

這段程式碼主要是使用 `PyTorch-Ignite` 來簡化 `PyTorch` 模型的訓練和評估流程。以下是這段程式碼的主要步驟和意義：

1. **\*\*設置模型和數據\*\***：

- 定義了用於 `MNIST` 數據集的模型，包括數據預處理和數據加載。
- ``get_model()`` 函數創建了一個多層感知器模型，包含隱藏層和最終的 `Softmax` 輸出層。

2. **\*\*建立訓練和評估引擎\*\***：

- 使用 ``create_supervised_trainer()`` 和 ``create_supervised_evaluator()`` 函數創建了訓練和評估引擎。這些引擎負責模型的訓練和驗證過程，並計算相應的指標（如準確率和損失）。

3. **\*\*事件處理和日誌記錄\*\***：

- 為訓練過程設置了事件處理器，這些處理器在特定事件發生時被觸發，比如每 100 次迭代記錄一次損失，每個 `epoch` 完成時記錄一次驗證結果。

4. **\*\*模型檢查點和最佳模型保存\*\***：

- 設置了模型檢查點，以便在訓練過程中保存模型狀態，並能在訓練中斷後恢復。
- 設置了根據驗證準確率保存最佳模型的處理器，以便後續的預測或再訓練。

5. **\*\*TensorBoard 記錄和可視化\*\***：

- 使用 ``TensorboardLogger`` 記錄訓練過程中的損失和驗證指標，並通過 `TensorBoard` 可視化這些指標。

總結來說，這段程式碼展示了如何利用 **PyTorch-Ignite** 簡化 **PyTorch** 中常見的訓練和評估工作，包括設置訓練和驗證引擎、管理訓練過程、保存模型檢查點、記錄訓練過程中的指標並使用 **TensorBoard** 進行可視化。