

深度學習

Lab Assingment 2

繳交期限：2025 年 11 月 16 日 23:59 (星期日)

1. 目的

本作業採用 **Fashion-MNIST** 數據集，以單層網路架構的梯度下降法實作四分類問題。
經由本作業，同學可達成以下學習目標：

- 瞭解多類別邏輯迴歸演算法(Multinomial Logistic Regression / Softmax Regression)的原理，並學習其權重(Weights)和偏差值(Biases)的更新方式。
- 觀察線性模型在不同訓練資料組合下的表現差異與限制。
- 經由程式實作，熟悉 NumPy 的矩陣運算與數值操作。

2. 課程章節與實驗環境

- 課程章節: Week 6 - Multinomial Logistic Regression
- 實驗環境: 限定使用 Python 程式語言

3. 實驗與要求

(1) 程式撰寫要求

- 同學需自行撰寫梯度下降法 (可為 Stochastic / Batch / Mini-Batch) 的單層網路多類別邏輯迴歸演算法。
- 不可使用任何現成的套裝軟體或 AI 工具，例如 GitHub Copilot、ChatGPT、scikit-learn、TensorFlow、PyTorch 等。
- 輸出層激活函數 (Activation Function)： **Softmax 函數**。
- 損失函數 (Loss Function)： **k 類交叉熵損失 (Cross-Entropy)**: $-\sum_{j=1}^k y_j \log a_j$
其中 y_j 是 One-Hot Encoding 標籤，其值為 0 或 1， a_j 為神經元輸出值， k 為類別個數。

(2) 實驗資料與任務

本次作業採用含有 10 個類別的 **Fashion-MNIST** 數據集，影像大小為 28×28 灰階圖像(即 784 維輸入屬性)，每筆資料附有 0 – 9 的類別標籤，共有 60,000 筆訓練資料和 10,000 筆測試資料。



本作業僅使用部分 **Fashion-MNIST** 資料集，請同學辨識不同類別組合的數據集。
為了觀察模型在不同類別組合下的表現，分為兩組任務：

組別	類別 (label)	說明	難度	分數比例
Group A (明顯不同的物品)	0: Tshirt/top 1: Trouser 8: Bag 9: Ankle boot	類別差異大， 線性可分性佳	中	80%
Group B (外觀類似的上衣類)	0: Tshirt/top 2: Pullover 4: Coat 6: Shirt	類別相似， 線性可分性差	中高	20%

助教分別為 Group A 及 Group B 資料集預先抽樣：

- 有標籤資料 (train + validation)：每類 2,500 筆，共 10,000 筆
- 無標籤測試資料 (test)：每類約 500 筆，共 2,000 筆

請依照助教指示之輸入格式要求。

(3) One-Hot Encoding

將類別標籤轉換為 One-Hot 向量表示：

例如：四類別問題中，第一、二、三、四類的標籤分別為： $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ 。

(4) 輸出要求

訓練結束時，請顯示或輸出以下資訊：

- 學習率 (learning rate)、訓練世代 (epoch) 數、batch size(使用 Mini-Batch 時)。
- 訓練與驗證集之準確率 (accuracy) 與損失值 (loss)。
- 測試資料的預測結果 (每筆資料之預測類別編號)。
- 繪製訓練與驗證集 loss / accuracy 曲線。

請依照助教指示之輸出格式要求。

(5) 訓練與停止條件

- 建議使用 80% 訓練集、20% 驗證集。
- 訓練可於以下情況停止：
 - 設定的最大世代數 (epoch)。此條件為避免無窮迴圈，不可視為找到合理模型。
 - 訓練損失降至足夠小或不再改善。
 - 驗證準確率下降而訓練準確率上升 (出現 overfitting)。

(6) 實務建議與提示

為順利完成實驗，請注意以下幾項實務重點：

- **資料正規化 (Normalization)**
 - 請將輸入影像的像素值除以 255.0，將範圍縮放至 [0, 1]。
 - 若不進行正規化，梯度可能過大導致訓練不穩定。
- **學習率設定 (Learning Rate)**
 - 初期學習率不宜過大，建議從 0.01 或 0.005 開始嘗試，並觀察 loss 變化調整。

- **Softmax 穩定化技巧 (Numerical Stability)**

- 若 Softmax 出現 overflow 或 NaN，請採用穩定化技巧：
原始的 Softmax 函數定義為：

$$\text{softmax}(n_i) = \frac{e^{n_i}}{\sum_{j=1}^k e^{n_j}}$$

將分子和分母同除以 $e^{\max(n)}$ ，運算結果不變：

$$\text{softmax}(n_i) = \frac{e^{n_i}/e^{\max(n)}}{\sum_{j=1}^k e^{n_j}/e^{\max(n)}} = \frac{e^{n_i - \max(n)}}{\sum_{j=1}^k e^{n_j - \max(n)}}$$

令 $c = \max(n)$ ，則數值穩定的 Softmax 公式為：

$$\text{softmax}(n_i) = \frac{e^{n_i - c}}{\sum_{j=1}^k e^{n_j - c}}$$

- **Mini-Batch 訓練參數建議**

- Batch size 可設定於 32 ~ 128 之間。

- **結果檢查**

- 訓練與驗證準確率應隨世代增加而上升。
- 若訓練集準確率高但驗證集下降，代表模型可能過度擬合。例如右圖應在 C 處的世代數停止以避免過度擬合。

