1 CIFAR10 数据集构建

jittor 已经内置了 CIFAR10 数据集，继承自 Dataset 类，直接创建即可将数据下载到本地的默认目录下。为了本任务和后续任务的可扩展性，我并没有直接继承 CIFAR10 数据集，而是复制并在 jittor 源码的基础上修改，以便实现后续的数据随机保留、过采样和增强。不过在基本的训练分类任务中，代码不需要修改，直接使用即可。所使用的训练集和测试集的划分也是基于 CIFAR 默认的测试集和训练集。

具体数据集扩展细节会在后文描述。

2 神经网络结构简述

由于本次任务旨在学习并使用国产深度学习框架，并不对精度有着细致的要求，所以我并没有追求神经网络结构的复杂性，而是选择了以往学习实践过程中较为常用的一个普通的卷积神经网络。在默认的参数下，神经网络的基本结构和参数如下。

```
Classifier
conv1: Conv(3, 64, (5, 5), (1, 1), (0, 0), (1, 1), 1, float32[64,], None, Kw=None, fan=None, i=None, bound=None)
bn1: BatchNorm(64, 1e-05, momentum=0.1, affine=True, is_train=True, sync=True)
conv2: Conv(64, 256, (5, 5), (1, 1), (0, 0), (1, 1), 1, float32[256,], None, Kw=None, fan=None, i=None, bound=None)
bn2: BatchNorm(256, 1e-05, momentum=0.1, affine=True, is_train=True, sync=True)
pool: MaxPool2d(
    _layer: Pool((2, 2), (2, 2), padding=(0, 0), dilation=None, return_indices=None, ceil_mode=False, count_include_pad=False, op=maximum)
)
fc1: Linear(6400, 256, float32[256,], None)
fc2: Linear(256, 96, float32[96,], None)
fc3: Linear(96, 10, float32[10,], None)
```

3 非平衡 CIFAR 数据集构建

为了方便后续的数据增强和过采样，我将数据集的构建过程封装在了一个成员函数中，以便后续的调用。在默认的情况下，CIFAR10 类的构造函数会调用该函数，以便在初始化时就将 label 在 0 到 4 范围内的数据直接删除，并将其保存在一个列表中，随后，再随机从这五类中各自取出 500 个数据即可。

4 非平衡数据集下的改进方法

在非平衡数据集下，模型的测试准确率确实出现了显著的下降，主要原因在于模型并没有学到数据缺失类的基本特征，从而在分类测试集对应样本时表现出了较差的效果，表现为整体准确率的降低。我尝试从两个角度解决这个问题，数据增强和对损失函数的改进。

4.1 数据增强

数据增强是一种常用的方法，它可以通过对原始数据进行一定的变换，从而生成新的数据，从而增加数据集的大小。数据增强的基本原则是在不改变原始数据集中某个样本的语义的情况下，通过对单个样本的重复选取，随机翻转、旋转等方法，在有限的数据量下尽可能增大模型的泛化能力。模型的泛化能力来源于数据增强并不改变样本本身的语义，比如说，一个飞机在翻转、旋转后，应当还是飞机，而不是其他的东西。

本实验的数据增加基于 numpy 的 flip 和 rot90 函数，通过一次性随机采样 500 个样本，对他们采用不同的方式增强并放到数据集中来实现。

4.2 损失函数改进

我采用的另一种思路是通过使用带权交叉熵损失函数引导模型用更符合逻辑的步长进行梯度下降，这一方法可以结合数据增强一同使用。对于缺失的数据集样本，由于模型不能够从其中学到足够的特征，所以应该在交叉熵函数中给予更大的权重，从而引导模型更好地梯度下降。

本实验的损失函数改进是在 jittor 自身的交叉熵损失函数的基础上进行的，方法是将其中的权重向量进行修改，使得缺失的类别的权重更大。具体的权重需求可以自行调整，默认的情况下为 10.

5 实验结果与总结

数据指标	原始数据集	非平衡	过采样	数据增强
Loss	0.7982	2.0931	1.6027	1.3902
Acc.	77.45%	57.32%	61.51%	63.98%

表 1: 使用交叉熵损失函数在不同数据处理方式下的实验结果

数据指标	原始数据集	非平衡	过采样	数据增强
Loss	0.8683	1.8712	1.6143	1.4278
Acc.	75.33%	58.07%	61.43%	63.23%

表 2: 使用带权交叉熵损失函数在不同数据处理方式下的实验结果

可以发现，带权交叉熵损失函数在一定程度上确实可以提升非平衡数据集条件下的精确度，但是在使用了数据增强的方法下，带权交叉熵损失函数反而会对模型精度带来一定的下降。另外，对于原始没有缺失的数据集，带权交叉熵损失函数会对模型的精度带来比较大的损失。

整体来说，对于缺失的数据集，并没有一个非常完美的解决方案，带权交叉熵函数在这一任务上的表现较差，部分原因可能是 jittor 自身实现的交叉熵损失函数本身不太稳定。不过，就目前的实验结果而言，数据增强还是能起到一定的作用。

6 其他说明

本实验默认的训练参数已在 `parser.py` 文件中配置好，测试代码完整性可以运行 `python task2.py --debug`