代码实践

- 阅读了解了代码的框架，熟悉代码执行的流程，输出所在的位置
- 搭建了三层全连接的DNN网络，进行50个epoch的训练

```
# 模型参数
INFO ] Model summary:
NET(
  (stft): ConvSTFT()
  (istft): ConviSTFT()
  (fc1): Linear(in_features=257, out_features=637, bias=True)
  (fc2): Linear(in_features=637, out_features=637, bias=True)
  (fc3): Linear(in_features=637, out_features=257, bias=True)
)

# 训练结果
2022-07-14 19:42:27
[/home/disk1/user2/mxy/HolidayWork/nnet/libs/trainer.py:247 -
INFO ] Loss(time/N, lr=1.250e-04) - Epoch 50: train =
-14.8638(0.27m/561) | dev = -14.4024(0.04m/89) | no impr, best
= -14.4028
2022-07-14 19:42:27
[/home/disk1/user2/mxy/HolidayWork/nnet/libs/trainer.py:260 -
INFO ] Training for 50/50 epoches done!
```

- 运行`compute_si_snr.py`，得到模型的测试结果

```
(base) user2@node7:/home/disk1/user2/mxy/HolidayWork$ ./test.sh ./sps_tas/spk1.scp ./data/tt/spk1.scp
100%|                                                                                | 824/824 [00:29<00:00, 27.99it/s]
SI-SDR(dB) Report:
NG: 824/17.541
```

阅读论文

Conv-TasNet: Surpassing Ideal Time - Frequency Magnitude Masking for Speech Separation

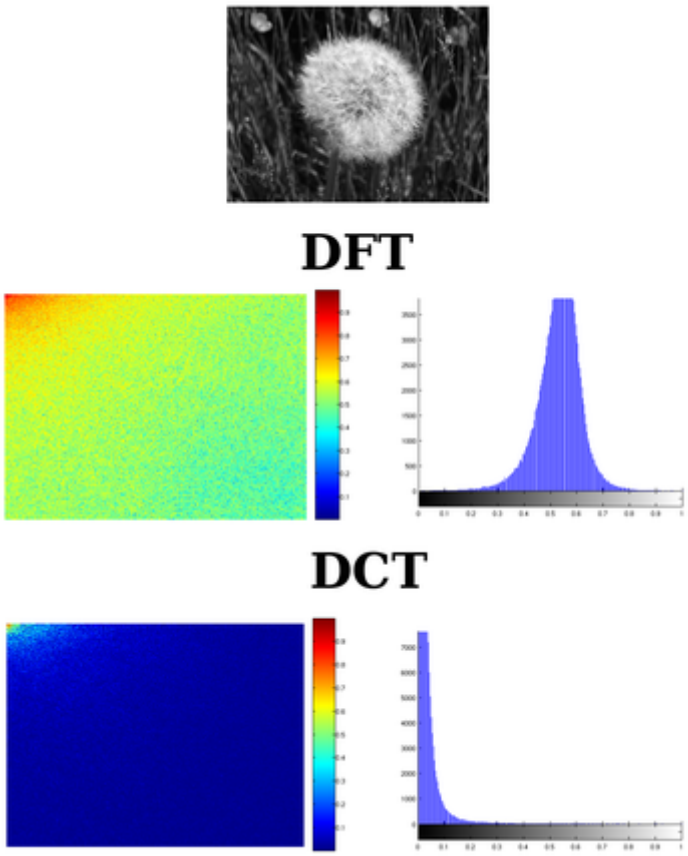
> 2019: Yi Luo and Nima Mesgrami

I. Introduction

1. 语言处理需要自动地进行语言分离，但分离新的说话人的准确率仍有所不足
2. short-time Fourier transform(STFT): 快速傅里叶变换

   short-time inverse Fourier transform(iSTFT): 快速反傅里叶变换

   discrete cosine transform(DCT): 离散余弦变换



3. 先前的方法：
   1. 将时域信号通过STFT转化到频域进行处理，其方法是通过干净的声音作为训练的数据集，使用非线性拟合去估计每个声音的频谱
   2. 另一种方法，通过一个权重mask乘到频谱中去恢复每个人的声音
   3. 缺陷：
      - STFT是一种普遍的信号转换方式，对于语言分离并不是最优的
      - 需要使用干净的数据来预先训练
      - 对声音的分辨率要求高，需要一个长的暂时窗去进行傅里叶变换，这个要求增加了系统的延迟。在大部分系统中这个延时为32ms，在音乐分离中这个延时更高
4. 如果只在时域进行处理，其效果与频域相比不够好

有人尝试将深度学习与时域处理想结合，其方法是：将使用STFT进行特征提取那一步换成使用学习去得到声音的特征表示，两个方法都是在提取特征信息，这种尝试即为TasNet

TasNet现在已经得到了和T-F系统差不多或者是更好的结果

5. time-domain audio separation network(TasNet)
6. Independent Component Analysis(ICA)
7. LSTM:
    1. 之前的TasNet中使用LSTM网络用来对声音信号进行分离，
    2. 缺陷：
        - 使用更小的核会增大输出的长度，使LSTM的训练过程难以进行

            > 输入的大小一直在变，模型的参数数量就也在变了，很难搞

        - LSTM中含有大量的参数，增加了计算代价
        - LSTM对时间的耗费比较长，当选择不同的起点时，导致了不稳定的分离准确率

            > 感觉Conv-TasNet主要解决了时间的问题，进行加窗虽然不知道是什么意思，但是之前的网络加窗需要的窗比较长，然后Conv-TasNet需要的窗就没那么长，随之而来的好处就是分离的准确率提高、计算代价减小等

8. stacked dilated convolution:

    是一种技巧，参照论文：https://openaccess.thecvf.com/content_CVPR_2019/papers/Schuster_SDC_-_Stacked_Dilated_Convolution_A_Unified_Descriptor_Network_for_CVPR_2019_paper.pdf

9. 使用卷积的另一个好处，可以进行**并行处理**，减小模型大小
10. depthwise separable convolution:

    参照论文：https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html

11. 论文的一般结构：

    The rest of the paper is organized as follows. We introduce the proposed Conv-TasNet in Section II, describe the experimental procedures in Section III, and show the experimental results and analysis in Section IV.

II. Convolutional Time-Domain Audio Separation Network

1. 分为三部分：encoder, separation, decoder

接下来计划

- 对网络模型的参数进行优化，并将每次的参数和运行结果记录在表格中
- 继续读论文，了解语音处理中的一些基本知识和所使用框架中用到的一些技巧