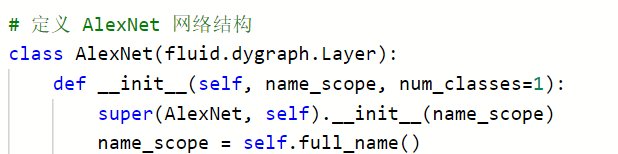
**实验：AlexNet分类**

**张靖祥 2017304010413 计算机172**

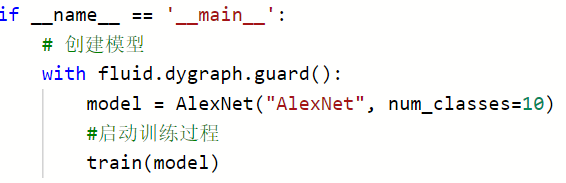
1. 程序分析

首先，本案例程序没有使用静态网络的搭建方法，可见，这是动态流图网络。传统的tensorflow是静态流图，在一些循环结构的时候很难去实现，当时我采用了多次循环并且用零矩阵相乘的方法才得以实现。飞浆的动态网络可以直接解决这一个问题。但是由于换了一个框架，因此还需要很多的熟悉过程。

下图可以看出，继承了fluid.dygraph.layer类，并重构了其初始化函数，获取了名字空间



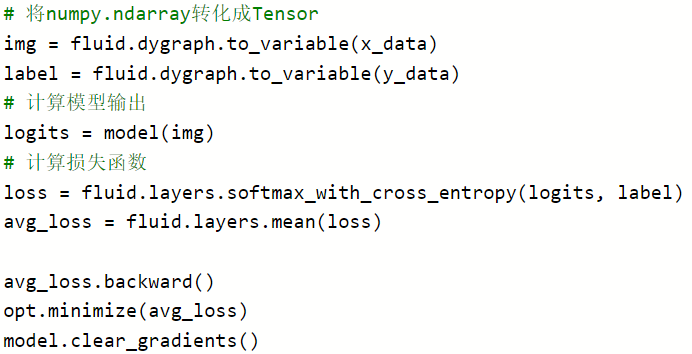
这里直接创建了AlexNet的类，



进行了初始化



整个训练流程更加python化，方便操作，但是同时其内部的机理更加难以窥透



1. 网络模型



与原定义一样，采用了AlexNet网络模型

1. 学习过程显示

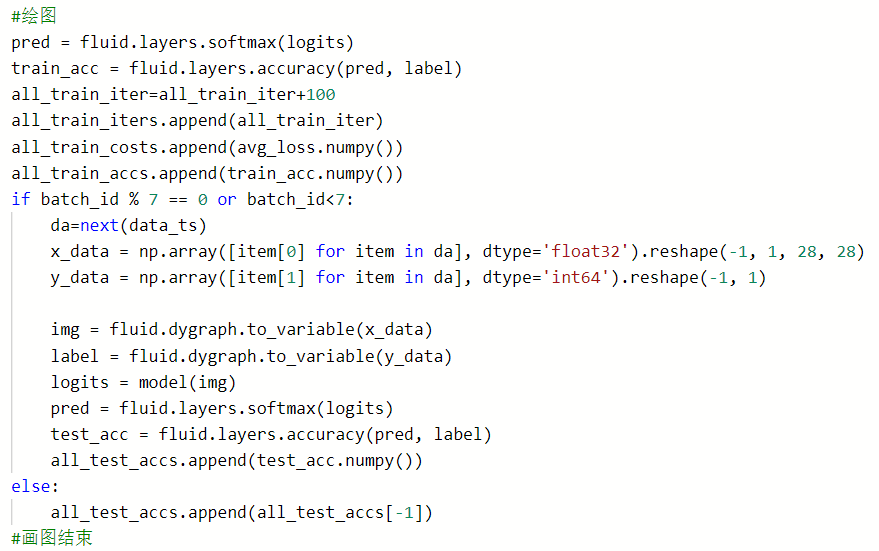
采用了直接绘制的方式



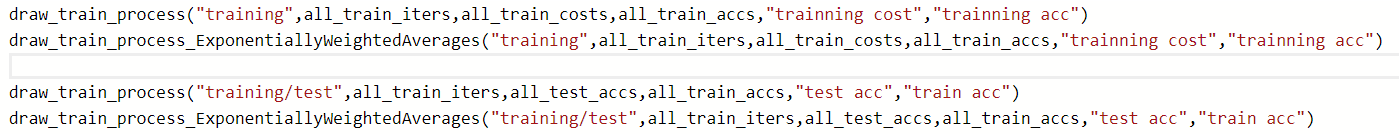
采用了加权指数平均的方式



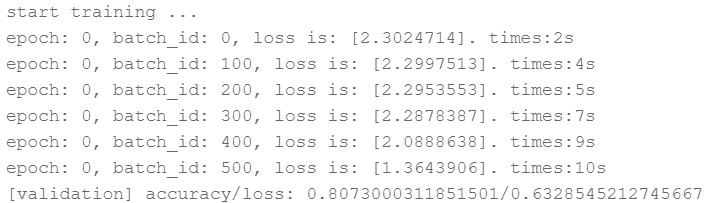
训练过程中插入绘图代码：



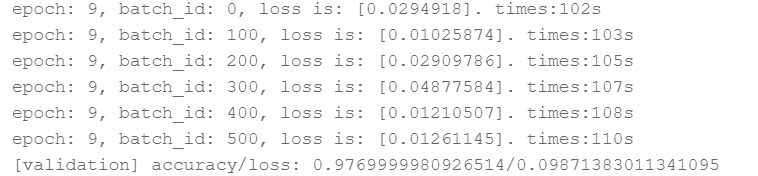
训练结束后进行绘制：



1. 训练结果（GPU训练）

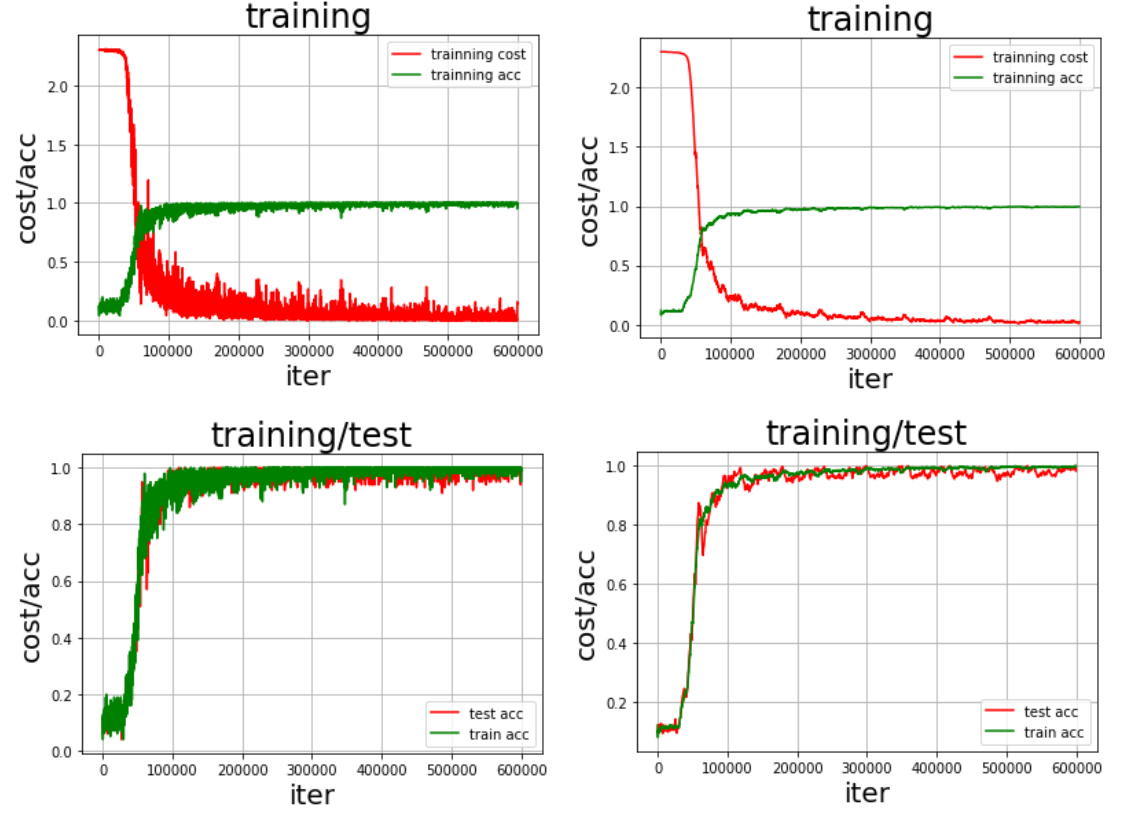


… … … … … …



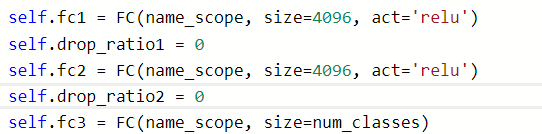
之所以训练了10个epoch，是因为第10到15个epoch几乎不在变化，没有必要继续训练了。

下图中左面的两个为直接绘制，右面的两个为平滑版本。上面的两个为训练和损失的图像，下面的两个为训练和测试的准确率变化情况。两个同步变化，

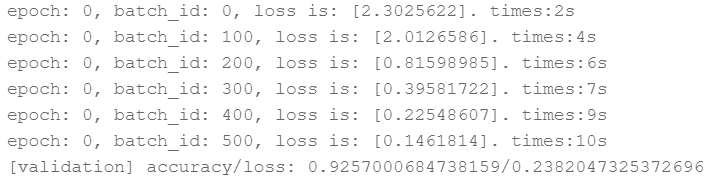


1. 不使用Dropout的测试

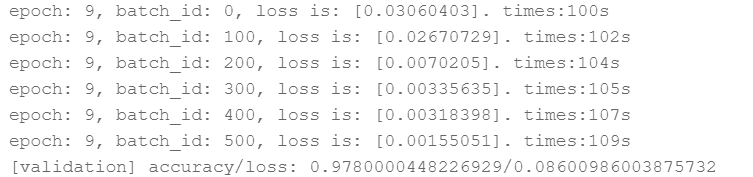
由于数据极其简单，因此我认为不会出现过拟合情况，所以我将Dropout设置为0



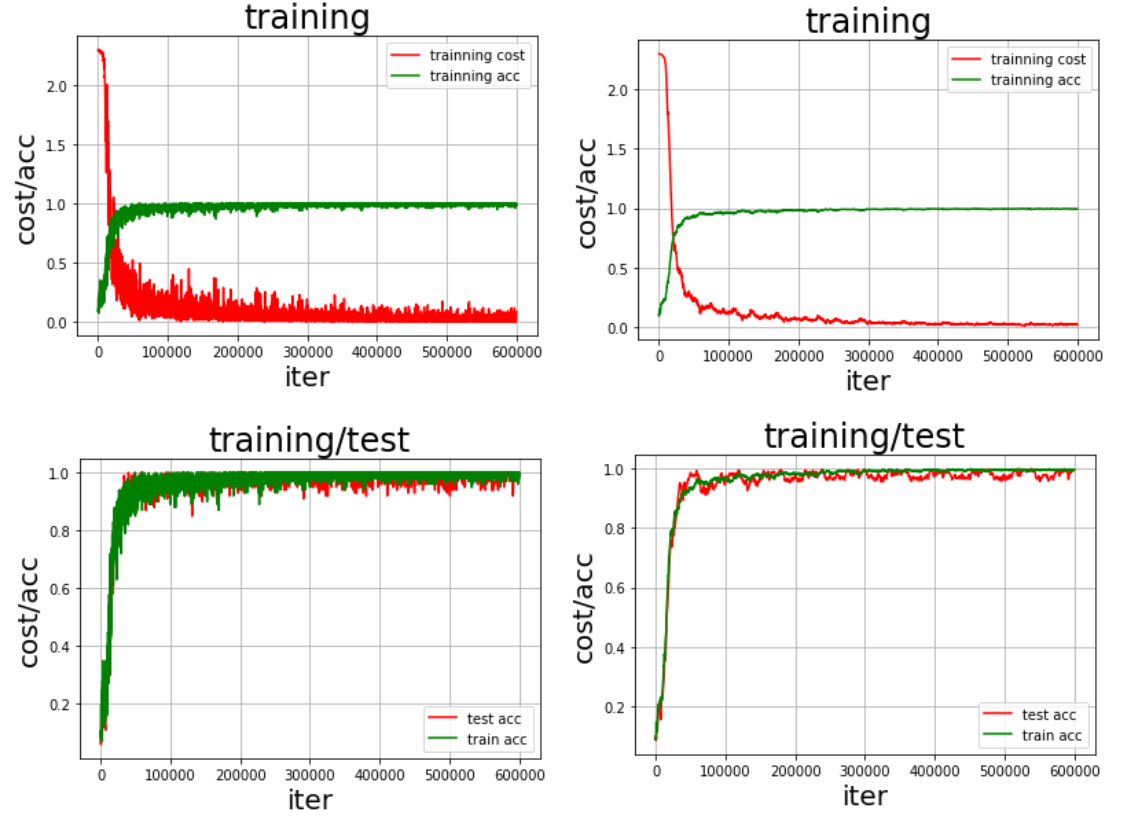
结果：



… … … … … …

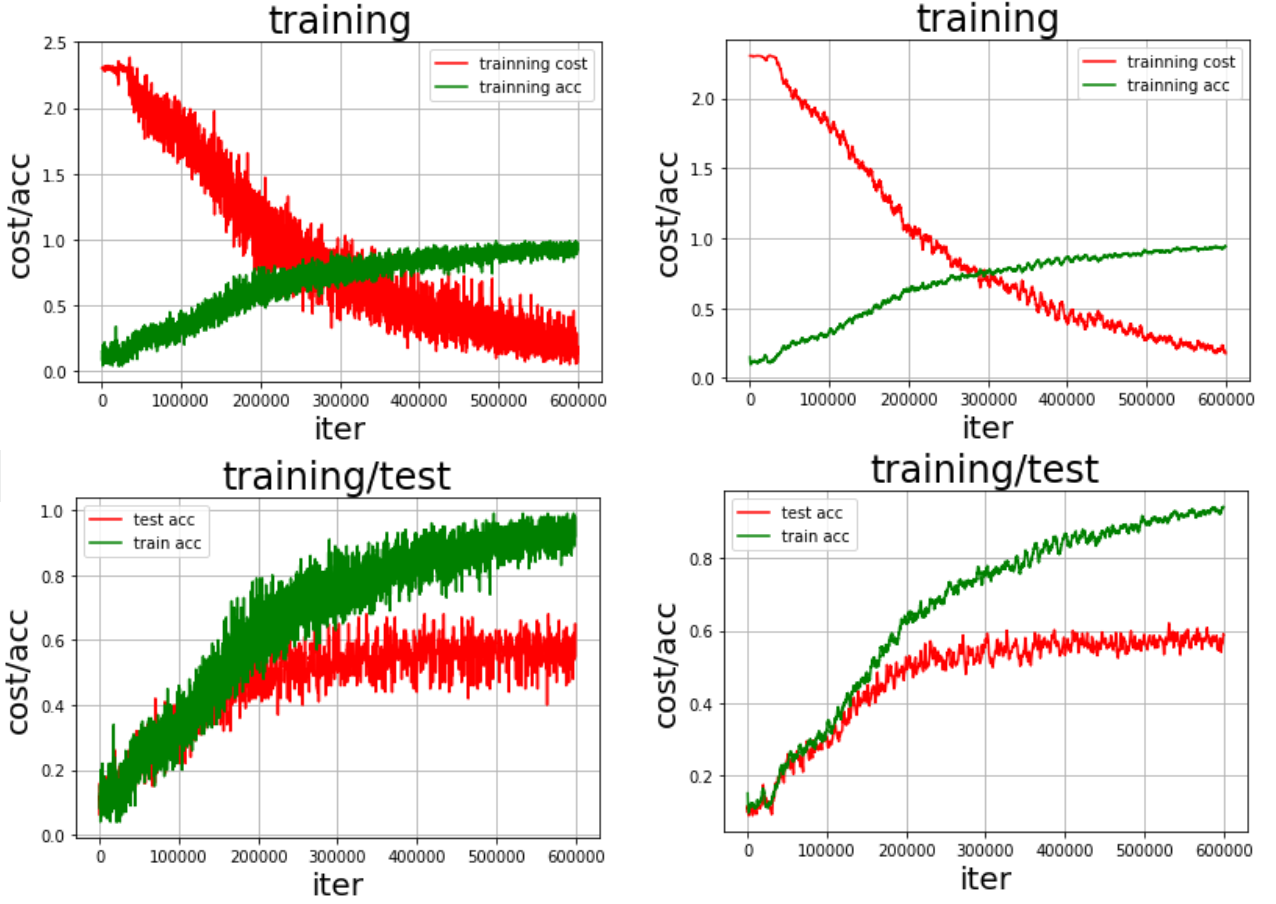


如下图，与预期一致，收敛速度更快了。



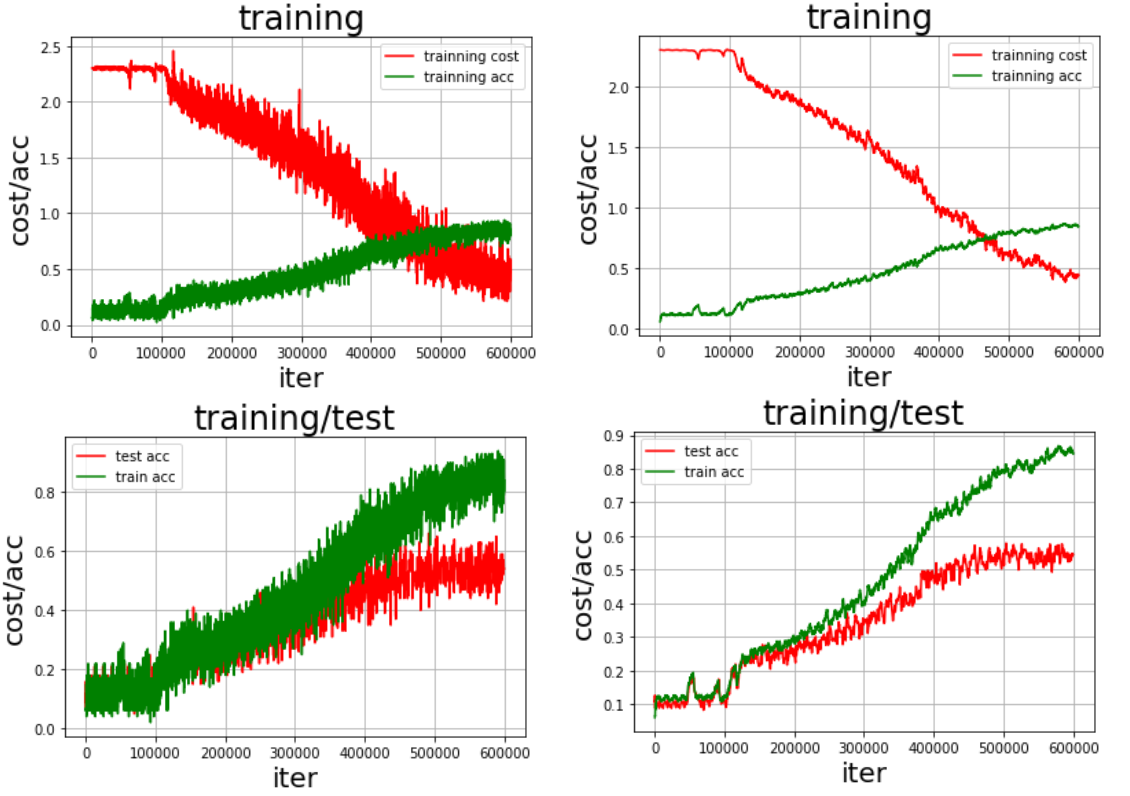
1. 另一个版本的手写图像识别（训练集12000，测试集50000）

参数：学习率0.03，周期：50，drop rate：0，优化器：moment，如下图（损失和准确率的图）

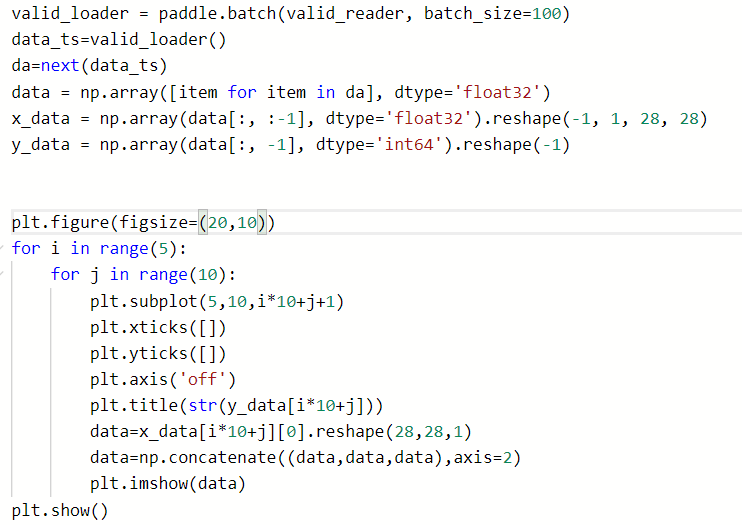
****

准确率低到百分之60，令人惊奇

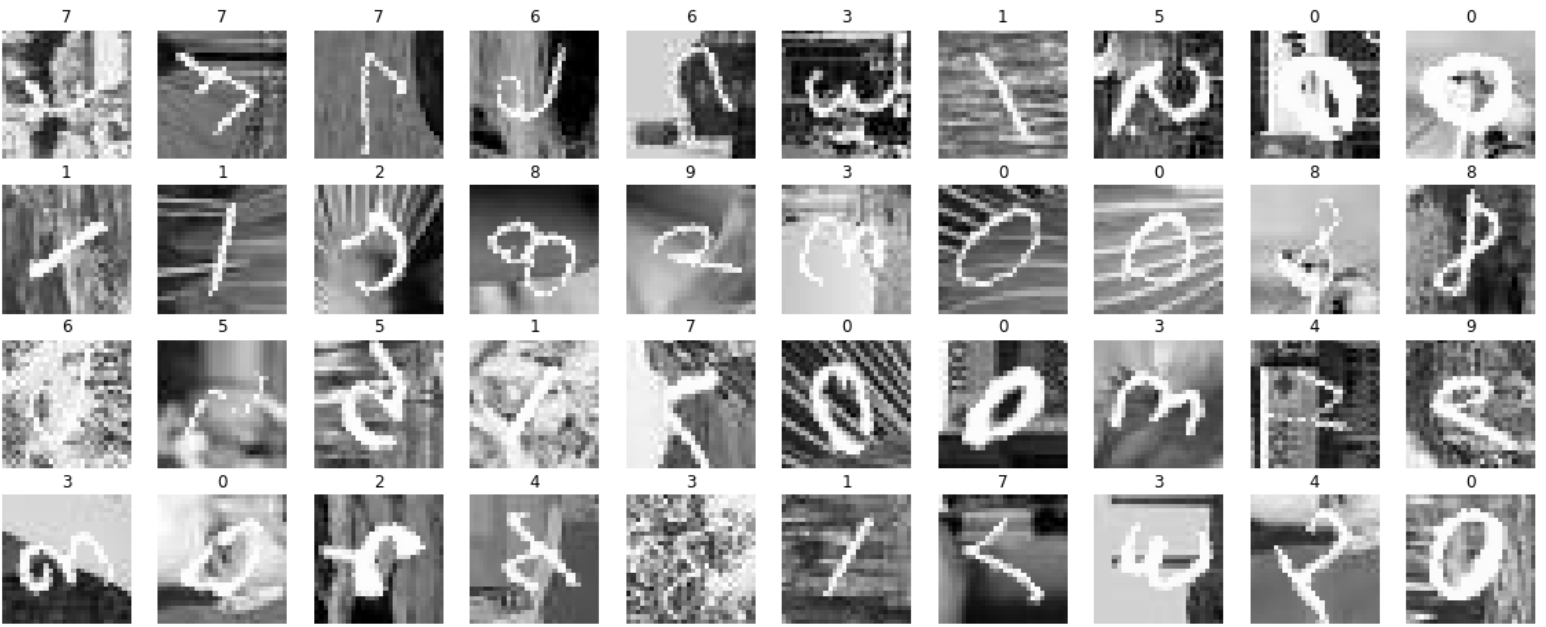
参数：学习率0.03，周期：50，**drop rate：0.5**，优化器：moment，如下图（损失和准确率的图）



显示下这么差劲的数据



训练集如下



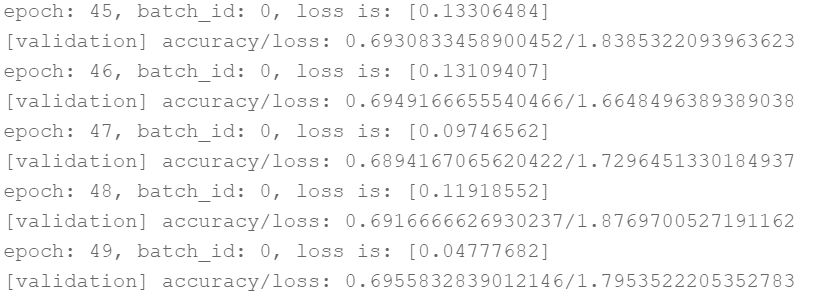
测试集如下

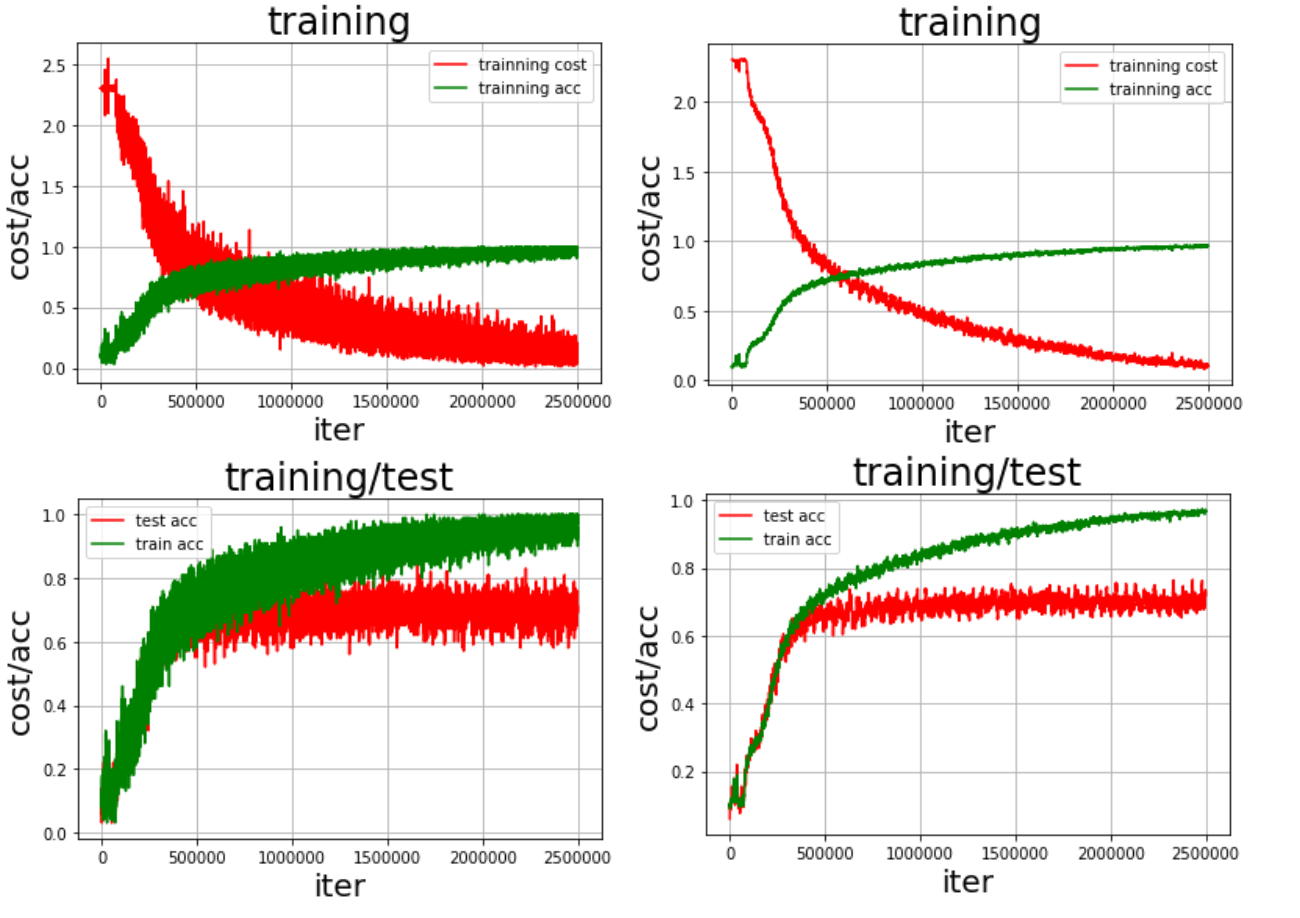


图像各种旋转，一堆噪声

**注：测试与训练数据给反过来，得到一下结果。**

参数：学习率0.03，周期：50，drop rate：0，优化器：moment，如下图（损失和准确率的图）





测试集准确率百分之70左右