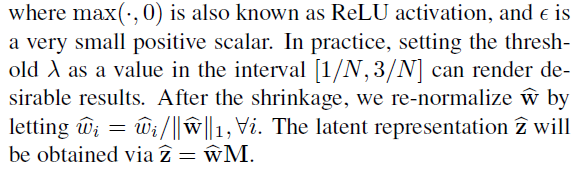
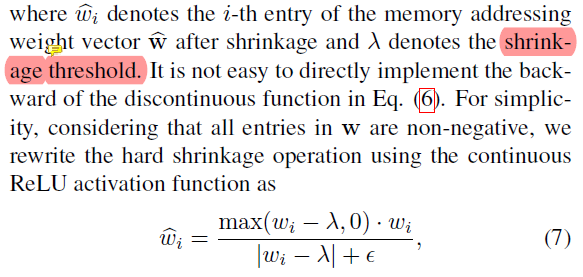
# 项目使用详细文档

## TIED

作为TIED的重要参数有2个，一个是memory\_size，另一个是shrink\_thres。

为了让memory模块能够被更少但是更相关的记忆项线性表示，从而获得更有价值的表征，我们对小权重项进行截断。如下图所示。该shrink threshold应当处于之间，我们直接取为2/N。（当然这是一种充满先验的取值方法）



因此在代码TIED\_ED.py中我们在18行类初始化时设置为self.shrink\_thres = 2/layer\_sizes[0]。

## 分析有可能从87掉到85的原因

### 融合方式

为方便理解与描述，我们设对于同一子段（Subsegment）的重构结果为预测结果为，其对应的真实数据（ground truth）分别为。是重构和预测的权重。

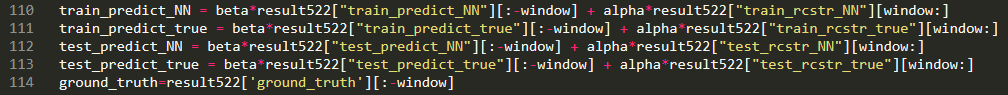
* 方式A: 直接将重构data和预测data相加融合，计算统一误差：
* 方式B：分别计算误差，再进行相加：
* 方式C：不融合·只计算预测误差
* 方式D：不融合·只计算重构误差

表 0-1

|  |
| --- |
| 值得注意的是，在此之上的加权和不加权又是新的taxonomy的维度，总共有8种可能，在详细这样总结之前，我忽略了将他们逐个实验的重要性和迫切性，耽误了一些时间。因此下面对我曾经做过的改动做一个回溯。 |

### 曾经做过的改动

**（1）2019.9-2020.1：方式A**

****在2019年9月-2020年1月，我们都使用cum.py对预测误差和重构误差进行融合，如下所示（from zyx服务器: /home/zyx/hyt/Code/cum.py）

**（2）2020.1-至今：方式B**

2020年1月，我在没有整理清楚以上formulation的情况下（或者说当时并不知道这会有很大的影响）做了一个造成后续混乱的改动。

Conditional\_test\_add\_pure\_merge\_interval\_try.py中我对wgt\_error\_out进行了一部分修改变为了方式B

但其问题在于：

我只对加权部分进行了这样的修改，从而导致不加权部分只使用了预测部分进行误差度量。（包括何雨婷在WADI上做的各项工作）

### 现在要做的事情

把这几种方式通过接口的方式进行定义并逐个实验，进行记录。

刚试了一下，也出现了之前那种只能搜到一个很好的值的问题，但是我非常清楚地记得在暑假并没有出现这样的事情。

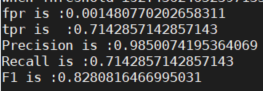
**未下采样的数据：**

不加权情况下：

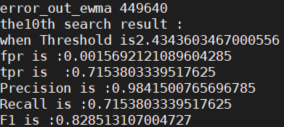
**D: Predict**



126 epoch



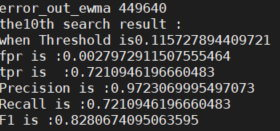
**A：**



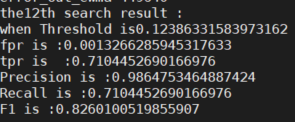
**B：**



**C：Reconstruct**



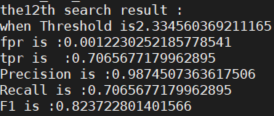
加权方式的A



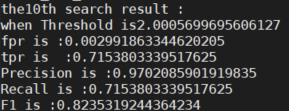
4种方式都试过了，都是相似的问题。

**下采样的数据：**

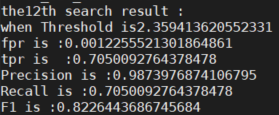
**A:**



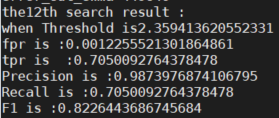
B:



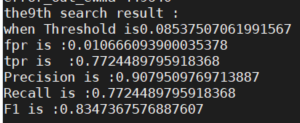
C：Reconstruct



D: Predict



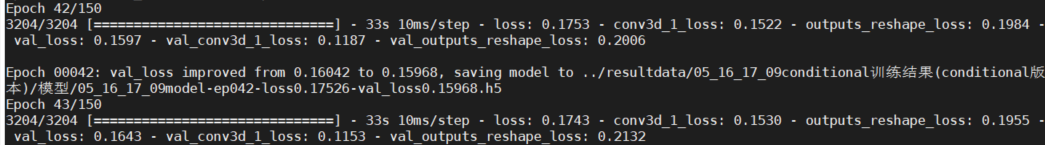
加权方式的C



连下采样也没有上次好了…这次相比上次做的改动：shrink\_thres（0.03->0.04）和固定的随机数种子

## 模型设计

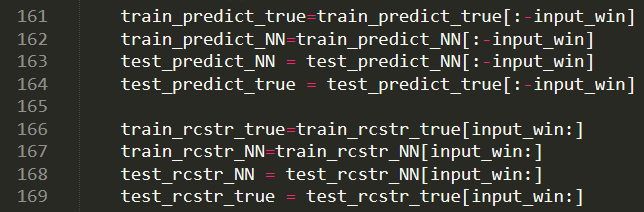
之前经常说预测比重构难，难在哪？



看loss就知道了 outputs\_reshape\_loss是预测损失，conv3d\_1\_loss是重构损失；无论是val还是training，预测loss都比重构loss要大

## 数据形状

“掐头去尾要中段”

对于input数据和predict数据分别删去前window个（秒）数据和后window个（秒）数据进行度量；而ground\_truth则要同时去除前window个和后window个；以只度量中间的部分数据的异常检测结果。

From 24服务器: test\_in\_one\_model.py和auc24\_universal\_find\_best\_in\_trained\_models\_graph\_WGT\_prd2.py