**目录**

2019-04-22

1. 学习caffe中的多个分类label的输入问题。默认情况下caffe的训练数据集是单分类器。有几种方法修改为多标签分类器：

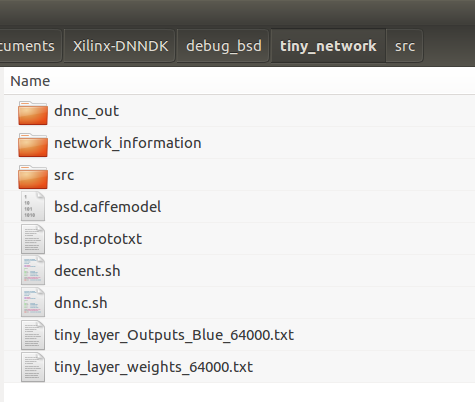
对于DNNDK可以使用如下图作为calibration.txt的样本：



但是，words.txt对应是什么文件？

1. 下面是使用tiny\_network，然后修改了src/main.cpp后，得到的tiny网络的输出：

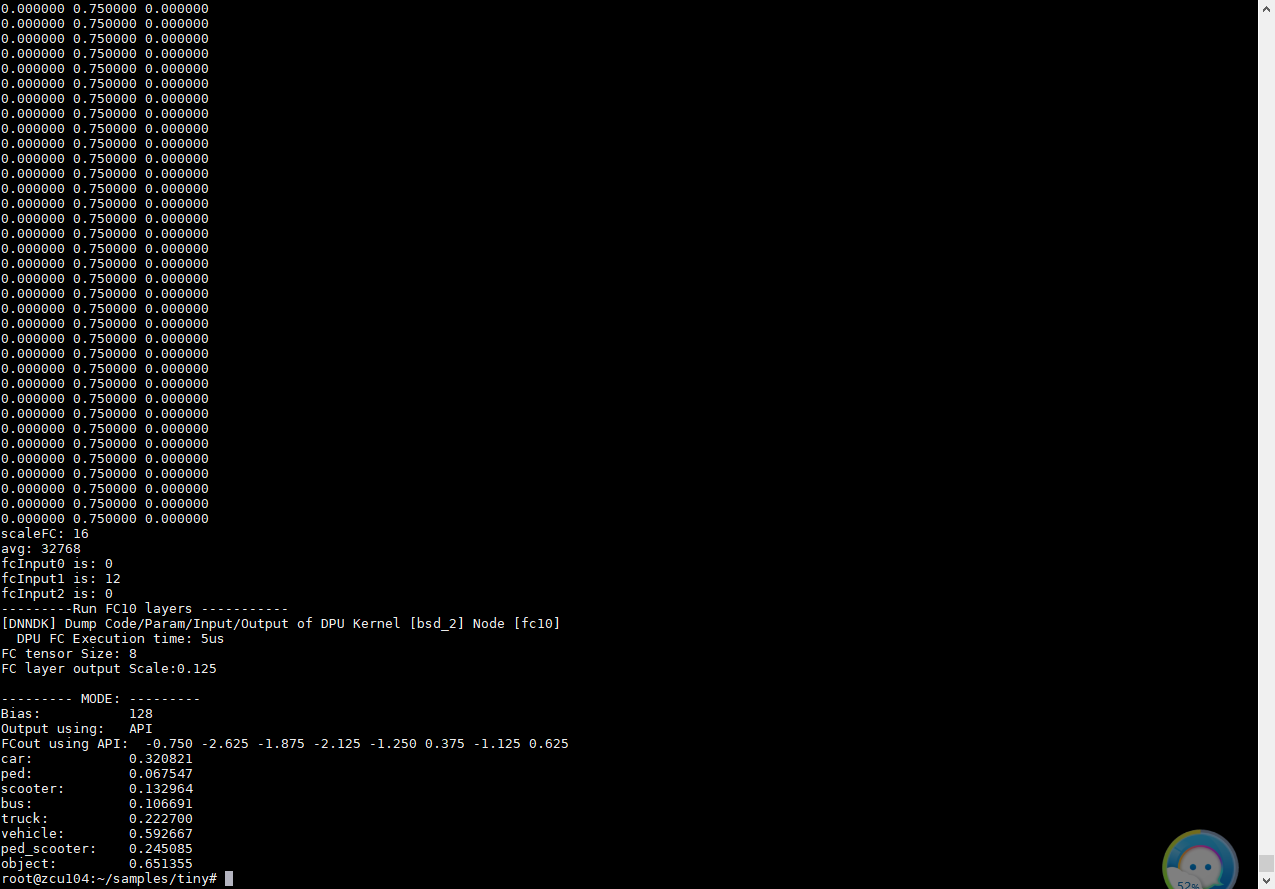
（详细的网络结构文件/参数文件/生成网络信息/main.cc/Makfile/详见debug\_bsd/tiny\_network目录！）



蓝色图像，网络输入为RGB的signed char类型(**CV\_SC8C3**), R/G/B = -128,-128,127



运行的结果：



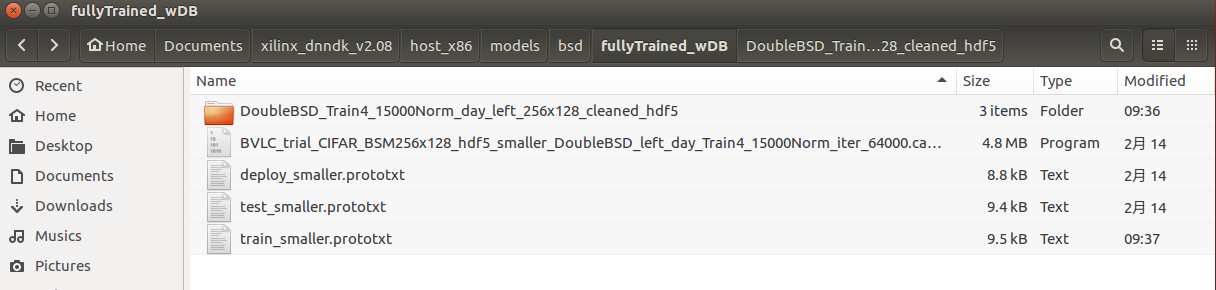
这里生成了中间各层的weights&bias，可以与amandeep生成的weights&bias进行对比。

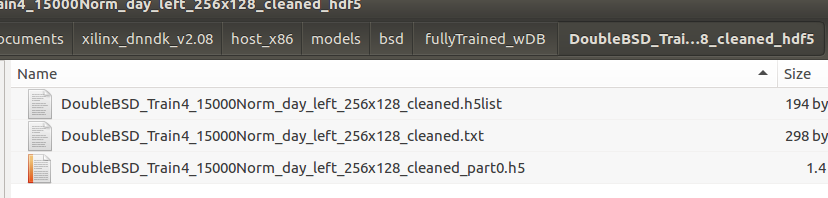
并可以进行各层的计算，看是否有问题。

**20190507**

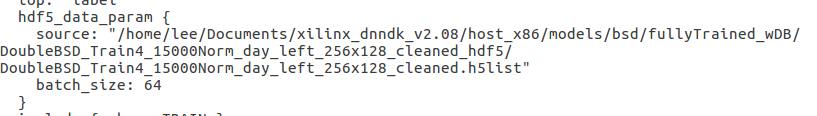
**使用正确的数据训练集得到正确的网络结果：**

**1.**  Amandeep找到了使用hdf5格式数据作为DNNDK的输入校准源数据。不需要calibration.txt进行校准。使用fully\_Trained\_wDB/ DoubleBSD\_Train4\_15000Norm\_day\_left\_256x128\_cleaned\_hdf5/ 下面的3个文件

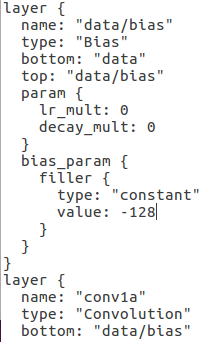




1. 其中的\*\*\*.h5list中需要修改为\*\*\*.h5文件的路径。
2. train\_smaller.prototxt中的数据集源文件需要指定\*\*\*.h5list的地址。如下：

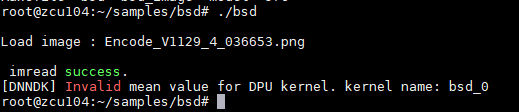


1. DECENT 不认识“Bias”层，需要删除。然后修改conv1a的输入。如下图：



1. 这时decent.sh可以通过，然后需要删除deploy.prototxt中的最后3层(计算loss，来得到学习正确率的)。
2. 最后可以得到elf文件。
3. 从calibration计算的loss可以看到，这个没有修改BIAS版本的错误率loss较高。所以可能到不到正确的结果。(Amandeep又重新进行了训练)

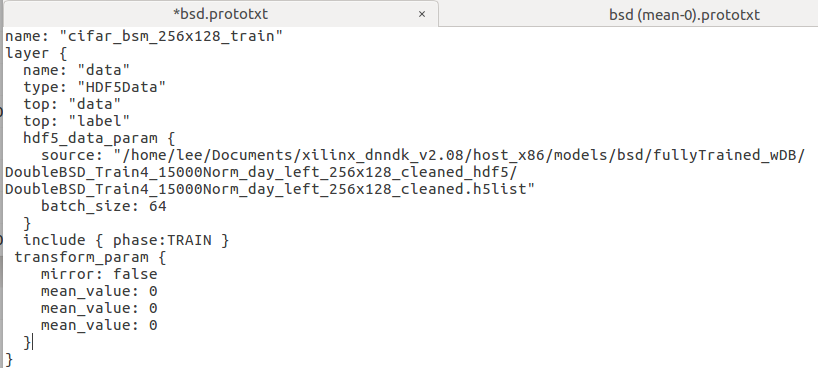
**2.** 运行./bsd出现错误，如下图：



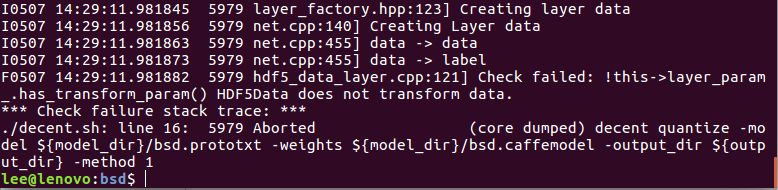
mean value是无效的。

**解决方法:**

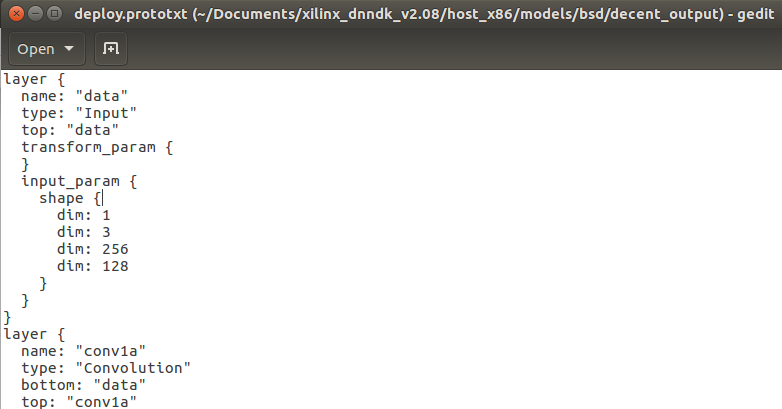
* 在train.prototxt里面添加mean\_value:值，再进行DECENT会报错：



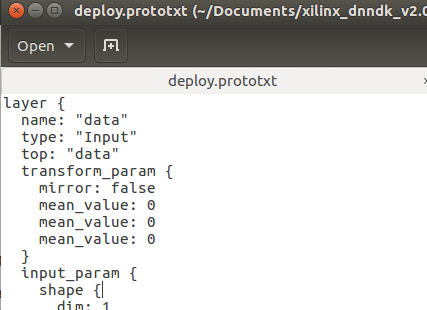
提示：



* 需要在DECENT生成的deploy.prototxt中进行修改：



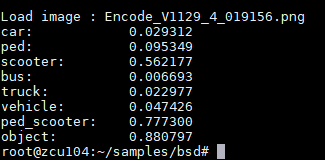
添加transform\_param{} 里面的mean\_value：0。

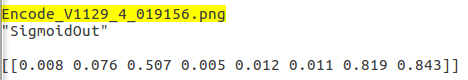


然后即可执行DNNC.SH运行得到elf文件。

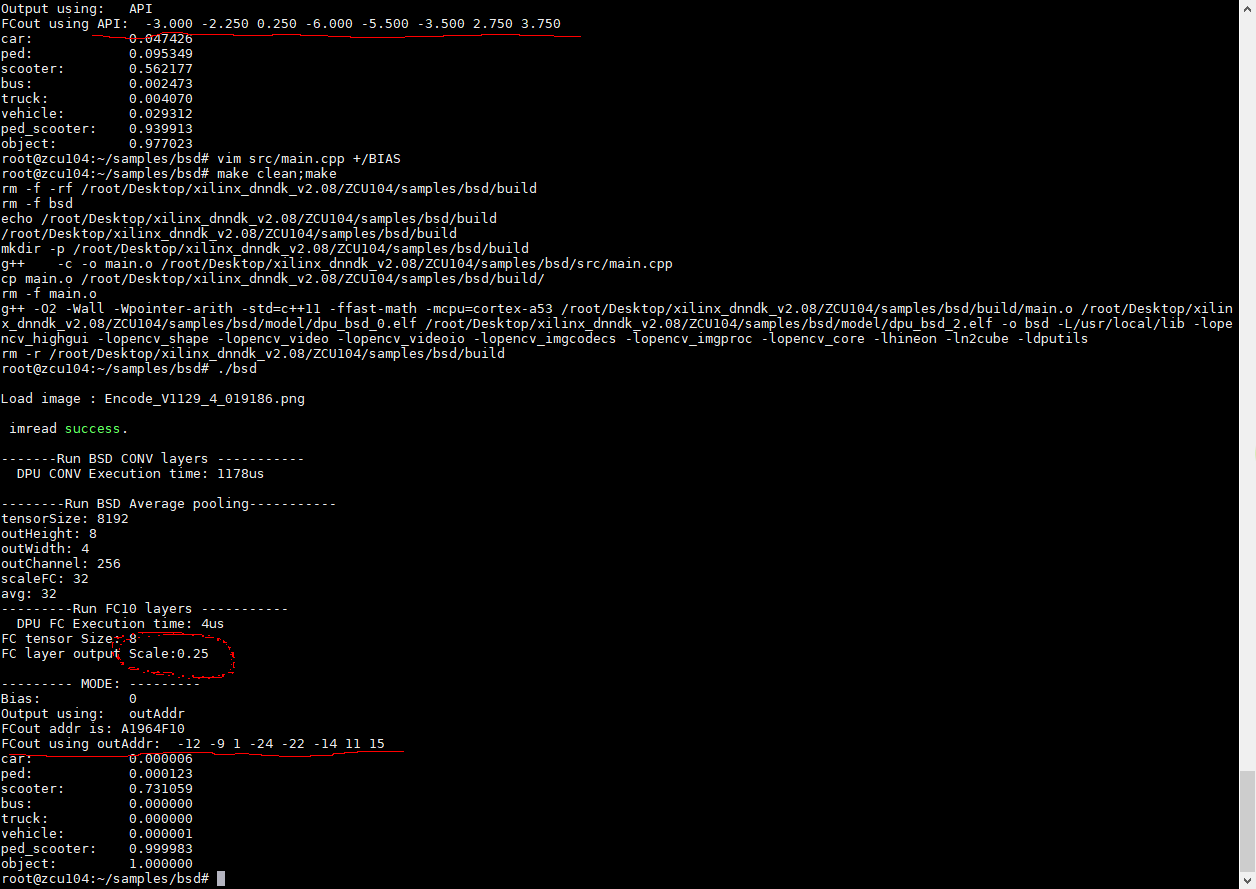
得到结果：原始图片和ZCU104&PC端执行结果如下：







**3.** 对比了使用API和outAddr来获取fc层输出，可以从上面看到，API和outAddr的值是相差一个scaleFC因子1/4的倍数。如下图：



**4. 某些图片和PC端的计算结果还是有比较大的出入的。**

**20190508**

**读取视频转换成图片进行BSD**

**1.**