

## 23 春 Python 与深度学习基础

### 第二次作业要求

#### ● 作业内容:

#### 一、 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。

##### 1. 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，有 GPU 环境。

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

##### 2. 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 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列出你的工作。

提示:

- 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。
- 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。

- 这部分代码量在 20 行左右。但取决于你的实现。
- （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：大约需要 15~20 个 epoch

- (5) 分别在无 GPU、1 个 GPU、多个 GPU 环境下，重复上述过程，观察和量化评价训练速度上的差异，并做讨论。

提示：多 GPU 可以使用数据并行或分布式数据并行，如果你用到多个服务器上的 GPU，那只能用分布式数据并行。

- (6) 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。

## 二、 复现 Word-level Language Model 并讨论

### 1. 完成以下任务：

- (1) 参考 [https://github.com/pytorch/examples/tree/main/word\\_language\\_model](https://github.com/pytorch/examples/tree/main/word_language_model)，复现训练和文本生成的过程。要求使用 Transformer 模型。提供实验截图。
- (2) 利用 TensorBoard 或其他工具绘制 Transformer 的结构。
- (3) 阅读论文

[https://proceedings.neurips.cc/paper\\_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html)，并参考其他资料（如 <https://jalamar.github.io/illustrated-transformer/>），讨论：Transformer 和 CNN 在捕捉上下文依赖上有什么差异？

**截止时间：2023 年 5 月 28 日（星期日） 23:59:59**

## ● 提交内容：

本次作业需提交三项内容：

1. **代码。**代码需放在 `github`, `gitee`, 学校 `git (git.ustc.edu.cn)` 或其他任何可以公开访问的代码库上。源代码包括每个模型的代码、训练以及测试代码，训练好的模型文件(.pth 文件)。
2. **设计报告。**完成作业中要求的任务。描述完成过程中遇到的困难及解决方法，描述作业的收获（或者吐槽）。A4 纸，小四字号，单倍行距，标准页边距（上下 2.54cm, 左右 3.17cm），图文并茂，建议五页以内。
3. **视频。**以视频的方式，展示测试的过程和结果，建议录屏。（Windows10 下可以使用 Win+G 启动录屏，如果文件过大，可以使用格式工厂 `format factory` 转码转大小）。

## ● 评分细则：

### 1. 代码（60 分）

代码中至少要符合任务要求的全部代码：

➤ 模型代码（20 分）：

➤ 训练代码（20 分）

训练流程规范清晰，可视化训练过程中 `loss` 和准确率的变化

➤ 测试和分析代码（20 分）

测试代码逻辑清晰，结果要保证较高的准确率以及可视化分析。

### 2. 报告（40 分）

➤ 实验细节（20 分）

➤ 简单说明代码所实现的功能。

描述代码中所用的 `API`，如参数说明，参数限制，返回数据等。

描述代码中设计的函数和类。

描述实验过程中遇到的难点与解决方案。

➤ 实验总结（20 分）

对实验结果的分析，及代码设计过程中的心得体会。

## ● 提交方式：

作业内容统一由电子邮件提交。

作业邮箱：pythondl@163.com

邮件标题写明学号姓名和第几次作业，正文中写明代码库链接，附件附上报告和视频，**不要压缩**，邮箱中**不要附代码文件**，代码我们只以链接中的为准。考虑到学校邮箱有着附件容量的限制，可以使用其他邮箱发送，但请务必注明学号姓名。

**作业期限：北京时间 2023 年 5 月 28 号 23: 59， 每迟 一天扣 2 分**

提交邮件示例：

