

数据增广

实验内容: 了解图像分类的数据增广方式, 练习常用的数据增广函数操作.

实验过程:

1) 在”实验 03 图像分类的迁移学习”的基础上, 思考数据增广的接口在代码中的位置, 选择 ResNet-18(或 VGG-16 模型), 准备重构代码.

2) 利用 torchvision 包, 实现随机 90 度旋转, 水平翻转, 上下翻转, 色调变化. [1][2]

3) 利用 albumentations 包, 实现模糊化和扭曲处理[3].

```
# 参考[3]
import albumentations as A
transform = A.Compose([
    A.OneOf([
        A.MotionBlur(p=0.2),
        A.MedianBlur(blur_limit=3, p=0.1),
        A.Blur(blur_limit=3, p=0.1),
    ], p=0.2),
    A.OneOf([
        A.OpticalDistortion(p=0.3),
        A.GridDistortion(p=0.1),
    ], p=0.2),
])
```

3) 自行实现自定义的三通道随机噪声(-R,R), 利用 floor 和 ceil 函数限制数值的向上和向下溢出. 其中 R 是预设的一个数值范围.

4) 打印出训练损失曲线图, 对比没有数据增广的情况.

5) 填写下表(放在实验报告中).

模型	训练准确率	测试准确率
ResNet-18 无数据增广		
ResNet-18 有数据增广		

6) 思考题: 黑白图像能不能用第(1)步色彩变化和第 3 步的方法? 怎么实现?

参考:

[1] Getting started with transforms v2

https://pytorch.org/vision/stable/auto_examples/transforms/plot_transforms_getting_started.html

[2] Torchvision Docs > Transforming and augmenting images > ColorJitter

<https://pytorch.org/vision/stable/generated/torchvision.transforms.v2.ColorJitter.html#torchvision.transforms.v2.ColorJitter>

[3] Defining a simple augmentation pipeline for image augmentation

<https://albumentations.ai/docs/examples/example/>