

## 在Tiny-ImageNet数据集上训练Resnet模型

Tiny-ImageNet是一个小型的、常用于教学的数据集，包括200个类型，10万张训练图片和1万张验证图片。在本次作业中，以pytorch提供的示例（<https://github.com/pytorch/examples/blob/main/imagenet/main.py>）为基础，在其上修改以适配Tiny-ImageNet，并且限定模型为resnet18。

### 准备工作：

你需要有可以运行的Pytorch环境，其中安装了torch、torchvision两个主要的python包及其他依赖。最好有GPU环境，可以加快训练过程，但在本作业中也不是那么必需。GPU可以通过<https://www.bitahub.com>申请账号（联系助教以获得算力支持），或者其他你熟悉的平台。

Tiny-ImageNet的下载地址在<https://image-net.org/data/tiny-imagenet-200.zip>，或者通过QQ群的共享文件获得

### 完成以下任务：

1、根据Tiny-ImageNet图片大小（3\*64\*64），计算图片经过各层处理后的中间结果的大小。请列出各层的名称及输出的大小。

提示：你可以根据模型结构计算，也可以借助TensorBoard绘制Graph（即模型结构）的功能来完成。

2、在<https://github.com/pytorch/examples/blob/main/imagenet/main.py>的基础上，做必要的改动以使得其可以在Tiny-ImageNet上训练。请提交改动后的代码，并使用git工具（git patch或git diff）来生成一份改动说明以帮助助教快速地批改作业。

提示：

（1）代码中还包含了分布式数据并行的代码，这部分功能在本课程中不要求掌握，你可以忽略它。

（2）原代码中对应的是ImageNet，它有1000类，也就是output有1000维，你需要修改成200维，通过很简单的设置。

（3）数据集（torch.utils.dataset）是你需要重点处理的对象，Tiny-ImageNet的文件结构中，训练数据集是可以直接使用原有代码的，但验证数据集不行，需要自行修改。如果你使用原有代码，那么验证数据集中每个样本的路径是正确的，但标签是错误的。一种供参考的方案是：利用wnids.txt和val/val\_annotations.txt来重新修订每个样本的标签，注意这个标签是一个整数值，它背后所代表的意义要和训练数据集中的数字完全对应上，否则，你在验证数据集上的loss和精度是错误的。

（4）这部分代码量在20行左右。但取决于你的实现。

（5）原代码中包含了对图片进行伸缩和裁剪的代码，这是因为ImageNet数据集的原始数据并不确定是多大，也不一定是正方形，但在Tiny-ImageNet则不同，已经事先做过拉伸和裁剪了，这部分代码并不需要。（当然，数据处理本身就是case-by-case的，你也可以继续探索更优的方式。）

3、在代码中增加torch.utils.tensorboard的代码，以能在TensorBoard中观察训练集Loss、训练集精度、验证集Loss、验证集精度的变化。

提示：<https://pytorch.org/docs/stable/tensorboard.html>

4、运行程序，将resnet18在训练集上的精度（Top5）训练到95%以上，并对第3步中的曲线进行截图，编写实验报告，分析曲线变化情况。

提示：

(1) 大约需要15~20个epoch

(2) 在我的测试环境中，CPU会比GPU慢5倍，训练一个epoch需要8分钟。

5、至少保存2个训练过程中模型的checkpoint，并使用代码中的--evaluate选项，对比两次评估的差异，并至少找出其中10张评判结果不同的图片。

**截止时间：2022年5月16日（星期一） 19:30**