

单位代码： 10359

学 号： 2018180112

密 级： 公开

分类号： TP181

合肥工业大学

Hefei University of Technology

硕士学位论文

MASTER'S DISSERTATION

论文题目： 基于神经网络的情感语音合成方法

学位类别： 专业硕士

专业名称： 计算机技术

作者姓名： 代子彪

导师姓名： 安鑫 副教授 王伟 高级工程师

完成时间： 2021 年 4 月

合 肥 工 业 大 学

专业硕士学位论文

基于神经网络的情感语音合成方法

作者姓名： 代子彪

指导教师： 安鑫 副教授

王伟 高级工程师

专业名称： 计算机技术

研究方向： 情感语音合成

2021 年 4 月

A Dissertation Submitted for the Degree of Master

Emotional speech synthesis based on neural network

By

Dai Zibiao

Hefei University of Technology

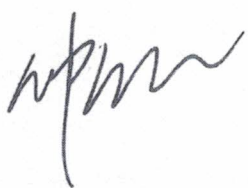
Hefei, Anhui, P.R.China

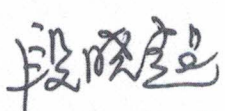
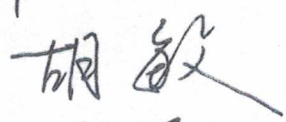
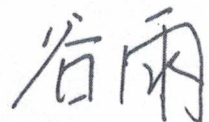

05, 2021


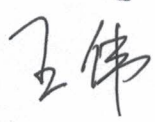
合 肥 工 业 大 学

本论文经答辩委员会全体委员审查, 确认符合合肥工业大学
学学历硕士学位论文质量要求。

答辩委员会签名 (工作单位、职称、姓名)

主席:  张敏七 教授

委员:  中国电科 38 所 研究员
 合肥工业大学 教授
 合肥工业大学 教授
 中科院合肥智能所 副研究员

导师:  合肥工业大学 副教授
 合肥皇图时空科技 高级工程师
有限公司

学位论文独创性声明

本人郑重声明：所呈交的学位论文是本人在导师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。对本文成果做出贡献的个人和集体，本人已在论文中作了明确的说明，并表示谢意。

学位论文中表达的观点纯属作者本人观点，与合肥工业大学无关。

学位论文作者签名：

代子彪

签名日期：

2021年5月27日

学位论文版权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用学位论文的规定，即：除保密期内的涉密学位论文外，学校有权保存并向国家有关部门或机构送交论文的复印件和电子光盘，允许论文被查阅或借阅。本人授权合肥工业大学可以将本学位论文的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名：

代子彪

指导教师签名：

史超

签名日期：2021年5月27日

签名日期：2021年5月27日

论文作者毕业去向

工作单位：北京有创居网络技术有限公司

联系电话：010-83460000

E-mail: sanfang@bytedance.com

通讯地址：北京市丰台区林荫路

邮政编码：101299

街13号信息大厦802室

致 谢

时光荏苒，三年的研究生生活即将结束。这段时间的工作经历和学术研究的过程，我深深体会到了写作论文时的那份宁静和思考。回首三年的求学历程，对那些引导我、帮助我、激励我的人，我心中充满了感激。

首先要感谢导师安鑫教授，是他教会了我如何静下心来做学术，如何从不同的方向去深入的思考问题。在三年的研究生生活中，安老师使我懂得了怎样将书本上的知识应用到实际工程上，使我能从实际出发重新审视学习到的知识，在学习和生活上都给予了我无微不至的关心和知道。安老师开阔的学术思想和对待生活的态度将成为我终生学习的榜样。

同时，我要感谢所有教导过我、关心过我的老师。你们为我的学业倾注了大量心血，你们为人师表的风范令我敬仰，严谨治学的态度令我敬佩。感谢一直关心与支持我的同学和朋友们，三年来，我们朝夕相处，共同进步，感谢你们给予我的所有关心和帮助。同窗之谊，我将终生难忘！在此也要感谢我生活学习了3年的学校——合肥工业大学，学校给了我一个宽阔的学习平台，让我不断吸取新知识，充实自己。

最后，真挚的感谢在百忙之中抽出时间评阅本篇论文的专家和教授。感谢你们提出的专业评审意见，感谢你们无私伟大的工匠精神。

作者：代子彪

2021年4月17日

摘 要

作为人机语音交互的出口,语音合成的效果直接影响到人机交互的体验。一个高质量的、稳定的语音合成系统能够让机器更加地拟人化,使人机交互过程更加自然。目前,大多数很多优秀的致力于提高中性语音成的质量的 TTS 模型已经被提出,例如 Tacotron2 和 WaveNet。但这些模型大多数使用的是 RNN 或者 LSTM 作为编码器和解码器,这种自回归的结构导致这些模型在训练和预测时很慢。此外,随着智能化语音合成系统的不断完善,人们对增强语音自然度的要求也越来越高。近几年,针对情感语音的分析与合成正成为新的研究热点,越来越多的研究员致力于研究如何合成富有表现力的情感语音。但是,在情感语音合成领域,开源的情感数据集很少,且大多数数据来自不同的发言人,导致可用来训练的数据集规模都很小,一定程度上限制了基于深度学习的情感语音合成模型的效果。

针对以上问题,本文主要工作如下:

(1) 针对基于 RNN 的神经网络语音合成模型训练和预测效率低下以及长距离信息丢失的问题,提出了一个基于 Bert 的端到端的语音合成模型 Bert TTS,该模型能合成高质量的英语音频。并且,该模型使用预训练的 Bert 作为编码器,在提高训练速度的同时能有效解决 RNN 那样长距离信息丢失问题。

(2) 针对不同情感的代表特征向量的选择问题,提出了一种基于情感数据集内部各情感的样本向量间距离的方法。该方法在考虑同一情感数据样本分布的同时,也考虑了该情感数据样本与其他情感数据样本的距离。并通过实验证明该方法优于基于均值的特征向量表示法。

(3) 针对情感语音数据集规模小的问题,提出了一种基于中性 TTS 通过在小批量情感语音数据集上微调来合成情感语音的方法。

实验表明本文提出的 Bert TTS 模型能够在得到与 Tacotron2 模型相近效果的基础上,把训练速度提升一倍左右。同时,本文提出的基于中性语音合成模型通过微调在小批量数据集上合成情感语音合成方法能够合成清晰的情感语音,在 MOS 打分测试中总体获得了 3.77 分。

关键词: 情感计算; 语音合成; 循环神经网络; Seq2Seq; WaveGlow; 注意力机制

As the exit of human-machine speech interaction, the effect of speech synthesis directly affects the experience of human-machine interaction. A high-quality and stable speech synthesis system can make the machine more anthropomorphic and make the human-computer interaction process more natural. Currently, many excellent TTS models have been proposed to improve the quality of neutral speech, such as Tacotron2 and WaveNet, but most of these models use RNN or LSTM as encoder and decoder, and this autoregressive structure causes these models to be slow in training and prediction. In addition, with the continuous improvement of intelligent speech synthesis systems, there is an increasing demand for enhanced naturalness of speech. In recent years, the analysis and synthesis of emotional speech is becoming a new research hotspot, and more and more researchers are working on how to synthesize expressive emotional speech. However, in the field of emotional speech synthesis, there are few open-source emotional datasets, and most of the data come from different speakers, resulting in small datasets available for training, which limits the effectiveness of emotional speech synthesis models based on deep learning methods to some extent.

In response to the above problems, the main work of this thesis is as follows:

(1) To address the problems of inefficient training and prediction of RNN-based neural network speech synthesis models and long-distance information loss, an end-to-end Bert-based speech synthesis model, Bert TTS, is proposed, which can synthesize high-quality English audio. Moreover, the model uses pre-trained Bert as an encoder, which can effectively solve the long-distance information loss problem like RNN while improving the training speed.

(2) For the problem of selecting representative feature vectors for different emotions, a method based on the emotion distance ratio within the emotion dataset is proposed. The method considers both the distribution within each emotion sample and other emotion samples around it. And the method is proved to be superior to the mean-based feature vector representation method through experiments.

(3) To address the problem of small size of emotional speech dataset, a method is proposed to synthesize emotional speech based on neutral TTS by fine-tuning it on a small batch of emotional speech dataset.

Experiments show that the Bert TTS model proposed in this paper can improve the training speed by about double while obtaining similar results as the Tacotron2 model.

Meanwhile, the neutral speech synthesis model proposed in this paper is able to synthesize clear emotional speech by fine-tuning the synthesis method on a small batch data set, and obtains an overall score of 3.77 in the MOS scoring test.

KEYWORDS: Emotional computing; Speech synthesis; Recurrent neural network(RNN); Seq2seq; Waveglow; Attention mechanism

目录

第一章 绪论	1
1.1 研究背景和意义.....	1
1.2 语音合成研究现状.....	1
1.2.1 语音合成相关技术概述	2
1.2.2 基于波形拼接的方法	2
1.2.3 HMM 模型和 STRAIGHT 合成技术.....	2
1.2.4 深度神经网络 TTS 模型	3
1.2.5 情感语音合成	5
1.3 本文工作内容.....	5
1.4 论文结构	6
第二章 相关知识和关键技术	8
2.1 循环神经网络 (Recurrent Neural Network, RNN)	8
2.1.1 网络架构	8
2.1.2 梯度消散和梯度爆炸	9
2.1.3 长期依赖问题	10
2.2 基于编码-解码的序列到序列架构 (Seq2seq 架构)	11
2.3 注意力机制 (Attention Mechanism)	13
2.4 Tacotron2 模型	14
2.4.1 模型架构	14
2.4.2 Encoder	14
2.4.3 注意力网络	15
2.4.4 Decoder	15
2.4.5 Tacotron2 的不足.....	16
2.5 预训练语言模型-BERT.....	16
2.5.1 模型架构	16
2.5.2 预训练语言模型	16
2.5.3 BERT 中的 Attention	17
2.6 WaveGlow	19
2.6.1 声码器	19
2.6.2 WaveGlow	19
2.7 本章小结	20
第三章 基于 BERT 的端到端语音合成模型-BERT TTS.....	21
3.1 基于 BERT 的端到端语音合成方法.....	21

3.1.1 文本特征提取和音频特征提取	22
3.1.2 Encoder	24
3.1.3 Decoder	27
3.1.4 Mel Linear 和 Stop Linear	28
3.2 实验与结果与分析	28
3.2.1 数据集	28
3.2.2 实验设置	29
3.2.3 特征预测情况	29
3.2.4 评估	30
3.2.5 训练时间对比	31
3.3 本章小结	32
第四章 基于 BERT TTS 的端到端情感语音合成方法	33
4.1 基于 BERT TTS 的端到端情感语音合成	33
4.1.1 情感特征表示方法	34
4.1.2 基于中性 TTS 模型微调	35
4.2 实验与结果分析	36
4.2.1 数据集	36
4.2.2 实验设置	37
4.2.3 注意力对齐	38
4.2.4 情感表达	38
4.2.5 语音合成质量评估	39
4.3 本章小结	39
第五章 总结与展望	41
5.1 总结	41
5.2 展望	41
参考文献	42
攻读硕士学位期间的学术活动及成果情况	47

插图清单

图 2.1	RNN 网络架构	8
图 2.2	\tanh 及其导数的函数图像	10
图 2.3	RNN 数据传递方式	11
图 2.4	Seq2seq 模型架构.....	11
图 2.5	注意力机制结构图	13
图 2.6	Tacotron2 架构.....	14
图 2.7	BERT 架构	17
图 2.8	WaveGlow 架构	20
图 3.1	基于 BERT 的语音合成架构	21
图 3.2	特征预测情况	29
图 3.3	不同长度的语音合成样本 MOS 比较	31
图 3.4	训练时损失变化	32
图 4.1	基于 BERT TTS 的情感语音合成模型架构	33
图 4.2	经 t-SNE 降维后的样本分布情况	34
图 4.3	基于中性 TTS 系统的微调方式	36
图 4.4	情感表达偏好测试结果	38

表格清单

表 3.1	实验结果 MOS 评分	30
表 4.1	情感语音数据集组成	37
表 4.2	各情感类型实验结果 MOS 评分	39

第一章 绪论

1.1 研究背景和意义

语音合成 (Speech Synthesis), 又叫作文语转换 (Text-to-Speech^[1], TTS) 技术, 是指计算机通过分析将目标文本翻译成目标语种流畅语音的技术。在人机语音交互系统中, 语音合成俨然是其核心技术之一^[2], 同时语音合成也是语音处理技术中的一个重要方向。作为人机语音交互的出口, 语音合成的效果直接影响到人机交互的体验。一个高质量的、稳定的语音合成系统能够让机器更加地拟人化, 使人机交互过程更加自然。

按照人类言语功能的不同层次, 语言合成也可分成三个层次: 1) 从文字到语音的合成 (Text-To-Speech); 2) 从概念到语音的合成 (Concept-To-Speech); 3) 从意向到语音的合成 (Intention-To-Speech)。这三个层次反映了人类大脑中形成说话内容的不同过程, 涉及人类大脑的高级神经活动。为了合成高质量的音频数据, 不仅需要制定各种规则, 也要对输入文本的上下文有一定得理解, 这涉及到自然语言处理的相关问题。在以前, 文语转换的过程是先将输入文字序列转换为韵律序列, 再通过语音合成器来生成时域波形。其中, 文本到音素的转换过程中需要语言学知识的参与; 在将韵律序列转化为时域波形的时候就需要用到先进的语音合成技术来实时的合成高质量音频。一般来说, 文语转换系统不仅要应用数字信息处理技术, 也要有语言学知识的参与。因此, 任何文语转换系统都需要一套复杂的文字序列到音素序列的转换程序, 语音合成任是它的最重要的部分, 充当了机器的“嘴巴”, 任何人机交互过程都需要语音合成器的参与。

情感是人际沟通的重要手段, 是人适应生存的心理工具, 也是激发心理活动和行为的动机。情感从生物进化的角度被分为基本情感和复杂情感。其带有很强的主观性, 因此对其进行分析, 建立有效的情感理论, 对情感语音分析与合成具有重要意义。近几年, 情感语音的合成是热门方向, 越来越多的研究员致力于研究如何合成带有情感的语音。随着智能化语音合成系统的不断完善, 人们开始深入对增强语音情感强度的研究。研究表明, 以语音的韵律和声学特征为指导因素是现有情感语音合成的主要方向。情感语音合成的研究, 主要有以下 3 个基本问题: 1) 情感语音的情感特征和声学参数如何提取; 2) 情感声学特征与情感状态的映射如何建立; 3) 将文本分析与情感因素结合的预测机制如何建立。目前, 较多的研究致力于提高中性语音成的质量, 并且, 很多优秀的 TTS 模型能合成高质量的中性音频。但情感语音合成方向的研究还任重道远。

1.2 语音合成研究现状

1.2.1 语音合成相关技术概述

综观语言合成技术的研究已有二百多年的历史，但是真正有实用意义的近代语音合成技术是随着计算机技术和数字信号处理技术的发展而发展起来的，主要是让计算机能够产生高清晰度、高自然度的连续语音。近几十年来国际和国内的研究主要集中在按规则文语转换，即将书面语言转换成口头语言。

传统的 TTS 系统由前端和后端组成，前端负责文本分析和语言特征提取，例如分词、部分语音标记、多词消歧和韵律结构预测等；后端则是根据前端的语言功能（例如语言声学参数建模、韵律建模和语音合成）构建的，用于语音合成。在过去几十年里，基于波形拼接的方法以及统计参数的方法占据了语音合成系统的半壁江山。但是，它们的共同缺陷是流程复杂，且对语音库要求相对较高，同时需要研究人员掌握较高的语言和语音学的基础知识。此外，基于这两种方法合成的音频在音律和发音上通常存在毛刺且不稳定的问题，听起来没有人类发音那么自然^[3-6]。

1.2.2 基于波形拼接的方法

波形拼接技术是一种通过波形处理，使得言语的超音段特征发证改变，而音段特征（谱包络）保持不变的时间维处理技术。这种技术最大限度的保留了发音人真实的音质，自然度和清晰度都很高，符合研究人员的预期。

但是，这种直接拼接的方式使得合成的音频听起来十分生硬，波形拼接导致拼接边界不平滑，拼接处往往会产生各种各样的异常，合成效果不够稳定。并且，建库对内存需求较大，构建周期长，扩展性差，不适合用作嵌入式应用。如果待合成的句子中大部分粒子单元都存在于语音库中，那么合成的语音的自然度会非常高，甚至可以达到商业水平。但是它的代价是较为占内存，且模型需要精心设计，对不在语音库中单元几乎没有泛化能力。

1.2.3 HMM 模型和 STRAIGHT 合成技术

接上文所述，基于波形拼接的语音合成方法对语料库要求很高，并且对模型的整体设计要求也较为严格，因为它的所有拼接单元都来自于库。并且，模型需要花费大量的时间去训练。随着基于统计的语音合成方法日渐成熟，基于拼接的方法慢慢被遗弃。基于统计参数的语音合成方法的基本思想是，首先对输入的训练语音进行参数分解，然后对声学参数建模，并构建参数化训练模型，生成训练模型库，最后在模型库的指导下，预测待合成文本的语音参数，将参数输入声码器合成目标语音^[7-8]。

隐马尔科夫模型（HMM）结合谐波加噪声模型一起，解决了这个问题。这种方法在当时也被认为是最有效的基于统计的建模方法。它的具体流程有三个部分：1）选择合适的特征向量来标记语音库中的音频数据，训练模型；2）使用模型将输入文本数据映射为特征向量；3）将第二步生成的特征向量送入一个滤波器组，用

于将该特征向量映射语音数据。该方法所需的音频库很小，并且灵活度高，十分契合移动嵌入式平台。

基于参数的方法解决了基于拼接式的合成方法中语音合成边界人不平滑的问题，但是构建基于参数的模型需要非常丰富的专业领域知识。因而设计困难，并且学习成本很高，它的各模块都必须独立训练，再将他们拼接。这就导致了可能存在误差叠加的问题，使得生成的语音经常模糊听不清而且十分不自然。

1.2.4 深度神经网络 TTS 模型

深度神经网络 DNN (Deep Neural Networks) 属于多层神经网络 MLP (Multiple Layer Perception)，二者在结构上大致相同。不同的是深度学习网络在做有监督学习的时候先做非监督学习，然后将非监督学习到的权值当作有监督学习的初值进行训练。

起初，DNN 网络最先被用在语音识别领域。Google 通过将其使用在他们的语音搜索业务上，使得整体识别率提升 10% 以上。这吸引了许多研究人员的目光，使得越来越多的人开始发掘 DNN 在语音信号增强、机器翻译等语音相关领域的应用方向。深度学习的本质，是构建包含大量隐藏层的深度神经网络来学习到能够拟合训练数据各维特征参数，免去了人工处理特征数据的过程，从而获得高效的分类或预测手段，常被用在语音以及图像处理这些特征边界模糊的任务上。在语音合成领域，最早使用 DNN 对文本信息与对应的音频参数的关系进行建模。DNN 的使用解决了使用其他模型时上下文建模效率低下、上下文数据与输入数据空间割裂、过拟合和音质差等问题。

近年来，随着人工神经网络的迅速发展，端到端^[9]的语音合成模型取得了更好的效果，例如 Tacotron^[10]以及 Tacotron2^[11]等。它们直接从文本产生 Mel 频谱图，然后再通过 Griffin-Lim^[12]算法或者 WaveNet^[13]的声码器合成音频结果。通过端到端的神经网络，合成的音频质量有了极大的提高，甚至可以与人类录音相媲美。通常，端到端的 TTS 模型包含两个组件，一个编码器和一个解码器。编码器尝试从给定的输入序列（单词或音素）中提取关键信息，将它们映射到语义空间中，并生成一系列包含上下文信息的隐藏状态编码，然后解码器从这些上下文信息中提取关键信息来构建解码器隐藏状态然后输出梅尔频谱帧。这种基于注意力机制^[14-15]的 Seq2Seq^[16]结构的编码器和解码器通常由循环神经网络 (RNN) 搭配其他网络组成，比如 LSTM 以及 GRU 等。

Oord 等人^[13]提出了一种能产生原始音频波的神经网络模型 WaveNet。WaveNet 是时域波形的生成模型，其产生的音频质量开始可以与真实人类语音相提并论，并且已经在某些完整的 TTS 系统中使用过。WaveNet 中涉及复杂的文本分析系统以及强大的词典，因为它的输入需要大量领域知识才能产生。

Wang 等人^[10]提出了一个端到端的文语转化模型 Tacotron，该模型可直接从文

本和音频对合成音频数据。Tacotron 是一种序列到序列^[16] (Seq2seq) 的体系结构, 用于从字符序列中产生幅度谱图, 它通过用单个替换这些语言和声学特征的产生来训练过程, 仅从数据端到端的训练神经网络。为了生成音频数据, Tacotron 使用 Griffin-Lim 算法来将 Mel 频谱转化为音频输出。论文中提到, 这只是未来神经语音编码器方法的一个占位符, 因为 Griffin-Lim 产生的特征像质和音频质量比 WaveNet 之类的方法低。

Shen 等人^[11]引入了 Tacotron2, 这是一个用于将文本转化为对于语种音频的神经网络架构。它主要包含 2 个部分, 一个由 RNN 构成的序列到序列特征预测网络, 该网络将输入文本数据映射到 Mel 标度图谱中。另一部分是一个声码器, 该模型的声码器是一个修改后的 WaveNet 模型, 它可以利用 Mel 频谱来合成语音音频。该模型在 MOS 评分中取得了 4.53 分, 是当时的最好成绩。这个模型结合了 Tacotron 和 WaveNet 各部分的优势, 能合成高质量的音频。

Arik 等人^[9]提出了一个基于神经网络的端到端语音合成模型 Deep voice1, 它可以实现将文本转化为目标语种的音频。随后, Arik 等人^[17]在 Deep voice1 基础上二次迭代, 提出了一种使用低维数据嵌入来增强文本到语音的训练的模型 Deep voice2, 该模型可以产生不同语调的声音。该模型与 DeepVoice1 有类似的模型架构, 但是该模型合成的音频质量显著提高。并且, 该模型能够从小批量数据集中学习到很好的效果, 它可以仅从每个发言人半个小时左右的音频数据中学习到该发言人独特的声音。Ping 等人^[18]提出了一个基于 Seq2seq 架构的端到端语音合成模型 Deep voice3, 改模型采用全卷积得方式实现了文本到 Mel 频谱的转化。这个模型可以将不同的文本特征 (例如字符、因素等) 转化为不同的声码器参数 (例如 Mel 频谱图、基频谱图等), 然后将这些参数作为音频波形合成模型的输入。

上述模型大多是基于 RNN 来进行建模的, 然而 RNN 作为一种自回归模型, 其第 i 步的输入包含了第 $i-1$ 步输出的隐藏状态, 这种时序结构限制了训练和预测过程中的并行计算能力。此外, 这种结构还会导致当输入序列过长时来自许多步骤之前的信息在传递过程中逐渐消失进而使生成的上下文信息存在偏差的问题。为了解决上述问题, Aaron 等人^[19]提出了一种并行训练模型, 他们引入了一种叫做概率密度蒸馏的方法, 从一个训练过的 WaveNet 中训练一个并行前馈网络。该方法整合了逆自回归流和波形网的特性, 这些特性保证了 WaveNet 训练的有效性以及自回归流的有效采样。Prenger 等人^[20]提出了一个能够从梅尔频谱图生成语音的基于流的网络 WaveGlow。它结合了基于流的生成模型 Glow^[21]的高效推理和预测以及热门语音合成模型 WaveNet 训练快、效果好的优点, 从而提供高质量音频的合成, 并且无需进行自回归。WaveGlow 不需要使用教师网络指导, 而使用单个损失函数进行训练, 可以最大化训练数据的似然性, 使训练过程简单、稳定。

综上所述, 基于深度神经网络的语音合成方法往往能取得非常好的效果, 近年

来更多的研究致力于研究并行计算的方法避免使用 RNN 带来的训练慢、长距离信息丢失等弊端。

1.2.5 情感语音合成

文本语音转换 (TTS) 系统的目的是合成类似于人的语音信号, 以便可以清楚地传达语言和副语言信息。得益于基于深度学习的 TTS 技术的最新进展, 从合成语音中理解文本的上下文含义已不再是问题。然而, 仍然难以通过合成语音来表达诸如情感之类的语言信息。情感语音合成是当下的热门技术, 合成带情感语音有两种方法^[22], 即情感信息在中性语音合成前或后被嵌入进去。前者将情感特征标签添加到输入文本中, 从提取上下文信息时获取情感信息; 第二种方式先合成不包含情感信息的中性语音, 然后再通过一定得技术添加情感表达信息。

在基于深度学习的端到端 TTS 系统 (例如 Tacotron) 中, 合成语音的表达是通过调节附加嵌入向量来控制的, 这些嵌入向量隐式提供通常由另一个神经网络生成的与韵律相关的潜在特征^[23-26]。但是, 由于这些系统仅模仿了参考音频的通用说话风格, 因此很难将用户定义的情感类型分配给合成语音。Lee 等人^[23]直接将情感标签提供给 Tacotron 系统的解码器, 以便将其与 pre-net 的输出连接在一起。尽管该方法显示了在合成语音中表达情感的可行性, 但是灵活控制情感表达也具有挑战性。

更好的方法是通过分析情感标记数据库的分布来学习每种情感的代表性嵌入样式。由于从相同情感类别获得的嵌入向量包含相似的韵律信息, 因此它们倾向于形成聚类。Kwonet 等人使用代表条件向量有效地生成了情感语音, 该代表条件向量是通过平均属于每个情感类别的样式嵌入向量而获得的。但是, 实验过程中发现该方法不能清楚地表现出每种情感的鲜明特征。由于样式嵌入向量在嵌入域中的分布高度分散, 因此它们的均值不是一个好的代表向量。

1.3 本文工作内容

考虑到基于 RNN 深度网络端到端语音合成模型存在训练缓慢、长距离信息丢失等问题, 本文提出了一种使用 BERT^[19]来替换 TTS 模型中编码器的新模型。该模型结合 Tacotron2 和 BERT 的优势, 使用预训练模型 BERT 作为编码器从输入的文本中提取上下文信息, 并通过微调的方式来拟合下游任务。BERT 模型在 11 项 NLP 任务上取得最佳成绩已经表明其具有很强的泛化能力和特征表示能力, 其具有的自注意力机制可以同时计算输入序列的注意力相关性, 在提高并行能力的同时缓解了 RNN 长距离信息丢失的问题。本文的模型使用文本作为输入, 输出为梅尔频谱图, 并使用 WaveGlow 作为声码器来合成音频。

情感语音合成是一门新兴技术, 但是基于深度学习的算法需要大量的情感语音数据, 这些数据通常很难收集且成本很高。特别地, 用很少的数据挑战性地对不同

说话者、不同风格或不同情绪的语音变化建模。在本文中，研究了如何在预训练的基于深度学习的中性 TTS 模型上进行微调，来与其他发言人的一小部分情感数据集进行语音合成。然后，本文通过使用较小的情感数据集对中性 TTS 模型进行微调，研究将这种模型应用于情感 TTS 的可能性。

本文的研究内容主要包括以下三个方面：

- 1) 提出了一个基于 BERT 的端到端的语音合成模型-BERT TTS，该模型能合成高质量的音频。并且，该模型使用预训练的 BERT 作为编码器，在提高训练速度的同时能有效解决 RNN 那样长距离信息丢失问题；
- 2) 提出了一种不同于常用的基于均值的情感特征向量表示的方法，该方法同时考虑了各情绪样本内部以及周围其他情绪样本的分布。并通过实验证明该方法优于基于均值的特征向量表示法；
- 3) 在预训练好的中性语音合成模型 BERT TTS 的基础上，通过在小批量情感语音数据集上微调来合成情感语音，验证了基于中性 TTS 系统在小批量数据集上微调来合成情感语音的可行性。

1.4 论文结构

本文共有五个章节，文章内容和结构组织如下：

第一章为绪论，主要介绍了论文的研究背景和意义，简单介绍了语音合成发展的主流方向，以及基于深度学习的语音合成领域研究的新要求和挑战。接下来从传统的语音合成方法、基于深度学习技术的语音合成方法以及情感语音合成方法入手，阐述了当前国内外研究现状。

第二章介绍了语音合成常用基础架构以及当下热门的语音合成模型两部分内容。先分别介绍了 RNN 结构、用来处理时序序列的 Seq2seq 架构以及列举了常见的注意力机制结构，并说明阐述了基于 RNN 的深度神经网络模型的局限性。随后介绍了热门语音合成模型 WaveGlow，深入分析了它的架构和优缺点。最后介绍了热门端到端语音合成框架 Tacotron2 以及预训练语言模型 BERT 的架构和优势。

第三章提出了一个基于 Tacotron2 和 BERT 的端到端语音合成模型，他能合成高质量的英文音频。主要包括对该模型编码器、解码器以及对 2 个线性预测器的描述。同时，针对本文提出的模型进行实验评估，并且对实验过程收集的数据进行分析，也介绍了本实验所使用的实验设置、数据集以及模型评估方法等，并对实验结果进行了分析。

第四章先简要描述了情感语音合成当下存在数据集规模小的问题，随后提出了一种基于中性 TTS 系统在小批量情感数据集上通过微调来合成情感语音的方法。基于第三章提出中性语音合成音频 BERT TTS 模型微调能合成高质量的情感语音。为了进一步提高情绪表达的清晰度，本章也提出了一种高效的情感代表特征表示。该特征表示法基于内部间情感比，同时也考虑了情感样本内部和类别间样式权重

之间的嵌入距离。通过志愿者偏好实验证明了本文提出的情感向量表示法优于基于传统的基于均值的情感向量表示法，并且也设计 MOS 评分实验验证了本文提出的情感语音模型能合成清晰的语音，在 MOS 打分实验中获得了总体 3.77 的评分。这也验证了基于中性 TTS 系统在小批量数据集上微调来合成情感语音的可行性。

最后，第五章总结了本文的研究工作并说明了当前研究成果的不足，展望了下一步的研究方向，为后续的研究工作打下基础。

第二章 相关知识和关键技术

本部分首先介绍当前热门的端到端语音合成模型的重要组成部分，包括 RNN、Seq2Seq 结构以及注意力机制等，随后简要介绍了一个语音合成声码器模型 WaveGlow。

2.1 循环神经网络 (Recurrent Neural Network, RNN)

2.1.1 网络架构

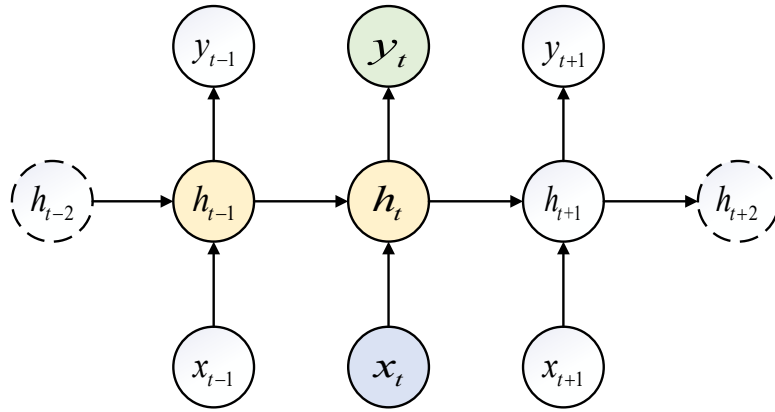


图 2.1 RNN 网络架构

Fig 2.1 Network architecture of RNN

循环神经网络或 RNN (Rumelhart 等人^[27]) 是一类用于处理序列数据的神经网络,其架构如图 2.1 所示。通常情况下,语音数据会随时间序列发生变化,普通深度神经网络无法对语音数据进行建模。循环神经网络作为一种可以处理时序变化数据的神经网络被提出,RNN 是被设计用来处理序列 $x^{(1)}, \dots, x^{(t)}$ 的深度神经网络结构,相较于其他网络, RNN 可以处理可变长度的序列,且序列的长度可以很长。

循环神经网络是一种自回归的结构,这种结构可以用来处理时间维度的数据。其中数据从 $t=1$ 到 $t=\tau$ 的每个时间步的传递方程如下所示:

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)} \quad (2.1)$$

$$h^{(t)} = \tanh(a^{(t)}) \quad (2.2)$$

$$o^{(t)} = c + Vh^{(t)} \quad (2.3)$$

$$\hat{y}^{(t)} = \text{soft max}(o^{(t)}) \quad (2.3)$$

其中 \hat{y} 为标准化后概率的输出向量, $h^{(0)}$ 为初始的隐藏状态。偏置向量 b 和 c 连同权重矩阵 U 、 V 和 W , 分别对应于输入到隐藏、隐藏到输出和隐藏到隐藏层的连接权重。

2.1.2 梯度消散和梯度爆炸

当待训练网络图的深度非常深的时候, 优化算法往往会面临一个难题就是长期依赖问题。由于很深的网络结构使得先前的信息在传递过程中慢慢消散, 模型就失去了对先前信息的掌握, 这让后续的学习变得十分困难。待训练网络过深的问题不仅仅存在于前馈神经网络中, 也同样存在于 RNN 中。并且由于 RNN 模型是参数共享的, 模型网络在各时间序列重复使用相同的权重来构建计算图, 这使得梯度消散和爆炸的问题更加严重。根本问题在于, 在很多阶段的传播后的梯度值更倾向于消失或爆炸的情况。下面举例分析。

首先假设神经元在前向传导过程中没有激活函数, 则前三个时间点的隐藏状态有:

$$\begin{aligned} h_1 &= W_{IH}x_1 + W_{HH}h_0 + b_h, y_1 = W_{HO}h_1 + b_0 \\ h_2 &= W_{IH}x_2 + W_{HH}h_1 + b_h, y_2 = W_{HO}h_2 + b_0 \\ h_3 &= W_{IH}x_3 + W_{HH}h_2 + b_h, y_3 = W_{HO}h_3 + b_0 \end{aligned} \quad (2.5)$$

对于一个序列训练的损失函数为:

$$L(y, \hat{y}) = \sum_{t=0}^T L_t(y_t, \hat{y}_t) \quad (2.6)$$

其中 $L(y, \hat{y})$ 为 t 时刻的损失。利用 $t=3$ 时刻的损失对 W_{IH} , W_{HH} , W_{HO} 求偏导, 有:

$$\begin{aligned} \frac{\partial L_3}{\partial W_{HO}} &= \frac{\partial L_3}{\partial y_3} \frac{\partial y_3}{\partial W_{HO}} \\ \frac{\partial L_3}{\partial W_{IH}} &= \frac{\partial L_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial W_{IH}} + \frac{\partial L_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial W_{IH}} + \frac{\partial L_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_{IH}} \\ \frac{\partial L_3}{\partial W_{HH}} &= \frac{\partial L_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial W_{HH}} + \frac{\partial L_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial W_{HH}} + \frac{\partial L_3}{\partial y_3} \frac{\partial y_3}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W_{HH}} \end{aligned} \quad (2.7)$$

因此, 不难得出对于任意时刻 t , W_{IH} , W_{HH} 的偏导为:

$$\frac{\partial L_t}{\partial W_{IH}} = \sum_{k=0}^t \frac{\partial L_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \left(\prod_{j=k+1}^t \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_k}{\partial W_{IH}} \quad (2.8)$$

$\frac{\partial L_t}{\partial W_{HH}}$ 同理可得。对于 $\frac{\partial L_t}{\partial W_{HH}}$, 在存在激活函数的情况下, 有:

$$\prod_{j=k+1}^t \frac{\partial h_j}{\partial h_{j-1}} = \prod_{j=k+1}^t f'_H(h_{j-1}) W_{HH} \quad (2.9)$$

假设激活函数为 \tanh , 下图刻画了 \tanh 函数及其导数的取值范围:

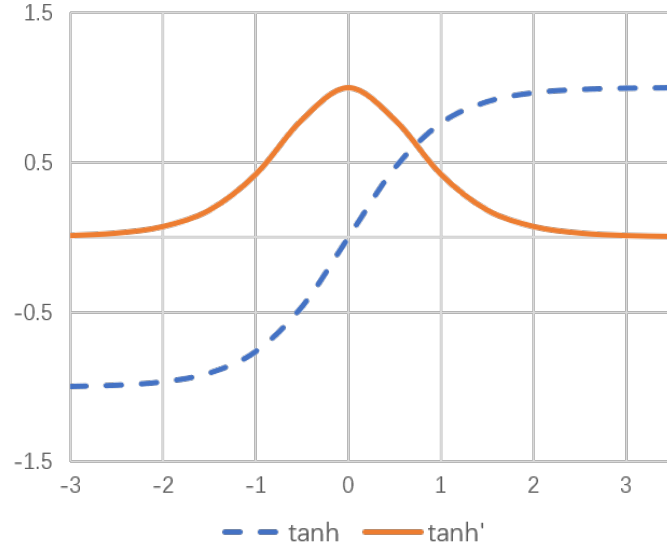

 图 2.2 \tanh 及其导数的函数图像

 Fig 2.2 Function graph of \tanh and \tanh'

由上图 2.2 可知, $0 \leq \tanh' \leq 1$, 同时当且仅当 $x=0, \tanh'(x)=1$ 。因此:

1. 当 t 较大时, $\prod_{j=k+1}^t f'_H(h_{j-1})W_{HH}$ 趋近于 0, 则会产生梯度消散问题;
2. 当 W_{HH} 较大时, $\prod_{j=k+1}^t f'_H(h_{j-1})W_{HH}$ 趋近于正无穷, 则继续向下传递会产生梯度爆炸的问题。

许多资料提供了更深层次的讨论 (Hochreiter 等^[28]; Doya 等^[29]; Bengio 等^[30]; Pascanu 等^[31])。

不同研究人员 (Hochreiter 等人^[28]; Bengio 等人^[30]) 在各自的科研活动中均发现了 RNN 梯度消失和爆炸问题。许多研究希望在训练过程中使得模型参数停留在梯度消失或爆炸未发生的空间来避开这个问题。实际实验过程中, 为了学习到长距离的信息并且对波动具有鲁棒性, RNN 在训练过程中一定会进入到梯度消失的区域。

2.1.3 长期依赖问题

即使 RNN 中的网络参数是稳定的, 长期依赖的问题仍然比短期相互作用困难, 因为 RNN 中存在多个 Jacobian 相乘的情况。RNN 中存在循环结构, 每一时刻的隐藏状态的计算取决于上一时刻的隐藏状态的输出, 这种训练结构使得 RNN 可以有效的保存、记忆和处理过去的语义信息。

但有的时候, 仅需利用最近的信息来处理当前的任务。例如: 考虑一个用于利用之前的文字预测后续文字的语言模型, 如果想预测

The clouds are in the sky.

中的最后一个词。在这句话中，不需要长距离的信息，这个词很显然是 **sky**。在这种相关信息很短的序列中，使用 RNN 可以起到非常好的训练效果。

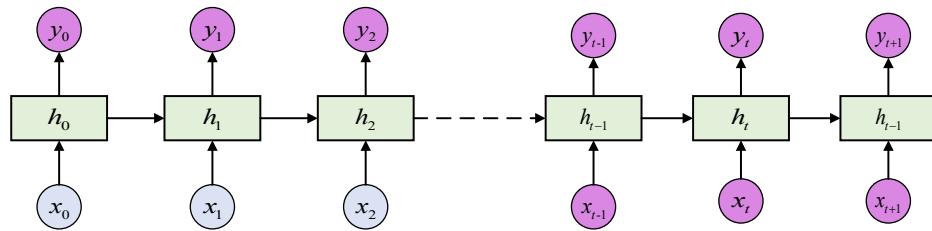


图 2.3 RNN 数据传输方式

Fig.2.3 Data transfer of RNN

但也有很多的情况需要更多的上下文信息，考虑需要预测的文本为

I grew up in France ... I speak fluent **French**.

较近的信息表明待预测的位置应该是一种语言，但想确定具体是哪种语言需要更远位置的“grew up in France”的背景信息。理论上，RNN 可以处理这种序列信息，但在实践中 RNN 却很难解决这个问题。

具体来说，每当模型能够表示长期依赖时，训练时的梯度幅度值相较于短期相互作用会变得很小。但是，这并不意味着无法训练。这表示学习长期依赖可能需要很长的时间，因为长期依赖关系的信息十分容易被短期相关性产生的波动遮蔽。Bengio 等人^[30]的实践实验表明，当输入序列的长度增加时，基于梯度下降的优化效果会下降的很明显，当序列长度达到 15 左右时，使用 SGD 损失函数已经无法成功完成传统 RNN 的训练。

2.2 基于编码-解码的序列到序列架构 (Seq2seq 架构)

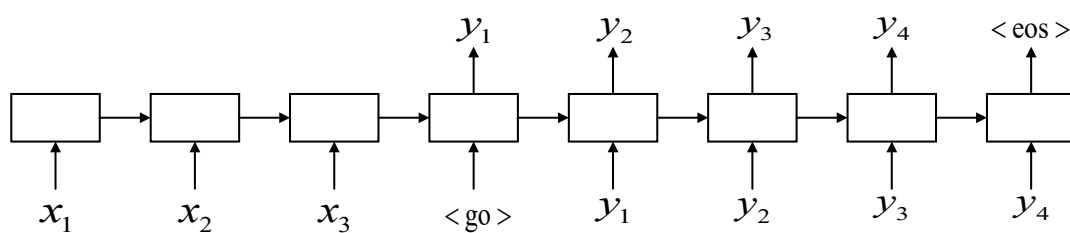


图 2.4 Seq2seq 模型架构

Fig.2.4 System Architecture of Seq2seq Model

最初，Cho 等人^[31]提出了用于在两个可变长度的序列间相互映射的最简单的 RNN 架构，之后不久由 Sutskever 独立开发，并且第一个使用这种方法获得翻译的最好结果。作者将该架构命名为编码-解码结构，或序列到序列 (Seq2seq) 架构，如图 2.4 所示。作者的想法十分简明：编码器 Encoder 用于编码输入序列的上下文信息，将任意长度的序列信息编码成固定长度的上下文向量；Decoder 是解码器，它从编码器输出的上下文向量中提取信息，并解码为输出序列。

在 RNN 提取之前，架构约束输入序列的长度 n_x 和输出序列的长度 n_y 必须一致，即 $n_x = n_y = \tau$ 。RNN 的创新之处在于打破了这种约束，从而可以适应更多的 NLP 任务。在 Seq2seq 的架构中，两个 RNN 共同训练来最大化条件概率 $\log P(y^{(1)}, \dots, y^{(n_y)} | x^{(1)}, \dots, x^{(n_x)})$ 。编码器 RNN 的最后一个状态 h_{n_x} 通常被当作从输入序列提取出的上下文向量 C ，并传递给解码器作为解码器 RNN 的输入。

Seq2Seq 模型将输入序列 (x_1, x_2, \dots, x_T) 映射为输出序列 $(y_1, y_2, \dots, y_{T'})$ ，这一过程由编码输入与解码输出两个环节组成。编码器负责把输入序列编码为一个固定长度的上下文向量，这个向量作为输入传给解码器，输出目标序列 $(y_1, y_2, \dots, y_{T'})$ 。且每个时刻的输出 y_t 由先前时刻的输出 $(y_1, y_2, \dots, y_{t-1})$ 共同决定。大多数情况下，输入序列长度与输出序列长度不相同，即 $T \neq T'$ 。在神经机器翻译(NMT)中，此转换基于条件概率 $p(y_1, y_2, \dots, y_{T'} | x_1, x_2, \dots, x_T)$ 将语言 A 的输入语句转换为语言 B 的输出语句，即

$$h_t = \text{encoder}(h_{t-1}, x_t) \quad (2.10)$$

$$s_t = \text{decoder}(s_{t-1}, y_{t-1}, c_t) \quad (2.11)$$

这里的 c_t 是嵌入后的文本信息通过 attention 计算出来的上下文向量

$$c_t = \text{attention}(s_{t-1}, \mathbf{h}) \quad (2.12)$$

$p(y_1, y_2, \dots, y_{T'} | x_1, x_2, \dots, x_T)$ 可以通过如下方式计算

$$p(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t | y_1, \dots, y_{t-1}, \mathbf{X}) \quad (2.13)$$

并且

$$p(y_t | y_1, \dots, y_{t-1}, \mathbf{X}) = \text{softmax}(f(s_t)) \quad (2.14)$$

这里的 $f(\cdot)$ 表示一个全连接层变换。对于翻译任务，此 softmax 函数需要计算每个维度对应位置上词汇表中每个单词的概率。但是，在 TTS 任务中不需要 softmax 函数，直接将解码器输出的隐藏状态输入到频谱图转换模型中。

这种架构存在一个明显的缺陷，编码器 RNN 将提取的上下文信息编码为一个固定长度的向量，很难概括上下文信息，且这个固定的长度很难确定。Bahdanau 等人^[14]在机器翻译任务中观察到了这种现象。他们首次提出使用可变长度的上下文向量 C 来代替固定长度的向量序列。此外，他们还在文章中引入了注意力机制，来计算上下文向量序列 C 的元素与输出序列的元素相关性，将在下一节介绍。

2.3 注意力机制 (Attention Mechanism)

接上文所说, 使用固定长度的上下文向量 C 这需要使用足够大的 RNN, 并且需要用足够多的训练步骤来提取上下文信息才可能实现。如 Cho 等人^[31]和 Sutskever 等人所表明的。然而, 更有效的办法是一次读取整个句子或段落, 再从上下文中提取有效的信息来一次翻译一个词, 每次聚焦输入序列的不同位置来产生下一个输出词所需要的所有上下文语义。这正是 Bahdanau 等人^[14]第一次引入的想法。

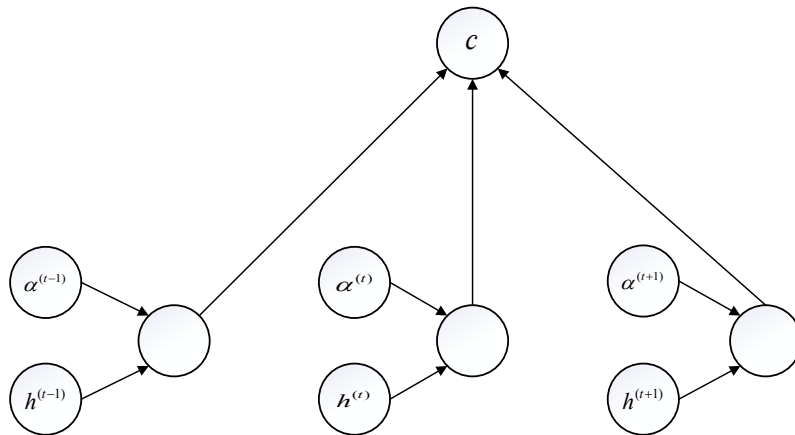


图 2.5 注意力机制结构图

Fig.2.5 Structure diagram of Attention mechanism

图 2.5 中显示了注意力机制的结构图, 图中各时间步关注输入序列的不同位置信息。由 Bahdanau 等引入的现代注意力机制, 本质上是加权平均。注意力机制特征向量 $h^{(t)}$ 按权重 $\alpha^{(t)}$ 进行加权平均, 再拼接到一起就得到了上下文向量 C 。在一些应用中, 特征向量 h 是神经网络的隐藏单元, 有时候可以用输入序列 x 替换。权重 $\alpha^{(t)}$ 由模型内部产生。它们的取值在 $[0,1]$ 之间, 表示相关性, 并且旨在仅仅集中在单个 $h^{(t)}$ 周围。权重 $\alpha^{(t)}$ 通常由模型内部计算相关性得分后用 *soft max* 函数激活产生。基于加权平均的注意力机制可以使用优化算法训练优化来获取, 因为它是平滑、可微的近似。

现代注意力机制的系统需要具备三个组件: 1) 读取器读取输入数据序列并将其转换为机器可理解的向量表示, 某一维向量需要反映每个词的位置; 2) 存储器存储读取器输出的特征向量列表。3) 最后一个为读取存储器的内容来顺序的执行任务, 每个时间步聚焦于某几个具有不同权重的存储器元素的内容。

以翻译为例, 当两种不同语言的句子相互翻译的时候, 不同语言中相应的词对齐时, 可以使对应的词嵌入编码想关联。许多研究表明, 可以将两种不同语言的词嵌入表示方式相关联(Kočiský 等人^[33]), 相较于传统的基于短语表中词频计数的方式可以产生更低的对齐错误率。Klementiev 等人^[34]的工作也对跨语言词向量进行了研究。

2.4 Tacotron2 模型

2.4.1 模型架构

Tacotron2 是 2017 年提出来的一个基于神经网络的端到端语音合成框架，其架构如图 3.1 所示。它包含两个部分：声谱预测网络和声码器（Vocoder）。声谱预测网络是一个 Encoder-Attention-Decoder 网络，用于将输入的字符序列翻译为 Mel 频谱的帧。声码器是 WaveNet 的修订版，用于将预测的 Mel 频谱帧序列即生成音频。

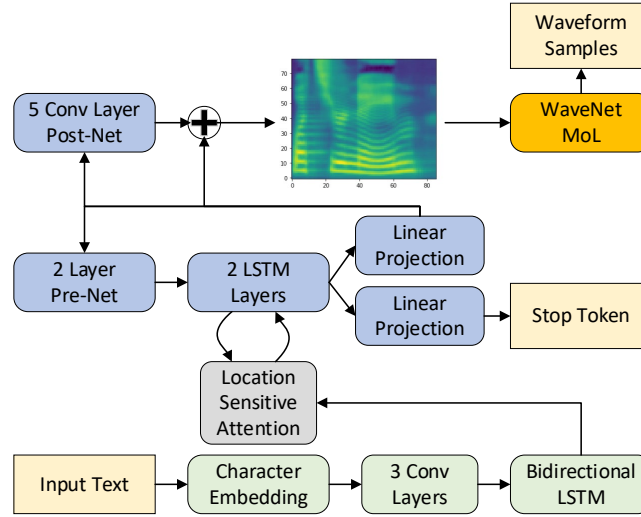


图 2.6 Tacotron2 架构

Fig.2.6 System Architecture of Tacotron2

2.4.2 Encoder

编码器模块包含一个字符嵌入层（Character Embedding），一个 3 层卷积以及一个双向 LSTM 层。输入得文本序列在经过 Embedding 层后被编码成 512 维的字符向量，然后降该字符向量穿过一个 3 层卷积层来从输入文本中提取上下文信息，该卷积层每层包含 512 个维度为 5×1 的卷积核，即每个卷积核一次卷积 5 个字符。卷积层后接批归一化层，接着使用 ReLu 函数进行激活。最后一个卷积层的输出向量用作下一层的输入，被输入双向的 LSTM 层，并用该层的输出作为编码特征向量，该 LSTM 层包含 512 个单元，双向各 256 个单元。上一时刻 $t-1$ 的梅尔频谱帧 y_{t-1} （预测时是上一时刻输出的预测帧，训练时是真实帧）首先使用 2 层全连接网络（2 Layer Pre-Net）进行处理，它的输出与 $t-1$ 时刻的上下文向量 c_{t-1} 连接在一起输入到 2 层的 LSTM 中。编码器输出计算公式为

$$f_e = \text{ReLU}(F_3 * \text{ReLU}(F_2 * \text{ReLU}(F_1 * E(X)))) \quad (3.1)$$

$$H = \text{Encoder Recurrency}(f_e) \quad (3.2)$$

其中, F_1, F_2, F_3 为 3 个不同的卷积核, $ReLU$ 为每一个卷积层上的非线性激活, E 表示对字符序列 X 做 embedding, $EncoderRecurrency$ 表示双向 LSTM。

2.4.3 注意力网络

Tacotron2 中使用了基于位置敏感的注意力机制 (Attention-Based Models for Speech Recognition), 是对[15]中提到的注意力机制的扩展。这样处理可以使用解码过程中累积注意力权重作为一个额外的特征, 因此使得模型在预测过程中不会出现子序列重复或遗漏的情况。接着用 32 个维度为 31×1 的卷积核卷积得到位置特征, 然后把位置特征和输入序列投影到维度为 128 的隐层表征, 再计算得到注意力权重。

Tacotron2 中使用的是混合注意力机制, 在对齐中加入了位置特征:

$$e_{i,j} = \text{score}(s_i, c\alpha_{i-1}, h_j) = v_a^T \tanh(Ws_i + Vh_j + Uf_{i,j}) \quad (3.3)$$

其中, s_i 为当前解码器的隐藏状态, $f_{i,j}$ 是上一时刻的注意力权重 α_{i-1} 经过卷积得到的位置特征, $f_i = F * c\alpha_{i-1}$, h_j 是当前时刻编码器的隐藏状态, v_a, W, V, U, b 为待训练参数矩阵。

2.4.4 Decoder

解码器 (Decoder) 是一个自回归循环神经网络, 它从编码器的输出序列中训练来预测输出 Mel 声谱图。上一时刻预测出的 Mel 频谱首先被传入一个预处理网络层, 该预处理层每层是由 256 个隐藏 ReLU 单元组成的双层全连接层, 实验中发现该预处理层对于学习注意力对齐是必要的。预处理网络层的输出向量与注意力层的输出向量先进行拼接, 在输入到包含 1024 个单元的单向 LSTM 层。LSTM 的输出向量再与编码器输出的注意力上下文向量进行拼接, 接着传递给一个线性投影层来预测当前时刻的 Mel 频谱帧。目标 Mel 频谱帧先经过一个包含 5 层卷积层的后处理层, 将结果通过残差连接叠加到先前的 Mel 频谱帧上, 用来改善频谱重构过程中可能出现的梯度消失问题。后处理层每层由 512 个维度为 5×1 的卷积核组成, 各层都使用 \tanh 函数激活。在预测 Mel 频谱帧的同时, 拼接解码器 LSTM 输出的向量与注意力上下文向量, 经过一个线性投影层后投影成一个数值后传递给 sigmoid 层, 输出的结果可以预测停止标记位, 通过该标记位来判断是否应该停止整个预测过程。计算公式为

$$y_{final} = y + y_r \quad (3.4)$$

其中 y 为原始输入

$$y_r = \text{PostNet}(y) = W_{ps} f_{ps} + b_{ps} \quad (3.5)$$

其中 $f_{ps} = F_{ps,i} * x$ ，其中 x 为上个卷积层的输出或者解码器的输出

2.4.5 Tacotron2 的不足

虽然 Tacotron2 效果很好，但它还存在了一些弊端亟待解决。例如，在面对复杂的单词（例如，decorum）的时候，模型很难预测出正确的发音。甚至，在合成某些特定的单词的时候它甚至会随机生成爆音。Tacotron2 的预测速度很慢，因此不能实时生成语音。此外，Tacotron2 还无法调整生成的语音的情绪，例如使声音包含开心或厌恶的情感在里面。

并且，Tacotron2 中编码器和解码器的主体都是使用的 LSTM，LSTM 本质上也是 RNN 的一种变体。RNN 作为一种自回归模型，其第 i 步的输入包含了第 $i-1$ 步输出的隐藏状态，这种时序结构限制了训练和预测过程中的并行计算能力。此外，这种结构还会导致当输入序列过长时来自许多步骤之前的信息在传递过程中逐渐消失进而使生成的上下文信息存在偏差的问题，当然 LSTM 的信息丢失问题相对 RNN 不那么严重。

2.5 预训练语言模型-BERT

2.5.1 模型架构

BERT (Bidirectional Encoder Representations from Transformers) [19] 是基于 Self-Attention 的预训练语言模型，它的内部结构简单并且可以并行计算，其架构如图 3.2 所示。它是 Transformer 模型的编码器部分，由两个子网：Self-Attention 层和 Feed Forward 层构成，每个子网后接一个残差连接与归一化层。由于 BERT 内部存在自注意力机制，它可以并行的来训练输入文本的双向深度表示。其次，可以在预训练好的 BERT 模型后接一个全连接层经过微调该层来适配下游任务，而不需要大量的特定于任务的额外训练，可以大大减少训练参数和训练时间。BERT 中存在的自注意力机制可以做到并行计算，避免了传统的 RNN 那样的自回归运算过程。得益于自注意力机制的并行计算，BERT 可以更加全面的来获取句子级别的语义，因为它同时能够提取词在句子中的多个不同层次的双向关系特征。BERT 模型在概念上很简单，在实际试验过程中得到的结果也很好。结果表明，在 11 种 NLP 任务上 BERT 都刷新了最新效果，并且在 SQuAD1.1（机器阅读理解顶级水平测试）中的两个衡量指标上，获得了最高分。

2.5.2 预训练语言模型

在当下的 NLP 研究领域，随着计算机算力的越来越强大，越来越多的通用语言表征的预训练模型 (Pre-trained Models, PTMs) 逐渐涌现出来。预训练语言模型

对下游的 NLP 任务十分有帮助，可以避免从头开始训练新的模型，大大节约了训练耗时。预训练模型有 3 个好处：1) 预训练模型可以从大规模语料中学习得到通用的语言表示，并用于下游任务；2) 预训练模型提供了更优的模型初始化方法，使得模型拥有更好的泛化能力同时还可以提高模型的收敛速度；3) 预训练模型迁移到小批量数据集上微调时可以有效的避免过拟合^[35]。

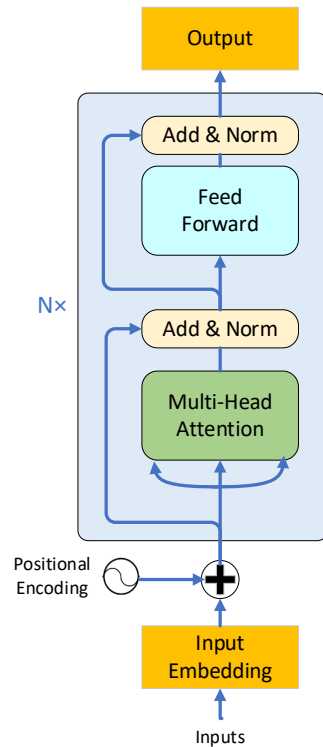


图 2.7 BERT 架构

Fig.2.7 System Architecture of BERT

BERT 发布之后，在 NLP 各个任务上取得了较好的结果。对于预训练 BERT 再微调的方法可以在不同任务上泛化能力很好的原因，主要有如下 3 点：1) 预训练可以为模型达到一个好的初始化效果，相比于随机初始化，其可以有更加广的最优解范围 (wider optima) 和更快达到最优解 (easier optimization)；2) 由于 flat 和 wide 的 optima，训练损失以及泛化误差连续性好，对过拟合有很好的鲁棒性，微调 BERT 可以产生更好的结果；3) BERT 的 lower layers 在微调过程中更难改变 (invariant)，说明靠近输入的层可以学到更多的语言表征^[36] (transferable representations of language)。

2.5.3 BERT 中的 Attention

BERT 这篇文章中提出了 2 种 Attention 机制，分别是 Self Attention 和 Multi-head Attention，下面分别介绍。

2.5.3.1 Self-Attention

接下来本文将介绍 BERT 中的自注意力机制 (Self-attention)，其思想和传统的注意力机制类似，但是 Self-attention 是 Transformer 用来将输入序列中每个单词用相关性权重联系起来。举个例子：

The animal didn't cross the street because it was too tired.

这里的 it 到底代表的是 animal 还是 street，对于人们来说很容易判断，但是对于机器它们无法将 it 和 animal 联系起来。通过自注意力机制可以计算每个词跟其他词的相关性，比如学习 it 和其他词相关性的时候 animal 的相关系数很大，接近 1。

Self-Attention 详细计算公式如下

$$Attention(Q, K, V) = \text{soft max}(\frac{QK^T}{\sqrt{d_k}})V \quad (3.6)$$

其中的 Q, K, V ：

- 1) 在编码器的自注意力机制中， Q, K, V 都来自同一个地方，它们是上一层编码器的输出。对于第一层编码器，它们就是字符嵌入和位置编码相加得到的输入；
- 2) 在解码器的自注意力机制中， Q, K, V 也是自于同一个地方，它们是上一层编码器的输出。对于时刻 0 编码器的输入，同样也是字符嵌入和位置编码相加得到的输入。但是对于编码器，往往不希望它能获得下一时刻 (即将来的信息，不想让他看到它要预测的信息)，因此需要进行 Sequence Masking；
- 3) 在编码器的注意力部分 K, V 来自于解码器的输出，并且 $K = V$ ， Q 表示来自于编码器的上一时刻的输出，；
- 4) Q, K, V 的维度都是一样的，分别用 d_Q, d_K, d_V 表示。

2.5.3.2 Multi-head Attention

原论文^[36]中提到，将 Q, K, V 通过一个线性映射之后分成 h 份，单独对每一份进行 self-attention 训练效果会更好。然后，把各个部分的结果拼接起来 (拼接的方式有很多种，针对不同的任务采取不同的拼接方式效果会更好)，再经过一个线性层的映射来获取最终的输出。这就是所谓的 multi-head attention。上面的超参数 h 就是多头注意力机制中头的数量。Multi-head Attention 的公式如下：

$$MultiHead(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_n)W^0 \quad (3.7)$$

其中

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (3.8)$$

本文的实验中, $d_{model}=512, h=8$, 即 $d_Q = d_K = d_V = d_{model} / h = 512 / 8 = 64$ 。

2.6 WaveGlow

2.6.1 声码器

声码器是对前段模型的预测输出的进一步转换, 完成语音频谱特征空间到音频波形空间的映射。声码器可分为神经网络和非神经网络声码器。非神经网络声码器是指通过算法设计完成频域到时域的转换, 比如 Griffin-Lim 算法、Word (Morise 等人^[38]) 声码器等等; 神经网络声码器是基于深度学习网络, 将卷积网络 (Waibel 等人^[39])、循环神经网络等网络结构进行拼接及堆叠, 结构化误差并进行反向传播, 调整网络参数, 让模型能够学习到频谱特征空间到时域矢量特征空间的映射关系, 从而完成语音合成。

2.6.2 WaveGlow

Waveglow 从数据分布的角度来分析语音合成过程。它基于生成式模型和数据分布角度考虑, 输入空间 (即梅尔频谱特征空间) 与输出空间 (即音频时域信息) 处于不同的空间分布, 那么可以基于条件概率进行建模, 假设用 x 表示时域采样值, $p(z)$ 表示高斯分布, 条件概率公式为:

$$q(x) = \int_{-\infty}^{+\infty} q(z)q(x|z)dz \quad (2.15)$$

最大化取对数后的概率分布函数, 即学习输入空间与输出空间的映射关系。Waveglow 将生成式模型的思想应用到频谱特征合成上: 首先, 将 $q(x|z)$ 等效为狄拉克分布, $q(z)$ 等效为 0 均值同纬度的高斯分布, 借鉴 glow 和 realnvp 的生成式流模型中, 通过随机高斯采样来生成不同样本的结果, waveglow 假设 z 为高斯分布, 设计网络将其与频谱特征结合, 生成音频采样值。简而言之, 设计模型希望能实现 $x = g(z, y)$ 和 $z = f(x, y)$ 的双射关系, 其中函数 g 和函数 f 互为反函数, y 为频谱特征, 使得 z 的高斯采样能够具有导向性。模型将 $q(z)$ 视为标准正态分布, 假设数据维度为 D , 推出概率分布函数如下所示:

$$\frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2}\|z\|^2} \quad (2.16)$$

根据双射函数与概率密度转换关系, 对概率分布函数进一步求解为对 x 和 y 的积分方程, 如下所示:

$$q(z) = \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2}\|f(x,y)\|^2} \left| \det \left[\frac{\partial f}{\partial x, y} \right] \right| \quad (2.17)$$

模型将最大化目标 z 的概率分布，转换成了对 x 和 y 的求解。根据公式和公式得出，要实现映射关系网络必须保证两点：双射关系，雅可比矩阵可求。Waveglow 网络将每层网络设定为可逆卷积，将权重矩阵初始化为正定矩阵，保证雅可比矩阵可求，这样就是实现了特征映射的双射关系。同时，为了保证跨纬度信息融合和防止维度坍塌，参照 glow 中设计耦合层来让网络学习各维度信息交互；为了使得信息充分混合，waveglow 将模型设为多步流式的网络结构，每一步流是一个可逆卷积，再接上仿射耦合层，同时通过参差连接加速收敛。整体模型结构如下图 2.6 所示。

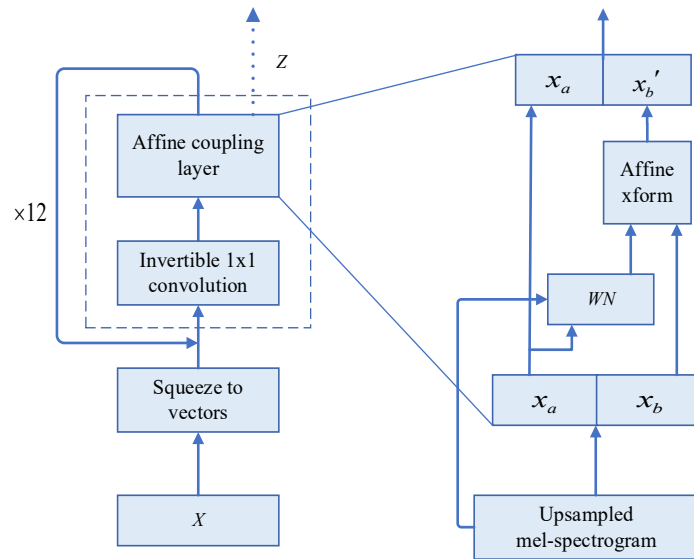


图 2.8 WaveGlow 架构

Fig.2.8 System Architecture of WaveGlow

然而，Waveglow 模型中设计为多步流式，导致网络模型较深，训练参数过大，造成了模型收敛速度过慢的问题。并且，在模型推测阶段，需要分配较大内存，占用较大的计算资源。

2.7 本章小结

本章分别介绍了语音合成常用基础架构以及当下热门的语音合成模型两部分内容。先分别介绍了 RNN 结构、用来处理时序序列的 Seq2seq 架构以及列举了常见的注意力机制结构，并说阐明了基于 RNN 的深度神经网络模型的局限性。随后介绍了热门语音合成模型 WaveGlow，深入分析了它的架构和优缺点。最后介绍了热门端到端语音合成框架 Tcotron2 以及预训练语言模型 BERT 的架构和优势。

第三章 基于 BERT 的端到端语音合成模型-BERT TTS

为了改善 Tacotron2 模型所存在的问题，本章提出了一种基于 BERT 的端到端语音合成模型。该模型在提高训练效率的同时不存在长距离信息丢失的问题。并且，本章的实验对 Tacotron2 的结束标记位预测模块进行了改进，改善了 Tacotron2 偶尔无法正确预测结束帧的问题。

本章将先介绍基于神经网络的端到端语音合成模型 Tacotron 以及预训练语言模型 BERT。然后介绍本文提出的基于 BERT 的端到端语音合成模型的整体架构，由于是基于 Seq2seq 架构，所以包含编码器和解码器部分。对于每一部分，都将针对第 n 条数据介绍详细的建模过程。最后介绍实验步骤和实验结果，并对实验数据展开分析得出结论。

3.1 基于 BERT 的端到端语音合成方法

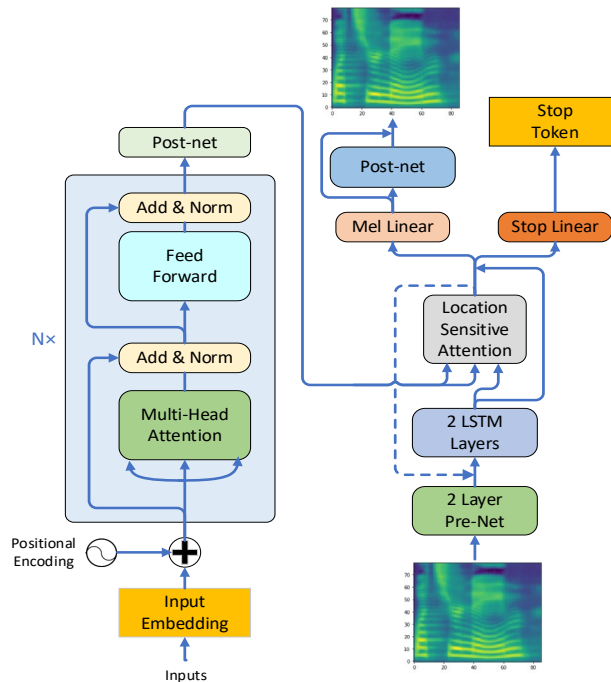


图 3.1 基于 BERT 的语音合成架构

Fig.3.1 System Architecture of TTS Model base on BERT

与基于 RNN 的模型相比，使用 BERT 作为解码器有 4 个优点。首先，基于在大数据集上经过训练的模型 BERT 通过微调额外的输出层来适配下游任务，这使得待训练参数量更少；其次，Self-Attention 可以并行处理编码器的输入，不需要进行自回归计算从而减少非常多的计算时间，提升训练效率；第三，Self-Attention 机制可以同时从上下文中提取左右的信息来建立长期依赖关系，从而避免传统 RNN 那样长距离信息丢失的问题；最后，BERT 在输入文本向量中加入了位置信息编码，使输入信号在前向和后向传播时任意位置之间的间隔缩短到 1。这在神经网络 TTS

模型中有很大的帮助，比如合成波的韵律，它不仅是取决于周围的几个单词，还取决于句子级别的语义。

本节接下来详细介绍 BERT TTS 模型的架构，并详细分析每个部分的功能。总体架构如图 3.3 所示。

3.1.1 文本特征提取和音频特征提取

具体的文本向量和音频特征向量提取过程如下：

首先，根据维基百科所有文本数据建立字典，每一个字符或单词对应一个索引，形如 $\langle string, index \rangle$ 。对第 n 个文本 $Text_n$ 进行标准化处理：1) 去除特殊字符；2) 将缩写转换成全写，例如将‘mrs’改写成‘misess’；3) 将数字转化成对应的英文文本，比如‘9’改成‘nine’，得到预处理后的第 n 个文本 $Text'_n$ ；

将预处理后的第 n 个文本 $Text'_n$ 中的字符串转化为字符，并用 one-hot 向量表示每个字符，从而得到向量化后的第 n 个文本向量，记为 $C^n = \{c_1^n, c_2^n, \dots, c_i^n, \dots, c_m^n\}$ ，其中， c_i^n 表示第 n 个文本向量的第 i 个字符， $i=1, 2, \dots, m$ ， m 为字符串长度。字典中共 30522 个对象，即每个字符向量的维度为 30522；

再利用梅尔频率倒谱系数对第 n 条音频 $Audio_n$ 进行语音特征提取，得到第 n 条语音信息特征 $MFCC_n$ ，从而与向量化后的第 n 个文本向量 C^n 共同构成第 n 条训练数据 $W'(n) = \langle MFCC_n, C^n \rangle$ ；

本文中，使用梅尔倒谱系数（MFCC）作为语音的语音特征。具体的语音特征获取过程如下：

梅尔倒谱系数是在 Mel 标度频率域提取出来的倒谱参数，与频率 f 的关系可以表示式(3.9)：

$$Mel(f) = 2595 \times \lg \left(1 + \frac{f}{700} \right) \quad (3.9)$$

步骤 1.1、利用式(3.10)所以对音频数据进行预处理，来获取平滑的语音信号：

$$H(S) = 1 - \mu S^{-1} \quad (3.10)$$

上式中， μ 表示调整系数，且 $\mu=0.97$ ， S 表示原始语音信号；

步骤 1.2、对平滑的语音数据进行分帧处理，获得分帧后的语音信号 $S(n)$ ；分帧

处理的参数选择与音频的采样频率有关，本章所使用的数据集中音频的采样率为 22050 Hz，设置帧长为 256 个采样点，帧移取 128。

步骤 1.3、对分帧后的语音信号利用式(3.11)和式(3.12)进行加窗处理，通过式(3.11)的海明窗进行加窗之后，能够减少语音信号吉布斯效应的影响，从而获得加窗后的语音信号 $S'(n)$ ：

$$S'(n)=S(n)+W(n) \quad (3.11)$$

$$W(n)=\begin{cases} (1-a)-a \times \cos\left(\frac{2\pi n}{N-1}\right) & 0 \leq n \leq N-1 \\ 0 & n < 0 \text{ or } n > N \end{cases} \quad (3.12)$$

式(3.12)中， a 为调整系数， $a \in (0,1)$ ；本实验中， a 的取值为 0.46；

步骤 1.4、利用式(3.13)对加窗后的语音信号 $S'(n)$ 进行 FFT，获得倒谱的语音信号 $X_a(k)$ ：

$$X_a(k)=\sum_{n=0}^N W(n) \cdot e^{-j\frac{2\pi nk}{N}} \quad 0 \leq k \leq N \quad (3.13)$$

步骤 1.5、利用梅尔滤波器组对倒谱的语音信号 $X_a(k)$ 进行滤波，获得加卷的语音信号；

梅尔滤波器组实质上是满足式(3-13)的一组三角滤波器：

$$\text{Mel}[f(m)] - \text{Mel}[f(m-1)] = \text{Mel}[f(m+1)] - \text{Mel}[f(m-1)] \quad (3.14)$$

式(3.14)中， $f(m)$ 为三角滤波器的中心频率。定义一个滤波器组，该滤波器组满足式 (3.15)，通过这个梅尔滤波器组可以得到经过滤波后的频率信号：

$$H_m(k)=\begin{cases} 0 & , k < f(m-1) \\ \frac{2[k-f(m-1)]}{[f(m+1)-f(m-1)]+[f(m)-f(m-1)]} & , f(m-1) \leq k \leq f(m) \\ \frac{2[f(m+1)-k]}{[f(m+1)-f(m-1)]+[f(m+1)-f(m)]} & , f(m) \leq k \leq f(m+1) \\ 0 & , f(m+1) \leq k \end{cases} \quad (3.15)$$

步骤 1.6、利用离散余弦变换对加卷的语音信号进行解卷，获得静态的梅尔频率倒谱参数 SMFCC；将步骤 1.6 中得到的信号 $H(k)$ 通过式(3.16)进行离散余弦变换(DFT)，得到需要的静态 mfcc 参数 $SMFCC(n)$ ：

$$SMFCC(n) = \sum_{m=0}^{N-1} \log(m) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad n=1,2,\dots,L \quad (3.16)$$

式(3.16)中, L 为 mfcc 的系数阶数, 本实验中, L 取值为 12。

步骤 1.7、利用式(3.17)对静态的梅尔频谱率倒谱参数进行动态差分, 获得一阶差分的梅尔频率倒谱参数;

$$d_t = \begin{cases} S_{t+1} - S_t & t < k \\ \frac{\sum_{k=1}^k k(S_{t+k} - S_{t-k})}{\sqrt{2 \sum_{k=1}^k k^2}} & other \\ S_t - S_{t-1} & t \geq p - k \end{cases} \quad (3.17)$$

式(3.17)中, k 表示一阶导数的时间差, p 表示倒谱系数的阶数, d_t 表示第 t 个一阶差分, S_t 表示第 t 个倒谱系数。本实验中, k 取值为 1。

步骤 1.8、对一阶差分的梅尔频率倒谱参数进行动态差分计算, 获得二阶差分的梅尔频率倒谱参数 d_2MFCC , 即将步骤 1.9 中得到的一阶差分参数带入式(3.17)得到二阶差分参数。

步骤 1.9、利用式(3.18)对静态的梅尔频率倒谱参数、一阶差分的梅尔频率倒谱参数、二阶差分的梅尔频率倒谱参数进行结合获得的 $MFCC$ 既是音频的语音信息特征。

$$MFCC = \frac{N}{3} d_1mfcc + \frac{N}{3} d_2MFCC + \frac{N}{3} SMFCC \quad (3.18)$$

3.1.2 Encoder

在 Tacotron2 中, 编码器是双向 RNN, 本章提出的模型将使用 3.2 节介绍的 BERT 模型部分进行替换与完善。与原始的双向 RNN 相比, Multi-Head Attention 将注意力分散到了多个子空间中, 从而可以在多个不同维度建模输入序列之间的关系, 并且直接在任意两个 token 向量之间建立长期依赖关系, 因此每个 token 向量的输出考虑整个序列的全局上下文, 这对合成音频韵律至关重要。尤其在句子较长的情况下, 使得生成的样本听起来更加平滑自然。BERT 支持的输入句子长度最长为 512 维。另外, 采用 Multi-Head Attention 层代替 RNN 可以通过并行计算来提高训练和预测速度。在 BERT 后接一个维度为 512 的全连接层后处理网络 (Post-Net) 作为编码器的输出层, 经过训练然后去适配下游任务。实验过程中, 通过冻结编码器中 BERT 所有层的参数, 将预训练模型 BERT 作为一个特征提取器, 只重新训练其后接的 Post-Net 层的参数, 得到模型新的权重。

文献[40]中指出, 由于 BERT 是基于 WordPiece 分词器来分词的, WordPiece 分割句子时会把一个完整的词分割成几个子词, 在训练时这些被分开的子词会随机的被 mask。比如 playing 在基于 WordPiece 分词时会被分割为 play 和##ing, 如果此时在随机 mask 过程中##ing 被选中, 由于单个##ing 没有实际意义而导致训练效果没有那么好。随后 Google 推出了名为全词 Mask (Whole Word Masking, WWM) 的升级版。在全词 Mask 中, 如果一个完整的词的部分子词被 mask, 则同属该词的其他部分也会被 mask, 即当##ing 被选中 mask 时会同时将 play 也包含进去。

本章的实验选取的是他们开源的包含 WWM 的英文预训练模型 BERT-large, 其层数 N 为 24, 输出层隐藏状态维度为 1024。

编码器的具体建模过程如下:

编码器神经网络, 包括: 多头注意力层、两个残差连接与归一化层、双层全连接层、单层全连接层; 多头注意力层是由 h 个点积注意力组成; 双层全连接层、单层全连接层中设置有概率为 p 的 dropout 函数以及神经元的激活函数 \tanh 。本实施例中 $p=0.1$;

步骤 2.1、利用式(3.19)得到第 n 个文本向量 C^n 在 t 位置对应的位置信息编码 L_t^n , 从而得到第 n 个文本向量 C^n 的位置信息编码 $L^n = \{L_1^n, L_2^n, \dots, L_t^n, \dots, L_m^n\}$:

$$L_t^n = f(t)^i := \begin{cases} \sin\left(\frac{1}{\delta^{\frac{2i}{d}}} \cdot t\right), & i \text{ 为奇数} \\ \cos\left(\frac{1}{\delta^{\frac{2i}{d}}} \cdot t\right), & i \text{ 为偶数} \end{cases} \quad (3.19)$$

式(3.19)中, t 表示字符在向量化后的第 n 个文本向量 C^n 中的位置, $f(t)^i$ 表示第 i 个字符 c_i^n 在 t 位置的位置信息的计算函数, $:=$ 表示生成符号, δ 表示缩放尺寸, 在本实验中为 10000, d 表示单个字符向量的维度, 本实验中为 512;

步骤 2.2、将第 n 个文本向量 C^n 及其位置信息编码 L^n 在对应位置相加后得到第 n 个输入向量 X^n ; 再将第 n 个输入向量 X^n 输入多头注意力层的每个点积注意力中, 从而利用式(3.20)得出第 j 个点积注意力的输出 α_j :

$$\alpha_j^n = \text{soft max} \left(\frac{Q_j^n (K_j^n)^T}{\sqrt{d_K}} \right) \cdot V_j^n \quad (3.20)$$

式(3.20)中, Q_j^n 表示第 n 个输入向量 X^n 经过 $d \times d_k$ 维的线性变换矩阵 W^Q 的映射后所得到的查询向量, K_j^n 表示第 n 个输入向量 X^n 经过 $d \times d_k$ 维的线性变换矩阵 W^K 的映射后所得到的关键字向量, d_k 表示 K_j 的维度, $(K_j^n)^T$ 表示 K_j^n 的转置, V_j^n 表示第 n 个输入向量 X^n 经过 $d \times d_v$ 维的线性变换矩阵 W^V 的映射后所得到的值向量, d_v 表示 V_j 的维度, $\text{softmax}(\cdot)$ 表示归一化指数函数, $j=1,2,\dots,h$ 。本实验中 $d_k = d_v = 64$;

步骤 2.3、将 h 个点积注意力的输出 $(\alpha_1^n, \alpha_2^n, \dots, \alpha_j^n, \dots, \alpha_h^n)$ 进行拼接, 得到第 n 个向量矩阵 α^n , 从而利用式(3.21)得到多头注意力层最终的输出向量 O^n , 本实验中 $h=16$;

$$O^n = \text{Concat}(\alpha_1^n, \dots, \alpha_h^n) \cdot W^O \quad (3.21)$$

式(3.21)中, $\text{Concat}(\cdot)$ 表示拼接操作, W^O 表示维度为 $d \times d_k$ 的线性变换矩阵;

步骤 2.4、将多头注意力的输出向量 O^n 与其第 n 个输入向量 X^n 经过残差连接与归一化层, 从而利用式(3.22)得到输出向量 H^n :

$$H^n = \text{LayerNorm}(X^n + O^n) \quad (3.22)$$

式(3.21)中, $\text{LayerNorm}(\cdot)$ 表示层归一化函数;

步骤 2.5、将输出向量 H^n 输入双层全连接层中, 从而利用式(3.23)得到相应层的输出 I^n :

$$I^n = \max(0, H^n W_1 + b_1) W_2 + b_2 \quad (3.23)$$

式(3.23)中, W_1, W_2 表示维度为 $d \times d_{ff}$ 的两个待训练的参数矩阵, d_{ff} 表示全连接层隐藏节点的个数即该层的输出维度, b_1 表示第一偏置矩阵, b_2 表示第二偏置矩阵, $\max(\cdot)$ 表示取最大值函数。本实验中 $d_{ff} = 1024$;

步骤 2.6、将双层全连接层的输出 I^n 与输出向量 H^n 经过残差连接与归一化层, 从而利用式(3.22)得到第 n 个文本向量 C^n 的上下文向量 U^n ;

步骤 2.7、上下文向量 U^n 通过一层全连接层处理后得到编码器神经网络输出的维度为 $d \times d_{ff}$ 的缩放后的上下文向量 U^m , 并作为编码器的输出向量;

3.1.3 Decoder

与 Tacotron2 类似，解码器是一个自回归网络，采用的是具有位置敏感注意力机制的 2 层 RNN，可以一次从一帧的编码输入序列中预测 Mel 频谱帧。前一时刻解码器的输出先通过一个预处理网络，该预处理网络包含 2 个全连接层，每层有 256 个激活函数为 ReLU 的隐藏单元。实验表明，作为信息瓶颈的预处理网络对于注意力的学习十分重要。然后，先拼接预处理网络的输出向量以及注意力网络的输出向量，再通过 2 个单向 LSTM，每个具有 1024 个隐藏单元的。连接 LSTM 层的输出和注意力上下文向量，再将结果输入到线性变换层来进行投影预测目标 Mel 频谱帧。最后将当前时刻预测的 Mel 频谱图输入到包含 5 层卷积层的后处理网络中，并对结果进行残差连接以提高整体效果。每层后处理网络由 512 个形状为 5×1 的卷积核组成，并进行 Batch-Normalization，然后在除了最后一层之外的所有层上用 \tanh 函数激活。

具体的建模过程如下：

解码器神经网络，包括：预处理网络、2 个单向 LSTM 层、位置敏感注意力层、停止标记位预测层、Mel 预测层以及后处理网络；预处理网络包含 2 个全连接层，每个全连接层均有 d_{pre} 个 ReLU 隐藏单元；后处理网络 r 层卷积层，每层包含 d_{dec} 个维度为 $k \times 1$ 的卷积核。本实验中 $d_{pre} = 256, d_{dec} = 512, k = 5$ 。

步骤 3.2、定义 t 时刻解码器神经网络的输出为 y_t^n ，定义 t 时刻位置敏感注意力层的输出为 a_t^n ，当 $t=0$ 时，令解码器神经网络的输出 y_t^n 和位置敏感注意力层的输出 a_t^n 均为全 0 矩阵；

步骤 3.2、 t 时刻解码器神经网络的输出 y_t^n 通过预处理网络后得到预处理层的输出向量 e_t^n ；

将预处理层的输出向量 e_t^n 与 t 时刻位置敏感注意力层的输出 a_t^n 连接后通过 2 个具有 d_{enc} 个隐藏单元的单向 LSTM 层，得到输出向量 l_t^n ；本实验中 $d_{enc} = 1024$ ；

步骤 3.3、将 t 时刻单向 LSTM 层的输出向量 l_t^n 与编码器的输出向量 U^m 输入到位置敏感注意力层中得到该层输出的注意力向量 F_t^n ；

步骤 3.4、将注意力向量 F_t^n 通过维度为 $(d + d_{dec}) \times 1$ 的停止标记位预测层的处理后再经过 sig mod 函数激活处理后得到 t 时刻的停止标记值 $token_t^n$ ；

当 $token_t^n > threshold$ 时，表示预测结束，并将所有时刻解码器神经网络的输出 $(y_1^n, y_2^n, \dots, y_t^n)$ 整合为第 n 个输入向量 X^n 最终的目标梅尔频谱帧向量 Y^n 后，开始读入下一条输入文本；

当 $token_t^n \leq threshold$ 时，执行步骤 3.5，其中， $threshold$ 表示停止阈值，在本实验中 $threshold = 0.5$ ；

步骤 3.5、注意力向量 F_t^n 经过 Mel 预测层的处理后输出 Mel 向量 M_t^n ，将 Mel 向量 M_t^n 与注意力向量 F_t^n 进行残差连接后再输入到后处理网络中，经过 d_{dec} 个卷积层的处理后再进行 batch 归一化处理，且所述处理网络除了在最后一层卷积层之外的所有层上用 \tanh 激活函数，从而得到时刻 $t+1$ 的输出 y_{t+1}^n ；

步骤 3.6、将 $t+1$ 赋值给 t 后，返回步骤 3.2 执行。

3.1.4 Mel Linear 和 Stop Linear

本章提出的模型中使用了两个不同的线性层分别预测 Mel 频谱图和结束标志位。实验中发现，对于停止线性层而言，每个序列的末尾只有一个正样本，表示“停止”，而其他帧则有数百个，即负样本有数百个。这种不平衡在结束预测错误的情况下可能无法正常结束导致无限循环，Tacotron2 即存在这个问题。本章的实验中，在计算二进制交叉熵损失时，在尾部的停止标记上施加正权重（5.0-8.0），从而有效的解决了该问题。比如使用 Tacotron2 模型生成“Hello Hello”这句话的音频结果，会得到一个 11s 的音频，而本章提出的模型生成的音频符合预期。

3.2 实验与结果与分析

为了评估模型音频建模的性能，本章设计实验比对了 Tacotron2 以及=BERT TTS 模型合成音频的质量。在本部分首先介绍本章所使用的数据集和实验环境，接着介绍本章所使用的评估方法以及实验结果。本章做了主观配对测试来评估模型性能，通过主观意见平均分（Mean Opinion Score, MOS）来评估合成的音频质量。在主观配对比较实验中，志愿者听完每个模型合成的音频结果，会对所听到的音频的自然度进行 5 分制打分（分值越高越好），打分间隔为 0.5。本章分别对不同模型不同训练阶段的合成样本进行了 MOS 打分测试。

3.2.1 数据集

本章所做的实验均基于 LJSpeech-1.1 语音数据集。该数据集包含了 13100 个单声道演讲者的短音频片段，这些片段来自 7 本非小说类书籍。该数据集包含了在家庭环境中使用内置麦克风在 MacBook Pro 上记录的大约 24 小时的语音数据。

3.2.2 实验设置

本章使用数据集的语音信号采样率为 22050 Hz，采样位为 16bit，使用海明窗处理，帧长为 50ms，帧移为 12.5ms，预加重系数为 0.97。所有数据集中的句子都经过了预处理，去除了音频前后段的空白部分，以及对文本进行了标准化处理（例如将“9”转写成“nine”）。

本章的实验均采用了多 GPU（2 个 2080Ti，11G）并行训练，并且均开启了半精度浮点数（FP16）加速训练。本实验的训练过程包括使用 BERT 作为编码器对特征预测网络进行了单独训练，以及在相同训练集上单独训练 WaveGlow 来合成高质量的音频。

为了训练特征预测网络，选择在单个 GPU 上以 batchsize 大小为 64 执行了标准的极大似然训练程序（在训练过程中，解码器传递给下一时刻的不是预测值而是真实值）。并且冻结了在大数据集上预训练好的 BERT 模型参数，使其作为一个特征提取器从输入文本中提取上下文信息。在 BERT 后接一个 Post-Net 模块，它是包含一个隐藏层的全连接网络层。隐藏层的隐藏单元数目与输入单元的维度一样，为 512。该层采用 ReLu（Rectified Linear Units）激活函数，并使用 0.9 的 dropout 进行正则化处理。实验使用 Adam^[39]作为优化器，其中 $\beta_1 = 0.9$ ， $\beta_2 = 0.999$ ， $\varepsilon = 10^{-8}$ ，并且将学习率固定为 10^{-3} 。本章的实验中还应用了权重为 10^{-6} 的 L_2 正则化。

在训练 WaveGlow 时，使用原始音频的 Mel 频谱图作为 WaveGlow 网络的输入。对于 WaveGlow，使用带有 librosa mel 过滤器默认设置的 80 个 bin 的 Mel 频谱图，即每个 bin 通过过滤器长度进行了归一化，并且刻度与 HTK 相同。Mel 频谱图的参数是 FFT 大小为 1024，跳数为 256 和窗口大小为 1024。

3.2.3 特征预测情况

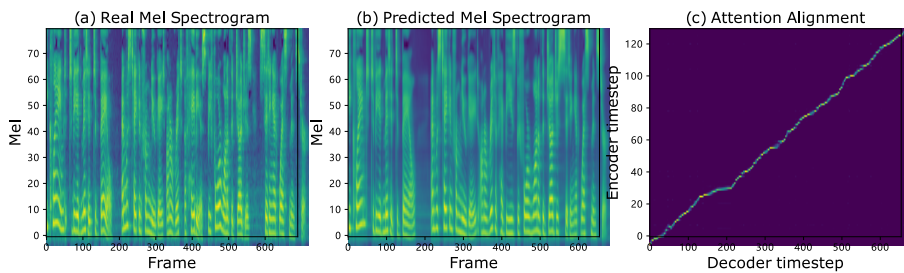


图 3.2 特征预测情况

Fig.3.2 The case of feature prediction

作为示例，图 3.4(a)、(b)给出了句子

He claimed to be admitted to bail, and was taken from Newgate on a writ of habeas
before one of the judges sitting at Westminster.

的真实的 Mel 声谱图和预测的 Mel 声谱图。通过观察可以看到合成的 Mel 声谱图和原音频十分接近，此外经过训练，模型也学习到了清晰平滑的特征网络对齐（如图 5(c)所示），这表明该模型在处理频谱图细节方面表现的很好。

3.2.4 评估

本文从内部数据集中选取了 10 句（非训练集中）具有不同长度的样本作为固定的评估集。在保证文本内容一致性并排除其他干扰因素的前提下，对不同模型不同训练阶段生成的这 10 个句子（包括真人录音）评估 MOS 得分。详细结果如表 1 所示。

表 3.1 实验结果 MOS 评分
Table 3.1 Result of MOS Score

Epochs	System	MOS	CMOS
500	Tacotron2	3.22	-
	Our Model	3.39	-
1000	Tacotron2	3.90	0.00
	Our Model	3.90	0.02
-	Ground Truth	4.54	-

从表 3.1 中可以看出，在迭代了 500 次的时候本章提出模型合成的音频效果明显优于 Tacotron2，这主要是由于本章的模型的收敛速度比 Tacotron2 更快。而在相同实验环境下迭代了 1000 个 epochs 后本章的模型获得了跟 Tacotron2 相同的 MOS 评分。因此，进一步采用相对评价意见打分 CMOS（The Comparison Mean Option Score）来对比 Tacotron2 模型和本章的模型的性能。在比较 CMOS 的测试实验中，参与人员每次听到 2 种音频（分别由本章提出的模型和 Tacotron2 合成），并使用[-3, 3]中的整数打分来评估前后音频的主观差距。可以看到本章的模型以 0.02 的差距获得了更高的评分。

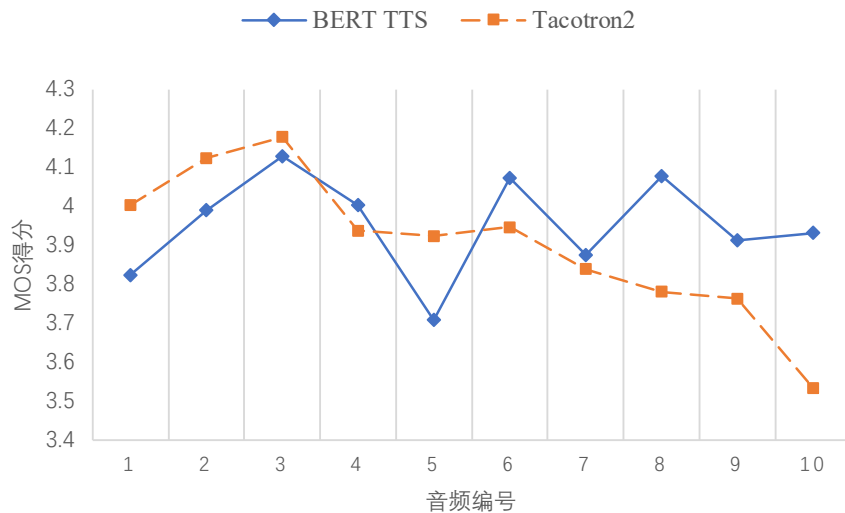


图 3.3 不同长度的语音合成样本 MOS 比较

Fig.3.3 Comparison of MOS for speech sample in different lengths

同时，为了分析输入文本的长度对模型性能的影响，本节对比了不同长度文本的音频合成结果的 MOS 得分情况。本节首先将上文选取的 10 个句子按长度递增编号为 1-10，每种长度的句子的 MOS 得分如图 3.5 所示。由于 BERT 最长支持 512 维的输入（不足 512 维进行补 0 操作），不存在 RNN 那样的长距离信息丢失的问题，在长音频合成中，本章的模型效果要明显优于 Tacotron2。从图 3.5 的结果中可以看到，当句子较长时，随着句子长度的增加，Tacotron2 模型的合成效果会变差，而本章提出的模型 BERT TTS 则没有这个问题。

3.2.5 训练时间对比

在训练过程中，通过插桩的方式记录了模型训练过程中每个训练 step 的耗时，以及每间隔 4 个训练 step 计算一次训练损失。在本节的实验环境下（2 个 GPU，FP16 启用，batch-size 为 64），新提出的模型的单个训练 step 耗时均值约为 0.35s，比在相同实验环境下的 Tacotron2（约为 0.65s）快了近一倍。

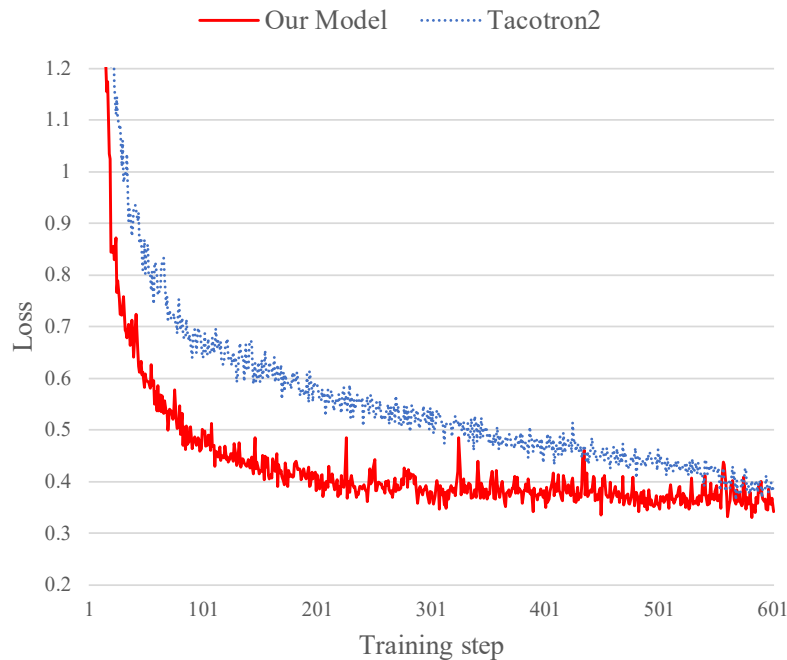


图 3.4 训练时损失变化

Fig.3.4 Changes in Training Loss

本文记录了训练过程中的损失函数，变化情况如图 3.6 所示，多次的试验结果表明，2 个模型的损失函数最终都会收敛到 0.3 附近。从图中可以看出，本章的模型比 Tacotron2 收敛的更快，也就是说本章提出的模型使用更少的训练步骤就可以得到效果较好的音频结果，这大大节约了第四章实验的耗时。

本章采用 BERT 作为编码器，它内部的 Self-Attention 结构可以并行的计算编码器的输出，因此可以节省大量的训练时间。并且，由于是基于预训练好的 BERT 微调来适配下游任务的，本章模型的收敛速度比 Tacotron2 更快。

3.3 本章小结

第三章提出了一个基于 Tacotron2 和 BERT 的端到端语音合成模型，他能合成高质量的英文音频。主要包括对该模型编码器、解码器以及对 2 个线性预测器的描述。本章提出的模型能充分利用 GPU 的并行计算能力从而获得更快的训练速度和预测速度。并且，它能从输入序列中获取远距离信息，使其在长文本语音合成中效果比 Tacotron2 等模型更好。同时，针对本章提出的模型进行实验评估，并且对实验过程收集的数据进行分析，也介绍了本实验所使用的实验设置、数据集以及模型评估方法等，并对实验结果进行了分析。实验结果表明，本章提出的模型能够在得到与 Tacotron2 模型相近效果的基础上，把训练速度提升一倍左右。此外，在长文本语音合成的场景下，本章提出的模型效果要明显优于 Tacotron2。

第四章 基于 BERT TTS 的端到端情感语音合成方法

4.1 基于 BERT TTS 的端到端情感语音合成

在情感语音合成领域，数据可用性是个问题。开源的情感数据集很少，且大多数数据来自不同的发言人，导致可用来训练的数据十分短缺。且语音合成所需的具有情感内容的高质量语音数据集很难收集，或是收集成本非常高。因此，与深度学习算法需要收敛的数据相比，情感语音可用数据量相对有限。解决数据量问题的有前途的方法是与知识转移相关的方法，例如转移学习，微调和多任务学习。在 TTS 领域中，许多学者也正在研究转移学习。在[42]中，他们成功地将知识从经过训练以区分说话者的模型转移到多说话者的 TTS 模型。这些例子激发了本文研究模型之间知识转移性的兴趣。

本章的目标是在使得模型合成中性语音的情况下，利用它来解决深度学习所需的数据量的内在问题，这是一个日益引起人们关注的话题。可以引用[43]中的无监督学习技术来更改带有样式标记的合成句子的韵律，或引用[23]修改 Tacotron 的体系结构以合成给定情感标签的情感语音。在本章中，通过在预训练好的基于深度学习的 BERT TTS 模型（第三章介绍的中性语音合成模型）进行微调，将中性 TTS 训练成可以合成情感语音的 TTS 系统。其整体架构如图 4.1 所示。

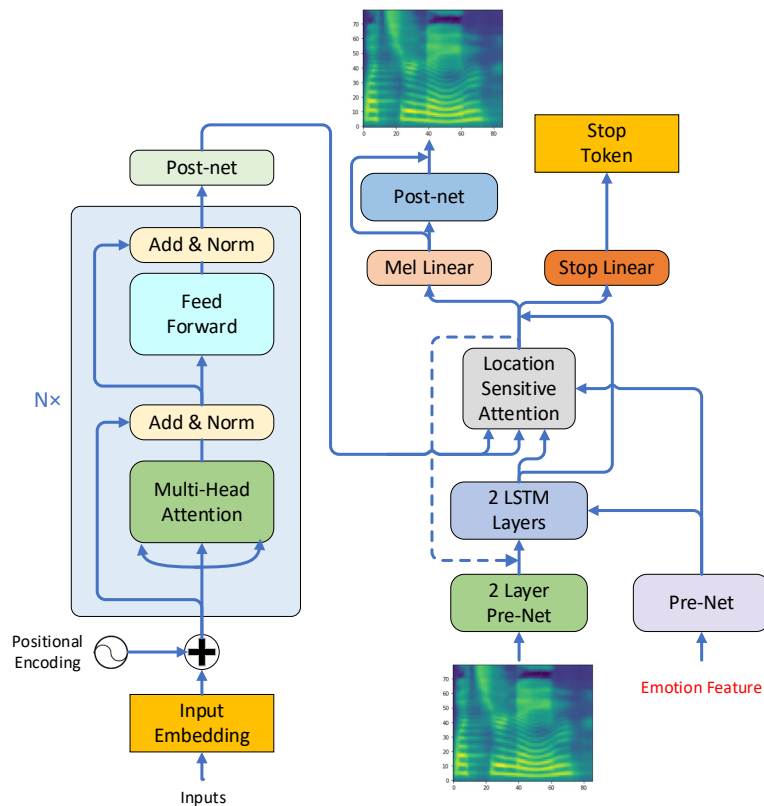


图 4.1 基于 BERT TTS 的情感语音合成模型架构

Fig.4.1 Model Architecture of Emotional speech synthesis base on BERT TTS

作为一个 Seq2seq 模型, BERT TTS 包含三个部分: 1) 编码器, 用于从输入文本中提取特征; 2) 基于注意力的解码器, 用于从输入文本的参与部分中生成梅尔谱图帧; 以及 3) 声码器-处理器合成语音波形。本章提出情绪化的 BERT TTS 来生成带有特殊规范(例如情感或个性)的语音, 以便模型可以在合成语音中具有变化。本章通过注入学习到的情感嵌入来实现情感 TTS:

$$h_t^{att} = \text{AttentionRNN}(x, h_{t-1}^{att}, e) \quad \& \quad h_t^{dec} = \text{DecoderRNN}(c_t, h_{t-1}^{dec}, e) \quad (4.1)$$

其中 x_t , c_t , h_t^{att} 以及 h_t^{dec} 分别是输入、注意应用上下文向量、注意 RNN 的隐藏状态和解码器 RNN 在时间步长的隐藏状态, e 是对应情感类型的代表向量, 本章提出了一个不同于传统的基于均值的情感特征向量表示法。

4.1.1 情感特征表示方法

文献[44]中的作者曾指出属于同一情感类别的具有语音信号韵律的风格嵌入矢量位置很近。因此, 如果来自同一样式层的特征向量源自相同的情感类别, 则它们往往会形成聚类。并且, 本章使用 t-SNE^[45]工具验证了该想法。先将提取各音频的梅尔频谱帧, 每个音频对应的梅尔频谱帧的维度为 $80 \times n$, 总的数据集中 n 的最大值为 2014。所以本章将梅尔频谱 80 维上每一维都 padding (后补 0) 成 2014, 再展平成维度为 1×16112 的数据, 最后调用 sklearn 库里的 TSNE 方法得到降维后的数据, 并绘制成图 4.2 所示分布图

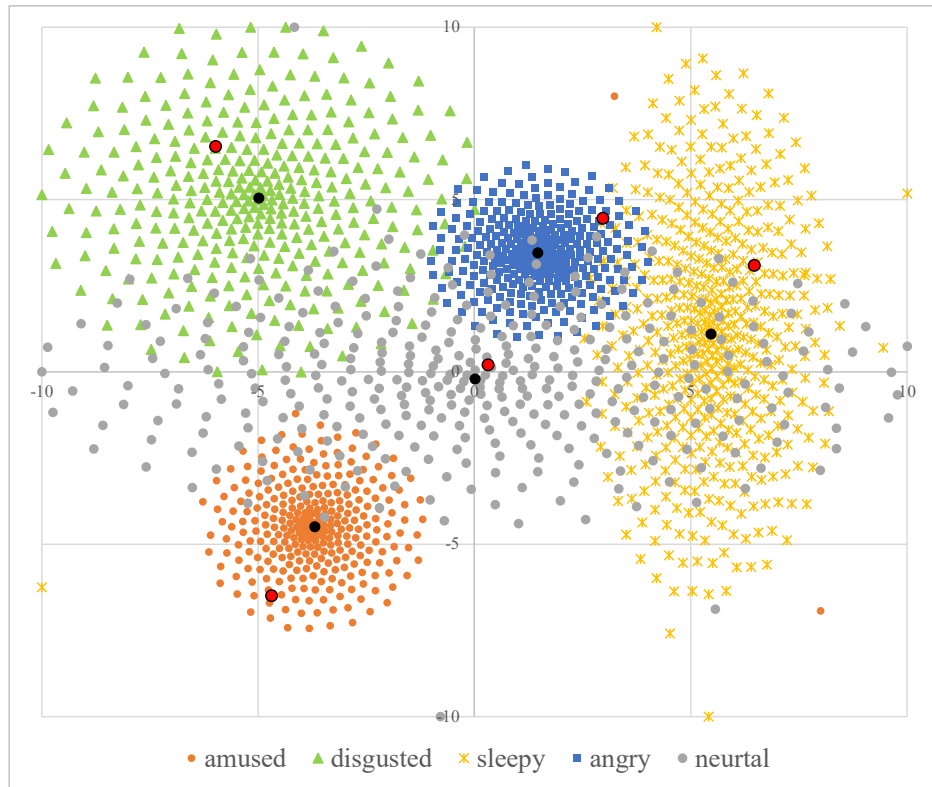


图 4.2 经 t-SNE 降维后的样本分布情况

Fig.4.2 Distribution of samples after dimensionality reduction by t-SNE

图 4.2 显示了 t 分布的随机邻域嵌入 (t-SNE)，可视化了来自 80 维样式标记权重向量的 2D 表示，这些向量根据其情感标签形成簇。从图中可以看出，除了中性语音与其他几种情绪的特征数据有交集，其他情绪形成的聚类分布都比较明显。为了从这些簇中合成情感言语，应该选择单个的特征向量来代表相应的情感类型。在本节中，提出一种方法来有效地确定每个情感类别的代表性权重向量，来清楚地呈现出不同情感独特的情感特征。

代表性特征向量应满足两个要求。首先，他们应该在不降低质量的前提下合成所需的情感言语。其次，代表权向量必须位于每个情感类别的边界之内，否则当模型的代币权重超出范围时，合成模型将无法在插值过程中生成语音。常用的是取均值：

$$r_e = \frac{1}{N_e} \sum_{x \in X_e} x \quad (4.2)$$

其中 N_e 和 x_i 分别是目标情感类别 e 的权重数和权重向量样本。但是，由于基于均值的方法没有考虑方差之类的权重分布，因此无法完全表示目标情感的信息。这在训练过程中也是不利的，因为他没有考虑周围其他情绪类别的分布，可能导致 2 个情感的代表向量距离很近，降低训练效果。本章决定采用各向量间距离比值来确定代表向量：

$$r_e = \frac{1}{2} \arg \max_r \frac{\sum_{x \in X_f} (r - x)^2}{\sum_{x \in X_e} (r - x)^2} + \frac{1}{2} \arg \max_r \frac{\sum_{x \in X_c} (r - x)^2}{\sum_{x \in X_e} (r - x)^2} \quad (4.3)$$

其中 X_e 表示目标情感的集群， X_f, X_c 分别表示距离目标情感最远和最近的情感集群。这样，情绪 e 的代表向量与距离最远的情绪和最近的情绪的距离最大化，而同一情绪内部的距离则最小化。这样可以使得情绪 e 的代表向量很好的和其他情绪区分开。并且，在计算得到 r_e 之后，本章会在当前情绪已有的样本中选择一个距离 r_e 最近的那个作为特征向量，避免了特征向量出现在聚簇范围之外的情况发生。图 4.2 中，黑点代表的是使用均值得到的向量，红点代表的是使用本章提出的方法得到的代表向量。从图中可以看出，使用本章的公式计算出来的代表向量比使用均值得到的向量的间距更大。

4.1.2 基于中性 TTS 模型微调

本章的目的是研究在少量情感语音数据集上对在大型数据集上进行预训练的中性 TTS 系统进行微调的可行性，并分析该模型能够适应这些条件的程度。本节将介绍整个微调流程，图 4.3 表示其总体思想。

该项工作中使用的数据集是 EmoV-DB(实验部分会详细介绍), 其中包含 5 种情绪的句子(中性、愉悦、生气、厌恶、慵懒)以及多个说话人。本章选择了一组语调接近 LJSpeech 中发言人的数据来微调本章的模型。在训练好的中性模型 BERT TTS 的基础上, 本章先使用其中的中性情感的音频通过微调来训练一个中性的 BTTS 模型, 使得模型熟悉该发言人的语调。这样会得到一个适应了当前说话人声音的中性 TTS 模型。再在中性 TTS 模型的基础上, 使用同一个说话人的其他情感的数据集来微调训练得到其他情感语音合成模型。

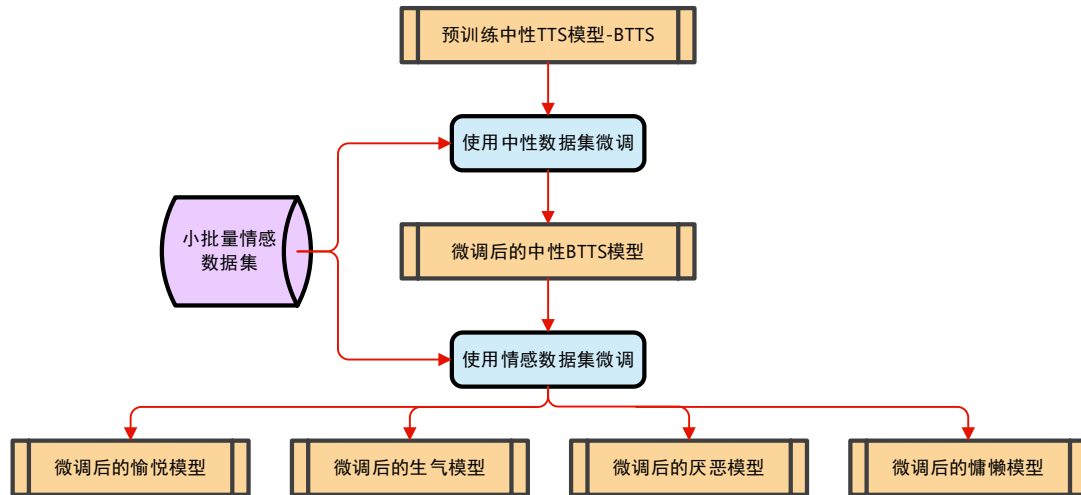


图 4.3 基于中性 TTS 系统的微调方式

Fig.4.3 Fine-tuning method based on neutral TTS system

在本节中, 将通过在小型数据集上微调预训练模型的一部分来说明如何在 TTS 上利用知识转移。首先, 要确定与新数据集进行微调的预训练模型的哪些部分需要改动, 以及要保持固定的部分。模型的第一部分 BERT 从输入文本中提取上下文信息。因此, 它不应该依赖于说话者身份的说话风格, 因为它只是经过训练从文本到上下文信息的映射。但是, 由于模型已经在 LJSpeech 数据集上进行了预训练, 因此可能会过度拟合该特定说话人的特性。本章要解决的第一个问题是确定 BERT TTS 是否可以将映射泛化为其他语音样式。由于文本不取决于讲话者的特征或讲话风格, 因此决定尝试仅训练音频部分。然而, 实验过程中发现合成的语音中存在一些节奏上的问题。经过多次试验表明这是因为注意力模块不适用于新说话人风格。因此, 选择对整个模块进行微调, 最后得到的模型能合成清晰的情感语音。

4.2 实验与结果分析

4.2.1 数据集

该项工作中使用的数据集是 EmoV-DB, 该数据库可在线获取。EmoV-DB 包含男女演员用英语说出的句子和法语中男性演员说出的句子。在东北大学校园的两个消声室中记录了英语演员, 而在蒙斯大学的消声室中记录了法国演员。每个演员都被要求说出 CMU-Arctic^[46]数据库中的句子子集。讲英语的人和来自 SIWIS^[47]数

据库的讲法语的人。演员们用 5 种情感类别讲出了这些句子，从而可以构建合成和语音转换系统。对于每个演讲者，在不同的会话中记录了不同的情绪。在这项工作中，实验中使用的是一位与 LJSpeech 数据集演员声音相似的英语女演员 bea 的数据集来对中性 TTS 系统进行情绪适应。

4.2.1.1 数据预处理

使用此模型进行预处理的重要方面是-采样频率-修剪音频文件的开头和结尾处的静音删除非语言表达（笑声，打哈欠等）。首尾的静噪修剪很重要，因为该模型使用了位置敏感注意力机制。通过假设字符的顺序与音频文件中的时间几乎线性相关，可以帮助注意力机制对齐。仅当语音文件的开头和结尾没有空音频时，这才是正确的。当使用未修剪静噪的数据集来训练模型的时候，得到的模型通常会丢掉前面的第一个单词，或者无法正确的预测结束帧。当音频文件中存在非语言表达（例如笑声，打哈欠或叹息声）时，也会发生相同的问题。实际上，在这些情况下，使用位置敏感注意的假设也没有得到验证。为了克服这个问题，首先为 Amused 数据集和 Sleepy 数据集手动筛选没有此类非语言表达的话语。然后仅针对 Amused 数据集，通过手动从剩余话语的一部分中删除笑声来添加到训练集中，得到的愉悦数据集总数为 238 句。最终结果如表 4.1 所示。可用来训练的情感语音数据时长都是以分钟为单位，最长的 Sleepy 数据集总时长为 36 分钟，而 LJSpeech 中中性音频数据的总长度约为 24 小时。

表 4.1 情感语音数据集组成

Table 4.1 Data set composition of Emotional speech

	Total duration [min]	Number of utterances
Amused	15	238
Angry	19	304
Disgusted	29	303
Sleepy	36	361
Neutral	23	357

4.2.2 实验设置

本实验数据集所使用的语音信号采样率为 44100 Hz，不同于 LJSpeech 数据集的 22050 Hz。采样位 16bit，使用海明窗处理，帧长 50ms，帧移 12.5ms，预加重系数 0.97。采用了多 GPU（2 个 2080Ti，11G）并行训练，并且均使用了半精度浮点数（FP16）加速训练。为了训练情感特征预测网络，实验中选择在单个 GPU 上以 batchsize 大小为 64 执行标准的极大似然训练程序。不同于 BERT TTS 模型的训练，实验只将 BERT 最后一层的参数加入训练，并且整个编码器部分的参数都参加微调，以期望训练得到一个合适的特征预测网络。本实验仍然使用 Adam 优化器，其中 $\beta_1 = 0.9$ ， $\beta_2 = 0.999$ ， $\varepsilon = 10^{-8}$ ，并且将学习率固定为 10^{-5} 。本实验还应用权重为 10^{-6}

的 L_2 正则化。

4.2.3 注意力对齐

实验过程中，在预训练好的 BERT TTS 模型上通过微调来适应中性情感语音时，该模型出现了生成语音的问题。在中性数据集上训练模型后，注意对齐在中间部分显示出不规则性。基于[23]中的结论，注意力对准的锐度与所生成语音的质量之间存在相关性，决定改进注意力对准的预测，并关注 BERT TTS 中的信息流。有两种信息来源可用于预测注意对齐。一种来自隐藏的关注状态 LSTM，另一种来自编码器 BERT。基于这两个来源，提出以下想法来改善对齐方式。

原始 BERT TTS 模型中的注意力部分通常会在多个解码器时间步中参与文本输入的相似部分，因为一个字符的发音通常需要一帧以上的声谱图。即使模型应更改注意力权重，下一个要关注的部分也将是当前关注文本的相邻部分。因此，在确定注意力权重时，该模型可以受益于具有先前参与的文本向量 c_{t-1} 的信息，该信息是文本编码的加权和。但是，原始的 BERT TTS 并没有利用这些信息。注意 RNN 仅采用前一时间步长的频谱图，几乎不包含信息 c_{t-1} 。基于此思想，将 c_{t-1} 与注意力 RNN 的输入 x_t 相连。现在，注意力 RNN 又输入一个 c_{t-1} ，如下所示：

$$h_t^{att} = \text{AttentionRNN}(x_t, h_{t-1}^{att}, c_{t-1}) \quad (4.4)$$

4.2.4 情感表达

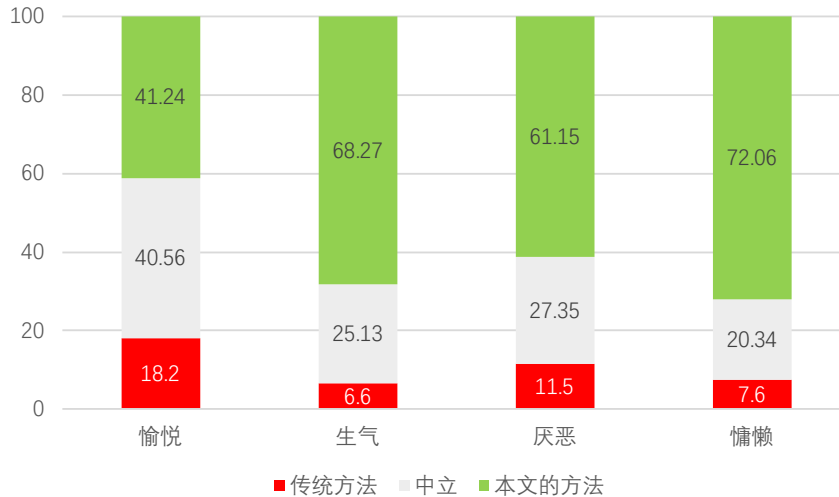


图 4.4 情感表达偏好测试结果

Fig.4.4 Results of the Emotional Expression Preference Test

本节首先进行了偏好测试，将本章提出的基于周围情绪的情感表达与传统的基于均值的方法进行比较。将分别按上述两种方法合成的相同文本以及情绪的音频

放在一起,要求参与者选择更能表达文本情绪的音频。如果参与者觉得它们之间没有情感上的差异,则可保持中立。在实验中,使用 10 个验证集中的句子来表达每个情感。测试结果如图 4.4 所示,在合成语音时,除幸福感类别外,本章所提出的方法均优于传统方法。由于模型记录到的幸福情绪信号具有与中性信号相似的韵律,因此两种方法之间的差异并不明显。结果证实,与传统的基于均值的方法相比,本章提出的方法有效地提取了每个情感类别的代表权重向量。

4.2.5 语音合成质量评估

为了测量合成语音的质量,在传统的基于均值的方法与本章提出的方法之间进行了主观均数评分(MOS)实验。在主观测试中,志愿者听完每个模型合成的结果,会对他们所听到的音频的自然度进行 5 分制打分(分值越大越好),打分间隔为 0.5,对按不同方法训练出来的模型合成的样本进行测试。同样,使用 10 个非训练集中的文本来合成不同情感标签的音频来参加测试。

得分结果如表 4.2 所示,在各情感下本章提出的方法合成的音频 MOS 得分均高于传统的基于均值的代表向量表示法,总体获得了 3.77 分。但是在基于情感数据集微调后模型合成的语音质量有所降低,略低于开始的中性模型的 3.9 分。主观猜测是因为中性模型已经拟合了之前说话人的说话风格,用来微调的数据集规模较小,无法让模型学习到合适的参数,使得合成的语音整体质量减低。但是,表 4.2 中给出的结果证明了本章的提出的情感特征表示方法的优越性,并且也说明了本章提出的使用中性 TTS 系统基于小数据集来微调来合成情感语音的方案是可行的。

表 4.2 各情感类型实验结果 MOS 评分

Table 4.2 MOS score of experimental results of various emotion types

Emotion	Conventional	Proposed
Amused	3.76	3.81
Angry	3.52	3.63
Disgusted	3.77	3.89
Sleepy	3.67	3.74
Total	3.68	3.77

4.3 本章小结

本章先简要描述了情感语音合成当下存在数据集规模小的问题,随后提出了一种基于中性 TTS 系统在小批量情感数据集上通过微调来合成情感语音的方法。基于第三章提出中性语音合成音频 BERT TTS 模型微调能合成高质量的情感语音。为了进一步提高情绪表达的清晰度,本章也提出了一种高效的情感代表特征表示。该特征表示法基于内部间情感比,同时也考虑了情感样本内部和类别间样式权重之间的嵌入距离。通过志愿者偏好实验证明了本章提出的情感向量表示法优于基

于传统的基于均值的情感向量表示法，并且也设计 MOS 评分实验验证了本章提出的情感语音模型能合成清晰的语音。在 MOS 打分实验中获得了总体 3.77 的评分，略低于 BERT TTS 的 3.90 分。同时，这也验证了基于中性 TTS 系统在小批量数据集上微调来合成情感语音的方法是可行的。

第五章 总结与展望

5.1 总结

作为人机语音交互的出口，语音合成的效果直接影响到人机交互的体验。一个高质量的、稳定的语音合成系统能够让机器更加地拟人化，使人机交互过程更加自然。本文针对情感语音合成面临的已有数据集规模小的问题展开研究，在提出了一个中性语音合成方法的基础上，将其进一步微调来进行情感语音的合成，主要工作如下：

(1) 针对基于 RNN 的神经网络语音合成模型训练和预测效率低下以及长距离信息丢失的问题，提出了一个基于 BERT 的端到端的语音合成模型 BERT TTS，该模型能合成高质量的英语音频。并且，该模型使用预训练的 BERT 作为编码器，在提高训练速度的同时能有效解决 RNN 那样长距离信息丢失问题。

(2) 针对不同情感的代表特征向量的选择问题，提出了一种基于情感数据集内部各情感的样本向量间距离的方法。该方法在考虑同一情感数据样本分布的同时，也考虑了该情感数据样本与其他情感数据样本的距离。实验证明该方法优于基于均值的特征向量表示法。

(3) 针对情感语音数据集规模小的问题，提出了一种基于中性 TTS 通过在小批量情感语音数据集上微调来合成情感语音的方法。通过实验验证了本文的情感语音模型效果，在 MOS 评分测试中总体获得了 3.77 分。

本文提出的模型可以用于智能对话系统，如个人助理、情感陪聊机器人、聊天机器人等。

5.2 展望

本文虽然使用 Bert 作为该 TTS 模型的 Encoder，但是 Decoder 仍然是自回归的，采用文献[48]中提出的 Transformer 作为编码器和解码器是后期的研究内容之一。此外，本文提出情感语音模型合成的情感强度是固定的，采用文献[44]中提出的插值方式来动态改变情感强度是另一个研究方向。最后，希望能够进一步改进当前模型使得它可以更好的契合中文语音合成任务。

参考文献

- [1] Taylor P. Text-to-speech synthesis[M]. New York: Cambridge University Press, 2009: 0-626.
- [2] Fung P., Schultz T. Multilingual spoken language processing[J]. IEEE Signal Processing Magazine, 2008, 25(3): 89-97.
井晓阳, 罗飞, 王亚棋. 汉语语音合成技术综述[J]. 计算机科学, 2012 (S3): 386-390.
- [3] Hunt A. J, Black A. W. Unit selection in a concatenative speech synthesis system using a large speech database[C]// Proceedings of the 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings. Piscataway: IEEE, 1996: 373-376.
- [4] Tokuda K., Yoshimura T., Masuko T., et al. Speech parameter generation algorithms for HMM-based speech synthesis[C]// Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. Piscataway: IEEE, 2000: 1315-1318.
- [5] Zen H., Tokuda K., Black A. W. Statistical parametric speech synthesis[C]// Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07. Piscataway: IEEE, 2007: 1229-1232.
- [6] Tokuda., Nankaku Y., Toda T., et al. Speech synthesis based on hidden Markov models[J]. The Institute of Electrical and Electronics Engineers, Piscataway: IEEE, 2013, 101(5): 1234-1252.
- [7] Zhang B., Quan C Q., Ren J F. Overview of Speech Synthesis in Development and Methods[J]. Journal of Chinese Computer System, 2016, 37(1): 186-192.
张斌, 全昌勤, 任福继. 语音合成方法和发展综述[J]. 小型微型计算机系统, 2016, 37(1): 186-192.
- [8] Jing X. Y, Luo F., Wang Y. Q. Overview of the Chinese Voice Synthesis Technique[J]. Computer Science, 2012 (S3): 386-390.
- [9] Arik S. Ö., Chrzanowski M., Coates A., et al. Deep voice: Real-time neural text-to-speech[C]// Proceedings of the International Conference on Machine Learning. PMLR, 2017: 195-204.
- [10] Wang Y., Skerry-Ryan R. J., Stanton D , et al. Tacotron: Towards End-to-End Speech Synthesis[C]// Proceedings of the Interspeech 2017. 2017.
- [11] Shen J., Pang R., Weiss R. J, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions[C]// Proceedings of the 2018 IEEE International

- Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE, 2018: 4779-4783.
- [12] Griffin D., Lim J. S. Signal estimation from modified short-time Fourier transform[J]. Proceedings of the 1984 IEEE Transactions on Acoustics Speech and Signal Processing, 1984, 32(2): 236-243.
- [13] Oord A., Dieleman S., Zen H., et al. WaveNet: A Generative Model for Raw Audio[C]// Proceedings of the 9th ISCA Speech Synthesis Workshop. 125-125.
- [14] Chrorowski J. K., Bahdanau D., Serdyuk D., et al. Attention-based models for speech recognition[J]. Advances in Neural Information Processing Systems, 2015, 28: 577-585.
- [15] Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Advances in Computer Science, 2014.
- [16] Sutskever I., Vinyals O., Le Q. V. Sequence to sequence learning with neural networks[J]. Advances in Neural Information Processing Systems, 2014, 27: 3104-3112.
- [17] Gibiansky A., Arik S. Ö., Damos G. F., et al. Deep Voice 2: Multi-Speaker Neural Text-to-Speech[C]// Advances in Neural Information Processing Systems, NIPS. 2017.
- [18] Ping W., Peng K., Gibiansky A, et al. Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning[C]// Advances in International Conference on Learning Representations. 2018.
- [19] Devlin J., Chang M. W., Lee K., et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 4171-4186.
- [20] Prenger R., Valle R., Catanzaro B. Waveglow: A flow-based generative network for speech synthesis[C]// Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE, 2019: 3617-3621.
- [21] Kingma D. P., Dhariwal P. Glow: Generative flow with invertible 1x1 convolutions[C]// Proceedings of the Advances in 2018 Neural Information Processing Systems. United states: NIPS, 2018: 10215-10224.
- [22] Hsu C. Y., Chen C. P. Speaker-dependent model interpolation for statistical emotional speech synthesis[J]. Proceedings of the EURASIP Journal on Audio, Speech, and

- Music Processing, 2012, 2012(1): 1-10.
- [23]Lee Y., Lee S Y., Rabiee A. Emotional End-to-End Neural Speech Synthesizer[C]// Proceedings of the 2017 Neural Information Processing Systems Foundation, NIPS 2017.
- [24]Wang Y., Stanton D., Zhang Y., et al. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis[C]// Proceedings of the International Conference on Machine Learning. PMLR, 2018: 5180-5189.
- [25]Zhang Y. J., Pan S., He L., et al. Learning latent representations for style control and transfer in end-to-end speech synthesis[C]// Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 6945-6949.
- [26]Lee Y., Kim T. Robust and fine-grained prosody control of end-to-end speech synthesis[C]// Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 5911-5915.
- [27]Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors[J]. Proceedings of the nature, 1986, 323(6088): 533-536.
- [28]Hochreiter S., Schmidhuber J. Long short-term memory[J]. Proceedings of the Neural computation, 1997, 9(8): 1735-1780.
- [29]Doya K. Bifurcations of recurrent neural networks in gradient descent learning[J]. Proceedings of the IEEE Transactions on neural networks, 1993, 1(75): 164.
- [30]Bengio Y., Simard P., Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. Proceedings of the IEEE transactions on neural networks, 1994, 5(2): 157-166.
- [31]Pascanu R., Gulcehre C., Cho K., et al. How to Construct Deep Recurrent Neural Networks[J]. Proceedings of the Computer Science, 2013.
- [32]Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Proceedings of the Computer Science, 2014.
- [33]Kočiský T., Hermann K. M., Blunsom P. Learning Bilingual Word Representations by Marginalizing Alignments[C]// Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014: 224-229.
- [34]Klementiev A., Titov I., Bhattarai B. Inducing crosslingual distributed representations of words[C]// Proceedings of COLING 2012. 2012: 1459-1474.
- [35]Qiu X., Sun T., Xu Y., et al. Pre-trained models for natural language processing: A survey[J]. Proceedings of the Science China Technological Sciences, 2020: 1-26.
- [36]Hao Y., Dong L., Wei F., et al. Visualizing and Understanding the Effectiveness of

- BERT[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019: 4134-4143.
- [37]Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need[C]// Proceedings of the Advances in 2017 Neural Information Processing Systems. United states: NIPS, 2017: 5998-6008.
- [38]Morise M., Yokomori F., Ozawa K. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications[J]. Proceedings of the IEICE TRANSACTIONS on Information and Systems, 2016, 99(7): 1877-1884.
- [39]Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization[J]. Proceedings of the Computer Science, 2014.
- [40]Cui Y., Che W., Liu T., et al. Revisiting Pre-Trained Models for Chinese Natural Language Processing[C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings. 2020: 657-668.
- [41]Waibel A., Hanazawa T., Hinton G., et al. Phoneme recognition using time-delay neural networks[J]. Proceedings of the IEEE transactions on acoustics, speech, and signal processing, 1989, 37(3): 328-339.
- [42]Jia Y., Zhang Y., Weiss R. J., et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018: 4485-4495.
- [43]Shen J., Pang R., Weiss R. J., et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions[C]// Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 4779-4783.
- [44]Um S. Y., Oh S., Byun K., et al. Emotional speech synthesis with rich and granularized control[C]// Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 7254-7258.
- [45]Maaten L., Hinton G. Visualizing data using t-SNE[J]. Journal of machine learning research, 2008, 9(11).
- [46]Kominek J., Black A. W. The CMU Arctic speech databases[C]// Proceedings of the Fifth ISCA workshop on speech synthesis. 2004.
- [47]Honnet P. E., Lazaridis A., Garner P. N., et al. The siwis french speech synthesis database? design and recording of a high quality french database for speech synthesis[R]. Proceedings of the Idiap, 2017.
- [48]Li N., Liu S., Liu Y., et al. Neural speech synthesis with transformer network[C]//

Proceedings of the 31st Annual Conference on Innovative Applications of Artificial Intelligence. PALO ALTO, CA: AAAI Press, 2019: 6706-6713.

攻读硕士学位期间的学术活动及成果情况

1) 参加的学术交流与科研项目

- (1) 服务机器人的情感认知与表达关键技术研究（编号：U1613217），国家自然科学基金联合资助基金资助项目，2017-2020

2) 参加的学术交流与科研项目

- (1) 安鑫 代子彪 李阳 孙晓 任福继. 基于 BERT 的端到端语音合成方法[J]. 计算机科学.
- (2) 安鑫 代子彪 李阳 孙晓. 基于深度学习的语音合成方法（申请号：202110430708.0）

特别声明

本学位论文是在我的导师指导下独立完成的。在研究生学习期间，我的导师要求我坚决抵制学术不端行为。在此，我郑重声明，本论文无任何学术不端行为，如果被查出有任何学术不端行为，一切责任完全由本人承担。

学位论文作者签名：代子彪

签字日期：2021 年 5 月 23 日