基于MindSpore在LeNet-5网络训练Mnist数据集

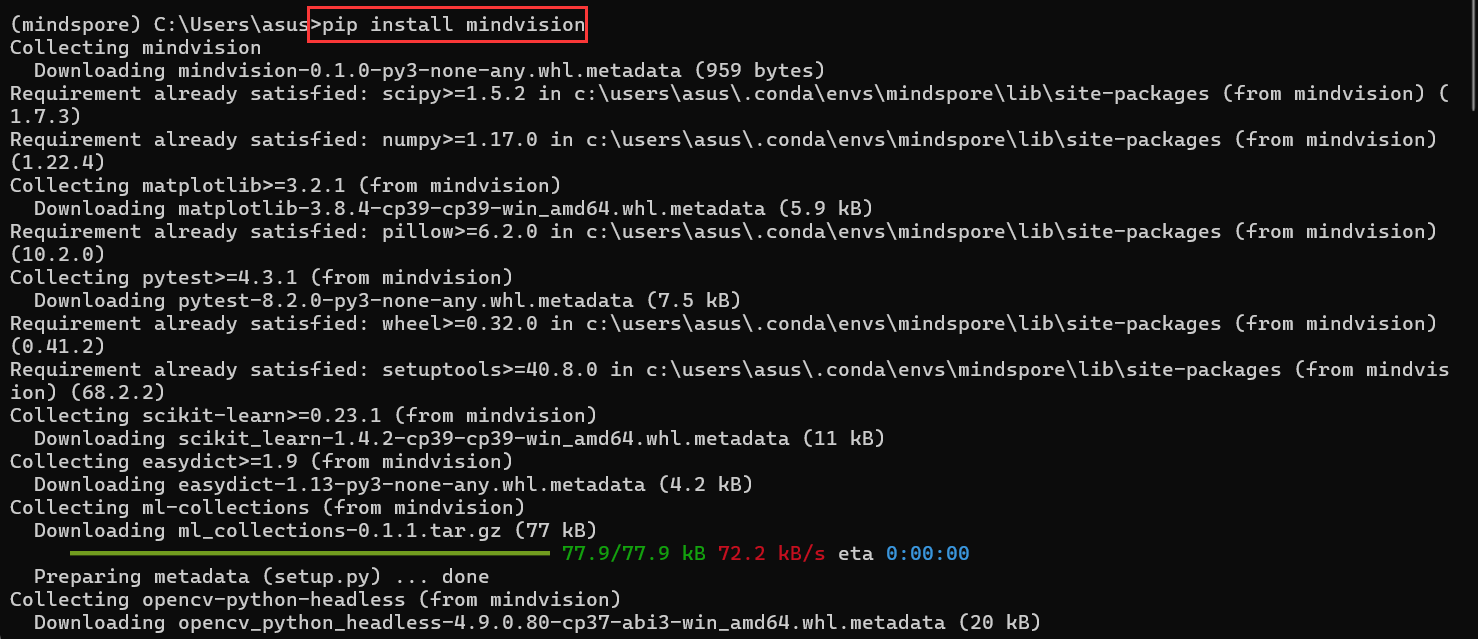
1. **实验目的**

MindSpore是一个全场景深度学习框架，旨在实现易开发、高效执行、全场景覆盖三大目标，其中易开发表现为API友好、调试难度低，高效执行包括计算效率、数据预处理效率和分布式训练效率，全场景则指框架同时支持云、边缘以及端侧场景。本篇报告就是基于MindSpore中的LeNet5网络模型，数据集Mnist进行训练和测试。

1. **实验过程**

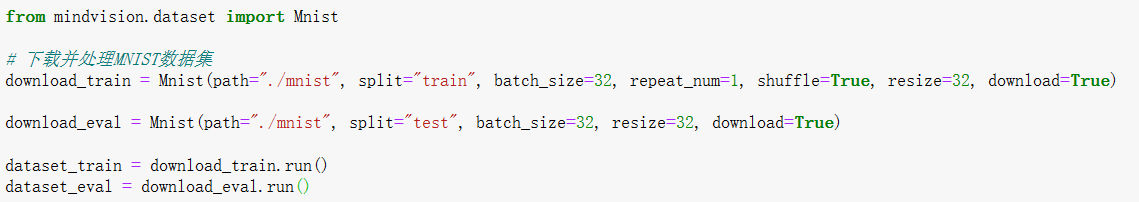
2.1 环境配置

此次训练代码依赖mindvision，可使用命令pip install mindvision安装。



2.2 数据集的导入

将mindvision安装完成以后，就可以进行数据集的导入：



参数说明：

path：数据集路径。

split：数据集类型，支持train、 test、infer，默认为train。

batch\_size：每个训练批次设定的数据大小，默认为32。

repeat\_num：训练时遍历数据集的次数，默认为1。

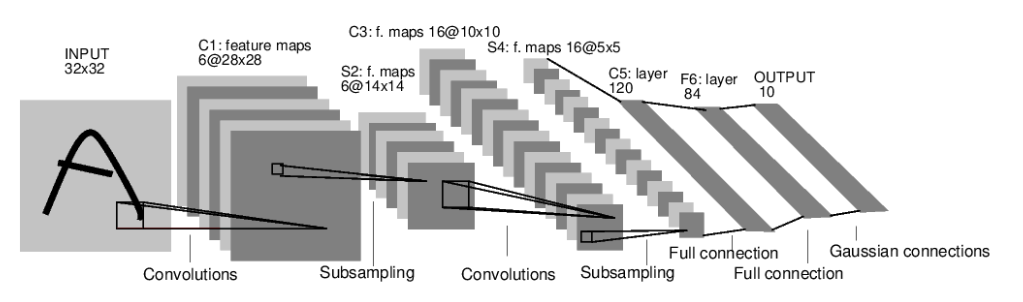
shuffle：是否需要将数据集随机打乱（可选参数）。

resize：输出图像的图像大小，默认为32\*32。

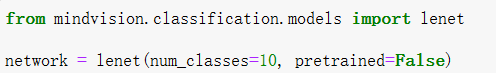
download：是否需要下载数据集，默认为False。

2.3 创建模型

按照LeNet的网络结构，LeNet除去输入层共有7层，其中有2个卷积层，2个子采样层，3个全连接层。

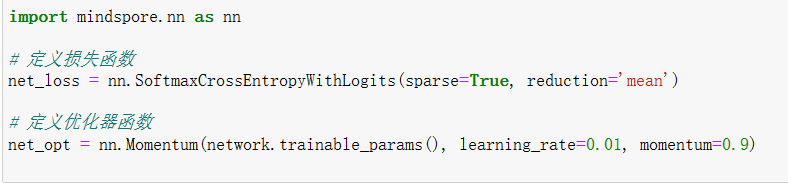


MindSpore Vision套件提供了LeNet网络模型接口lenet，定义网络模型如下：



2.4 定义损失函数和优化器

要训练神经网络模型，需要定义损失函数和优化器函数。损失函数这里使用交叉熵损失函数SoftmaxCrossEntropyWithLogits。优化器这里使用Momentum。

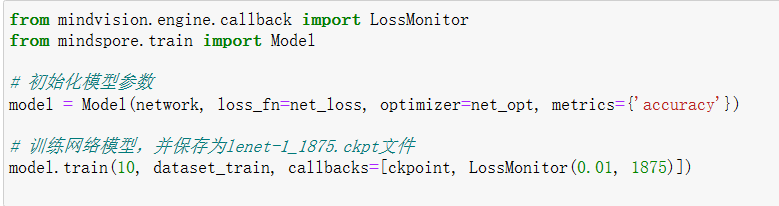


2.5 训练及保存模型

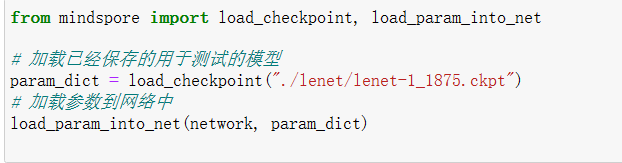
在开始训练之前，MindSpore需要提前声明网络模型在训练过程中是否需要保存中间过程和结果，因此使用ModelCheckpoint接口用于保存网络模型和参数，以便进行后续的Fine-tuning（微调）操作。



通过MindSpore提供的model.train接口可以方便地进行网络的训练，LossMonitor可以监控训练过程中loss值的变化。

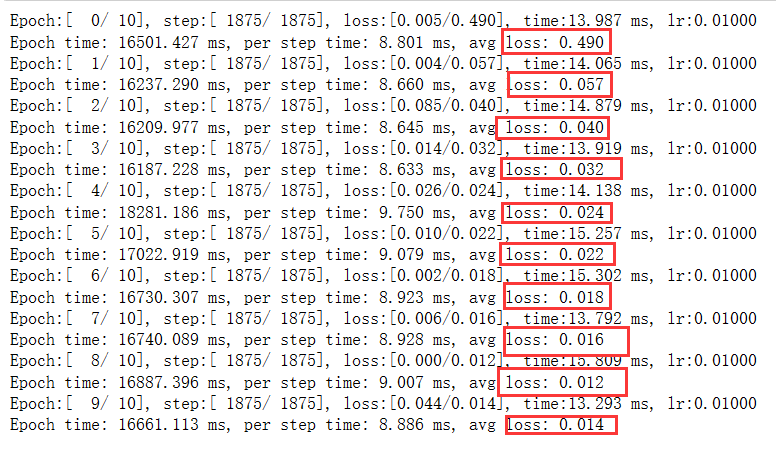


2.6 加载模型

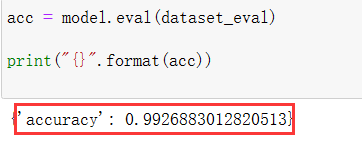


1. **实验结果**

训练过程中会打印loss值，loss值会波动，但总体来说loss值会逐步减小，精度逐步提高。每个人运行的loss值有一定随机性，不一定完全相同。



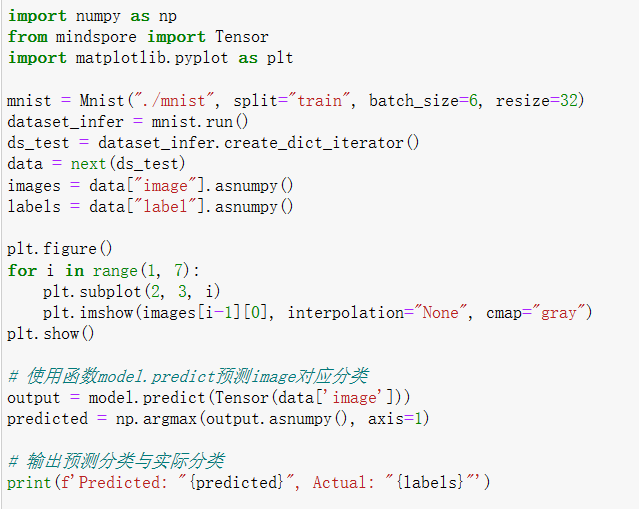
通过模型运行测试数据集得到的结果，验证模型的泛化能力：使用model.eval接口读入测试数据集。使用保存后的模型参数进行推理。



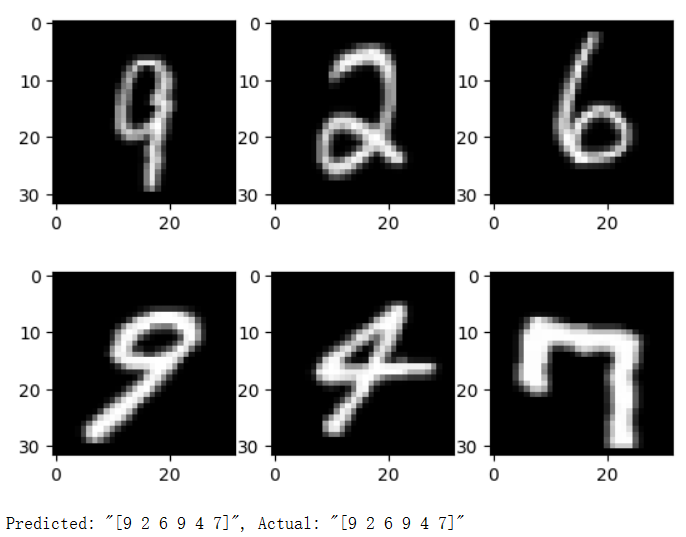
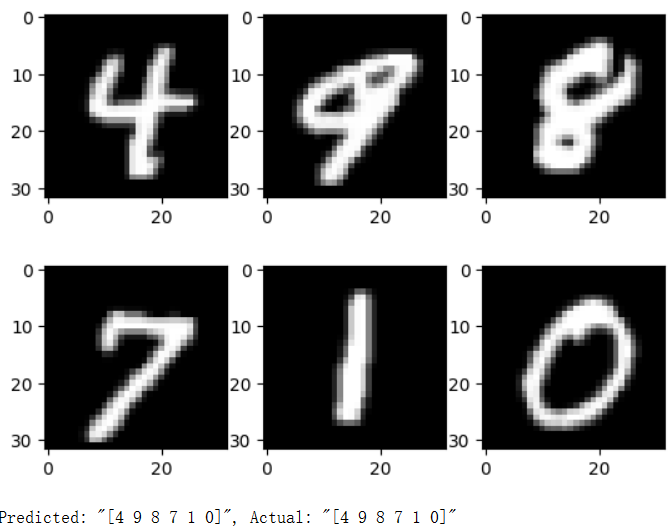
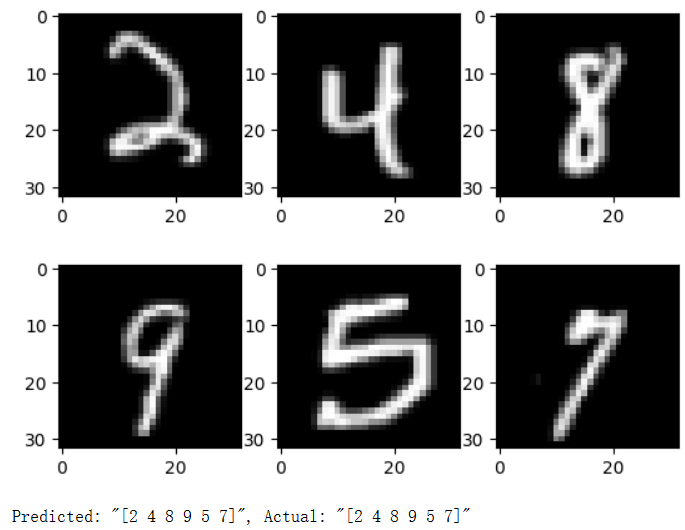
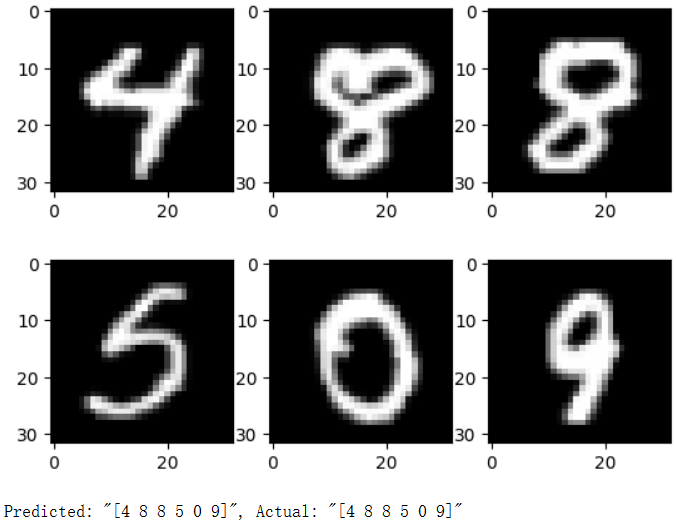
可以在打印信息中看出模型精度数据，示例中精度数据达到99%，模型质量良好。随着网络迭代次数增加，模型精度会进一步提高。

我们使用生成的模型进行单个图片数据的分类预测，具体步骤如下：

被预测的图片会随机生成，每次运行结果可能会不一样。代码使用了Tensor模块，阅读更多有关[张量Tensor](https://www.mindspore.cn/tutorials/zh-CN/r1.7/beginner/tensor.html)的信息。



下面是四次的实验结果：

从这四次测试结果可以看出，此次训练以后的准确度还是相当高的。