1. 实现说明

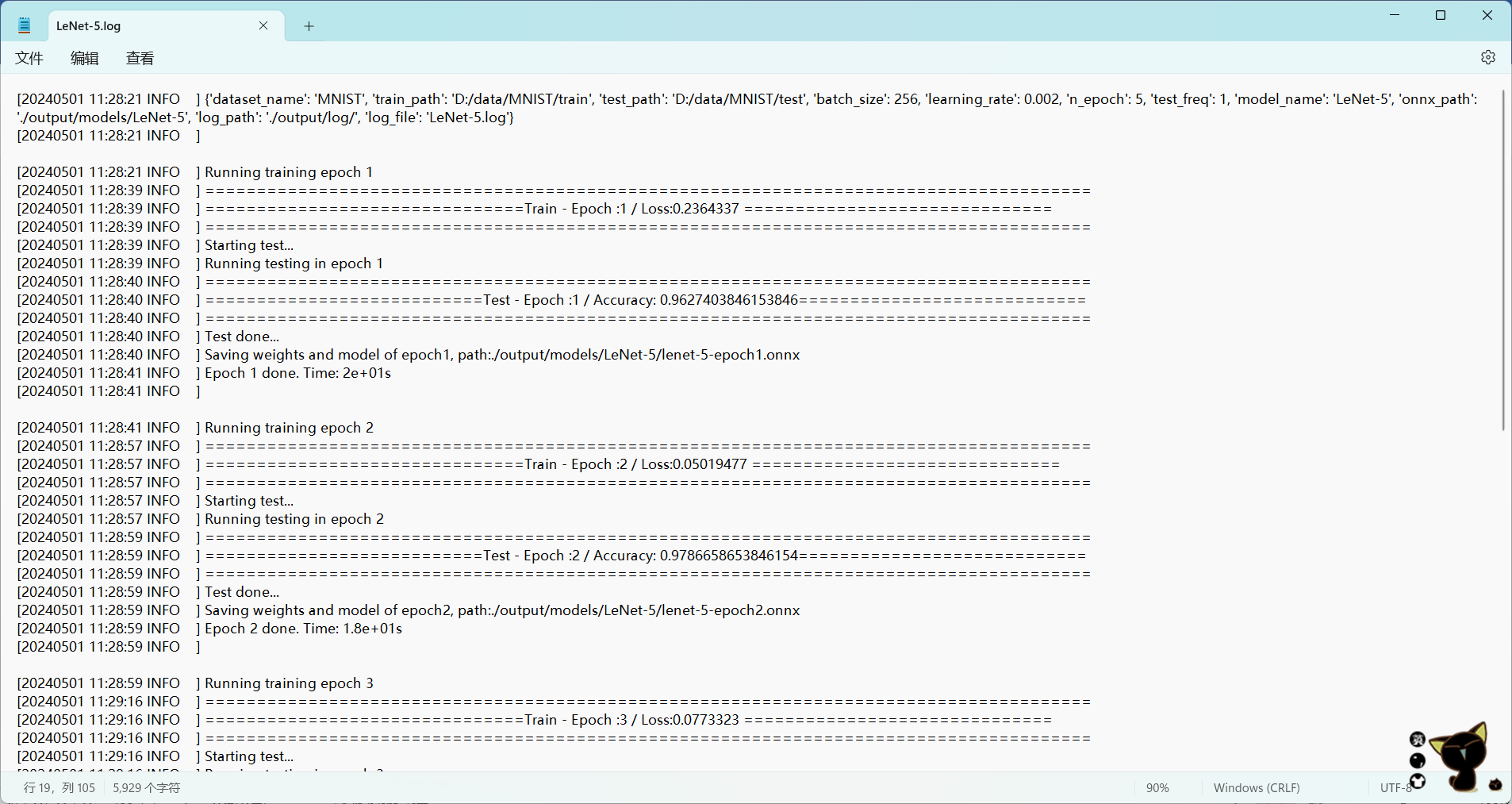
代码中网络模型的相关代码均使用了mindspore框架。辅助功能中使用了logging日志模块，numpy辅助计算。

首先，我们定义了一个参数字典，用于保存数据集路径，批处理大小，学习率，周期数等训练和测试过程中需要用到的参数，主要是方便日后修改代码可以在这个字典中做到统一修改，而不需要去优化器代码段单独修改学习率一个超参数。

然后我们使用mindspore.dataset中的方法创建数据集加载器。从LeNet-5的类方法中创建模型实例，并在mindspore.nn中创建交叉熵损失函数和准确率指标，作为训练和测试的衡量指标。使用mindspore.train.Model方法将网络模型和优化器，损失函数联系在一起。在训练和测试阶段只需要调用Model.train()和Model.eval()传入周期数和数据集就可以实现自动化训练和测试。Mindspore中的LossMonitor()函数并不能满足记录日志的需求，在这个函数的内部实现中，它是直接获得了网络模型在训练中的loss并使用print()函数输出。所以我重写了该方法，裁剪了获取loss的代码片段，并记录在日志中。

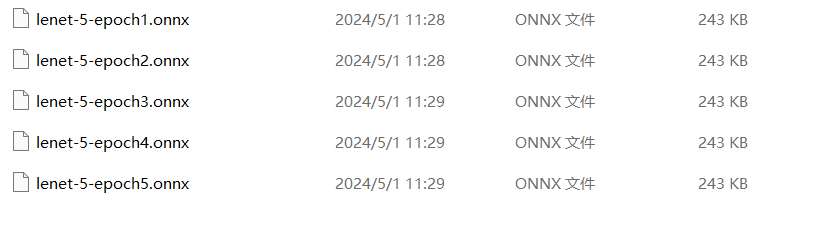
最后，在测试阶段，使用Model.eval()可以直接获取准确率这个指标，也很方便的进行日志记录。测试阶段结束后使用mindspore.train.serialization.export进行模型保存，导出为onnx格式。

1. 运行结果及说明
2. 日志文件



第一行是记录的超参数字典的内容，后面开始记录每一个epoch中训练的loss值和测试的准确度以及对应epoch下模型保存的路径。

1. 模型保存



1. 结果说明

代码运行结果正常，可以完成LeNet-5在MNIST数据集上的训练和测试功能。在整个流程中训练的loss值稳步下降，测试的准确度稳步上升。第5个epoch的测试准确度由开始的0.9627上升到0.9863。说明代码功能正常。