小数据下的音素级别说话人嵌入的语音合成自适应方法

徐志航[1),2)] 陈 博[1),2)] 张 辉[3)] 俞 凯[1),2)]

[1)](上海交通大学人工智能研究院人工智能教育部重点实验室 上海 200240)

[2)](上海交通大学计算机科学与工程系跨媒体语言智能实验室 上海 200240)

[3)](苏州思必驰信息科技有限公司 江苏 苏州 215000)

摘 要 在语音合成中,使用少量的用户录制数据进行说话人自适应一直面临着一个问题:如何在不过分降低合成声音的自然度的情况下,提高合成声音的相似度.现有的句子级别、帧级别说话人嵌入等自适应方法在合成训练集外说话人声音时会出现低相似度的问题.使用少量的用户录制数据微调预训练的语音合成模型的自适应方法尽管能提升合成音频的相似度,但是也常伴随着自然度的下降.为了解决这个问题,本文提出了一种基于音素级别的说话人嵌入的语音合成自适应方法.在训练阶段,从真实的特征片段中提取音素级别的说话人嵌入,控制语音合成模型的训练.在自适应阶段,通过对说话人嵌入预测网络进行快速自适应,在推理阶段代替真实音频得到音素级别说话人嵌入帮助模型合成音频.实验使用了少量真实的用户录制数据,对现在主流的不同粒度的说话人嵌入方法进行了性能比较.实验表明,相比较各种不同的说话人嵌入方法,本文提出的方法在不更新语音合成模型的情况下保持自然度不明显下降,并取得了最好相似度;在更新语音合成模型的情况下,该方法同时达到了最好的自然度和相似度.分析发现音素级别的说话人嵌入方法在几乎不增加自适应训练时间的情况下,提供了更好的模型自适应初始点,有效地提高了自适应模型合成声音的质量.

关键词 语音合成;说话人嵌入;时长模型;小数据;说话人自适应

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2022.01003

Speech Synthesis Adaption Method Based on Phoneme-Level Speaker Embedding Under Small Data

XU Zhi-Hang[1),2)] CHEN Bo[1),2)] ZHANG Hui[3)] YU Kai[1),2)]

[1)](*MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai* 200240)

[2)](*Lab of Cross-Media Language Intelligence, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240)

[3)](*AiSpeech Ltd, Suzhou, Jiangsu* 215000)

Abstract In speech synthesis, the use of a small amount of user-recorded data for speaker adaptation has always been faced with a problem: how to synthesize highly similar speeches without excessively reducing the naturalness of the synthesized speeches. The existing utterance-level and frame-level speaker embedding methods face the problem of low similarity when synthesizing speeches of testing speaker, and the use of a small amount of user recorded data to fine-tune the pre-trained speech synthesis model can improve the similarity of synthesized audio, but it is often accompanied by a decrease in naturalness. To solve this problem, we propose a novel adaptation method for speech synthesis based on phoneme-level speaker embedding. In the training stage, the phoneme-level speaker embedding is extracted from the real feature fragments to control the training of the

收稿日期:2020-11-23;在线发布日期:2021-08-09.**徐志航**,硕士研究生,主要研究方向为语音合成. E-mail: 416766213@qq.com. **陈 博**,博士研究生,主要研究方向为语音合成、声音转换等. **张 辉**,硕士,主要研究方向为语音合成. **俞 凯**(通信作者),博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为认知型对话系统、语音合成、语音识别、理解及机器学习等. E-mail: kai.yu@sjtu.edu.cn.

speech synthesis model. In the adaptation stage, we quickly adapt the speaker embedding predictor network, replacing the real audio in the inference stage to obtain phoneme-level speaker embedding. We use a small amount of real user-recorded data to conduct experiments, and compare the performance of common speaker embedding methods in different grains. Experiments show that compared with various speaker embedding methods, our method maintains no significant decrease in naturalness without updating the speech synthesis model, and achieves the best similarity; in the case of updating the speech synthesis model, our method achieves the best naturalness and similarity at the same time. The analysis found that the phoneme-level speaker embedding method provides a better initial point of model adaptation without increasing the adaptive training time, and effectively improves the quality of the synthesized speeches of the adaptive model.

Keywords text to speech; speaker embedding; duration model; small data; speaker adaptation

1 引 言

近年来,随着移动设备的普及,使用语音的人机交互场景变得越来越常见.语音作为人类最重要和最自然的交流方式,是人机交互系统最自然的入口,广泛地应用在不同的人机交互场景中.完整的基于语音的人机交互系统包括用户的询问、机器识别和理解、自然语言生成、语音合成(Text-To-Speech, TTS),最终以音频信号的方式反馈给用户.因此合成高清晰度、高自然度和多样性的语音是人机交互系统中不可或缺的一环.

在深度学习的帮助下,端到端文本转语音系统已经可以合成高质量的语音[1-3].谷歌提出的WaveNet[4]、Tacotron[1]端到端语音合成框架在语音合成领域已经产生了颠覆性的影响,其合成音频几乎可以骗过人耳.但语音合成问题的尚未完全解决,在多样性的语音合成、更加可控的语音合成、小数据语音合成等个性化语音合成的研究方向上仍然有很广阔的研究空间.个性化语音合成中一个常见的使用场景是仅使用少量的用户语音数据,定制个性化的语音助手、地图导航等.这种场景下的录制数据通常很少,并且用户录制数据本身就存在问题和缺陷,例如:语音质量差、背景噪音、房间混响、错字漏字等.据我们所知,至今还没有完全鲁棒的方法可以搭建小数据上的端到端语音合成系统[5-6].

在实际应用场景中,用户录制的数据一般不足以从零训练一个端到端语音合成模型,因而最常用的解决办法是从预训练的模型进行说话人自适应(Speaker Adaptation)[2,7-14].说话人自适应中最直接的做法是根据测试数据对预训练的模型参数进行全部或者部分的重训练和更新[8,10].

由于测试数据存在标注错误、数据量稀少、噪音混响等缺陷,参数更新后的模型会出现过拟合、声音质量不稳定的问题.其中有一种特殊的说话人自适应方法只更新模型当中说话人嵌入,自适应说话人嵌入的方法不更新语音合成模型的其他参数,因此能保证合成语音的自然度[7,15-17],但这种方法提供的说话人音色信息有限,很难合成高相似度的语音.

为了解决这些问题,本文提出了基于音素级别的说话人嵌入的语音合成自适应方法,并将其实现在了基于独立时长模型的端到端语音合成框架之上.在训练阶段,模型从真实的参考音素特征片段当中提取音素级别的说话人嵌入进行训练.在自适应阶段,仅需对预训练的说话人嵌入预测网络进行自适应训练.在测试阶段使用说话人嵌入预测网络预测音素级别说话人嵌入合成音频.

本文中使用的独立的时长模型和[18-19]思路相同,但是实现上都略有不同.为了契合小数据的自适应任务,本文没有采用音素内的局部注意力机制[18],也没有采用说话人相关的时长模型[19],而是采用了说话人无关的时长模型以提高系统的鲁棒性.

本文的最主要贡献是:

(1)提出了细粒度的音素级别说话人嵌入的高效的自适应方法.该方法采用参考编码器和说话人嵌入预测网络分别在训练和测试阶段生成音素级别的说话人嵌入,在保证自然度的同时,稳定提高了在测试说话人上的相似度.

(2)引入了独立的说话人无关的时长模型代替注意力机制,保证测试说话人仍能合成韵律稳定、自然的声音.

(3)对现在主流的说话人嵌入方法进行了性能

比较.在不更新与更新声学模型参数的两种实验比较下,本文提出的音素级别的说话人嵌入方法都取得了最好的相似度,在更新声学模型参数的情况下甚至取得了最好的自然度.

本文第 2 节将回顾语音合成的技术的现状,介绍语音合成的说话人自适应技术;在第 3 节中将详细介绍提出的音素级别说话人嵌入上的说话人自适应的模型方法;第 4 节给出实验的设置、基线系统以及评价标准的介绍;第 5 节将给出实验结果和分析的描述;第 6 节为本文的结论.

2　相关工作

2.1　语音合成技术

传统的文字转语音系统一般分为前端文本处理,后端声学模型和声码器.在现有语音合成系统中,前端对文本上下文信息的表征包含词长、词性、节奏、元辅音等文本信息.后端通过隐马尔可夫模型对基频、梅尔倒谱系数和频带非周期性成分等声学特征进行时序建模.但是构建这样的不同模块需要较强的领域专业知识,而近几年提出的端到端的语音合成框架降低了语音合成的门槛.

2016 年 9 月,谷歌 DeepMind 团队提出的 WaveNet[4] 端到端语音合成模型.紧接着,谷歌陆续提出了 Tacotron[1]、Parallel WaveNet[20]、Tacotron2[21] 等多种端到端合成系统.在端到端语音合成框架方面,百度的研究团队提出了基于卷积神经网络的多说话人的端到端语音合成框架[2,22].很快,有更多的研究者提出了更多不同的端到端合成框架,包括基于教师-学生机制的 Transformer 语音合成框架[23] 和非自回归的 FastSpeech 框架[19,24].

语音合成的另一个重要研究方向就是如何对神经声码器进行进一步的优化和加速.其中有围绕着对类 WaveNet 的自回归神经声码器的加速和轻量化[9,25-26] 等.加速的另一方向是通过放弃自回归的方式来加速模型的训练[20,27-28].

2.2　说话人表征技术

说话人表征的概念来自于说话人确认、说话人分辨任务,近年来逐渐开始被运用到了语音合成领域[7,16,29].说话人表征提取则是从声学特征中通过统计模型提取说话人相关信息,并对不同的说话人抽象出相应的特征表示,称为说话人嵌入(Speaker Embedding).

在深度学习开始流行之前,说话人建模最常用、效果最好的方法是基于高斯混合模型-通用背景模型的 i-vector[30].随后随着深度学习的发展,d-vector[13] 提出了利用深度神经网络进行说话人表征学习的新框架.x-vector[31] 是目前最流行的基于段级别优化方法的说话人表征,相对于 d-vector 而言,引入了段级别的优化方法之外,也采用了建模能力更强的时延神经网络作为说话人表征的提取器.

而在语音合成任务中,有时也会使用说话人表征对说话人音色进行建模.其中最常用的表征向量有:独热表(One-Hot Table)和查找表(Look-Up Table)[2].与此同时,当需要更加复杂的表征时,就可能需要使用参考音频和参考编码器[32-33] 来对模型进行联合训练.这些说话人表征会被设计作为额外的条件输入或者控制变量[11,15,34],影响的模型训练和输出,而后续的说话人自适应也会围绕着说话人嵌入展开讨论.

2.3　说话人自适应技术

小数据上的语音合成任务受限于其稀少的训练数据,一般无法直接训练一个端到端模型.以 Tacotron 模型为例,一般需要 10 个小时的单说话人的干净训练数据,或者使用多个说话人的数据并引入说话人相关的参数加以控制.完成初始模型的训练后,就可以使用目标说话人的数据,在预训练的模型上进行自适应训练,更新全部或者部分模型参数,来拟合新的说话人数据.

在说话人自适应当中,可以更新说话人嵌入来控制声音的音色.一般来说这种方法在集内数据,或者相似分布数据上有很好的自然度.但是该方法无法直接更新其他模型参数,通常对于新数据欠拟合,导致合成的音频经常会和原始说话人不够相似[7].一些研究人员提出了使用在说话人识别领域更加先进的说话人嵌入[16] 和数据增强的办法[29] 增加说话人嵌入本身的覆盖程度.文献[17]提出了基于注意力机制的说话人嵌入方法,相比较句子级别、说话人级别的说话人嵌入,这种方法直接提取和利用帧级别的特征,得到更加丰富说话人嵌入.文献[12]提出了使用说话人表征技术提取全局和音素相关的局部说话人嵌入,并使用注意力机制动态融合的做法,其方法和本文最大的区别是本文使用参考音频和嵌入预测网络的方式生成音素级别的说话人嵌入,而不是基于说话人确认中的表征技术.

除了只更新说话人嵌入,也可以使用目标说话人的文本音频数据,对网络中的其他全部或部分参数进行更新.很多研究[15,35-37] 都使用测试数据微调全

部或者部分的预训练模型.一些研究者将说话人编码网络与声学模型[10,15]或声码器[8]联合训练,并使用测试数据[36,38]进行微调.由于使用少量数据对大规模的模型参数进行微调容易导致[15]过拟合,因此一些自适应技术如说话者自适应训练[39]、LHUC[40]等方法也有被使用到语音合成自适应当中[41-42].

总结来说,为了得到用户定制的个性化语音合成系统,最常用的办法是在单(多)说话人语音合成系统的基础上进行说话人自适应,以此来达到对新说话人的建模.但是说话人自适应技术依然存在一些问题,基于现有说话人嵌入的自适应方法的相似度不足,而使用少量数据直接更新模型参数的经常会导致过拟合从而自然度下降或合成的不稳定.

3 模型方法

本节详细介绍了提出的基于音素级别的说话人嵌入的语音合成自适应方法.整个模型主要分为:端到端语音合成框架、参考编码器网络、时长预测网络、说话人嵌入预测网络.

在训练阶段,模型需要输入文本和对应的参考特征,通过参考编码器提取音素级别的说话人嵌入,训练端到端语音合成框架和参考编码器网络,同时对时长预测网络和说话人嵌入预测网络进行预训练.在自适应阶段,需要使用测试说话人的数据对说话人嵌入预测网络进行自适应训练,让其拟合目标说话人音素级别的说话人嵌入的分布.在测试阶段,仅输入文本,使用说话人嵌入预测网络代替参考编码器提供音素级别的说话人嵌入合成音频.

为了方便介绍,本文约定训练数据文本(音素)序列为 $\boldsymbol{t}=(\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_n)$,音频信号为 $\boldsymbol{A}$,其对应的特征序列为 $\boldsymbol{O}=(\boldsymbol{o}_1,\boldsymbol{o}_2,\cdots,\boldsymbol{o}_m)$.

3.1 端到端语音合成模型框架

端到端语音合成模型框架由基于 Tacotron2 声学模型和 LPCNet 声码器组成.Tacotron2 模型主要由编码器,注意力模块和解码器组成,本文参考文献[18]中对于 Tacotron 的注意力机制的修改,采用独立时长模型,但同时保留了解码器的自回归预测机制以避免过平滑问题,声学模型示意图绘于图 1.

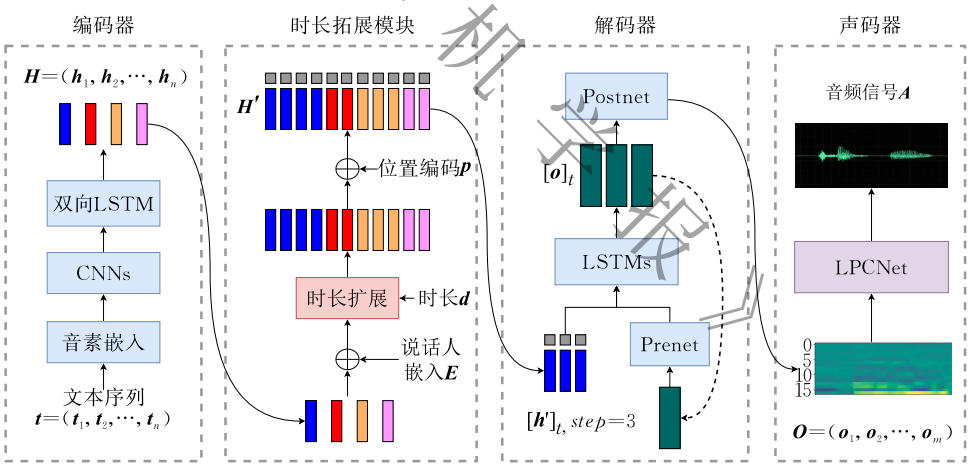


图 1 使用独立时长模型的端到端语音合成模型框架结构(⊕表示维度拼接)

在输入编码器前,文本会经过文本标准化,通过拼音字典转换成带调拼音音素序列,然后使用一个查找表把音素序列转换成可训练的音素嵌入,经过 5 层卷积神经网络和 1 层双向 LSTM,得到了编码器输出 $\boldsymbol{H}=(\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_n)$.

原 Tacotron2 模型使用的注意力机制中,注意力机制网络会在解码器的每一步中将编码器输出 $\boldsymbol{H}$ 整合为固定长度的上下文向量.但是我们的实验中,为了保证合成音频不会因为注意力机制崩溃,我们使用了音素的真实时长对编码器输出进行时长拓展.编码器输出 $\boldsymbol{H}=(\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_n)$ 进行时长拓展将得到和声学特征长度 m 相同的隐层序列 $\boldsymbol{H}'$.这样对于解码器来说,编码器输出和输出特征的长度将严格对齐.因为在推理阶段无法获得对应文本的时长信息,我们会额外训练一个时长预测网络,输入 $\boldsymbol{H}$ 预测对应音素的发音时长.

Tacotron2 模型的解码器是一个自回归递归神经网络,假设当前解码器时刻为 t,它输入前一帧的解码输出 $\boldsymbol{o}_{t-1}$ 和编码器输出 $\boldsymbol{h}'_t$ 预测当前帧声学特征 $\boldsymbol{o}_t$.前一帧的输出 $\boldsymbol{o}_{t-1}$ 会先经过一个带 dropout 的非线性层预处理网络 Prenet,和当前帧帧对应的编码器输出 $\boldsymbol{h}'_t$ 进行维度拼接,经过两层解码网络

LSTM，预测得到当前帧的声学特征 $\boldsymbol{o}_t$. 为了继续使用 Tacotron 中的多步解码的加速技巧，将输出目标 $\boldsymbol{O}$ 进行多帧折叠. 假设折叠步数为 $step$，此时 t 时刻解码器的输入记作 $[\boldsymbol{h}']_t=(\boldsymbol{h}'_{t\times step+1},\boldsymbol{h}'_{t\times step+2},\cdots,\boldsymbol{h}'_{(t+1)\times step})$，$t$ 时刻的解码目标变为 $[\boldsymbol{o}]_t=(\boldsymbol{o}_{t\times step},\boldsymbol{o}_{t\times step+1},\cdots,\boldsymbol{o}_{t\times step+step-1})$，而整个解码器的解码步数就得到了 $step$ 倍的减少.

Tacotron2 模型的解码器网络为了更好地训练文本和声学特征之间的对其关系，会对注意力机制计算后的上下文向量也使用自回归生成. 其实现方法是在解码器的两层 LSTM 之前输入前一时刻的上下文向量，在 LSTM 之后输入当前时刻的上下文向量. 由于本文使用的独立时长模型不需要计算注意力，因此没有使用这种上下文向量的自回归生成，而是直接在解码器的 LSTM 中输入时长拓展后的当前时刻的编码器输出.

最后，解码器的输出 $\hat{\boldsymbol{O}}$ 还会经过一个由 5 层卷积神经网络组成的后处理网络 Postnet 得到残差 $\hat{\boldsymbol{O}}_{res}$ 并求和，计算得到最终预测特征 $\hat{\boldsymbol{O}}_{post}=\hat{\boldsymbol{O}}+\hat{\boldsymbol{O}}_{res}$.

3.2　时长拓展模块

因为音素序列 $\boldsymbol{t}$ 和声学特征序列 $\boldsymbol{O}$ 之间有长度的差别，在语音合成模型当中一般会显式或者隐式地引入时长模型. 时长模型简单来说建模了一个文本（音素）的在当前上下文中对应的发音长度. 因为声学特征提取一般是分帧的，所以这里的时长一般指声学特征的帧数.

在训练阶段，因为训练数据和对应文本是固定的，我们可以使用 ASR 的强制对齐信息从音频特征当中提取对应的时长信息 $\boldsymbol{d}=(\boldsymbol{d}_1,\boldsymbol{d}_2,\cdots,\boldsymbol{d}_n)$，其中 d_i 表示第 i 个音素的时长. 在时长拓展模块中，会对编码器隐层 $\boldsymbol{H}$ 和对应的时长序列 $\boldsymbol{d}$ 进行重复拓展：

$$\boldsymbol{H}'=\underbrace{(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_1)}_{\boldsymbol{d}_1},\underbrace{(\boldsymbol{h}_2,\cdots,\boldsymbol{h}_2)}_{\boldsymbol{d}_2},\cdots,\underbrace{(\boldsymbol{h}_n,\cdots,\boldsymbol{h}_n)}_{\boldsymbol{d}_n}$$

在测试阶段，会使用时长预测网络提供集外文本的时长信息. 时长预测网络输入编码器隐层 $\boldsymbol{H}$，经过 1 层双向 LSTM，再经过一次线性映射得到时长预测值序列 $\hat{\boldsymbol{d}}=(\hat{\boldsymbol{d}}_1,\hat{\boldsymbol{d}}_2,\cdots,\hat{\boldsymbol{d}}_n)$.

为了区分时长扩展之后的编码器隐层中同一个音素当中的不同帧，本文采用了传统参数语音合成当中常用的相对位置编码方法. 假设音素 t_i 对应的编码器隐层 $\boldsymbol{h}_i$ 经过时长拓展之后为 $\underbrace{(\boldsymbol{h}_i,\boldsymbol{h}_i,\cdots,\boldsymbol{h}_i)}_{\boldsymbol{d}_1}$，其位置编码就可以表示为 $\boldsymbol{p}_i=(1/\boldsymbol{d}_i,2/\boldsymbol{d}_i,\cdots,\boldsymbol{d}_i/\boldsymbol{d}_i)$. 同时，为了让时长的分布更接近高斯分布、更易训练，在训练时候会将时长转换成 log 域，在完成预测后再转回线性域并向上取整.

本小节最后，对本文采用的说话人无关时长预测网络的设计思路进行解释. 在本文不使用说话人相关的时长模型是有意为之，包括采用独立的时长模型代替注意力机制，都是为了提升最终在目标说话人上的声音合成质量. 从评价标准讨论，在真实的小数据说话人声音复刻使用场景中，用户对合成声音的自然度、相似度的要求最高，而说话节奏（即时长）和目标说话人相近并不是其核心指标. 从数据情况讨论，用户录制数据无法避免出现错字漏字等标注问题，因此测试数据的时长信息并不一定准确的，而且实际录制中用户的说话节奏可能也并不一定自然；退一步假设标注完全正确，用户说话节奏自然，时长提取完全准确，在实验中使用该时长信息对时长模型进行说话人自适应训练，也同样会面临着数据量过小而过拟合的问题.

3.3　参考编码器网络

如图 2 所示，在上述端到端语音合成框架的基础上，本文引入了参考编码器网络提取音素级别的说话人嵌入. 参考编码网络输入当前目标声学特征序列 $\boldsymbol{O}$，输出和音素序列相同长度的说话人嵌入 $\boldsymbol{E}=(\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_n)$.

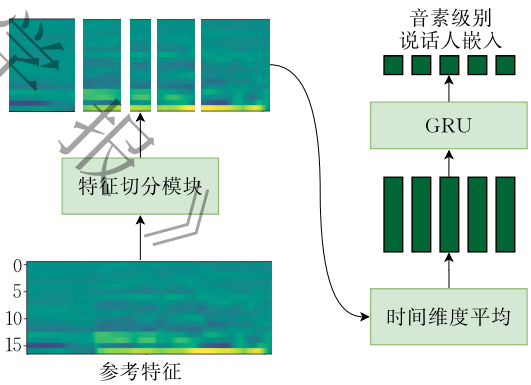


图 2　参考编码器网络示意图

在参考编码器网络中，声学特征 $\boldsymbol{O}$ 主要经过两个步骤得到对应的说话人嵌入 $\boldsymbol{E}$. 首先通过时长信息将声学特征 $\boldsymbol{O}$ 切分为长度和音素长度相同的片段序列 $\boldsymbol{S}=(\boldsymbol{s}^{(1)},\boldsymbol{s}^{(2)},\cdots,\boldsymbol{s}^{(n)})$，其中 $\boldsymbol{s}^{(i)}$ 表示第 i 个音素对应的声学特征片段. 将每个 $\boldsymbol{s}^{(i)}$ 经过时间维度的平均可以得到特征均值序列 $\{\bar{\boldsymbol{s}}\}_n=(\bar{\boldsymbol{s}}_1,\bar{\boldsymbol{s}}_2,\cdots,\bar{\boldsymbol{s}}_n)$，再经过一层 GRU 和一层线性映射就得到了本文提出的音素级别的说话人嵌入 $\boldsymbol{E}$.

音素级别的说话人嵌入 $\boldsymbol{E}$ 会与编码器隐层 $\boldsymbol{H}$ 进行维度拼接. 而在后续实验中，为了保证实验的可比性，句子级别的说话人嵌入和帧级别的说话人嵌入

都会经过相应的拓展与编码器隐层 $\boldsymbol{H}$ 或者时长拓展后的隐层 $\boldsymbol{H}'$ 拼接后进入解码器参与训练和测试.

关于本文采用的音素级别的说话人嵌入的使用,与音素片段的拼接语音合成模型有些相似,与传统的拼接合成模型的区别在于本文方法将片段的挑选、拼接、平滑都整合为使用神经网络操作.需要注意的是,和拼接合成不同的是,文本信息在整个过程中除了作为片段挑选的依据,该信息还会作为音素片段均值的补充信息,提供包括音调变化等局部的变化信息,帮助模型合成语义清晰的声音.去除文本的消融实验可见 5.4 节.

本文采用的参考编码器并没有使用复杂的模型结构,其核心结构操作是基于时长切分的时间平均.这种做法也启发于说话人表征技术[13,31],它可以去除局部变化的信息,留下局部稳定的信息,即音素级别的音色、口音和部分语义信息.在小数据说话人复刻的任务出发,我们认为学习口音或者发音习惯也可以很好提高听感上的相似度,因此并没有对音素级别说话人嵌入中的各类信息进行解耦合,或者是去除口音信息.

3.4 说话人嵌入预测网络

在测试时,集外文本一般没有文本对应的参考音频作为参考编码器的输入,来帮助模型合成音频.为了解决这个的问题,本文提出了一个从音素序列 $\boldsymbol{t}$ 到音素级别说话人嵌入 $\boldsymbol{E}$ 的说话人嵌入预测网络,来拟合这种上下文相关的音素级别说话人嵌入分布,如图 3(a)所示.

说话人嵌入预测网络输入音素序列 $\boldsymbol{t}$,使用与编码器独立的音素嵌入,经过 3 层卷积神经网络得到说话人嵌入预测值 $\hat{\boldsymbol{E}}=(\hat{\boldsymbol{e}}_1,\hat{\boldsymbol{e}}_2,\cdots,\hat{\boldsymbol{e}}_n)$,具体的网络参数细节可以参考网络配置表 1.

我们尝试过对测试数据的真实特征片段进行简单操作以达到代替参考音频的效果,但是效果都不够理想.方法一,根据输入的文本 $\boldsymbol{t}$,从测试数据中随机选择音素(假设不存在音素不覆盖问题)的对应特征片段组成"伪"参考音频的特征序列 $\boldsymbol{S}_{rand}=(\boldsymbol{s}_{rand}^{(t_1)},\cdots,\boldsymbol{s}_{rand}^{(t_n)})$.这种方法没有考虑上下文信息的特征片段,非常容易在合成音频的音素连接处出现音调跳变,合成效果很不自然;方法二,使用测试数据中所有的当前音素片段拼接组成"伪"参考音频的特征序列 $\boldsymbol{S}_{all}=(\boldsymbol{s}_{all}^{(t_1)},\cdots,\boldsymbol{s}_{all}^{(t_n)})$.这种做法可以缓解音调跳变问题,但是大量的特征片段平均后过于平滑,从而导致合成音频相似度有明显的下降.

因此,本文提出的说话人嵌入预测网络将"挑选"特征片段的问题交由神经网络解决,可以有效缓解使用随机真实特征片段造成的音调跳变的不稳定性,同时也保留了上下文相关的信息,不会过于平滑而丢失了太多说话人音色.

3.4.1 混合高斯密度网络

我们使用混合高斯密度网络来建模说话人嵌入的输出分布.参考文献[43],将说话人嵌入预测网络的最后一层隐层映射成混合高斯的三个分布参数 $\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\sigma}$,其输出分布可以写为

$$p(\boldsymbol{y}|\boldsymbol{x},\mathcal{M})=\sum_{k=1}^{K}\boldsymbol{\omega}_k(\boldsymbol{x})\mathcal{N}(\boldsymbol{y};\boldsymbol{\mu}_k(\boldsymbol{x}),\boldsymbol{\sigma}_k^2(\boldsymbol{x})),$$

$$\boldsymbol{\omega}_k(\boldsymbol{x})=\frac{\exp(z_k^{(\omega)}(\boldsymbol{x},\mathcal{M}))}{\sum_{l=1}^{K}\exp(z_l^{(\omega)}(\boldsymbol{x},\mathcal{M}))},$$

$$\boldsymbol{\sigma}_k(\boldsymbol{x})=\exp(z_k^{(\sigma)}(\boldsymbol{x},\mathcal{M})),$$

$$\boldsymbol{\mu}_k(\boldsymbol{x})=z_k^{(\mu)}(\boldsymbol{x},\mathcal{M}),$$

其中 K 混合高斯中的高斯数目,$\boldsymbol{x}$ 是混合高斯密度网络的输入,$\boldsymbol{y}$ 是网络的输出变量,z 表示激活函数,$\mathcal{M}$ 表示模型参数.在训练时,我们直接将优化对数似然值 $\log p(\boldsymbol{y}|\boldsymbol{x},\mathcal{M})$.在合成的时候,我们可以对混合高斯分布进行采样,但是为了保证合成声音的稳定性,本文仅对多个头的高斯均值进行了权重平均.混合高斯的数目参考了文献[44],同样验证了混合高斯数目 $K=2$ 时最适合训练与自适应.

3.4.2 说话人嵌入预测网络训练

为了减小自适应训练的时间开销,提高自适应的鲁棒性,会在训练语音合成主模型的同时,训练说话人嵌入预测网络.多说话数据上的说话人嵌入预测网络会使用一个全局的可训练说话人查找表编码(Look-Up Table Encoding),并拼接到说话人嵌入预测网络每一层卷积的输入上.在训练阶段会根据说话人编号,训练一组全局的查找表.在自适应阶段,会固定选择查找表中的一个编码值进行更新.

在训练数据上完成说话人嵌入预测网络的预训练之后,在自适应阶段就可以使用少量的目标说话人数据对说话人嵌入预测网络进行快速的自适应训练.在实现中,说话人嵌入预测的损失误差的梯度将会在反传至参考编码器前被截断,防止影响参考编码器和声学模型的训练.

本小节最后,对音素级别的说话人嵌入对测试说话人的泛化鲁棒性进行说明.尽管目标说话人的本文覆盖程度可能非常有限,但是音素级别的说话人嵌入只提供局部的平均音色,语音中的音调与部分语义仍然受编码器端的文本信息控制.并且本文

在多说话人的训练数据上对说话人嵌入预测网络进行了充足的预训练，对于在目标说话人数据中没有出现过的文本，仍能合成鲁棒且语义清晰的声音.

3.5　训练和测试流程

整个模型训练的损失值$\mathcal{L}$包括：语音合成模型的声学特征重构误差$\mathcal{L}_{recon}$、时长模型的预测误差$\mathcal{L}_{dur}$、说话人嵌入网络的预测误差$\mathcal{L}_{emb}$：

$$\mathcal{L}_{recon}=\|\boldsymbol{O}-\hat{\boldsymbol{O}}\|_2+\|\boldsymbol{O}-\hat{\boldsymbol{O}}_{post}\|_2,$$

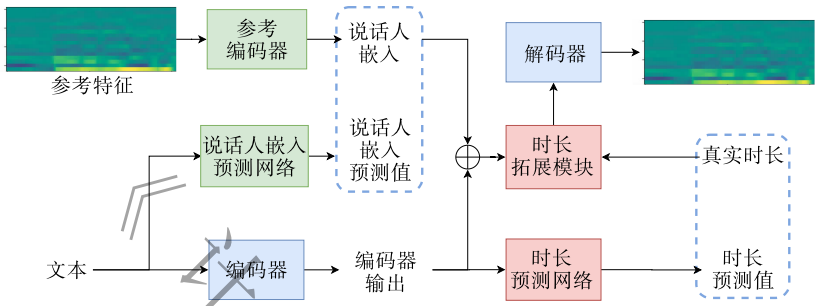
$$\mathcal{L}_{dur}=\|\boldsymbol{d}-\hat{\boldsymbol{d}}\|_2,$$

$$\mathcal{L}_{emb}=\log p(\boldsymbol{O}|\boldsymbol{t},\mathcal{M}),$$

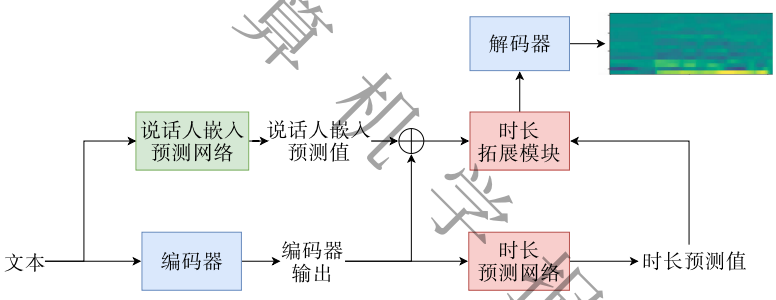
$$\mathcal{L}=\mathcal{L}_{recon}+\lambda_{dur}\times\mathcal{L}_{dur}+\lambda_{emb}\times\mathcal{L}_{emb},$$

其中λ_{dur}和λ_{emb}是对应损失值的权重参数.

整个模型的训练、测试如图 3(a)、(b)所示. 在训练阶段需要输入文本和参考特征，参考编码器将通过真实时长提取音素级别的说话人嵌入，和编码器的输出维度拼接后经过时长拓展，通过解码器输出对应的目标声学特征，并且完成说话人嵌入预测网络和时长预测网络的预训练. 在自适应训练阶段，仅计算$\mathcal{L}_{emb}$来更新说话人嵌入预测网络，而无需模型的其他部分.



(a) 模型训练阶段结构示意图



(b) 模型测试阶段结构示意图

图 3　图为本文提出的音素级别的说话人嵌入的语音合成自适应方法的训练和测试阶段的示意图(蓝色虚线框表示需要计算的损失值$\mathcal{L}_{emb}$和$\mathcal{L}_{dur}$. 在训练中，模型使用真实时长和真实参考特征提取音素级别的说话人嵌入，训练网络参数. 在测试阶段会通过自适应得到的说话人嵌入预测网络和时长预测网络进行合成)

在测试阶段，只需要输入文本到编码器网络和说话人嵌入预测网络，使用预测时长进行时长拓展后，通过解码器预测得到目标说话人的声学特征.

模型的整个数据的预处理、训练、自适应和测试的流程整理如算法 1 所示.

算法 1.　基于音素级别的说话人嵌入语音合成自适应方法.

输入：多说话人训练数据集$\mathcal{D}_1=\langle\boldsymbol{A},\boldsymbol{t}\rangle,\mathcal{D}_2,\cdots,\mathcal{D}_s$，自适应目标说话人数据集$\mathcal{D}'$，集外测试文本$\hat{\boldsymbol{t}}$

输出：自适应目标说话人的合成音频信号$\hat{\boldsymbol{A}}$

1. 预处理阶段：

1.1. 从音频信号$\boldsymbol{A}$中提取声学特征$\boldsymbol{O}$，得到训练数据集用$\mathcal{T}=(\boldsymbol{O},\boldsymbol{t})$和自适应数据集$\mathcal{T}'=(\boldsymbol{O}',\boldsymbol{t}')$.

1.2. 使用训练数据集$\mathcal{T}$训练基于 HMM 的 ASR 模型，从对齐信息中提取时长$\boldsymbol{D}$并对$\boldsymbol{O}$进行切分得到音素特征片段$\boldsymbol{S}$.

2. 训练阶段：

2.1. 使用真实的音素特征片段$\boldsymbol{S}$和时长$\boldsymbol{D}$，计算$\mathcal{L}$训练音素级别的说话人嵌入控制的端到端语音合成模型.

2.2. 使用真实的音素特征片段$\boldsymbol{S}$和时长$\boldsymbol{D}$，计算$\mathcal{L}_{dur}$和$\mathcal{L}_{emb}$训练时长预测网络和说换人嵌入预测网络.

3. 自适应阶段：使用自适应数据集$\mathcal{T}'$，计算$\mathcal{L}_{emb}$自适应训练说话人嵌入预测网络.

4. 测试阶段：输入集外测试文本$\hat{\boldsymbol{t}}$，合成声学特征$\hat{\boldsymbol{O}}$，再经过预训练的声码器得到音频输出$\hat{\boldsymbol{A}}$.

4　实验设置

本节将介绍实验的所用到的数据、特征处理、模

型参数、训练细节以及最后模型性能的评价方法.

4.1 实验数据

在本实验中,我们收集了 24 位男性和 52 位女性说话人的中文音频作为训练数据,男性合计 61 个小时,女性合计 141 个小时.在测试阶段,我们收集了 20 位用户(10 位男性,10 位女性)录制的真实数据,各 40 句话(合计 3～4 min),这些数据和真正的训练数据在音质和标注的准确度上有较明显的差距:用户录制的音频有环境混响、噪音爆音、错字漏字等问题.

为了进一步说明训练数据和测试数据在语速上区别,我们对数据中的音素时长进行了统计,分别计算了说话人的声韵母时长的均值和标准差,并以二维散列图的形式表示在图 4 中.比较图 4 中训练集和测试集的韵母点,可以发现测试集说话人的韵母发音时长均值与训练集分布类似.比较图 4 中训练集和测试集的声母点,训练集中的说话人的时长均值集中在 6 帧之前,而测试集的声母均值分布非常分散.由此可见,训练集的说话人语速主要取决于韵母,而测试集说话人语速将同时取决于声韵母.

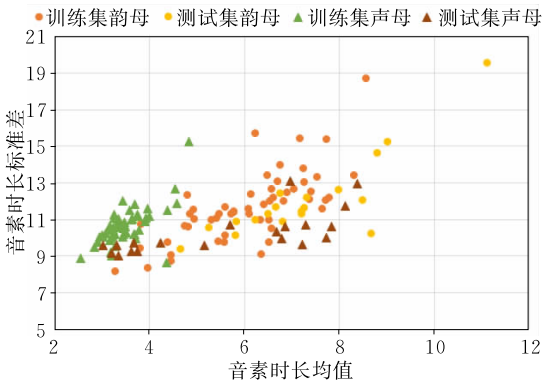


图 4 训练集和测试集的音素时长统计量散列图

测试数据上需要先进行预处理操作,包括使用 rnnoise[45] 方法对音频进行降噪、对提供给用户录制的朗读稿文本进行文本归一化和音素转换.在时长提取上,我们对 kaldi[46] 工具中的 aishell[47] 脚本进行了修改,替换语音识别的音素集合为语音合成音素集合,从而免去了不同音素集合之间映射的麻烦.同时,我们强制转换每个句子为一个特殊的长词,使得 kaldi 的强制对齐不会在文本中插入额外的停顿标识.kaldi 脚本中的 HMM 模型训练仅使用了上述训练数据,不包括测试数据.如果 kaldi 脚本在强制对齐的过程中,beam 大小超过 20 的音频数据会被丢弃,间接筛选了文本错误过多的数据.

4.2 LPCNet 声码器

LPCNet[35] 声码器的声学特征包括如下 20 维:18 维 bark 尺度的倒谱系数和 2 维基频参数(周期和相关性参数).其的特征提取参数和原始论文保持一致,即 16k 采样率、20 ms 的窗口大小、10 ms 的帧偏移、8 bits 量化和 $\alpha=0.85$ 预加重系数.

LPCNet 声码器的训练数据即端到端语音合成模型的训练数据,而在训练多说话人的 LPCNet 模型时,采用在采样率网络中引入基于查找表的说话人嵌入的方法.对于训练集外的说话人,将选择一个默认的说话人编号对应的说话人嵌入进行合成.这种折中的做法并不会非常影响最终合成音频的自然度和相似度,因此在测试阶段并没有对声码器进行额外的说话人自适应训练.

4.3 训练细节

为了平衡模型的性能和效率,本节实验采用了如下超参数:解码器的折叠步数为 3,batch 大小为 32,初始学习率为 1e-3,使用 noam 学习率衰减策略,Adam 优化器的参数为 $\beta_1=0.9$,$\beta_2=0.999$,$\epsilon=$ 1e-6.损失误差中的权重参数设置为 $\lambda_{dur}=1$,$\lambda_{emb}=0.01$.为了加快训练速度,实际实验中使用了 4 张 2080ti 显卡进行并行训练.训练时间为 2 天左右,总计大约 200 个 epoch,100k 步左右完全收敛.

在自适应阶段,降低 batch 大小到 8,同时固定学习率为 1e-4,训练 100 个 epoch,约计 500 步.自适应训练时间开销将在后文 5.3 节中详细讨论.

网络的参数细节描述如表 1 所示.其中 Conv 1d-a-b 表示一维卷积神经网络,核大小为 a,通道数据为 b;BN 表示 Batch Normalize;BLSTM 表示双向 LSTM;FC 表示一层全连接网络;2GMM 表示混合高斯数目为 2 的混合高斯密度输出网络.

表 1 模型超参数细节

编码器		Embedding 512 Conv 1d-5-512-BN-ReLU×3 1 layer BLSTM, 256 cells each directions
解码器	Prenet	FC-256-ReLU-Dropout(0.5)×2
	LSTMs	FC-512 2 layer LSTM,512 cells, decode steps 3
	Postnet	Conv1d-5-512-BN-ReLU×5 Conv1d-5-20, residual connection
参考编码器		FC-64 1 layer GRU, 64 cells
时长预测网络		1 layer BLSTM, 16 cells each directions FC-1
说话人嵌入预测网络		Embedding 256, SpeakerEmbedding 64 Conv1d-5-256-BN-Tanh-Dropout(0.5)×3 FC-2GMM

4.4 模型评价

4.4.1 模型设置

本节选择了 3 个不同粒度不同实现的说话人嵌入的方法进行对比实验，声学模型都选择上述的基于时长的端到端语音合成模型框架. 实验涉及到的若干系统名称缩写解释如下①：

(1) OracleVocode. 声码器反合成原始特征.

(2) Xvec[7]. 句子级别的说话人嵌入，使用基于预训练的说话人网络提取的 x-vector 提取说话人嵌入. 在训练时，使用对应句子的 x-vector；在测试时，使用说话人平均的 x-vector 做为输入.

(3) UttEmb[15]. 句子级别的说话人嵌入，使用联合训练的参考编码器网络提取说话人嵌入. 由于与原论文使用的声学特征不同，我们将 2 维卷积神经网络替换为了 1 维卷积网络. 在训练时，使用对应句子的声学特征输入参考编码器；在测试时使用随机挑选的真实特征序列作为参考编码器的输入.

(4) Attentron[17]. 帧级别的说话人嵌入，使用基于注意力机制提取音色信息的联合训练参考编码器提取说话人嵌入. 为了防止过拟合，在训练时概率使用对应句子的声学特征输入参考编码器；在测试时，使用随机挑选的多句真实特征序列的拼接作为参考编码器的输入.

(5) PhnEmb. 本文提出的音素级别的说话人嵌入，采用基于显式时长切分提取音色信息所联合训练得参考编码器. 在训练时，使用真实特征片段输入参考编码器；在测试阶段，使用自适应后的说话人嵌入预测网络.

(6) ＋Adapt. 在更新说话人嵌入的基础之上，同时更新解码器网络中的 LSTM 的网络参数进行自适应训练.

4.4.2 评价指标

语音合成领域主要使用主观打分进行合成系统的评价，同时也使用一些客观指标进行辅助分析，其中包括：

(1) 平均意见分数(Mean Opinion Score，MOS)是语音合成中最常用的主观评价标准，可以对合成的音频分别进行自然度和相似度的评价. MOS 一般要求评测者对音频进行 1～5 分的打分. 自然度 MOS 要求评测者对音频的自然程度进行打分；相似度 MOS，则会给出一个参考音频，要求评测者对合成音频与参考音频的相似程度进行打分.

(2) 梅尔倒谱失真(Mel Ceptrum Distortion，MCD)是评估合成声音质量的一种客观方法，计算了倒谱序列之间的数值差异. 为了减少引入时长和动态时间扭曲(Dynamic Time Wraping)带来的误差，在对集内文本进行指标计算时使用真实时长以保证和真实特征长度一致.

(3) x-vector 余弦相似度(x-vector cosine similarity) x-vector[31]是基于神经网络提取的说话人嵌入，通过计算合成音频和真实音频在 x-vector 空间的余弦距离，近似地比较合成声音和真实声音的相似程度. 其计算方法是，给定两个说话人嵌入向量 $\boldsymbol{x}$，$\boldsymbol{y}$，

$$sim = 1 - \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{|\boldsymbol{x}| \cdot |\boldsymbol{y}|},$$

实现中，余弦相似度的计算采用了 Resem-blyer② 计算得到的嵌入值进行打分.

5 实验结果

本节将对上述系统进行不同方面的实验分析，包括未见过说话人的音色提取能力比较、集外文本的合成音频的自然度和相似度对比、自适应训练的耗时和收敛情况的分析，以及部分消融实验的结果和分析.

5.1 测试说话人集内文本评价结果

首先，为了验证音素级别的说话人嵌入方法的提取未见过的说话人音色的能力，我们先使用了测试说话人的真实数据作为参考音频，合成和参考音频内容相同的音频进行评测. 其中为了区别在音素级别的说话人嵌入方法中使用参考编码器和说话人嵌入预测网络进行合成的两种方法，我们将使用参考编码器的系统命名为 PhnEmb＋real ref.

为排除声码器在合成过程中引入的误差，对真实特征经过声码器反合成的音频将作为评价的目标音频. 客观指标 MCD 和 x-vector 余弦相似度的计算使用全部 20 个测试说话人总计约 800 句话，统计结果计算了 95% 置信度的区间范围，结果如表 2 所示.

表 2　测试说话人集内文本的指标结果(＋real ref 表示使用了真实的特征片段作为参考编码器的输入)

系统	MCD↓	x-vector 余弦相似度↑
Xvec	9.880±0.337	0.807±0.012
UttEmb	9.495±0.338	0.795±0.012
Attentron	8.726±0.322	0.844±0.009
PhnEmb＋real ref	6.088±0.243	0.890±0.007
PhnEmb	7.937±0.311	0.854±0.009

① 文献[12]提出的音素相关的说话人嵌入方法因为需要对每个说话人建立音素级别的查找表，在我们的较多的训练说话人下的开销过大，因此并没有选作基线系统.

② https://github.com/resemble-ai/Resemblyzer

从 MCD 和 x-vector 余弦相似度可以看出，对于基于句子级别的说话人嵌入系统，Xvec 和 UttEmb 系统复原原始音频特征的能力相对有限，帧级别的说话人嵌入 Attentron 和音素级别的说话人嵌入 PhnEmb＋real ref 都比这两种句子级别的说话人嵌入方法有细致的音色提取能力，能在从未见过的说话人数据上合成出更接近原始说话人的音频.

比较我们提出的音素级别的说话人嵌入系统 PhnEmb＋real ref 和基于注意力机制计算的帧级别说话人嵌入系统 Attentron 可以发现，帧级别的说话人嵌入方法在说话人嵌入的粒度上更具优势，但是从实验结果看到其客观指标均差于音素级别的说话人嵌入系统 PhnEmb＋real ref，甚至差于不使用真实参考特征而使用说话人嵌入预测网络的系统 PhnEmb.

检查 Attentron 模型对参考音频的对齐情况发现，尽管输入了真实的参考音频，但是其对齐情况并不理想. 其对齐图大致呈现对角线但是局部出现镂空的情况. 而本文提出的音素级别说话人嵌入使用了独立的时长模型，能较为准确地提取每一个音素的说话人信息，这可能解释了 PhnEmb＋real ref 和 PhnEmb 系统在该实验上的性能优势.

5.2 测试说话人集外文本评价结果

本实验选择测试说话人中的 6 位说话人（3 名男性、3 名女性）对上述系统分别生成了 48 句话进行评测. 最终我们一共收集到了 34 位母语中文的志愿者对自然度和相似度的打分，结果如表 3 所示. 我们也给出了部分合成的测试音频用于展示效果①.

表 3 测试说话人集外文本的指标结果

系统	自然度-MOS↑	相似度-MOS↑
OracleVocode	3.92±0.22	—
Xvec	3.95±0.18	2.38±0.22
UttEmb	**4.03**±0.18	1.85±0.15
Attentron	3.44±0.19	2.41±0.18
PhnEmb	3.91±0.18	**3.36**±0.22
Xvec＋Adapt	3.10±0.19	3.37±0.21
UttEmb＋Adapt	3.72±0.18	3.34±0.21
Attentron＋Adapt	2.77±0.18	3.15±0.22
PhnEmb＋Adapt	**4.13**±0.15	**3.80**±0.19

在不更新语音合成模型参数的系统中，比较三个基线系统和本文提出的音素级别的说话人嵌入自适应方法可见，本文提出的方法有明显的相似度提升，并且自然度没有明显下降. 其中句子级别的说话人嵌入系统 UttEmb 和 Xvec 尽管有相对较高的自然度，但是其相似度有明显的劣势.

在更新语音合成模型参数的系统中，三个基线系统因为更新了解码器中 LSTM 的网络参数，在相似度上得到了提升，但是自然度都有了不同程度的下降. 而我们提出的音素级别的说话人嵌入依然保持着最好的相似度的同时，其自然度也在各个系统中达到最好的水平.

其中帧级别的说话人嵌入系统 Attentron 没有达到比较理想的效果. 其在测试说话人的集外文本上的对齐并不清晰，呈现零星的散点的情况，因此其相似度并没有额外的优势，且合成质量可能也受到了不稳定的说话人嵌入的影响而有所下降.

5.3 自适应训练时间分析

在实际的说话人自适应的产品使用当中，除了关心合成声音的自然度和相似度，同时也关注对于一组新数据，其自适应训练的时间开销，以及自适应训练的收敛速度. 本小姐分别对如下 5 组系统进行了自适应训练，统计其时间开销如表 4 所示.

表 4 每个 epoch 的训练时间开销统计平均值

系统	平均每 epoch 训练时间/s
PhnEmb	0.12
Xvec＋Adapt	3.78
UttEmb＋Adapt	3.82
Attentron＋Adapt	4.62
PhnEmb＋Adapt	3.91

相比较需要更新合成模型参数的方法，对说话人嵌入预测网络自适应训练的时间开销非常小. 这得益于自适应训练的网络仅为说话人嵌入预测网络，损失计算和梯度传递不需要经过自回归的合成模型. 同时说话人嵌入预测网络使用的是可以底层加速的卷积网络，因而也可以进一步地加快训练速度. 考虑到例如 Xvec 等系统其实也需要一些特征提取的时间，音素级别的说话人嵌入自适应方法的自适应时间其实可以近似忽略.

考虑更新合成模型参数的方法，从平均每个 epoch 的训练时间开销看出，音素级别的说话人嵌入和句子级别的说话人嵌入的时间开销差距并不大，而帧级别的说话人嵌入系统 Attentron 需要对比较长的参考音频逐帧计算注意力机制，他的时间开销就明显增加.

对后四个更新合成模型参数的系统在所有 20 个测试说话人上的自适应训练情况进行进一步分析，对其最终重构误差 $\|\boldsymbol{O}-\hat{\boldsymbol{O}}_{post}\|_2$ 在每个 epoch 上进行

① https://dazenhom.github.io/htmls/phn-emb.html

了统计平均，可以得到自适应训练过程中，平均重构误差随 epoch 数的变化趋势图 5.

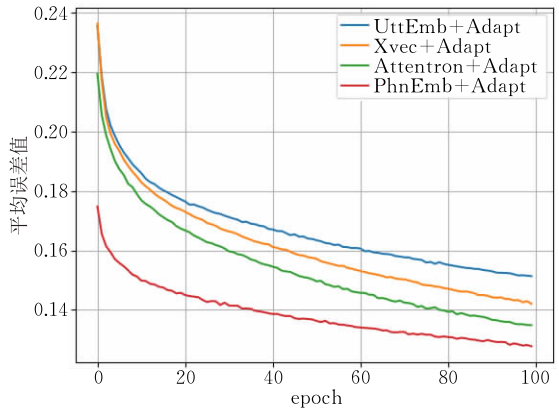


图 5　自适应训练过程中 $\|\boldsymbol{O}-\hat{\boldsymbol{O}}_{post}\|_2$ 平均误差值随 epoch 数变化趋势图

可以看出句子级别的说话人嵌入方法 UttEmb 和 Xvec 的起始误差最大，帧级别说话人嵌入方法 Attentron 的起始误差稍显更低，而我们提出的音素级别的说话人嵌入方法的起始误差明显最低. 这个优势使得在需要更新语音合成模型参数时，训练相同的 epoch 数的停止标准下能更快收敛，在需要收敛到一定重构误差值的停止标准下时间开销更少.

同时这也部分解释了音素级别的说话人嵌入在小数据说话人自适应中的优势. 它提供了一个显著更好的模型初始点，能够更快地帮助模型找到正确的梯度方向，完成自适应训练. 同时音素级别的说话人嵌入提供的非音色信息可能也缓解了不准确的数据标注的问题，因此可以得到更好的自然度.

5.4　模型消融实验

本节对本文中使用的三个实验设置：文本嵌入、独立的时长模型和混合高斯密度网络进行消融实验，以验证上述三种设置对最后模型性能的影响，其系统符号表示为：

(1) PhnEmb-text. 去除文本嵌入和编码器网络，解码器仅输入音素级别说话人嵌入.

(2) PhnEmbAtt. 替换独立的说话人无关时长模型为原始 Tacotron 所使用的注意力机制，此处采用了更为稳定的 DCA 注意力机制[48].

(3) PhnEmbL2. 使用 L2 损失函数代替混合高斯密度网络的对数自然值训练说话人嵌入预测网络.

实验采用 MCD 和 x-vector 余弦相似度作为评价标准，在测试说话人的集内数据进行测试，结果如表 5 所示. 对于前两种设置仅考虑模型从真实音频特征片段中提取说话人信息并恢复音频特征的能力，因此使用真实音频片段作参考音频. 而第三种设置需要考虑不同的预测网络对说话人嵌入的拟合效果，故使用了说话人嵌入预测网络进行合成比较.

表 5　测试说话人集内测试数据的指标结果
(real ref 表示使用真实特征片段进行合成)

系统	MCD ↓	x-vector 余弦相似度 ↑
PhnEmb+real ref	6.088±0.243	0.890±0.007
PhnEmb-text+real ref	6.621±0.276	0.872±0.012
PhnEmbAtt+real ref	6.001±0.185	0.879±0.007
PhnEmb	7.937±0.311	0.854±0.009
PhnEmbL2	8.176±0.337	0.835±0.009

比较系统 PheEmb+real ref 和 PhnEmb-text+real ref 可以看到，没有文本嵌入的输入，声音相似度并没有很明显的下降，但是其 MCD 有一个明显的上升. 从合成的音频听感上发现，音频语义的清晰程度受到了较为严重的影响，会出现发音错误、吞音等问题，而这个问题在使用预测得到的说话人嵌入上变得更加严重. 因此仅仅输入音素级别的说话人嵌入是不够的，文本端的输入依然是必需的. 从信息提取的角度设想，音素级别的说话人嵌入给定了当前说话人当前音素发音的平均音色，而文本端提供了音调、发音规律等信息，将平均音色正确地扩充为一个时间上连续平滑的音素特征片段，保证合成声音的语义清晰正确.

比较系统 PhnEmb+real ref 和 PhnEmbAtt+real ref 的 MCD 可以看到，使用分离的时长模型的性能相较使用基于注意力机制的模型并没有明显的下降，这个结论和文献[18,24]中是类似. 独立的时长模型所带来的好处是可以保证测试文本的时长可控的同时，不会出现因为注意力机制所导致的对齐崩溃问题[18,24]. 这在我们合成从未见过的测试说话人的声音时是非常有好处的，系统就不会由于不稳定的说话人嵌入导合成崩溃，极大提升了系统的鲁棒性.

比较系统 PhnEmb 和 PhnEmbL2 的 MCD 可以看到，在经过自适应训练之后，使用混合高斯作为输出分布有更好的数据拟合. 同时我们类似地观察了其损失值的变化曲线发现，混合高斯密度也提供了更好的初始点，可以帮助预训练的模型更快地训练收敛.

5.5　说话人相关的时长模型对比实验

本节对提出的说话人无关时长模型和说话人相关时长模，在测试说话人上的合成效果进行对比实验和分析. 为了防止可能出现的时长模型和声学模

型不匹配问题，本节选择更新声学模型的系统作为实验的对照组，符号标记如下所示：

(1) PhnEmb+Adapt(PA). 使用说话人无关时长模型，更新解码器中的 LSTM，细节前文已述.

(2) PhnEmb+Adapt+DurAdapt(PAD). 使用说话人相关时长模型，在 PhnEmb+Adapt 系统基础上，在自适应阶段同时更新时长预测网络.

为了验证说话人语速对两种时长模型合成声音产生的影响，根据图 4 所示的统计结果，分别选择了三组不同语速的说话人(快、中等、慢)，每个性别各 2 位(男女各 1 位)测试说话人. 为了突出语速的影响，本实验选择了快组和慢组中最极端的说话人进行实验. 评价指标为自然度偏好测试，需要听测者比较同样文本内容的两句音频，并选择自然度更好的一句. 实验对共 30 组对比音频进行听音测试，其实验结果统计如表 6 所示.

表 6 时长模型的偏好测试结果 /%

语速类型	偏好 PA	偏好 PAD	无偏好
快语速	60	17	23
中等语速	80	3	17
慢语速	97	0	3

从表 6 可以看出，使用说话人相关的时长模型多少都会带来一定程度的自然度损失. 当语速较慢时，更新时长模型带来的自然度下降非常明显；当说话人的语速较快时，对自然度的影响相对变小.

该结果也符合对测试数据的观察和表 4 的统计结果. 语速慢的说话人的数据中存在着大量的不自然的停顿和拖音，这些停顿可能在强制对齐时被切入了声母部分，从而导致了表 4 中出现的测试说话人的声母时长的不自然的散列分布.

综上所述，使用并不准确的时长更新时长模型，会导致性能下降，这也是本文倾向于使用说话人无关的时长模型的初衷. 而说话人无关的时长模型，也可以作为缓解测试数据中的标注错误的一种方案被使用.

6 结 论

本文提出了一种基于音素级别的说话人嵌入的语音合成自适应方法，通过从真实的参考特征中提取音素级别的说话人嵌入，控制模型生成自然且相似的声音. 在自适应时，我们通过对预训练的说话人嵌入预测网络进行自适应训练，学习从文本上下文到对应音素级别的说话人嵌入的映射，代替真实特征片段用于集外文本的推理.

本文使用真实用户录制的少量数据进行自适应训练. 实验表明，相比较如今的各种不同的说话人嵌入方法，本文提出的方法在不更新主要网络参数的情况下保持自然度的同时，得到了最好相似度且自然度并没有明显下降. 与此同时，在更新声学模型的情况下，该方法达到了最好的自然度和相似度. 分析发现音素级别的说话人嵌入在不显著增加自适应训练时间的情况下，提供了更好的模型自适应初始点，有效地提高了集外说话人的合成声音质量.

该研究的未来工作包括：(1) 对本文提出的音素级别的说话人嵌入进行解耦合，对目标说话人的音色、口音、韵律等进行细致的控制；(2) 将音素级别的说话人嵌入方法应用到其他语音合成框架上，如参数 LSTM、FastSpeech 等模型上；(3) 本文使用了说话人无关的时长模型，如何更好地利用测试说话人的时长信息将需要更进一步研究.

致 谢 本文主要研究内容为徐志航在苏州思必驰信息科技有限公司实习期间完成. 感谢思必驰公司提供的基础设施支持和宝贵的技术讨论. 特别感谢马娆、李翰正、曾海峰等同事在工作期间给出的宝贵意见和技术支持！

参 考 文 献

[1] Wang Y, Skerry-Ryan R, Stanton D, et al. Tacotron: Towards end-to-end speech synthesis//Proceedings of the 18th Annual Conference of the International Speech Communication Association. Stockholm, Sweden, 2017: 4006-4010

[2] Gibiansky A, Arik S, Diamos G, et al. Deep voice 2: Multi-speaker neural text-to-speech//Advances in Neural Information Processing Systems. Long Beach, USA, 2017: 2962-2970

[3] Taigman Y, Wolf L, Polyak A, et al. VoiceLoop: Voice fitting and synthesis via a phonological loop. arXiv preprint arXiv:1707.06588, 2017

[4] Oord A V D, Dieleman S, Zen H, et al. WaveNet: A generative model for raw audio//Proceedings of the 9th ISCA Speech Synthesis Workshop. Sunnyvale, USA, 2016: 125

[5] Chung Y A, Wang Y, Hsu W N, et al. Semi-supervised training for improving data efficiency in end-to-end speech synthesis//Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019). Brighton, UK, 2019: 6940-6944

[6] Fong J, Gallegos P O, Hodari Z, et al. Investigating the robustness of sequence-to-sequence text-to-speech models to

imperfectly-transcribed training data//Proceedings of the 20th Annual Conference of the International Speech Communication Association. Graz, Austria, 2019: 1546-1550

[7] Jia Y, Zhang Y, Weiss R, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis//Advances in Neural Information Processing Systems. Montréal, Canada, 2018: 4480-4490

[8] Chen Y, Assael Y, Shillingford B, et al. Sample efficient adaptive text-to-speech. arXiv preprint arXiv: 1809.10460, 2018

[9] Kalchbrenner N, Elsen E, Simonyan K, et al. Efficient neural audio synthesis//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 2415-2424

[10] Nachmani E, Polyak A, Taigman Y, et al. Fitting new speakers based on a short untranscribed sample//Proceedings of the 35th International Conference on Machine Learning. Stockholm, Sweden, 2018: 3680-3688

[11] Zhao Y, Saito D, Minematsu N. Speaker representations for speaker adaptation in multiple speakers BLSTM-RNN-based speech synthesis//Proceedings of the 18th Annual Conference of the International Speech Communication Association (Interspeech). San Francisco, USA, 2016: 2268-2272

[12] Fu R, Tao J, Wen Z, et al. Phoneme dependent speaker embedding and model factorization for multi-speaker speech synthesis and adaptation//Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019). Brighton, UK, 2019: 6930-6934

[13] Doddipatla R, Braunschweiler N, Maia R. Speaker adaptation in DNN-based speech synthesis using d-vectors//Proceedings of the 18th Annual Conference of the International Speech Communication Association (Interspeech). Stockholm, Sweden, 2017: 3404-3408

[14] Bollepalli B, Juvela L, Alku P. Speaking style adaptation in text-to-speech synthesis using sequence-to-sequence models with attention. arXiv preprint arXiv: 1810.12051, 2018

[15] Arik S, Chen J, Peng K, et al. Neural voice cloning with a few samples//Advances in Neural Information Processing Systems. Montréal, Canada, 2018: 10019-10029

[16] Cooper E, Lai C I, Yasuda Y, et al. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings //Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020). Barcelona, Spain, 2020: 6184-6188

[17] Choi S, Han S, Kim D, et al. Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding//Proceedings of the 21st Annual Conference of the International Speech Communication Association. Shanghai, China, 2020: 2007-2011

[18] Yu C, Lu H, Hu N, et al. Durian: Duration informed attention network for multimodal synthesis. arXiv preprint arXiv: 1909.01700, 2019

[19] Ren Y, Hu C, Tan X, et al. FastSpeech 2: Fast and high-quality end-to-end text to speech. arXiv preprint arXiv: 2006.04558, 2020

[20] Oord A, Li Y, Babuschkin I, et al. Parallel WaveNet: Fast high-fidelity speech synthesis//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 3918-3926

[21] Shen J, Pang R, Weiss R J, et al. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions//Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, Canada, 2018: 4779-4783

[22] Ping W, Peng K, Gibiansky A, et al. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. arXiv preprint arXiv: 1710.07654, 2017

[23] Li N, Liu S, Liu Y, et al. Neural speech synthesis with transformer network//Proceedings of the AAAI Conference on Artificial Intelligence: Volume 33. Honolulu, USA, 2019: 6706-6713

[24] Ren Y, Ruan Y, Tan X, et al. FastSpeech: Fast, robust and controllable text to speech//Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 3171-3180

[25] Mehri S, Kumar K, Gulrajani I, et al. SampleRNN: An unconditional end-to-end neural audio generation model. arXiv preprint arXiv: 1612.07837, 2016

[26] Valin J M, Skoglund J. LPCNET: Improving neural speech synthesis through linear prediction//Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019). Brighton, UK, 2019: 5891-5895

[27] Ping W, Peng K, Chen J. ClariNet: Parallel wave generation in end-to-end text-to-speech. arXiv preprint arXiv: 1807.07281, 2018

[28] Prenger R, Valle R, Catanzaro B. WaveGlow: A flow-based generative network for speech synthesis//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019). Brighton, UK, 2019: 3617-3621

[29] Cooper E, Lai C I, Yasuda Y, et al. Can speaker augmentation improve multi-speaker end-to-end TTS?//Proceedings of the 21st Annual Conference of the International Speech Communication Association. Shanghai, China, 2020: 3979-3983

[30] Garcia-Romero D, Espy-Wilson C Y. Analysis of i-vector length normalization in speaker recognition systems//Proceedings of the 12th Annual Conference of the International Speech Communication Association. Florence, Italy, 2011: 249-252

[31] Snyder D, Garcia-Romero D, Sell G, et al. X-vectors: Robust DNN embeddings for speaker recognition//Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, Canada, 2018: 5329-5333

[32] Wang Y, Stanton D, Zhang Y, et al. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 5180-5189

[33] Skerry-Ryan R, Battenberg E, Xiao Y, et al. Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 4693-4702

[34] Huang Z, Lu H, Lei M, et al. Linear networks based speaker adaptation for Speech Synthesis//Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, Canada, 2018: 5319-5323

[35] Kons Z, Shechtman S, Sorin A, et al. High quality, lightweight and adaptable TTS using LPCNet//Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech 2019). Graz, Austria, 2019: 176-180

[36] Deng Y, He L, Soong F. Modeling multi-speaker latent space to improve neural TTS: Quick enrolling new speaker and enhancing premium voice. arXiv preprint arXiv: 1812.05253, 2018

[37] Bollepalli B, Juvela L, Alku P, et al. Lombard speech synthesis using transfer learning in a Tacotron text-to-speech system//Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech). Graz, Austria, 2019: 2833-2837

[38] Variani E, Lei X, Mcdermott E, et al. Deep neural networks for small footprint text-dependent speaker verification//Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Florence, Italy, 2014: 4052-4056

[39] Miao Y, Zhang H, Metze F. Speaker adaptive training of deep neural network acoustic models using i-vectors. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(11): 1938-1949

[40] Swietojanski P, Li J, Renals S. Learning hidden unit contributions for unsupervised acoustic model adaptation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2016, 24(8): 1450-1463

[41] Wu Z, Swietojanski P, Veaux C, et al. A study of speaker adaptation for DNN-based speech synthesis//Proceedings of the 16th Annual Conference of the International Speech Communication Association. Dresden, Germany, 2015: 879-883

[42] Luong H T, Yamagishi J. Scaling and bias codes for modeling speaker-adaptive DNN-based speech synthesis systems//Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT). Athens, Greece, 2018: 610-617

[43] Zen H, Senior A. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis//Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Florence, Italy, 2014: 3844-3848

[44] Zhang J X, Ling Z H, Liu L J, et al. Sequence-to-sequence acoustic modeling for voice conversion. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2019, 27(3): 631-644

[45] Valin J M. A hybrid DSP/deep learning approach to real-time full-band speech enhancement//Proceedings of the 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP). Vancouver, Canada, 2018: 1-5

[46] Povey D, Ghoshal A, Boulianne G, et al. The kaldi speech recognition toolkit//Proceedings of the IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. Waikoloa, USA, 2011

[47] Bu H, Du J, Na X, et al. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline//Proceedings of the 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA). Seoul, Korea, 2017: 1-5

[48] Battenberg E, Skerry-Ryan R, Mariooryad S, et al. Location-relative attention mechanisms for robust long-form speech synthesis//Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020). Barcelona, Spain, 2020: 6194-6198

XU Zhi-Hang, M. S. candidate. His research interest is speech synthesis.

CHEN Bo, Ph. D. candidate. His research interests include speech synthesis and voice conversion.

ZHANG Hui, M. S. His research interest is speech synthesis.

YU Kai, Ph. D., professor. His research interests include cognitive spoken dialogue systems, speech synthesis, speech recognition, understanding and machine learning, etc.

Background

Recently, the end-to-end Text-To-Speech (TTS) system has approached high speech quality and naturalness. It draws the interests of companies and research groups on creating personal-designed speech assistants using the speech data recorded by clients. However, the data from clients are usually in a small amount and recorded in their daily life. As far as we know, building an end-to-end TTS system with a small amount of data is still not a well-solved problem. It is also challenging to build an end-to-end TTS system with a small amount of dirty speech. Adaptation from a pretrained TTS model is a popular and commonly used.

This paper works on speaker adaptation method of speech synthesis under small data. These client-recorded data have several severe problems including poor speech quality, background noise, pronunciation mistakes. Adaptation on utterance level speaker embedding cause less similarity. And direct parameter updating may cause over-fitting and other unstable speech synthesizing. Our proposed phoneme level speaker embedding method provide a new aspect of using different grains speaker embedding and improve the quality of synthesized speeches both on similarity and naturalness, which can be applied on every TTS models for speaker adaptation.