ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Softmax Regression Project

Báo Cáo Kỹ Thuật
23KHMT1-Nhập môn Học Máy

Nhóm sinh viên thực hiện:

Nhóm 5

Giảng viên hướng dẫn:

Lê Nhựt Nam

Võ Nhật Tân

Thành phố Hồ Chí Minh, Ngày 11 tháng 12 năm 2025

MỤC LỤC

MỤC LỤC i

1 Ý TƯỞNG THỰC HIỆN 2

1.1 Tổng quan đồ án 2

1.2 Mục tiêu 2

1.3 Ý tưởng giải quyết 2

2 PHÂN TÍCH KHÁM PHÁ VÀ TIỀN XỬ LÍ DỮ LIỆU 4

2.1 Phân tích khám phá và tiền xử lí dữ liệu 4

3 CƠ SỞ LÝ THUYẾT VÀ CÀI ĐẶT MÔ HÌNH 7

3.1 Cơ sở lý thuyết 7

3.1.1 Tham số mô hình 7

3.1.2 Hàm Softmax 7

3.1.3 Hàm mất mát Cross-entropy 8

3.1.4 Cập nhật trọng số với Gradient descent 8

3.2 Cài đặt mô hình (SoftmaxRegression) 9

4 TRÍCH XUẤT ĐẶC TRƯNG 11

4.1 Đặc trưng dựa trên cường độ điểm ảnh đã được chuẩn hóa 11

4.1.1 Định nghĩa 11

4.1.2 Quy trình xử lý 11

4.1.3 Tổng quan luồng hoạt động: 11

4.1.4 Các cải thiện được kỳ vọng: 11

4.2 Trích xuất đặc trưng cạnh - Sobel 12
4.2.1 Định nghĩa 12
4.2.2 Quy trình xử lý 12
4.2.3 Tổng quan luồng hoạt động: 13
4.2.4 Các cải thiện được kỳ vọng 14
4.2.5 Hình ảnh minh hoạ với cách trích xuất đặc trưng cạnh Sobel 14
4.3 Giảm chiều dữ liệu - Phân tích thành phần quan trọng (PCA) 14
4.3.1 Định nghĩa 14
4.3.2 Quy trình xử lý 15
4.3.3 Tổng quan luồng hoạt động 16
4.3.4 Các cải thiện được kỳ vọng 16
4.3.5 Hình ảnh minh hoạ với cách trích xuất đặc trưng giảm chiều PCA 18

5 ĐÁNH GIÁ VÀ PHÂN TÍCH KẾT QUẢ 19

5.1 Cài đặt thực nghiệm 19
5.2 Metric đánh giá 19
5.2.1 Accuracy 19
5.2.2 Precision (Macro) 20
5.2.3 Recall (Macro) 20
5.2.4 F1-score 20
5.2.5 Confusion Matrix 21
5.3 Kết quả và Heatmap 22
5.4 Đánh giá và thảo luận kết quả 24
5.5 Thảo luận về điểm mạnh và điểm yếu của Softmax Regression 25
5.5.1 Điểm mạnh 25
5.5.2 Điểm yếu 26

6 ỨNG DỤNG 27

7 KẾT LUẬN **33**

7.1 Tóm tắt nội dung và kết quả chính 33

7.1.1 Kết quả thực nghiệm cho thấy 33

7.2 Hạn chế và hướng cải thiện 34

TÀI LIỆU THAM KHẢO **35**

THÔNG TIN NHÓM

Danh sách thành viên và phân công

STT	MSSV	Họ tên	Vai trò	Phân công	Phần trăm
01	23127181	Nguyễn Trần Quốc Duy	Thành viên	Trích xuất giảm chiều PCA	100%
02	23127207	Đặng Đăng Khoa	Thành viên	Develop application	100%
03	23127271	Võ Ngọc Bích Trâm	Nhóm trưởng	Mô hình softmax regression	100%
04	23127486	Phan Quốc Thịnh	Thành viên	Trích xuất cạnh Sobel	100%
04	23127493	Võ Hoàng Thương	Thành viên	Trích xuất flatten	100%

CHƯƠNG 1

Ý TƯỞNG THỰC HIỆN

1.1. Tổng quan đồ án

Đồ án xây dựng mô hình dùng để dự đoán các giá trị chữ số từ 0-9 từ hình ảnh sử dụng các kỹ thuật trích xuất đặc trưng ảnh khác nhau và tích hợp lên Ứng dụng thực tế.

1.2. Mục tiêu

Giúp cho sinh viên hiểu được cách xây dựng một thuật toán Softmax Regression đơn thuần bằng thư viện Numpy, nhằm có cái nhìn rõ hơn về cơ sở lý thuyết đã học, phương thức cài đặt, cũng như một số cách rút trích đặc trưng ảnh cơ bản và biết cách ứng dụng, tích hợp vào Ứng dụng đơn giản.

1.3. Ý tưởng giải quyết

Quá trình thực hiện đồ án được triển khai theo các bước chính sau:

1. **Thu thập và tiền xử lý dữ liệu:** Dọc tập dữ liệu MNIST có sẵn sau đó kiểm tra và xử lý các giá trị khuyết hoặc bất thường.
2. **Khám phá dữ liệu (EDA):** Sử dụng thống kê, biểu đồ để tổng quan dữ liệu.
3. **Xây dựng mô hình:**
   - Xây dựng mô hình Softmax Regression với các phương pháp trích xuất vector đặc trưng khác nhau.
   - Sử dụng phương pháp Gradient Descent để cập nhật và tìm ra trọng số và bias tối ưu nhất.
4. **Đánh giá mô hình:** So sánh các mô hình bằng các chỉ số đánh giá khác nhau như: accuracy,

precision, recall, F1-score, confusion matrix.

5. **Kết luận:** Đưa ra nhận xét và lựa chọn mô hình phù hợp nhất để tích hợp vào ứng dụng.

CHƯƠNG 2

PHÂN TÍCH KHÁM PHÁ VÀ TIỀN XỬ LÍ DỮ LIỆU

Tập dữ liệu được sử dụng trong đồ án này là tập dữ liệu **MNIST**, một trong những bộ dữ liệu kinh điển và được sử dụng rộng rãi nhất trong lĩnh vực nhận dạng hình ảnh và học sâu, đặc biệt trong các mô hình phân loại ảnh.

MNIST bao gồm các hình ảnh chữ số viết tay từ 0 đến 9, được thu thập, chuẩn hóa và xử lý lại từ tập dữ liệu **MNIST** gốc. Đây là bộ dữ liệu đơn giản, nhỏ gọn và phù hợp để thử nghiệm, đánh giá các thuật toán phân loại như Logistic Regression, Softmax Regression, SVM, MLP, CNN,. . .

Tổng cộng có **70.000 ảnh** chữ số viết tay. Chia thành hai phần:

- **60.000 ảnh** dùng cho tập huấn luyện (training set)
- **10.000 ảnh** dùng cho tập kiểm tra (test set)

2.1. Phân tích khám phá và tiền xử lí dữ liệu

Tổng quan dữ liệu:

- Số mẫu train: 60000
- Số mẫu test: 10000
- Kích thước mỗi ảnh 28 × 28 pixel tương ứng với 28 dòng 28 cột
- Kiểu dữ liệu ảnh: uint8
- Giá trị pixel min/max (train): 0 255
- Giá trị pixel min/max (test) : 0 255

- Các nhãn train duy nhất: [0 1 2 3 4 5 6 7 8 9]
- Các nhãn test duy nhất: [0 1 2 3 4 5 6 7 8 9]

Kiểm tra giá trị thiếu và inf trên tập train:

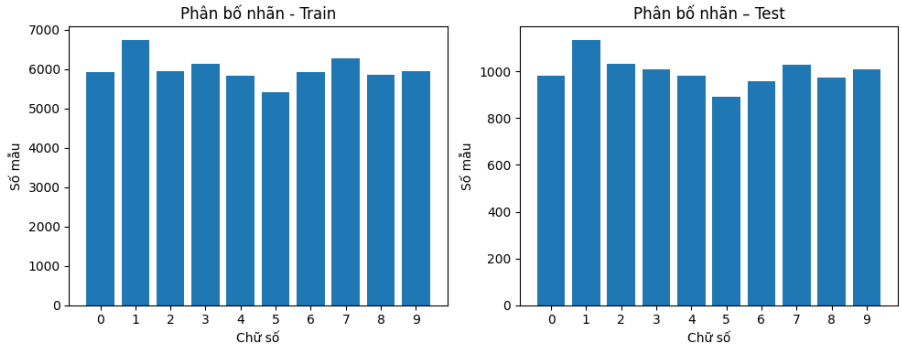
- Số NaN trong ảnh : 0
- Số NaN trong nhãn: 0
- Số Inf trong ảnh : 0
- Số Inf trong nhãn: 0

Kiểm tra giá trị thiếu và inf trên tập test:

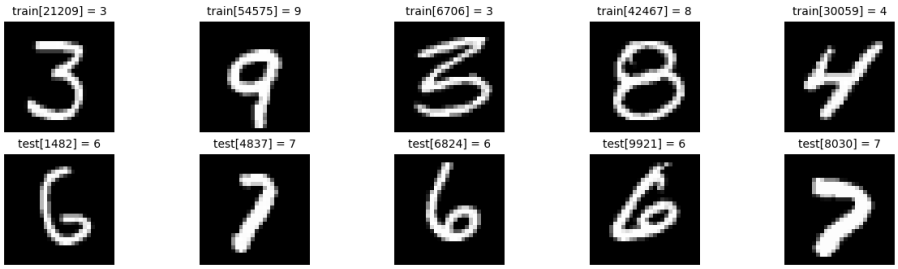
- Số NaN trong ảnh : 0
- Số NaN trong nhãn: 0
- Số Inf trong ảnh : 0
- Số Inf trong nhãn: 0

Nhận xét: Vì không có missing value và outlier nên không cần phải tiền xử lí đối với các trường hợp này.

Phân bố nhãn và hiển thị một số ảnh ngẫu nhiên trên tập train và test:



Hình 2.1: Phân bố nhãn trên tập train và test



Hình 2.2: Một số ảnh ngẫu nhiên trên tập train và test

CHƯƠNG 3

CƠ SỞ LÝ THUYẾT VÀ CÀI ĐẶT MÔ HÌNH

3.1. Cơ sở lý thuyết

3.1.1. Tham số mô hình

- Input (ảnh) được chuyển thành vector đặc trưng:

$$x \in \mathbb{R}^d$$

- Số lớp: K

- Tham số của mô hình:

$$W \in \mathbb{R}^{K \times d} \text{ (trọng số)}, \qquad b \in \mathbb{R}^K$$

- Logits:

$$z = Wx + b.$$

- Xác suất các nhãn dự đoán:

$$\hat{y} = \text{softmax}(Wx + b), \qquad \hat{y} \in \mathbb{R}^K, \qquad \hat{y}_i \in [0, 1], \qquad \sum_{i=1}^{K} \hat{y}_i = 1.$$

3.1.2. Hàm Softmax

- Hàm softmax cho lớp k được định nghĩa như sau:

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

- Trong đó: $z = Wx + b$

3.1.3. Hàm mất mát Cross-entropy

- Quá trình mô hình học dựa trên hàm loss cross-entropy.
- Cross-entropy được định nghĩa như sau:
  - Với: x là input, y là nhãn thật (dạng one-hot), y_k là giá trị tại vị trí k trong y.

$$\mathcal{L}(x, y) = -\sum_{k=1}^{K} y_k \log(\hat{y}_k).$$

  - Với tập huấn luyện N mẫu:

$$\mathcal{J}(W, b) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, y^{(i)})$$

3.1.4. Cập nhật trọng số với Gradient descent

- Với 1 sample:

$$\frac{\partial \mathcal{L}}{\partial W} = (\hat{y} - y)x^T,$$
$$\frac{\partial \mathcal{L}}{\partial b} = (\hat{y} - y).$$

- Với mini-batch/toàn bộ dữ liệu:

$$\frac{\partial \mathcal{J}}{\partial W} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathcal{L}_i}{\partial W},$$
$$\frac{\partial \mathcal{J}}{\partial b} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \mathcal{L}_i}{\partial b}.$$

- Update:
  - Với 1 mẫu:

$$W \leftarrow W - \alpha \frac{\partial \mathcal{L}}{\partial W}, \qquad b \leftarrow b - \alpha \frac{\partial \mathcal{L}}{\partial b}.$$

  - Với N mẫu:

$$W \leftarrow W - \alpha \frac{\partial \mathcal{J}}{\partial W}, \qquad b \leftarrow b - \alpha \frac{\partial \mathcal{J}}{\partial b}.$$

- Với α: là learning rate.

3.2. Cài đặt mô hình (SoftmaxRegression)

Danh sách hyperparameters

`learning_rate`: (α) Tốc độ cập nhật trọng số.

`n_iterations`: Số vòng lặp huấn luyện (Epochs).

`n_classes`: Số lượng lớp đầu ra (với MNIST là 10 lớp).

`weights`: Ma trận trọng số W, sẽ được khởi tạo khi bắt đầu train.

Hàm kích hoạt Softmax

Chuyển đổi điểm số (score) thành xác suất.

$$\hat{y}_k = \frac{e^{z_k}}{\sum_j e^{z_j}}.$$

Lưu ý: Khi cài đặt,

$$z_{\text{shifted}} = z - \max(z).$$

Việc trừ đi giá trị lớn nhất giúp tránh hiện tượng tràn số (overflow) khi tính hàm mũ e^z với các giá trị z lớn.

One_hot_encode

Chuyển đổi nhãn dạng số nguyên (ví dụ: $y = 2$) sang vector xác suất mục tiêu (ví dụ: $[0, 0, 1, 0, \dots]$). Điều này cần thiết để tính toán hàm mất mát Cross-Entropy.

Huấn luyện mô hình (`fit`)

Sử dụng thuật toán *Gradient Descent*:

1. **Khởi tạo trọng số:** Tạo ma trận W kích thước `(n_features, n_classes)` với toàn giá trị 0.

2. **Vòng lặp:**

- **Lan truyền xuôi (Forward Pass):** Tính toán điểm số

$$Z = XW + b$$

và xác suất dự đoán

$$\hat{Y} = \text{softmax}(Z).$$

- **Tính hàm mất mát (Compute Loss):** Sử dụng hàm **Cross-Entropy Loss**:

$$J(W) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log(\hat{y}_{ik}).$$

$$J(b) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log(\hat{y}_{ik}).$$

Lưu ý: Cộng thêm `1e-8` vào logarit để tránh lỗi $\log(0)$.

- **Tính Gradient (Backward Pass):** Tính đạo hàm của hàm mất mát theo trọng số W và bias b:

$$\nabla_b J = \frac{1}{N}(\hat{Y} - Y).$$

$$\nabla_W J = \frac{1}{N}X^T(\hat{Y} - Y).$$

- **Cập nhật trọng số (Update Parameters):**

$$W \leftarrow W - \alpha \cdot \nabla_W J.$$

$$b \leftarrow b - \alpha \cdot \nabla_b J.$$

Dự đoán (`predict`)

Sau khi mô hình đã học được trọng số W tối ưu:

1. Tính xác suất cho dữ liệu mới:

$$\hat{Y} = \text{softmax}(XW + b).$$

2. Chọn lớp có xác suất cao nhất: `np.argmax`.

CHƯƠNG 4

TRÍCH XUẤT ĐẶC TRƯNG

4.1. Đặc trưng dựa trên cường độ điểm ảnh đã được chuẩn hóa

4.1.1. Định nghĩa

Phương pháp trích xuất đặc trưng này chỉ đơn giản là chuẩn hóa mọi pixel từ giá trị [0,255] về [0,1] và flatten để đưa vào mô hình

4.1.2. Quy trình xử lý

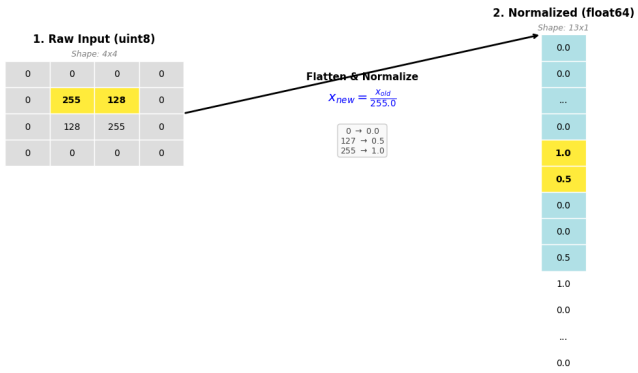
- **Biến đổi:**
  - **Input:** ảnh gốc 28×28 (giá trị 0–255).
  - **Bước 1:** chuyển sang float, chia 255 $\rightarrow$ mỗi pixel $\in [0, 1]$.
  - **Bước 2:** flatten $28 \times 28 \rightarrow$ vector 784 chiều.
- **Vector đặc trưng thu được:** $\mathbf{x} \in \mathbb{R}^{784}$, mỗi phần tử là cường độ xám chuẩn hóa.

4.1.3. Tổng quan luồng hoạt động:

Quy trình trích xuất đặc trưng ảnh với phương pháp chuẩn hóa cường độ điểm ảnh được thể hiện ở hình 4.1

4.1.4. Các cải thiện được kỳ vọng:

- Các feature cùng thang đo $[0, 1] \rightarrow$ gradient descent ổn định hơn.
- Tránh giá trị lớn 0 - 255 gây tràn số trong softmax, giúp mô hình hội tụ tốt hơn.



Hình 4.1: Quy trình trích xuất đặc trưng ảnh với phương pháp Chuẩn hóa cường độ điểm ảnh

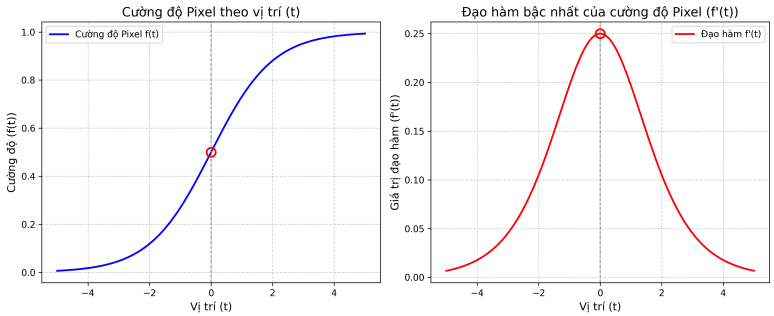
4.2. Trích xuất đặc trưng cạnh - Sobel

4.2.1. Định nghĩa

Phương pháp Sobel là một kỹ thuật phát hiện cạnh, giúp làm nổi bật đường biên của chữ số và bỏ qua các thông tin nhiễu hoặc vùng nền không quan trọng.

4.2.2. Quy trình xử lý

- **Chuyển đổi kiểu dữ liệu:** Các điểm ảnh đầu vào được chuyển sang dạng số thực (`float64`) để đảm bảo độ chính xác khi tính toán đạo hàm.
- **Tính Gradient:**
  - **Nền tảng lý thuyết:** Phương thức Sobel hoạt động dựa trên nguyên lý tính `đạo hàm của cường độ điểm ảnh`. Đạo hàm bậc nhất sẽ đạt giá trị cực đại tại nơi có sự thay đổi độ sáng mạnh.[3]



Hình 4.2: Sự thay đổi cường độ điểm ảnh tại cạnh và Đạo hàm

– Sobel sử dụng phép tích chập (Convolution) giữa ảnh gốc I với hai hạt nhân (kernels) 3×3 để tính toán sự thay đổi này theo hai hướng: dọc và ngang.

– Công thức toán học của các Kernel:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \quad \text{và} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

– **Sobel X (G_x):** Phát hiện các cạnh dọc (sự thay đổi độ sáng theo chiều ngang).

– **Sobel Y (G_y):** Phát hiện các cạnh ngang (sự thay đổi độ sáng theo chiều dọc).

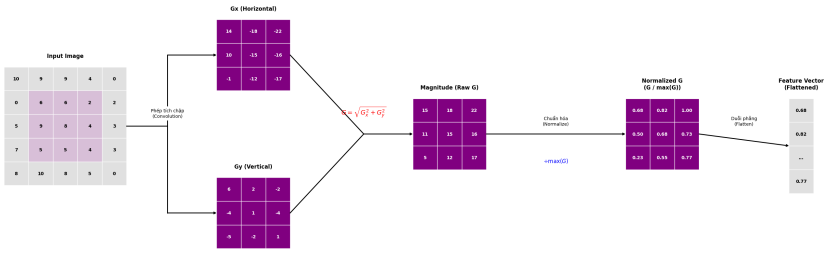
- **Tính Biên độ (Magnitude):** Kết hợp hai thành phần gradient để thu được độ mạnh tổng quát của cạnh tại mỗi điểm ảnh, sử dụng công thức Pytago:

$$G = \sqrt{G_x^2 + G_y^2}.$$

- **Chuẩn hóa (Normalization):** Chia toàn bộ ma trận biên độ cho giá trị lớn nhất để đưa các giá trị về khoảng $[0, 1]$. Điều này giúp thuật toán Gradient Descent hội tụ ổn định hơn.

- **Làm phẳng (Flattening):** Duỗi ma trận kết quả 28×28 thành vector đặc trưng 1 chiều có kích thước 784.

4.2.3. Tổng quan luồng hoạt động:



Hình 4.3: Quy trình trích xuất đặc trưng ảnh với phương pháp Sobel

Quy trình trích xuất đặc trưng ảnh với phương pháp Sobel được thể hiện ở hình 4.3

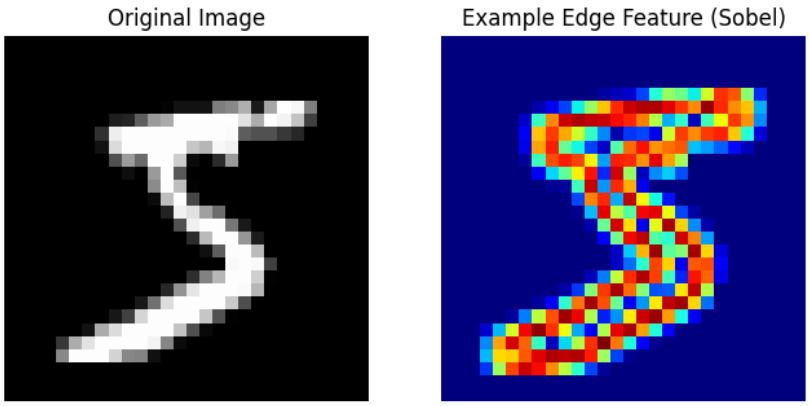
4.2.4. Các cải thiện được kỳ vọng

Loại bỏ nhiễu do cường độ sáng/ màu sắc của ảnh: Bằng cách tính toán đạo hàm bậc nhất theo hai phương ngang (G_x) và dọc (G_y) rồi tổng hợp thành biên độ $G = \sqrt{G_x^2 + G_y^2}$, thuật toán loại bỏ các mảng màu/độ sáng đồng nhất (nền) và chỉ làm nổi bật các cạnh bao quanh.

Ổn định với màu sắc/cường độ sáng: Dù ảnh là sáng hay tối, cường độ tại biên không có quá nhiều nhạy cảm, giúp mô hình hoạt động ổn định.

Chuẩn hóa biên độ trở nên tối ưu với thuật toán Gradient Descent: phù hợp với quá trình cải thiện hàm mất mát, cải thiện tốc độ hội tụ với các mô hình sử sử dụng thuật toán Gradient Descent.

4.2.5. Hình ảnh minh hoạ với cách trích xuất đặc trưng cạnh Sobel



Hình 4.4: Ảnh gốc với ảnh sử dụng trích xuất cạnh Sobel

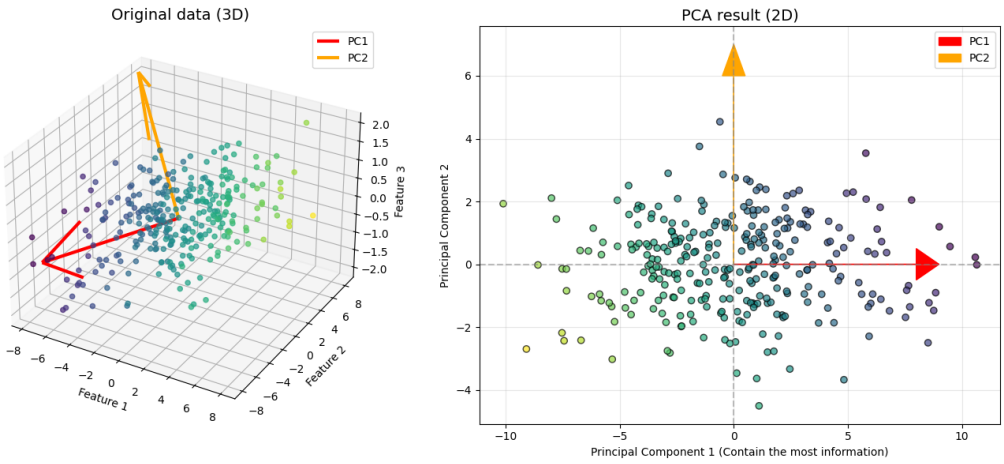
4.3. Giảm chiều dữ liệu - Phân tích thành phần quan trọng (PCA)

4.3.1. Định nghĩa

- PCA (Principal Component Analysis – Phân tích thành phần chính) là một kỹ thuật giảm chiều dữ liệu và trích xuất đặc trưng bằng cách tìm ra các hướng (hay thành phần chính) trong không gian dữ liệu mà trên đó, các quan sát thể hiện sự biến động, phương sai lớn nhất.

[2]

- Với tập dữ liệu MNIST: mỗi ảnh $28 \times 28 \rightarrow 784$ chiều.
- Sử dụng PCA giảm còn 50-100 chiều.
- Sau đó train Softmax Regression nhanh hơn và đôi khi giúp mô hình ổn định hơn.



Hình 4.5: Ví dụ minh họa phân tích thành phần quan trọng (PCA)

4.3.2. Quy trình xử lý

- Với ma trận dữ liệu:

$$X \in \mathbb{R}^{N \times d}.$$

- Quá trình hoạt động của PCA bao gồm các bước cơ bản sau: [2]
  - **Bước 1: Chuẩn hóa dữ liệu:** Dữ liệu được chuẩn hóa về cùng một thang đo bằng cách trừ đi giá trị trung bình.

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i, \qquad X_c = X - \mu.$$

  - **Bước 2: Tính ma trận hiệp phương sai:** Bước này nhằm tính toán ma trận hiệp phương sai để xác định mối quan hệ và mức độ tương quan giữa các biến, nhận diện thông tin dư thừa.

$$C = \frac{1}{N} X_c^T X_c.$$

– **Bước 3: Phân rã trị riêng và vector riêng:** Từ ma trận hiệp phương sai, PCA tìm ra các `eigenvectors (vector riêng)` và `eigenvalues (trị riêng)`.Các vector riêng xác định các hướng có độ biến thiên dữ liệu (phương sai) lớn nhất, còn các trị riêng cho biết lượng thông tin mà mỗi hướng mang lại.

$$Cu_k = \lambda_k u_k,$$

trong đó:

$$u_k \text{ là vector riêng,} \qquad \lambda_k \text{ là trị riêng.}$$

– **Bước 4: Chọn thành phần chính:** tức lấy m vector riêng ứng với m trị riêng lớn nhất:

$$U_m = [u_1, \ldots, u_m].$$

– **Bước 5: Chiếu dữ liệu lên không gian mới:** Dữ liệu gốc được chiếu lên các thành phần chính đã chọn. Kết quả thu được là một tập dữ liệu có số chiều thấp hơn nhưng vẫn giữ được hầu hết các đặc trưng quan trọng, giúp trực quan hóa dễ dàng hơn và tăng hiệu quả cho các thuật toán học máy.

$$Z = X_c U_m \in \mathbb{R}^{N \times m}.$$

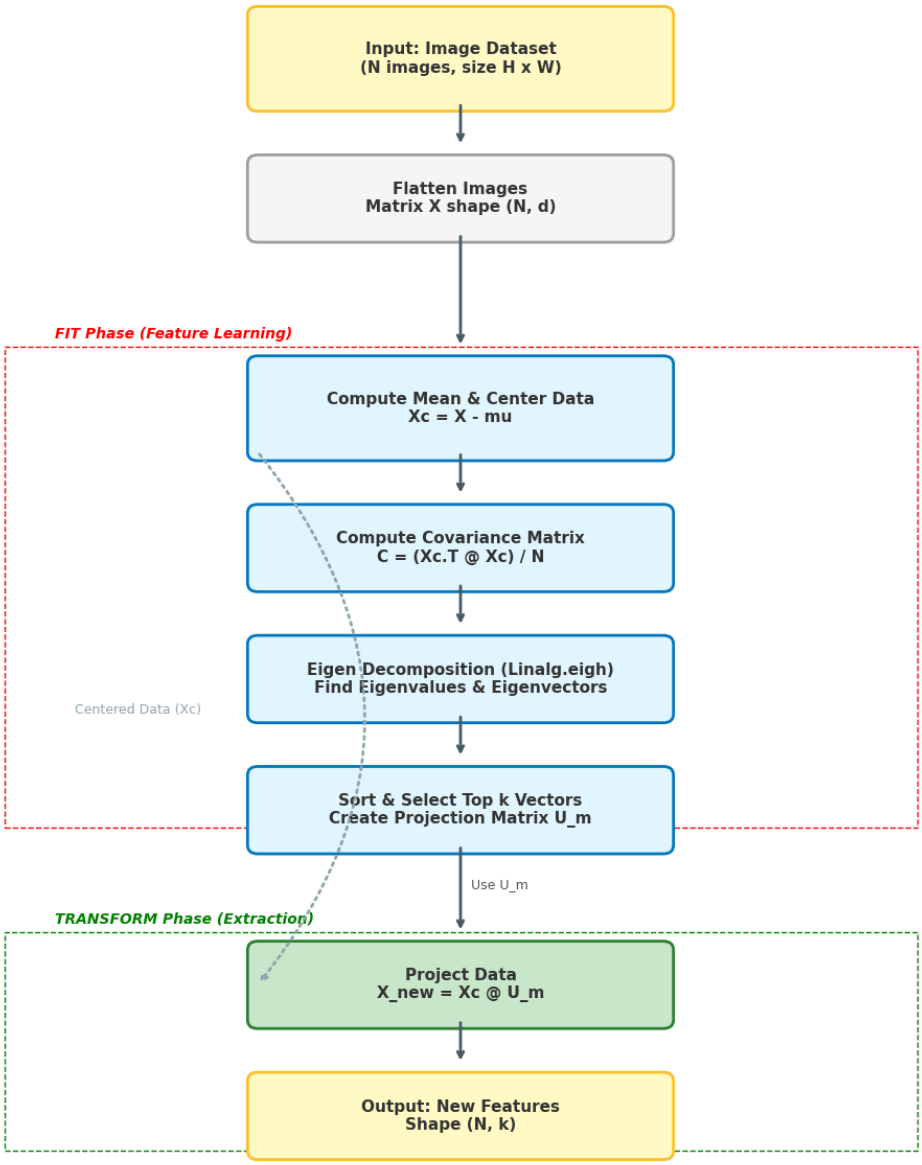
4.3.3. Tổng quan luồng hoạt động

Luồng hoạt động của trích xuất đặc trưng bằng cách giảm chiều - PCA được thể hiện qua hình 4.6

4.3.4. Các cải thiện được kỳ vọng

- **Loại bỏ thành phần gây nhiễu:** Trong dữ liệu ảnh, đặc biệt là chữ viết tay, các thành phần có phương sai thấp thường đại diện cho các chi tiết nhiều, thừa ở nền hoặc không quan trọng. PCA đóng vai trò như bộ lọc giữ lại các chi tiết quan trọng cho quá trình dự đoán.

- **Giải quyết một phần 'lời nguyền chiều cao' (curse of dimensionality):** [1]: PCA là kỹ thuật giảm chiều dữ liệu bằng cách chiếu các điểm quan sát lên một không gian con có số chiều nhỏ hơn nhưng vẫn giữ lại phần lớn phương sai của dữ liệu. Nhờ đó, mô hình học máy phải làm việc trong một không gian có ít đặc trưng hơn, giúp:



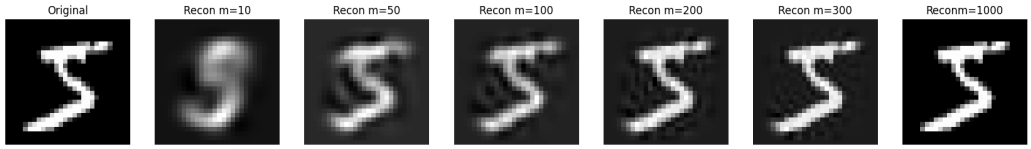
Hình 4.6: Quy trình trích xuất đặc trưng ảnh với phương pháp PCA

– Giảm hiện tượng dữ liệu bị thưa (data sparsity) ở không gian chiều cao, từ đó giảm nhu cầu về số lượng mẫu huấn luyện.

– Hạn chế overfitting do số chiều/đặc trưng quá lớn so với số lượng mẫu.

– Cải thiện hiệu suất tính toán (số chiều giảm ⇒ chi phí xử lý ma trận, tính khoảng cách, huấn luyện mô hình cũng giảm).

4.3.5. Hình ảnh minh hoạ với cách trích xuất đặc trưng giảm chiều PCA



Hình 4.7: Sự thay đổi của ảnh ảnh so với ảnh sử dụng trích xuất đặc trưng giảm chiều bằng PCA (10, 50, 100, 200, 300)

Một vài hình ảnh minh hoạ với cách trích xuất đặc trưng giảm chiều PCA khi giam xuống 10, 50, 100, 200, 300 chiều được thể hiện ở hình 4.7

CHƯƠNG 5

ĐÁNH GIÁ VÀ PHÂN TÍCH KẾT QUẢ

5.1. Cài đặt thực nghiệm

Trong phần này, chúng tôi trình bày cấu hình dữ liệu, mô hình và các siêu tham số sử dụng trong quá trình huấn luyện và đánh giá.

- **Dữ liệu**: Bộ dữ liệu MNIST gồm tổng cộng 70.000 ảnh chữ số viết tay (đen–trắng), trong đó:
  - 60.000 ảnh được dùng làm tập huấn luyện.
  - 10.000 ảnh được dùng làm tập kiểm tra.

  Bài toán là phân loại 10 lớp tương ứng với các chữ số từ 0 đến 9.
- **Mô hình**: Sử dụng mô hình hồi quy Softmax (Softmax Regression) cho bài toán phân loại nhiều lớp.
- **Siêu tham số (hyperparameters)**:
  - Learning rate: 0.12
  - Số vòng lặp huấn luyện (epochs): 150

5.2. Metric đánh giá

5.2.1. Accuracy

Accuracy (độ chính xác tổng thể) đo tỉ lệ mẫu mà mô hình dự đoán đúng trên toàn bộ tập dữ liệu:

$$\text{Accuracy} = \frac{\text{Số mẫu dự đoán đúng}}{\text{Tổng số mẫu}}$$

5.2.2. Precision (Macro)

Với bài toán phân loại nhiều lớp, ta xác định **precision** cho từng lớp c như sau:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}$$

Trong đó:

- TP_c: số mẫu *thuộc* lớp c và được mô hình *dự đoán đúng* là c.
- FP_c: số mẫu *không thuộc* lớp c nhưng bị mô hình *dự đoán nhầm* là c.

Precision (macro) được tính bằng cách lấy trung bình cộng precision của tất cả các lớp:

$$\text{Precision}_{\text{macro}} = \frac{1}{K} \sum_{c=1}^{K} \text{Precision}_c$$

với K là số lượng lớp.

5.2.3. Recall (Macro)

Tương tự, **recall** cho từng lớp c được định nghĩa là:

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c}$$

Trong đó:

- FN_c: số mẫu *thuộc* lớp c nhưng *không được* mô hình dự đoán là c (bị dự đoán sang lớp khác).

Recall (macro) là trung bình cộng recall của tất cả các lớp:

$$\text{Recall}_{\text{macro}} = \frac{1}{K} \sum_{c=1}^{K} \text{Recall}_c$$

với K là số lượng lớp.

5.2.4. F1-score

- F1 là trung bình điều hòa giữa Precision và Recall giúp cân bằng giữa 2 giá trị Precision và Recall được định nghĩa là:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Nếu Precision cao mà Recall thấp → F1 sẽ không cao.
- Nếu Recall cao mà Precision thấp → F1 cũng không cao.
- Chỉ khi cả hai đều tương đối tốt, F1 mới cao.

5.2.5. Confusion Matrix

Confusion matrix (ma trận nhầm lẫn) là một ma trận kích thước $K \times K$ dùng để mô tả chi tiết cách mô hình phân loại các lớp, với K là số lượng lớp.

Mỗi phần tử M_{ij} trong ma trận biểu diễn:

$$M_{ij} = \text{số mẫu } \textit{thực tế} \text{ thuộc lớp } i \text{ nhưng được mô hình } \textit{dự đoán} \text{ là lớp } j$$

Thông thường:

- Mỗi **hàng** tương ứng với **nhãn thật** (ground truth).
- Mỗi **cột** tương ứng với **nhãn dự đoán** (predicted label).

Các phần tử trên **đường chéo chính** M_{cc} (với $c = 1, \ldots, K$) là số mẫu được phân loại **đúng** cho từng lớp. Các phần tử **ngoài đường chéo** M_{ij} với $i \neq j$ thể hiện số mẫu bị **nhầm** từ lớp i sang lớp j.

Từ ma trận nhầm lẫn, ta có thể suy ra các đại lượng như:

- $TP_c = M_{cc}$: số lượng dự đoán đúng cho lớp c.
- $FP_c = \sum_{i \neq c} M_{ic}$: số mẫu bị dự đoán nhầm *thành* lớp c.
- $FN_c = \sum_{j \neq c} M_{cj}$: số mẫu *thuộc* lớp c nhưng bị dự đoán nhầm sang lớp khác.

Nhờ đó, confusion matrix cung cấp cái nhìn trực quan về:

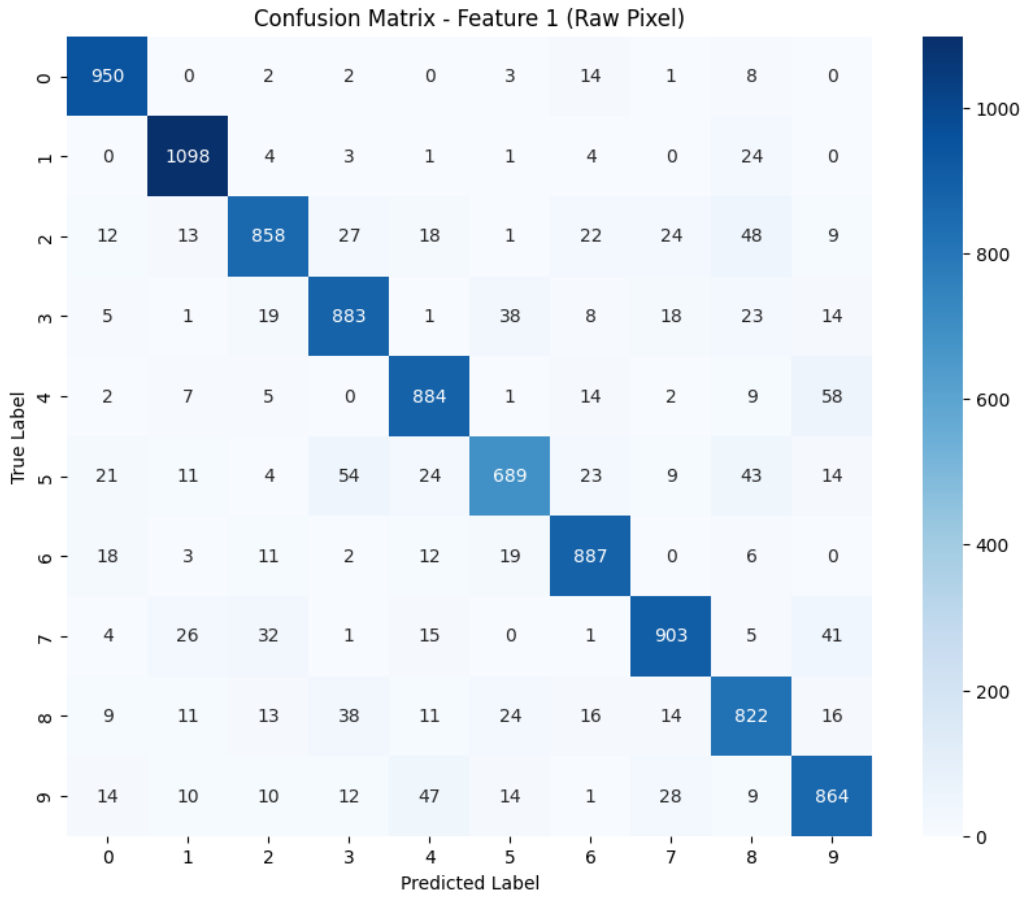
- Lớp nào mô hình phân loại tốt (nhiều giá trị trên đường chéo).
- Lớp nào thường bị nhầm với lớp nào khác (các ô ngoài đường chéo có giá trị lớn).

Feature	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
Feature 1 (Raw Pixel)	0.8838	0.8828	0.8820	0.8817
Feature 2 (Edges)	0.8467	0.8470	0.8442	0.8442
Feature 3 (PCA)	0.8780	0.8776	0.8761	0.8759

Bảng 5.1: Kết quả đánh giá các phương pháp trích chọn đặc trưng trên mô hình Softmax Regression

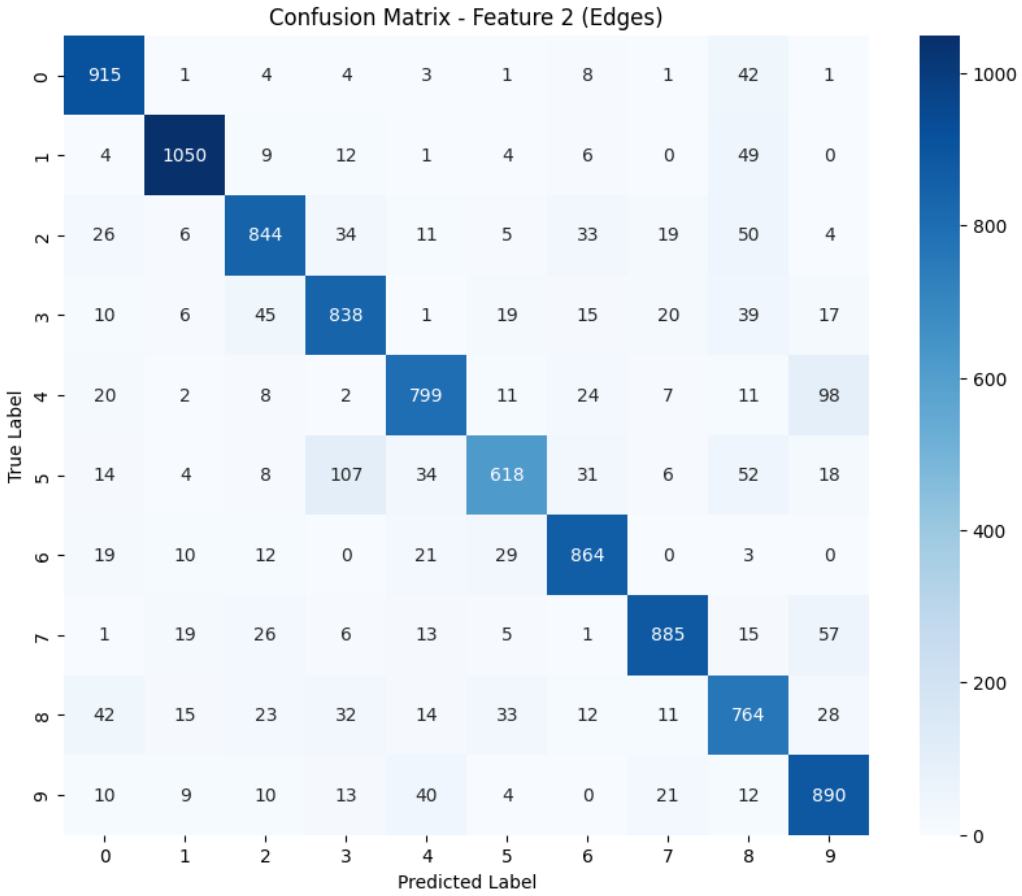
5.3. Kết quả và Heatmap

- **Heatmap kết quả với đặc trưng raw pixel** (ảnh sau khi được chuẩn hoá và làm phẳng):



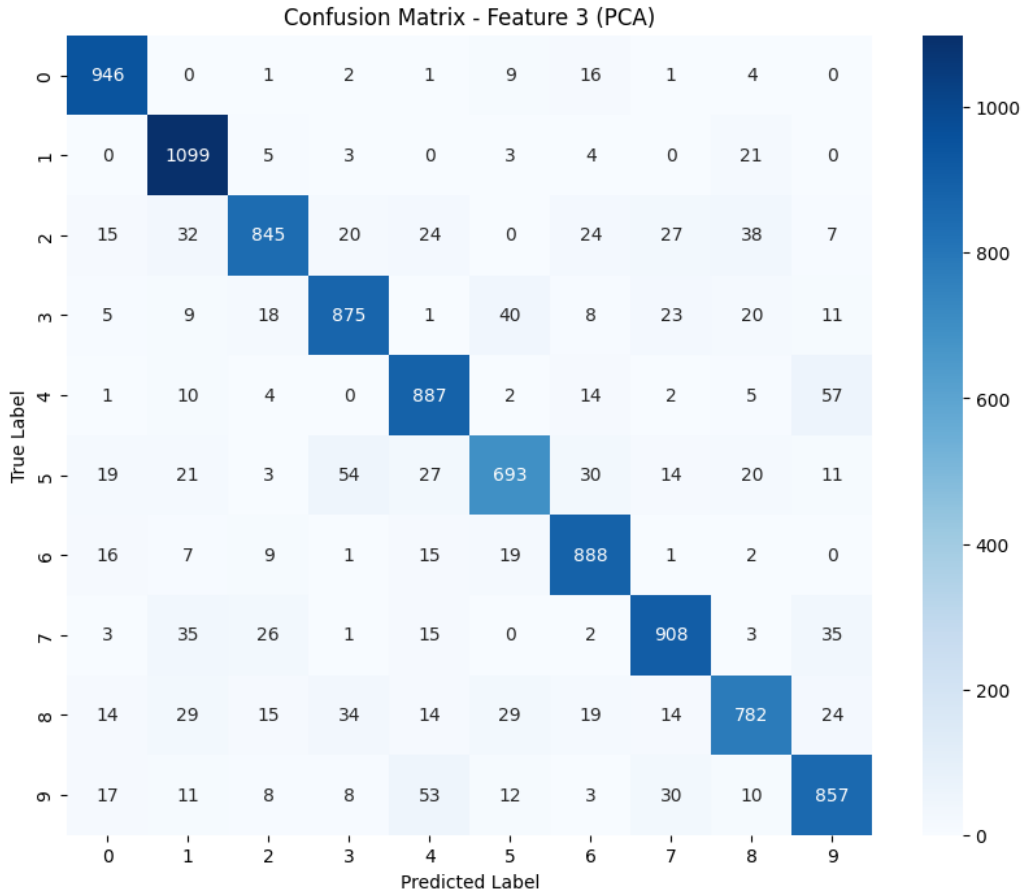
Hình 5.1: Heatmap ma trận nhầm lẫn của mô hình sử dụng đặc trưng raw pixel (chuẩn hoá và flatten).

- **Heatmap kết quả với đặc trưng cạnh (Edges) trích xuất bằng Sobel:**



Hình 5.2: Heatmap ma trận nhầm lẫn của mô hình sử dụng đặc trưng cạnh (Edges) trích xuất bằng Sobel.

- **Heatmap kết quả với đặc trưng sau khi giảm chiều bằng PCA:**



Hình 5.3: Heatmap ma trận nhầm lẫn của mô hình sử dụng đặc trưng sau khi giảm chiều bằng PCA.

5.4. Đánh giá và thảo luận kết quả

Dựa trên các chỉ số đánh giá trong Bảng 5.1 và các heatmap ma trận nhầm lẫn 5.1 5.1 5.3, có thể rút ra một số nhận xét như sau:

- **So sánh giữa các loại đặc trưng**:
  - Đặc trưng **raw pixel** cho kết quả tốt nhất về cả Accuracy, Precision (macro) và Recall (macro). Điều này cho thấy mô hình Softmax Regression vẫn khai thác khá hiệu quả thông tin trực tiếp từ mức pixel khi dữ liệu đã được chuẩn hoá.
  - Đặc trưng **Edges (Sobel)** cho kết quả thấp hơn đáng kể. Việc chỉ giữ lại thông tin cạnh giúp làm nổi bật biên dạng chữ số nhưng đồng thời làm mất đi nhiều thông tin về cấu trúc

bên trong, dẫn đến mô hình tuyến tính như Softmax Regression khó phân tách một số lớp gần nhau.

  - Đặc trưng **PCA** cho kết quả chỉ thấp hơn một chút so với raw pixel, trong khi số chiều đã được giảm đi đáng kể. Từ góc độ cá nhân, nhóm đánh giá đây là một sự đánh đổi hợp lý giữa hiệu năng và chi phí tính toán/lưu trữ: mô hình vẫn giữ được phần lớn chất lượng phân loại nhưng làm việc trên không gian đặc trưng gọn hơn.

- **Mối quan hệ giữa Accuracy, Precision (macro) và Recall (macro)**: Các giá trị Accuracy, Precision (macro) và Recall (macro) khá gần nhau trong cả ba thiết lập. Điều này gợi ý rằng mô hình không thiên lệch mạnh về một vài lớp cụ thể và hiệu năng tương đối đồng đều giữa các lớp (do macro-average cho mỗi lớp trọng số như nhau). Với bộ dữ liệu cân bằng như MNIST, đây là kết quả phù hợp với kỳ vọng.

- **Quan sát từ heatmap**: Quan sát các heatmap ma trận nhầm lẫn, lỗi phân loại chủ yếu tập trung ở những cặp chữ số có hình dạng gần giống nhau (chẳng hạn như *4-9*, *3-5*, *7 - 9*). Khi dùng đặc trưng Edges, hiện tượng nhầm lẫn này thường rõ hơn vì mô hình chỉ nhìn thấy biên cạnh mà không có thông tin "độ đậm/nhạt" bên trong nét chữ. Ngược lại, với raw pixel và PCA, mô hình có thêm ngữ cảnh về hình dáng tổng thể, nên giảm bớt một phần nhầm lẫn.

5.5. Thảo luận về điểm mạnh và điểm yếu của Softmax Regression

5.5.1. Điểm mạnh

- Hiệu suất với cách trích xuất raw pixel theo bảng 5.1 là cao nhất chứng tỏ mô hình tận dụng tốt các quan hệ tuyến tính đơn giản.

- PCA giảm chiều (cụ thể giảm chiều còn 100) nhưng vẫn duy trì hiệu năng cao (giảm ít so với raw pixel theo bảng 5.1) cho thấy softmax ổn định trên không gian đặc trưng tuyến tính. Từ đó, cho thấy softmax chỉ cần một không gian đặc trưng có phương sai lớn nhất, không cần toàn bộ 784 chiều.

- Mô hình không phụ thuộc vào các chiều nhiễu nhỏ, vì kết quả với PCA (5.1) vẫn ổn định (giảm khá ít so với trích xuất đặc trưng bằng raw pixel).

- Đối với các cách trích xuất đặc trưng khác nhau thì độ chính xác thay đổi cho thấy mô hình cho phản hồi rõ ràng về chất lượng đặc trưng vì thế mô hình softmax regression sẽ khá phù hợp với các bài toán phân tích feature quality.

5.5.2. Điểm yếu

- Nhầm lẫn nhiều với các lớp có hình dạng tương tự ở hình 5.1, 5.2, 5.3 (chẳng hạn như *4-9*, *3-5*, *7 - 9*) số lượng nhầm lẫn khá nhiều. Từ đó cho thấy, Softmax Regression phụ thuộc mạnh vào sự tách tuyến tính giữa các lớp. Nếu đặc trưng không tổ chức theo cách tuyến tính rõ ràng, mô hình suy giảm nhanh chóng.

- Edge features gây giảm hiệu năng 5.1 đáng kể cho thấy softmax không phù hợp khi đặc trưng không mô tả hình dạng tổng thể.

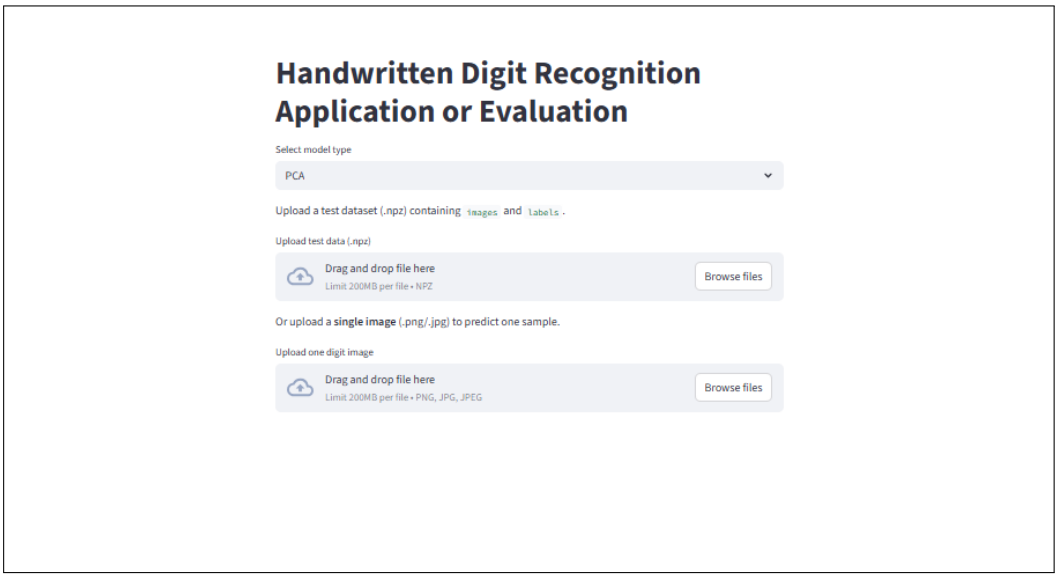
- PCA làm mất một phần thông tin hữu ích và độ hiệu quả mô hình cũng giảm cho thấy Softmax không tận dụng được các quan hệ phi tuyến, nên khi PCA loại bỏ thông tin không nằm trên các vector phương sai lớn nhất, mô hình bị ảnh hưởng và làm giảm độ hiệu quả,...

- Khi thay đổi cách trích xuất đặc trưng như: Raw pixel, Edges - Sobel hay giảm chiều - PCA thì độ hiệu quả mô hình cũng thay đổi cho thấy Softmax không tự sinh đặc trưng, mà hoàn toàn phụ thuộc vào đặc trưng đầu vào. Nếu đặc trưng không đủ khả năng biểu diễn sự khác biệt giữa các lớp, mô hình không thể tự học thêm cấu trúc phức tạp. Softmax Regression hoàn toàn phụ thuộc vào feature engineering mà không thể tự trích xuất đặc trưng phức tạp như các mô hình sâu.

CHƯƠNG 6

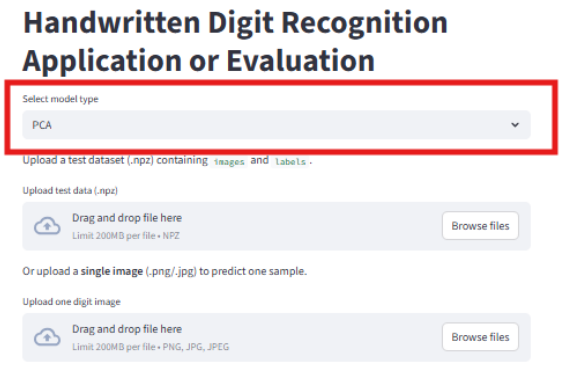
ỨNG DỤNG

- Giao diện chính của ứng dụng:



Hình 6.1: Giao diện chính

- Nơi chọn cách trích xuất đặc trưng. Khi bấm vào sẽ hiện ra 3 cách trích xuất chủ yếu:

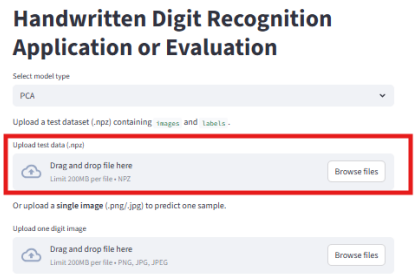


Hình 6.2: Trích xuất đặc trưng

- 3 cách trích xuất đặc trưng ảnh (Raw Pixels, Edges (Sobel), PCA):

Hình 6.3: Trích xuất đặc trưng

- Cách tải file .npz

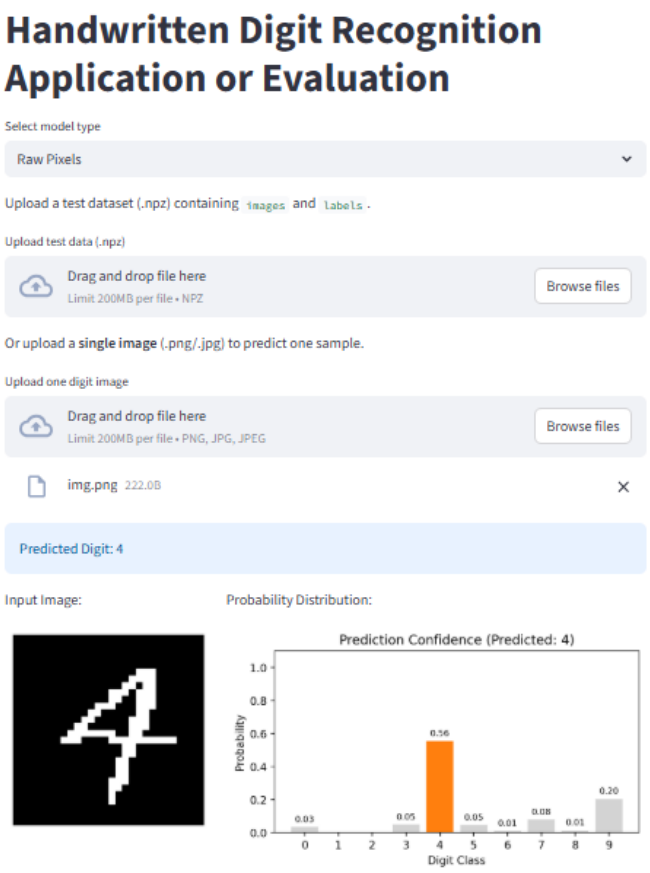


Hình 6.4: Nơi tải file .npz

- Cách upload ảnh lên:

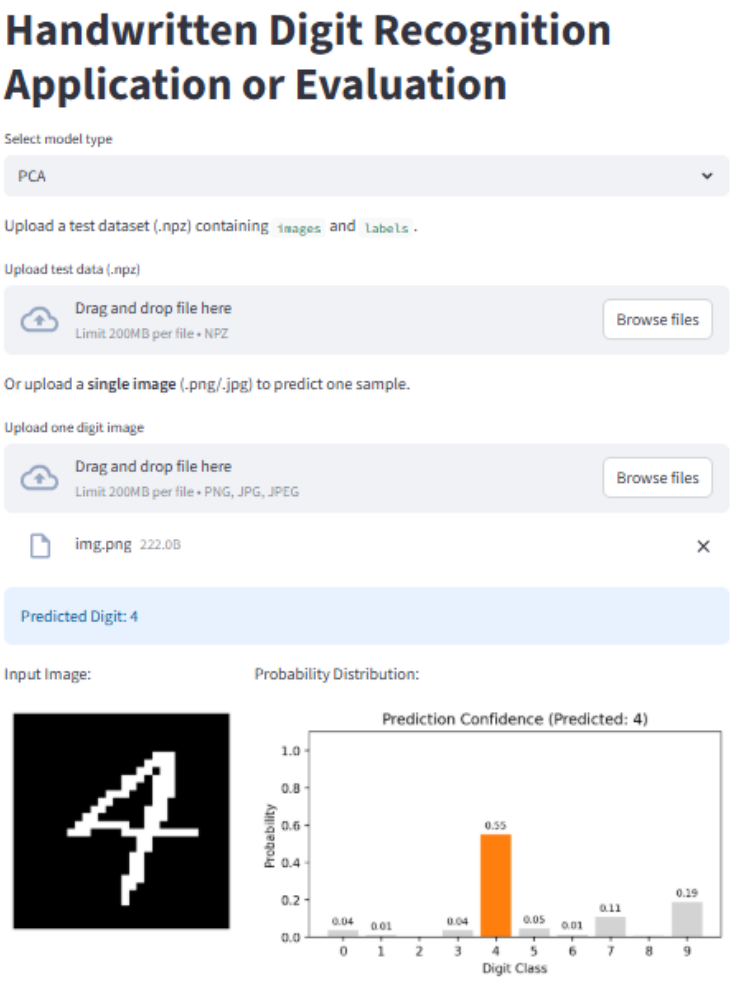
Hình 6.5: Upload ảnh

- Kết quả:



Hình 6.6: Dự đoán ảnh của trích xuất bằng cách flatten

Hình 6.7: Dự đoán ảnh của trích xuất bằng cách edges



Hình 6.8: Dự đoán ảnh của trích xuất bằng cách giảm chiều pca

- **Thông tin thêm về ứng dụng:** Đây là một ứng dụng cơ bản để dự đoán nhãn chữ viết tay từ hình ảnh được xây từ streamlit (một thư viện python) - giúp xấy web cơ bản. Một số chức năng chính: chọn cách trích xuất đặc trưng (Raw Pixels, Edges (Sobel), PCA), mô hình: softmax regression và cho dự đoán nhãn từ hình ảnh và phân bố xác xuất các nhãn,...

CHƯƠNG 7

KẾT LUẬN

7.1. Tóm tắt nội dung và kết quả chính

Trong đồ án này, chúng em đã xây dựng và đánh giá mô hình *Softmax Regression* cho bài toán phân loại chữ số viết tay trên bộ dữ liệu MNIST. Ba dạng đặc trưng đầu vào được khảo sát gồm:

- Đặc trưng **raw pixel** (chuẩn hoá và làm phẳng ảnh),
- Đặc trưng **edges** trích xuất bằng toán tử Sobel,
- Đặc trưng **PCA** sau khi giảm chiều không gian đặc trưng.

7.1.1. Kết quả thực nghiệm cho thấy

- Đặc trưng **raw pixel** đạt kết quả tốt nhất về Accuracy, Precision (macro), Recall (macro) và F1-score (macro) chứng tỏ mô hình tuyến tính vẫn có thể khai thác hiệu quả thông tin trực tiếp từ mức pixel khi dữ liệu được chuẩn hoá tốt.
- Đặc trưng **edges (Sobel)** cho hiệu năng thấp hơn rõ rệt, do việc chỉ giữ lại biên cạnh làm mất đi một phần thông tin cấu trúc bên trong chữ số.
- Đặc trưng **PCA** cho kết quả chỉ kém nhẹ so với raw pixel nhưng có ưu điểm là giảm số chiều, giúp mô hình gọn hơn về mặt lưu trữ và tính toán.

Nhìn chung, cả ba metric Accuracy, Precision (macro), Recall (macro) và F1-score (macro) đều ở mức tương đối đồng đều giữa các lớp, phù hợp với tính chất cân bằng của bộ dữ liệu MNIST.

7.2. Hạn chế và hướng cải thiện

- Softmax Regression là mô hình tuyến tính, do đó khả năng biểu diễn ranh giới phi tuyến giữa các lớp còn hạn chế. Điều này lý giải vì sao độ chính xác chưa đạt tới mức rất cao như các mô hình sâu hiện đại (CNN).
- Trong tương lai, có thể kết hợp đặc trưng edges hoặc PCA với các mô hình phi tuyến (ví dụ: mạng nơ-ron nhiều lớp, CNN) để kiểm tra liệu việc trích chọn đặc trưng có giúp mô hình mạnh hơn tận dụng được cấu trúc hình học của chữ số hay không.

TÀI LIỆU THAM KHẢO

[1] Niklas Donges. *The Curse of Dimensionality in Machine Learning*. https://www.datacamp.com/blog/curse-of-dimensionality-machine-learning. Accessed: 2025-12-10. 2019.

[2] Nguyễn Minh Hải. *Principal component analysis (PCA) là gì? Cách thức hoạt động*. Truy cập ngày 22/11/2025. 2025. URL: https://vnptai.io/vi/blog/detail/principal-component-analysis-la-gi.

[3] OpenCV. *Sobel Derivatives*. https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html. Accessed: 2025-12-09. n.d.