之前我跟您深入讨论过通过神经网络来进行固体力学的参数反演，我最终选取FNO（Fourier Neural Operator）网络来实现参数反演，我想让您生成一份代码，下面我介绍一下背景及要求：

一、数据文件

1. 实验模拟数据我们存放在Excel表中，其名字为”Data.xlsx”；
2. 我们会将其和Python代码文件置于同一个文件夹中。

二、数据格式

1、由于数据是等间距位移采样，因此我们并没有记录位移数据，也就是x轴值，我们只记录载荷值，也就是Y轴值，换句话说，在Excel表中，我们默认同一列的数据的它的x轴是一样的。

2、Excel 表中，每行第一列和第二列分别对应的是力学参数屈服强度和硬化指数,它作为输出。

3、Excel表中，每行从第三列到第204列对应的是载荷数值，代表的是载荷-位移曲线，从

第三列到第103列对应的是加载阶段，第104列到204列对应的是卸载阶段，它是作为输入。

1. Excel有121行，也就是采样了121条数据。121条数据中要留取一部分数据作为测试数据，剩余的留下训练，比例按照你认为的合适去选取。
2. 代码要求
3. 代码的逻辑清晰、可读性强；
4. 代码模块化、健壮性强、易于维护；
5. 代码中训练参数设置（如训练轮次、批大小、学习率等等）、加载读取数据、训练、推理、测试要分离
6. 代码中注释用中文，要合理恰当得体，使得代码可读性强，易于后期维护
7. 神经网络组件可以适当模块化分离方便后期修改。
8. 在考虑模型表现前提下，可以对输入输出数据进行适当标准化或归一化处理，并在代码中保留反变换逻辑以便还原真实物理量。
9. 请实现模型保存与加载机制，支持训练过程中保存最佳模型，支持断点续训，并在测试阶段自动加载。
10. 请在训练过程中记录loss曲线，适当保存训练日志，并支持可视化预测结果与真实值的对比，便于调试和分析。
11. 设置随机种子以确保结果可复现；
12. FNO网络结构可参考论文设置合理默认值，并将关键超参数（如层数、模数、通道数）提取为可配置参数
13. 数据为1D时间序列（载荷-位移曲线），请实现适合1D序列的FNO网络。

请根据以上内容深度思考，尽量生成生成一份Bug少、高质量的Python代码，辛苦了。我会上传FNO的论文《Fourier Neural Operator for Parametric Partial Differential Equations》以及数据的EXcel表，方便你思考。现在请您大显身手吧！  
  
  
之前我与您深入讨论过通过神经网络进行固体力学的参数反演，最终选择FNO（Fourier Neural Operator）网络实现参数反演。请生成一份高质量的Python代码，满足以下背景和要求：

一、数据文件

1. 实验模拟数据存储在Excel表“Data.xlsx”中。

2. Excel文件与Python代码文件位于同一文件夹。

二、数据格式

1. 数据为等间距位移采样，仅记录载荷值（Y轴），不含位移值（X轴）。Excel表中，同一列数据的X轴位移值相同。

2. Excel表每行数据格式：

- 第1列：屈服强度（输出，浮点数）。

- 第2列：硬化指数（输出，浮点数）。

- 第3至204列：载荷-位移曲线（输入，浮点数），其中第3至103列为加载阶段，第104至204列为卸载阶段。

3. Excel表包含121行数据（121条样本）。请将数据划分为训练集和测试集，默认比例为80%训练、20%测试（可根据任务复杂度调整，需在代码注释中说明依据）。

4. 数据可能存在噪声，建议在数据加载时添加简单的数据清洗逻辑（如检查缺失值或异常值）。

三、代码要求

1. 代码结构：逻辑清晰、模块化、可读性强、易于维护。使用函数或类将数据加载、预处理、模型定义、训练、推理、测试、可视化等功能分离。

2. 数据预处理：

- 为了加强模型表现，对输入（载荷值）和输出（屈服强度、硬化指数）可以进行适当归一化，保留反变换逻辑以还原真实物理量。

- 检查数据完整性（如每行列数是否为204，是否存在缺失值）。

3. FNO网络：

- 参考论文《Fourier Neural Operator for Parametric Partial Differential Equations》

- 数据为1D时间序列（载荷-位移曲线），请实现适合1D序列的FNO网络。

- 关键超参数（如层数、模数、通道数）提取为可配置参数。

- 确保输入数据格式适配FNO。

4. 训练参数：

- 设置随机种子确保结果可复现。

5. 模型保存与加载：

- 保存最佳模型（基于验证集最低MSE损失），支持断点续训。

- 每隔100个epoch保存一次检查点，保存路径为“./checkpoints/”。

- 测试阶段自动加载最佳模型。

6. 日志与可视化：

- 记录训练和验证的loss曲线，保存为csv文件（路径：./logs/loss.csv）。

- 可视化预测结果与真实值的对比，绘制载荷-位移曲线的预测与真实曲线（使用Matplotlib，生成折线图）。

- 计算并输出测试集的评估指标（MSE、MAE、R²）。

7. 依赖与环境：

- 基于PyTorch实现，支持GPU/CPU运行。

- 使用DataLoader优化数据加载，减少内存占用。

8. 健壮性与错误处理：

- 处理Excel文件读取失败、数据格式错误等异常情况。

- 验证输入数据维度一致性，抛出友好错误提示。

9. 注释与文档：

- 使用中文注释，清晰、简洁、得体，说明每个模块的功能和关键步骤。

- 提供简单的代码使用说明（如运行方式、依赖安装）。

请根据以上内容生成一份Bug少、高质量的Python代码，注释清晰，结构模块化，满足固体力学参数反演需求。感谢您的努力！

之前我与您深入讨论过通过神经网络进行固体力学的参数反演，最终选择FNO（Fourier Neural Operator）网络实现参数反演。请生成一份高质量的Python代码，满足以下背景和要求：

一、数据文件

1. 实验模拟数据存储在Excel表“Data.xlsx”中。

2. Excel文件与Python代码文件位于同一文件夹。

二、数据格式

1. 数据为等间距位移采样，仅记录载荷值（Y轴），不含位移值（X轴）。Excel表中，同一列数据的X轴位移值相同。

2. Excel表每行数据格式：

- 第1列：屈服强度（输出，浮点数）。

- 第2列：硬化指数（输出，浮点数）。

- 第3至204列：载荷-位移曲线（输入，浮点数），其中第3至103列为加载阶段，第104至204列为卸载阶段。

3. Excel表包含121行数据（121条样本）。请将数据划分为训练集和测试集，默认比例为80%训练、20%测试（可根据任务复杂度调整，需在代码注释中说明依据）。

4. 数据可能存在噪声，建议在数据加载时添加简单的数据清洗逻辑（如检查缺失值或异常值）。

三、代码要求

1. 代码结构：逻辑清晰、模块化、可读性强、易于维护。使用函数或类将数据加载、预处理、模型定义、训练、推理、测试、可视化等功能分离。

2. 数据预处理：

- 为了加强模型表现，对输入（载荷值）和输出（屈服强度、硬化指数）可以进行适当归一化，保留反变换逻辑以还原真实物理量。

- 检查数据完整性（如每行列数是否为204，是否存在缺失值）。

3. FNO网络：

- 参考论文《Fourier Neural Operator for Parametric Partial Differential Equations》

- 数据为1D时间序列（载荷-位移曲线），请实现适合1D序列的FNO网络。

- 关键超参数（如层数、模数、通道数）提取为可配置参数。

- 确保输入数据格式适配FNO。

4. 训练参数：

- 设置随机种子确保结果可复现。

5. 模型保存与加载：

- 保存最佳模型（基于验证集最低MSE损失），支持断点续训。

- 每隔100个epoch保存一次检查点，保存路径为“./checkpoints/”。

- 测试阶段自动加载最佳模型。

6. 日志与可视化：

- 记录训练和验证的loss曲线，保存为csv文件（路径：./logs/loss.csv）。

- 可视化预测结果与真实值的对比，绘制载荷-位移曲线的预测与真实曲线（使用Matplotlib，生成折线图）。

- 计算并输出测试集的评估指标（MSE、MAE、R²）。

7. 依赖与环境：

- 基于PyTorch实现，支持GPU/CPU运行。

- 使用DataLoader优化数据加载，减少内存占用。

8. 健壮性与错误处理：

- 处理Excel文件读取失败、数据格式错误等异常情况。

- 验证输入数据维度一致性，抛出友好错误提示。

9. 注释与文档：

- 使用中文注释，清晰、简洁、得体，说明每个模块的功能和关键步骤。

- 提供简单的代码使用说明（如运行方式、依赖安装）。

请根据以上内容生成一份Bug少、高质量的Python代码，注释清晰，结构模块化，满足固体力学参数反演需求。感谢您的努力！