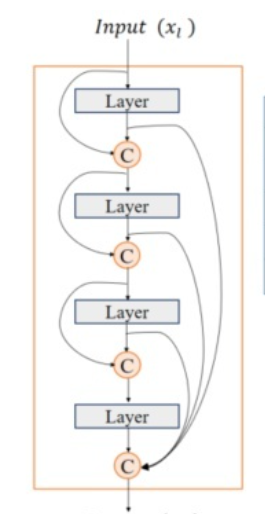


|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Domain | PESQ | STOI | SSNR | CSIG | CBAK | COVL | Para(maillion |
|  |  |  |  |  |  |  |  |  |
| TSTNN（改进前的模型） | T | *2.96* | *95* |  | 4.33 | 3.53 | 3.67 | 0.92 |
| 去掉前馈 |  | **3.02** | **95** |  | **4.33** | 3.55 | **3.68** | **0.92** |
| 三层的TSTM |  | **2.96** | 94 |  | 4.32 | 3.52 | **3.67** | **0.74** |
| 两层的TSTM |  | **2.96** | 94 |  | 4.32 | 3.52 | **3.67** | **0.74** |
| 残差密集TSTM |  | **2.96** | **95** |  | 4.32 | 3.52 | **3.67** | **0.74** |
|  |  |  |  |  |  |  |  |  |

轻量级残差密集架块：每个layer都是一个两级transformer模块（Two-stage tranfs block）

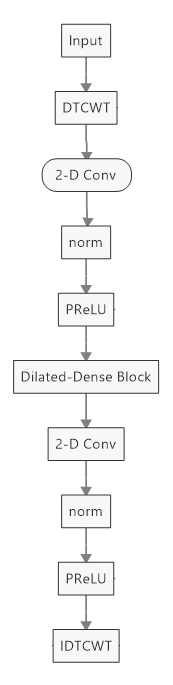
元学习



兄弟你好！

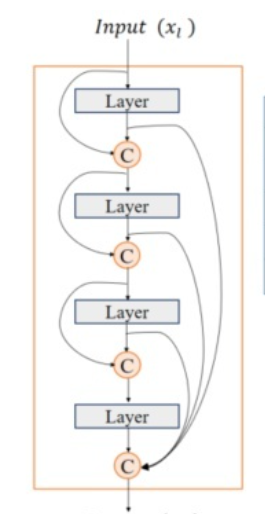
以下是新实验的一些**改动：**

第一点：在encoder中加入小波变换，我用的是DTCWT变换（你可以参见：https://pytorch-wavelets.readthedocs.io/en/latest/dtcwt.html），你可以跑一跑其他小波变换的实验。



对语音信号先进行双树复杂离散小波变换，设置小波变换系数为3，该变换有内置滤波器，提取语音信号的低频、中频信息，过滤掉带有高频信息的噪声序列。将低频、中频信息分别通过卷积层、激活函数等，在encoder最后通过双树复杂离散小波变换的逆变换实现语音信号的重建。

第二点：在encoder后面的TSTM模块中，在不同的四层两级transformer模块中使用轻量级残差连接块，下图是我的具体的实现，其中每个layer是一个两级的TSTM。



第三点：原作者用了四层TSTM，你可以跑一下两层、三层的实验，这些实验我都跑了，但是不是单独跑的，我跑的时候加上了小波变换跑的，具体的实验结果如本文档的第一个表格所示。**模型在老师的服务器上，但我已经登不上去了，所以新实验你就直接重头跑吧，本来我也没跑多少实验。**

第四点：将元学习加入模型中。

你每次跑实验的时候，一定要严格控制实验变量，就是比如你跑一个两层的TSTM的时候，你一定不要去加小波变换跑实验，而是单独的去跑两层的TSTM。消融实验一定要做全，比如三个点abc，你就要跑a、b、c、a+b、a+c、b+c、a+b+c，并且其他实验变量一定不能变。另外，这个模型训练框架有问题，bs最大设置为2，跑的时候需要22g的内存，不然跑不起来。