

這段程式碼展示了如何使用 **PyTorch** 進行深度學習模型的實作，包括基本的數據處理、模型建立、訓練、以及自定義層。以下是程式碼的主要內容及其意義：

#### 1. **PyTorch 基本操作**：

- 使用 **PyTorch** 的張量操作（如加法、減法、乘法）來計算簡單的數學公式。
- 展示了如何設置 `requires_grad` 屬性以便追蹤梯度計算。

#### 2. **梯度計算**：

- 通過 `loss.backward()` 方法計算損失函數對模型參數的梯度。
- 確保了梯度計算的正確性，並展示了如何使用這些梯度來進行參數更新。

#### 3. **模型建構與訓練**：

- 使用 `nn.Sequential` 簡單地構建多層模型。
- 配置模型的層、初始化方法和優化器。
- 展示如何訓練一個 XOR 問題的模型，包括訓練過程中的損失和準確率的可視化。

#### 4. **自定義模型和層**：

- 創建了 `MyModule` 類來構建自定義的神經網絡模型。
- 實現了 `NoisyLinear` 類來添加噪聲到線性層，以檢測模型在訓練過程中的穩健性。

#### 5. **可視化**：

- 使用 `matplotlib` 進行訓練過程中的損失和準確率的可視化。
- 使用 `mlxtend` 庫中的 `plot_decision_regions` 函數來可視化分類模型的決策邊界。

總體來說，這段程式碼展示了 **PyTorch** 的核心功能，包括自動微分、模型建構與訓練、以及如何自定義模型和層。這些步驟對於理解和使用 **PyTorch** 進行深度學習任務是非常重要的。