VGG16不拟合

并非不拟合，只是拟合慢，前期很久都看不到拟合。

解决方法：用网上的算力，用BatchNorm2d（分布差不多，梯度不会爆炸或消失）

不对劲：BatchNorm2d之后有大约一半的值<0,岂不是有一半要失活。（有点像有选择地dropout，只训练比较大的那部分。）

另一个现象：训练集的acc不怎么变化时，验证集的acc也会提升。

GPU内存泄露:

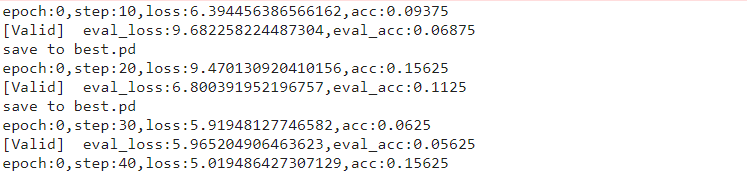
累积信息时不能用CUDA的tensor

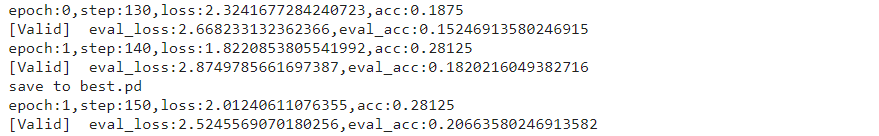
将losses.append(eval\_loss)改成losses.append(float(eval\_loss))

Linear不注意也容易内存爆炸(nm的复杂度)

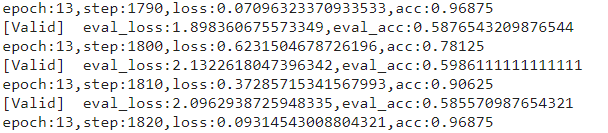
VGG16+batchnorm2d

第一个epoch结束后,可看到loss明显下降。

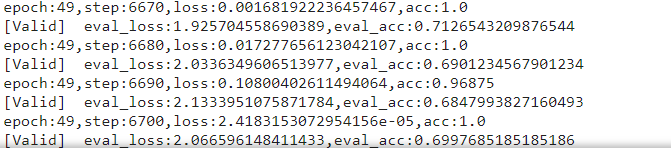




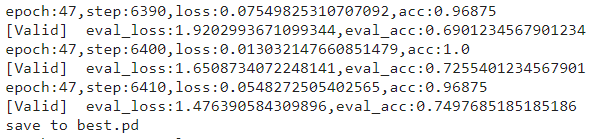
13epoch后 loss有时到0.1附近



50epoch后还有下降趋势,应该没有训练完

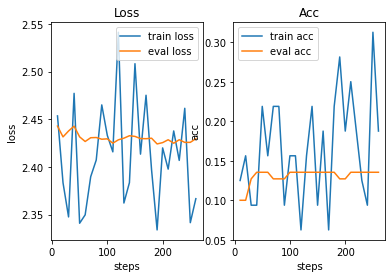


48epoch验证集有一个较好的表现



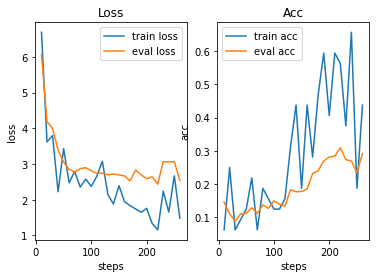
注：之后发现了一个bug，最后一层4096转12的线性层没用上，但是仍能得到答案。

标准VGG16,前两个epoch情况:



所以我会误判不能收敛。

VGG加入BatchNorm2d:

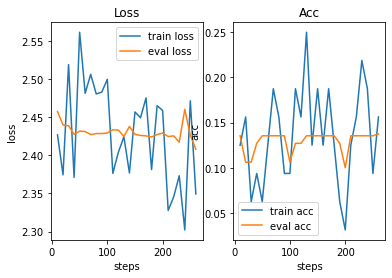


虽然初始loss高,但有明显收敛,且ACC也有明显提高

池化层只压缩宽和高

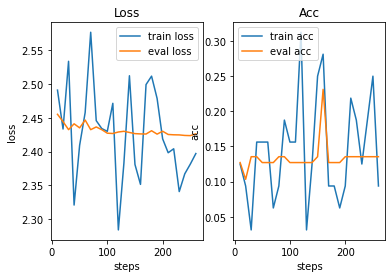
1\*1卷积，改变通道数

加残差层，但用1\*1改变输入通道的维数

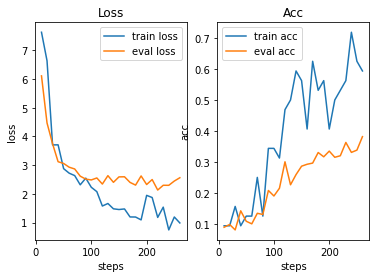


没有用

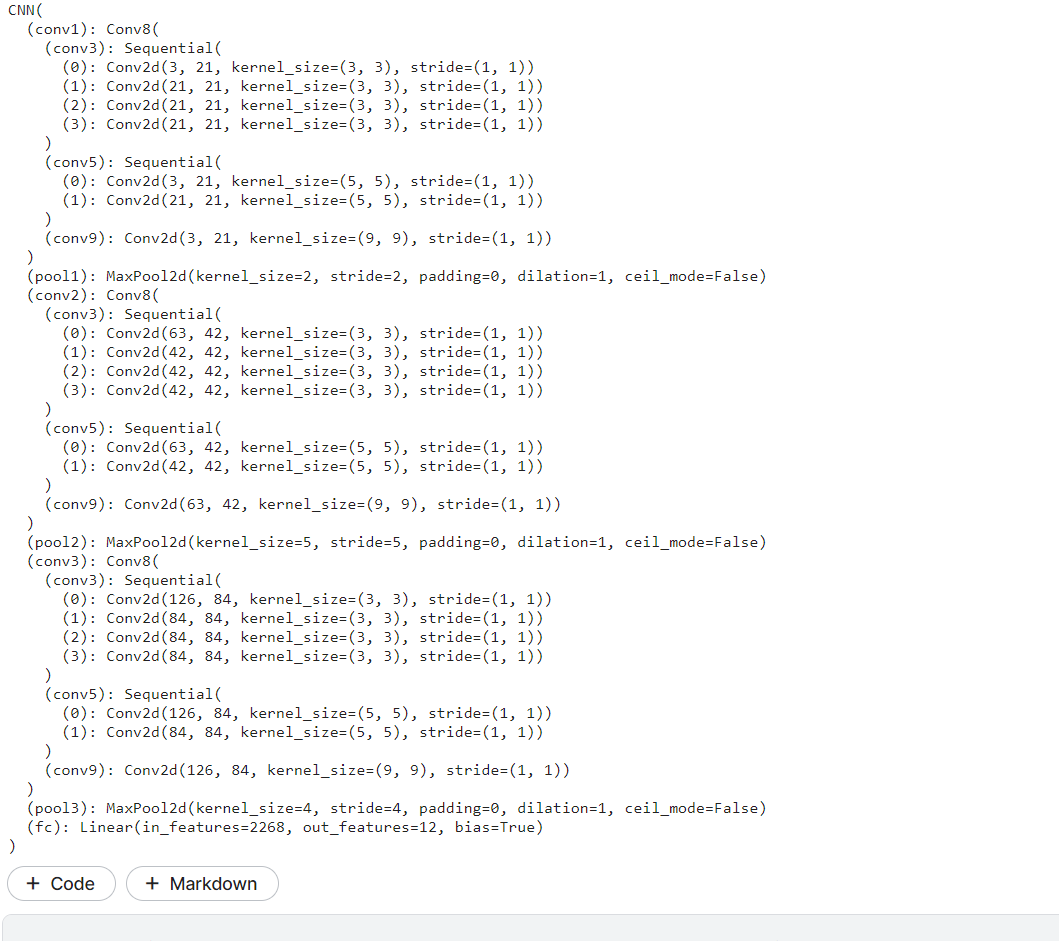
不改变输入通道数，只在3层同类卷积用残差层，没有用。

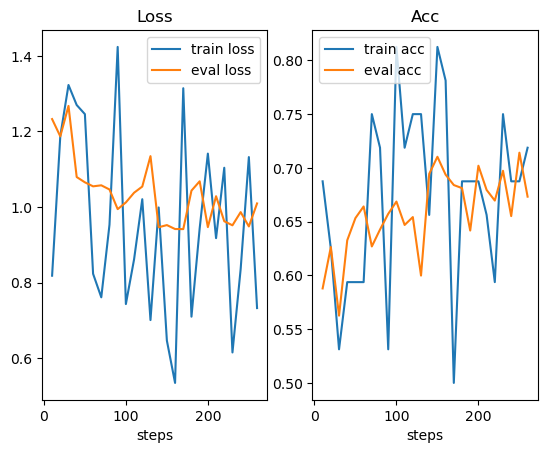
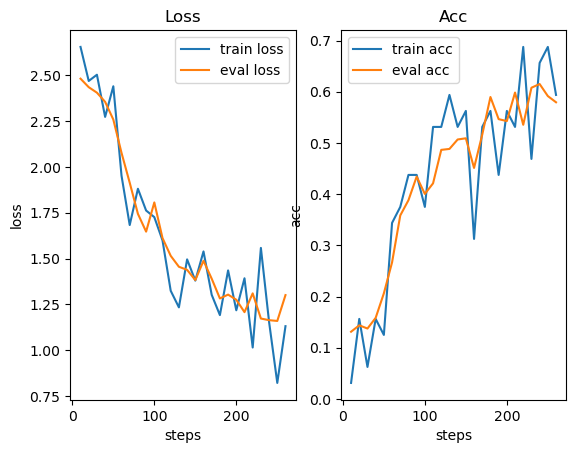


只在3层同类卷积用残差层且加Batchnorm2d

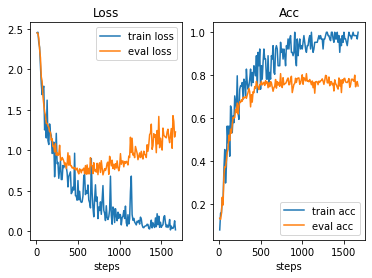


新弄的一个模型:





1-2轮 3-4轮



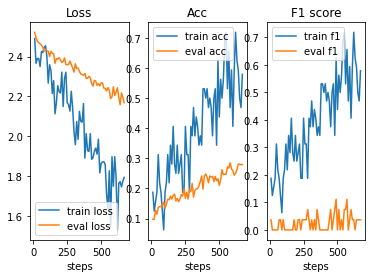
25轮

10多轮时loss最高

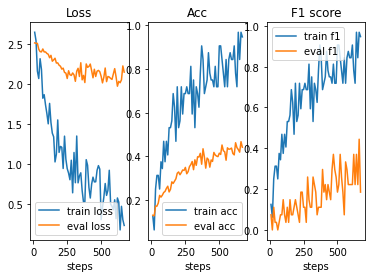
卷积层加Batchnorm2d和ReLU

线性层加Sigmoid 和 Dropout

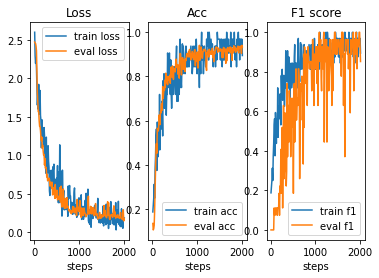
5轮情况



去掉Sigmoid函数,拟合更快了。



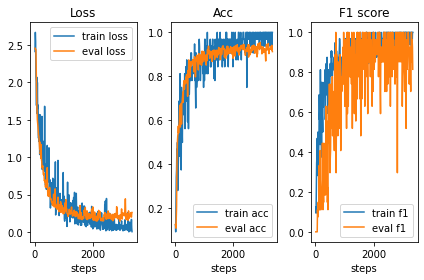
加eval(),15轮:



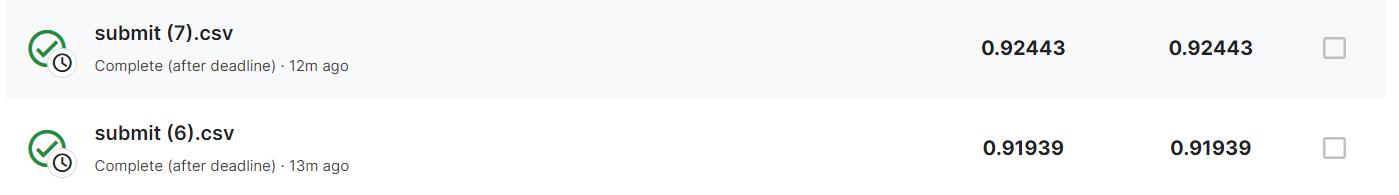
Final.pd 和best.pd成绩(best保留的是maxacc)



因为15轮仍未出现过拟合的情况,所以训练25轮,且这轮保留maxf1和maxacc



0.91是best.pd，0.92是final.pd



仔细一想应该保存最小loss才对。

自己和kaggle notebook的转换

1. 数据集路径
2. Batch size大小

先整合,少了一个循环,out\_channels循环

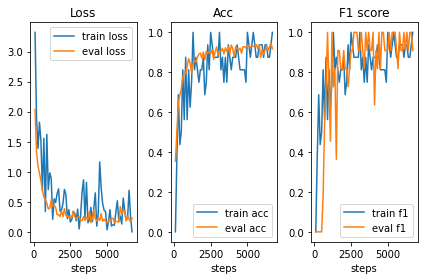
前向传播很快，但反向传播很慢。不是反向传播慢，而是每步就验证慢（相当于把整个验证集跑了一遍）。

不，反向传播确实慢

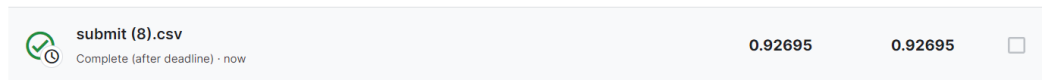
回到昨天的CNN

调整:取min\_eval\_loss为best.pd, 25epochs,调高通道数,补加一层线性层

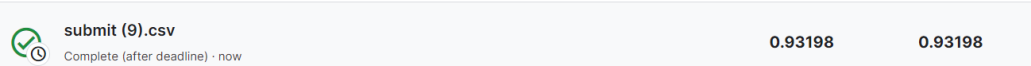
学习率:1e-4



Final.pd成绩:



Best.pd成绩(取最小eval loss):

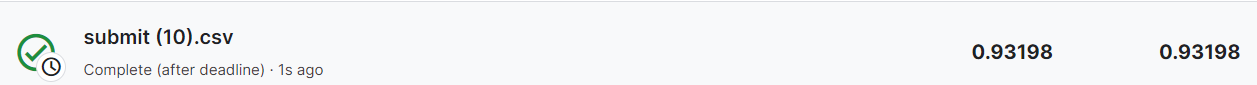


调低学习率(=1e-5),继续训练5轮

出现意外,没得到图像:

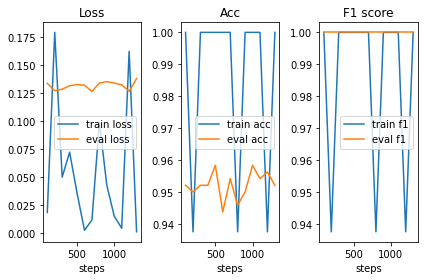
验证集loss稳定在0.12,f1稳定在1.0

Final.pd:

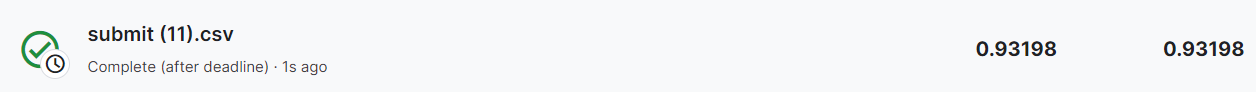


Best.pd,因意外丢失

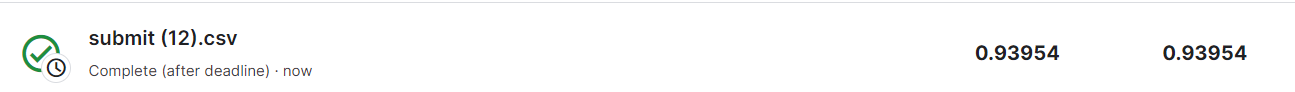
Lr=5e-6再训练5轮



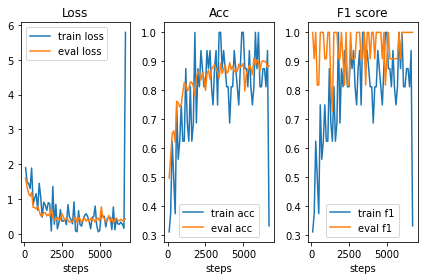
Final.pd:



Best.pd:



使用图像处理后:



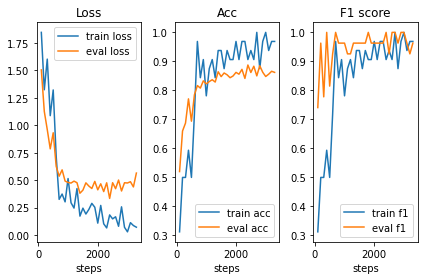
Best:



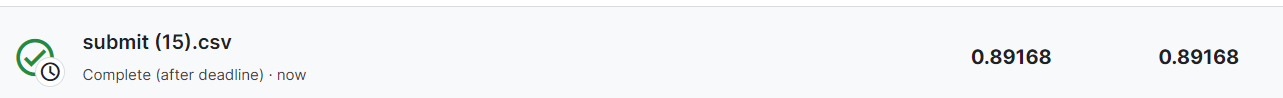
Final:



小通道,图片处理:



Final:



Best:

