# 人脸识别研发文档

作者：piginzoo

时间：2021/9/17

人脸识别研发文档 1

概述 3

Arcface 3

10.12 训练人脸成功 3

9.28 解决nan和不收敛问题，之，单玩mnist 4

9.17 第一次训练结果 8

## 概述

这个文档，主要用来记录研发过程中的问题和解决方法，方便跟踪项目。

此文档不讲实现，具体实现参考我的技术博客和项目arcface-pytorch中的README.md，比较详细记述了实现原理和代码细节。

## Arcface

AAArcface算法是相对比较成熟的一个人脸特征抽取算法，是2019年的论文，是基于cos距离的分支里最优秀的一个算法。

代码是fork自一个github开源代码，做了大量的修改和增强：

* 追加注释，对每行代码理解，与论文严格对应并注释
* 构建了上述的训练docker容器，方便pytorch1.7-gpu版本的训练
* 实现了一个ealystop，来方便早停
* 改造了Visualizer，方便与容器外的visdom服务器通讯
* 还增加了tensorboard的支持，作为visdom的备份（visdom不好用）
* 改造了数据集加载，配合使用下面说的数据集加载
* 为了对比验证，增加了Mnist数据集的测试，分别测试一般的全连接，和arcface的约束
* 增加了mnist的可视化plot，画了一个二维的环形图，用于和论文的效果做对比

使用CelebA数据集作为训练集：[CelebA数据集](<https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html> )

测试集仍然使用原代码的lfw数据集，用于计算acc：[LfW数据集](<http://home.ustc.edu.cn/~yqli1995/2019/09/28/lfw/> )

### 10.12 训练人脸成功

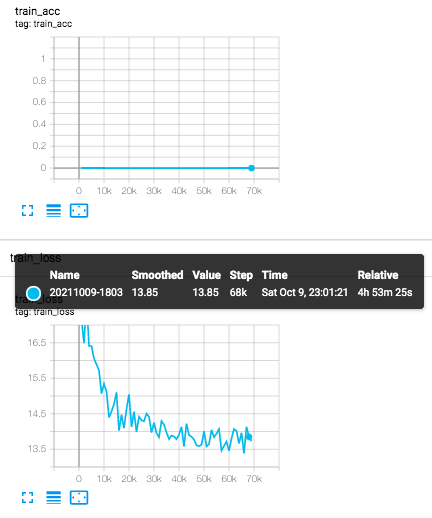
重新训练人脸识别，之前通过mnist的演练，已经对代码、原理、bugs等都有了深刻的把握，因此这次重新切回来训练人脸。

遇到了以下的问题：

1. nan问题

训练过程中，还是会出现loss突然nan掉，后来，通过github上的issue上的一个建议，搞清楚，是因为在计算其中的分母的时候出现了除零，需要修改夹子函数clamp(0→1e-10)，来避免了了除零。

1. 不收敛问题

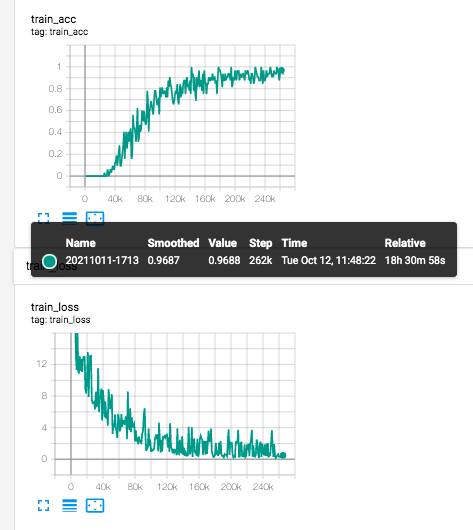


总是不收敛，train acc都上不去，后来查看论文，以及去insightface，也就是arcface作者的代码去寻找灵感，做了如下修改：

* 用了resnet的avg层，放弃了5x5x2048的方式，减少了全连接的参数量
* m=1.2，s=64
* 设置easy\_margin = True，(夹角超过180度，CosFace 代替 ArcFace)

最终收敛结果：

训练大概10个小时，1万个分类，train acc接近到1，test acc大概是82%，还和论文有一定差距，需要再优化。



最好的阈值: 0.5756394，最好的ACC：0.8222698072805139

测试3000对人脸，（最好）正确率0.82，(适配出来的最好的阈值0.58)

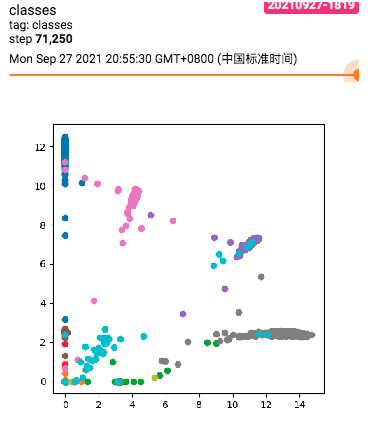
测试acc：0.8222698072805139

### 9.28 解决nan和不收敛问题，之，单玩mnist

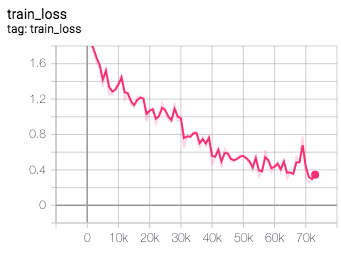
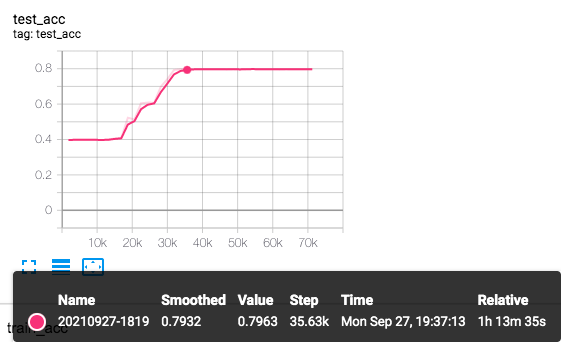
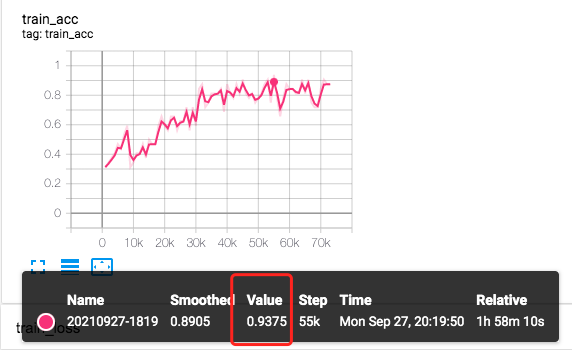
由于不收敛，所以改去train mnist，试图通过小数据集，可以摸清楚细节。

结果改成了mnist，还是40-50%的正确率，到大概几千个batches后，就崩了，nan，loss也nan，resnet出来的features也nan，怀疑是2维度无法表达10维度的分类，不过后来发现不是。

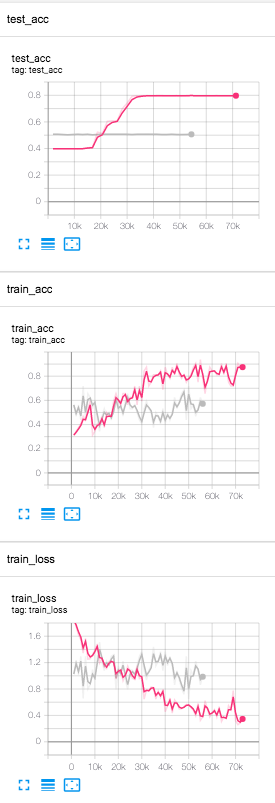
绝望后，再简化成crossentropy+liner-fc，替换掉了focaloss+arcface loss，果然不崩了，但是，正确率也没有想象的那么好，才到验证集80%，训练集90%的acc。



收敛后的2维可视化。



修改后，发现效果更差：（灰色），感觉可能是**去掉normalize**导致的



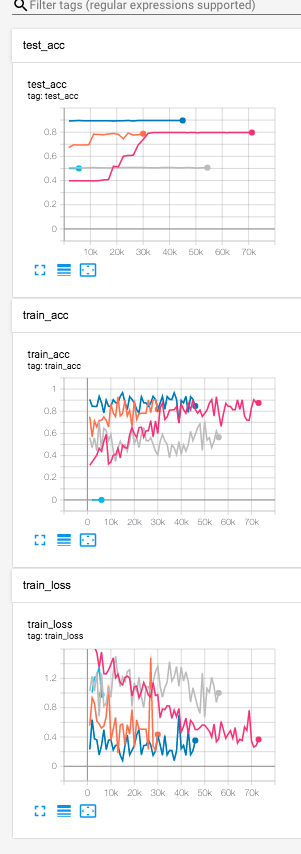
果然是正则化的问题，正则化之后，确实好多了，可以train数据集冲到94%的正确率了。

然后，我有又改了，把2维度表示，**改成了512维度**表示，

果然，正确率上去了，train的acc到了97%，test的acc到了90%，但是还是不理想啊，我反思一下，可能一个是resnet18？或者，是因为我给丫弄成3通道了，肯定不如人家一通道好用。



做个比较吧：



红色：没有分开步长，全都是0.0001

灰色：2维度，但是去掉了正则，分开了步长：resnet0.0001，new0.1

桔色：重新加回了正则

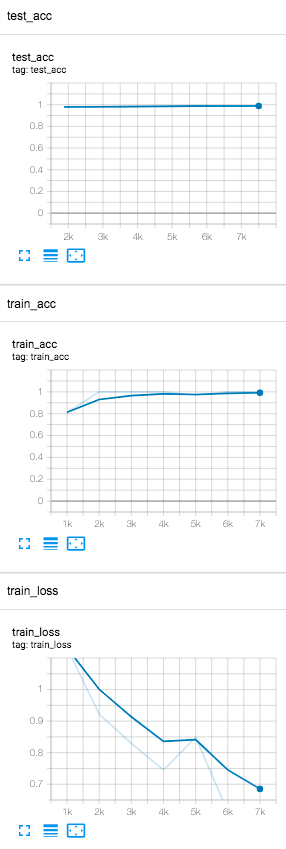
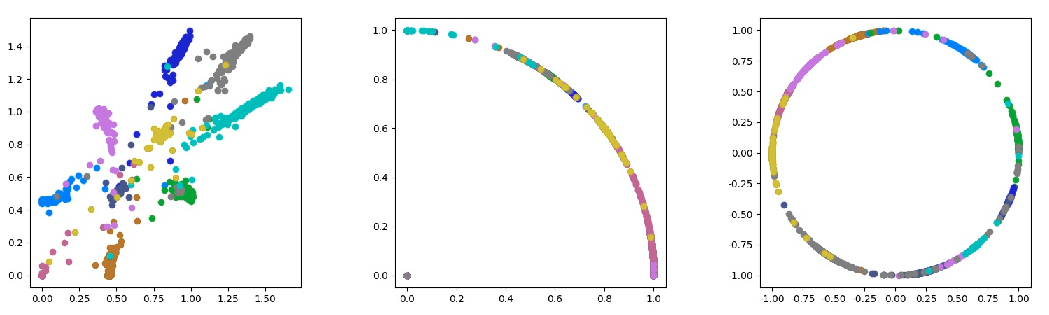
蓝色：最好，加回了正则，且改成了输出feature 512维度



接下来，我简化了网络，只是用resnet输出的512x7x7，接一个全链接到2，然后用softmax输出了，就是下面第一张图，归一化后是第二张图，发现全连接后用了relu所以是图二，去掉relu，得到了图三，就是我想要的效果。

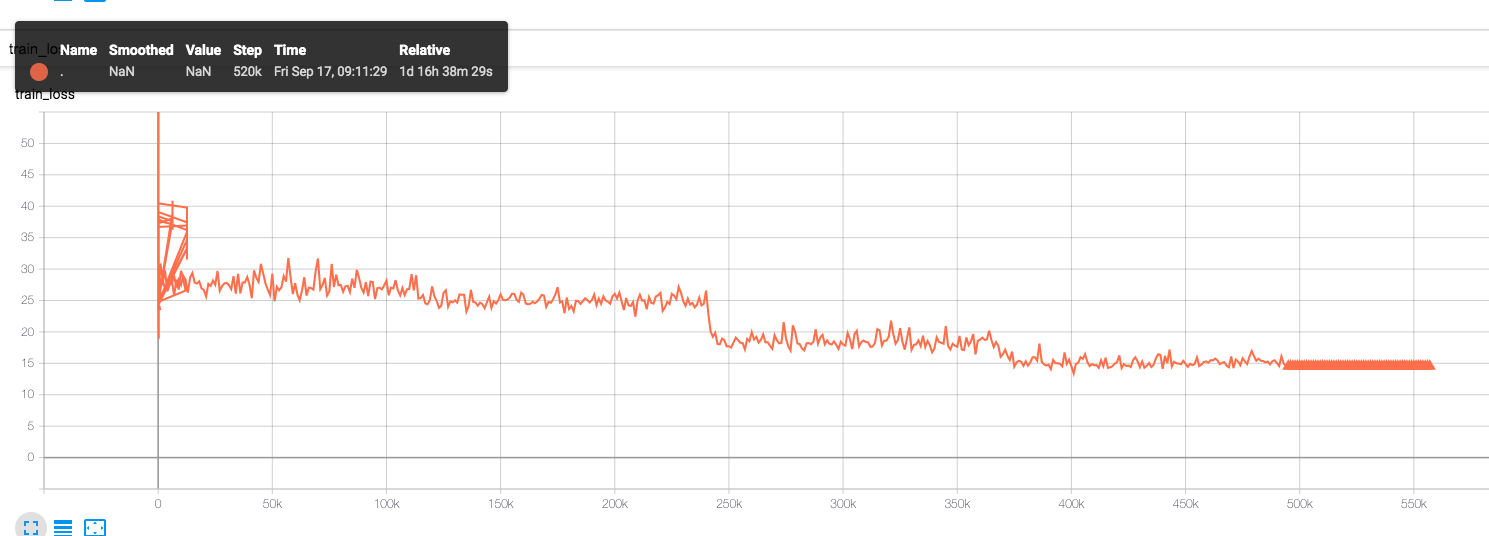
看，类别们虽然可以分开，但是是粘连在一起的。

而且，正确率也上去了，非常高了，测试集99.12%，训练集acc到1了，loss也降到0.6.



### 9.17 第一次训练结果

重构了代码，加入了tensorboard显示正确率和loss：



训练了28小时，最后崩了，而且loss下降的也有限，也不是一个逐渐下降的过程，感觉是有问题的。

  
train和test的acc正确率一直都是0，也很不正常。