然后构造词语切分有向无环图。这样,每一个词对应图中的一条有向边。若赋给相应的边长一个权值(该权值可以是常数,也可以是构成的词的属性值),然后针对该切分图,在从起点到终点的所有路径中求出最短路径,该最短路径上包含的词就是该句子的切分结果。图 5.3 以句子“他说的确实在理”为例,给出了 3-最短路径的求解过程。

基于规则的方法,一般都是人工预先加入规则,或者在辞典中添加相应的信息辅助分词。基于规则的方法通过模拟人对句子的理解,达到识别词的效果,基本思想是语义分析,句法分析,利用句法信息和语义信息对文本进行分词。自动推理,并完成对未登录词的补充。

最近几年研究较多的分词方法都是基于统计的方法。基于统计的方法主要采用一个已经标注好的训练语料库,找到在一定的上下文信息条件下具有最大概率值的词串。统计学中的许多方法都可以用来进行分词。

例如,基于统计语言模型(N-gram)的分词方法(冯冲,陈肇雄等,2006)、基于隐马尔可夫模型(Hidden Markov Model,HMM)的分词方法、基于最大熵模型(Maximum Entropy, ME)的分词方法(Xue and Converse,2002; Low, Ng et al.,2005)、基于条件随机场模型(Conditional Random Fields,CRF)的分词方法(Zhao et al.,2006)、基于双向长短时记忆网

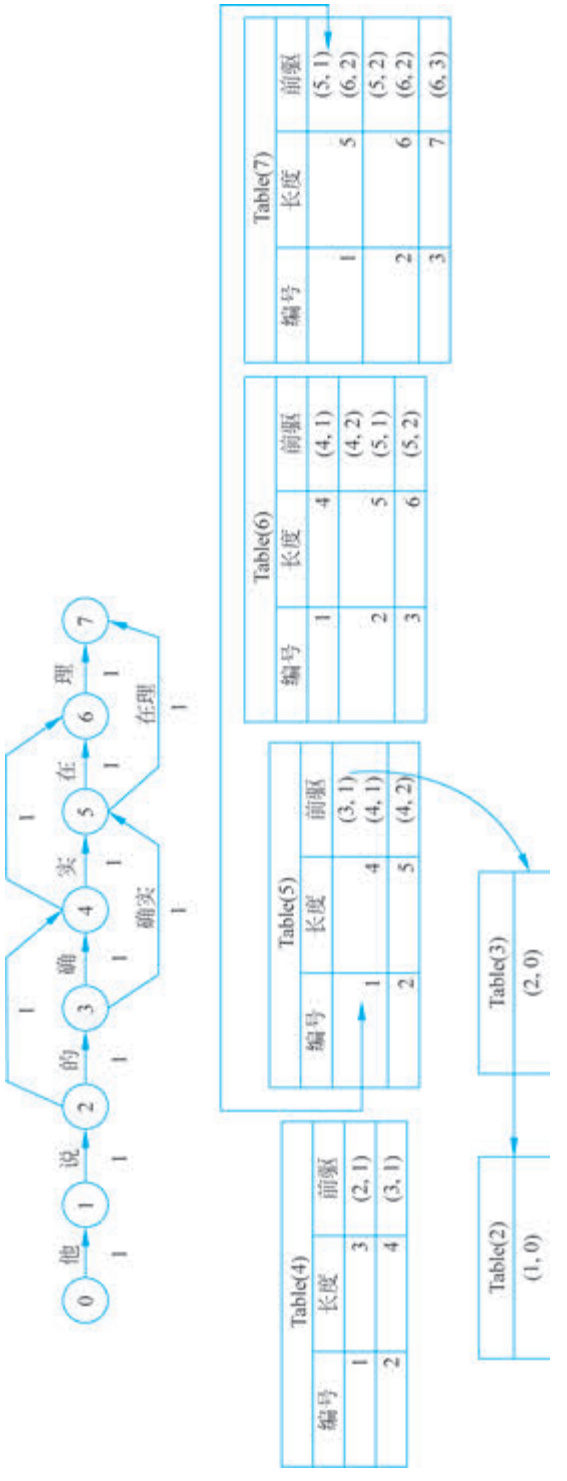

图 5.3　句子“他说的确实在理”的求解过程（N＝3）

络与条件随机场相结合(Bidirectional Long Short Term Memory-Conditional Random Fields, BLSTM-CRF)的分词方法(Huang Z, Xu W et al., 2015)等,这些方法在分词方面都取得了不错的效果。

3) 词性标注基本问题及难点

汉语词性标注同样面临很多棘手的问题。

词性标注的主要研究内容是兼类词的识别,兼类词是指具有多种词性的词。张虎等(2004)对北京大学计算语言学研究所在网上公布的 200 万汉字语料进行了统计,结果表明兼类词虽只占到 11%,但兼类词的词次却占到了 47%。中文中常用词的兼类现象很严重,因此相对而言,词性标注也是文本处理中的难点。

如教育是一项崇高的事业。她教育了我。其中,“教育”这个词在第一句中为名词,第二句中为动词。词性标注的任务就是根据上下文信息确定词的词性,并加以标注。

与汉语分词规范类似,目前为止还没有一个统一的被广泛认可的汉语词类划分标准。例如,LDC 标注语料中,汉语词性标注集有 33 个词类(Xia, 2000);北京大学语料库加工规范中有 26 个基本词类、74 个扩充词类(俞士汶等, 2003a);山西大学提出的汉语词类标记集共有 25 个词类(刘开瑛, 2000)。

词性标注集的选择也是非常重要的。一般来说,语音合成系统中的词性标注集与自然语言处理中的标注集是相同的,词性分类较细;但是,词性是于句法分析中的,如果合成系统没有用到较深层次的句法信息,可以考虑采用缩减的词性集,既可以缩小模型大小,也有可能带来性能上的提升。

4) 词性标注方法

基于规则的方法是最早提出的词性标注方法,它手工编制包含繁杂的语法或语义信息的词典和规则系统。这种方法不仅费时费力,而且带有很大的主观性,难以保证规则的一致性。更大的问题是处理歧义长句、生词、不规范句子的能力非常脆弱,词性标注准确率不高。

基于统计的方法是基于机器学习的思想,从 20 世纪 80 年代中期到现在,许多人采用这种思路进行词性标注,取得了很好的效果。主要方法有有限状态自动机的方法(Emmanuel and Yves, 1997)(Roche and Schabes, 1995)、基于隐马尔可夫模型的方法(刘颖, 2001; Peifeng, 2005)、基于最大熵的方法(Adwait Ratnaparkhi, 1996)、条件随机场的方法、基于转换的错误驱动学习的方法(Brill, 1994)、人工神经网络(Schmid, 1994)等。

5.2.3 韵律处理

韵律是物理和语音效应相结合的复杂系统,是用来表达在日常生活语言交流中的态度、设想以及注意力。它包括内涵意义和外延意义。内涵意义指的是说话者所表达的情感或者听者从中猜测的情感等等信息,而外延意义指的是口头或者书面信息的语义内容。韵律在指导听者理解外延意义上有很重要的支持作用,而且在提示内涵意义或者说话者对话语、听者以及整个交流的态度都起主要的作用。

韵律部分的研究是一个复杂的系统工程,涉及语言学、语音学、心理学、语用学等学科的综合知识。一个语音单元除了由元音和辅音按时间顺序排列的音段成分之外,还必须包括一定的超音段成分,否则这个音节就不可能成为有区别意义的有声语言。韵律的声学参

数一般包括基频、时长、能量。目前对韵律研究的重点是音高、音长、音强三个超音段参数在连续语流中的分布规律及其相互的作用，而基本的研究方法仍是基于对生理特征的分析及对大语料库的统计分析。

对于一个 TTS 系统而言，韵律预测是十分重要的，它承启文本分析模块的分析信息，生成对合成系统具有指导意义的声学参数，是 TTS 系统中一个必不可少的模块。韵律预测的方法分为基于规则的方法和基于数据驱动的方法两类。早期的韵律预测采用基于规则的方法。研究人员需要大量的语音学试验的数据，试图从中发现语音在不同的上下文环境中，有着怎样不同的韵律表现，总结规律，从而指导韵律参数的预测。这种方法虽然能较好地预测韵律参数，但是也有很多的限制：基于规则的方法大多是和语言相关的，不具有通用性；开发周期较长，需要花费较长的时间统计分析；不具有灵活性；等等。因此，语音合成后期的研究大多采用统计的方法来预测韵律参数。多种基于数据驱动的方法，例如基于神经网络、基于分类与回归树等方法相继被提出。通过替换训练数据，就能得到针对不同语言、不同风格的韵律模型。

1）基频模型

基频一直是语音合成研究的焦点。研究表明，基频曲线对于不同的音节或音节组合，有其基本的规律，有相对稳定的变化模式，这些为进一步的连续语流的音高曲线（语调）的研究奠定了基础。连续语音的音高曲线融入了发音人的生理特征、感情、语义、语境以及很多的个人特征信息。赵元任先生的“大波浪小波浪”学说以及“橡皮带”理论（赵元任，1932）是语调研究的奠基学说，初步说明了语调的本质规律。沈炯（1985）则进一步扩充了这种思想，提出了语调调节的“双线模型”。Fujisaki（2004）、Kochansaki（2003）等结合发音生理机制及表面现象，提出了控制语调的具体模型。这些理论及相应的模型都能够反映连续语流音高曲线的基本规律，从而提高了语音合成的自然度。

2）Fujisaki 模型

Fujisaki 模型是藤崎博也（Fujisaki）于 1971 年所提出的一种韵律模型，这是较早的采用数学公式生成基频曲线的模型。Fujisaki 模型属于一种典型的重叠模型（Superposition Model），该模型认为完整的基频曲线是由几部分叠加而产生的。如图 5.4 所示为 Fujisaki 模型的原理图，从图中可以看出，完整的基频曲线由两部分叠加产生，一部分用来模拟基频曲线的整体走势（Phrase Command），如在陈述句中所出现的基频曲线下倾现象，该部分由几个冲激函数通过二阶系统产生；另一部分描述基频曲线的局部走势（Accent Command，对于汉语而言也可称为 tone command），如在英语中的重音现象，该部分由几个阶跃函数通过二阶系统产生。两部分数字在对数域叠加从而产生了完整的基频曲线。

Fujisaki 模型的数学表达式为

$$\ln F_0(t)=\ln F_b+\sum_{i=1}^{I}\mathrm{Ap}_i G_{\mathrm{p}}(t-T_{0i})+\sum_{j=1}^{J}\mathrm{Aa}_j[\mathrm{Ga}(t-T_{1i})-\mathrm{Ga}(t-T_{2i})] \tag{5.1}$$

$$G_{\mathrm{p}}(t)=\begin{cases}\alpha^2 t\exp(-\alpha t), & t\geqslant 0\\ 0, & t<0\end{cases} \tag{5.2}$$

$$\mathrm{Ga}(t)=\begin{cases}\min[1-(1+\beta t)\exp(-\beta t),\gamma], & t\geqslant 0\\ 0, & t<0\end{cases} \tag{5.3}$$

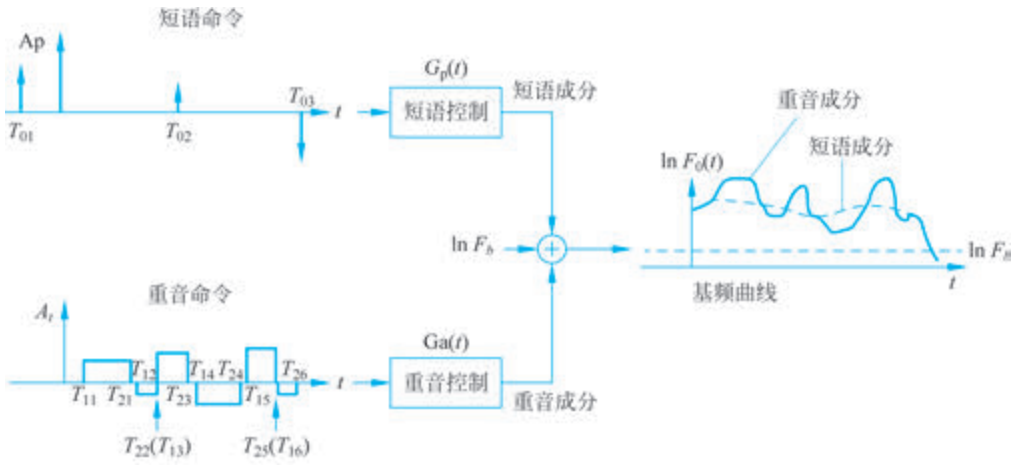

图 5.4　Fujisaki 模型

式(5.1)中的 F_b 为基频基准值,第二项 $G_p(t)$ 为短语命令(Phrase Command),第三项 Ga(t)为重音命令(Accent Command)。其余公式中的参数含义可参见表 5.2。

表 5.2　Fujisaki 模型参数含义

参　数	含　义
Fb	基频基准值
I	短语命令个数
J	字调命令个数
Api	第 i 个短语命令的幅度
Atj	第 j 个音节字调命令的幅度
T0i	第 i 个短语命令的时间
T1j	第 j 个音节字调命令的开始时间
T2j	第 j 个音节字调命令的结束时间
α	短语控制命令的衰减系数,一般取 2/s
β	字调控制命令的自然角频率,一般取 20/s
γ	字调控制命令的允许最大值,一般取 0.9

Fujisaki 模型有着较好的生理基础,可以近似地认为发音过程中喉头处甲状软骨的平移运动而引起的声带长短的变化具有冲激响应的形式,而转动引起的长度变化具有阶跃响应的形式。这种叠加的思想与赵元任的“大波浪和小波浪”之说是非常相似的。

最初的 Fujisaki 模型是基于规则的,通过指定冲激函数和阶跃函数的位置、大小以及二阶系统的参数从而得到模型的参数。后来人们又提出通过训练的方法来得到 Fujisaki 模型的参数。Fujisaki 模型起初是针对日语设计的一种基频模型,后来逐渐被推广到德语、汉语(H. Mixdorff et al.,2000)等各种语言。

3) PENTA 模型

许毅(2005)认为言语中的基频有多种表意功能,这些功能相互独立并且在基频中同时平行存在:有的以局部曲线为表现方式(声调,基频重音,边界调),有的以调节调域为表现方式(焦点和新话题)。汉语中的每一个音节,包括轻声和弱读音节都有其局部的基频目

标，而发音时说话人会在当前的调域范围内试图实现本音节的基频目标，因此汉语中的声调是通过与音节同步的基频目标来实现的，而实现过程会受到多种发音局限的约束。基于这种思想，许毅教授提出了针对汉语音节的平行编码及目标实现模型（Parallel Encoding and Target Approximation，PENTA）。

该模型认为言语的韵律是一个并行编码（Parallel Encoding）的过程，是言语的各项功能在基频曲线上的表现，如图 5.5 所示。

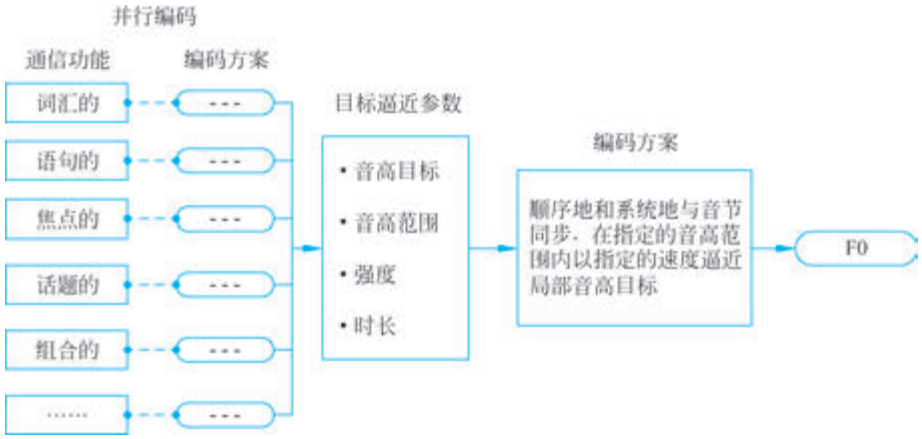

图 5.5　PENTA 模型结构图

由于 PENTA 模型主要是针对汉语的，以言语通信功能为出发点，选择音高目标作为编码对象，因此获得了中文语音合成领域的广泛关注。但是因为其自身理论尚有许多需完善的地方，因此该模型尚未得到广泛应用。

4）时长模型

早在 20 世纪 90 年代初，Jan Van Santen（1993）提出的基于统计的时长模型（Sums of Products，SoP）在部分西方拉丁语系的语言中已经有不少研究和应用，尤其在英文中音素的时长预测领域应用广泛（Santen，1994）。

SoP 模型的基本思想是高层韵律描述对时长的影响是由两种方式作用而共同得到的。其中一种是每个高层韵律描述属性对时长的作用是独立的，而另一种是多个高层韵律描述属性之间存在着联合作用的关系。SoP 模型更适用于大数据量，大特征空间的时长预测。在 SoP 模型中，考虑到语音在表现方面的对数特性，独立影像的作用可以作为一种加性的作用存在，而联合影像的作用可以作为乘积项的作用存在。因此，对于某一个元音在特定的高层韵律描述下的时长信息可以表示成一系列和决定该元音的高层韵律描述属性相关的系数表达式来体现（Santen，1993）：

$$\mathrm{DUR}(f)=\sum_{i\in T}\prod_{j\in I_j}S_{i,j}(f_j) \tag{5.4}$$

其中，T 是相加项集，I_j 是相乘项集，$S_{i,j}(f_j)$ 是每个相乘项中的乘数项。相加项集中的每项对应于一个韵律属性，如声韵母类型、调型等，相加项对时长的作用之间的关系是独立的，可以直接叠加。相乘项的每一项对应于一个韵律属性交互，即相乘项的每个乘数项因子对时长的作用之间相互影响，具有交互性。当 $T=\{1,2,\cdots,N\}$ 并且 $I_j=\{1\}$ 时，就可以

变成一个相加模型，反之则称为一个乘积模型。

SoP 模型的建议一般需要三个步骤：第一步是时长影响变量的选择；第二步是变量间的交互作用对时长影响的确定；第三步是建立时长模型。对于模型的建立，有以下几点关键技术值得注意：

（1）韵律属性的数据有名义尺度的，也有顺序尺度的，属性值不能直接对应到模型项上。因此需要对类别型韵律属性分类，建立一个分类树，使得每个属性值都是有实际语言学意义的，这样使得模型的可内插性成为可能。普遍采用的方法是将类别型韵律属性分类，使得名义尺度顺序化、数字化。

（2）韵律属性数目越多，属性空间会成倍增大，对数据的要求就越高。因此在样本数据量有限的情况下，必须控制模型中使用的韵律属性的数目，因此需要找到一种有效的方法分析哪些属性或者属性的组合对时长的影响更大。

（3）注意韵律属性之间的交互性。由于模型相乘项是由交互属性组合构成的，属性的交互性研究因此是非常重要的。作者设计了残差计算的算法来定量分析交互作用的大小，采用的残差值是以单个格子为单元的所有时长平均的一种空间距离算法，在某种程度上解决了样本数目不足以及特征空间不平衡等问题。

SoP 模型与时长数据的有序性有数学上的联系，有很好的可内插性，可以在某种程度上解决数据稀疏问题，因为每一项都是有序因子参数的乘积，而它们之间的影响是有规律可循的，这样对于数据稀疏的问题可以较好解决，另外该模型有很好的抗噪声特性。

5）韵律结构预测

（1）基于规则的方法：通过对语音学和语言学的研究总结一些通用的韵律规则，利用这些先验知识，可以建立一个规则韵律生成系统。通常规则系统包括两方面（赵晟，2002），一是汉语的通用规则，如汉语的 4 个调的基本形状，上声连接的变调规则，时长变化，语气语调的音高变化等；二是目标说话人的特定韵律特征规则，如个人的基本调型、调域、语速停顿规则。下面将以汉语普通话为例来介绍基于规则的韵律预测方法。

在连续语流中，每个字的发音会相互影响，连续语流中一个字的发音的声调与这个字单独发音时的声调会有不同，在合成的连续语流中，只有具有这种声调变化，才能使合成的语音具有较好的可懂度，否则将只会是单字语音的生硬连接。汉语普通话语句中的变调以二字词的连续变调最为重要，因为二字词在整个汉语词汇就占了约 74.3%。它的调型基本上是两个原调型的相连的序列，但受到连读影响使前后两调或缩短或变低。

时长也是语音的重要特征之一，它对合成语音的自然度和可懂度都有一定的影响。汉语中时长主要体现在韵母的调型段长度上，调长和调型是密切相关的，通常认为，上声音长最长，阴平、阳平次之、去声最短。在连续语流中调长的变化和声调一样，也是受到上下文环境的影响。例如，轻读音节的调长通常比重读时缩短近一半；在二字组中，后一音节的调长要比前一个音节的调长稍短等。在按规则汉语合成中，可将调长和调型一致起来，即，凡是平调，声调的调长适中，凡是降声调的调长较长，凡是降调的调长较短，轻声调长最短。声母的时长比较稳定。此外，根据试验语音学提供的经验，句子的最后一个音节的调长应比通常情况长 20%左右。除时长外，音节之间的间隙也对合成语音效果有一定的影响，适当地增加静音间隙，会使语音听起来更为生动。

不同的句式和语气会有不同的韵律表现。例如，朗读式的陈述句有比较明显的音高下

倾趋势；不带疑问词的疑问句需要靠句尾音高来提升等。而不同的情感的语音对基频和时长都有不同的影响。

此外，不同的人的发音，由于生理、地域等影响，不同发音人具有不同的调高、调域、时长等。普通的男性偏低、女性和小孩偏高。有的地方语速慢，有的地方语速快。对合成不同说话人的语音，需要对这个说话人做专门的分析，建立一个适合于个人的语音韵律规则体系用于语音合成。

(2) 基于机器学习的方法：基于人工规则的韵律生成，虽然有效地利用了专家知识，但是韵律的复杂度远大于现在已经掌握的人工规则，尤其是特定语言环境和特定人发音的一些细节。随着大语料库制作技术的出现，以及计算机硬件技术的发展和一些高效机器学习算法的成熟，使得录制大规模的语料库，用数据驱动的方法进行机器学习来发掘其中的韵律规则成为一种可能。数据驱动的方法可以尽可能地利用已有的语料资源，自动化的机器学习可以发掘更多更细的规则，对不同的数据库做快速的自适应，大幅降低研究者繁杂的分析工作。只需设计比较一致的训练语料数据库，机器学习策略，对机器学习的结果进行分析和筛选就可以得到一个比较好的韵律预测器，同时给出的机器分析结论也有助于提炼一些人工规则。训练的输入数据包括音节的调型、拼音、前后的韵律环境等等，输出可以是相应的基频、时长的原始或规整值。

TTS 发展到现在，很大程度上可以说是一个人工智能系统，从前端的文本分析和自然语言理解，后端的韵律声学参数的预测生成，都不同程度地用到了机器学习的技术。大量的机器学习和数据挖掘技术被应用到语音合成研究上来。如人工神经网络(Artifical Neural Network)、决策树(Decision Tree)等。利用这些机器学习策略就可以建立一些有效的韵律预测系统。

(3) 基于神经网络的方法：人工神经网络的信息处理过程分为三部分，首先完成输入信号与神经元连接强度的内积运算，然后将其结果通过激励函数，再经过阈值函数判决，如果输出值大于阈值门限，则该神经元被激活，否则处于抑制状态。通常神经网络由网络的拓扑结构、神经元特性和学习或训练规则这三个因素所决定。

神经网络按拓扑结构可分为分层网络模型和相互连接型网络模型。前者将神经元分成若干层，各层顺序连接，第 i 层的输入仅与第 $i-1$ 层输出相关联。后者允许任意两个神经元之间存在相互关联。

神经网络的学习算法一般分为有监督和无监督两种。有监督学习算法要求同时给出输入和正确的输出，网络根据当前输出与所要求的目标输出的差进行网络调整。这类算法中最著名的是反向误差传播算法(Back-Propagation)。

学习是神经网络的主要特征之一，目前较为流行的有感知机学习算法、玻尔兹曼机学习算法等。

神经网络已广泛应用于语音技术的各个领域，在韵律预测上也有很多研究。下面介绍了一个基于 4 层神经网络的韵律预测模型(Sin-Horng Chen，1998)，其方法流程图如图 5.6 所示。

如图 5.6 所示，该模型为 4 层神经网络，包含有 1 个输入层，2 个隐层和 1 个输出层。同时它们可以被划分为两部分：第一部分是输入层和第一个隐层；第二部分是输出成和第二个隐层。下面将详细介绍这两部分的工作原理。

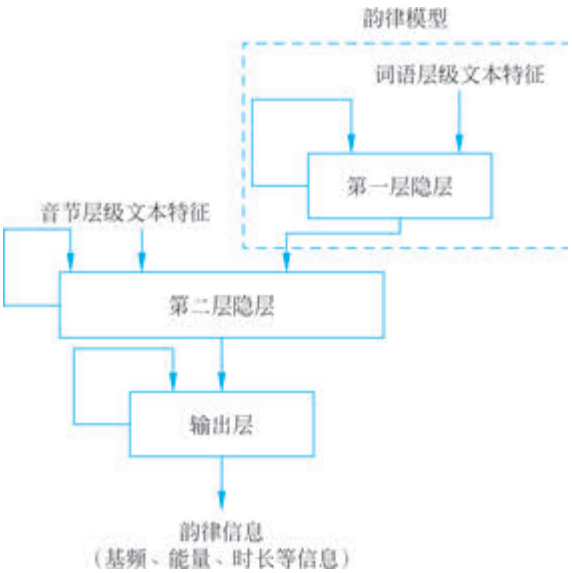

图 5.6　基于神经网络的韵律预测模型流程图

第一部分主要承担解析韵律短语结构的任务。该部分的输入单词级别的文本特征，例如当前单词及前后单词的词性、词长、句中相对位置和绝对位置等信息。词性集合包含 42 种不同的词性，其中包含 15 种动词词性、8 种名词词性、10 种副词词性、2 种连词词性和 7 种其他词性。

第二部分负责输出各种不同的韵律参数。该部分包含了 1 个输出层和 1 个隐层。该部分的输入为音节级别的文本特征和上一个隐层的输出信息。与第一部分不同的是，第二部分的所有输出都将反馈作为输入信息。音节级别的文本信息包括声韵母类型、声调、在词中位置、在短语中位置、前音节信息和后音节信息。输出信息为基频轨迹参数、能量等级和时长参数。

需要注意的是，这些输入信息进入神经网络之前必须先进行预处理，因为神经网络只能处理数值型或者是布尔型数据，对一些离散型数据（调型、声韵母类型）必须先预处理成一个布尔序列。输出一般取音节最大值点的值和位置、最小值点的值和位置，以及起点和终点的基频值、时长等，有了这些关键点位置和值就可以大致还原出原始的韵律信息。神经网络的训练比较简单，设计好输入输出数据，以及网络结构就可以进行训练，由于语音数据并不是很理想平稳的，而且易受外界因素影响有较大的波动，所以预测的结果通常依赖数据的稳定性而不是网络设计的复杂性。

（4）基于 BLSTM-CRF 的韵律预测方法：基于 BLSTM-CRF 的韵律边界预测模型总体架构的第一层为词矢量嵌入层，即将原始输入字映射到嵌入层中进行处理，后续输入 BLSTM-CRF 层。该层接收一个嵌入层的序列作为输入，并预测句子的韵律边界。其方法流程如图 5.7 所示。

词矢量嵌入层，通过增加一个嵌入层来引入对于特定任务的词矢量表示。在韵律节奏预测的过程中，对于不同类型的节奏预测，预训练的词矢量嵌入层会有所不同。当输入 U 个词组成的一句话 $w_1, w_2, \cdots, w_u$，通过词矢量嵌入层转换为一组词矢量序列 x_1，

$x_2,\cdots,x_u$。

BLSTM 层，通过嵌入层生成的词矢量作为两个相反方向的 LSTM 输入。对于每一个时刻 t，每个 LSTM 都以前一时刻的隐藏状态和当前时刻的词矢量作为输入，并输出一个新的隐藏状态。而 BLSTM 通过将两个方向的隐藏状态连接到一起来表示一个词。

CRF 层，通过使用隐藏状态表示 h_t 可以直接对每个词进行相互之间状态独立的预测，如基于 BLSTM-RNN 的预测方法。不过在韵律预测任务中，连续标记之间存在依赖关系，所以需要对句子进行联合建模，这样有利于联系前后的标记信息。所以在 BLSTM 层之后加入 CRF 层，使得模型可以在所有可能的序列中得到最优的路径。在训练的过程中，不断使正确标记的序列得分最大化的同时使其他序列的得分最小化。

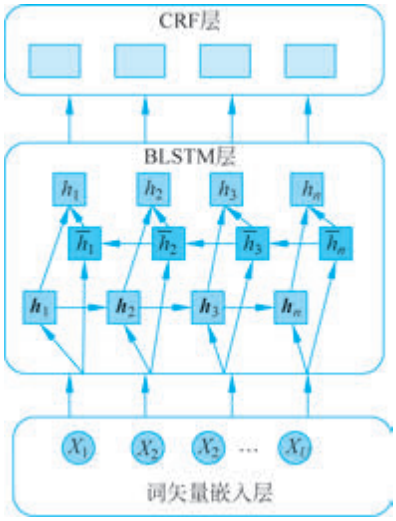

图 5.7　基于 BLSTM-CRF 的韵律预测方法流程图

5.3　拼接语音合成

基于大语料库的波形拼接语音合成技术已经成为语音合成技术的主流技术（Hunt et al.，1996；Chou et al.，2002；Christophe et al.，2002），其技术思想就是将在大量自然语流中的丰富的语音单元按照一定的规则拼接，得到高自然度的语音。可以想象，由于合成波形几乎原封不动地取自原始语料库，因此语料库的设计对合成结果的自然度至关重要。针对目标语音，语料库应能采用尽量少的语料覆盖尽量多的语言现象，包含丰富多变的韵律信息。语料库设计的合理性直接决定合成时是否能够选出所需要的合成单元，最后能否得到高自然度的语音。在语料库内对每个单元都将保留若干在不同韵律环境下的变体，在合成过程中通过前端对文本的分析获得当前单元的韵律环境属性集，再通过某种基元选取算法选出与当前环境最为匹配的变体，把这些变体拼接起来得到合成语音。可见语料库中包含的单元变体越多，越有可能精细地反映出韵律环境的细微变化，合成结果的表现力就会越强，越接近真人发声的结果。

由上述分析可知，在实现波形拼接合成系统的过程中，除了要设计出一个包含足够多

的可用的单元的语料库以外，还要解决的问题包括：

（1）选用什么样的单元作为拼接的基本声学单元。

（2）合成过程中如何将语料库中正确的单元挑选出来。

（3）选出这些单元之后，如何合理地拼接才能使得合成结果连贯、自然。

本节将分别对这三个问题进行讨论，并提出一种基于韵律环境约束的基元选取算法。

5.3.1 拼接系统中采用的基本声学单元

拼接语音合成系统中可使用的基本声学单元包括音素（声韵母）、半音节、音节、双音素、词或短语。在选择系统的合成单元时需要考虑下面四个目标。

目标一：单元之间的拼接损失要尽可能小。

这一点是拼接合成的基本原理所要求的。直观的考虑是采用尽量长的单元，如词、短语甚至句子，以这样的单元来合成文本，需要的拼接点少，从而也降低了平均拼接损失。但实际上，拼接处的不连续是任何拼接系统都无法避免的，少数几处明显的失真就会恶化听者对整个句子的印象，于是在系统中可以允许轻微的拼接损失，只要所选择的单元从本质上可以较为平滑地拼接即可。例如，在听感上元音间谱的不连续要比擦音间得更加明显，而音节之间的拼接则明显比音节内部声韵母的拼接更平滑。目前常用的办法是每个单元在音库中都保留多个样本，合成时从中挑选出拼接损失最小的。

目标二：单元应使得系统具备较强的通用性。

语音合成的最终目的是使得系统能够自然流畅的合成任意输入的文本，这要求音库中的单元应能覆盖任何可能出现的语言现象。对汉语来说，如果选择词或者短语作为基本单元，则要求音库至少应包含所有单字词，还要包含常用的二字词、多字词、成语甚至常用的短句以提高合成质量。这样固然可以通过无限度地扩大音库规模而提升音质，但在诸多场合下是得不偿失的，特别是当存储空间比较有限的时候。但在某些特定领域内，应用这样的单元可以得到较好的效果，如天气预报、机场管理、报时等。

目标三：单元应能在较低的代价下实现韵律的修改。

拼接合成直接选用音库中的原始波形进行拼接，实际选出的单元在韵律上可能与期望中的不相符合，这就需要采用某种韵律修改的算法对基频、音长等韵律信息进行修正。在信号处理的过程中必然会导致音质的损失，选择单元时应考虑这方面的问题，使得候选单元与目标单元的韵律模式尽量相近，如选用有调音节而不是无调音节。

目标四：单元应能满足统计训练的要求。

在语料库规模一定的情况下，采用的单元越小，每个单元的样本就会越多，越有利于采用统计机器学习的方法进行各种模型的训练，训练的效果也会越好。

1）音素（Phoneme）

从语音学上讲，音素是最小的语音单位，因此在选择拼接声学单元时，人们直观地会将它作为首要选择。尤其是在早期的语音合成系统受计算机存储空间与运算速度的限制的情况下，使用音素可以使语料库的设计更为灵活，便于设计出体积较小的音库。但是由于音素受相邻语音环境的协同发音影响较大，因此在自然语流中，其声学变体非常之大，这就给选音拼接带来了极大的困难。所选的样本不合适时，会导致相邻音素间的基频和频谱不连续，从而降低合成结果的音质。

2）声韵母(Initial-final)

声母-韵母组合是汉语普通话中一种特定的语言结构，一般来说一个声母和一个韵母组成音节，也有零声母音节。例如，“开”(kai)字可采用两个声韵母单元 k 和 ai 来合成。同音素类似，采用声韵母的好处在于可以把音库做得很小，但自然度和可懂度都非常低。因为在汉语中音节内部声韵母的协同发音效应尤为严重，采用声韵母拼接无法对这种协同发音进行建模，从而在拼接点处导致较为强烈的谱的扭曲。而且调型对音质也有很大影响，如果不区分韵母的声调，那么在韵律调整过程中也将导致音质的损失。解决的办法是对声韵母依与其共处同一音节内部的声韵母类型进一步细分，从而对协同发音进行建模(见第 3 章)。以分类后的环境相关的声韵母作为基本单元。

3）半音节(Demi-syllable)

顾名思义，将完整的音节从中间切开即得到两个半音节单元。这样做的好处是可以把声韵母间的过渡段完整地包含在前面的半音节之内。例如，“开”(kai)字可分为 ka 和 ai 两个单元，从而可以针对韵母对声母的影响进行建模。在半音节框架下，原始语料的切分也较为容易，可直接取音节的中点作为半音节的边界，便于自动切分。为追求更精确的结果，也可对自动切分的结果进一步手工调整，将边界调至元音频谱最平稳的位置。

4）音节(Syllable)

音节由元音和辅音组成，是中文语音合成系统里使用最为广泛的基本单元，其原因来自下面三个方面。

从声学角度来讲，音节可以有效地对其内部声韵母间的协同发音进行建模。汉语音节基本上都是 C-V(辅音-元音/声母-韵母)结构，声韵母间的过渡段都包含在音节内部，可以避免大部分协同发音影响。另外，汉语音节之间相对比较独立，便于从声学层面进行一些分析和处理。音节内部元音时长较长且能量较强，是音节的主要成分，因此语音合成的基频模型基本上都是以音节作为模型的框架。

从语言学角度来讲，音节与汉字存在一一对应的关系。人们在对输入文本进行分析时，直观地把汉字作为分析的基本单元，从而希望能够在系统内部保持这种对应关系。

从工程角度来讲，由于汉语普通话的音节数目足够小(不考虑音调，有 400 多个，即使考虑音调也仅有 1600 多个)，所以便于控制音库的规模，同时也将有足够的数据用于进行各种统计模型的训练。

对于一些西方语系语言，音节数目众多，结构比较复杂(可能出现 V-C、C-V-C 等结构)，非但不能避免协同发音影响，反而会导致音库容量急剧增大。

5）双音素(Diphone)

双音素在以英语为代表的西方语音合成里面较为常用。取相邻两音素各自谱最平稳的位置作为双音素的两个边界，例如双音素 h-iy 从擦音 h 的中部开始到元音 iy 的中部截止。单词 book 可由双音素序列/sil-b/，/b-uh/，/uh-k/，/k-sil/进行合成。若音素个数为 N，则两两组合可得到双音素的个数约为 N^2。在英语中很多音素组合实际上是不可能存在的，所以实际的双音素数目为 1300 多个。双音素之所以被广泛使用是因为它可以对音素间的协同发音进行建模，而拼接点都是音素内部最平稳的地方，所以相对较为连续，合成结果音质较好。

在中文系统中，Bell 实验室和 AT&T 两家机构的中文语音合成系统采用双音素作为

基本单元。采用音节作为基本单元能对音节内部的协同发音进行建模，但音节之间仍存在不连续的现象。采用双音素能够较好地解决这一问题。

6）词或短语

无论是在东方还是西方语系中，词或短语都会与某个特定的语义相对应，这有助于提升合成语音的可懂度。另外，从声学层面来看，词或短语彼此之间协同发音的相互影响较弱，因此合成结果的自然度也将大幅提高。但由于词的类别非常多，且在不同环境下声学表现也不尽相同，因此若使用词或短语作为声学单元，所需的语料资源量将会非常巨大。即使语料资源能够满足，这样巨大的语音合成系统也会丧失其应用价值。

综合上述对常用拼接单元的分析可知，它们相比之下各有优劣。因此在构造波形拼接所需要的语料库时，可以把不同单元结合起来，扬长避短。

5.3.2 基于目标代价和拼接代价的基元选取与拼接合成

合成系统输入的文本通过前端的文本分析与韵律预测模块后，转换成一串目标单元序列，每个目标单元都携带着与其对应的特征信息。下面要做的就是从音库中为每个目标单元挑选出与之相近的样本作为备选，同时还应保证选出的样本之间过渡得尽量平滑。基于上述目的，Alan W Black 提出了基于目标代价和拼接代价的基元选取算法。

1）基于目标代价和拼接代价的基元选取算法

假设目标单元序列为 $T=\{t_1,t_2,\cdots,t_i,\cdots t_n\}$，音库中与其相对应的候选单元为 $U=\{u_1,u_2,\cdots,u_i,\cdots u_n\}$，$i$ 为音节的序号，n 为待合成的目标单元的数目。目标代价 $C^t(u_i,t_i)$ 用于估计音库中候选单元 u_i 与其目标单元 t_i 的差距；拼接代价 $C^c(u_{i-1,u_i})$ 用于估计相邻的两个候选单元 u_{i-1} 与 u_i 在拼接点处的差距，如图 5.8 所示。

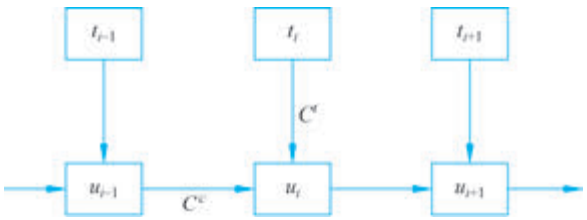

图 5.8 基元选取算法示意图

则用某个候选单元序列 U 去合成目标单元序列 T，需付出的总的代价为

$$C(U,T)=\sum_{i=1}^{n}C^t(u_i,t_i)+\sum_{i=2}^{n}C^c(u_{i-1},u_i)+C^c(S,u_1)+C^c(u_n,S) \tag{5.5}$$

其中，$C^c(S,u_1)$ 和 $C^c(u_n,S)$ 表示起始和结尾音节与静音段的拼接代价，用于定义拼接的边界条件。则最优单元序列为

$$\hat{U}=\underset{U}{\operatorname{argmin}}C(U,T) \tag{5.6}$$

可采用 Viterbi 动态规划算法来求解此式。把目标单元序列看作 HMM 中的状态序列，对每个目标单元来说，它的候选单元就是它所有可能的状态值，结合目标代价和拼接代价对音库中所有可能的候选单元路径进行搜索，以总代价最少的那条路径作为最终用于合成的单元序列。Viterbi 算法的时间复杂度为 $O(m^2n)$，其中 n 为目标单元的数目，m 为每

个目标单元的平均候选单元的数目。因此待合成的句子较长时，为实现实时的合成，需要在运用 Viterbi 搜索之前，对候选单元进行筛选，只允许与目标单元足够相似的候选单元参与搜索。

2）目标代价函数

目标代价函数用于衡量候选单元与目标单元的相似性，这种相似性体现于两方面：音段特征和超音段特征。

从音段特征来分析，虽然不同的候选单元可能具备相似的基频曲线和时长，但它们的音段特征有可能差距较大。例如，比较“好人”中的“好”字与“好天气”中的“好”字，二者的韵律环境有可能非常相似，但“好人”中的“好”字后接声母“r”隶属于发音擦音，同“好”字间协同发音效应较强；而“好天气”中的“好”字后接声母“t”隶属于不送气塞音，同“好”字音协同发音效应较弱。因此这两个“好”字的音段特征差距较为明显。这里，可通过规则代价表的方法来确定不同音段之间进行相互替换时的目标代价的大小。考虑的音素包括前接音节韵母的类型和后接音节声母的类型。

从超音段特征来分析，目标代价可考虑的因素包括：

（1）当前音节的声调。

（2）当前音节的韵律层级位置。

（3）当前音节所属的韵律词及韵律短语长度。

（4）前后静音段的长度。

（5）当前音节的重音。

（6）基频曲线的距离。

（7）音长的差距。

对离散的特征，可指定规则代价表作为该特征的目标代价函数；对连续的特征，如基频可计算基频曲线的欧氏距离作为目标代价函数。

综合音段特征及超音段特征，为每个候选单元及目标单元都生成一个 p 维的特征矢量，则候选单元 t_i 的目标代价为

$$C^t(t_i,u_i)=\sum_{j=1}^{p}\omega_j^t C_j^t(t_i,u_i) \tag{5.7}$$

其中，$C_j^t(t_i,u_i)$是第 j 维分量的目标代价，应归一化于 0～1，ω_j^t 是其所占总目标代价的权重，且 $\sum\omega_j^t=1$。通过调整不同子特征目标代价的权重，可保证在音段特征和超音段特征的联合表现上，候选单元与目标单元最为接近。

3）拼接代价函数

相邻两个候选单元的拼接代价用于衡量拼接点处的不连续性。这种不连续性体现于三方面：频谱的不连续性、基频曲线的不连续性及能量的不连续性。

频谱的不连续性可以通过计算在拼接处相邻两候选样本的某种声学参数的距离来衡量。常用的声学参数包括 Mel Frequency Cepstral Coefficients（MFCCs）和 Line Spectral Frequencies（LSFs）。此处的距离定义应能客观反映人在听觉上对语音不连续的感觉，常用的距离定义包括 Euclidean 距离和 Kullback-Leibler 距离。

能量的不连续性可通过计算拼接时短时能量的差异来衡量。

对中文而言，相邻单元间的基频不连续性不可简单地通过计算拼接处基频的差值来衡量，本书将在 2.4 节中给出一种基于数据驱动的计算基频不连续性的方法。

综上所述，相邻两个候选单元 u_{i-1}，u_i 的拼接代价函数 $C^c(u_{i-1},u_i)$可定义为 q 个子代价 $C_j^c(u_{i-1},u_i)$的加权和，如下式所示：

$$C^c(u_{i-1},u_i)=\sum_{j=1}^{q}\omega_j^c C_j^c(u_{i-1},u_i) \tag{5.8}$$

4）代价函数的训练

可知，目标代价和拼接代价中的权值对合成结果起到至关重要的作用。前面给出了一种基于回归训练(Regression Training)的权重集训练方法，该方法利用线性回归原理，学习给定单元所有样本间的信赖关系，从而获得权重值。具体实现方法如下所示。

(1) 对基元在语料库中的每个候选单元通过以下步骤进行操作：

将该候选单元视为目标单元；

计算目标单元与基元在语料库中其他候选单元之间的目标距离；

挑选最优匹配的前 n 个候选单元，一般 n 取 20；

确定目标单元与前 n 个最优候选单元之间的目标代价 $C_j^t(t_i,u_i)$。

(2) 将目标单元与基元在语料库中其他候选单元之间的目标距离和目标单元与前 n 个最优候选单元之间的目标代价进行存储。

(3) 采用线性拟合算法，将目标单元与前 n 个最优候选单元之间的目标代价通过线性组合来预测目标单元与基元在语料库中其他候选单元之间的目标距离，训练出来的权值就是目标代价的权值。

(4) 对每个基元都采用上述步骤来训练权值。

基于回归训练的权重集训练方法给出了一条自动获取权重的途径，且计算复杂度可以接受，但在训练数据量较少的情况下，得到的结果并不准确。因此目前仍采取自动训练与人工测听相结合的办法设定代价函数中的权值。

5）拼接合成

得到候选基元的最佳路径后，在路径上的单元就可以在略做平滑处理后直接拼接起来。如果没有追求音高和音长的目标值，除了在接点非常有限的平滑处理，没有在合成语音上加入人工的影响，所以听起来非常接近原始的发音人的声音，保持着原来的说话风格。如果需要调整音高和音长，可以采用基音同步叠加算法(PSOLA)进行调整，合成语音也具有较高的保真度。

5.3.3 基于深度学习的拼接语音合成

1）单元选择的拼接语音合成

谷歌公司提出了一种显著改进基于单元选择的拼接语音合成的神经网络模型，如图 5.9 所示。该模型采用基于序列到序列 LSTM 的自动编码器，将每个单元的声学和语言特征压缩为固定大小的矢量(称为嵌入)。通过将目标损失表示为嵌入空间中的 $L2$ 距离，促进了合成单元选择，在提高拼接语音合成质量的同时，保持了较低的计算成本和延迟。

在训练时，声学和语言特征都是可用的，但是推理时只有语言特征。谷歌公司设计了一个网络，该网络能够在训练期间利用网络的输入，但在运行时仍然可以在没有声学特征

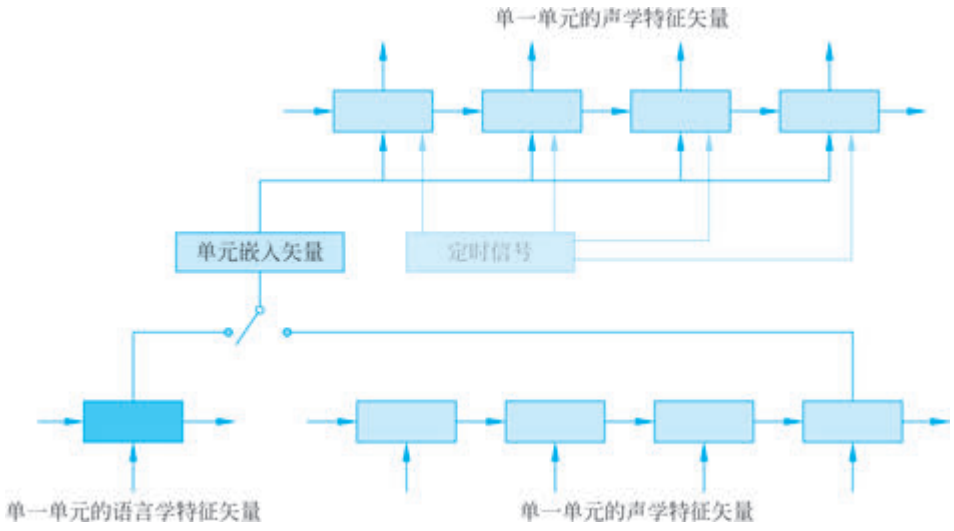

图 5.9　谷歌公司的单元选择拼接语音合成神经网络模型框架

的情况下正常工作。

(1) 学习嵌入：谷歌公司所提出网络模型的拓扑如图 5.9 所示。使用由 LSTM 单元组成的序列到序列自动编码器网络来学习嵌入。该网络由两个编码器组成：第一个编码器对语言序列进行编码，该序列由每个(音素或双音素大小)单元的单个特征矢量组成。它是一种多层循环 LSTM 网络，读取一个输入语言特征矢量并为每个单元输出一个嵌入矢量。第二个对每个单元的声学序列进行编码。它也是一个循环多层 LSTM 网络。其输入是完整单元的参数化声学特征序列，并且在看到输入序列的最终矢量后输出一个嵌入矢量。这是时间瓶颈，其中来自多个时间帧的信息被压缩成单个低维矢量表示。两个编码器的嵌入输出大小相同。插入一个开关，以便解码器可以连接到声学或语言编码器。在训练期间，根据某个固定概率为每个单元随机设置开关。

解码器经过训练，可以根据所连接的解码器的嵌入来估计语音的声学参数。它的输入由重复足够多次的嵌入矢量组成，以匹配单元中的帧数，并且将粗编码定时信号附加到每个帧，该信号告诉网络它通过了多远的单元。

该网络使用随时间的反向传播和随机梯度下降进行训练，并且在解码器的输出处使用平方误差损失。为了进一步鼓励编码器生成相同的嵌入，在损失函数中添加了一个附加项，以最小化两个编码器生成的嵌入之间的平方误差。

(2) 分区嵌入：单元选择系统的特征之一是能够对不同信息流、频谱、非周期性、F0、发声和持续时间的相对重要性进行加权。使用单个解码器将导致将所有这些流编码到嵌入中，从而无法重新加权流。为了实现重新加权，嵌入被划分为单独的流，并且每个分区连接到其自己的解码器，该解码器单独负责预测该流的特征。

(3) 等距嵌入：谷歌公司引入等距嵌入作为附加约束，以便嵌入空间内的 $L2$ 距离即成为单元之间声学距离的直接估计，又在独立的网络训练运行中更加一致。将单元对之间的动态时间扭曲 (DTW) 距离定义为使用 DTW 算法对齐的声学空间中帧对之间的 $L2$ 距离之和。谷歌公司在网络的损失函数中添加一项，使得两个单元的嵌入表示之间的 $L2$ 距离

与相应的 DTW 距离成正比。使用 DTW 对齐小批量中不同句子的音素，以生成 DTW 距离矩阵。相应的 $L2$ 距离矩阵是在音素嵌入之间计算的。这两个矩阵之间的差异被添加到网络的损失函数中以实现最小化。

(4) 最近邻预选：当构建语音合成系统时，语音训练数据中的每个单元嵌入都被存储在数据库中。在语音合成系统运行时，目标句子的语言特征被输入语言编码器中，以获得相应的目标嵌入序列。对于每个目标嵌入，预先从数据库中选择 k 个最近的单元。这些预选的单元被排列成一个网格，并执行维特比搜索，以找到连接成本最小化的最佳单元序列，以满足整体目标和连接成本的最小化要求。目标损失是通过计算语言编码器预测的目标嵌入矢量与数据库中存储的单元嵌入矢量之间的 $L2$ 距离来衡量的。这有助于确保语音合成的准确性和清晰度。

2) 基于混合单元选择的拼接语音合成

苹果公司提出的一种混合单元选择语音合成系统如图 5.10 所示，该系统满足了 Siri 的自然、个性和表现力要求。这一系统已经成功部署在 iOS 和 macOS 等多种语言的桌面和移动设备上，包括 iPhone、iPad、Mac 等。该系统遵循经典的单元选择框架，并充分利用深度学习技术来提高性能。具体而言，系统采用了深度和循环混合密度网络，用于在单元选择过程中预测各自成本的目标和串联参考分布，以提高合成语音的质量。

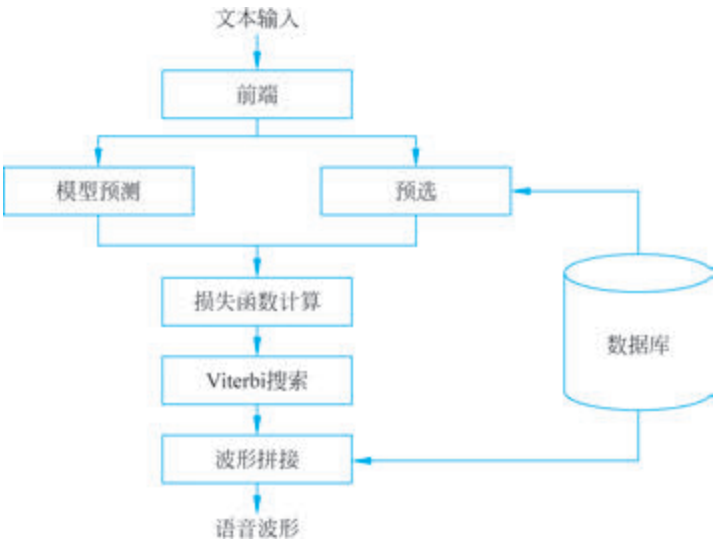

图 5.10　苹果单元选择合成系统框架

苹果设备上 TTS 系统遵循典型的单元选择框架，该框架使用前端生成语言特征、低延迟预选、实现串联的统计模型以及找到最佳单元序列的 Viterbi 搜索的目标成本，以及波形串联以生成最终的合成波形。作为一个混合系统，它受益于基于深度学习的统一声学模型来预测声学和韵律特征分布并实现串联和目标成本。这个框架可以产生比统计参数语音合成更高的质量，并且具有可接受的占用空间和延迟。此外，允许在没有互联网连接的情况下进行合成。苹果针对低延迟和质量进行了一些额外优化。

(1) 前端：也称为文本处理，主要目标是生成原始输入文本的语音转录以及多种语言特征(标点符号、音节、重音)，以指导韵律预测和单元选择步骤产生可理解且自然的语音。

苹果使用传统的、很大程度上基于规则的系统来生成必要的语言特征。

(2) 预选：考虑到有限的设备内存，苹果单元选择合成系统在内存使用方面受到严格限制，远超过典型的桌面或服务器环境。为此，苹果单元选择合成系统采用了强制执行更为严格的内存策略的操作系统。系统广泛地依赖只读内存映射数据，以确保操作系统有效地管理内存资源，并尝试将执行单元选择所需的驻留数据最小化。这一目标通过两步的预选流程来实现。

音素上下文匹配：为每个半音素识别了一组尽可能准确地匹配目标语音上下文的候选单元。这一过程从五音素匹配的子集开始，然后扩展为包括三音素和双音素匹配的子集，最后添加单音素匹配，直到达到或超过候选单元的阈值数量。

指纹匹配：在音素上下文匹配中选出的候选单元中，考虑了每个单元的多个布尔特征，例如单元是否重读、词首/词尾、短语词首/词尾、句子词首/词尾等。这些特征以位集的形式表示为单元指纹。随后，根据上下文特定的目标指纹来评分单元指纹。使用这个分数和上下文搜索的分数，可以将搜索范围缩小到 100 个单元，这些单元将用于维特比搜索。

(3) 模型预测：苹果单元选择合成系统使用概率方法来实现目标和串联成本，以最大限度地减少手动调整。采用深度和循环混合密度网络(MDN)来预测目标和串联分布，分别用于实现目标和串联成本。MDN 提供了一个对条件密度函数进行建模的框架，使用当前时间步的前端语言特征，通过多层非线性变换来预测声学和韵律特征分布。深度前馈 MDN 的公式为

$$p(\boldsymbol{y}_t \mid \boldsymbol{x}_t; \lambda) = \sum_{k=1}^{K} \alpha_k \, \mathcal{N}(\boldsymbol{y}_t; \mu_k; \Sigma_k) \tag{5.9}$$

其中，$\boldsymbol{x}_t$ 和 $\boldsymbol{y}_t$ 是当前时间步输入和输出的特征，λ 表示多层神经网络的模型参数，K 是高斯分量的数量，α_k、μ_k 和 Σ_k 是 a 的权重、均值和协方差高斯分量。

前馈 MDN 不具备对时间依赖性进行建模的能力，这对于建模语音尤其是基频来说非常重要。为此，苹果使用循环混合密度网络对基频进行建模。循环混合密度网络的公式为

$$p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \cdots, \boldsymbol{x}_1; \lambda) = \sum_{k=1}^{K} \alpha_k \, \mathcal{N}(\boldsymbol{y}_t; \mu_k, \Sigma_k^2) \tag{5.10}$$

$p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \cdots, \boldsymbol{x}_1; \lambda)$依赖所有先前的输入特征来预测输出分布，因此具有对顺序数据进行建模的能力。

(4) 长单元：用于以一组预先确定的单元替代合成话语的一部分的方法，这些单元充当单个大单元。在苹果的单元选择合成系统中，长单元的选择基于语音匹配，并利用了机器学习输入和合成数据管道的共同特征。这种方法显著减少了系统中长单元的数量，并将其重点放在极其关键的句子和非语音元素上。由于长单元的使用在语音数据和录制时间方面都具有高成本，因此只有极小一部分合成话语包含了任何预定义的长单元。

(5) 优化：为了使系统在设备上低延迟运行，苹果单元选择合成系统对单元索引进行了优化，并利用了并行化。单元索引对于延迟和占用空间都起着重要作用。苹果单元选择合成系统的两步预选只需要参考语音的上下文表和指纹部分，并且只有前者需要在音素上下文匹配步骤中进行广泛搜索。为了简化上下文查找，数据库中的所有单元都排列在索引中，按其五音素上下文排序，顺序为 < phoneme、predecessor、successor、pre-predecessor、post-successor >。这使系统能够将大多数匹配的子集表示为索引中连续的单元范围，并通

过简单的二分搜索找到子集。

5.4 统计参数语音合成

针对基于大规模语料库的拼接合成方法的弊端，基于参数合成的可训练语音合成方法（Trainable TTS）被提了出来，其中最成功的是 HMM 的可训练语音合成方法（HMM based Speech Synthesis System，HTS）（Tokuda，1995）。近年来，深度神经网络模型被应用于机器学习各个研究领域，并且都取得了相比传统方法显著的优势。基于神经网络的建模方法在统计参数语音合成中的应用也逐步深入，成为语音合成的主流方法。

基于参数合成的可训练语音合成方法的基本思想是，基于一套自动化的流程，根据输入的语音数据进行训练，并形成一个相应的合成系统。该方法基于统计建模和机器学习，根据一定的语音数据进行训练并快速构建合成系统。由于这种方法可以在不需要人工干预的情况下，自动快速地构建合成系统，而且对于不同发音人、不同发音风格，甚至不同语种的依赖性非常小，非常符合多样化语音合成方面的需求，因此早期得到研究人员的认可和重视，并在实际应用中发挥出重要作用。

5.4.1 基于隐马尔可夫的参数语音合成方法

基于 HMM 的语音合成方法主要分为两个阶段：训练阶段和合成阶段。图 5.11 为基于 HMM 的语音合成方法的系统框图。

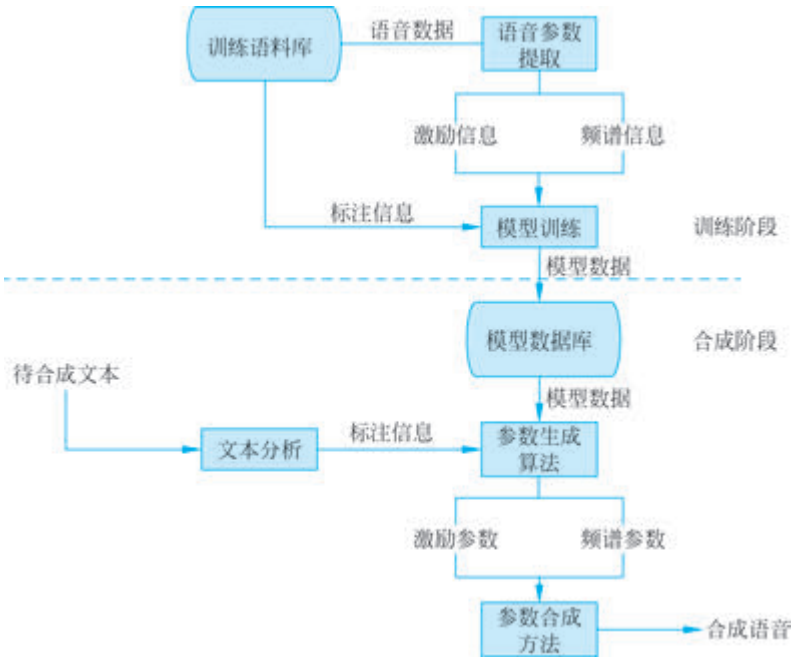

图 5.11 基于 HMM 的语音合成方法的系统框图

在 HMM 训练前，首先要对一些建模参数进行配置，包括建模单元的尺度、模型拓扑结构、状态数目等，还需要进行数据准备。一般而言，训练数据包括两部分：声学数据以及标

注数据，其中声学数据包括频谱信息和激励信息；标注数据主要包括音段切分和韵律标注，现在采用的都是人工标注的。但是与拼接合成的方法不同，其中的切分信息并不是很重要，自动切分的结果基本上就可以满足要求。在拼接合成中，如果切分边界不准确，会直接反映到输出结果的拼接单元上，对合成结果影响较大。但对于可训练的语音合成方法，个别切分边界的不准确会被后面的参数训练过程弱化，因此可训练的语音合成方法对边界错误信息的鲁棒性较高。对于韵律标注，则需要人工进行，不过这部分的工作量相对较小。

除了定义一些 HMM 参数以及准备训练数据以外，模型训练前还有一个重要的工作就是对上下文属性集和用于决策树聚类的问题集进行设计，即根据先验知识来选择一些对声学参数（谱、基频和时长）有一定影响的上下文属性并设计相应的问题集，如前后调、前后声韵母等（Odell，1995）。需要注意的是，这部分工作是与语种相关的。除此之外，整个 HTS 的建模训练和合成流程基本上与语言种类无关。

如图 5.12 所示，基于 HMM 的语音合成系统分为训练部分和合成部分。

在训练部分，首先要对训练语料进行一定的预处理，如上面提到的分析语音参数，对语料进行标注等。之后就是 HMM 的训练。基于 HMM 的语音合成相当于 HMM 语音识别的逆过程。HTS 中 HMM 的训练也与识别中基本相同，分为模型初始化、单个模型训练、结合上下文训练、上下文聚类、聚类后训练等。在得到了 HMM 之后，就是合成部分。

在合成部分，根据输入的文本分析结果，得到对应的模型序列。这些含有上下文信息的模型被串联在一起形成句子模型（Sentence HMM）。之后，根据模型中时长模型的参数分布和要合成语音的长度，得到状态序列。通过 HTS 特有的结合动态特征的参数生成算法，合成出平滑的语音参数序列，通过声学模型还原成语音波形。图 5.12 为基于 HMM 语音合成方法的示意图。

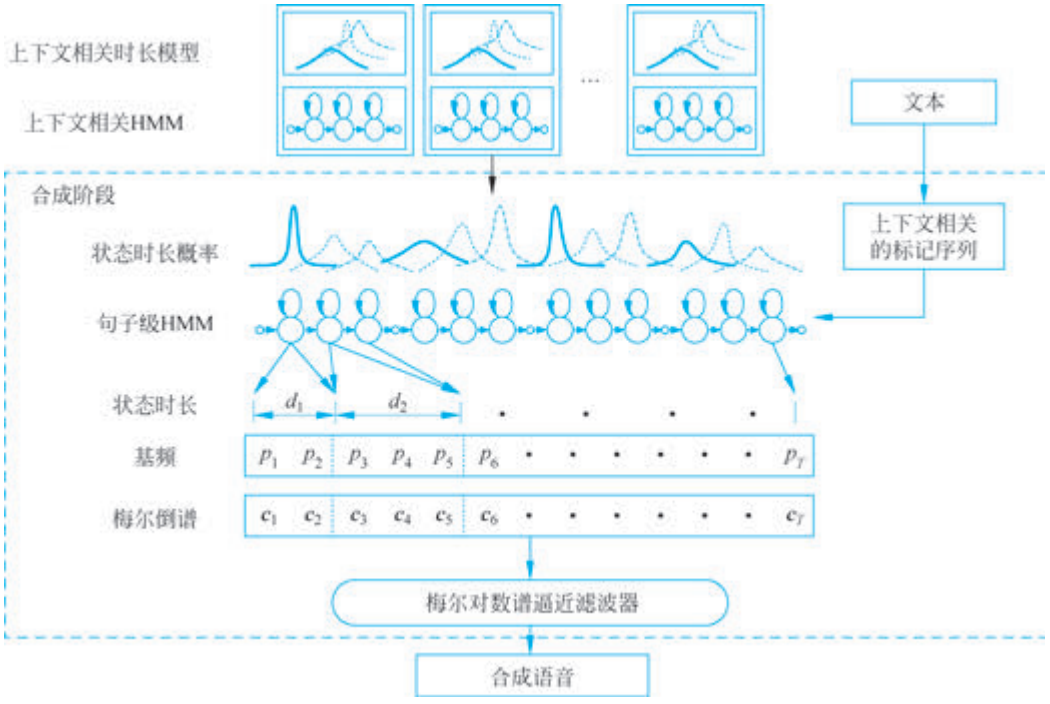

图 5.12　基于 HMM 语音合成方法的示意图

1）基于 HMM 语音合成的相关算法

相对于拼接合成方法，基于 HMM 的语音合成涉及的主要算法有基于多空间分布(Multi-Space Distribution，MSD)的 HMM 模型训练(Tokuda et al.，2002)，结合动态特征的参数生成算法(Tokuda et al.，1995a，1995b，2000，Masuko，1996)。

2）结合动态特征的参数生成算法

如图 5.13 所示，在得到了 HMM 模型序列之后，如何得到参数序列，就是结合动态特征的参数生成算法的任务。在给定 HMM 模型参数 λ 的情况下，求得最优的观测序列 $\boldsymbol{O}=[o_1^{\mathrm{T}},o_2^{\mathrm{T}},\cdots,o_T^{\mathrm{T}}]^{\mathrm{T}}$ 相当于使得下式最大化：

$$\boldsymbol{O}=\underset{O}{\operatorname{argmax}}P(\boldsymbol{O}\mid\lambda)=\underset{O}{\operatorname{argmax}}\sum_{Q}P(Q\mid\lambda)P(\boldsymbol{O}\mid Q,\lambda) \tag{5.11}$$

其中

$$\sum_{Q}P(Q\mid\lambda)P(\boldsymbol{O}\mid Q,\lambda)\cong\max_{Q}P(Q\mid\lambda)P(\boldsymbol{O}\mid Q,\lambda) \tag{5.12}$$

为计算简便，上式可以近似分两步最大化

$$Q_{\max}=\underset{Q}{\operatorname{argmax}}P(Q\mid\lambda) \tag{5.13}$$

$$O_{\max}=\underset{O}{\operatorname{argmax}}P(\boldsymbol{O}\mid Q_{\max},\lambda) \tag{5.14}$$

对于式 $Q_{\max}=\underset{Q}{\operatorname{argmax}}P(Q|\lambda)$，在给定语音长度 T 的情况下，相当于求取

$$\log P(Q\mid\lambda,T)=\sum_{k=1}^{K}\log p_k(d_k) \tag{5.15}$$

其中，d_k 为对应状态的时长。根据时长模型(Yoshimura et al.，1998)：

$$d_k=\xi(k)+\rho\cdot\sigma^2(k) \tag{5.16}$$

$$\rho=\left(T-\sum_{k=1}^{\kappa}\xi(k)\right)\Big/\sum_{k=1}^{\kappa}\sigma^2(k) \tag{5.17}$$

$\xi(k)$、$\sigma^2(k)$为对应状态的时长分布的均值和方差，ρ 为时长放缩因子。

在不考虑动态参数的情况下，

$$\log P(\boldsymbol{O}\mid Q,\lambda)=-\frac{1}{2}\boldsymbol{O}^{\mathrm{T}}\boldsymbol{U}^{-1}\boldsymbol{O}+\boldsymbol{O}^{\mathrm{T}}\boldsymbol{U}^{-1}\boldsymbol{M}+K \tag{5.18}$$

其中

$$\boldsymbol{U}^{-1}=\operatorname{diag}[U_{q_1,i_1}^{-1},U_{q_2,i_2}^{-1},\cdots,U_{q_T,i_T}^{-1}] \tag{5.19}$$

$$\boldsymbol{M}=[\mu_{q_1,i_1}^{\mathrm{T}},\mu_{q_2,i_2}^{\mathrm{T}},\cdots,\mu_{q_T,i_T}^{\mathrm{T}}]^{\mathrm{T}} \tag{5.20}$$

分别为协方差和均值。

最大似然输出的观察值 $\boldsymbol{O}$ 应为各个状态的均值：

$$\frac{\partial\log P(\boldsymbol{O}\mid Q,\lambda)}{\partial\boldsymbol{O}}=0\Rightarrow\boldsymbol{O}=\boldsymbol{M} \tag{5.21}$$

$\boldsymbol{M}$ 为对应状态序列的均值序列，如图 5.13 所示。

加入动态参数之后，观察值包括静态参数和动态参数：$o_t=[c_t^{\mathrm{T}},\Delta c_t^{\mathrm{T}},\Delta^2 c_t^{\mathrm{T}}]^{\mathrm{T}}$，$\boldsymbol{O}=\boldsymbol{W}C$。$\boldsymbol{W}$ 为计算动态参数的窗矩阵(Tokuda，1995)。式$\frac{\partial\log P(\boldsymbol{O}|Q,\lambda)}{\partial\boldsymbol{O}}=0\Rightarrow\boldsymbol{O}=\boldsymbol{M}$ 变为

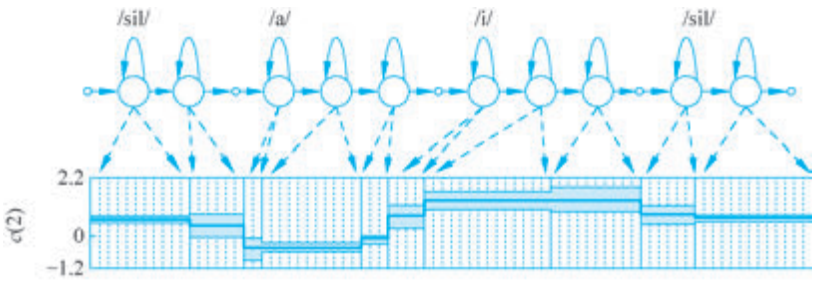

图 5.13　不考虑动态参数下的输出观察序列(Tokuda,2006)

$$\frac{\partial \log P(\boldsymbol{WC} \mid Q,\lambda)}{\partial C}=0 \Rightarrow \boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}^{-1}\boldsymbol{WC}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{U}^{-1}\boldsymbol{M}^{\mathrm{T}} \tag{5.22}$$

解上面的方程可以得到观察序列,即输出的合成语音参数序列。图 5.14 给出了一个连续的输出参数序列合成出的轨迹频谱的例子。

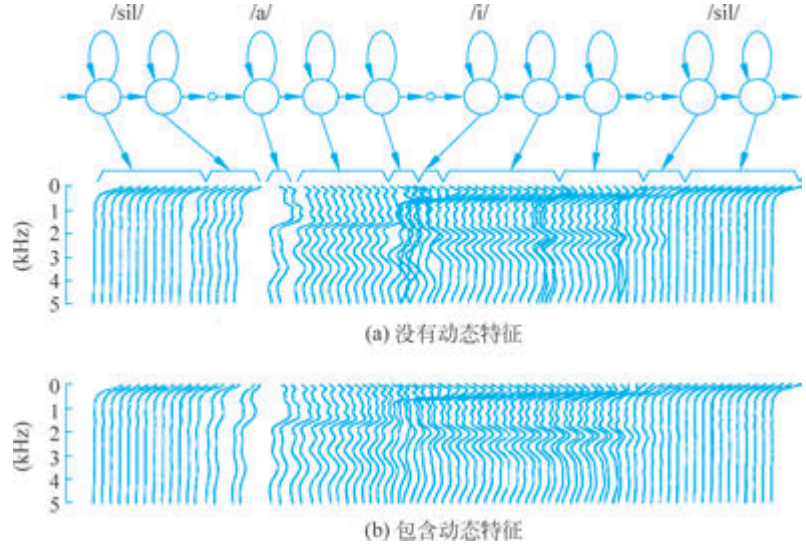

图 5.14　合成出的轨迹频谱(Tokuda,1996)

事实上,通过以上方法生成的参数轨迹相比于原始语音分析得到的参数轨迹显得过于平滑,因而最终的合成音起伏偏少、听起来发闷。为了缓解这个问题,Toda 等(2007)提出了兼顾全局方差(Global Variance,GV)的参数生成方法。传统的参数生成算法的目标是在给定 HMM 序列时最大化输出概率,它的效果是使参数序列尽可能接近高斯成分的均值;即使增加了动态参数的约束,生成的参数序列轨迹也是高斯成分的均值轨迹的小幅度简单平滑,因此参数轨迹在高斯均值轨迹附近小幅震荡,它的方差相比原始语音的轨迹小很多。GV 同时考虑 HMM 的似然值和生成的参数轨迹的方差,生成的参数轨迹具有较大的 GV,更接近真实语音的情况。

3) 基于多空间分布的 HMM 模型的基频建模

已有的一些基频建模方法,例如二次曲线模型、Fujisaki 模型和 Pitch Target 模型等,都可以对基频进行比较好的建模并指导韵律预测。基于 HMM 模型的语音合成中,为了考

虑基频信息与频谱信息的状态对齐，基频信息和频谱信息被统一建模。然而，由于基频数据的维数在时间轴上不固定（在清音段或静音段为一个符号，在浊音段为一维实数，见图 5.15），传统的 HMM 无法对其进行建模。

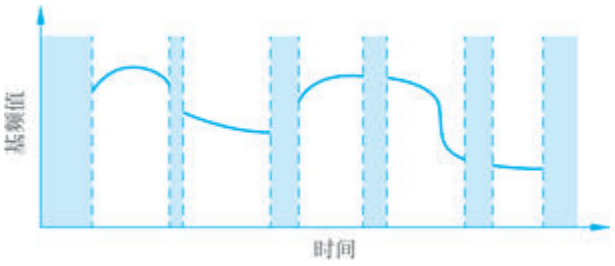

图 5.15　基频曲线示意图（维度随时间变化）

为了将基频信息和频谱信息统一建模，基于多空间分布的 HMM 模型被提了出来（Tokuda，2000）。

首先，假设样本空间$\boldsymbol{\Omega}$是由 G 个不同维度的空间组成的：

$$\boldsymbol{\Omega}=\bigcup_{g=1}^{G}\boldsymbol{\Omega}_g \tag{5.23}$$

其中，$\boldsymbol{\Omega}_g$ 为一个 n_g 维的实空间 $\mathbf{R}^{n_g}$。这样，每个子空间实际上可以有不同的维度，一些子空间的维度可以相同，见图 5.16。对于每个空间$\boldsymbol{\Omega}_g$，有一个计权值 ω_g，并且 $\sum_{g=1}^{G}\omega_g=1$。如果 $n_g>0$，则空间$\boldsymbol{\Omega}_g$ 有一个概率密度函数 $N_g(x)$，$x\in\mathbf{R}^{n_g}$。对于样本空间的一个样本矢量 $\boldsymbol{o}$，定义为包含空间索引矢量 $\boldsymbol{X}$ 和属于 $\mathbf{R}^n$ 空间的观察矢量 $\boldsymbol{x}$，即 $\boldsymbol{o}=(\boldsymbol{X},\boldsymbol{x})$。

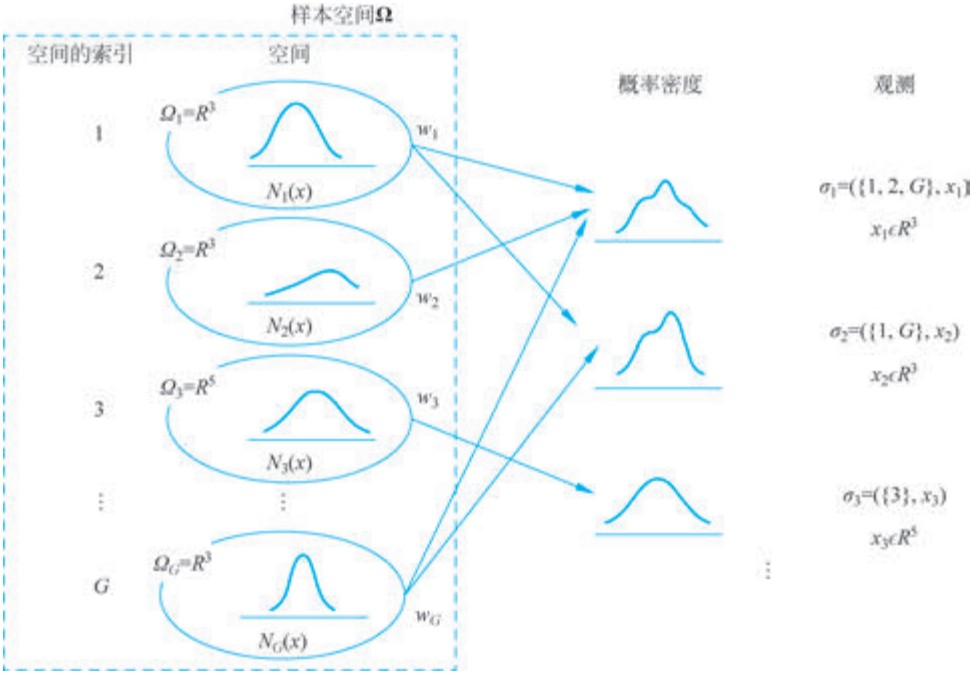

图 5.16　基于多空间分布（MSD）示意图（Tokuda，1999）

一个观测值 o 的输出概率，可以表示为

$$b(o)=\sum_{g\in S(o)}\omega_g N(V(o)) \tag{5.24}$$

其中，$S(o)=X$，$V(o)=x$。定义当 $n_g=0$ 时，$N_g(x)\equiv 1$。

由于在浊音段时基频为一维连续输出值，在静音段和清音段为一个符号，没有输出值，因此在基频建模时，可以设置 $n_g>0(g=1,2,\cdots,G-1)$，$n_g=0(g=G)$。实际建模时 G 常取值 2，即 MSD-HMM 中有两个空间，一个为一维连续空间用来描述浊音基频，另一个为零维空间用来标识静音和清音（如图 5.17 所示）（Tokuda，1999）。

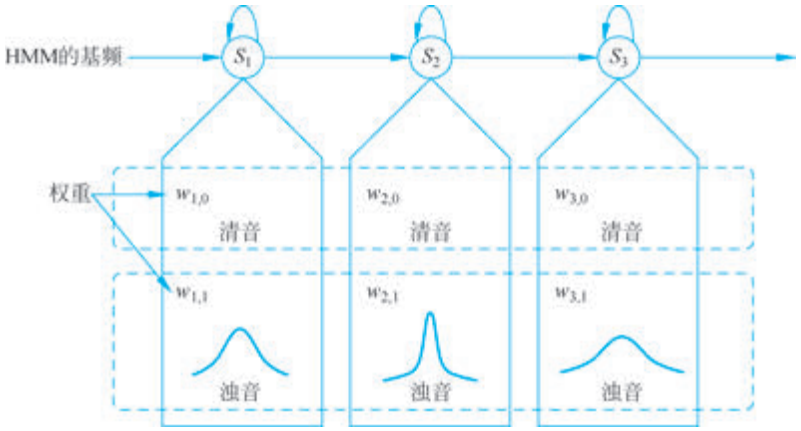

图 5.17　基于 MSD-HMM 的基频建模

实际的基于 HMM 的语音合成系统，将频谱参数和基频参数结合在一起。加上上面提到的动态参数，所有参数的构成如图 5.18 所示。

图 5.18　基于 HMM 的语音合成的参数构成

其中，第一个参数流为谱参数流，包括频谱参数的静态参数，一阶二阶差分。这一参数流为连续分布。后面第二、三、四参数流分别为基频参数的静态和一阶二阶差分，为多空间分布。值得指出的是，参数的训练是频谱参数和基频参数是组合在一起训练，但是基于上下文的决策树聚类时，由于影响两者的因素不同，是分开进行聚类的。具体的模型参数估计算法可参考相关文献（Tokuda，2002）。

5.4.2 基于深度学习的参数语音合成方法

深度学习在语音领域最具标志性的应用是 DNN-HMM 混合语音识别框架的提出，如图 5.19 所示，该系统显著地降低了识别错误率。研究者开始意识到深度学习在语音领域的巨大应用潜力，深度学习逐渐被应用到语音的各个领域。尽管 DNN-HMM 的混合语音识别框架能带来显著的性能提升，但是搭建该框架是一个非常复杂、环节众多且需要大量专家知识的过程。此外，HMM 仍然有一阶马尔可夫假设，这与 RNN 的长时记忆能力相矛盾。为了充分利用神经网络的建模能力，研究者自 2014 年开始研究不依赖 HMM 的结构更简单的端到端语音识别框架。在该框架中，深度神经网络模型直接对语音序列建模，无需太多假设，已经取得了业内的最佳水平。受语音识别研究的启发，深度学习技术开始在统计参数语音合成领域崭露头角。一方面，在传统的管道式框架中，基于神经网络的声学建模技术正在逐渐取代基于 HMM 的声学建模成为主流的建模方法，并得到了长足的发展；另一方面，强有力模型的提出推动着端到端声学建模的发展，端到端的统计参数语音合成正在吸引研究者的注意，并且取得了突破性的进展。

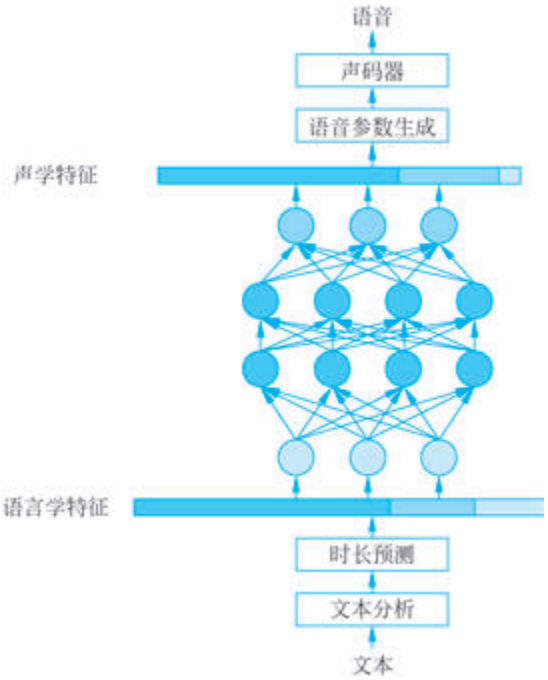

图 5.19 基于深度学习的管道式语音合成系统框架

基于 HMM 的参数语音合成的局限主要是由以下两方面因素导致的：声学建模的不充分性以及参数生成时的过度平滑问题。针对这两个问题，研究者利用深度学习技术做了大量改进。

基于 HMM 的参数语音合成对声学建模的不充分性主要体现在，对上下文相关的音子进行决策树聚类有一定的局限性：一是从深度学习的观点看，决策树是一个浅层的简单函数，并不能建模复杂的输入（上下文语言学标签）输出（语音参数）关系；二是决策树会分割

输入空间以及训练数据，每个叶子节点由相对应的训练数据建模，这会导致数据倾斜问题，然而，一个叶子节点的数据与其他节点的建模也相关。为了消除对决策树依赖，提高声学模型精度，Heiga Zen 提出了基于全连接神经网络的声学建模技术。该方法以上下文语言学标签的矢量编码作为输入，直接生成声学特征，通过神经网络的建模能力有效表征了输入输出的复杂映射关系，并且避免了声学特征空间的划分。由于该方法提出的系统结构简单高效，因此得到了广泛的研究与应用。在此基础上，研究者对模型进行了进一步改进。为了能够建立上下文语言学序列的长期相关性，Fan 将双向长短时记忆循环神经网络(Bidirectional Long Short-Term Memory Recurrent Neural Network，BLSTM-RNN)应用到声学建模中。FCN 假设每帧特征是独立的，这忽略了语音的序列本质。在 BLSTM-RNN 结构中，当前输出帧不仅与左侧输入有关，还考虑了与右侧输入的关系，这样在考虑整个句子上下文的基础上，建模的准确度得到了提高。

基于统计参数的语音合成与基于大规模语料库的拼接合成相比，互相的优缺点如表 5.3 所示。

表 5.3　拼接合成与统计参数语音合成的优缺点

拼接合成	统计参数合成
优　　点	优　　点
直接语音拼接合成，音质很高，接近真实语音	生成的语音平滑稳定，整体效果较强。参数合成易于根据需要调整输出语音参数。占用资源小，易于应用于嵌入式系统
缺　　点	缺　　点
拼接点处不连续，平滑问题比较难解决。结果较难改变，占用资源较大	参数合成语音，音质相对较差，有机器声。由于基于统计方法，有过平滑现象

相对于拼接合成语音系统，参数系统具有灵活，输出多样，资源占用率小，构建代价低的优点。制约参数方法发展的主要瓶颈在于输出结果的音质上。由于用到的是参数合成，合成出的语音有一定的音质下降。同时，由于参数合成中用到了统计方法，导致输出的语音参数存在过平滑现象，这同样是输出语音音质下降的原因。

5.5　局部端到端语音合成

尽管基于深度学习的统计参数语音合成取得较好因依赖人工设计特征，音质较低且难以捕捉复杂的时序关系，而局部端到端语音合成模型通过自动特征提取，直接从文本生成高质量、自然的语音，提升了模型的表现力和简化了流程架构。局部端到端语音合成主要分为两类：声学模型端到端语音合成与声码器端到端语音合成。

5.5.1　声学模型端到端语音合成

声学模型端到端语音合成是指文本分析和声学模型由一个模型建模，输入文本序列输出声学参数，然后送入一个声码器生成语音波形，如图 5.20 所示。声码器端到端语音合成是指声学模型和声码器由一个模型建模，需要一个额外的文本分析模块对文本进行分析处理，再送入后端模型生成语音波形。

图 5.20　声学模型端到端语音合成

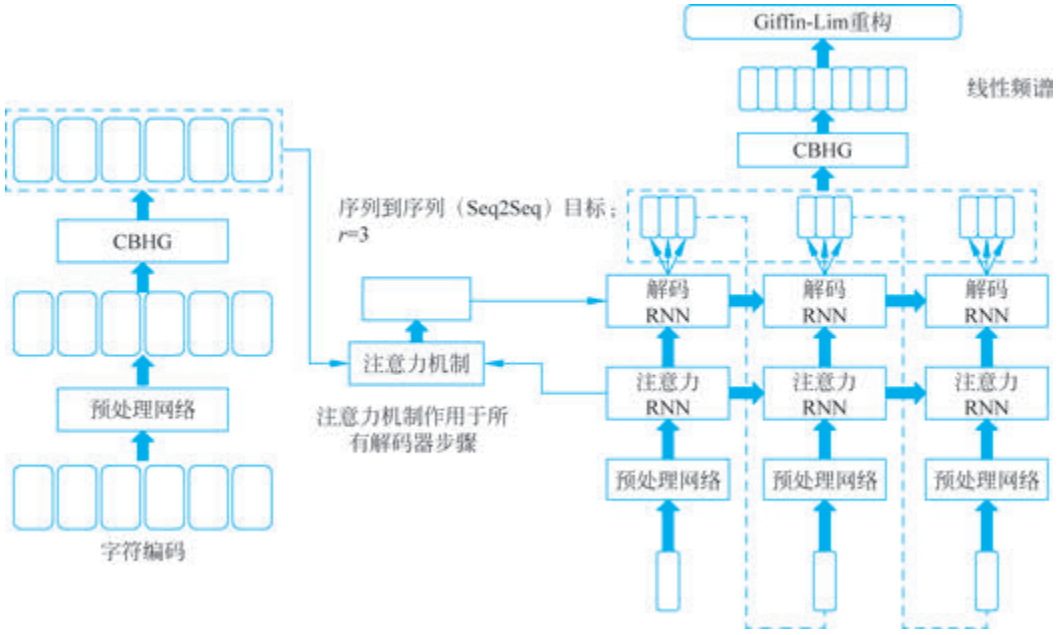

图 5.21　Tacotron 框架

1. Tacotron

Tacotron 的骨干部分是一个有注意力机制的 Seq2Seq 模型。如图 5.21 所示描绘了该模型架构，它包含一个编码器、一个基于注意力机制的解码器和一个后处理网络。从高层面上说，模型把字符作为输入，产生的声谱帧数据随后被转换成波形。下面详细描述这些组件。

编码器。编码器的目的是提取文本的鲁棒序列表达。使用高效的编码器实现可以提取出更具表达力的高级别表示，从而有助于模型泛化。编码器的输入是字符序列或者音素序列（对于汉语，输入可以为带调的声韵母序列），输入的每个字符或者音素都是 One-Hot 矢量并被嵌入一个连续矢量中。然后对每个字符矢量施加一组非线性变换，统称为“Pre-Net”。此外，在实际系统中，一般使用带 Dropout 的瓶颈层（Bottleneck Layer）作为 Pre-Net 以帮助收敛并提高泛化能力。CBHG（Convolutional Bank，Highway networks，GRU）模块将 Pre-Net 的输出变换成编码器的最终表达，并传给后续的注意力模块。该模块中的卷积堆能充分捕捉输入序列中不同层次的 N-gram 信息及韵律模式，提高发音准确度和自然度。在 Tacotron 中，该模块可以直接处理英文字符序列。我们发现基于 CBHG 的编码器不仅减少了过拟合，它还能比标准的多层 RNN 编码器产生更少的错音。

CBHG 模块如图 5.22 所示，包含一个一维卷积滤波器组，后跟一个高速公路网络（Srivastava et al.，2015）和一个双向门控循环单元（GRU）（Chung et al.，2014）循环神经网

络(RNN)。CBHG 是一个强大的模块以提取序列的特征表达。首先在输入序列上用 K 组一维卷积核进行卷积,这里第 k 组包含 Ck 个宽度是 k($k=1,2,\cdots,K$)的卷积核。这些卷积核显式地对局部上下文信息进行建模(类似于 Unigrams,Bigrams,直到 K-grams)。卷积输出结果被堆叠在一起然后再沿时间做最大池化以增加局部不变性,注意我们令步长为 1 以维持时间方向的原始分辨率。然后把得到的结果序列传给几个定长一维卷积,其输出结果通过残差连接(He et al.,2016)和原始输入序列相叠加。所有卷积层都使用了批标准化(Ioffe and Szegedy,2015)。卷积输出被送入一个多层高速公路网络来提取高层特征。最上层我们堆叠一个双向 GRU RNN 用来前后双向提取序列特征。CBHG 是受机器翻译(Lee et al.,2016)论文的启发,但与之不同之处包括使用非因果卷积,批标准化,残差连接以及步长为 1 的最大池化处理。这些修改能够提高模型的泛化能力。

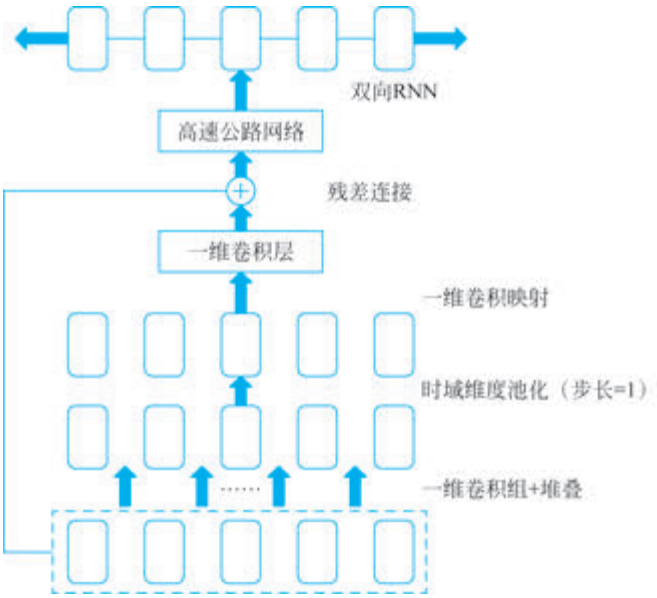

图 5.22　CBHG 框架

解码器。这里使用基于内容的注意力解码器(Vinyals et al.,2015),在这个解码器中,一个有状态的循环层在每个时间步骤上都产生一次注意点查询,再把上下文矢量和 Attention RNN 单元的输出拼接在一起,作为解码器 RNN 的输入。使用带有纵向残差连接的 GRUs 堆栈(Wu et al.,2016)作为解码器,值得注意的是,这里的残差连接能够加速模型收敛。另外,解码器的目标输出,也是一个重要的设计选择。虽然我们可以直接预测原始线谱图,但这对于学习语音信号和原始文本对齐的目标是一个高度冗余的表示。因为这个冗余,我们为 Seq2Seq 解码和波形合成选择了一个不同的目标输出。语音合成作为一个可训练或者可确定的逆向过程,只要能够提供足够的语音可理解性和足够的韵律信息,Seq2Seq 的目标输出就可以被大幅度压缩。尽管类似倒谱这样更小的带宽或者更简洁的目标输出也可行,但现在一般采用带宽为 80 的 Mel 频谱作为解码器的目标输出。我们使用了一个后处理网络把 Seq2Seq 的目标输出转换为线性谱,再通过波形生成模块来生成语音波形。解码器的第一步是在一个“全零帧”上开始调节。在推断时,解码器的第 t 步处理,预

测结果的最后一帧被作为解码器第 $t+1$ 步的输入。注意这里选择最后一帧输入到下一步处理中只是一种选择而已，也可以选择一组 r 帧的全部作为下一步的输入。

后处理网络。其任务是把 Seq2Seq 的输出转换成可以被合成为波形的目标表达。在原始端到端模型被提出来时，使用 Griffin-Lim 作合成器，因此后处理网络要学习的是如何预测在线性频率刻度上采样的频谱幅度（线性谱）。构建后处理网络的另一个动机是它可以看到全体解码结果序列，对比普通 Seq2Seq 总是从左到右运行，它可以获得前后双向信息用于纠正单帧预测错误。这里一般使用 CBHG 模块作为后处理网络，尽管一个更简单的架构可能也会工作得很好。后处理网络的概念是高度通用的，它可以用来预测不同的目标输出如声码器参数，也可以作为像 WaveNet 那样的神经声码器（van den Oord et al.，2016；Mehri et al.，2016；Arik et al.，2017）来直接合成波形样本。

如图 5.23 所示，Tacotron2 系统由两个组件组成。具有注意力的循环序列到序列特征预测网络，它根据输入字符序列预测梅尔频谱图帧序列的修改版本 WaveNet，根据预测的梅尔频谱图帧生成时域波形样本。

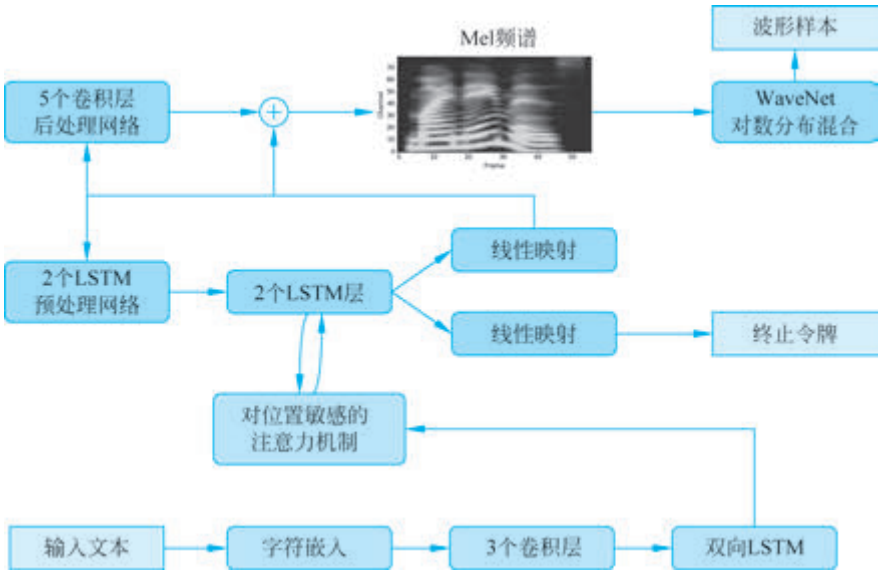

图 5.23　Tacotron2 框架

频谱图预测网络由具有注意力机制的编码器和解码器组成。编码器将字符序列转换为隐藏特征表示形式，解码器使用该表示形式来预测频谱图。编码器的输出被注意力网络使用，该网络将完整的编码序列总结为每个解码器输出步骤的固定长度的上下文矢量。使用了位置敏感注意力，它扩展了加性注意机制，使用了来自先前解码器时间步的累积注意力权重作为附加特征。解码器是一个自回归递归神经网络，它逐帧从编码的输入序列中预测 Mel 频谱图。解码器 LSTM 输出和注意力上下文的串联被投影到一个标量，并通过一个 Sigmoid 激活函数进行传递，以预测输出序列是否已经完成。

最后使用修改过的 WaveNet 架构将 Mel 频谱图特征表示反转为时域波形样本。

2. TransformerTTS

尽管端到端的神经文本转语音(TTS)方法(如 Tacotron2)被提出并取得了最先进的性能，但仍然存在两个问题：训练和推理过程效率低；难以使用当前循环神经网络(RNNs)对长期依赖进行建模。

受 Transformer 网络在神经机器翻译(Neural Machine Translation，NMT)中成功的启发，TransformerTTS(图 5.24)引入并改进了多头注意力机制，以替代 Tacotron2 中的 RNN 结构和原始注意力机制。通过多头自注意力的帮助，编码器和解码器中的隐藏状态可以并行地构建，从而提高训练效率。同时，任意两个不同时刻的输入之间可以直接通过自注意力机制进行连接，有效解决了长程依赖问题。

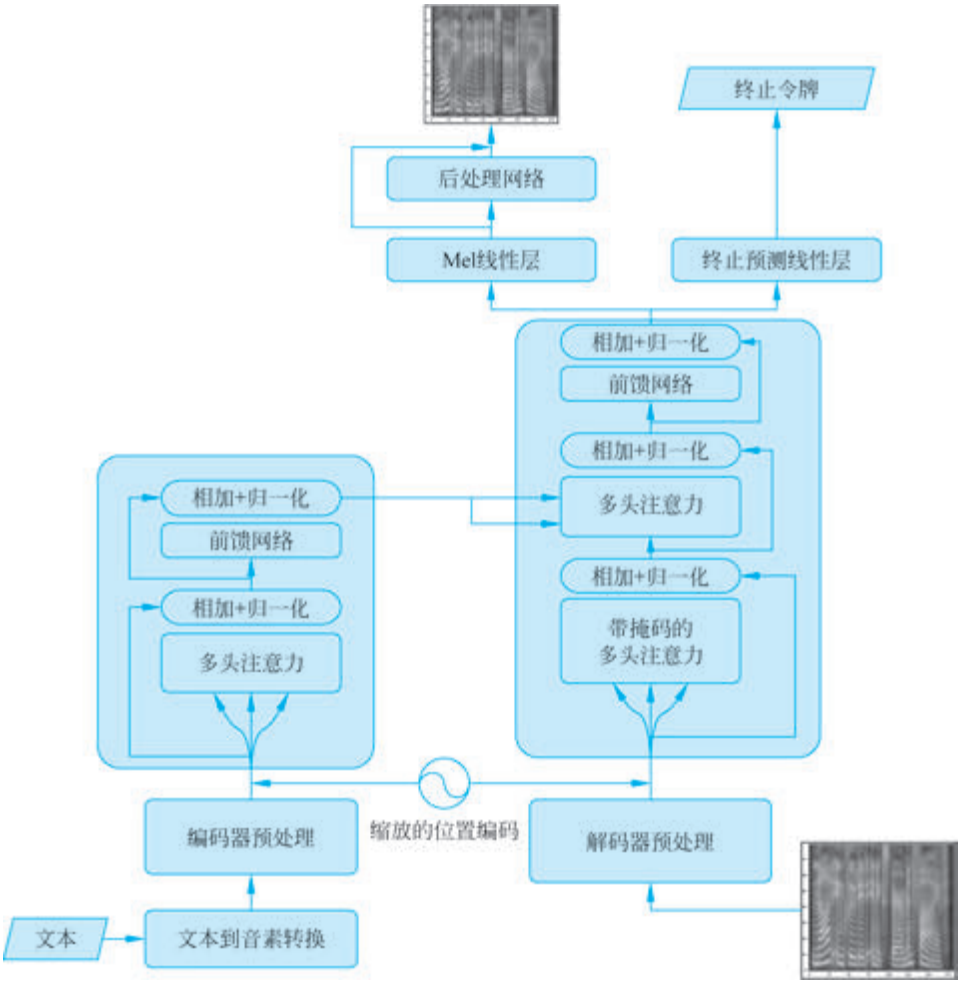

图 5.24　TransformerTTS 框架

与 RNN 的模型相比，使用 Transformer 进行 TTS 有两个优点。首先，通过去除循环连接，它使得训练可以并行进行，因为解码器的输入序列的帧可以同时提供。其次，自注意力机制允许将整个序列的全局上下文引入每个输入帧中，直接建立起长距离依赖关系。最

后，Transformer 将信号在输入和输出序列中任意位置之间的路径长度缩短为 1。这对于神经 TTS 模型非常有帮助，如合成波形的韵律不仅仅依赖邻近的几个单词，还依赖句子级的语义信息。

(1) 位置编码：Transformer 不包含任何循环和卷积操作，因此如果我们对编码器或解码器的输入序列进行重新排序，将得到相同的输出。为了考虑序列的顺序，我们通过三角位置嵌入将有关帧相对或绝对位置的信息注入。

$$\mathrm{PE}(\mathrm{pos},2i)=\sin\left(\frac{\mathrm{pos}}{1000^{\frac{2i}{d_{\mathrm{model}}}}}\right) \tag{5.25}$$

$$\mathrm{PE}(\mathrm{pos},2i+1)=\cos\left(\frac{\mathrm{pos}}{10000^{\frac{2i}{d_{\mathrm{model}}}}}\right) \tag{5.26}$$

其中，pos 是时间步索引，$2i$ 和 $2i+1$ 是通道索引，d_{model} 是每个帧的矢量维度。在 NMT 中，源语言和目标语言的嵌入来自语言空间，所以这些嵌入的尺度是相似的。然而，在 TTS 场景中，源域是文本，目标域是 Mel 频谱图，因此使用固定的位置嵌入可能会对编码器和解码器的预网络施加严重的约束。我们使用可训练的权重来应用这些三角位置嵌入，以使这些嵌入能够适应编码器和解码器的预网络输出的尺度：

$$x_i=\mathrm{prenet}(\mathrm{phoneme}_i)+\alpha\mathrm{PE}(i) \tag{5.27}$$

(2) 编码器预网络：在 TransformerTTS 模型中，将音素序列输入同一网络中，这个网络被称为“编码器预网络”。每个音素都有一个可训练的 512 维嵌入，每个卷积层的输出有 512 个通道，接着是批归一化和 ReLU 激活函数，还有一个 Dropout 层。此外，在最终 ReLU 激活之后，添加了一个线性投影，因为 ReLU 的输出范围是$[0,+\infty)$，而这些三角位置嵌入的每个维度都在$[-1,1]$上。在非负嵌入上添加以 0 为中心的位置信息会导致波动不在原点附近，从而损害模型性能，因此，添加了一个线性投影以保持中心的一致性。

(3) 解码器预网络：Mel 频谱图首先输入一个由两个全连接层组成的神经网络(每个层有 256 个隐藏单元)，该网络使用 ReLU 激活函数，被称为“解码器预网络”，在 TTS 系统中起着重要作用。音素具有可训练的嵌入，因此它们的子空间是自适应的，而 Mel 频谱图的子空间是固定的。解码器预网络的责任是将 Mel 频谱图投影到与音素嵌入相同的子空间，以便可以衡量相似性，从而使注意力机制能够起作用。

(4) 编码器：将 Tacotron2 中双向 RNN 替换为 Transformer 编码器。与原始的双向 RNN 相比，多头注意力将一个注意力分割成多个子空间，以便能够从多个不同的角度建模帧之间的关系，并直接建立起任意两个帧之间的长时依赖，因此每个帧都考虑了整个序列的全局上下文。这在合成音频韵律中尤为重要，另外，使用多头注意力而不是原始的双向 RNN 可以实现并行计算，提高训练速度。

(5) 解码器：使用 Transformer 解码器，一是添加自注意力，二是使用多头注意力代替位置敏感注意力。多头注意力可以从多个角度集成编码器隐藏状态并产生更好的上下文矢量。Mel 线性层、停止线性层和后网络与 Tacotron2 一样，使用两个不同的线性投影来分别预测 Mel 频谱图和停止标记，并使用一个 5 层 CNN 来产生一个残差以改善 Mel 频谱图的重建。在计算二元交叉熵损失时，对尾部的正停止标记施加正权重(5.0～8.0)，从而有效解决了这个正负样本不均匀的问题。

3. FastSpeech1/2

之前基于神经网络的 TTS 系统，Mel 频谱图是自回归生成的。由于 Mel 频谱图的长序列和自回归性质，这些系统面临着几个挑战：

(1) Mel 频谱图生成的推理速度慢。尽管基于 CNN 和 Transformer TTS 可以加快基于 RNN 的模型的训练速度，但所有模型都会根据先前生成的 Mel 频谱图生成 Mel 频谱图，并且推理速度较慢，因为给定 Mel 频谱图序列的长度通常为数百或数千。

(2) 合成语音通常不稳健。由于错误传播以及自回归生成中文本和语音之间的错误注意对齐，生成的 Mel 频谱图通常存在单词跳过和重复问题。

(3) 合成语音缺乏可控性。以前的自回归模型自动生成一张一张的 Mel 频谱图，没有明确利用文本和语音之间的对齐。因此，通常很难直接控制自回归生成中的语音速度和韵律。

考虑到文本和语音之间的单调对齐，为了加速 Mel 声谱图的生成，FastSpeech 以文本(音素)序列作为输入并非自回归地生成 Mel 声谱图。它采用基于 Transformer 中的自注意力和一维卷积的前馈网络。由于 Mel 频谱图序列比其对应的音素序列长得多，为了解决两个序列之间长度不匹配的问题，FastSpeech 采用长度调节器，根据音素持续时间对音素序列进行上采样(即每个音素对应的 Mel 频谱图的数量)以匹配 Mel 频谱图序列的长度。该调节器建立在音素持续时间预测器的基础上，该预测器可以预测每个音素的持续时间。

FastSpeech 是基于 Transformer 中的自注意力和一维卷积的前馈结构。这个结构称为前馈 Transformer(FFT)，如图 5.25(a)所示。前馈 Transformer 将多个 FFT 块堆叠起来，用于将音素转换为 Mel 频谱图，音素端有 N 个块，Mel 频谱图端也有 N 个块，之间还有一个长度调节器，用于连接音素序列和 Mel 频谱图序列之间的长度差异。每个 FFT 块由自注意力和一维卷积网络组成，如图 5.25(b)所示。自注意力网络包括多头注意力以提取交叉位置信息。

长度调节器(如图 5.25(c)所示)用于解决前馈 Transformer 中音素序列和 Mel 频谱图序列之间长度不匹配的问题，同时也用于控制语音速度和部分韵律。音素序列的长度通常小于对应的 Mel 频谱图序列的长度，而且每个音素对应于几个 Mel 频谱图。将对应于一个音素的 Mel 频谱图的长度称为音素时长。基于音素时长 d，长度调节器将音素序列的隐藏状态扩展 d 倍，使得隐藏状态的总长度等于 Mel 频谱图的长度。将音素序列的隐藏状态表示为 $H_{\text{pho}}=[h_1,h_2,\cdots,h_n]$，其中 n 是序列的长度。将音素时长序列表示为 $D=[d_1,d_2,\cdots,d_n]$，其中，$\sum_{i=1}^{n}d_i=m$，m 是 Mel 频谱图序列的长度。将长度调节器表示为 LR。

$$H_{\text{mel}}=\text{LR}(H_{\text{pho}},D,\alpha) \tag{5.28}$$

其中，α 是一个超参数，用于确定扩展序列 H_{mel} 的长度，从而控制语速。

音素持续时间预测对于长度调节器非常重要。如图 5.25(d)所示，持续时间预测器由一个具有 ReLU 激活函数的 2 层一维卷积网络组成，每一层后跟层归一化和丢弃层，以及一个额外的线性层输出一个标量，即预测的音素持续时间。注意，该模块叠加在音素端的 FFT 块上，并与 FastSpeech 模型一起进行联合训练，以使用均方差(MSE)损失预测每个音素的 Mel 频谱图长度。在对数域中预测持续时间，这使得它们更加符合高斯分布并且更易于训练。请注意，经过训练的持续时间预测器仅在 TTS 推理阶段中使用，因为我们可以直

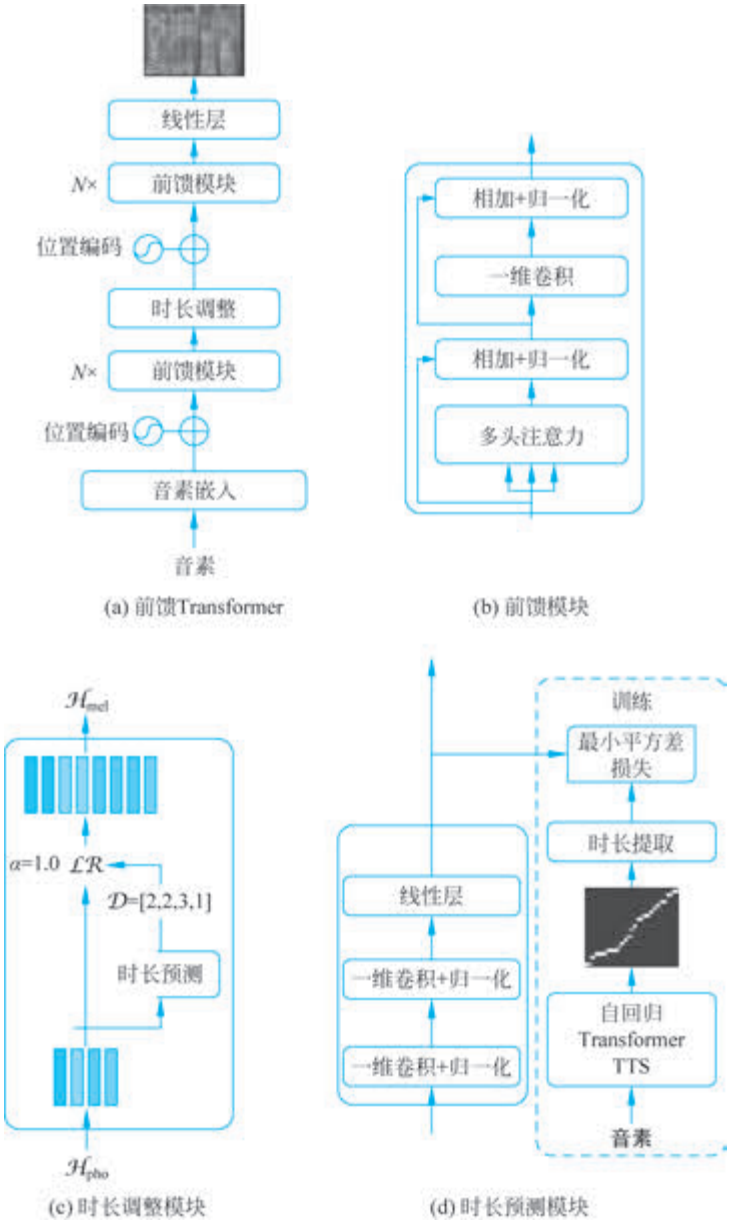

图 5.25　FastSpeech 框架

接使用从自回归教师模型中提取的音素持续时间进行训练。为了训练持续时间预测器，从自回归教师 TTS 模型中提取真实的音素持续时间，如图 5.25(d)所示。描述详细步骤如下：

首先，训练一个基于自回归编码器-注意力-解码器的 TransformerTTS 模型。

然后，对于每个训练序列对，从训练过的教师模型中提取编码器-解码器注意力对齐。由于多头自注意力的存在，会有多个注意力对齐，并且并非所有注意力头都表现出对角线性质(音素和 Mel 频谱图序列是单调对齐的)。焦点率 F 用来衡量一个注意力头距离对角线的程度：$F=\frac{1}{S}\sum_{s=1}^{S}\max_{a\leqslant t\leqslant T}a_{s,t}$，其中 S 和 T 分别是真实 Mel 频谱图和音素的长度，

$a_{s,t}$ 表示注意力矩阵中第 s 行第 t 列的元素。计算每个注意力头的焦点率，并选择具有最大 F 的头作为注意力对齐。

最后，根据持续时间提取器，提取音素持续时间序列 $D=[d_1,d_2,\cdots,d_n]$，其中

$$d_i=\sum_{s=1}^{S}[\text{argmax}_t \quad a_{s,t}=i] \tag{5.29}$$

FastSpeech 具有以下几个缺点：教师-学生蒸馏流程复杂耗时；从教师模型中提取的持续时间不够准确，从教师模型中蒸馏出的目标 Mel 频谱由于数据简化而导致信息损失，这两点都限制了语音质量。

FastSpeech 2 解决了 FastSpeech 中的问题，并通过以下方式更好地解决 TTS 中的一对多映射问题：①直接使用真实的目标进行训练，而不是从教师模型简化的输出；②引入更多的语音变体信息（例如语调、能量和更准确的持续时间）作为条件输入。具体而言，我们从语音波形中提取持续时间、语调和能量，直接将它们作为条件输入进行训练，并在推理过程中使用预测的值。在此基础上进一步设计了 FastSpeech 2s，这是首次尝试直接并行地从文本生成语音波形。

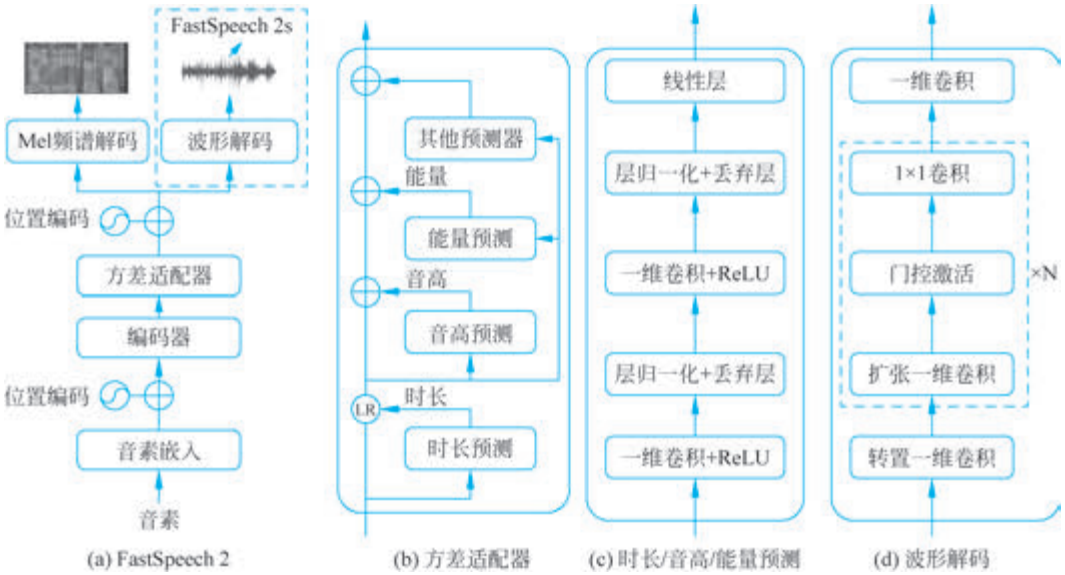

图 5.26　FastSpeech 2 框架

FastSpeech 2 的整体模型架构如图 5.26(a)所示。编码器将音素嵌入序列转换为音素隐藏序列，然后方差适配器将不同的变化信息（如持续时间、音高和能量）添加到隐藏序列中，Mel 频谱解码器将适应后的隐藏序列并行转换为 Mel 频谱序列。使用前馈 Transformer 块作为编码器和 Mel 频谱解码器的基本结构，它由自注意力层和一维卷积组成，这与 FastSpeech 中的结构相同。FastSpeech 2 与 FastSpeech 的不同之处在于，它采取了几项改进措施。首先，FastSpeech 2 移除了教师-学生蒸馏流程，并直接使用真实 Mel 频谱图作为模型训练的目标，这可以避免蒸馏后 Mel 频谱图中的信息损失，并提高声音质量的上限。其次，FastSpeech 2 的方差适配器不仅包括持续时间预测器，还有音高和能量预测器。其中，①持续时间预测器使用强制对齐法得到的音素持续时间作为训练目标，这比从

自回归教师模型的注意力图中提取的持续时间更准确；②附加的音高和能量预测器可以提供更多的变化信息，对于缓解 TTS 中的一对多映射问题非常重要。最后，为了进一步简化训练流程并使其成为完全端到端的系统，FastSpeech 2s 直接从文本生成波形，而不需要级联的 Mel 频谱生成（声学模型）和波形生成（声码器）。在下面的小节中，我们将详细描述方法中方差适配器和直接声波生成的设计。

方差适配器的目标是将方差信息（如持续时间、音高、能量等）添加到音素隐藏序列中，从而提供足够的信息来预测文本到语音合成中的一对多映射问题中的变体语音。方差信息包括：①音素持续时间，代表说话声音的持续时间；②音高，是传达情感的关键特征，极大地影响语音韵律；③能量，表示 Mel 频谱图的帧级幅度，直接影响语音的音量和韵律。相应地，方差适配器包括：①持续时间预测器（即长度调节器，如 FastSpeech 中所用）；②音高预测器；③能量预测器，如图 5.26(b)所示。在训练中，将从录音中提取的持续时间、音高和能量的真值作为输入隐藏序列中，以预测目标语音。同时，使用真实的持续时间、音高和能量作为目标来训练持续时间、音高和能量预测器，这些预测器在推理中用于合成目标语音。如图 5.26(c)所示，持续时间、音高和能量预测器具有类似的模型结构（但具有不同的模型参数），包括一个包含 ReLU 激活函数的两层一维卷积网络，每层后面都有层归一化和丢弃层，以及一个额外的线性层将隐藏状态投影到输出序列上。

持续时间预测器将音素隐藏序列作为输入，并预测每个音素的持续时间，代表了这个音素对应多少个 Mel 频谱帧，并将其转换为对数域以便于预测。持续时间预测器通过均方误差（MSE）损失进行优化，并将提取的持续时间作为训练目标。为了提高对齐准确性，减少模型输入和输出之间的信息差异，使用 Montreal 强制对齐（MFA）工具来提取音素持续时间，而不是使用预先训练的自回归 TTS 模型在 FastSpeech 中提取音素持续时间。

先前基于神经网络的 TTS 系统中的音高预测通常直接预测音高轮廓。然而，由于实际音高的高度变化，预测音高值的分布与实际音高值的分布非常不同。为了更好地预测音高轮廓的变化，FastSpeech 2 使用连续小波变换（Continuous Wavelet Transform，CWT）将连续的音高序列分解为音高谱图，并将音高谱图作为音高预测器的训练目标，该预测器通过 MSE 损失进行优化。在推理中，音高预测器预测音高谱图，然后使用反向连续小波变换（inverse Continous Wavelet Transform，iCWT）将其转换回音高轮廓。为了在训练和推理中将音高轮廓作为输入，将每帧的音高 F0（训练/推理的真值/预测值）在对数尺度上量化为 256 个可能值，并将其转换为音高嵌入矢量 $\boldsymbol{P}$，并将其添加到扩展的隐藏序列中。

FastSpeech 2 将每个短时傅里叶变换（STFT）帧的振幅的 L2 范数作为能量。然后，将每个帧的能量均匀量化为 256 个可能值，并将其编码为能量嵌入 e，并与音高类似地添加到扩展的隐藏序列中。FastSpeech 2 使用能量预测器来预测能量的原始值，而不是量化后的值，并使用 MSE 损失进行能量预测器的优化。

4. Grad-TTS

扩散概率模型的扩散型过程可以看作满足随机微分方程（Stochastic Differential Equation，SDE）的随机过程。

$$\mathrm{d}X_t = b(X_t, t)\mathrm{d}t + a(X_t, t)\mathrm{d}W_t \tag{5.30}$$

其中，W_t 是标准布朗运动，$t \in [0, T]$，T 为有限时间范围，并且系数 b 和 a（分别称为漂移

和扩散)满足一定的可测条件。

容易找到这样的随机过程,当 $T\to\infty$ 时,对于任意的初始数据分布 Law(X_0),最终分布 Law(X_T)收敛到标准正态分布 $N(0,1)$(1 是 $n\times n$ 的单位矩阵,n 是数据的维度)。任何具有这种性质的扩散型过程都称为前向扩散,扩散概率建模的目标是找到一个逆向扩散,使其轨迹与前向扩散的轨迹紧密相似,但时间顺序相反。用适当的神经网络对逆向扩散进行参数化,可以实现这一过程。在这种情况下,生成就变成了从 $N(0,1)$ 中采样随机噪声,然后使用任何数值求解器。通常使用简单的一阶 Euler-Maruyama 方案求解描述逆向扩散动力学的 SDE。如果前向和逆向扩散过程的轨迹相近,则生成样本的分布将非常接近数据 Law(X_0)的分布。

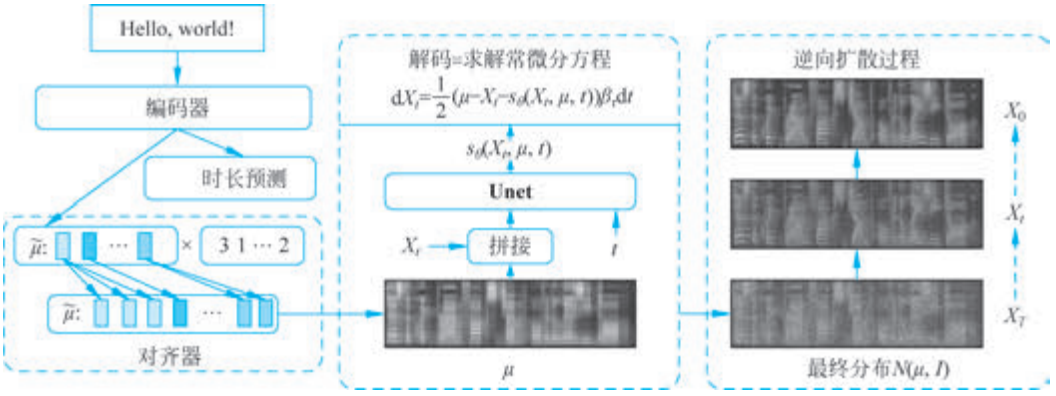

图 5.27　Grad-TTS 框架

Grad-TTS 的声学特征生成器由三个模块组成:编码器、持续时间预测器和解码器,如图 5.27 所示。

(1) 推理过程:一个长度为 L 的输入文本序列 $x_{1:L}$ 通常由字符或音素组成,Grad-TTS 推理过程的目标是生成 Mel 频谱图 $y_{1:F}$,其中 F 是声学帧的数量。在 Grad-TTS 中,编码器将输入文本序列 $x_{1:L}$ 转换为用于持续时间预测器生成编码文本序列 $\tilde{\mu}_{1:L}$ 与逐帧特征 $\mu_{1:F}$ 之间的硬单调对齐 A 的特征序列。函数 A 是一个单调满射映射,映射区间为$[1,F]\cap\mathbb{N}$ 和 $[1,L]\cap\mathbb{N}$,定义 $\mu_j=\tilde{\mu}_{A(j)}$,其中 $j\in[1,F]$为任意整数。简而言之,持续时间预测器告诉我们每个文本输入元素持续的帧数。A 的单调性和满射性保证了文本按照正确的顺序发音,不会跳过任何文本输入。与所有带有持续时间预测器的 TTS 模型一样,通过将预测的持续时间乘以某个因子,可以控制合成语音的速度。

然后,输出序列 $\mu=\mu_{1:F}$ 被传递给解码器,它是一个扩散概率模型。具有参数 θ 的神经网络 $s_0(X_t,\mu,t)$定义了一个常微分方程(ODE)。

$$\mathrm{d}X_t=\frac{1}{2}(\mu-X_t-s_\theta(X_t,\mu,t))\beta_t\,\mathrm{d}t \tag{5.31}$$

使用一阶 Euler 方案逆向求解。序列 μ 也用于定义终止条件 $X_T\sim N(\mu,I)$。噪声计划 β_t 和时间跨度 t 是预定义的超参数,其选择主要取决于数据,而在 Euler 方案中的步长 h 是一个可以在 Grad-TTS 训练后选择的超参数。它表示输出 Mel 频谱图质量和推理速度之间的权衡。

(2) 训练过程：Grad-TTS 训练的一个目标是最小化对齐的编码器输出 μ 和目标 Mel 频谱图 y 之间的距离，因为刚才描述的推理方案建议从随机噪声 $N(\mu, I)$开始解码。直观上，如果我们从与目标 y 在某种意义上已经接近的噪声开始解码，解码会更容易。

如果考虑到对齐的编码器输出 μ 是编码器启动所需的输入噪声的参数化表示，那么自然地将编码器输出 $\widetilde{\mu}$ 视为正态分布 $N(\widetilde{\mu}, I)$，从而导致负对数似然编码器损失：

$$\mathcal{L}_{\text{enc}} = -\sum_{j=1}^{F} \log \varphi(y_j; \widetilde{\mu}_{A(j)}, I) \tag{5.32}$$

其中，$\varphi(\cdot; \widetilde{\mu}_i, I)$是 $N(\widetilde{\mu}_i, I)$的概率密度函数。尽管也可能使用其他类型的损失函数，但由于这种概率解释，Grad-TTS 选择了$\mathcal{L}_{\text{enc}}$(实际上等同于均方误差准则)。原则上，甚至可以在 Grad-TTS 中完全没有任何编码器损失，并且让进一步描述的扩散损失完成生成逼真的 Mel 频谱图的所有工作，但在实践中，没有$\mathcal{L}_{\text{enc}}$的情况下，Grad-TTS 无法学习对齐。

编码器损失$\mathcal{L}_{\text{enc}}$必须相对于编码器参数和对齐函数 A 进行优化。由于很难进行联合优化，$\mathcal{L}_{\text{enc}}$采用了 Glow-TTS 提出的一种迭代方法。优化的每个迭代由两个步骤组成：在固定编码器参数的情况下搜索最佳对齐 A^*；固定该对齐 A^* 并进行一步随机梯度下降，以优化损失函数相对于编码器参数的情况。在该方法的第一步中使用了单调对齐搜索。MAS 利用动态规划的概念来找到最佳的单调满射对齐。

为了在推理中估计最佳对齐 A^*，Grad-TTS 使用了时长预测网络。与 Glow-TTS 中一样，Grad-TTS 使用对数域中的均方误差准则(MSE)来训练时长预测器 DP。

$$d_i = \log \sum_{j=1}^{F} \mathbb{I}_{\{A^*(j)=i\}}, \quad i = 1, 2, \cdots, L \tag{5.33}$$

$$\mathcal{L}_{\text{dp}} = \text{MSE}(\text{DP}(sg[\widetilde{\mu}]), d) \tag{5.34}$$

其中，$\mathbb{I}$ 是一个指示函数，$\widetilde{\mu} = \widetilde{\mu}_{1:L}$，$d = d_{1:L}$，而停止梯度操作符 $sg[\cdot]$被应用于时长预测器的输入，以防止$\mathcal{L}_{dp}$对编码器参数产生影响。

DPM 相关的损失：设 $\Sigma = I$，因此噪声数据的分布简化，其协方差矩阵仅变为标量乘以单位矩阵 $\boldsymbol{I}$。

$$\lambda_t = 1 - \mathrm{e}^{-\int_0^t \beta_s ds} \tag{5.35}$$

整体的扩散损失函数$\mathcal{L}_{\text{diff}}$是在不同时间点 $t \in [0, T]$上估计噪声数据的对数密度梯度时，加权损失的期望。

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{X_0, t}\left[\lambda_t \mathbb{E}_{\xi_t}\left[\left\| s_\theta(X_t, \mu, t) + \frac{\xi_t}{\sqrt{\lambda_t}} \right\|_2^2\right]\right] \tag{5.36}$$

其中，X_0 代表训练数据中采样目标 Mel 频谱图 y，t 从$[0, T]$的均匀分布中采样，ξ_t 服从 $N(0, I)$，而公式为

$$X_t = \rho(X_0, I, \mu, t) + \sqrt{\lambda_t}\xi_t \tag{5.37}$$

使用$\boldsymbol{\epsilon}_t = \sqrt{\lambda_t}\xi_t$ 进行替换，可以得到上述公式。使用带有权重 λ_t 的损失函数，根据常见的经验法则，这些权重应与 $1/\mathbb{E}[\| \nabla \log p_{0t}(X_t | X_0) \|_2^2]$成比例。

总之，训练过程包括以下步骤：

固定编码器、持续时间预测器和解码器参数，并运行 MAS 算法，以找到最小化$\mathcal{L}_{\text{enc}}$的对

齐 A^*；

固定对齐 A^* 并最小化相对于编码器、持续时间预测器和解码器参数的损失函数 $\mathcal{L}_{enc}+\mathcal{L}_{dp}+\mathcal{L}_{diff}$；

重复这两个步骤直到收敛。

5.5.2　声码器端到端语音合成

声码器端到端语音合成是指声学模型和声码器由一个模型建模，需要一个额外的文本分析模块对文本进行分析处理，再送入后端模型生成语音波形，如图 5.20 所示。

1. MelGAN

作为生成对抗网络在声码器任务的应用之一，我们首先简略介绍生成对抗网络(Generative Adversarial Network，GAN)，这是一种用于生成样本的机器学习模型。它由两个主要组件组成：生成器(Generator)和判别器(Discriminator)。生成器和判别器是通过对抗训练来相互竞争和提升的。生成器的目标是生成与真实样本相似的合成样本，而判别器的目标是区分真实样本和生成样本。两个组件通过反复博弈的过程进行训练，以提高各自的性能。其训练过程可以简述如下：

(1) 生成器接收一个随机噪声矢量作为输入(有时还有条件信息矢量一同作为输入)，并生成一个合成样本，如图 5.28 所示；

(2) 判别器接收真实样本和生成样本，并尝试将它们区分开来；

(3) 判别器的输出被用作衡量生成器生成样本的质量的指标；

(4) 生成器根据判别器的反馈调整自己的生成策略，以生成更逼真的样本。

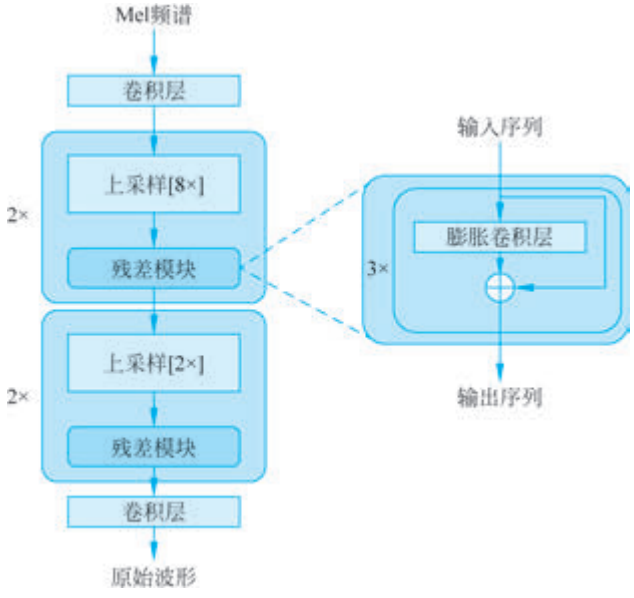

图 5.28　MelGAN 生成器框架

通过反复迭代训练生成器和判别器，使生成器能够生成更接近真实样本的合成样本，

而判别器能够更准确地区分真实样本和生成样本。关于 GAN 模型的更多细节，读者可以自行了解，事实上，与 GAN 模型之后的生成模型（如 VQVAE、Diffusion）相比，GAN 模型的框架，以及训练和推理过程相对简单。因此，本节中更为关注生成器和鉴别器的网络结构，我们将重点介绍这些内容。

传统的 GAN 模型中，生成器通常接收一个随机噪声矢量作为输入，这个噪声矢量是从一个全局的噪声分布中采样得到的。生成器通过将这个噪声矢量转换成与真实数据相似的样本，当然，如果含有条件信息，也要一起作为输入。

对于声码器这一任务，要求在时域维度上对谱图的每帧进行几百倍上采样恢复到波形长度。对于 GAN 而言，如果条件信息非常强，噪声输入就不再重要。已知声码器的上采样过程是一对多的映射，那么 Mel 频谱图是否也属于强条件信息呢？提出者得出的实验结论“依旧属于”似乎是反常的，即取消全局噪声矢量作为输入，反而能取得更好的效果。

上采样的过程使用转置卷积的堆栈。每个转置卷积后面都是一堆具有膨胀卷积的残差块。膨胀卷积层堆叠的感受野随着层数的增加呈指数增长，这样可以扩大感受野的范围。类比图像生成领域中各个像素感受野中的重叠。更大的感受野有利于提取音频各部间更好的长程相关性。此外，转置卷积的相关研究中，“棋盘效应”是一个需要避免的常见问题，提出者通过实验刻意设计生成器的参数，确保转置卷积层中，卷积核的大小是 Stride 的倍数，同时卷积核的大小逐层幂次增长。

用于图像生成的流行 GAN 架构中，多使用实例规范化（Instance Normalization）的方法，提出者进行了相关实验，尝试了实例规范化、包括谱归一化（Spectral Normalization）的方法。最终确定在生成器的所有层中使用权重归一化（Weight Normalization）的方法。

MelGAN 中提出的这一鉴别器（如图 5.29 所示），称为多尺度鉴别器（Multi-Scale Discriminator，MSD）在后来的 GAN 系列声码器中依然用途广泛。它的特征是存在 D1、D2、D3 共 3 个结构一样的鉴别器分别对原音频，对原始音频进行 2 和 4 倍的下采样之后的音频进行操作。这一下采样是通过核大小为 4 的步幅平均池化层（Strided Average Pooling）完成的。因为音频有不同的层次，对不同尺度的音频，鉴别器可以专门针对不同频率范围的音频特性。例如，在下采样音频上运行的鉴别器无法获取高频成分，因此只学习基于低频成分的鉴别特征是有偏差的。

对于单个的鉴别器本身，除了 Output，每层中还输出了特征图（Feature Map），值得说明的是，类比图像任务当中基于 Image Patch 的概念，MSD 鉴别器是基于窗口的。标准 GAN 鉴别器学习在整个音频序列的分布之间进行分类，而基于窗口的鉴别器学习在小音频块的分布之间进行分类，这已被证明可以捕获基本的高频结构，并且可以应用于可变长度的音频序列。同样，鉴别器中也采用权重归一化（Weight Normalization）的方法。

MelGan 的损失函数为铰链损失（Hinge Loss）：

$$\min_{D_k}\mathbf{E}_x[\min(0,1-D_k(x))]+\mathbf{E}_{s,z}[\min(0,1+D_k(G(s,\boldsymbol{z})))],\quad \forall k=1,2,3 \tag{5.38}$$

$$\min_{G}\mathbf{E}_{s,z}\left[\sum_{k=1,2,3}-D_k(G(s,\boldsymbol{z}))\right] \tag{5.39}$$

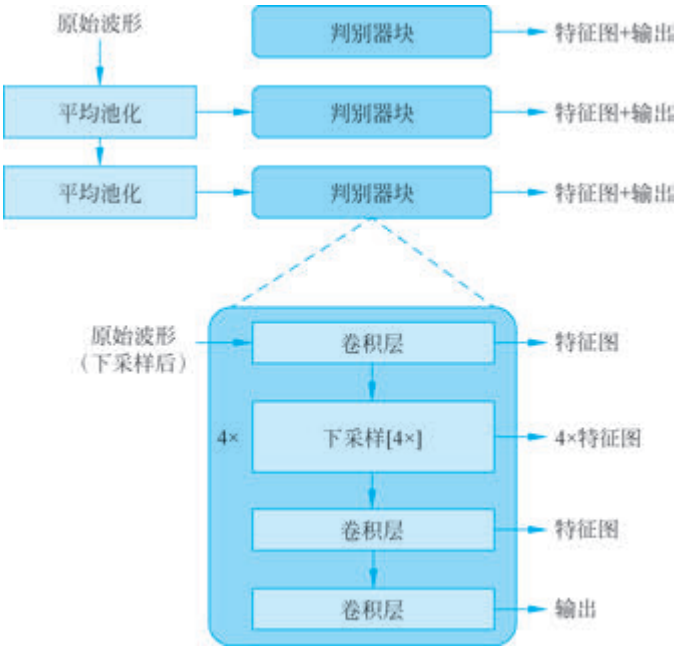

图 5.29　MelGAN 多尺度鉴别器框架

其中，x 代表原始波形，s 代表条件信息（例如 Mel 频谱图），z 代表高斯噪声矢量。事实上，由上文叙述已知，z 作为实际输入是不存在的。

特征匹配损失函数（Feature Matching Loss）：每一个鉴别器的每一层都会输出特征图（Feature Map），对原始样本和生成器生成的样本分别鉴别，将对应特征图之间的 $L1$ 距离的平均值最小化。

$$L_{\mathrm{FM}}(G,D_k)=\mathbf{E}_{x,s}\left[\sum_{i=1}^{T}\frac{1}{N_i}\left\|D_k^{(i)}(x)-D_k^{(i)}(G(s))\right\|\right] \tag{5.40}$$

$D_k^{(i)}$ 代表第 k 个鉴别器的第 i 个特征图输出，一共有 T 层，N_i 代表每层所有单元的数目，在所有鉴别器块的每个中间层上均使用这一特征匹配。

因此，最终生成器优化目标为

$$\min_G\left(\mathbf{E}_{s,z}\left[\sum_{k=1,2,3}-D_k(G(s,z))\right]+\lambda\sum_{k=1}^{3}L_{\mathrm{FM}}(G,D_k)\right) \tag{5.41}$$

2. HifiGAN

在 MelGAN 的基础上，理解 HifiGAN 是相对容易的，后者在各种意义上都是前者的继承，因此，本节重点叙述这一工作最大的创新之处：多周期鉴别器（Multi-Period Discriminator，MPD）。HifiGAN 多周期鉴别器框架如图 5.30 所示。

图 5.29 是周期为 3 的子鉴别器，事实上，为了避免重叠，还设置了 2、3、5、7、11 等一系列周期。MPD 是子鉴别器的混合，每个子鉴别器只接收输入音频的等间隔样本，以周期 p 给出。子鉴别器通过查看输入音频的不同部分来捕获其中的隐式结构。

首先将长度为 T 的 1D 原始音频重塑为高度为 T/p 和宽度为 p 的 2D 数据，然后对重

塑后的数据应用 2D 卷积，但是宽度轴上的核大小限制为 1，以独立处理周期性样本。鉴别器的每一层都使用了权重归一化。

以上即多周期鉴别器的结构。HifiGAN 使用了两种鉴别器，另一个是 MelGAN 中提出的多尺度鉴别器(Multi-Scale Discriminator，MSD)，HifiGAN 多尺度鉴别器框架如图 5.31 所示。

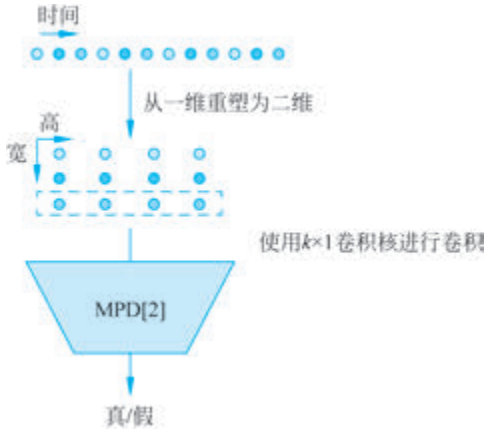

图 5.30　HifiGAN 多周期鉴别器框架

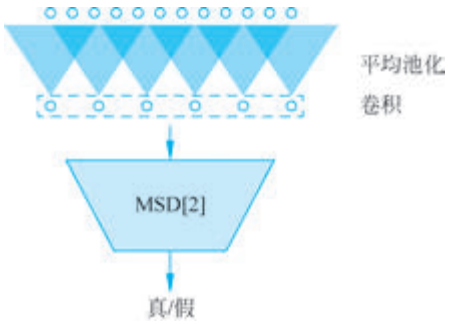

图 5.31　HifiGAN 多尺度鉴别器框架

图 5.31 是多尺度鉴别器中，对原始波形 4 倍下采样时的子鉴别器。可以注意到，多周期鉴别器对原始波形的不相交样本进行操作，而多尺度鉴别器对平滑波形进行操作。

除去以上鉴别器外，相较 MelGAN，生成器损失函数也还多了一项，即 Mel 频谱图距离损失(Mel-Spectrogram Loss)，具体而言，即最小化生成器合成波形的 Mel 频谱图与原真实波形的 Mel 频谱图之间的 $L1$ 距离。

3. BigVGAN

BigVGAN 的生成器延续 GAN 系列的架构，而鉴别器包括 3 种，除去 MelGAN 提出的多尺度鉴别器(MSD)，HifiGAN 提出的多周期鉴别器(MPD)以外，还有一种援引自 Won Jang 等提出的多分辨率鉴别器(Multi-Resolution Discriminator，MRD)，具体来说，该鉴别器也是由多个子鉴别器组成，每个子鉴别器将生成音频波形和原音频波形，通过 STFT 变换转换到频域，计算频谱图的差异实现鉴别功能。不同的子鉴别器采用不同窗长的 STFT 变换，这样基于不同分辨率下的二维频谱图上完成鉴定，并求均值。这几种鉴别器的组合可以构成不同的 GAN 结构方案。而 BigVGAN 的提出者认为用 MRD 代替 MSD 可以改善音质，减少音高和周期性伪影。BigVGAN 将 Snake 激活函数应用至声码器任务中。

考虑音频波形的特点，高周期性是其中之一，可以自然地表示为原始周期分量的组合(例如狄利克雷条件下的傅里叶级数)。在 HifiGAN 等的工作中，依靠膨胀卷积层来学习不同频率下所需的周期分量，其非线性激活函数(例如 Leaky ReLU)可以产生一些新细节，但不能提供任何周期性相关的信息。提出者发现使用 Leaky ReLU 激活函数尽管在训练集分布之内，例如可见过的记录环境中，能产生高质量的语音信号。但对于分布以外的场景，如未见过的记录环境、非语音发声等，产生语音的质量会显著下降。

为了利用语音高周期性这一特点，引入一种周期性 Snake 激活函数替换 Leaky ReLU。

在此之前，这种周期性激活函数已经在温度和金融数据预测任务中被证明有效，其公式为

$$f_\alpha(x)=x+\frac{1}{\alpha}\sin^2(\alpha x) \tag{5.42}$$

其中，α 是控制信号周期成分的频率可训练的参数，α 越大，频率越高。$\sin^2(x)$的使用确保了单调性，并使其易于优化。

Snake 激活函数适应了建模原始波形所需的周期性，它可以为连续的时间信号产生任意的高频细节，但这些细节无法由网络的离散时间输出表示，这可能会导致混叠伪影。这种副作用可以通过应用低通滤波器来抑制。操作方法是沿时间维度对信号进行 2 倍上采样，应用 Snake 激活函数激活，然后对信号进行 2 倍下采样，这是受奈奎斯特-香农采样定理启发的常见做法。每个上采样和下采样操作都伴随着使用低通滤波器，采用 Kaiser 窗口函数和 Sinc 滤波器。

4. DiffWave

作为去噪概率扩散模型(Denoising Diffusion Probabilistic Models, DDPM)在声码器任务的应用之一，考虑到本书的其他章节，例如 FastDiff-TTS，已经做过解释，因此这里不对去噪概率扩散模型进行详细的阐述。如果读者对该模型比较生疏，应该先进行一定的了解。事实上，DiffWave 中使用的概率扩散模型的前向过程与后向过程采用了与 DDPM 几乎完全相似的结构。

对于 DDPM 结构模型而言：其训练过程即前向加噪过程，训练对象即“噪声预测器网络”。推理过程即反向解噪过程。因此，这里侧重于介绍 DiffWave 的噪声预测器结构，以及该方法可适用的若干任务。DiffWave 作为一个非自回归的波形生成模型，声码器，即 Mel 频谱图到波形转换只是该模型的多种应用方法之一。

本方法中，扩散模型的训练是依据以下公式：

$$\min_\theta L_{\text{unweighted}}(\theta)=\mathbf{E}_{x_0,\boldsymbol{\epsilon},t}\left\|\boldsymbol{\epsilon}-\varepsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon},t\right)\right\|_0^2 \tag{5.43}$$

这也是“噪声预测器”的目标函数。$\bar{\alpha}_t$ 等为噪声策略参数。

预测器的输入包含两个部分，扩散到第 t 步时的状态 $x_t=\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$和步数 t。事实上，当面临条件生成模式时，还会有 Conditioner 作为输入。

DiffWave 的网络结构沿袭自 WaveNet，3 种输入如图 5.32 所示。对于 Input，即扩散过程中第 t 步时的带噪语音 x_i，初始通道为 1，长度 L 即该段语音的波形点数目。对于 Output，即扩散过程中第 t 步时的预测噪声，其形状与 Input 一样。

对于双向膨胀卷积层，由图 5.32 可知，网络整体由 N 道残差模块构成，在第 i 道残差模块中，双向膨胀卷积层卷积核大小定为 3，而膨胀系数设定为 $i \bmod n$。这一方法继承自 WaveNet，将膨胀系数取为大小为 n 的循环$[1,2,4\cdots\cdots 2^{n-1}]$，通常而言，$n$ 小于 N，例如提出者在无条件生成实验中，采用的一组参数即 36 层残差模块，循环周期为$[1,2,4\cdots\cdots 2048]$。此外，由于 DiffWave 并非 WaveNet 那样的自回归模型，故采用双向机制，这使得计算新的波形点时，感受野(即通过数层膨胀卷积层后对该波形点的生成有贡献影响的一组波形点)既包括此时之前的信息，也包括此时之后的信息，双向膨胀卷积的感受野如图 5.33 所示。

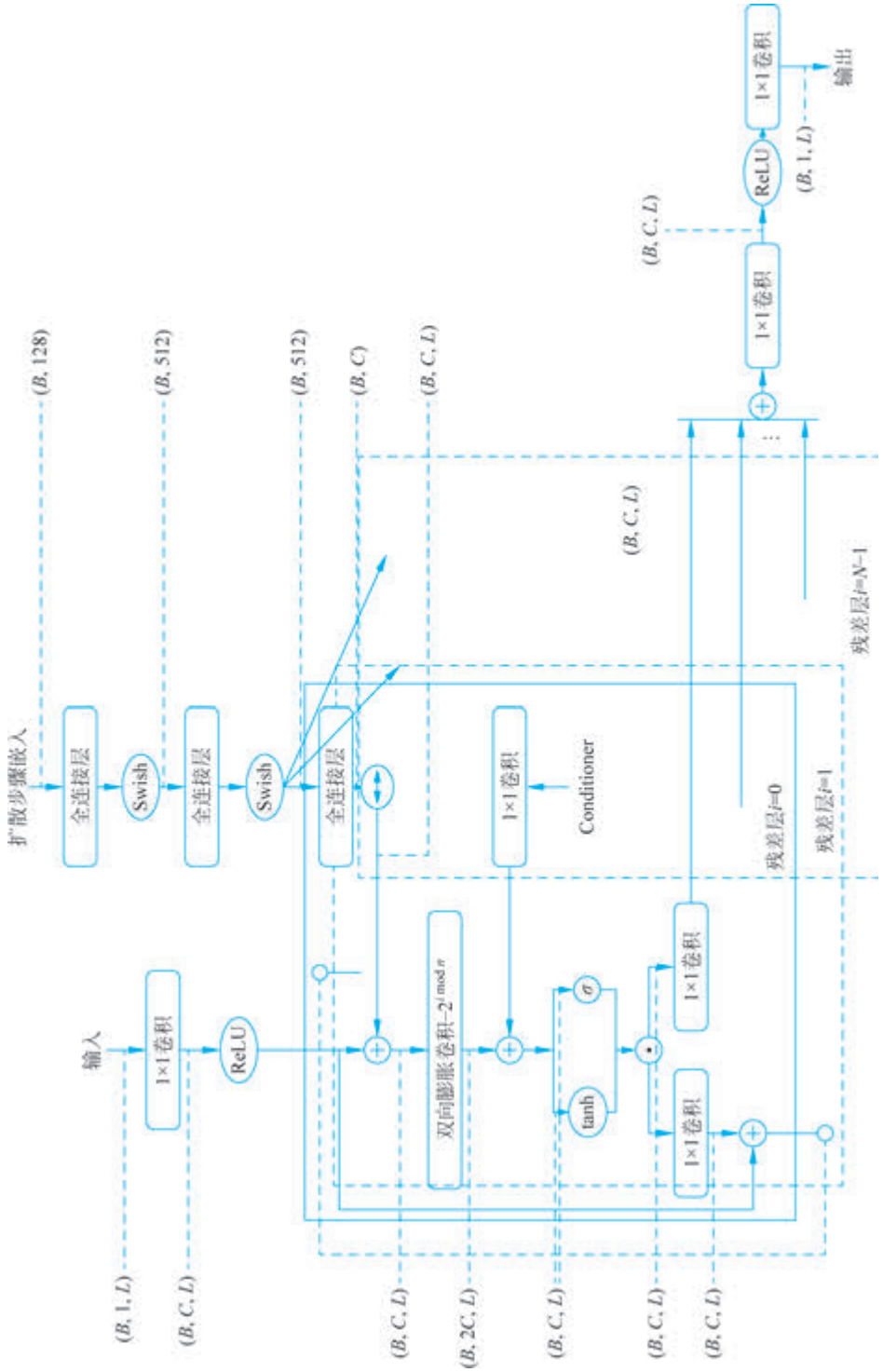

图 5.32　DiffWave 的网络结构

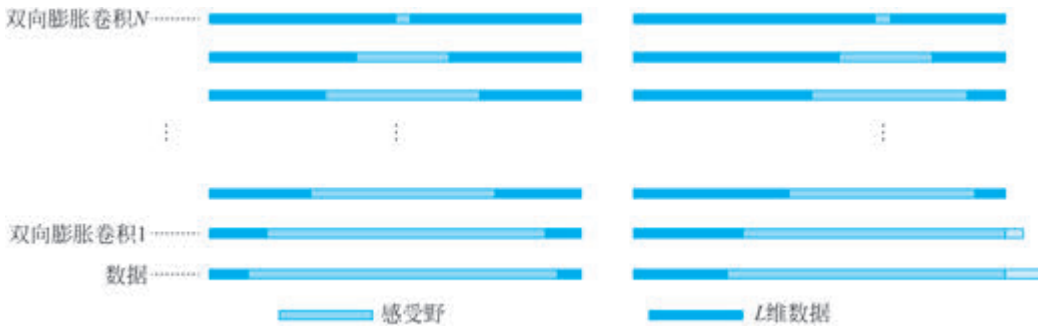

图 5.33　双向膨胀卷积的感受野

双向卷积层后是类似 WaveNet 的门激励单元，在该神经网络中起激励作用，其中 σ 指 Sigmoid 函数。

对于 Diffusion-step，即扩散步数 t，采用 128 维的嵌入矢量作为输入，嵌入矢量的编码方式如以下公式所示

$$t_{\text{embeding}}=\left[\sin\left(10^{\frac{0\times4}{63}}t\right),\cdots,\sin\left(10^{\frac{63\times4}{63}}t\right),\cos\left(10^{\frac{0\times4}{63}}t\right),\cdots,\cos\left(10^{\frac{63\times4}{63}}t\right)\right] \tag{5.44}$$

随后，前两个全连接层对于所有的残差层共享参数，第 3 层全连接层将通道维度映射到 C 维。随后，在长度维度上广播到 L，然后加入每个残差层中。这 3 个全连接层将步数 t 转换为了与音频张量（当然此时通道已扩充为 C）相同的形状。

另外，我们应该意识到，扩散步骤 t 除了以这样的方式融入网络外，根据扩散模型原理，代表噪声策略的一系列参数阿尔法，贝塔等也与 t 有关，这也将影响训练与推理过程。

以上是 DiffWave 网络的说明，接下来将介绍 DiffWave 的相关应用任务。

（1）无条件生成：这种模式下，DiffWave 训练与推理中都不必接收 Conditioner 就可以生成语音。根据扩散模型的角度看来，从完全的高斯噪声中逐步去噪生成随机的音频。符合生成模型的定义，这样得到的音频符合训练集的数据分布，且不归属于训练集本身。以提出者进行的实验为例，抽取 Speech Commands 数据集中的数字录音子集（SC09 数据集）进行训练，随机生成训练集中从未出现的数字录音。值得说明的是，提出者指出，双向膨胀卷积的尺寸和层数的增加将会增加训练的不稳定性，导致结果变差。

（2）全局条件生成：全局条件生成是指条件信息由全局离散标签给出（例如说话者 ID 或单词 ID，前者在一些指定说话人风格的场景中很常见），以提出者进行的类条件生成实验为例，依旧以 SC09 数字录音数据集进行训练，以数字的标签作为 Conditioner。以上标签设置为 128 维的嵌入矢量，在每个残差层中，都通过 1×1 卷积映射到 $2C$ 的通道，作为偏置项加在膨胀卷积之后的输出上，因此要与这时的音频张量形状相同。

（3）局部条件生成：作为本章的主线内容，就语音合成而言，神经声码器可以通过 Mel 频谱图或者文本到波形架构中的某种隐藏状态合成波形。这样的条件不具有全局性，显然某帧的 Mel 频谱只负责对应时域帧的波形。Mel 频谱图首先经过转置二维卷积进行上采样，达到与音频张量相同的长度 L，随后亦是通过 1×1 卷积映射到 $2C$ 的通道，作为偏置项加在膨胀卷积之后的输出上。

（4）加速采样：DiffWave 的相关论文中介绍了反向解噪时采用的一种加速采样方法。

扩散概率模型的推理过程通常要反向解噪完训练时设定的步数，这将消耗大量推理时间，事实上，扩散模型推理的“慢”是影响扩散模型实际应用的重要因素之一。

为了缓解这一问题，注意到采样时最有效地去噪步骤发生在 $t=0$ 附近，因此设计更少去噪步数（例如相对训练时 200 步的 T 而言）的快速采样算法 T_{infer}（例如 6 步），关键思想是将 T 步逆向过程“折叠”为一个经过精心设计的方差调度的 T_{infer} 过程。

具体来说，用户可以完全自行定义一个噪声策略 $\{\eta_t\}_{t=1}^{T_{\text{infer}}}$，作为训练过程中噪声策略 $\{\beta_t\}_{t=1}^{T_{\text{infer}}}$ 的替代，显然，还可以定义 γ，用于对标 α：

$$\gamma_t=1-\eta_t,\bar{\gamma}_t=\sum_{s=1}^{t}\gamma_s,\bar{\eta}_t=\frac{1-\bar{\gamma}_{t-1}}{1-\bar{\gamma}_t}\eta_t\quad(t>1,\bar{\eta}_1=\eta_1) \tag{5.45}$$

接下来，需要确定，这些用户自行定义的某步噪声策略，究竟等同于原噪声策略的哪一步。总步数由 T 变成 T_{infer}，两个数列一定存在着某种放缩关系。规定 T_{infer} 数列中的第 s 步，对应 T 数列中的第 t_s^{align} 步（计算后这通常是一个浮点数），通过插值公式可作如下映射，公式中的 t 根据 if 条件确定：

$$t_s^{\text{align}}=t+\frac{\sqrt{\bar{\alpha}_t}-\sqrt{\bar{\gamma}_s}}{\sqrt{\bar{\alpha}_t}-\sqrt{\bar{\alpha}_{t+1}}} \tag{5.46}$$

其中，$\sqrt{\bar{\gamma}_s}\in[\sqrt{\bar{\alpha}_{t+1}},\sqrt{\bar{\alpha}_t}]$ 推理过程，即反向解噪时，循环递进 T_{infer} 数列，当第 s 步时，从完全高斯噪声中逐渐解得当前带噪音频 x_s，x_s 是从参数为 $\mu_\theta^{\text{fast}}(x_s,s)$，$\sigma_\theta^{\text{fast}}(x_s,s)$ 的高斯噪声中采样而来的，两个参数计算如下：

$$\mu_\theta^{\text{fast}}(x_s,s)=\frac{1}{\sqrt{\gamma_s}}\left(x_s-\frac{\eta_s}{\sqrt{1-\bar{\gamma}_s}}\boldsymbol{\epsilon}_\theta(x_s,t_s^{\text{align}})\right),\quad\sigma_\theta^{\text{fast}}(x_s,s)=\tilde{\eta}_s^{\frac{1}{2}} \tag{5.47}$$

直到得到 x_0。

5. WaveGrad

WaveGrad 也是去噪扩散概率模型的声码器，去噪扩散概率模型（DDPM）的基本理论同样不再赘述。WaveGrad 的训练即为前向加噪过程，推理过程即为反向去噪过程。这里重点关注作为“噪声预测器”的网络结构部分。与 DiffWave 不同，WaveGrad 的网络结构使它作为语音波形生成器只承担神经声码器这一种局部条件生成任务。网络结构如图 5.34 所示。

列出训练时的目标函数公式为

$$\mathbf{E}_{\bar{\alpha},\boldsymbol{\epsilon}}[\|\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}}y_0+\sqrt{1-\bar{\alpha}}\boldsymbol{\epsilon},x,\sqrt{\bar{\alpha}})-\boldsymbol{\epsilon}\|_1] \tag{5.48}$$

值得说明的是，WaveGrad 模型的介绍中，一些同等意义的参数的符号与 DiffWave 中是不一样的，尽管两个模型采用的扩散原理相同，请勿产生符号表示上的混淆。如果读者依然感到困惑，可以先行详细了解概率扩散模型的基本原因。

对照目标函数公式中噪声预测器 $\boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}}y_0+\sqrt{1-\bar{\alpha}}\boldsymbol{\epsilon},x,\sqrt{\bar{\alpha}})$ 与网络结构图，$y_n=\sqrt{\bar{\alpha}_t}y_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$ 作为输入之一，即当前加噪或解噪第 t 步时的音频（含噪）状态。x 即从文本到波形中的中间声学特征，例如 Mel 频谱图。$\sqrt{\bar{\alpha}}$ 作为噪声策略，是与 n 相关的参数，同样也施加在网络结构之中，作用于 FiLM 模块。

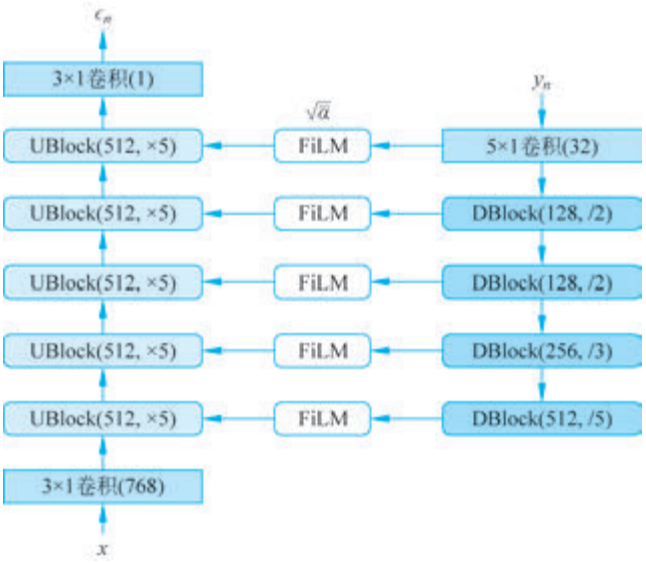

图 5.34　WaveGrad 的网络结构

训练过程中，我们需要成对的 Mel 频谱图即 x 和音频波形即 y_0，训练以上预测器对采样噪声的预测准确度。以第 n 步加噪训练为例，接受 x 和 y_0，以及当前的 $\sqrt{\bar{\alpha}}$，预测得到噪声 ϵ_n，并与实际添加的随机采样噪声 ϵ 计算损失函数，反向传播更新预测器的网络参数。推理过程中，通过该训练好的网络，从完全高斯噪声中随机采样的 y_n 开始，结合待处理的 Mel 谱图 x，预测噪声并解褪回 y_0。

在以上网络结构图中，音频波形 y_n 与输出的噪声 ϵ_n 有相同的张量形状。在这一过程中，声学特征 Mel 频谱图 x 也需要变更到相同的形状才行，因此存在时间维度的上采样过程，通道数目即 Mel 频谱的频域特征数，也要降低为 1 个通道。这一过程由 5 层上采样模块逐步完成，与此同时，音频波形 y_n 也在做下采样，每一级都对应了正经过了某层上采样的 x，这样的每一对组合都有对齐的长度，通过 FilM 模块进行信息的沟通和融合。

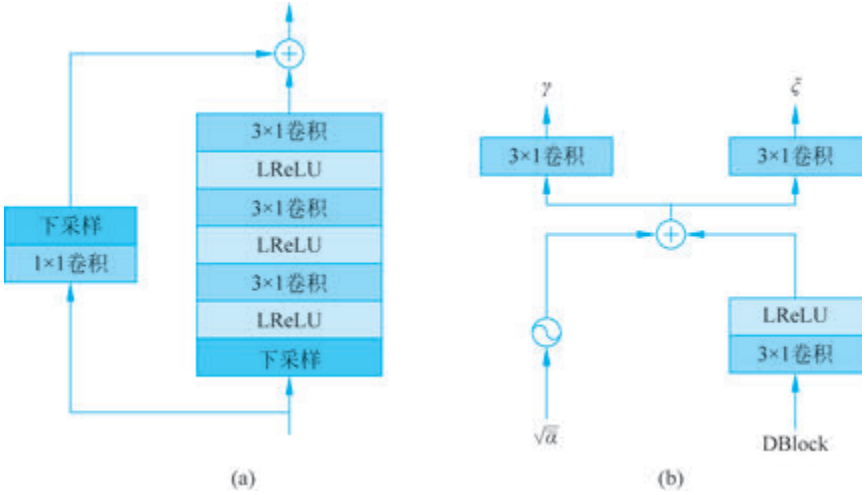

图 5.35　WaveGrad 下采样模块与 FiLM 模块的细节

如图 5.35(a)展示了下采样模块的细节。如图 5.35(b)展示了 FiLM 模块的工作机制，DBlock 即下采样之后的模块，和噪声策略$\sqrt{\alpha}$通过这一结构得到γ和ξ两个参数，这两个参数作为放缩矢量和位移矢量再融入上采样模块，具体如图 5.36 所示。

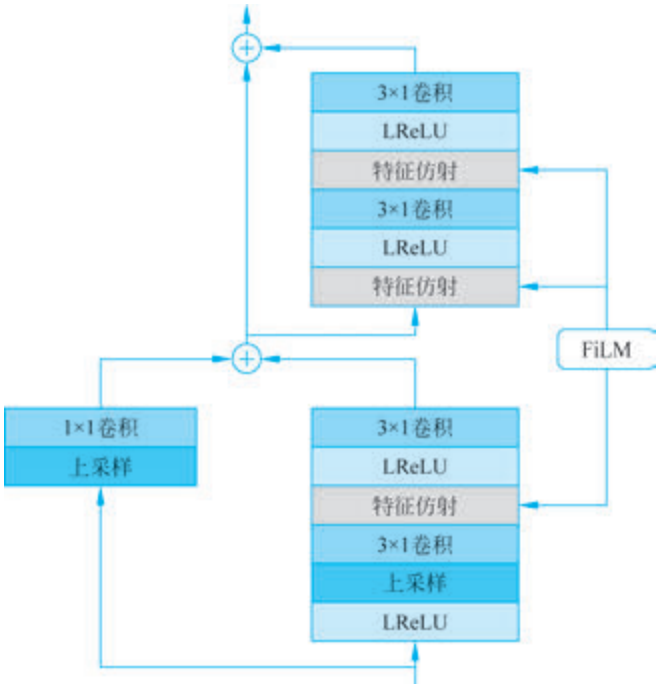

图 5.36　WaveGrad 上采样模块细节

计算公式可以表示为

$$\gamma(D,\sqrt{\alpha})\odot U+\xi(D,\sqrt{\alpha}) \tag{5.49}$$

式(5.49)中，D 是下采样模块的输出，U 是对应上采样模块的中间输出，$\odot$指矩阵对应元素相乘。

以上即 WaveGrad 的网络结构。

5.6　完全端到端语音合成

与局部端到端语音合成不同，完成端到端语音合成将文本分析、声学模型与声码器统一建模，输入文本，通过一个语音合成模型直接生成语音波形，如图 5.37 所示。代表性的完全端到端语音合成模型是基于多头注意力机制的端到端模型 EfficientTTS 和基于变分推理的端到端模型 VITS。

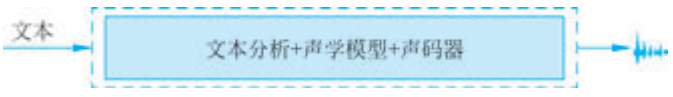

图 5.37　完全端到端语音合成

5.6.1　基于多头注意力机制的端到端模型 EfficientTTS

EfficientTTS 是一种完全端到端模型，即可以实现文本到波形的端到端训练。该方法的基础是 Encoder-decoder 的序列到序列模型。本章节会较为详细地依次描述 Efficient 的基础描述，各部分架构计算细节与功能、训练与推理的流程。

在语音合成任务中，输入文本序列 $\boldsymbol{x}=\{x_1,x_2,\cdots,x_{T_1-1}\}$ 变换为输出序列 $\boldsymbol{y}=\{y_1,y_2,\cdots,y_{T_2-1}\}$。实际上，$\boldsymbol{y}$ 可以是波形序列（端到端模式），也可以是声学特征（例如 Mel 谱图，y_j 表示某帧）。$\boldsymbol{x}\rightarrow\boldsymbol{y}$ 合成过程中，首先，输入文本序列通过编码器网络 $f:\boldsymbol{h}=f(\boldsymbol{x})$ 转换为隐藏状态序列 $\boldsymbol{h}=\{h_1,h_2,\cdots,h_{T_1-1}\}$。对于每个待生成的输出时间步，均用注意力机制搜索 $\boldsymbol{h}$ 的所有元素来生成上下文矢量 $\boldsymbol{c}$，生成第 j 个时间步时：

$$c_j=\sum_{i=0}^{T_1-1}\alpha_{i,j}*h_i \tag{5.50}$$

式(5.50)中出现了对齐矩阵（注意力分布）：$\boldsymbol{\alpha}=\{\alpha_{i,j}\}\in\mathbb{R}^{(T_1,T_2)}$，传统上，该矩阵可以直接通过几种形式的注意力机制公式计算。随后 $\boldsymbol{c}$ 将会通过解码器网络 $g:\boldsymbol{y}=g(\boldsymbol{c})$ 完成序列到序列的生成过程。简而言之，输出第 j 个时间步的元素 y_j 对应上下文矢量 $\boldsymbol{c}_j$，而 $\boldsymbol{c}_j$ 与输入序列的所有隐藏状态 $\{h_1,h_2,\cdots,h_{T_1-1}\}$ 均有关。很显然，$\alpha_{i,j}$ 代表输出第 j 个时间步 y_j 对输入序列第 i 个元素 x_i 的注意力程度。

由于编码器和解码器网络的变换过程中，$\boldsymbol{h}$ 与 $\boldsymbol{x}$，$\boldsymbol{y}$ 与 $\boldsymbol{c}$，长度都一致。但是式(5.50)。运算过程中出现了长度变换，输入长度 T_1 普遍远小于输出长度 T_2。因此对齐显得非常重要。

考虑文本到波形变换中的先验知识（下文均以生成波形而非声学特征进行讲解）。首先该过程的对齐是单调的，对于先后出现的两个输出波形点 y_j 和 $y_{j+\Delta L}$ 与它们各自在输入序列中对齐的位置 u 和 v（u、v 都是区间$[0,T_1-1]$内的浮点数），必定有 $u<v$。其次是连续性，如果 ΔL 刚好等于 1，也就是连续两个输出波形点，那么 $v-u$ 一定也不超过 1，这是由于 TTS 是短序列到长序列的转变。最后是完全性，即输入序列一定会完全被覆盖，这意味着，第一个输出波形点一定对齐位置 0，而最后一个输出波形点一定对齐 T_1-1。

为了解决对齐问题，EfficientTTS 定义了索引映射矢量（Index Mapping Vector，IMV）$\boldsymbol{\pi}=\{\pi_j\}$，$j\in[1,2,\cdots,T_2-1]$，定义为输入序列的索引矢量 $\boldsymbol{p}=\{0,1,\cdots,T_1-1\}$ 的全体加权求和：

$$\pi_j=\sum_{0}^{T_1-1}\alpha_{i,j}\cdot p_i \tag{5.51}$$

第 j 时间步输出的波形点 y_j 在输入序列中的索引映射位置数值为 π_j。作为一种整数列 $j\in[1,2,\cdots,T_2-1]$到实数区间 $\pi_j\in[0,T_1-1]$的“映射关系”。提出者在数学上证明（略），它符合单调性、连续性、完全性，可将其视为某种“对齐”，符合以下数学约束：

$$0\leqslant\Delta\pi_i\leqslant 1 \tag{5.52}$$

$$\pi_0=0 \tag{5.53}$$

$$\pi_{T_2-1}=T_1-1 \tag{5.54}$$

其中，$\Delta\pi_i=\pi_i-\pi_{i-1}$，$1\leqslant i\leqslant T_2$。

由上两段可以知道，$\boldsymbol{x}\rightarrow\boldsymbol{y}$ 合成过程中效果良好的关键在于得到好的注意力矩阵 $\boldsymbol{\alpha}$，传统上，该矩阵可以直接通过几种形式的注意力机制公式计算，但是这样得到的 α 并没用到 TTS 任务中对齐的先验知识。EfficientTTS 的核心思想，就在于利用索引映射矢量 IMV 和式(5.52)～式(5.54)提及的约束条件，将注意力机制公式直接计算出的未保证约束先验的 α，转变为保证约束先验的新 α，达成优化目的。

融入约束条件有以下两种方式：

(1) 软单调对齐(Soft Monotonic Alignment，SMA)，是依旧直接使用注意力机制公式直接计算出的 α，但整体训练中添加一项损失函数，从而尽可能满足约束条件，损失函数是

$$L_{\mathrm{SMA}}=\lambda_0\left\|\left|\Delta\pi\right|-\Delta\pi\right\|_1+\lambda_1\left\|\left|\Delta\pi-1\right|+(\Delta\pi-1)\right\|_1+\lambda_2\left\|\frac{\pi_0}{T_1-1}\right\|_2+\lambda_3\left\|\frac{\pi_{T_2-1}}{T_1-1}-1\right\|_2 \tag{5.55}$$

显然，$\Delta\lambda$ 越满足式(5.52)～式(5.54)约束 L_{SMA} 越接近 0，当约束完全满足时，$L_{\mathrm{SMA}}=0$。因此软单调对齐不需要变更任何网络结构和其他内容，但是缺点也非常明显。在训练的开始阶段难以产生单调对齐，这种约束手段并不严格。因此，EfficientTTS 采用硬间隔对齐方法。

(2) 硬单调对齐(Hard Monotonic Alignment，HMA)，这是能严格保证约束的方法。某种意义上，这是通过修改索引映射矢量 IMV 的定义来达成目的。这种方法会变更原注意力矩阵 $\boldsymbol{\alpha}$ 为新的注意力矩阵 $\boldsymbol{\alpha}'$。

首先，根据原注意力矩阵 $\boldsymbol{\alpha}_{i,j}$，以及索引矢量 $\boldsymbol{p}=\{0,1,\cdots,T_1-1\}$，通过式(5.56)计算 π_j，并记作 π'_j，它不是单调的，我们通过激活函数和 $\Delta\pi>0$ 需要让它变单调。

$$\begin{aligned}&\Delta\pi'_j=\pi'_j-\pi'_{j-1},\quad 0<j\leqslant T_2-1\\&\Delta\pi_j=\mathrm{ReLU}(\Delta\pi'_j),\quad 0<j\leqslant T_2-1\\&\pi_j=\begin{cases}0, & j=0\\\sum\limits_{m=0}^{j}\Delta\pi_m, & 0<j\leqslant T_2-1\end{cases}\end{aligned} \tag{5.56}$$

同时，数值限定在$[0,T_1-1]$里

$$\pi_j^*=\pi_j\cdot\frac{T_1-1}{\max(\pi)}=\pi_j\cdot\frac{T_1-1}{\pi_{T_2-1}} \tag{5.57}$$

就这样，在原先 IMV 的基础上延伸了若干步，于是认为现在定义的 π_j^* 才是真正的索引映射矢量 IMV。这样定义下的 IMV 保证是满足约束的。

但是我们最终需要的是新的注意力矩阵$\boldsymbol{\alpha}'$，通过利用以 π^* 为中心的高斯核重建：

$$\boldsymbol{\alpha}'_{i,j}=\frac{\exp(-\sigma^2(p_i-\pi_j^*)^2)}{\sum\limits_{m=0}^{T_1-1}\exp(-\sigma^{-2}(p_m-\pi_j^*)^2)} \tag{5.58}$$

其中，σ 是超参数，这样得到的$\boldsymbol{\alpha}'$一定符合单调性约束，就这样，达到了优化的目的。

接下来，将通过 EfficientTTS 训练与推理的流程图讲解其框架细节，如图 5.38 所示。

提取文本和 Mel 频谱图的隐藏表示的高维矢量，它的输出分别记作 $\boldsymbol{k}=\{k_1,k_2,\cdots,k_{T_1-1}\}$和 $\boldsymbol{q}=\{q_1,q_2,\cdots,q_{T_2-1}\}$，二者的维度均为 D。

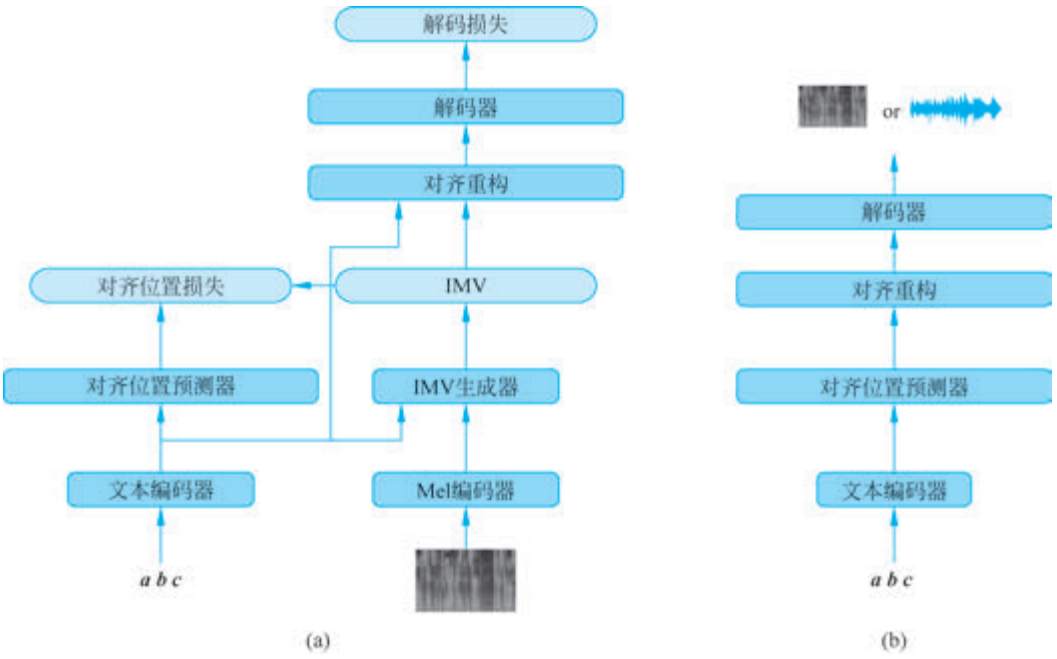

图 5.38　Efficient 训练(a)与推理(b)流程图

该部分流程仅见于训练阶段，作用是根据文本编码器与 Mel 编码器的输出 $\boldsymbol{k}$ 和 $\boldsymbol{q}$ 计算得到索引映射矢量 IMV，具体过程如下。

使用缩放点积注意力模型为打分函数，并投入 Softmax 函数，计算原始未经单调约束的注意力矩阵 $\boldsymbol{\alpha}$，这是注意力机制融入序列到序列模型的常见方法。

$$\boldsymbol{\alpha}_{i,j}=\frac{\exp(-D^{-0.5}(q_j\cdot k_j))}{\sum_{m=0}^{T_1-1}\exp(-D^{-0.5}(q_j\cdot k_m))} \tag{5.59}$$

如果使用软单调对齐(Soft Monotonic Alignment，SMA)，则 IMV 采用式(5.59)定义并计算 π_j 即可。但 EfficientTTS 使用硬单调对齐(Hard Monotonic Alignment，HMA)，则 IMV 采用式(5.56)的定义，并通过式(5.57)计算 π_j^* 。

对于 $\forall j\in[1,2,\cdots,T_2-1]$均如是计算，得到$\boldsymbol{\pi}^*$ 为后续使用的索引映射矢量 IMV，该量符合单调约束。总体训练过程中，IMV 后续在对齐位置预测器的训练和对齐重建模块中都将用到，为了方便表示，后续所有的$\boldsymbol{\pi}^*$ 统一简化记作$\boldsymbol{\pi}$ 。

该网络由卷积构成，在训练和推理过程中都用到了，接收文本编码器的输出 $\boldsymbol{k}$，输出逆索引映射矢量 $\boldsymbol{c}=\{c_i\}(i\in[1,2,\cdots,T_1-1])$。接下来将解释逆索引映射矢量的定义，以及引入该量的必要性，解释过程中也会引入对齐重建模块的内容。这两个组件的设计是彼此依存的。

回顾我们的语音合成目的，即在推理的过程中获取满足单调约束的注意力矩阵$\boldsymbol{\alpha}'$，硬单调对齐的部分中，通过式(5.58)可知，只要有索引映射矢量$\boldsymbol{\pi}_j$，就能得到 $\alpha'_{i,j}$。但是，根据 IMV 生成模块章节可知，$\boldsymbol{\pi}_j$ 的计算需要原始注意力矩阵$\boldsymbol{\alpha}_{i,j}$，而推理过程中，仅有文本编码器的输出 $\boldsymbol{k}$，因此无法通过计算 IMV 的方式得到 $\alpha'_{i,j}$。回顾 π_j 的定义，第 j 时间步输出的

波形点 y_j 在输入序列中的索引映射位置数值，整数列 $j\in[1,2,\cdots,T_2-1]$到实数区间 $\pi_j\in[0,T_1-1]$的“映射”。因此引入一种“逆索引映射”，整数列 $i\in[1,2,\cdots,T_1-1]$到实数区间 $c_i\in[0,T_2-1]$的“映射”，并用一个卷积神经网络系统迂回达成目的。c_i 指第 i 个输入序列 Token 在输出序列中的索引映射位置的数值。c_i 和 π_i 的关系如图 5.39 所示。

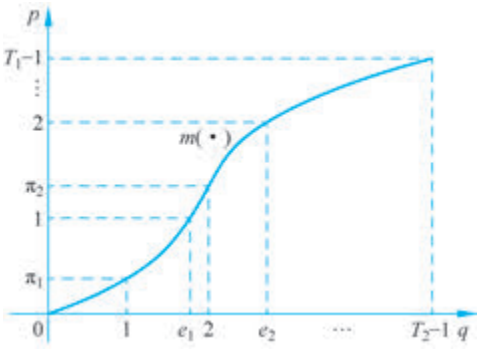

图 5.39　c_i 和 π_j 的关系图(纵轴 p 是输入索引，横轴 p 是输出索引)

依照这样类似反函数的定义，通过索引映射矢量$\boldsymbol{\pi}$求逆索引映射矢量 $\boldsymbol{e}$ 的方法如下，首先计算中间权重矩阵 $\boldsymbol{\gamma}$：

$$\boldsymbol{\gamma}_{i,j}=\frac{\exp(-\sigma^{-2}(p_i-\pi_j)^2)}{\sum_{n=0}^{T_2-1}\exp(-\sigma^{-2}(p_i-\pi_n)^2)} \tag{5.60}$$

并最终由此计算得 e_i：

$$e_i=\sum_{n=0}^{T_2-1}\gamma_{i,n}\cdot q_n \tag{5.61}$$

可见，$\boldsymbol{e}$ 的分辨率与 $\boldsymbol{h}$ 一致，后续可以通过该 $\boldsymbol{e}$ 重建$\boldsymbol{\alpha}'$。

训练管道中，根据 IMV 生成器模块计算的$\boldsymbol{\pi}$，并由式(5.60)、式(5.61)计算标准的 $\boldsymbol{e}$ 作为训练对齐位置预测器网络的目标，实际训练是对 $\Delta e=e_i-e_{i-1}$ 算对齐位置损失(Aligned Position Loss)：

$$L_{\text{ap}}=\|\log(\Delta\hat{e}+\epsilon)-\log(\Delta e+\epsilon)\|_1 \tag{5.62}$$

推理管道中，通过已训练好的对齐位置预测器网络输出得到 $\boldsymbol{e}$。

训练管道中，已经由 IMV 生成器模块和预测器模块计算了$\boldsymbol{\pi}$和 $\boldsymbol{e}$，前者可以直接由式(5.58)重建$\boldsymbol{\alpha}'$。但是推理管道中，只有逆索引映射矢量 $\boldsymbol{e}$，为了保持与推理的一致性，训练管道中也采用 $\boldsymbol{e}$ 来重建$\boldsymbol{\alpha}'$，通过利用以 $\boldsymbol{e}$ 为中心的高斯核计算如下：

$$\alpha'_{i,j}=\frac{\exp(-\sigma^{-2}(e_i-q_j)^2)}{\sum_{m=0}^{T_1-1}\exp(-\sigma^{-2}(e_m-q_j)^2)} \tag{5.63}$$

其中，$\boldsymbol{q}=\{0,1,\cdots,T_2-1\}$是输出序列的索引，根据完全性，序列长度 $T_2=e_{T_1-1}+\Delta e_{T_1-1}$，事实上，对齐位置预测器直接输出的都是 Δe_i，还需逐步累加到 e_i。

推理管道和训练管道都使用式(5.63)重建得 $\boldsymbol{\alpha}'$。

根据 $\boldsymbol{\alpha}'$和式(5.50)，可以根据 $\boldsymbol{h}$ 计算上下文矢量 $\boldsymbol{c}$，$\boldsymbol{c}$ 通过解码器网络 $\boldsymbol{y}=g(\boldsymbol{c})$得到最

终输出。最终输出 $\boldsymbol{y}$ 可以是 Mel 频谱图(此时整个系统即一个语言模型),这样编码器 $g(\cdot)$不会有时间长度上的变化,$\boldsymbol{c}$ 和 $\boldsymbol{y}$ 的长度均为 Mel 频谱图的帧数。当然,$\boldsymbol{y}$ 也可以是波形(此时即文本到波形的端到端系统),这样编码器 $g(\cdot)$在时间长度上还需数百倍的上采样,但是有专门的声码器,例如 5.5.2 节的 HifiGAN、BigVGAN 等,能较好地完成这样的上采样过程,可以作为编码器网络,在端到端管道训练中仅进行微调。

5.6.2　基于变分推理的端到端模型 VITS

基于变分推理的端到端模型(Variational Inference Text-to-Speech,VITS)作为一个完全端到端语音合成系统,它通过将条件变分自编码器(Variational Auto-Encoder,VAE)与对抗学习相结合,解决了传统语音合成模型流程复杂、训练困难等问题,如图 5.40 所示。

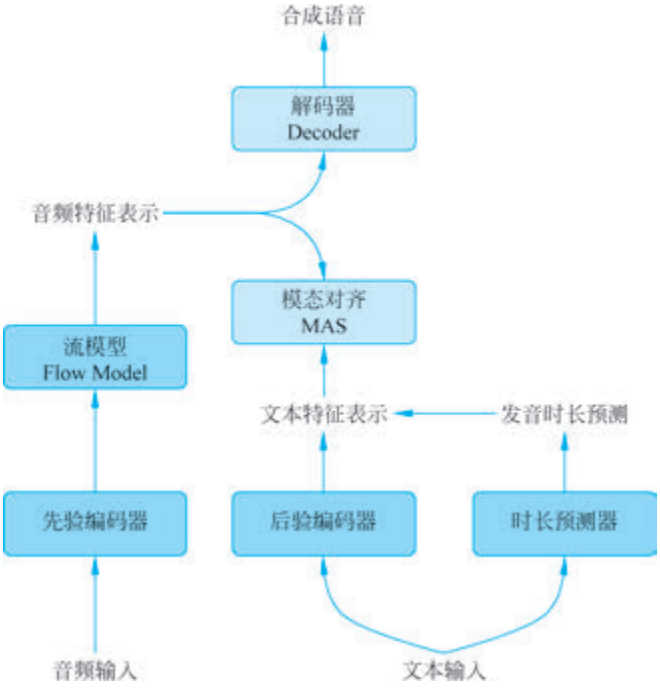

图 5.40　基于变分推理的端到端模型

VITS 模型主要包含三个模块:后验编码器(Posterior Encoder)、先验编码器(Prior Encoder)和解码器(Decoder)。其中后验编码器负责将输入的音频信号编码为潜在表示;先验编码器接收文本序列作为输入,生成对应的先验分布,两个编码器的输出通过单调对齐搜索(Monotonic Alignment Search,MAS)算法进行对齐。解码器则将潜在表示转换回音频波形。针对 VAE 模型的高斯分布假设所带来的描述力弱的问题,VITS 在隐空间引入了流(Flow)方法对分布进行变换,并使用判别器进行对抗训练以提升合成音频的自然度。

VITS 模型突破了传统语音合成模型需要多个独立模块串联的限制,实现了真正的端到端训练,同时合成音质相比传统方法有显著提升。但该模型缺乏对韵律信息的直接建模,导致生成语音过于平淡。

5.7 基于大语言模型的端到端语音合成

基于大语言模型的端到端语音合成利用大语言模型(Large Language Model，LLM)直接从文本生成自然语音，不需要单独优化文本分析、韵律预测、声学模型和声码器等模块，避免了多阶段合成带来的误差积累。相较于非大语言模型的端到端语音合成方法，这类方法借助语言模型的强大生成能力，在自然性、情感表达、上下文理解和流畅度方面表现更为优越。典型的方法包括使用大语言模型与神经网络相结合，直接对文本到音频的映射进行建模，此外，这种端到端语音合成技术还能灵活地控制语速、情感、语调等特征，更加贴近人类语言的多样性需求。

5.7.1 基于神经音频编解码器的零样本语音合成 VALL-E

VALL-E 是基于神经音频编解码器的零样本语音合成(VALL-E)使用从预训练的神经音频编解码器模型生成的离散编码来训练神经编解码器语言模型，并将 TTS 视为条件语言建模任务，而不是连续信号回归如图 5.41 所示。VALL-E 首先使用预训练的神经编解码模型将每个音频样本编码为离散编码，量化后，神经编解码器能够重建波形。得到音频的离散编码之后，将零样本 TTS 转换为条件编解码器语言建模任务，训练神经语言模型来生成以音素序列 $\boldsymbol{x}$ 和声学提示矩阵为条件的声学编码矩阵。声学提示矩阵是通过相同的神经编解码器以目标说话人的语音作为输入获得的。

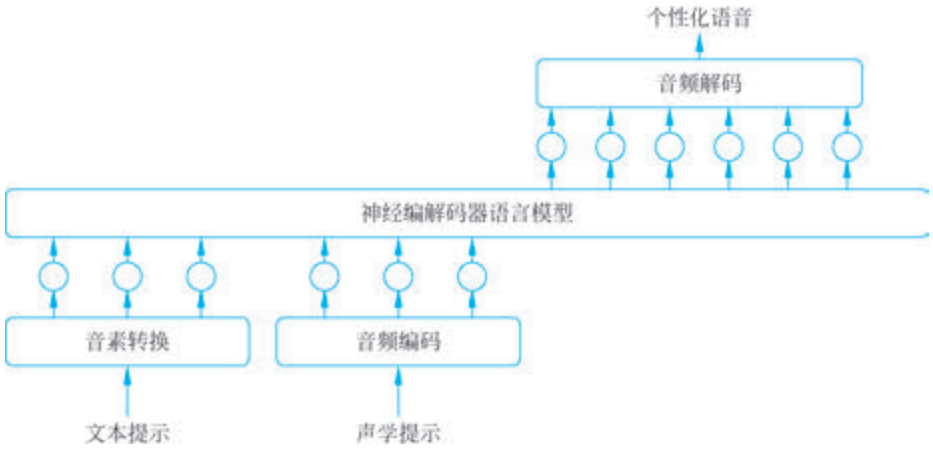

图 5.41 基于神经编解码器语言模型的零样本语音合成

由于神经编解码器通过残差矢量量化将语音编码为了 8 个不同的编码序列，因此 VALL-E 以分层方式设计了两种条件语言模型。对于来自第一个量化器的离散编码，训练仅自回归（AR）解码器的语言模型。它以音素序列和声音提示为条件，自回归生成目标语音的第一层离散编码。对于从第二个到最后一个量化器的离散编码，训练非自回归(NAR)语言模型，以音素序列、声音提示矩阵和前一层的离散编码作为条件，非自回归生成其他 7 个量化器的离散编码。

VALL-E 模型首次采用神经音频编解码器得到的语音离散编码作为语音合成中音频

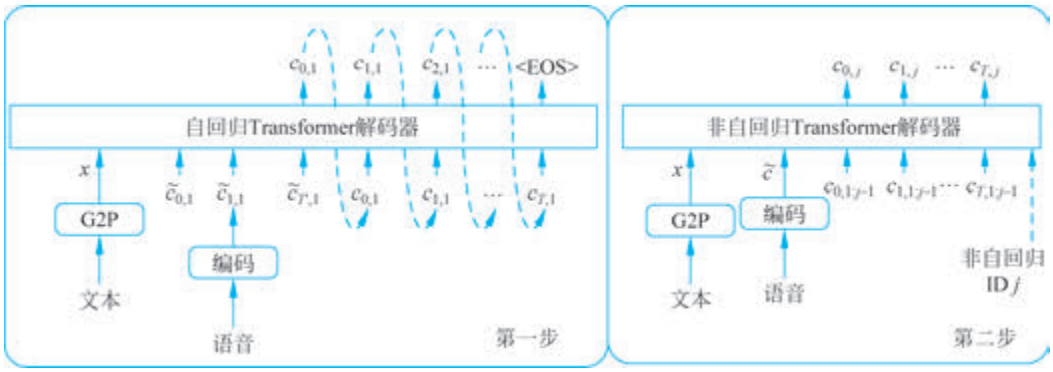

图 5.42　神经编解码器语言模型的两阶段训练

的表征，通过借助语言模型的上下文学习能力，能够以很短的目标说话人音频作为提示来克隆其音色，此外还可以将声学环境和说话者的情感在合成的语音中得到保持。然而两阶段的训练使得具有较高的训练复杂度和推理复杂度，自回归带来了鲁棒性上的一些缺陷，有概率出现丢字、漏字、重复现象，如图 5.42 所示。

5.7.2　基于神经语音编解码器的零样本语音合成 NaturalSpeech 3

NaturalSpeech 3 是一个更具自然度的零样本文本到语音系统，它通过将语音分解为代表不同属性的子空间，并分别生成这些属性，来实现自然语音的零样本合成。如图 5.43 所示，该系统设计了一个新颖的神经语音编解码器（Factorized Vector Quantization Codec，FACodec），该方法将语音分解为 5 个独立属性：持续时间、韵律、内容、声学细节和音色。尽管持续时间是韵律的一部分，但由于系统采用了非自回归的生成方式，该方法单独对持续时间进行建模，并利用内部工具获得音素级的持续时间信息。而对其他属性，首先，本文通过隐式学习并解纠缠出代表不同属性的子空间。随后，系统使用因子化扩散模型分别生成这些属性的表示，并最终通过编解码器根据这些生成的属性重建语音波形。最后，该系统利用一个分解扩散模型（Factorized Diffusion Model），该模型由音素编码器和语音属性扩

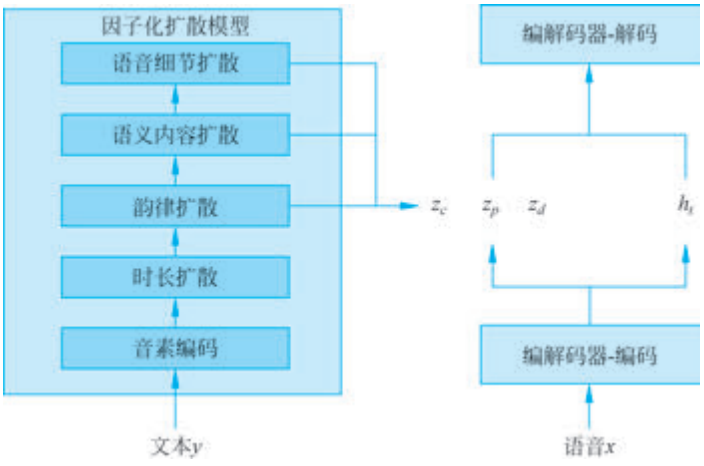

图 5.43　用于语音属性分解的神经语音编解码器和分解扩散模型系统概述

散模块组成，具有相同的离散扩散公式，可以基于相应的提示生成每个子空间中的属性。具体而言，按照持续时间、韵律、内容、声学细节的顺序，仅使用相应的属性提示，并在子空间内应用离散扩散。

NaturalSpeech 3 是旨在生成高质量、多样化的自然语音。它在单说话人和多说话人场景中实现了更接近人类水平的自然度，采用编码器/解码器结构和分解扩散模型来生成语音。与前代相比，NaturalSpeech 3 引入了 FACodec，将语音分解为独立子空间，简化了建模过程，如图 5.44 所示。此外，NaturalSpeech 3 允许通过不同的提示控制不同的语音属性，增强了系统的控制性。在性能上，通过扩展到 10 亿个参数和 20 万小时的训练数据，展示了其在大规模数据和模型规模下的可扩展性。这些优势使得 NaturalSpeech 3 在语音合成领域具有显著的进步和潜力。

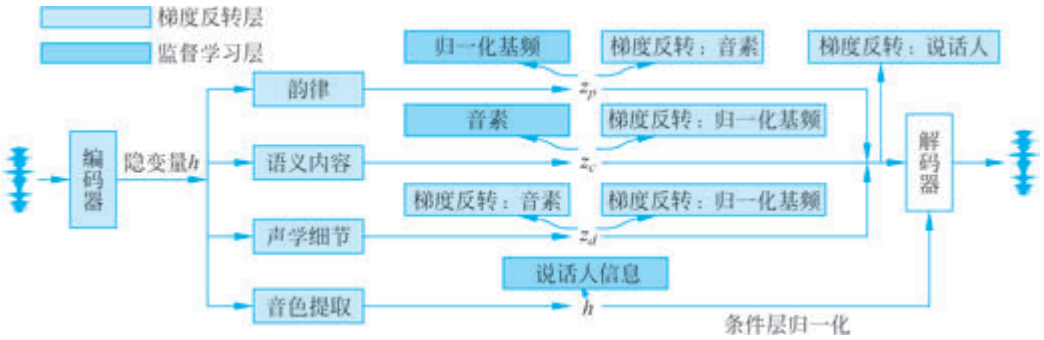

图 5.44　用于属性分解的 FACodec 框架

5.7.3　基于大规模预训练模型的端到端语音合成 SoundStorm

SoundStorm 模型由谷歌公司 DeepMind 团队提出，它通过将扩散模型与自回归采样相结合，解决了传统音频扩散模型采样速度慢的问题。它可视为大规模预训练模型在语音合成领域的应用，旨在通过端到端的方式生成高质量、自然的语音合成。SoundStorm 模型包括三个主要模块：压缩编码器(Compression Encoder)、扩散解码器(Diffusion Decoder)和自回归采样器(Autoregressive Sampler)。其中，压缩编码器负责将原始音频信号压缩到一个更加紧凑的离散表示空间；扩散解码器在这个压缩空间中进行扩散过程，生成高质量的音频表示；自回归采样器则采用创新的分块并行策略，大幅提升了音频生成的速度。在具体实现中，模型使用了 Encodec 作为音频压缩编码器，并采用了条件扩散模型来实现音频的生成，通过提出的块并行采样策略，使得生成速度提升了数个数量级。该模型突破了传统音频扩散模型在推理速度上的瓶颈，同时保持了高质量的音频生成能力。

通过大规模的预训练，SoundStorm 能够在多个语音合成任务中生成自然且多样化的语音，包括情感表达、语速变化等。该模型能够支持流式生成，并实现低延迟、高质量的语音合成效果，在实际应用中具有广泛的前景，如智能语音助手、虚拟客服、个性化语音合成等。

5.7.4　基于自回归语言建模的多功能语音合成模型 Seed-TTS

字节跳动公司提出了一种大规模自回归语音合成模型(Seed-TTS)，能够生成与人类语

音几乎没有区别的语音。Seed-TTS 在语音上下文学习方面表现出色，在说话者相似度和自然度方面的表现与真实的人类语音相媲美，对各种语音属性（例如情感）有着卓越的可控性，并且能够为域外说话人生成高度表现力和多样化的语音。Seed-TTS 提出了一种用于语音分解的自蒸馏方法，以及一种强化学习方法来增强模型的鲁棒性、说话者相似性和可控性。

Seed-TTS 系统由 4 个主要模块组成（图 5.45）：语音离散化编码器、离散编码语言模型、离散编码扩散模型和声学声码器。首先，语音离散化编码器将语音信号转换为语音离散编码序列。然后，训练离散编码语言模型，在文本和语音离散编码的配对序列上进行训练。在推理过程中，自回归地生成语音离散编码。接着，这些生成的离散编码由扩散模型进行处理，以增强声学细节。最后输出被传递到声码器以预测最终波形。

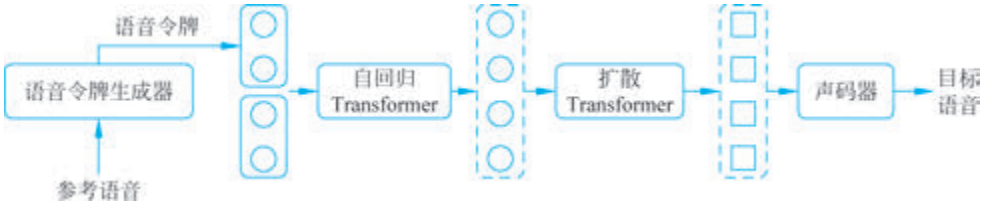

图 5.45　Seed-TTS 的整体框架

Seed-TTS 经历 3 个训练阶段：预训练、微调和后训练。预训练阶段的目标是最大化场景和说话人的覆盖范围，同时为通用语音建模建立强大的骨干。微调阶段包括说话人微调和指令微调。说话人微调的重点是增强选定的一组说话人的性能，而指令微调的目的是提高可控性和交互性。后训练通过强化学习进行，从整体上改进模型。

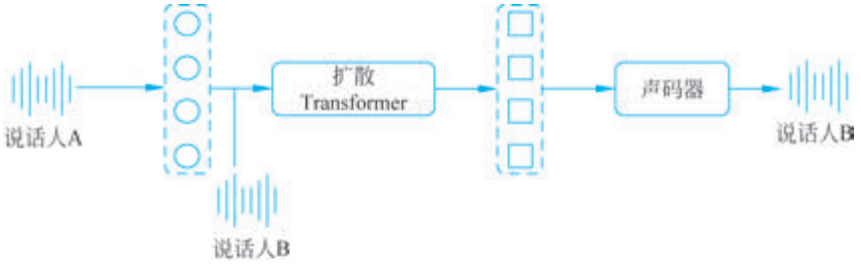

图 5.46　Seed-TTS 系统中零样本语音转换的流程

Seed-TTS 引入了一种自蒸馏方法，旨在提高音色的可控性。语音分解是指将语音分解为各种独立的、解开的属性的过程，对于零样本语音转换和零样本 TTS 至关重要。Seed-TTS 提出了一种自蒸馏方案来实现属性解耦，核心原理是创建受控语音对，这些语音对共享大部分信息，但在一个或几个特定目标属性上有所不同。利用此类数据对以及对模型架构的微调，使 Seed-TTS 模型能够实现高质量的属性解耦。以音色解耦为例，图 5.46 显示了 Seed-TTS 用于零样本语音转换（Voice Conversion，VC）任务的流程。

Seed-TTS 将利用外部奖励模型的强化学习方法，例如 Proximal Policy Optimization（PPO）和 REINFORCE 与不使用外部奖励模型的方法，例如 Direct Preference Optimization（DPO）进行比较，表明这两种方法都是有效的。前者允许对特定语音属性进行清晰的控制，

而后者则实现更加简单。

Seed-TTS 仍然有一些局限性，有时在需要细致入微的情感和情境理解的场景中存在局限性。此外，场景覆盖率仍有改进的空间，如在唱歌或给出包含背景音乐或过多噪音的提示时表现不佳。

5.8 语音合成评价方法

5.8.1 整体性能评测方法

由于语音合成系统最终是要应用于某一领域，使用者关心的是语音合成系统的最终语音输出质量，所以需要从整体上对合成系统的性能进行评比。TC-Star 从第三届的评测开始，就全部采用整体系统的评测，而不再对单个的模块进行评测。目前国际上常用的语音合成系统整体性能评测方法有：

(1) 成对比较测试(Paired Comparison，PC)：被测系统两两相比，听音人需要在二者之间选择一个相对好的，不存在二者相同的情况。

(2) 倾向性测试(Preference Test，PT)：与 PC 类似，也是两两之间相比，但是打分方式不同。一般结果分为：3＝Much Better，2＝Better，1＝ Slightly Better，0＝About the same，－1＝Slightly Worse，－2＝Worse，－3＝Much Worse。

(3) 自由尺度打分(Magnitude Estimation，ME)：最早使用 ME 测试方法对语音合成系统进行测试的是 Pavlovic 和 Espesser(1990)。听音人根据被测系统的输出，根据自己的判断给系统打分，事先不给参考标准，得分也可以在计算机屏幕上用不同长度的线段来表示。

(4) 固定尺度打分(Categorical Estimation，CE)：和 ME 不同的是，CE 一般采用 7、10 或 20 分制，对每一分代表的意义事先加以说明，比如：7 分代表优秀，6 分代表良好等，测试时，听音人根据语音合成系统的输出分类打分。

(5) 反应时间测试(Reaction Time，RT)：记录从被测系统输出声音以后到听音人做出响应的时间，用来衡量听音人接收信息的难易程度。

(6) 主观印象分(Mean Opinion Score，MOS)：听音人根据合成系统的输出给出自己的主观印象分。MOS 评测中，一般采用 5 分制：1＝Bad，2＝Poor，3＝Fair，4＝Good，5＝Excellent。如表 5.4 所示。MOS 评测要求有较多数量的听音人，同时要设计较为科学的测试材料和测试项目。这样，测试的结果才能真正地反映出系统的性能。

表 5.4 MOS 分大致等级及其对应的解释

MOS 分数	质量	解　　释
4.0～4.5	电话质量	接近清晰，“真人”发音质量
3.5～4.0	交流	自然，可懂度高，可以用来电话交流，语音质量降低很明显。
2.5～3.5	合成	通常可懂，不能识别出说话者，自然度较差

从上面可以看出，对于系统整体性能评测，一般使用听音人对被测系统的输出语音打分，尽管打分的方法、尺度有所不同。在中文合成语音性能全国评测中，目前用 MOS 分来

衡量参测系统的自然度，这和国际上通用的测试方法是一致的。但是，需要指出的是，整体性能测试不能清楚地指出系统的问题所在。语音合成系统目前的输出语音质量还无法和语音编码的语音质量相比较，因此需要一些不同层次特定性能测试方法来对合成系统的弱点进行诊断。

5.8.2　语音合成评价内容

TC-Star 的评测内容划分很细，相对比较科学。主要包括：整体评分，听音人听音时的努力程度、可懂度、发音、清晰度、语速、自然度、长时间舒适度、听者的舒适度、语流的平稳度。这些方面可以说是对合成系统的指标要求，既有总体上的要求，也有一些指标主要是针对层面的要求。其中整体评分和可懂度是相对重要的两个方面。

国际电信联盟(International Telecommunication Union，ITU)也给出了 ITU-TS P8S 建议书草案，即《语音输出设备质量的主观评价》。该草案对系统的测试使用 10～30 个语音测试材料，分 8 方面：整体印象、可接受度、收听效果、理解程度、清晰度、发音标准度、讲话速度、声音的悦耳程度，用 5 分制对系统进行打分，然后根据系统的得分来衡量系统的好坏。

可懂度的评测与整体评分略有不同，它采用听写的形式。评测一般采用的都是有意义的句子，但是如何衡量句子意义对结果的影响是一件很不容易的事。因此，Nye 和 Gaitenby(1974)构造了 100 个语法固定但语义不规则的句子，称作 Haskins 句法句，用这种句子可较灵敏地测试不同的系统。

基于同样的考虑，原欧洲共同体 SAM 项目使用了语义不可预测句(Semantic Unpredictable Sentence，SUS)(Beno，C.，M. Grice and V. Hazan，1996)对不同语言的文语转换系统进行测试(1989)。SUS 共有 5 种不同的句法结构，从词典中按词性不同随机选取一些词构成句子。这种测试方法最主要的优点是适用于不同的语言。测试材料比较公平，并且可以组成许多句子。从对英语、德语和法语语音合成系统使用 SUS 测试的结果来看，SUS 测试的缺点较少，一些词频影响和记忆问题主要是由于词典的限制引起的。SUS 的测试方法一直沿用至今，Blizzard 的评测中主要涉及 3 个方面：MOS 分、错误率、SUS 测试。

句子的可懂度测试还可以通过句子判断方法来完成，也就是反应时间测试。这种方法要求听音人尽快对句子的正确与否作出判断。听音人的响应时间和判断的准确度可以灵敏地反映在句子层面上的可懂度。TC-Star 的中的测试与这些测试方法类似。

国家规定的文语转换和语音识别系统评测规范中指出，语音学模块评测内容包括语音清晰度(Articulation)测试和语音自然度(Naturallity)测试两部分。语音清晰度是指输出语音是否容易听清楚；语音自然度是指输出语音听起来是否自然。语音清晰度测试又可进一步分为音节清晰度测试、单词清晰度测试和单句清晰度测试。

整个文语转换评测除了进行语音学模块的评测之外，还应当进行语言学模块的评测。语音学模块评测内容包括语音清晰度测试和语音自然度测试。语言学处理在语音合成中起着重要作用，分词、多音字、韵律等处理是否正确，都直接影响合成系统的输出，所以它们都是评测的目标。具体地说，语言学模块评测内容包括切词、多音字、数字串、符号和单位等的文本处理能力的测试。

Barber 等(1989)进行了意大利语的字位到音位的转换规则测试，Willemse(1987)对荷

兰语字位到音位转换规则。Barber 等(1989)用同样的意大利语测试材料对语音合成系统词重音规则进行了测试。Baart 和 Heemskerk(1988)对荷兰语合成系统词素分析的性能进行了评测。

但是在重音的测试方面,要做到自动评测就比较困难。一方面,词典中的词一般是有限的,而句子的集合可以说是无限的,同时句子还和上下文有关系;另一方面,一个句子的重音位置大部分不是唯一的。也就是说,不同的人可能认为重音在不同的地方。因此,在这种情况下,评测只能采用"专家"评判的方法。Van Bezooijen 和 Pols(1989)对荷兰语句子的重音标定算法进行了评测。

语音合成系统的评测是一个很重要的工作,目前的评测大多采用主观测试,为测试要建立言语资料库和言语输入、输出系统评价方法标准化等工作。这些都使得评测成为一项内容复杂、规模巨大、费时费力的工作。系统开发人员不能随时对系统进行评测,以便观察系统是否有所改进。已经有大量研究人员和机构投入这方面的研究中。

5.9 本章小结

本章系统介绍了语音合成的相关概念和技术,包括语音合成的前端处理、传统的语音合成技术、端到端语音合成技术、基于大语言模型的端到端语音合成以及语音合成评价指标等内容。目前,随着深度学习和大模型技术的迅速发展,语音合成的新技术、新方法层出不穷,合成语音可以达到与人类语音媲美的效果。目前,语音合成技术在自然度、情感表达、个性化定制、计算资源需求和多语言支持等方面仍存在一定不足。未来的发展趋势包括提高语音的情感表达和风格个性化、降低计算需求以支持实时生成、多语言和跨口音的无缝支持、实现语音与其他模态的融合,以及通过自监督学习和少样本学习降低数据需求。随着技术的不断进步,语音合成将朝着更自然、高效、个性化的方向发展,推动更广泛的应用。

习题

(1) 语音合成前端文本处理主要分为哪几个关键步骤?请简述各部分的主要功能。
(2) 参数语音合成与波形拼接语音合成相比优缺点有哪些?
(3) 波形拼接语音合成的关键技术有哪些?目标代价是如何定义与计算的?
(4) 端到端语音合成与管道式参数语音合成的结构主要区别是什么?有什么优势?
(5) 请简述语音合成中 Tacotron 的主要模型结构,并给出后续优化的建议。

参考文献

[1] ZEN H,TOKUDA K,BLACK A. Statistical parametric speech synthesis[J]. Speech Communication, 2009,51(11): 1039-1064.

[2] HUNT A W,BLACK A W. Unit selection in a concatenative speech synthesis system using a large speech database[C]//Proceedings of 1996 IEEE International Conference on Acoustics, Speech and

Signal Processing. Atlanta：IEEE，1996：373-376.

[3] WANG Y，SKERRY-RYAN R，STANTON D，et al. Tacotron：Towards end-to-end speech synthesis [C]//Proceedings of the 18th Annual Conference of the International Speech Communication Association. Stockholm：ISCA，2017：4006-4010.

[4] SHEN J，PANG R，WEISS R J，et al. Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions [C]//Proceedings of 2018 IEEE International Conference on Acoustics，Speech and Signal Processing. Calgary：IEEE，2018：373-376.

[5] ZEN H，SENIOR A，SCHUSTER M. Statistical parametric speech synthesis using deep neural networks[C]//Proceedings of 2013 IEEE International Conference on Acoustics，Speech and Signal Processing. Vancouver：IEEE，2013：7962-7966.

[6] ZEN H，SAK H. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis[C]//Proceedings of 2015 IEEE International Conference on Acoustics，Speech and Signal Processing. South Brisbane：IEEE，2015：4470-4474.

[7] FAN Y，QIAN Y，XIE F，et al. TTS synthesis with bidirectional LSTM based recurrent neural networks[C]//Proceedings of the 15th Annual Conference of the International Speech Communication Association. Singapore：ISCA，2014：1964-1968.

[8] QIAN Y，FAN Y，HU W，et al. On the training aspects of deep neural network (DNN) for parametric TTS synthesis [C]//Proceedings of 2014 IEEE International Conference on Acoustics，Speech and Signal Processing. Florence：IEEE，2014：3829-3833.

[9] WU Z，VALENTINI-BOTINHAO C，WATTS O，et al. Deep neural network employing multi-task learning and stacked bottleneck features for speech synthesis [C]//Proceedings of 2015 IEEE International Conference on Acoustics，Speech and Signal Processing. South Brisbane：IEEE，2015：4460-4464.

[10] FAN Y，QIAN Y，SOONG F K，et al. Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis[C]//Proceedings of 2015 IEEE International Conference on Acoustics，Speech and Signal Processing. South Brisbane：IEEE，2015：4475-4479.

[11] VALIN J M，SKOGLUND J. LPCNET：Improving neural speech synthesis through linear prediction [C]//Proceedings of 2019 IEEE International Conference on Acoustics，Speech and Signal Processing. Brighton：IEEE，2019：4384-4388.