2019年05月22日10:26:09

数据集vctk，说话人数量4个

实验目的：做成few-shot语音转换

实验过程：模型架构参考nvdia的 **Few-Shot Unsupervised Image-to-Image Translation**（FUNIT）

基础模型为stargan-vc，在本代码基础上改动，包括数据预处理，模型，训练和损失函数

- [x] 修改模型，用测试数据确保模型能够正确输出
- [x] 确定损失函数，用测试数据能够训练
- [x] 确定特征维度是否合适
- [x] 训练出模型
- [ ] 评价转换结果
- [ ] 是否能够绘制t-SNE图
- [ ] 主观测试

实验评价：主观评价自然度，清晰度，可选的t-SNE图。

实验过程记录：

输入特征大小36x256，内容编码器和说话人编码器暂时是一样的，他们输出的特征在通道上连接起来，再输入解码器，在测试时，由于输入源和目标的频谱帧数不同，导致不能连接，故采用了每256帧转换一次的方案，问题出在说话人编码器，其也是输入多帧，怎么和内容编码器的输出连接了？暂时的实现，只是权宜之计，后期应该将说话人编码器换成真正的编码器，输出向量。

- [x] 讲说话人编码器换成输出固定长度向量（需要训练一个说话人向量系统）
- [x] 损失函数调整、

2019年05月23日09:58:48

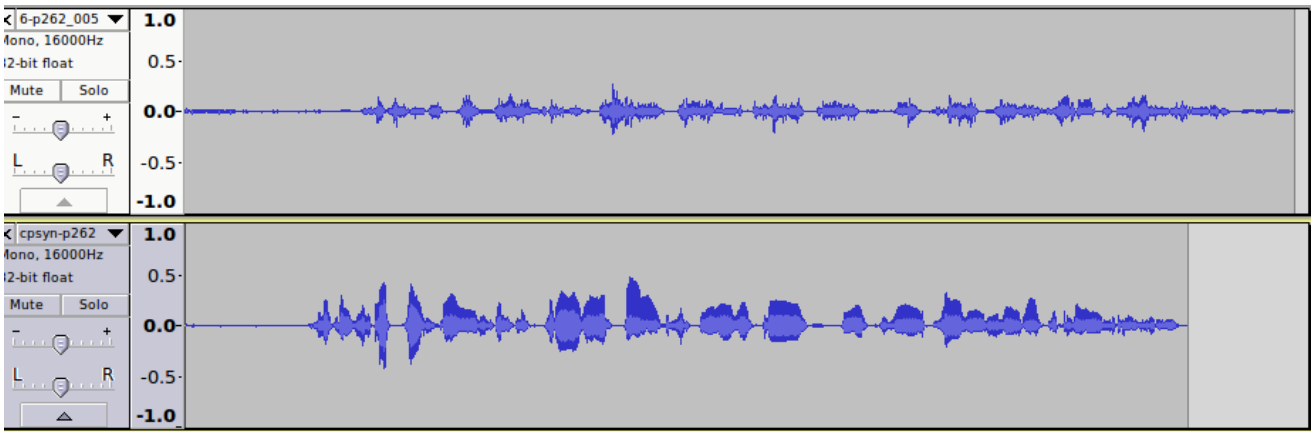
经过一夜的训练，模型并没有转换成功，输出和噪音一样，最有可能原因是：内容编码器（C）和说话人编码器（S）输出的都是一样的矩阵，即说话人个性特征没有成功编码并连接到C的输出。

本实验虽然想按照FUNIT论文上的结构来，但是代码还没有放出来，据论文中描述可知，FUNIT的基础代码是MUNIT，故现在参考MUNIT的做法，做以下改动：

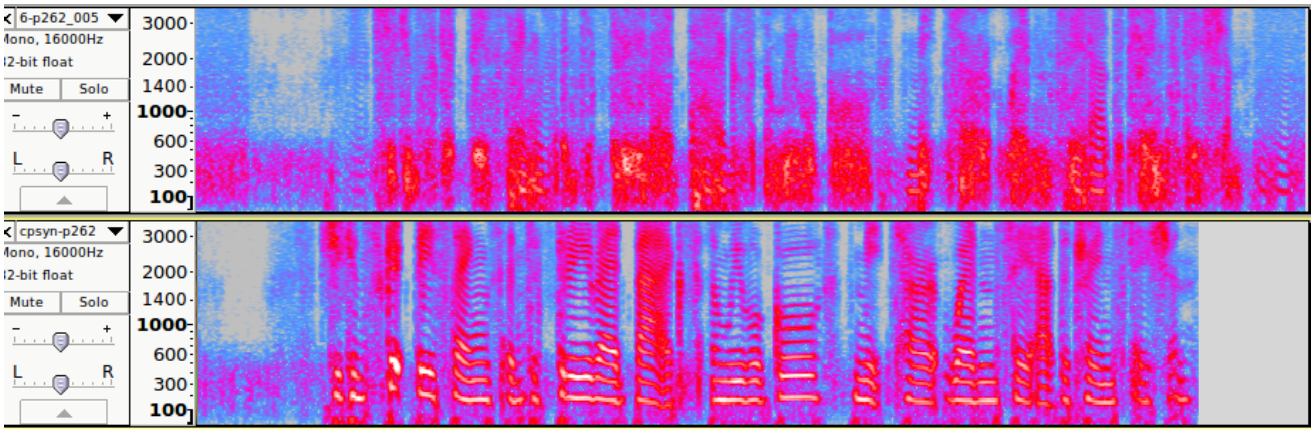
- [x] 将S换成MUNIT中的风格编码器，将输出的风格变成一个向量，AdaIN仿射
- [x] 修改Decoder中的激活函数为AdaIN

2019年05月24日09:55:24

下图是迭代训练300000次，模型转换结果和原始音频的波形图和mel频谱图。



在上面的波形图是转换后的，下面是原始波形，可见转换后的波形幅度小了很多，整体的形状差别较大。



从梅尔频谱可以看出，原始波形的共振峰很明显，而转换后的音频共振峰基本消失，频谱基本被噪音所覆盖。在测听时，可以听见说话内容，但是包含噪音，清晰度很差，自然度也很差。

经过这次改进模型，可以确定：该模型是能够进行训练和转换的。下面就要从各方面改进模型的效果了。

先从损失函数，优化方法考虑：

- [x] 分析原来的损失函数是否适用于该模型
- [x] 确定损失函数

2019年05月25日12:18:14

损失函数改为lsgan损失，如下：

判别器损失

```
# lsgan loss
r_out = self.D(mc_real)
mc_fake = self.G(mc_real)
f_out = self.D(mc_fake.detach())
d_loss = torch.mean((f_out - 0)**2) + torch.mean((r_out - 1)**2)

# Compute loss for gradient penalty.
alpha = torch.rand(mc_real.size(0), 1, 1, 1).to(self.device)
x_hat = (alpha * mc_real.data + (1 - alpha) * mc_fake.data).requires_grad_(True)
```

```
out_src = self.D(x_hat)
d_loss_gp = self.gradient_penalty(out_src, x_hat)

# Backward and optimize.
d_loss = d_loss + self.lambda_gp * d_loss_gp
```

生成器损失：

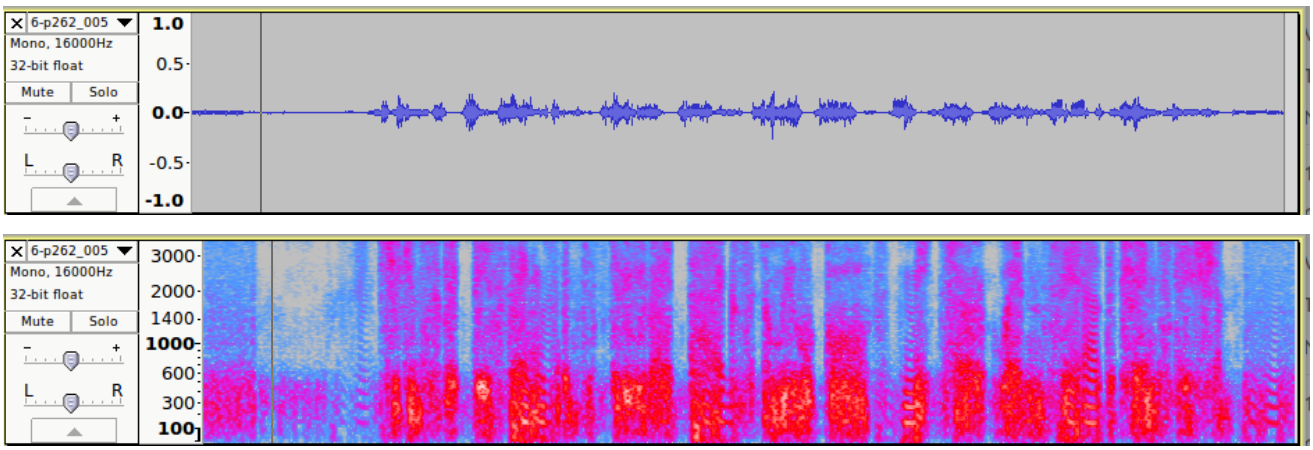
```
# lsgan loss
mc_fake = self.G(mc_real)
g_f_out = self.D(mc_fake)
g_loss_fake = torch.mean((g_f_out - 0)**2)

#reconstruction loss
g_loss_rec = torch.mean(torch.abs(mc_real - mc_fake))

# Backward and optimize.
g_loss = g_loss_fake + self.lambda_rec * g_loss_rec
```

其中lambda_rec=5， lambda_gp=5

去掉了以前适用于Cyclegan的损失，迭代训练500000轮，并且在卷积层之间放置了dropout2d，结果如下：



训练中判别器的损失稳定下降，最终在0.5左右波动，生成器的损失稳定下降，最终在1.5左右波动。比较本图和昨天的图，波形上大致相似，mel谱也十分相似，可知原来的损失函数和现在的损失函数都是可以使模型正常训练的，故模型的问题应该不在损失函数了。

为了更加确切的分析问题所在，对训练所得的模型进行专门的测试，方案：1.源和目标为同一个人；2.源和目标为一男一女。

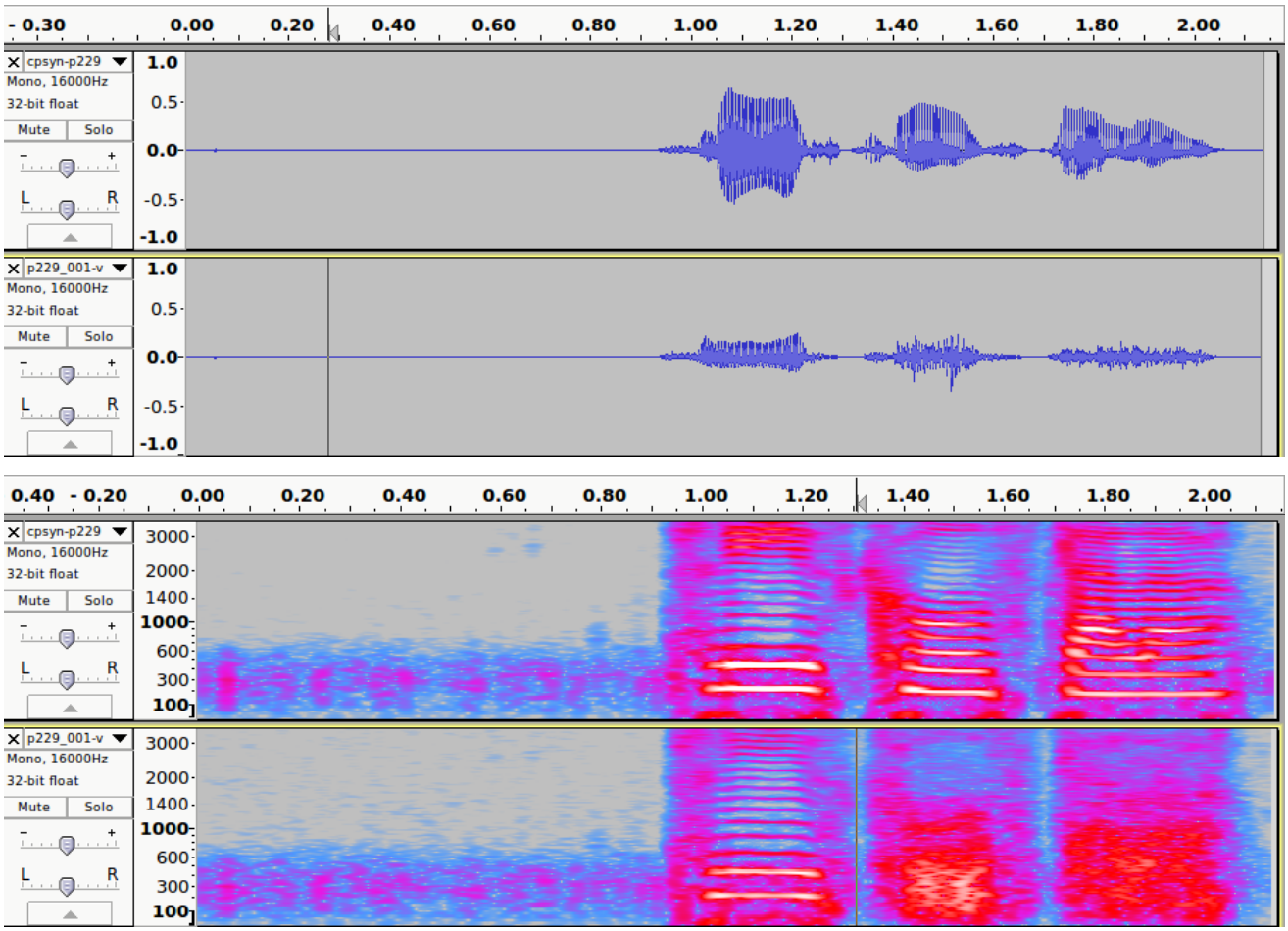
需要分析：

1. 模型能不能在方案1时正确的重建音频频谱和波形，音色不变；
2. 在方案2时模型能不能正确重建波形，并且转换成目标的音色；

下图设置为：

```
src = 'data/VCTK-Corpus/wav16/p229/p229_001.wav'
trg = 'data/VCTK-Corpus/wav16/p229/p229_001.wav'
```

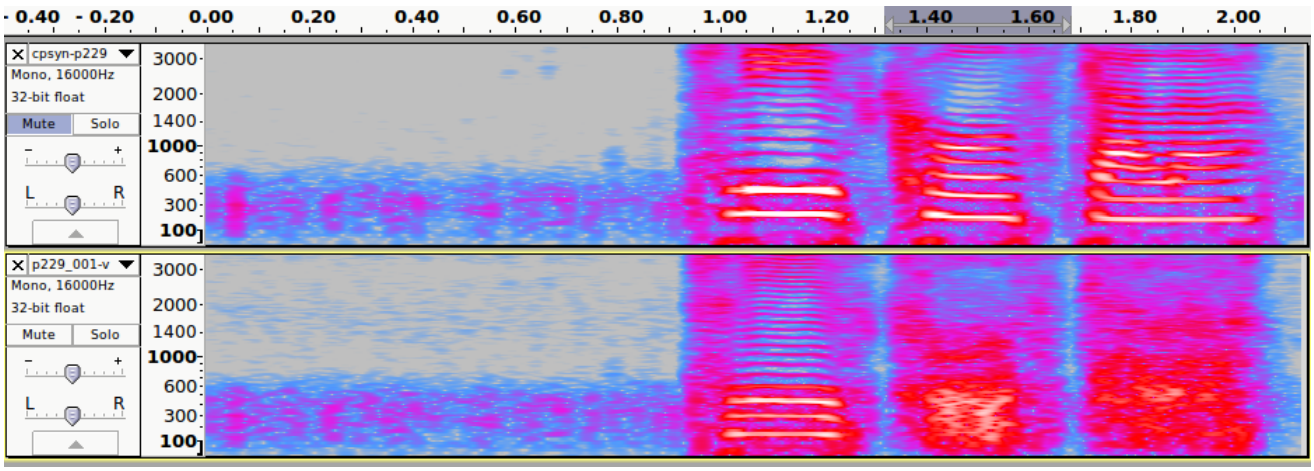
源和目标相同时，重建的波形如下图，1s-1.2s时，转换的波形相对正常的形状，频谱可以看到泛音成分，从1.4s-2.0s波形大不相似，泛音也被噪音所掩盖，听起来音色相似，但是噪音很大。



下图设置为：

```
src = 'data/VCTK-Corpus/wav16/p229/p229_001.wav'
trg = 'data/VCTK-Corpus/wav16/p232/p232_001.wav'
```

源和目标不相同时，听起来音色不太一样，噪声很大。



为什么**1s-1.2s**泛音分量较其他时间好？

2019年05月27日09:51:28

在训练一天多的时间里，出现：

```
Elapsed [1 day, 11:56:00], Iteration [1046600/2000000], D/loss: 51.9191, G/loss_fake:
108.9915, G/loss_rec: 25.3842, G/g_loss: 235.9123
Elapsed [1 day, 11:56:10], Iteration [1046700/2000000], D/loss: 400.9465, G/loss_fake:
528.5242, G/loss_rec: 87.6559, G/g_loss: 966.8038
Elapsed [1 day, 11:56:21], Iteration [1046800/2000000], D/loss: 1538.4707, G/loss_fake:
4283.6865, G/loss_rec: 95.8614, G/g_loss: 4762.9932
Elapsed [1 day, 11:56:31], Iteration [1046900/2000000], D/loss: 173.1725, G/loss_fake:
532.8336, G/loss_rec: 104.7104, G/g_loss: 1056.3857
Elapsed [1 day, 11:56:41], Iteration [1047000/2000000], D/loss: 263.6960, G/loss_fake:
743.8304, G/loss_rec: 93.6745, G/g_loss: 1212.2031
```

损失已经发散了。由于存在学习率衰减，这时候学习率已经很低了。

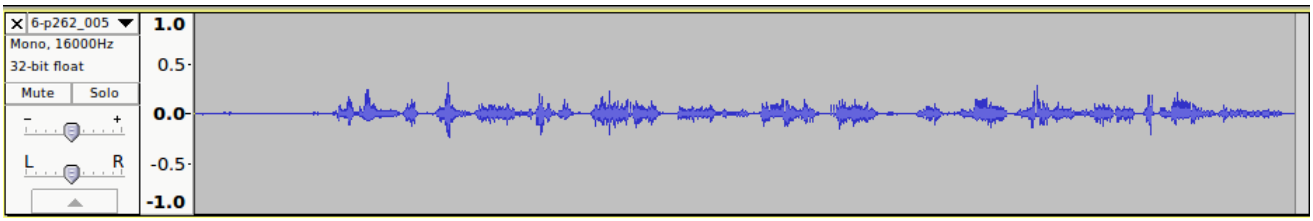
2019年05月28日09:46:41

调整了decoder的激活函数为selu，pad类型为reflect，训练到800000轮时没有出现损失发散情况。数据集扩大到10个人，转换后的语音依然丢失了说话人的个性特征，噪音很大。~~现在考虑能否去噪，方案是输入带噪声信号，输出频谱和纯净频谱做L1损失，带噪声信号可能是原来输出的，也可以时手工添加的噪声。~~

需要好好研究一下AdaIN变换那块原理。

2019年05月31日10:30:37

为了验证自己的网络结构不是导致声音带有噪声的原因，已经将生成器换成了MUNIT的生成器，训练后，输出结果如下



可见，结果和之前模型大致相同，声音依旧噪声很大，失去了说话人的个性特征。

2019年06月03日15:04:14

突然想到，说话人编码器没有训练，说话人编码器输出的特征直接结果一个MLP然后复制到解码器里，所以，根本就没有训练！这也解释了合成的音频丢失了说话人个性信息。所以，为了训练说换人编码器，现在有两个方案：

- [ ] 两阶段训练，先训练说话人编码器，在训练整个网络
- [ ] ~~联合训练，在训练GAN的过程中，逐步训练说话人编码器~~（代码改动较大，暂时放弃）

为了简单起见，先实施第一种方案，即两阶段训练方案。

2019年06月05日10:00:36

两阶段训练具体实施步骤：

参考论文**Generalized end-to-end loss for speaker verification**提出的GE2E损失和模型，使用world提取的MCC特征训练说话人编码器。

需要实施：

- [x] 预处理VCTK数据集，特征改为world提取的128维MCC特征
- [x] 训练说话人编码器
- [x] 将原说话人编码器换成已经训练好的新的编码器

2019年06月12日11:22:13

方向错误，应该是按照论文FUNIT的方案来实施，中途却改为了使用MUNIT的方案来实施，不管怎么做都是错的。

现在改正：首先彻底明白AdaIN的原理和实现（写博客），再实现FUNIT的网络结构，考虑借鉴架构，具体考虑语音特征特点来修改。