

這段程式碼展示了如何使用 `PyTorch Lightning` 進行高層次的模型訓練和管理。
具體步驟包括：

1. **定義 `PyTorch Lightning` 模型**：

- 定義了一個多層感知器（MLP）模型 `MultiLayerPerceptron`，該模型包含多個隱藏層，並使用 `torchmetrics` 來計算準確率。
- 實現了模型的前向傳播、訓練步驟、驗證步驟和測試步驟，並設置了優化器。

2. **設置數據加載器**：

- 定義了一個 `PyTorch Lightning` 的數據模塊 `MnistDataModule`，用於下載、準備和加載 MNIST 數據集。
- `MnistDataModule` 負責將數據集分為訓練集、驗證集和測試集，並提供相應的數據加載器。

3. **訓練模型**：

- 使用 `pytorch_lightning.Trainer` 來訓練模型，並設置模型檢查點（`ModelCheckpoint`）來保存最好的模型。
- 根據是否有 GPU，配置相應的訓練器來進行訓練。

4. **評估模型**：

- 在訓練後使用 `trainer.test()` 來評估模型的性能。
- 使用 `TensorBoard` 來跟蹤模型的訓練過程和性能。

5. **重新訓練和測試**：

- 從之前保存的檢查點重新加載模型，進行額外的訓練和測試。

這段程式碼展示了如何使用 **PyTorch Lightning** 提高模型開發效率，簡化模型訓練和評估過程。