

# 实验二

## 实验要求

使用 pytorch 或者 tensorflow 实现卷积神经网络，在 ImageNet 数据集上进行图片分类。研究 dropout, normalization, learning rate decay, residual connection, network depth 等超参数对分类性能的影响。

## 实验步骤

1. **网络框架**：要求选择 pytorch 或 tensorflow 其中之一，依据官方网站的指引安装包。（如果前面实验已经安装过，则这个可以跳过）
2. **数据集**：这次实验使用 Tiny-Imagenet-200 数据集，包含 200 个类，每个类有 500 张训练图像，50 张验证图像和 50 张测试图像。由于测试图像没有标签，因此使用数据集中的验证集当作测试集，并从训练集中手动划分新的训练集和测试集。下载链接：  
<http://cs231n.stanford.edu/tiny-imagenet-200.zip> 或 <https://rec.ustc.edu.cn/share/4bfe7ba0-cfab-11ed-9877-5f868bfa864a>
3. **模型搭建**：采用 pytorch 或 tensorflow 所封装的 module 编写模型，例如 torch.nn.Linear(), torch.nn.ReLU() 等，无需手动完成底层 forward、backward 过程。
4. **模型训练**：将生成的训练集输入搭建好的模型进行前向的 loss 计算和反向的梯度传播，从而训练模型，同时也建议使用网络框架封装的 optimizer 完成参数更新过程。训练过程中记录模型在训练集和验证集上的损失，并绘图可视化。
5. **调参分析**：将训练好的模型在验证集上进行测试，以 Top 1 Accuracy(ACC) 作为网络性能指标。然后，对 dropout, normalization, learning rate decay, residual connection, network depth 进行调整，再重新训练、测试，并分析对模型性能的影响。
6. **测试性能**：选择你认为最合适的（例如，在验证集上表现最好的）一组超参数，重新训练模型，并在测试集上测试（注意，这理应是你的实验中**唯一**一次在测试集上的测试），并记录测试的结果（ACC）。

## 实验提交

本次实验截止日期为 **4 月 16 日 23:59:59**，需提交代码源文件及实验报告到邮箱：  
[proton00@mail.ustc.edu.cn](mailto:proton00@mail.ustc.edu.cn)，具体要求如下：

1. 全部文件打包在一个压缩包内，压缩包命名为 **【学号】- 姓名 - exp2.zip**
2. 代码仅包含 .py 文件，请勿包含实验中间结果（例如中间保存的数据集等），如果有多个文件，放在 src/ 文件夹内。

3. 代码中提供一个可以直接运行的并输出结果的 `main.py`，**结果包括训练集损失、验证集损失随 epoch 改变的曲线（保存下来）和测试集的 ACC。**
4. 代码中提供一个描述所有需依赖包的 `requirements.txt`，手动列入代码中用到的所有非标准库及版本或者使用 `pip freeze > requirements.txt` 命令生成。
5. 实验报告要求 pdf 格式，要求包含姓名、学号。内容包括简要的**实验过程**和**关键代码**展示，对超参数的**实验分析**，最优超参数下的训练集、验证集**损失曲线**以及测试集上的**实验结果**。

## 参考资料

---

往届同学的实验代码和报告：[https://github.com/hehaha68/USTC\\_2022Spring\\_Introduction-to-Deep-Learning](https://github.com/hehaha68/USTC_2022Spring_Introduction-to-Deep-Learning)

提供的 Lab2\_demo.ipynb