

這段程式碼展示了如何使用 **PyTorch** 和 **Scikit-Learn** 來進行機器學習模型的建立、訓練與評估。下面是每個部分的總結：

1. **\*\*套件版本檢查\*\***：

- 檢查了所需的 **Python** 套件版本是否符合預期。

2. **\*\*線性回歸模型\*\***：

- 使用 **PyTorch** 手動實現了線性回歸模型，並通過梯度下降法來更新參數。圖形顯示了訓練數據點和預測的線性回歸直線。

3. **\*\*使用 `torch.nn` 和 `torch.optim` 模組進行模型訓練\*\***：

- 使用 `torch.nn.Linear` 類來建立線性回歸模型，並使用 **SGD** 優化器進行訓練。這段程式碼展示了如何利用 **PyTorch** 的高層 **API** 來簡化模型的訓練過程。

4. **\*\*建立多層感知機 (MLP) 進行 Iris 數據集的分類\*\***：

- 使用 `load_iris` 加載 **Iris** 數據集，並將其分為訓練集和測試集。建立了一個具有隱藏層的 **MLP** 模型來進行分類。訓練過程中記錄了損失和準確率的變化，並通過圖形展示了這些指標的變化。

5. **\*\*評估模型\*\***：

- 在測試集上評估了訓練好的模型，並計算了準確率。展示了如何保存和重新加載訓練好的模型。

6. **\*\*選擇激活函數\*\***：

- 討論了不同的激活函數，包括邏輯函數 (**Logistic function**)、**Softmax** 函

數、雙曲正切函數 (Tanh) 和修正線性單元 (ReLU)。提供了這些函數的計算示例和圖形表示。

總體來說，這段程式碼涵蓋了從模型建立、訓練、評估到使用不同激活函數的基本概念。