*《机器学习》课程项目*

*实施报告*

*题 目：* **模拟人声**

*组 号：*

*任课教师： 黄青华*

*组 长：*  20123915 朱天骋 20 %

*成 员：* 20124003 王逸格 20 %

*成 员：*  20124004 邓傲秋 20 %

*成 员：* 21123858 李晨曦 20 %

*成 员：*  20120783 卜宇扬 20 %

*联系方式：* 13621658460

*2023 年 2 月 26 日*

# 模拟人声

## 1 项目实施方案

### 设计目标

近年来，人工智能技术得到了飞速的发展，尤其是在自然语言处理领域的应用。语音变换技术作为自然语言处理的一个分支，可以将一段语音转换为另一个人的语音，目前已广泛应用于娱乐、语音助手、安全等领域，为人们带来了很多便利。在这种背景下，我们开发了这款能够变成其他人声音的课程项目，希望能够为大家提供一种新的语音沟通方式，同时也为AI技术在语音处理领域的应用做出贡献。对于本项目，我们希望可以达成的目标包括：

* 实现高质量的语音转换：在转换后的语音中保留语音中的内容和情感，同时使得语音听起来自然和流畅。
* 支持多人语音转换类型：用户可以选择不同的预训练模型，将输入音频转化成不同人的音频。
* 转换速度快：能够快速处理语音转换请求，以满足用户的需求。
* 可靠性高：保证语音转换的准确性和稳定性，防止出现不必要的错误和故障。
* 简洁美观的UI：为用户提供一个便捷的交互方式，提高用户体验。

### 设计思路

该项目的整体设计思路为：先收集目标人物的原音频，然后对原音频特征提取获得目标人物的音高音色音长等信息，再将这些信息送入神经网络模型得到属于目标的权重文件。当用户输入待转换音频后，我们同样对新音频进行特征提取，并用目标人物的特征信息进行替换合成，从而得到输出音频。最后我们需要为整个程序搭建一个简洁美观的UI界面。其中最关键的步骤毫无疑问便是提取特征和神经网络的选择。

### 1.3模型选择

通过搜集文献，我们首先将从音频中提取特征的方法锁定在了SOFT-VC[1]方法，这是一种。而在训练网络模型选择上，我们了解到目前主流的音频转换技术主要有：

* VITS[2]是一种并行端到端的语音合成方法，采用了变分推理，增加了归一化流和对抗性训练过程，从而提高了生成建模的表达能力。
* CycleGAN-VC[3]是一种基于循环一致性生成对抗网络（CycleGAN）的语音转换技术，通过构建两个生成器和两个判别器，同时学习两个方向的语音转换，有较强的鲁棒性和泛化能力。
* StarGAN-VC[4]与CycleGAN-VC类似，StarGAN-VC也采用了生成对抗网络（GAN）的方法来实现语音转换，但是它的网络结构和损失函数设计与CycleGAN-VC有所不同。
* DeepVoice3是一种基于编码解码的端到端语音转换技术，尽管其拟合效果和灵活性都比较好，但是需要大量的训练数据和计算资源。

最终，综合各模型的性能、训练数据要求、计算资源要求以及源码可读性等角度考虑，我们选择了SOFT-VC与VITS相结合的模型来实现本项目的功能需求。在后续的调研过程中，我们还发现了已经有人尝试对于两模型进行拼接，构建了一个基于vits与softvc的歌声音色转换模型[5]，尽管其效果并不理想的，所以本项目希望能在该模型基础之上，优化模型结构的同时，搭建一个高可用性的UI。

### 1.4实现方式

我们首先仔细阅读了SOFT-VC和VITS的文献，大致了解它们的基本原理，并通过学习已有模型进行了代码复现。我们发现其所有的训练超参数都被预先放在了config.json文件中，于是我们想能不能通过修改模型的通道数、卷积核个数大小形状、神经网络的层数和节点数等参数以优化变声器模型。

另外，我们选择了python下的gradio库来绘制UI界面。gradio是一个开源Python库，用于构建机器学习、数据科学演示和web应用程序，它可以围绕机器学习模型快速创建一个漂亮的用户界面。不仅如此，它还拥有丰富的交互接口，能完美满足我们输入音频的功能需求。

### 1.5关键技术

#### 1.5.1 SOFT-VC模型

在一个典型的语音转换系统中，目标是学习捕捉语言内容但丢弃说话者特定细节的功能。离散化提取（discretization）过程中也丢弃了一些语言内容——增加了转换语音中的发音错误。而软语音单元可以一定程度上解决这个问题，类似于计算机视觉中的软分配（soft-assignment in computer vision）训练一个网络来预测离散语音单元上的分布。通过对离散单元分配中的不确定性进行建模以达到保留更多的内容信息的目的，从而纠正错误的发音。

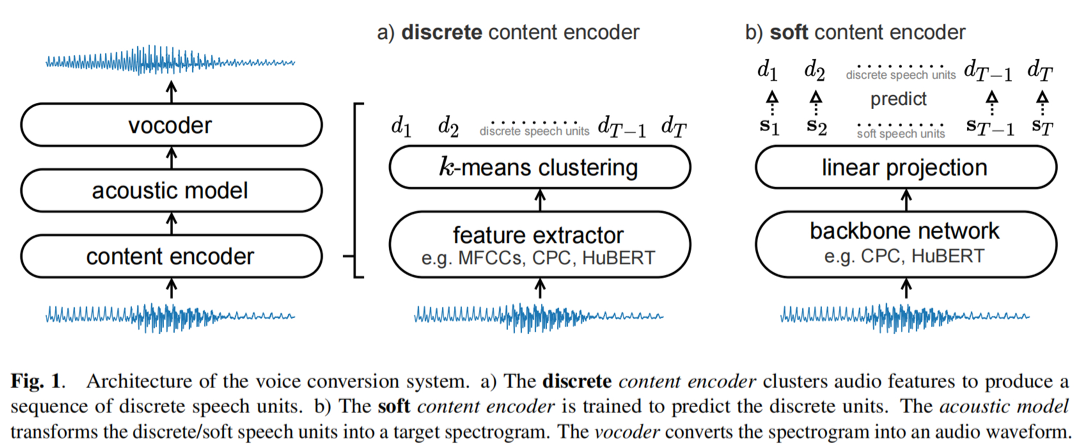
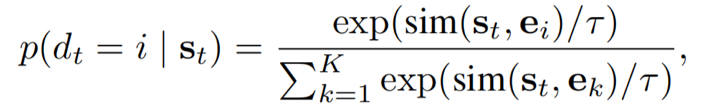


图1 声音转换系统结构

其中，Soft Content Encoder:对于软语音单元，很容易直接使用特征提取器的输出，而不需要聚类。其理念是，软语音单元提供了原始连续特征和离散单元之间的中间地带。一方面，离散的单位创造了一个信息瓶颈，迫使说话者的信息消失。因此，为了准确地预测离散单元，软内容编码器需要学习一个说话者独立的表示。另一方面，语音的空间并不是离散的。因此，离散化会导致一些内容信息的丢失。通过在离散单元上的分布进行建模，我们的目标是保持更多的内容信息并提高可理解性。上图的b中概述了软内容编码器的训练过程。给定一个输入话语，首先提取一系列离散的语音单元(d1, ..., dT)作为标签。接下来，一个主干网络（CPC或HuBERT）处理话语。然后，一个线性层投射输出以产生一系列的软语音单(s1, ..., sT) 。每个软单元参数化一个离散单元谱的分布：



其中，i是离散单元，ei是一个相应的可训练的嵌入向量，sim（s, e）为计算软单元和离散单元之间的余弦相似度,最小化分布和离散目标之间的平均交叉熵(d1, ..., dT)，以更新编码器。在测试时，软内容编码器将输入的音频映射到一系列的软语音单元(s1, ..., sT)，然后传递给声学模型。

#### 1.5.2 VITS模型

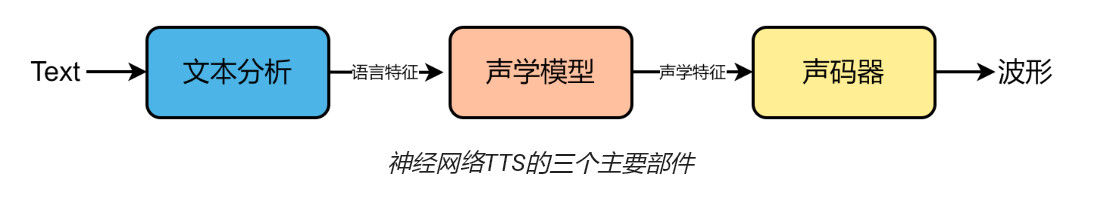


图2 神经网络TTS的三个主要部件

VITS（Variational Inference with adversarial learning for end-to-end Text-to-Speech）是一种语音合成方法(声学模型)，它使用预先训练好的语音编码器 (vocoder声码器) 将文本转化为语音。

VITS则采用VAE架构实现端到端的语音合成。为了进一步提高模型的表现力（expressive power），VITS将标准化流（normalizing flow）运用到条件先验分布（conditional prior distribution）上，另外引入对抗训练（adversarial training）。

除了要生成高质量的音频，TTS系统还要能做到“一到多”的合成效果，即一个人念多次相同的话会有不同的方式（音调、时长等）。为解决该问题，VITS中引入随机时长预测器（stochastic duration predictor）来生成具有不同rhythm的语音。

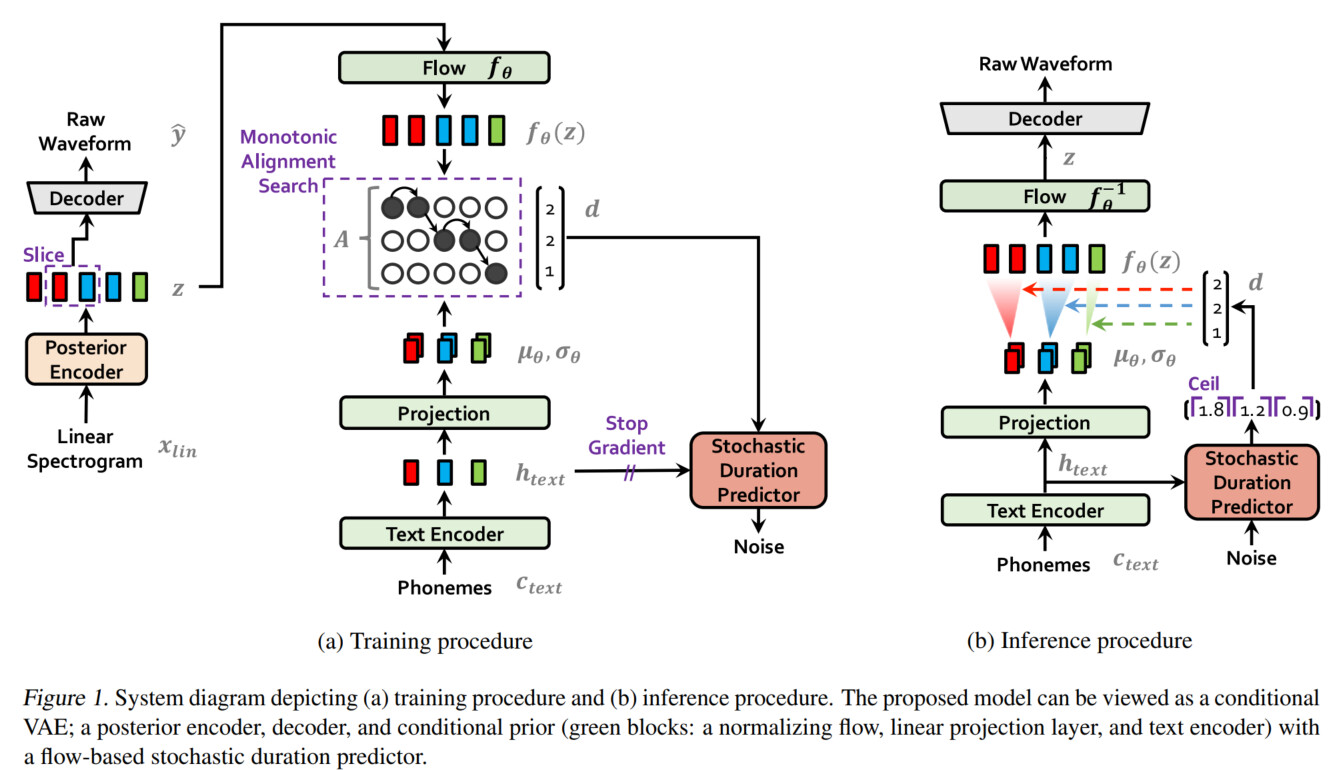


图3 VITS模型结构

VITS 的工作流程如下：

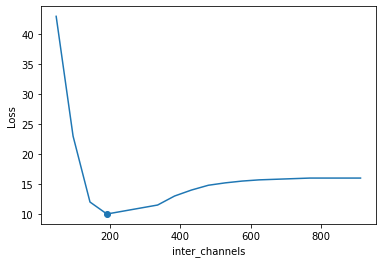
1. 将文本输入 VITS 系统，系统会将文本转化为发音规则。
2. 将发音规则输入预先训练好的语音编码器 (vocoder)，vocoder 会根据发音规则生成语音信号的特征表示。
3. 将语音信号的特征表示输入预先训练好的语音合成模型，语音合成模型会根据特征表示生成合成语音。

## 2 项目实施过程

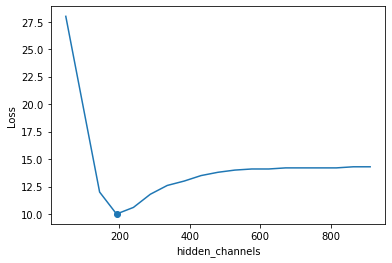
前期的文献调研和资料调查部分已经在前文介绍过就不详细展开了，这里主要介绍数据集准备、优化模型部分和前端制作。

数据集准备部分。我们首先获取了同一人总共5个视频共30分钟的音频源文件。受计算机硬件设备限制，我们需要先对音频文件进行切分，否则即使将batch\_size降到1，训练开销依然会超过GPU容量。于是，我们特地写了一个切分音频的程序，通过python的pydub库调取ffmpeg音频处理软件，将5个aac格式的音频源文件切分为了220个10s的wav格式的音频文件。为了满足训练要求，我们还对这些音频文件进行了重采样，并输入SOFT-VC模型提取特征，且这些特征信息被存储在了音频相应的npy和pt文件中

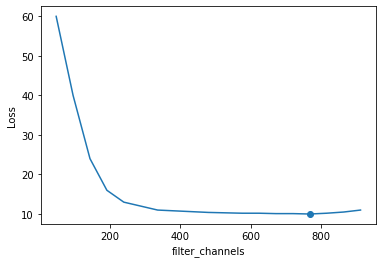
优化模型部分，由于每次训练都需要较久时间才能获得结果所以我们最后调整的超参数并不多。最后，本小组选定了中间层的通道数、隐藏层的通道数和滤波器通道数这三个超参数作为优化的入手点。对于每一个超参以48为步长，逐步获得训练结果并记录。调优的办法我们选择贪心算法，即每一步超参的选择都选择最优解，从而希望最终达到全局最优解的目的。我们在训练过程中对超参数与残差进行记录，并绘制了折现图展示了我们的调参结果，如图4所示。我们最终得到了超参数的最优解[192,192,768],此时训练模型仅为10.102，由此我们获得了最优的训练模型G\_20000.pth文件。



(a)



(b)



(c)

图4各个超参数与残差关系图

前端部分我们首先制作了前端的UI界面，通过自定义一个VitsGradio类封装gradio库的Block网页模块，输入参数以及交互控件。接着，我们需要为交互控件添加交互函数，而交互控件总共有三个，分别是：模型载入按钮，文件输入框和音频转换按钮。模型载入按钮的点击函数为从项目文件中读取config配置文件和预训练模型。文件输入框则可以在点击后打开系统文件夹以便用户上传需要变声的音频文件，用户也同样可以将文件直接拖入该框中进行输入。而音频转换按钮的点击函数就是去调用变声程序，直接输出结果音频。另外，转换的过程可能需要一定的时间，我们也同样设计了进度条用以表示的转换进度。

## 3 结果分析

首先对于音频转换结果进行分析。从主观角度分析，除了个别音节有噪声外，转换出来的音频能够完美的模仿变声目标的音色和发音习惯。我们为10名不同的同学播放了转化得来的音频，他们都能准确说出音频的变声目标，并且其中恰好有一半人没有分辨出这是合成音频而非变身目标本人的音频。从客观角度分析，我们节选了变声目标和另外一人所说的同一句话，如图5所示。尽管受到停顿、音长等影响，但是我们依然可以发现在变音后，其波形图明显与目标的音频波形图更为相似了。

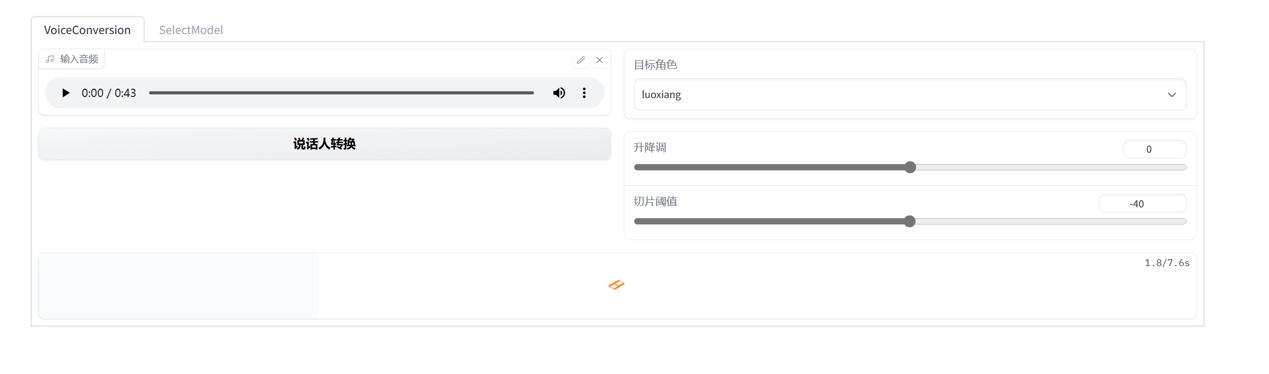


图5测试数据波形比较图

接着，我们也是顺利完成了UI界面的制作，如图3所示。其不仅满足了项目所需的基本共需求，也体现了简洁美观的特点。



(a)



(b)

图6 UI界面成果展示图

## 4 未来展望

尽管本项目满足了所有的项目功能需求，对于变声效果进行了优化，也完成了UI界面的制作，但是我认为依然有提升空间：

* 变声后音频存在部分噪音：后续选择不同滤波器对音频降噪，提高输出音频的质量。另外，也可以通过优化算法或者增加batch\_size的方式提升模型性能，来提升变声效果。
* 前端界面没有集成用户现场录音：可以继续学习gradio库，寻找录音的控件和函数，并拓展项目的功能。
* 音频转换效率有待提高：目前对于一分钟以内的音频进行转换需要10s左右时间，后续可以通过优化算法或者提升设备性能加快转换效率。

我们会在未来逐步对这些方面进行优化，从而完成一个令人满意能够模拟人物声音的程序。

## 参考文献

[1] B. van Niekerk, M. -A. Carbonneau, J. Zaïdi, M. Baas, H. Seuté and H. Kamper, "A Comparison of Discrete and Soft Speech Units for Improved Voice Conversion," ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 6562-6566, doi: 10.1109/ICASSP43922.2022.9746484.

[2]Kim, Jaehyeon, Jungil Kong and Juhee Son. “Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech.” *ArXiv* abs/2106.06103 (2021): n. pag.

[3]Kaneko, Takuhiro, H. Kameoka, Kou Tanaka and Nobukatsu Hojo. “Cyclegan-VC2: Improved Cyclegan-based Non-parallel Voice Conversion.” ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019): 6820-6824.

[4] Kameoka, H., Takuhiro Kaneko, Kou Tanaka and Nobukatsu Hojo. “StarGAN-VC: non-parallel many-to-many Voice Conversion Using Star Generative Adversarial Networks.” 2018 IEEE Spoken Language Technology Workshop (SLT) (2018): 266-273.