

這段程式碼介紹了感知器（Perceptron）與自適應線性神經元（Adaline）的實作範例，並展示了如何在 Python 中訓練機器學習演算法，尤其是二元分類任務。

這裡的核心概念包括：

1. **感知器**：一個早期的機器學習模型，用於解決二元分類問題。它通過學習權重和偏差來正確分類數據。
2. **Adaline**：一個基於梯度下降的感知器變體，使用損失函數（均方誤差）來優化模型。

感知器（Perceptron）實作

在 `Perceptron` 類別中，主要參數如下：

- `eta` 是學習率，決定每次權重更新的步長。
- `n_iter` 是訓練的迭代次數。
- `random_state` 用於隨機初始化權重。

感知器通過 `fit()` 方法進行訓練，主要過程包括：

1. 計算每個數據點的預測結果。
2. 根據預測錯誤來更新權重和偏差。
3. 每次錯誤都會計入錯誤列表，供後續分析。

Adaline 實作

Adaline 使用梯度下降來優化，和感知器類似，但它計算的是連續輸出的誤差，而不是僅僅依據預測錯誤來更新權重。

在 `AdalineGD` 類別中，訓練過程中：

1. 計算每個樣本的輸入加權和。
2. 通過激活函數（在這裡是線性函數）計算輸出。
3. 使用均方誤差來更新權重，並記錄損失值以便後續觀察模型收斂情況。

結果

程式碼通過可視化方式展示了感知器和 **Adaline** 模型的決策邊界，以及隨著迭代次數增加，模型的錯誤數量或損失的變化情況。

可視化

透過 `plot_decision_regions()` 函數，可以直觀地看到模型在二維平面上的分類結果與決策邊界。

這段程式適合用來理解機器學習分類模型的基本原理，以及如何利用 **Python** 來實作和可視化機器學習模型。