深度学习作业二报告

520030910342 柳纪宇

1. 数据集准备

CIFAR-10 数据集包括了 60000 张 32x32 的有色图，一共分为 10 个类，每个类有 6000 张图片。其中 50000 张图片为训练样本，剩下的 10000 张图片为训练样本。

下载得到的 CIFAR 数据集的文件类型为 Python 中的"pickled" 对象，因此我们需要使用官网中提供的 unpickle 函数将其转换为字典类型，从而从中提取出图片和对应的标签。

为了区分训练集和测试集，我先编写了父类 CIFAR，再编写了继承该父类的两个子类 Train_CIFAR 和 Test_CIFAR。

在父类 CIFAR 的 ___init___ 函数中，我遍历了 5 个训练数据集，对其中的图片文件进行 reshape 和 transpose 操作后将其加入列表 d_list 中，相应地将这些图片对应的标签加入列表 l_list。将这两个列表转换成 numpy 格式后保存至 self.trian_data 和 self.train_label 中。对训练数据集进行类似的操作，得到的数据和标签存储于 self.test_data 于 self.test_label 中。

子类 Train_CIFAR 继承了父类的 ___init___ 函数，在 ___getitem___ 函数中该 DataLoader 返回了 self.train_data 和 self.train_label 的第 i 个转换成 jittor 使用的 Var 类型的索引值。

子类 Test_CIFAR 也与此类似，不同点在于用于索引的列表更改成了 self.test_data 和 self.test_label。

2. 模型训练

2.1. 模型准备

我为分类任务编写的模型是一个三层卷积神经网络。

该模型由两个中间层和一个全连接层组成。第一个中间层由一个 Conv 层、一个 BatchNorm 层、一个 Relu 层和一个 MaxPool 层组成。第二个中间层的构造类似第一个，也由如上这些层构成，具体参数上的不同可参见 model.py。全连接层将中间层 2 输出的结果转换为一个一维矩阵传入，输出结果为一个 10 维向量，向量中数值最大的维度即预测得到的类。

2.2. 模型训练与测试

本次实验中的训练使用的方法是遍历训练集、计算误差、进行梯度回传更新参数。训练的 epoch 数为 50，学习率为 0.01，batch 大小为 20，优化器为 nn.SGD，使用的误差函数为交叉熵函数。每 100 次迭代输出一次平均误差，每个 epoch 结束后将该 epoch 的总误差加入列表 Train_Loss_List 中。

本次实验使用的测试方法为遍历测试集、计算预测准确率、更新最佳准确率。使用的 batch 大小为 1，每 1000 次迭代输出一次平均准确率，每个 epoch 结束后将该 epoch 的平均准确率加入列表 Test_Acc_List 中。

使用该模型训练得到的最佳准确率为 0.5766。使用 matplotlib 绘制 epoch-train_loss 折线图和 epoch-test_acc 折线图，分别存储到路径'./output/Train_loss.jpg'和'./output/Test_acc.jpg'。训练得到的模型存储到路径'./trained_model/model.pkl'。

3. 训练集切分

在这个部分，我对训练集中标签为 (0,1,2,3,4) 的数据进行下采样，保留原数据中的十分之一，将其与标签为 (5,6,7,8,9) 的数据进行合并生成新数据集 Ten_Percent_Train_CIFAR，其训练样本个数为 27500。

使用该数据集，保留与之前相同的超参数设置和模型设置进行模型训练，每个 epoch 记录一次训练总误差和测试准确率。最终得到的最佳准确率为 0.5705，可以看到相较未切分前模型准确率略有下降。使用 matplotlib 绘制 eopch-trian_loss 折线图和 epoch-test_acc 折线图，分别存储到路径'./output/masked_Train_loss.jpg'和'./output/masked_Test_acc.jpg'。

训练得到的模型存储到路径'./trained_model/masked_model.pkl'。

4. 模型优化

4.1. 数据增强

为了在 label 小于 5 的训练集数据只有原先十分之一的情况下仍然保持较高的准确率，我选择使用数据增强的方法来扩充数据集，以达到提高模型准确率的效果。

我一共设计了七种数据增强方式，分别为随机改变亮度、随机交换 RGB 通道、随机改变对比度、随机改变饱和度、随机转变为灰度图、随机翻转、随机裁剪。具体可见根目录下的 utils.py 文件。

在数据集加载过程中，若参数 data_augmentation 的值为 True，则选取约 25% 标签小于 5 的数据进行数据增强，将增强后的数据及其标签加入训练集数据中。

4.2. 模型改进

为了提高模型准确率，我参考 jittor 官方文档编写了 Resnet 模型，并在训练过程中使用 Resnet18 替代之前使用的 CNN 模型。在此基础上，我还引入了 jittor 中的预训练模型以提高模型准确率。

由于算力限制和对模型收敛性的估计，本次训练仅训练 20 个 epoch，可以看到最终测试准确率同样得到了收敛。

4.3. 训练结果

最终得到的最佳准确率为 0.6658，可以看到相较于先前模型准确率得到了很大的提升。使用 matplotlib 绘制 eopch-trian_loss 折线图和 epoch-test_acc 折线图，分别存储到路径'./output/enhanced_Train_loss.jpg'和'./output/enhanced_Test_acc.jpg'。

训练得到的模型存储到路径'./trained_model/enhanced _model.pkl'。

4.4. 总结

在本次实验中，我在 CIFAR 数据集的基础上使用自定义 CNN 模型进行图像分类实验，得到了%57.66 的准确率。在进行数据集切分后，准确率相较原先略有下降（降至%57.05）。为了提高模型准确率，我在引入数据增强的同时使用预训练后的 Resnet18 模型进行训练，最终得到的准确率结果为%66.58。