

這段程式碼展示了如何從頭開始實現一個多層感知器（MLP）來處理 MNIST 手寫數字數據集。以下是每個主要部分的總結：

1. **套件檢查**：檢查所需的 Python 套件版本是否符合要求。

2. **數據載入與處理**：

- 從 OpenML 載入 MNIST 數據集，將圖像數據標準化到 $[-1, 1]$ 範圍。
- 顯示了每個數字類別的第一個圖像，並顯示了數字"7"的 25 個不同版本。
- 將數據集劃分為訓練集、驗證集和測試集。

3. **模型定義**：

- 定義了一個多層感知器（MLP）模型，包括隱藏層和輸出層，並使用 Sigmoid 激活函數。
- 定義了前向傳播（forward）和反向傳播（backward）方法，用於計算模型的預測和梯度。

4. **訓練過程**：

- 實現了一個小批量生成器（minibatch generator）來處理訓練數據。
- 定義了一個訓練函數來進行模型訓練，並在每個時期計算訓練和驗證集上的損失和準確度。

5. **性能評估**：

- 繪製了訓練過程中的損失和準確度變化。
- 在測試集上評估了模型的最終性能，並顯示了一些錯誤分類的圖像和預測結果。

這段程式碼涵蓋了從數據處理、模型構建、訓練到性能評估的完整流程，展示了如何用 **Python** 實現一個基本的神經網絡模型來解決分類問題。