

朱老师，我给您汇报一下这周我的学习情况

1、复现了 DCCRN

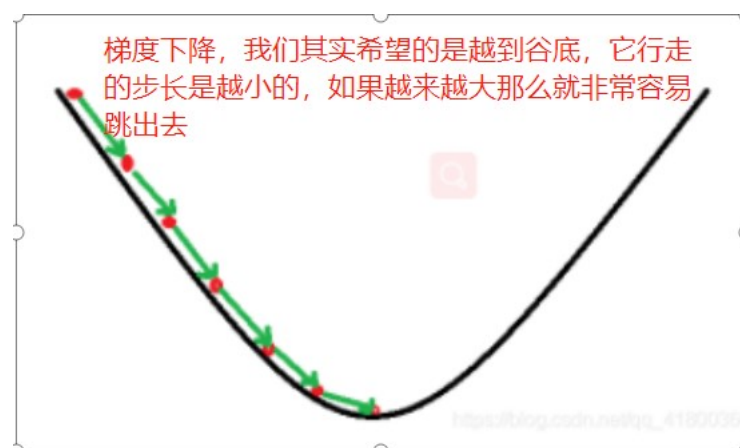
2、复现了两种 Complex BatchNormal 操作（DCCRN 用了其中的一种，但是我查阅资料发现有另一种计算思路，所以都实现了在两张卡上分开跑的，后期对比一下）

3、复现了 RTNet 模型

我终于明白那个 loss“不对劲”在哪儿了：准确来讲，那个 loss 设计上（从优化方向 and 我们的目的方向角度来讲是一直的，降低 loss 会让信噪比提升）。Loss 也是可以使用负值的。但是我再三考虑，想到了负值的不好的地方，以及这个 loss（SI-SNR）的缺点所在。

首先说为啥 loss 到负值就不好了：

按负值来讲，它可以无限小，但是在 loss 降低的同时它的绝对值是在不断扩大的。这里用一个比较俗套的图来讲一下：



所以这个 loss 可以用，但是其实对于模型的拟合来讲是不好的。目前我的改进思路是①要么去掉 log 并且用噪信比来作为 loss；②要么在训练过程中较大比例地减小学习率

1、然后针对第一个方案（去掉 log），可以从下面看到 clean-noise 情况下，loss 非常大 6321383，clean+noisy 的情况下就非常小了 0.01996。我想这也应该是为什么要加 log 的原因。。免得后面完全训练不动，所以这里我这两天在想，有没有能够避免掉负值，并且能够合理地调整两极分化情况的这么个函数（目前就想到了 sigmoid，不过因为这个函数将两级又过于降幅，估计效果也不好，还有待验证）

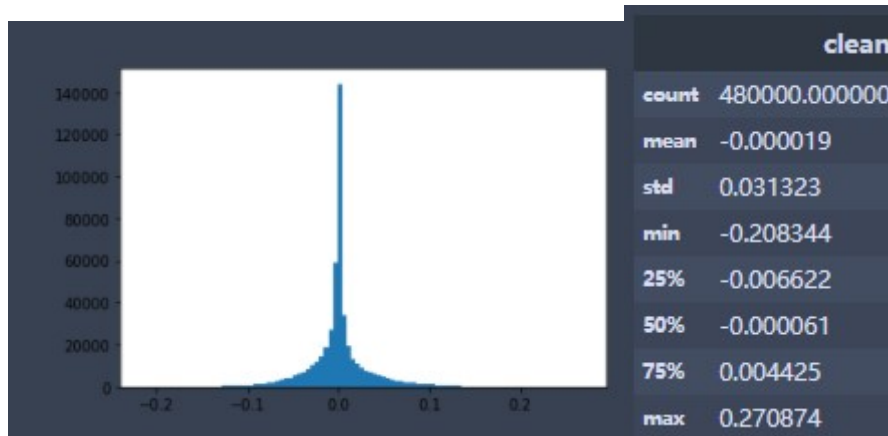
6321383.579542469	
49.940738254846295	clean+noise

0.019967700542900386	clean+noisy
-16.996715604146615	

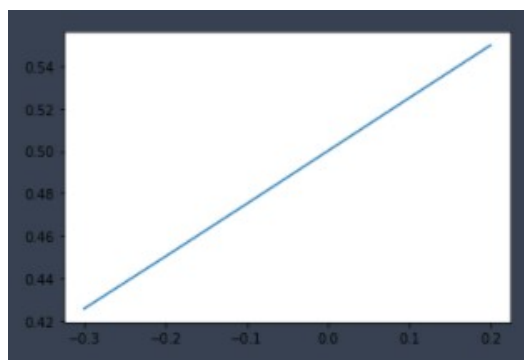
0.0	
-76.72977992558523	clean+clean

2、在训练过程中动态大幅调整学习率，这个得根据这几天跑模型的过程 loss 的变化情况后 续再做讨论

3、另一个想法就是上次吃饭的时候我给您说的：用熵的概念来定义（前面这些大部分都基于矩阵的思想来的）。目前是想了一个信息散度的情况。由于信息散度是建立在概率分布的前提下（有 log 操作，必须为正数），所以我这里想的是如何将一条语音信号（-0.3 左右 ~0.3 左右的脉冲信号）转换为一个有概率意义或者类似的一个数值，我想到的是 softmax 有一个概率转换的思想。但是我通过数据分布分析发现一个语音信号里面数据的分布大致是这样的



可以看到，大部分都是 0。绝对值 0.1 以上的值都非常少。如果用 softmax 的话，那么 0 值的概率就会非常大。。导致 0 的权重很大，如果用这个作为 loss，那么模型的注意力可能会被 0 值所吸引。所以需要有一个类似于一个线性的模型（不会改变数据原始的数据分布情况），然后我可视化了 sigmoid 的 -0.3~0.3 的部分



我觉得。。。这可能就是我要找的线吧。。能够将原始数据（有负数）合理地，几乎等比地映射到 0-1 之间。所以我就这个思路构造了一个损失函数（预测值相对于实际值的一个类似于信息散度的概念）

```
def sigmoid(x):
    return 1 / (1+np.exp(-x))
def relative_entropy(source, estimate):
    source = sigmoid(source)
    estimate = sigmoid(estimate)
    return np.sum(estimate*np.log(estimate/(source)))
```

```
379.89752
***** clean+noise
167.3202
***** clean+noisy
0.0
***** clean+clean
```

结果看起来也很 ok。

下周安排：

①将 DCCRN 模型训练好

②RTNet 训练好

③开始测试我上面的几个想法

④阅读文献，看看有没有其他测评思路