

# 深度学习导论2022.S Lab1

---

By ZhangYT 2022.3.17

## 一、实验目标

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、网络深度等超参数对分类性能的影响

## 二、代码结构

---data 数据文件夹，你需要将下载好的数据放在这个位置

-----test

-----train

-----val

---ImageNet.py Tiny ImageNet的DataLoader

---CNN.py CNN框架代码

---ResNet.py ResNet框架代码

---train.ipynb 训练代码

## 三、实验内容

为了方便同学们上手，第一次实验多数代码已给出，你可以选择直接阅读代码填空:)

当然，你们可以自己从头搭建，这样会加深理解(虽然不会有额外的分数)

P.S你们也可以使用TF,

### 数据集

考虑到ImageNet数据量过大，我选取了[Tiny ImageNet](#)。

Tiny Imagenet 有 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

### 数据加载

本部分工作内容已经在ImageNet.py中由TA完成，你可以参考在train.ipynb中的使用方法

### 数据预处理 Task0.1

你需要在train.ipynb完成 task 0.1 部分，来增广/正则化数据。(你也可以不用数据增广直接采用transform2)

## CNN框架 Task 1.1 Task 1.2

你需要补全 `CNN.py` 中的两个task

task1.1用于定义网络框架，请务必注意通道参数。友情提示，你需要在最后加全连接层。

task1.2 用于前向传播，这里非常简单，你只需要按照task1.1自己的定义即可。

## ResNet框架 Task 2.1 Task2.2 Task2.3

你需要补全 `ResNet.py` 中的三个task

为了便于理解，我已经写好了ResNetLayer的定义部分，你只需要在task2.1中写出前向传播即可

task2.2 task2.3类似CNN，只不过你需要按照ResNet的[定义](#)来写

## 网络参数 Task3.1-Task3.6

如果你希望自己来写训练过程，当然可以，你可以忽略下述task :)

你需要补全 `train.ipynb` 中的六个task

task3.1 用来更改学习率以及最大次数

task3.2 用来设定优化器，学习率调整，以及loss函数

task3.3 task3.4分别是validate test函数，你可以对应train来写

task3.5 用于early stopping

task3.6 用于调用 `fit` 函数，你只需要传入参数

## 五、提交内容

1. `CNN.py`
2. `ResNet.py`
3. `train.ipynb`
4. 报告，用来简单描述你做的fine tuning(dropout、normalization、learning rate decay、residual connection、网络深度等), 最后的test准确率，对比，以及Loss随着epoc改变的曲线(你可以用 `utils.py` 中的 `plot` 函数来画)

**压缩包命名** PB18XXX\_姓名\_Lab1.zip

在**2022.4.3 23:59:59 之前**提交到[deeplearning2022s@163.com](mailto:deeplearning2022s@163.com)