

# Phân loại hình ảnh thời trang bằng học máy

Hoàng Thiết Lâm Phạm Gia Nguyên Bùi Hữu Phước

December 18, 2024

## Giới thiệu

## Cơ sở lý thuyết

PCA  
TSNE  
Hàm Softmax  
Logistic Regression  
CNN  
Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu  
Thực nghiệm t-SNE  
Thực nghiệm PCA  
Thực nghiệm Logistic Regression  
Thực nghiệm CNN

## Kết luận

- **Phân loại ảnh** là một bài toán kinh điển trong lĩnh vực **Xử lý ảnh và Thị giác máy tính**.
- Bản báo cáo tập trung vào việc phân loại ảnh thời trang nhằm hỗ trợ người **khiêm khuyết về thị giác** trong việc mua sắm quần áo.
- Nội dung chính:
  - Sử dụng các **thuật toán giảm chiều** để:
    - Giảm độ phức tạp tính toán của bộ dữ liệu có số chiều cao.
    - Kiểm tra sự tồn tại của các phân cụm trong bộ dữ liệu.
  - Áp dụng và so sánh các **mô hình phân loại** khác nhau để đánh giá hiệu quả.

# Phân chia công việc giữa kỳ

## Giới thiệu

### Cơ sở lý thuyết

PCA  
TSNE  
Hàm Softmax  
Logistic Regression  
CNN  
Độ đo đánh giá

### Thực nghiệm và đánh giá

Dữ liệu  
Thực nghiệm t-SNE  
Thực nghiệm PCA  
Thực nghiệm Logistic Regression  
Thực nghiệm CNN

### Kết luận

STT	Công Việc	Người Phụ Trách
1	Lý thuyết và thực nghiệm Logistic Regression	Bùi Hữu Phước
2	Lý thuyết và thực nghiệm Convolutional Neural Network	Hoàng Thiết Lâm
3	Lý thuyết và thực nghiệm Principal Component Analysis	Phạm Gia Nguyên
4	Lý thuyết và thực nghiệm T-Distributed Neighbor Embedding	Hoàng Thiết Lâm

Table: Danh sách công việc và phân chia phụ trách

# Phân Tích Thành Phần Chính (PCA)

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- **PCA** là một phương pháp giảm số chiều của dữ liệu một cách hiệu quả.
- **Mục đích:** Tìm các thành phần chính (*Principal Components*) để giữ lại phần lớn phương sai của dữ liệu.

## Mục tiêu của PCA:

- Giảm số chiều của dữ liệu nhưng vẫn giữ lại phần lớn thông tin quan trọng.
- Tìm một hệ cơ sở trực chuẩn bao gồm các vector vuông góc (các thành phần chính).

# Dữ Liệu trong PCA

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Dữ liệu đầu vào:

- Ma trận dữ liệu  $\mathbf{X} \in \mathbb{R}^{n \times d}$ .
  - $n$ : Số lượng điểm dữ liệu.
  - $d$ : Số chiều của dữ liệu ban đầu.
- Số chiều mục tiêu  $k$ , với  $k < d$ .

## Dữ liệu đầu ra:

- Ma trận dữ liệu giảm chiều  $\mathbf{Z} \in \mathbb{R}^{n \times k}$ .
- Ma trận thành phần chính  $\mathbf{W} \in \mathbb{R}^{d \times k}$ .

# Các bước thực hiện PCA

## 1. Chuẩn hóa dữ liệu:

$$\mathbf{x}' = \frac{\mathbf{x} - \mu}{\sigma}$$

- $\mu$ : Vector trung bình,  $\sigma$ : Vector độ lệch chuẩn.
- Kết quả là ma trận chuẩn hóa  $\mathbf{X}' \in \mathbb{R}^{n \times d}$ .

## 2. Tính ma trận hiệp phương sai:

$$\mathbf{C} = \frac{1}{n-1} \mathbf{X}'^\top \mathbf{X}'$$

## 3. Tính giá trị riêng và vector riêng:

$$\mathbf{Cv}_i = \lambda_i \mathbf{v}_i$$

## 4. Chọn $k$ vector riêng tương ứng với $k$ giá trị riêng lớn nhất.

## 5. Chiếu dữ liệu:

$$\mathbf{Z} = \mathbf{X}' \mathbf{W}, \quad \mathbf{W} \in \mathbb{R}^{d \times k}$$

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



# Minh họa các bước PCA

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## PCA procedure

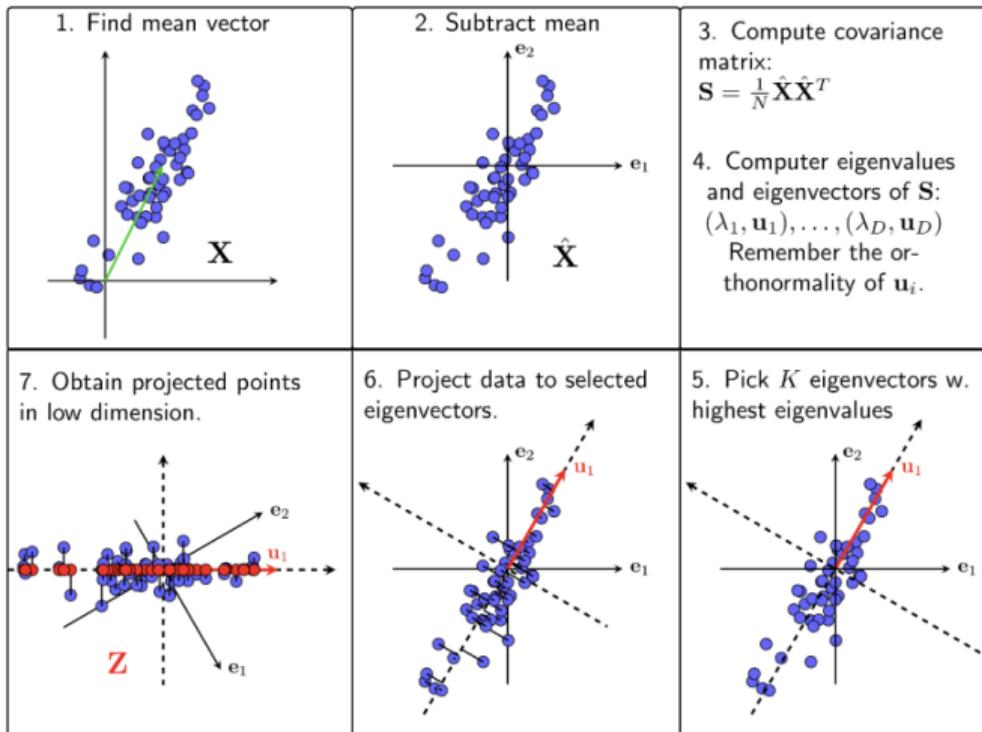


Figure: Các bước PCA

# Phương sai tích lũy trong PCA

## Phương sai tích lũy:

$$\text{Phương sai tích lũy} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{j=1}^d \lambda_j}$$

- Giúp xác định tỉ lệ thông tin (phương sai) giữ lại khi chọn  $k$  thành phần chính.
- Biểu đồ phương sai tích lũy giúp tìm điểm  $k$  tối ưu ("Elbow point").

## Biểu đồ phương sai tích lũy:

- Trục hoành: Số thành phần chính  $k$ .
- Trục tung: Phương sai tích lũy.

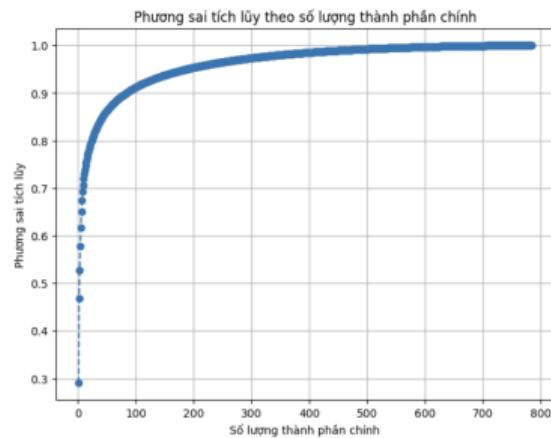


Figure: Biểu đồ phương sai tích lũy.

Giới thiệu

Cơ sở lý thuyết

PCA

T-SNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

## Phân rã giá trị suy biến (SVD):

- Cho ma trận dữ liệu chuẩn hóa  $\mathbf{X} \in \mathbb{R}^{n \times d}$ , SVD phân rã  $\mathbf{X}$  thành ba ma trận:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$$

- $\mathbf{U} \in \mathbb{R}^{n \times n}$ : Ma trận trực giao chứa các vector riêng của  $\mathbf{X}\mathbf{X}^\top$ .
- $\Sigma \in \mathbb{R}^{n \times d}$ : Ma trận đường chéo chứa các **giá trị suy biến** sắp xếp giảm dần.
- $\mathbf{V} \in \mathbb{R}^{d \times d}$ : Ma trận trực giao chứa các **vector riêng** của  $\mathbf{X}^\top\mathbf{X}$  (thành phần chính).
- SVD giúp tìm các thành phần chính mà không cần tính ma trận hiệp phương sai  $\mathbf{C}$ .

Giới thiệu

Cơ sở lý thuyết

PCA

T-SNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

## Mối quan hệ giữa SVD và PCA:

- Các cột của  $\mathbf{V}$  chính là các thành phần chính (*Principal Components*).
- Các giá trị trên đường chéo của  $\Sigma$  liên quan đến phương sai theo các thành phần chính.
- SVD là phương pháp  **ổn định và hiệu quả** khi xử lý dữ liệu lớn.

## Kết quả từ SVD:

- Các vector trong ma trận  $\mathbf{V}$  là **thành phần chính**.
- Giá trị suy biến trong  $\Sigma$  tương ứng với độ lớn của phương sai.
- Dữ liệu giảm chiều được tính như sau:

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_k$$

với  $\mathbf{V}_k \in \mathbb{R}^{d \times k}$  chứa  $k$  thành phần chính đầu tiên.

# t-Distributed Stochastic Neighbor Embedding

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Thuật toán **t-Distributed Stochastic Neighbor Embedding(t-SNE)** là một thuật toán **giảm chiều phi tuyến** và **trực quan hóa dữ liệu nổi tiếng**.
- So với kĩ thuật khác, t-SNE tốt hơn bởi việc tạo một không gian mà phát hiện được các cấu trúc ở những **độ đo khác nhau**.
- Là **biến thể** của Stochastic Neighbor Embedding, dễ tối ưu hơn, trực quan hóa dữ liệu tốt hơn.



# Thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- Cho  $\{X_i\}_{1 \leq i \leq n}$  là tập điểm dữ liệu với số chiều là  $p$ , số mẫu là  $n$ .
- Thuật toán bắt đầu với tính toán *phân phối xác suất đồng thời* trên *tất cả cặp* dữ liệu  $\{(X_i, X_j)\}_{1 \leq i \neq j \leq n}$ .
- *Mã trận đối xứng* biểu diễn các phân phối:  $\mathbf{P} = (p_{ij})_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$ .
- *Mã trận  $\mathbf{P}_{ij}$*  kiểm tra dữ liệu  $X_i$  và dữ liệu  $X_j$  có gần nhau không
- Với  $p_{ii} = 0$  với mọi  $1 \leq i \leq n$  và với  $i \neq j$  thì,

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}, \quad p_{j|i} = \frac{\exp\left(-\frac{\|X_i - X_j\|_2^2}{2\tau_i^2}\right)}{\sum_{\ell \in \{1, 2, \dots, n\} \neq i} \exp\left(-\frac{\|X_i - X_\ell\|_2^2}{2\tau_i^2}\right)} \quad (1)$$

- $\tau_i$  là *tham số hiệu chỉnh*, được chọn dựa trên một độ đo gọi là *perplexity*(đánh giá mô hình xác suất dự đoán mẫu tốt như thế nào) và *chiến lược tìm kiếm nhị phân*.

# Thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- Mục tiêu tạm thời: Giảm chiều về *hai chiều*
- Đối với không gian 2 chiều  $\{y_i\}_{1 \leq i \leq n} \subset \mathbb{R}^2$ .
- Tính *phân phối xác suất đồng thời* cho tất cả các cặp  $\{(y_i, y_j)\}_{1 \leq i \neq j \leq n}$ .
- Ma trận đối xứng biểu diễn các phân phối:  $\mathbf{Q} = (q_{ij})_{1 \leq i,j \leq n} \in \mathbb{R}^{n \times n}$
- Ma trận  $\mathbf{Q}_{ij}$  kiểm tra dữ liệu  $y_i$  và dữ liệu  $y_j$  có gần nhau không

$$q_{ij} = \frac{(1 + \|y_i - y_j\|_2^2)^{-1}}{\sum_{\ell, s \in \{1, 2, \dots, n\}, \ell \neq s} (1 + \|y_\ell - y_s\|_2^2)^{-1}} \quad (2)$$

# Thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- **P** và **Q** được coi là các *ma trận tương đồng*, tính toán khoảng cách theo từng cặp của dữ liệu nhiều chiều  $\{X_i\}_{1 \leq i \leq n}$  và dữ liệu hai chiều  $\{y_i\}_{1 \leq i \leq n}$ .
- Định nghĩa ma trận tương đồng: Ma trận tương đồng  $M_{ij}$  so sánh độ tương đồng giữa đối tượng tại vị trí  $i$  và tại vị trí  $j$ . Ma trận  $M \in \mathbb{R}^{n \times n}$
- Ma trận  $P_{ij}$  kiểm tra độ tương đồng với dữ liệu  $X_i$  và dữ liệu  $X_j$  ở số chiều không gian gốc
- Ma trận  $Q_{ij}$  kiểm tra độ tương đồng với dữ liệu  $y_i$  và dữ liệu  $y_j$  ở số chiều không gian đã giảm chiều

# Thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Thuật toán tìm  $\{y_i\}_{1 \leq i \leq n}$  mà tối thiểu hóa độ đo *Kullback-Leibler divergence* giữa  $\mathbf{P}$  và  $\mathbf{Q}$  với

$$(y_1, \dots, y_n) = \arg \min_{y_1, \dots, y_n} D_{KL}(\mathbf{P}, \mathbf{Q}) = \arg \min_{y_1, \dots, y_n} \sum_{i,j \in \{1,2,\dots,n\}, i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3)$$

- Độ đo *Kullback-Leibler divergence* xác định mức độ khác biệt giữa 2 phân phối xác suất P và Q. Nếu xét đến ma trận, P và Q thường là ma trận tương đồng.
- Độ đo này đo lường mức độ phân phối P *sai lệch* so với phân phối Q.
- Nếu P và Q *giống nhau hoàn toàn* thì  $D_{KL}(P, Q) = 0$
- *Không đối xứng*:  $D_{KL}(P, Q) \neq D_{KL}(Q, P)$  để nhấn mạnh các phần của P mà Q sai lệch

# Thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



$$(y_1, \dots, y_n) = \arg \min_{y_1, \dots, y_n} D_{KL}(\mathbf{P}, \mathbf{Q}) = \arg \min_{y_1, \dots, y_n} \sum_{i,j \in \{1,2,\dots,n\}, i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3)$$

- Mục tiêu của tối thiểu hóa  $D_{KL}(\mathbf{P}, \mathbf{Q})$ : Điều chỉnh các điểm dữ liệu ở không gian giảm chiều  $\mathbf{Q}$  khớp nhất với  $\mathbf{Q}$ , giữ lại các mối quan hệ ở không gian gốc.
- Khi cặp dữ liệu thứ  $i$  và  $j$  có  $p_{ij}$  cao nhưng  $q_{ij}$  nhỏ thì  $D_{KL}(\mathbf{P}, \mathbf{Q})$  lớn, yêu cầu việc tối ưu sao cho  $q_{ij}$  tăng lên.
- Thuật toán để giải bài toán tối ưu (3) là biến thể của Gradient Descent.

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



Phương trình cập nhật cho bài toán tối ưu (3) là:

$$y_i^{(k+1)} = y_i^{(k)} + hD_i^{(k)} + m^{(k+1)} \left( y_i^{(k)} - y_i^{(k-1)} \right) \quad \text{for } i = 1, \dots, n. \quad (4)$$

- $k$  là chỉ số lần lặp thứ mấy
- $h \in \mathbb{R}$  là tham số độ dài bước(chỉ định trước)
- $D_i^{(k)} = 4 \sum_{j \neq i} \left( y_j^{(k)} - y_i^{(k)} \right) S_{ij}^{(k)} \in \mathbb{R}^2$  là hệ số gradient ứng với điểm dữ liệu  $y_i$
- $S_{ij}^{(k)} = (p_{ij} - q_{ij}^{(k)}) / (1 + \|y_i^{(k)} - y_j^{(k)}\|_2^2) \in \mathbb{R}$
- $m^{(k)}$  là tham số momentum(động lượng)(chỉ định trước).

# Thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Khởi tạo:  $y_i^0$  với  $i \in \{1, 2, \dots, n\}$  được lấy từ *phân phối đồng nhất* trên  $[-0.01, 0.01]^2$  hoặc từ *phân phối Gaussian*  $N(0, \delta^2 I)$  với  $\delta > 0$ .
  - Những giá trị khởi tạo sẽ *tập trung quanh 0*, đảm bảo giá trị khởi tạo *không quá lớn*.
  - Phần động lượng  $m^{(k+1)}(y_i^{(k)} - y_i^{(k-1)})$  chỉ nhằm *tăng tốc độ hội tụ*, tránh bị kẹt ở *cực tiểu địa phương*.
  - Số lần lặp khuyến cáo là 1000.
  - Độ dài bước đặt ban đầu là 400 hoặc 800, sẽ được cập nhật mỗi lần lặp.
- Đơn giản hóa, xét công thức *Gradient Descent gốc*:

$$y_i^{(k+1)} = y_i^{(k)} + hD_i^{(k)}, \quad \text{for } i = 1, \dots, n \quad (5)$$

# Tối ưu hóa thuật toán t-SNE

## Giới thiệu

## Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

## Kết luận

- Vấn đề của thuật toán (5): Tốc độ *hội tụ chậm*, thậm chí *không hội tụ*
- Kỹ thuật để xử lý: Early Exaggeration(làm quá ban đầu).
- Gồm 2 giai đoạn: Early Exaggeration và Embedding(Biểu diễn)
- Giúp *tạo ra đặc trưng* để trực quan hóa và *tăng tốc độ hội tụ*



# Tối ưu hóa thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Bước làm quá ban đầu

- Với  $K_0$  lần lặp đầu tiên, hệ số  $p_{ij}$  trong hệ số gradient  $D_i^k$  (ứng với  $y_i$ ) được nhân với một hệ số làm quá  $\alpha > 0$ . Phương trình (5) trở thành:

$$y_i^{(k+1)} = y_i^{(k)} + hD_i^{(k)}, \quad \text{for } i = 1, \dots, n \quad (5)$$

- với  $S_{ij}^{(k)}(\alpha) = (\alpha p_{ij} - q_{ij}^{(k)}) / (1 + \|y_i^{(k)} - y_j^{(k)}\|_2^2) \in \mathbb{R}$
- Hằng số 4 trong  $D_i^k$  đã được hấp thụ vào độ dài bước h
- Tham số được khuyến cáo năm 2008:  $\alpha = 4$  và  $K_0 = 50$ , và vào năm 2014 là  $\alpha = 12$  và  $K_0 = 250$ .

# Tối ưu hóa thuật toán t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Bước làm quá ban đầu

- Với  $K_0$  lần lặp đầu tiên, hệ số  $p_{ij}$  trong hệ số gradient  $D_i^k$  (ứng với  $y_i$ ) được nhân với một hệ số làm quá  $\alpha > 0$ . Phương trình (5) trở thành:

$$y_i^{(k+1)} = y_i^{(k)} + hD_i^{(k)}, \quad \text{for } i = 1, \dots, n \quad (5)$$

- với  $S_{ij}^{(k)}(\alpha) = (\alpha p_{ij} - q_{ij}^{(k)}) / (1 + \|y_i^{(k)} - y_j^{(k)}\|_2^2) \in \mathbb{R}$
- Hằng số 4 trong  $D_i^k$  đã được hấp thụ vào độ dài bước h.
- Tham số được khuyến cáo năm 2008 là  $\alpha = 4$  và  $K_0 = 50$ , và vào năm 2014 là  $\alpha = 12$  và  $K_0 = 250$ .

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Các vấn đề của bước làm quá ban đầu

- Dù bước làm quá ban đầu đã giúp tăng tốc độ, tuy nhiên tại sao lại sản sinh kết quả tốt hơn vẫn chưa được hiểu rõ.
- Độ hiệu quả của không gian đã giảm chiều chưa chắc chắn.
- Việc lựa chọn các tham số ( $\alpha, h, K_0$ ) ảnh hưởng như thế nào.
- Cách chọn các tham số ( $\alpha, h, K_0$ ).

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Bước biểu diễn

- Sau khi hoàn thành bước làm quá ban đầu, tham số  $\alpha$  bị loại bỏ.
- Thuật toán (5) được khôi phục đến khi đạt giới hạn lặp
- Đầu ra cuối: Không gian 2 chiều  $\{y_i^{K_1}\}_{1 \leq i \leq n} \in \mathbb{R}^2$  với  $K_1$  là chỉ số lần lặp cuối cùng
- Không gian giảm chiều vẫn giữ được những cấu trúc quan trọng.

# Tối ưu hóa thuật toán t-SNE

## Giới thiệu

## Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu

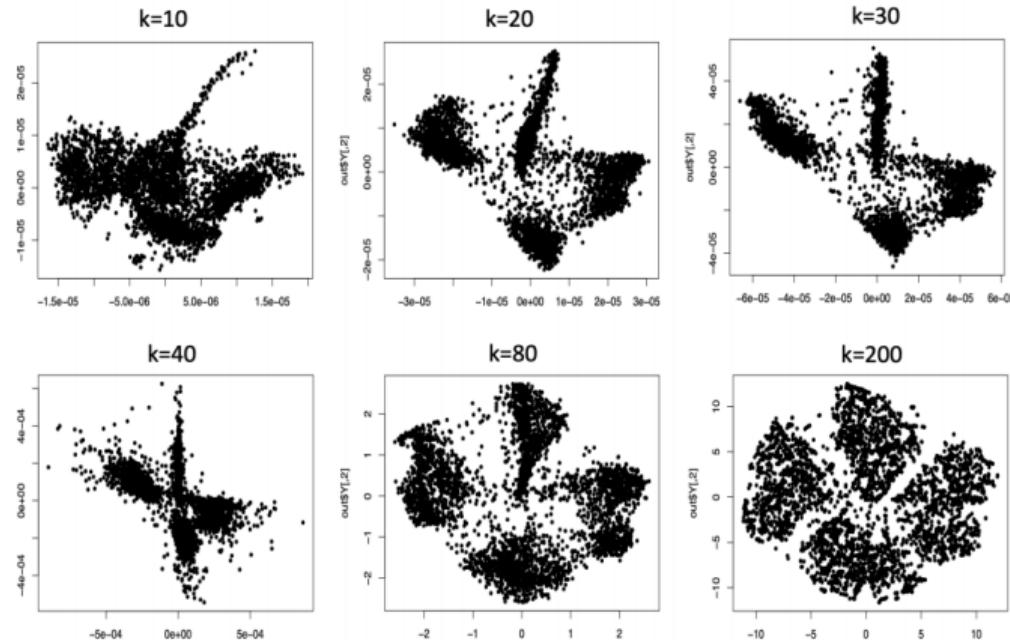
Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

## Kết luận



**Figure:** Hình biểu diễn các bước lặp của t-SNE từ tập dữ liệu MNIST. Mỗi mẫu dữ liệu ứng với ảnh viết tay của số 2/4/6/8. Tham số được sử dụng là perplexity = 30,  $\alpha = 12$ ,  $h = 200$ ,  $K_0 = 40$ . Ba hàng đầu ứng với bước làm quá ban đầu, ba hàng cuối ứng với bước biểu diễn.

# Hàm Softmax

## Giới thiệu

## Cơ sở lý thuyết

PCA  
TSNE  
Hàm Softmax  
Logistic Regression  
CNN  
Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu  
Thực nghiệm t-SNE  
Thực nghiệm PCA  
Thực nghiệm Logistic Regression  
Thực nghiệm CNN

## Kết luận

- Trong bài toán phân loại nhiều lớp, hàm Softmax được sử dụng để chuyển đổi đầu ra của mô hình thành xác suất cho từng lớp.



## Giới thiệu

## Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

## Kết luận



- Trong bài toán phân loại nhiều lớp, hàm Softmax được sử dụng để chuyển đổi đầu ra của mô hình thành xác suất cho từng lớp.
- Công thức của hàm Softmax được xây dựng trên tập  $\{z_1, z_2, \dots, z_C\}$ :

$$g(z_k) = \frac{\exp(z_k)}{\sum_{j=1}^C \exp(z_j)}; \text{ với } k = 1, 2, \dots, C.$$

Trong đó:

- $z_k$  là đầu ra logit của mô hình tại node thứ  $k$ .
- $C$  là số lớp của bài toán.

# Tính chất hàm Softmax

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

■  $g(z): \mathbb{R} \rightarrow (0, 1)$ .

■  $\sum_{k=1}^C g(z_k) = 1$ .

■ Xét trong  $\{z_1, z_2, \dots, z_C\}$ ,  $g(z)$  tăng chéo:

$$z_i > z_j \Leftrightarrow g(z_i) > g(z_j)$$

■ Hàm số đủ trơn và đạo hàm tính được dễ dàng.



# Tính chất hàm Softmax



Figure: Ảnh minh họa tính chất của hàm Softmax.

# Hàm Softmax ổn định

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Khi  $z_i$  quá lớn, việc tính toán  $\exp(z_i)$  có thể gây ra hiện tượng tràn số.



# Hàm Softmax ổn định

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Khi  $z_i$  quá lớn, việc tính toán  $\exp(z_i)$  có thể gây ra hiện tượng tràn số.
- Ta khắc phục bằng cách sử dụng phiên bản ổn định hơn của Softmax:

$$g(z_k) = \frac{\exp(z_k - M)}{\sum_{j=1}^C \exp(z_j - M)}; \text{ với } M \text{ là hằng số bất kỳ.}$$

- Trong thực nghiệm, M thường được chọn là:  $M = \max(z_i)$ .

# Softmax Regression

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Mô hình Softmax Regression là một mô hình phân loại trong học máy.
- Softmax Regression thuộc loại mô hình phân biệt và được xếp vào nhóm mô hình tuyến tính (linear model).
- Mô hình này có kiến trúc đơn giản nhưng vẫn hiệu quả và có vai trò quan trọng ngày nay.



Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Các biến quan sát  $\mathbf{x}$  thuộc không gian  $d$  chiều :  $\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^d$  và  $\bar{\mathbf{x}} = (1 \ \mathbf{x})^T$ .
- Nhãn tương ứng là  $y$ , với  $y \in \{1, 2, \dots, C\}$ .
- Với  $C$  lớp chúng ta cần  $C$  vector tham số  $\theta_1, \theta_2, \dots, \theta_C$ .

$$\theta_k = (\theta_{0,k} \ \theta_{1,k} \dots \theta_{d,k})^T$$

- Như vậy, các tham số tạo thành một ma trận  $\boldsymbol{\theta} \in \mathbb{R}^{(d+1) \times C}$ :

$$\begin{pmatrix} \theta_{0,1} & \theta_{0,2} & \dots & \theta_{0,C} \\ \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,C} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{d,1} & \theta_{d,2} & \dots & \theta_{d,C} \end{pmatrix}$$

# Kiến trúc của Softmax Regression

Giới thiệu

Cơ sở lý thuyết

PCA

T-SNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Multi-Class Classification with NN and SoftMax Function

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \\ \vdots \\ \mathbf{w}_K^\top \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

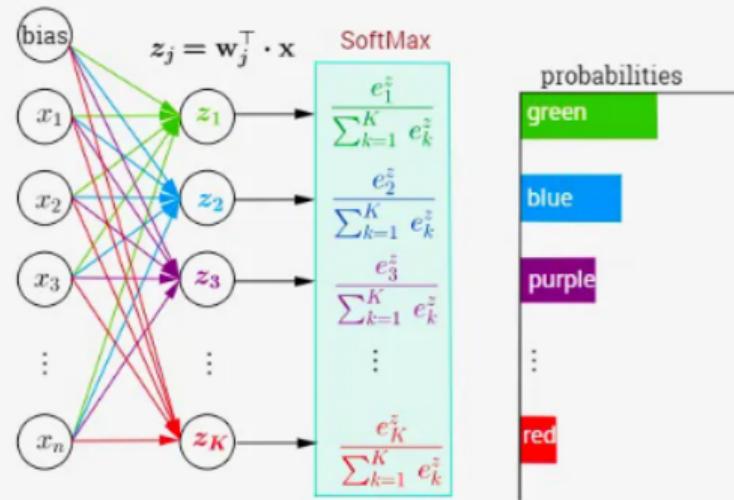


Figure: Mô hình Softmax Regression.

# Hàm mất mát của Softmax Regression

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Ta có tập huấn luyện ( $X, Y$ ):  $X = \{x_n\}_{n=1}^N; Y = \{y_n\}_{n=1}^N$ ,  $y_n \in \{1, 2, \dots, C\}$ .

- Hàm mất mát được xây dựng thông qua sử dụng MLE:

$$J(\theta) = -\frac{1}{N} \sum_{n=1}^N \left\{ \theta_{y_n}^T \mathbf{x}_n - \log \left[ \sum_{k=1}^C \exp (\theta_k^T \mathbf{x}_n) \right] \right\}$$



# Hàm mất mát của Softmax Regression

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Ta có tập huấn luyện ( $X, Y$ ):  $X = \{x_n\}_{n=1}^N; Y = \{y_n\}_{n=1}^N$ ,  $y_n \in \{1, 2, \dots, C\}$ .

- Hàm mất mát được xây dựng thông qua sử dụng MLE:

$$J(\theta) = -\frac{1}{N} \sum_{n=1}^N \left\{ \theta_{y_n}^T \mathbf{x}_n - \log \left[ \sum_{k=1}^C \exp (\theta_k^T \mathbf{x}_n) \right] \right\} + \frac{\lambda}{2} \sum_{k=1}^C \sum_{j=1}^d \theta_{j,k}^2 \quad (1)$$

- Bài toán tối ưu: Tìm  $\theta^* = \arg \min_{\theta} J(\theta)$ .

# Tối ưu hàm mất mát bằng thuật toán SGD

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

■  $\bar{\mathbf{y}_n}$  là vector one-hot-coding của  $y_n$ .

■  $p_{k,n} := P(y_n = k \mid \mathbf{x}_n; \theta) = g(\theta_k^T \bar{\mathbf{x}_n})$  là xác suất điểm dữ liệu thuộc lớp thứ  $k$  theo mô hình.



# Tối ưu hàm mất mát bằng thuật toán SGD

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- $\bar{\mathbf{y}_n}$  là vector one-hot-coding của  $y_n$ .
- $p_{k,n} := P(y_n = k \mid \mathbf{x}_n; \theta) = g(\theta_k^T \bar{\mathbf{x}_n})$  là xác suất điểm dữ liệu thuộc lớp thứ  $k$  theo mô hình.
- Đặt  $e_{k,n} := \bar{y}_{k,n} - p_{k,n}$
- Như vậy,  $\mathbf{e}_n := \bar{\mathbf{y}}_n - \mathbf{p}_n$  là vector sai số giữa one-hot-coding của nhãn và vector xác suất phân lớp của mô hình tương ứng với  $\mathbf{x}_n$ .

# Tối ưu hàm mất mát bằng thuật toán SGD

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



1. **Đầu vào:** tham số học  $\eta > 0$ ,  $(X, Y)$ . **Khởi tạo:**  $\theta \in \mathbb{R}^{(d+1) \times C}$  ngẫu nhiên.
2. Tráo đổi ngẫu nhiên thứ tự các cặp dữ liệu  $(x_n, y_n)$ . Lần lượt  $n = 1, 2, \dots, N$ , thực hiện:
3. Tính vector xác suất phân lớp đầu ra:

$$\mathbf{p}_n = (p_{1,n} \ p_{2,n} \ \dots \ p_{C,n})^T$$

# Tối ưu hàm mất mát bằng thuật toán SGD

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

4. Tính gradient  $\frac{\partial J(\theta)}{\partial \theta}$ :

$$\frac{\partial J(\theta)}{\partial (\theta)} = -\mathbf{x}_n \mathbf{e}_n^T + \lambda \theta'$$

Trong đó  $\theta' = \begin{pmatrix} 0 & 0 & \dots & 0 \\ \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,C} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{d,1} & \theta_{d,2} & \dots & \theta_{d,C} \end{pmatrix}$

5. Cập nhật tham số:

$$\theta := \theta - \eta \cdot \frac{\partial J(\theta)}{\partial (\theta)}$$

6. Nếu đạt đủ điều kiện dừng, kết thúc. Ngược lại, quay lại bước 2.

# Convolutional Neural Networks

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- Convolutional Neural Network(CNN) hay Mạng neuron tích chập là một mô hình rất nổi tiếng trong lĩnh vực *học sâu*.
- Thường sử dụng trong *xử lý ảnh*.
- Khả năng *tự động* trích xuất đặc trưng.

# Cấu trúc cơ bản

## Giới thiệu

## Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu

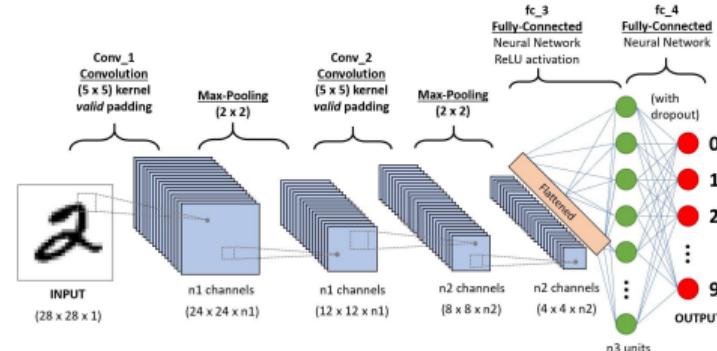
Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

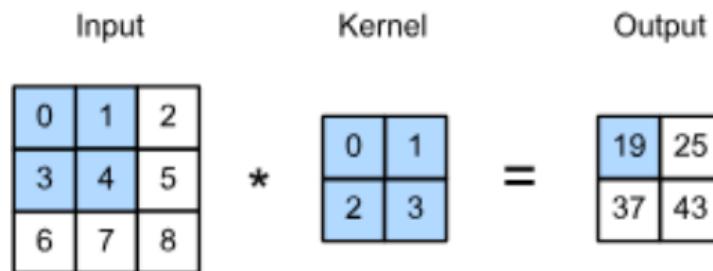
## Kết luận



- Các lớp được sắp xếp **tuần tự**
- Tầng **Convolution**
- Tầng **Pooling**
- Tầng **Flatten**
- Tầng **Fully-Connected**
- Tầng **Batch Normalization** và Tầng **Dropout**

# Tầng Convolution

Thực hiện phép tích chập giữa dữ liệu đầu vào(ảnh) và các bộ lọc nhằm trích xuất đặc trưng quan trọng

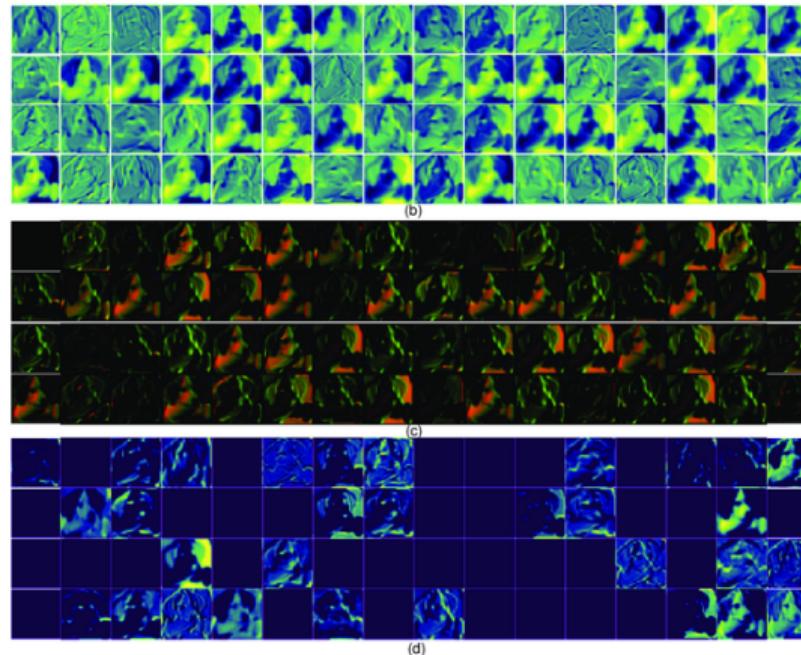


## Các thành phần:

- **Dữ liệu đầu vào:** Ảnh với kích thước (Height X Width X Channels)
- **Bộ lọc(Filter/Kernel):** Ma trận nhỏ, các giá trị khởi tạo ngẫu nhiên hoặc học được qua huấn luyện. Các loại bộ lọc ứng với các đặc trưng của ảnh
- **Phép tích chập:** Trượt bộ lọc trên ma trận đầu vào từ trái -> phải, trên -> dưới. Lấy phép nhân từng phần tử giữa ma trận con và bộ lọc -> lấy tổng.

# Tầng Convolution

- Kết quả của phép toán tích chập là một *hình ảnh*(có kích cỡ bé hơn ảnh gốc)
- Gồm các đặc trưng trích xuất được, gọi là *ảnh đặc trưng*(feature map)



Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



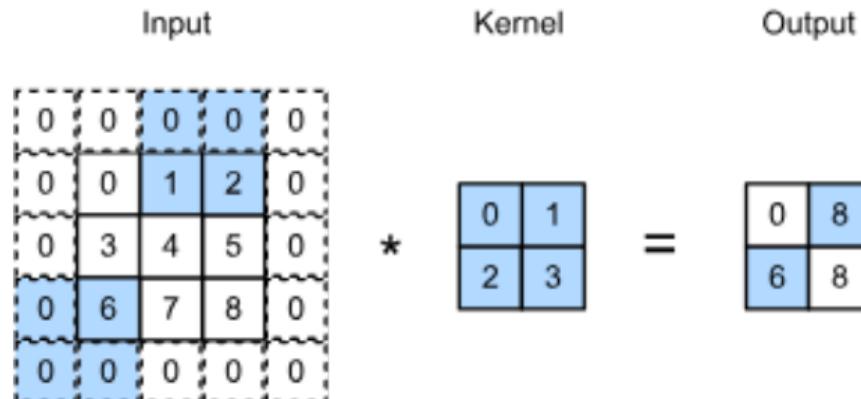
## Các siêu tham số:

- **Số lượng bộ lọc:** Càng nhiều bộ lọc, càng nhiều đặc trưng trích xuất.
- **Kích thước bộ lọc:** Thường có cỡ  $3 \times 3$ .
- **Bước nhảy (Stride):** Khoảng cách *giữa các lần trượt* của bộ lọc trên ảnh đầu vào. Mặc định bước nhảy = 1, có thể dùng bước nhảy lớn hơn để có ít lần trượt hơn
- **Padding:** Khi áp dụng tầng Convolution sẽ xảy ra hiện tượng không sử dụng pixels ở *rim ảnh*. Padding thêm các giá trị(0) nhằm *kiểm soát kích cỡ ảnh* đặc trưng đầu ra.

# Tầng Convolution

- $n_h \times n_w$  là kích thước ảnh đầu vào
- $k_h \times k_w$  là kích thước của bộ lọc
- $s_h, s_w$  là bước nhảy cho chiều dài và chiều rộng
- $p_h, p_w$  là số hàng padding cho chiều dài và chiều rộng
- Kích thước đầu ra là:

$$\lfloor (n_h - k_h + p_h + s_h) / s_h \rfloor \times \lfloor (n_w - k_w + p_w + s_w) / s_w \rfloor.$$



Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Tầng Pooling dùng để *giảm chiều dữ liệu*, giảm số lượng tham số.
- Tương tự tầng Convolution, sử dụng bộ lọc trượt qua ảnh đầu vào.
- Nhưng bộ lọc này *sẽ không có tham số*.
- Bộ lọc áp dụng hàm *tổng hợp các giá trị* trong vùng đang trượt, điền vào mảng đầu ra.

## 2 loại Pooling chính:

- Max Pooling: Trong vùng đang trượt qua, chọn ra pixel có giá trị lớn nhất
- Average Pooling: Trong vùng đang trượt qua, lấy giá trị trung bình các pixels

# Tầng Pooling

Giới thiệu

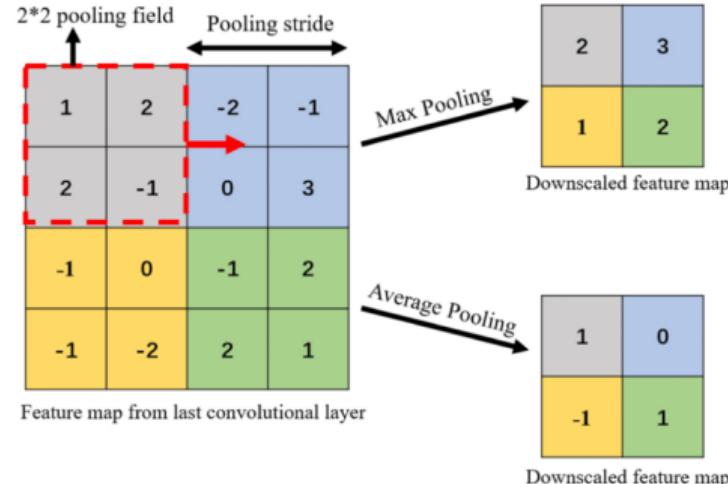
Cơ sở lý thuyết

PCA  
TSNE  
Hàm Softmax  
Logistic Regression  
CNN  
Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu  
Thực nghiệm t-SNE  
Thực nghiệm PCA  
Thực nghiệm Logistic Regression  
Thực nghiệm CNN

Kết luận



## Các siêu tham số:

- Kích thước bộ lọc: Thường có cỡ  $2 \times 2$ .
- Bước nhảy: Khoảng cách giữa các lần trượt
- Padding: Tương tự Padding ở tầng Convolution.

# Tầng Flatten

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

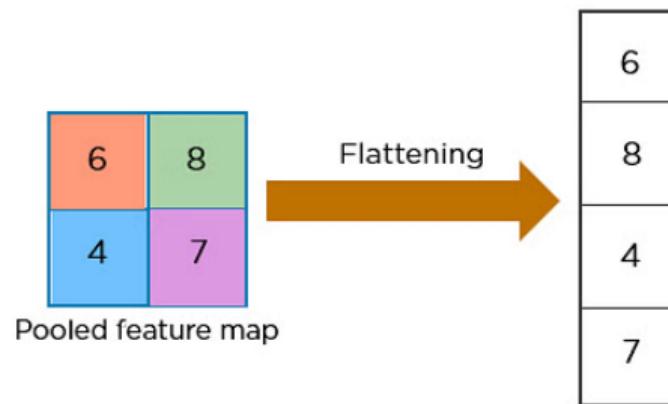
Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Làm phẳng ma trận về vector 1 chiều.
- Kết quả của tầng Flatten là đầu vào cho tầng Fully-connected.



# Tầng Fully-connected

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

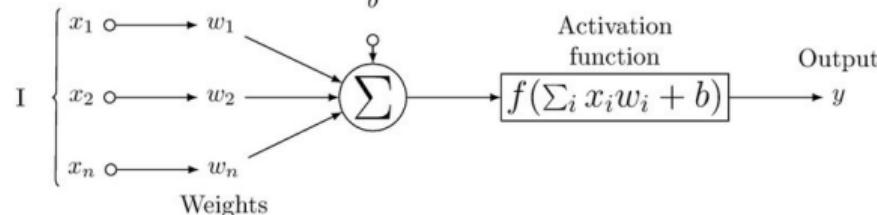
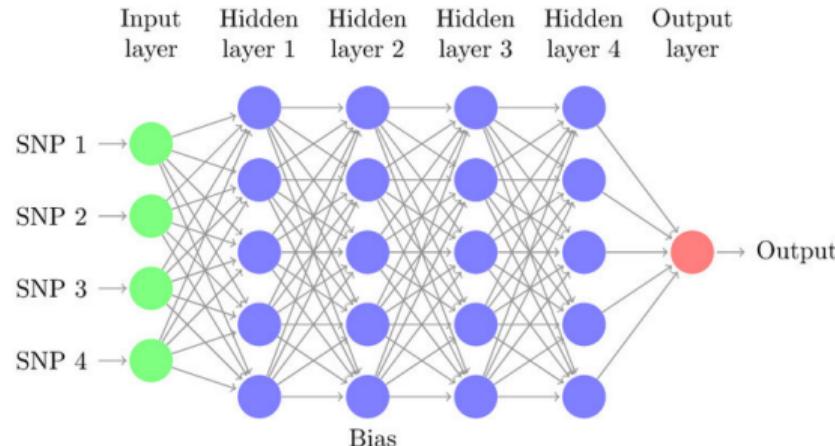


Figure: Minh họa cấu trúc của tầng Fully-connected

# Tầng Fully-connected

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Tầng này sử dụng cấu trúc của *Multi-layers Perceptron*(MLP)
- *Tất cả* các node trong mạng đều *liên kết* với nhau
- Đầu vào là *kết quả trích xuất đặc trưng* qua tầng Convolution, Pooling, Flatten.
- Trong các tầng Hidden, *hàm kích hoạt* được sử dụng để tạo ra tính *phi tuyến*.
- Kết quả *đầu ra* sẽ đưa qua *hàm kích hoạt* để xác định *xác suất dữ liệu* thuộc *một lớp*.
- Hàm kích hoạt Sigmoid: Phân loại *nhi phân*. Hàm kích hoạt Softmax: Phân loại *nhiều lớp*.
- Với  $L$  là số layer của Multi-layers Perceptron, số layer của tầng Fully-connected là  $L+1$ .

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- Đối với các mô hình MLP, *tất cả* các node trong mạng đều liên kết với nhau.
- Giúp mô hình học được *tất cả* các mối quan hệ giữa đầu ra và đầu vào.
- Tuy nhiên, việc thiếu dữ liệu kết hợp với tính chất liên kết trên khiến mô hình học được **đoạn** **rút** **ra** **từ** **dữ** **liệu** **nhiều**
- Những mối quan hệ đó chỉ có ở tập Training, không có ở tập Test => Overfitting
- Các phương pháp hiệu chỉnh cho Overfitting: Tham số hiệu chỉnh L1 L2, Dừng huấn luyện khi kết quả Validation giảm.
- Cách **tốt nhất** để hiệu chỉnh mô hình: **Lấy trung bình các dự đoán của tất cả các bộ tham số**.
- Tuy nhiên, cách này sẽ chỉ phù hợp với mô hình nhỏ và đơn giản.

# Tầng Dropout

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

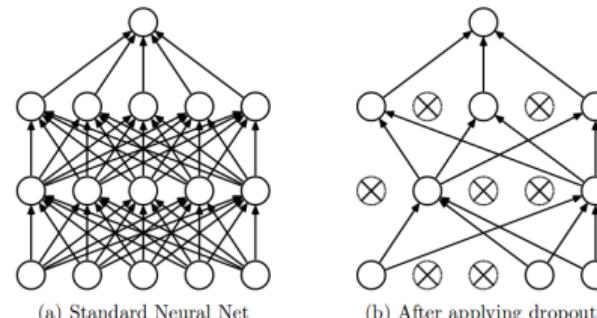
Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

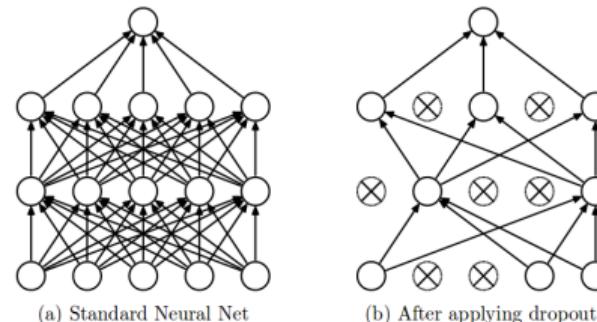
Kết luận

- Dropout giải quyết cả 2 vấn đề: Overfitting và tối ưu việc lấy trung bình nhiều bộ tham số.
- Bỏ bớt các nodes trong các layer.
- Các kết nối từ trước và sau node sẽ bị loại bỏ tạm thời
- Tạo nên cấu trúc MLP mới từ MLP gốc
- Các node bị loại bỏ dựa trên tham số xác suất p.



# Tầng Dropout

- Áp dụng Dropout cho MLP tương đương lấy mẫu một MLP thưa từ MLP gốc.
- MLP thưa gồm các node chưa bị loại bỏ ở quá trình dropout
- Với MLP gồm  $n$  nodes, có  $2^n$  MLP thưa
- Tất cả các phiên bản đều vẫn có chung trọng số, nên số lượng tham số là  $O(n^2)$ .
- Quá trình huấn luyện MLP với Dropout
  - Lấy mẫu một MLP thưa
  - Huấn luyện trên phiên bản thưa đó



Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

# Tầng Dropout

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



- Khi mô hình bị Overfitting, mô hình đã **lỡ học thông tin gây nhiễu**.
- Bản chất quá trình huấn luyện sử dụng tất cả các nodes, giảm hàm mất mát.
- Khi Overfitting xảy ra, một node có thể thay đổi để bù đắp sai sót của các node khác, dẫn đến hiện tượng **phụ thuộc phức tạp lẫn nhau**.
- Điều này khiến mô hình mất đi khả năng dự đoán trên dữ liệu mới.
- Bằng việc vô hiệu hóa ngẫu nhiên các nodes, sự hiện diện của từng neuron trở nên bớt tin cậy
- Điều này buộc các tầng phải **hoạt động độc lập**, phân bổ trách nhiệm xử lý đầu vào đều hơn.
- Tạo ra mô hình ít các phụ thuộc phức tạp hơn, **tổng quát hóa tốt hơn**.

# Tầng Batch Normalization

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Đại diện cho việc **chuẩn hóa các giá trị kích hoạt** của các node ở các tầng Hidden.
- Sao cho phân phối của các node là như nhau.
- Ví dụ với việc huấn luyện một MLP.
  - Nếu phân phối của tầng hiện tại thay đổi
  - Phân phối của giá trị kích hoạt ở tầng tiếp cũng thay đổi (Do đầu ra của tầng hiện tại là đầu vào của tầng sau)
- Hiện tượng thay đổi phân bố giá trị kích hoạt được gọi là: **internal covariance shift**.



# Tầng Batch Normalization

- Xét hình ở bên trái, trước hết sẽ phải chuẩn hóa giá trị đầu vào  $[x_1, x_2, \dots, x_n]$
- Sau đó chuẩn hóa đầu ra của tầng Hidden 1 để làm đầu vào của tầng Hidden 2
- Nếu chia MLP gốc thành nhiều mạng con thì
  - Áp dụng Batch Normalization tức là chuẩn hóa giá trị kích hoạt  $[h_1, h_2, \dots, h_n]$  của tầng Hidden 1 làm đầu vào cho tầng Hidden 2.

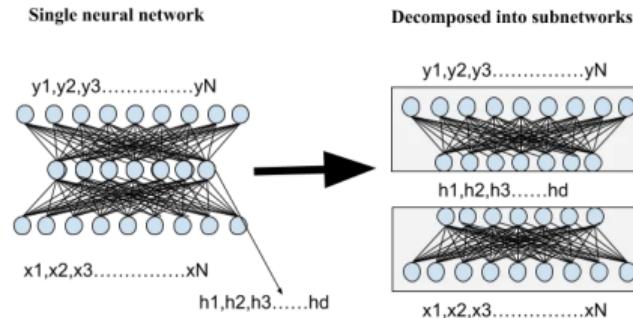


Figure: Minh họa chia MLP thành các mạng con

# Tầng Batch Normalization

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Chuẩn hóa một dữ liệu bất kì tức đưa  $x$  về có trung bình = 0, phương sai = 1.
- Với số lượng node trong một tầng Hidden là  $d$ , giá trị kích hoạt của tầng đó là  $x = [x_1, x_2, \dots, x_d]$  thì công thức chuẩn hóa là:

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}]}}$$

$\hat{x}^{(k)}$  là giá trị chuẩn hóa của node thứ  $k$  trong tầng Hidden đang xét.  
 $\mathbb{E}[x^{(k)}]$  là trung bình của node thứ  $k$ .  
 $\text{Var}[x^{(k)}]$  là phương sai của node thứ  $k$ .

# Tầng Batch Normalization

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Tuy nhiên, ta không muốn lúc nào các node đều phải có trung bình = 0, phương sai = 1.
- Ta muốn mô hình tự học các giá trị này.
- Xét hai tham số mới:  $\gamma, \beta$  để học các giá trị trung bình và phương sai.
- Các tham số này sẽ được học cùng lúc với trọng số và bias khi huấn luyện
- Công thức cuối cùng cho giá trị chuẩn hóa kích hoạt của node thứ k là:

$$y^k = \gamma^k \times \hat{x}^k + \beta^k$$



# Tầng Batch Normalization

- Tuy nhiên, để tối ưu quá trình huấn luyện, ta sẽ chỉ cập nhật tham số sau **từng phần nhỏ của dữ liệu**(mini-batch) chứ **không dùng toàn bộ dữ liệu**.
- Batch Normalization cũng áp dụng tương tự.

## Algorithm Quá trình huấn luyện với Batch Normalization và Mini-Batch $B$

**Đầu vào:** Giá trị Mini-Batch  $B = \{x_1, \dots, x_m\}$

**Đầu vào:** Tham số  $\gamma, \beta$

**Đầu ra:**  $\{y_i = BN_{\gamma, \beta}(x_i)\}$

1: Tính trung bình của Mini-Batch:  $\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$

2: Tính phương sai của Mini-Batch:  $\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$

3: **for**  $i = 1, \dots, m$  **do**

4:     Chuẩn hóa:  $\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$

5:     Điều chỉnh tỷ lệ và dịch chuyển:  $y_i \leftarrow \gamma \hat{x}_i + \beta$

6: **end for**

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

# Tầng Batch Normalization

- Tuy nhiên, khi dự đoán ta sẽ không áp dụng Batch Normalization như lúc huấn luyện. Vì ta chỉ xét một mẫu dữ liệu tại một lần dự đoán.
- Vì vậy, ta sẽ lưu lại trung bình và phương sai của node thứ k(lúc huấn luyện).
- Sử dụng các giá trị đó kết hợp với  $\gamma, \beta$  để dự đoán.

## Algorithm Quá trình dự đoán với Batch Normalization

```
1: for  $k = 1 \dots K$  do
2:   // Ký hiệu:  $x \equiv x^{(k)}, \gamma \equiv \gamma^{(k)}, \mu_B \equiv \mu_B^{(k)}, \dots$ 
3:   // Trung bình và phương sai trên các mini-batch:
4:    $\mathbb{E}[x] \leftarrow \mathbb{E}_B[\mu_B]$                                 // Kỳ vọng trung bình
5:    $\text{Var}[x] \leftarrow \frac{m}{m-1} \mathbb{E}_B[\sigma_B^2]$           // Phương sai
6:   // Phép biến đổi  $y$  trong  $N_{BN}^{inf}$ :
7:    $y \leftarrow \frac{\gamma}{\sqrt{\text{Var}[x]+\epsilon}} \cdot x + \left( \beta - \frac{\gamma \mathbb{E}[x]}{\sqrt{\text{Var}[x]+\epsilon}} \right)$ 
8: end for
```



Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

# Độ đo đánh giá

■ **Accuracy:** Tỷ lệ dự đoán đúng trên tổng số dự đoán.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

■ **Ma trận nhầm lẫn:** Là một kĩ thuật đánh giá hiệu năng của mô hình cho các bài toán phân lớp.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure: Ma trận nhầm lẫn trong phân loại nhị phân.

# Độ đo đánh giá

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Precision và Recall: Là 2 chỉ số quan trọng để đánh giá hiệu năng của mô hình.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) <b>Type II Error</b>	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) <b>Type I Error</b>	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	

Figure: Precision và Recall.

# Mô tả dữ liệu

- Dữ liệu sử dụng: FashionMNIST
- 60000 ảnh Training, 10000 ảnh Test.
- Mỗi ảnh là một ảnh xám, kích thước  $28 \times 28 \times 1$ .
- Có 10 nhãn.

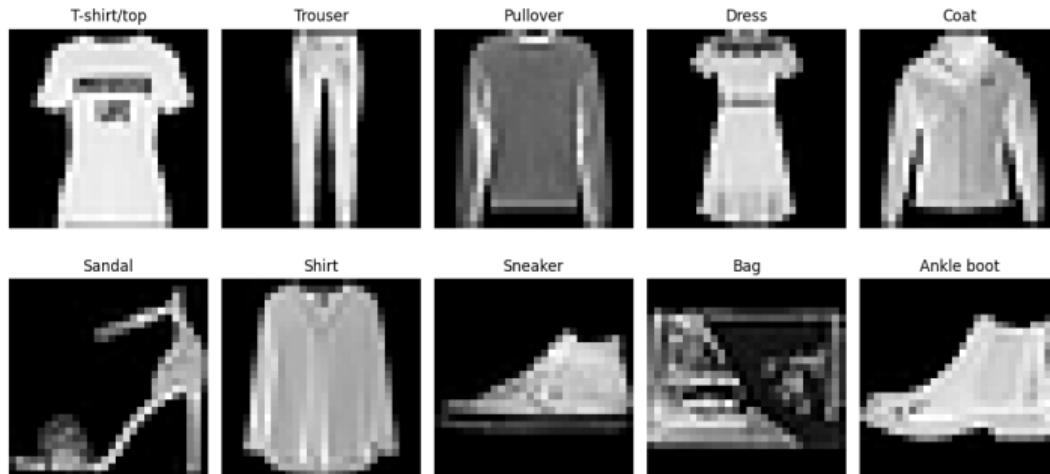


Figure: Ví dụ ảnh của từng nhãn

# Chia dữ liệu

Bộ dữ liệu này đã được chia thành 3 tập Training, Validation, Test với số lượng mẫu tương ứng là 48000, 12000, 10000.

Nhãn	Training	Validation	Test
T-shirt/top	4800	1200	1000
Trouser	4800	1200	1000
Pullover	4800	1200	1000
Dress	4800	1200	1000
Coat	4800	1200	1000
Sandal	4800	1200	1000
Shirt	4800	1200	1000
Sneaker	4800	1200	1000
Bag	4800	1200	1000
Ankle boot	4800	1200	1000

Table: Bảng thống kê dữ liệu FashionMNIST

# Chuẩn hóa và kiểm tra bộ dữ liệu

Chuẩn hóa dữ liệu từ 0-255 về 0.0-1.0 (Do các pixels có giá trị từ 0-255)

	<b>Maximum</b>	<b>Minimum</b>
<b>Trước chuẩn hóa</b>	255	0
<b>Sau chuẩn hóa</b>	1.0	0.0

Table: Giá trị trước và sau chuẩn hóa

Kiểm tra có ảnh nào có giá trị NaN(ảnh lỗi)

<b>Bộ dữ liệu</b>	<b>Ma trận dữ liệu(X)</b>	<b>Biến mục tiêu(y)</b>
<b>Train</b>	Không có giá trị NaN	Không có giá trị NaN
<b>Valid</b>	Không có giá trị NaN	Không có giá trị NaN
<b>Test</b>	Không có giá trị NaN	Không có giá trị NaN

Table: Kiểm tra giá trị NaN trong bộ dữ liệu

# Tham số cho t-SNE

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

Áp dụng giảm chiều trên tập Training(48000 mẫu) về 2 chiều và 3 chiều.  
Các bộ tham số sử dụng:

- *n\_components* = 2(hoặc 3).
- *learning\_rate* = 200.
- *init* = "random".
- *perplexity* = 30.
- *random\_state* = 42.
- *metric* = "euclidian"

Đầu tiên sẽ giảm chiều trên tập Training gốc.



# Trực quan hóa dữ liệu với t-SNE

Trực quan hóa dữ liệu gốc giảm xuống 2 chiều.

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

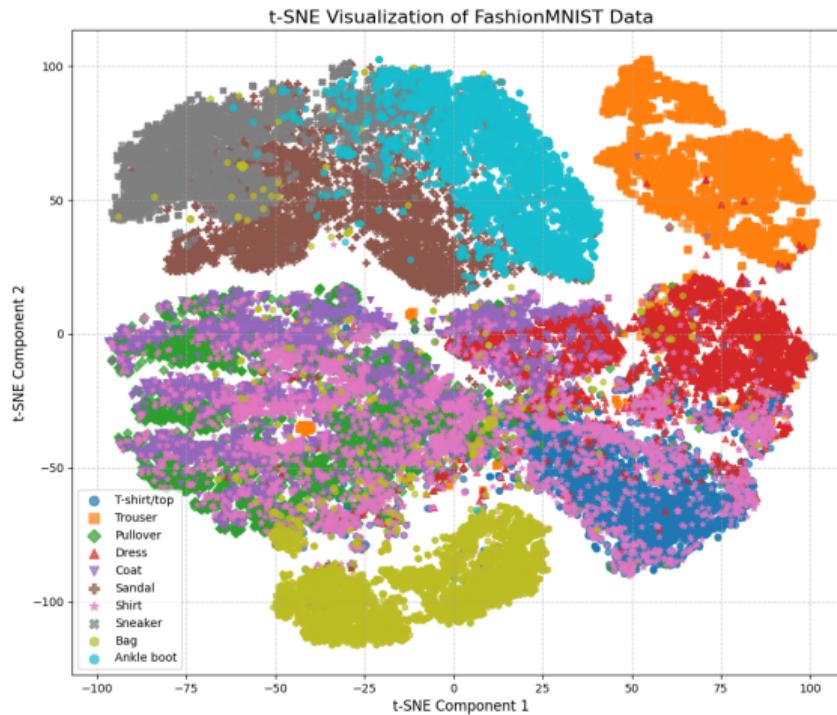
Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



# Trực quan hóa dữ liệu với t-SNE

Trực quan hóa dữ liệu gốc giảm xuống 3 chiều.

## Giới thiệu

## Cơ sở lý thuyết

- PCA
- TSNE
- Hàm Softmax
- Logistic Regression
- CNN
- Độ đo đánh giá

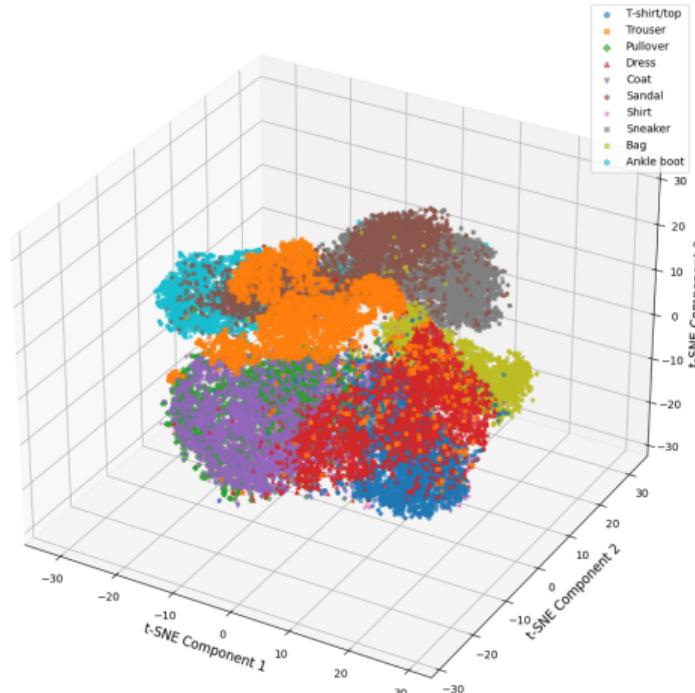
## Thực nghiệm và đánh giá

- Dữ liệu
- Thực nghiệm t-SNE
- Thực nghiệm PCA
- Thực nghiệm Logistic Regression
- Thực nghiệm CNN

## Kết luận

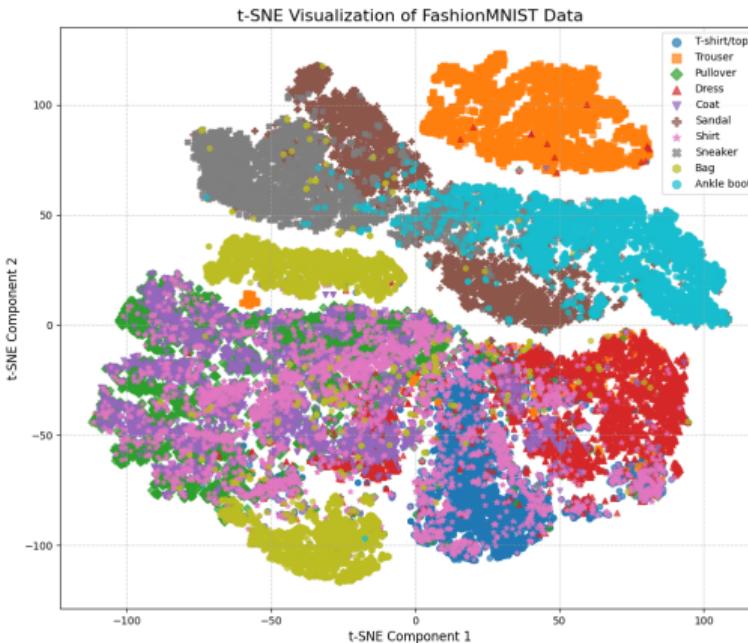


3D t-SNE Visualization of FashionMNIST Data



# Trực quan hóa dữ liệu với t-SNE

Vì dữ liệu ban đầu có **784 cột**, chúng tôi sẽ áp dụng **PCA giảm về 50 chiều**, sau đó áp dụng t-SNE với **bộ tham số như trước**. Trước hết là giảm về **2 chiều**



# Trực quan hóa dữ liệu với t-SNE

Trực quan hóa dữ liệu đã sử dụng PCA còn 50 thành phần chính, giảm xuống 3 chiều.

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

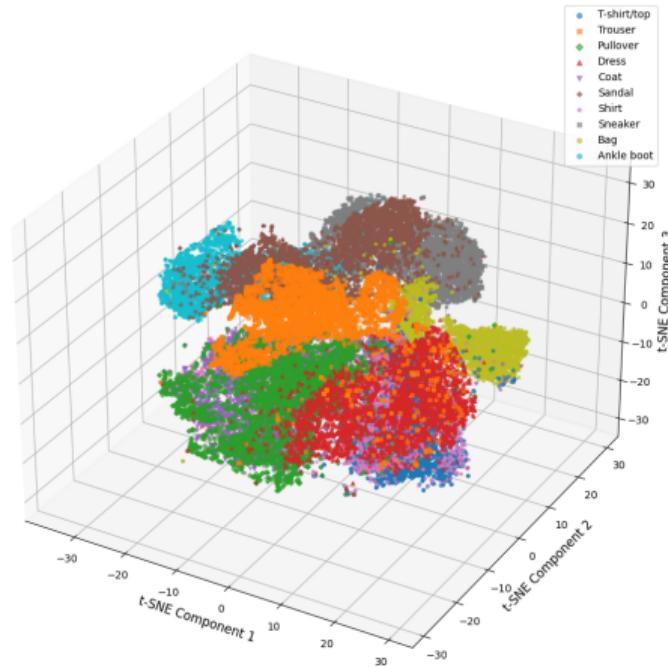
Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



3D t-SNE Visualization of FashionMNIST Data



# So sánh t-SNE: Dữ liệu gốc vs PCA (2 chiều)

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

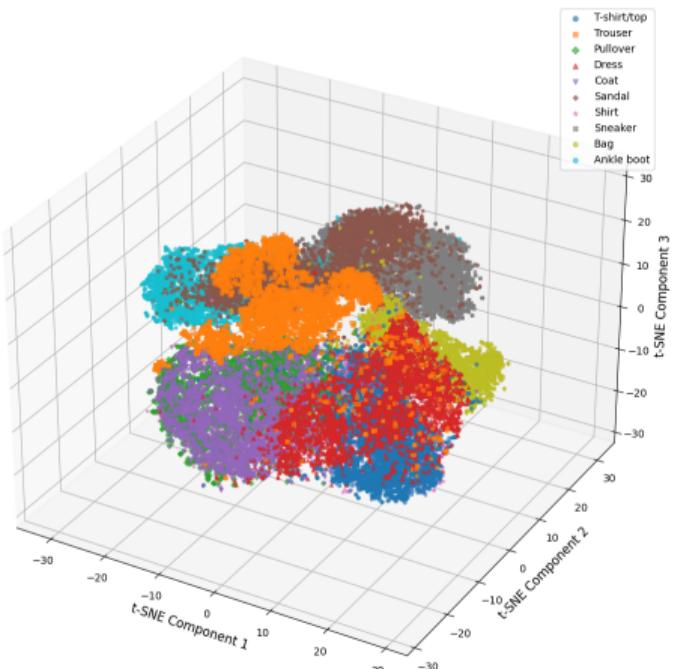
Kết luận



Figure: Trái: Dữ liệu gốc. Phải: PCA giảm về 50 chiều trước t-SNE.

# So sánh t-SNE: Dữ liệu gốc vs PCA (3 chiều)

3D t-SNE Visualization of FashionMNIST Data



3D t-SNE Visualization of FashionMNIST Data

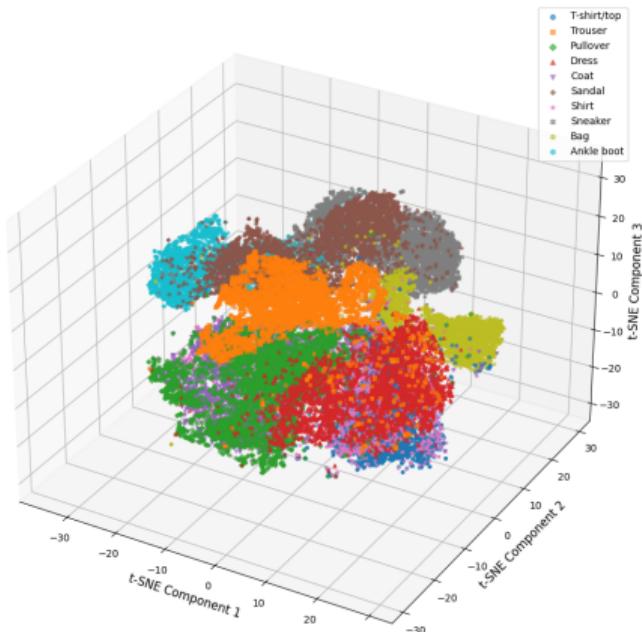


Figure: Trái: Dữ liệu gốc. Phải: PCA giảm về 50 chiều trước t-SNE.

# Thực Nghiệm PCA

- Phương sai giải thích cao ở đoạn đầu và có "đuôi dài" về phía sau.
- Phương sai tích lũy tăng nhanh ở 100 thành phần chính đầu tiên và chậm dần về sau, tạo ra điểm khuỷu tay.

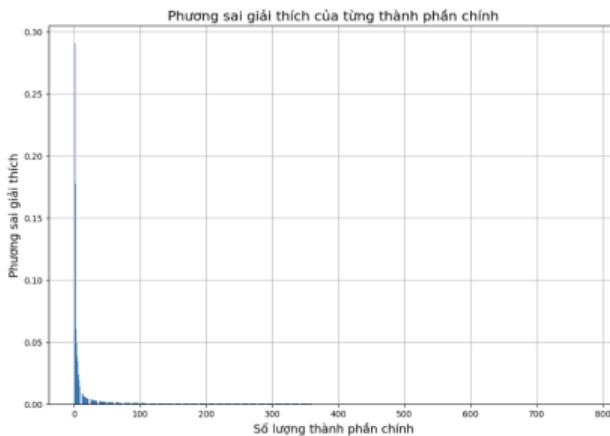


Figure: Phương sai giải thích.

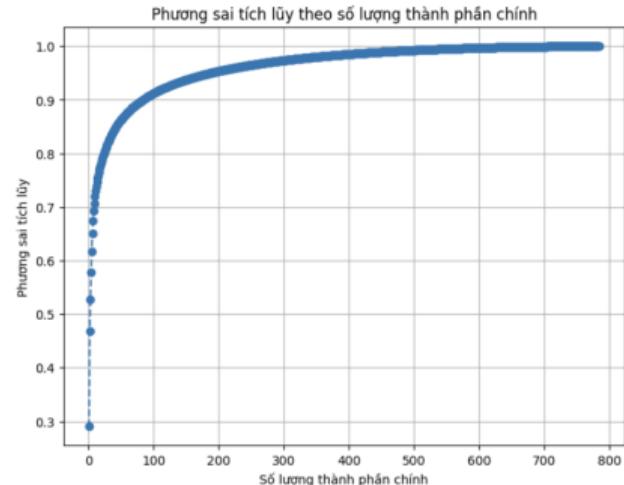


Figure: Phương sai tích lũy.

# Ảnh sau khi giảm chiều bằng PCA

## Giới thiệu

## Cơ sở lý thuyết

PCA

T-SNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

## Kết luận

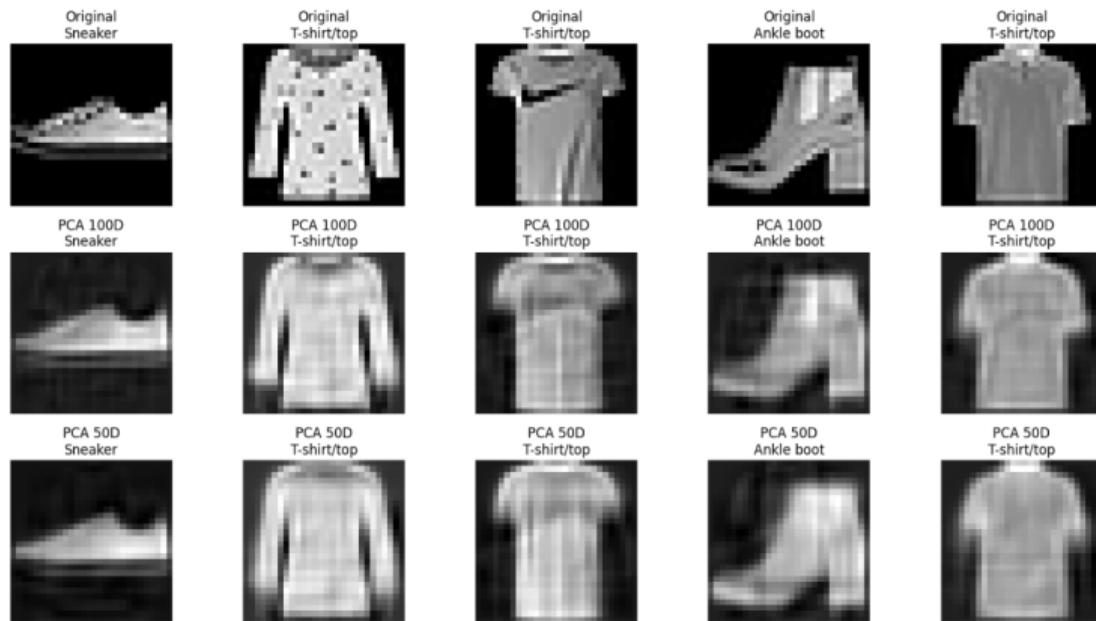


Figure: 5 ảnh ngẫu nhiên khi dùng PCA giảm về 50 và 100 chiều..

# Thực Nghiệm PCA

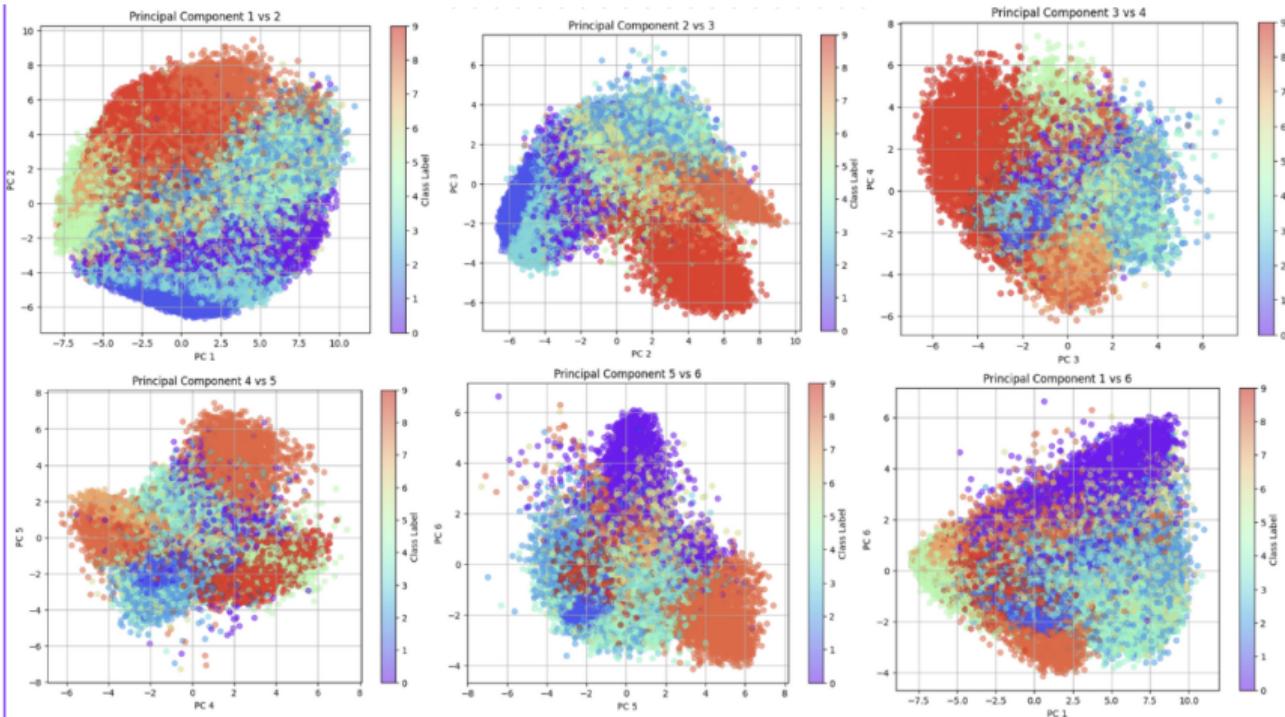


Figure: 6 cặp thành phần chính đầu tiên khi trực quan hóa

# Trực quan hóa 2D và 3D

## Giới thiệu

## Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

## Kết luận

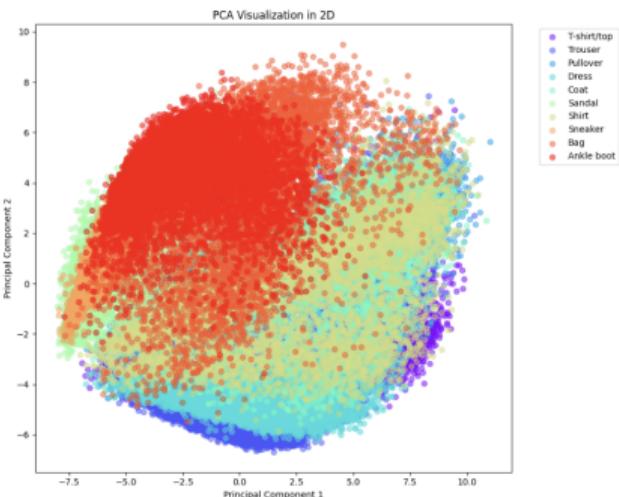


Figure: Trực quan hóa 2d.

PCA Visualization in 3D

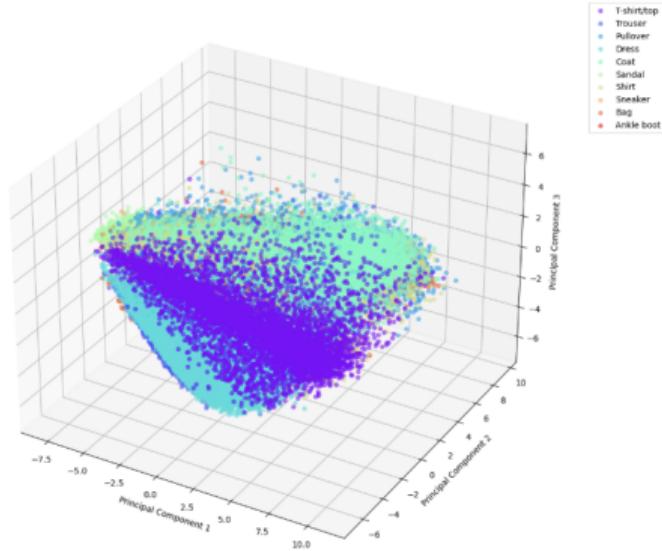


Figure: Trực quan hóa 3d.

# Thực nghiệm bằng Softmax Regression

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



Phương pháp Softmax Regression được tiến hành với:

- Bộ dữ liệu chưa giảm chiều và đã giảm về 46 chiều bằng PCA.
- Mô hình Logistic Regression của thư viện scikit-learn với các tham số chung: `multi_class = 'multinomial'`, `max_iter = 10000`.
- Tiến hành thực nghiệm trên cả 2 loại dữ liệu với các tỷ lệ `train:validation = 4:1, 7:3, 6:4`; các solver: `'lbfgs'`, `'saga'`, `'sag'`.
- Các chỉ số đánh giá bao gồm: Accuracy, Precision, Recall trên cả hai tập.

# Với tập dữ liệu chưa giảm chiều

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

STT	Train/Validation	Bộ tham số	Accuracy	Precision	Recall
1	4:1	solver='lbfgs'	0.889 / 0.850	0.888 / 0.850	0.889 / 0.850
2	4:1	solver='saga'	0.885 / 0.859	0.884 / 0.858	0.885 / 0.859
3	4:1	solver='sag'	0.887 / 0.858	0.886 / 0.857	0.887 / 0.858
4	7:3	solver='lbfgs'	0.892 / 0.840	0.891 / 0.838	0.892 / 0.840
5	7:3	solver='saga'	0.887 / 0.853	0.886 / 0.852	0.887 / 0.853
6	7:3	solver='sag'	0.889 / 0.851	0.888 / 0.850	0.889 / 0.851
7	6:4	solver='lbfgs'	0.898 / 0.836	0.897 / 0.834	0.898 / 0.836
8	6:4	solver='saga'	0.892 / 0.848	0.891 / 0.847	0.892 / 0.848
9	6:4	solver='sag'	0.893 / 0.845	0.893 / 0.844	0.893 / 0.845

Table: Thực nghiệm với các bộ dữ liệu và không dùng hiệu chỉnh L2.

# Với tập dữ liệu chưa giảm chiều

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

STT	Train/Validation	Bộ tham số	Accuracy	Precision	Recall
1	4:1	solver='lbfgs'	0.889 / 0.850	0.888 / 0.850	0.889 / 0.850
2	4:1	solver='saga'	0.885 / 0.859	0.884 / 0.858	0.885 / 0.859
3	4:1	solver='sag'	0.887 / 0.858	0.886 / 0.857	0.887 / 0.858
4	7:3	solver='lbfgs'	0.892 / 0.840	0.891 / 0.838	0.892 / 0.840
5	7:3	solver='saga'	0.887 / 0.853	0.886 / 0.852	0.887 / 0.853
6	7:3	solver='sag'	0.889 / 0.851	0.888 / 0.850	0.889 / 0.851
7	6:4	solver='lbfgs'	0.898 / 0.836	0.897 / 0.834	0.898 / 0.836
8	6:4	solver='saga'	0.892 / 0.848	0.891 / 0.847	0.892 / 0.848
9	6:4	solver='sag'	0.893 / 0.845	0.893 / 0.844	0.893 / 0.845

Table: Thực nghiệm với các bộ dữ liệu và không dùng hiệu chỉnh L2.

- Áp dụng hiệu chỉnh L2 với các mô hình 4, 7, 8 và 9.

# Tìm tham số C phù hợp

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Trong scikit-learn,  $C = \frac{1}{\lambda}$  với  $\lambda$  là hệ số hiệu chỉnh L2.
- Xác định tham số C thông qua lớp `model_selection.GridSearchCV`
- Mỗi solver được thử nghiệm với  $C \in \{10^{-8}, 10^{-7}, 10^{-6}, \dots, 100\}$ .
- Đánh giá chéo với tham số  $cv = 3$  trên bộ dữ liệu tương ứng của từng mô hình.



# Tìm tham số C phù hợp

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

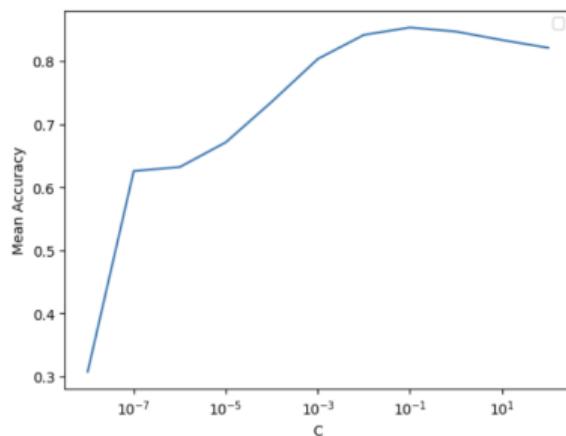
Thực nghiệm PCA

Thực nghiệm Logistic Regression

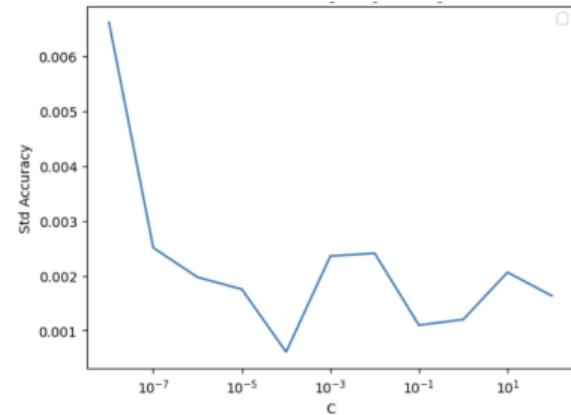
Thực nghiệm CNN

Kết luận

■ Với solver = 'lbfgs':



(a) Hệ số C và chỉ số mean accuracy (lbfgs).



(b) Hệ số C và chỉ số std accuracy (lbfgs).

Figure: So sánh mối quan hệ giữa hệ số C với mean accuracy và std accuracy.

# Tìm tham số C phù hợp

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

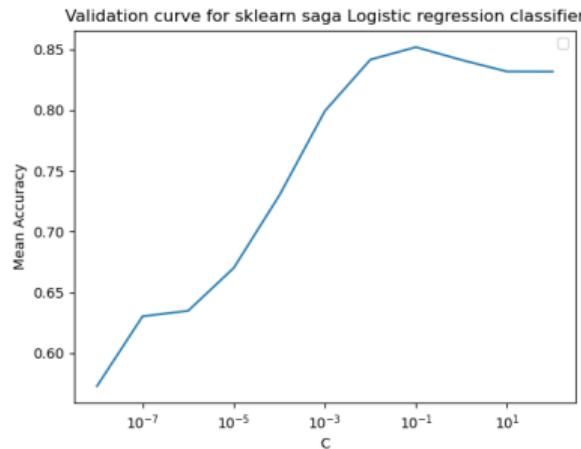
Thực nghiệm PCA

Thực nghiệm Logistic Regression

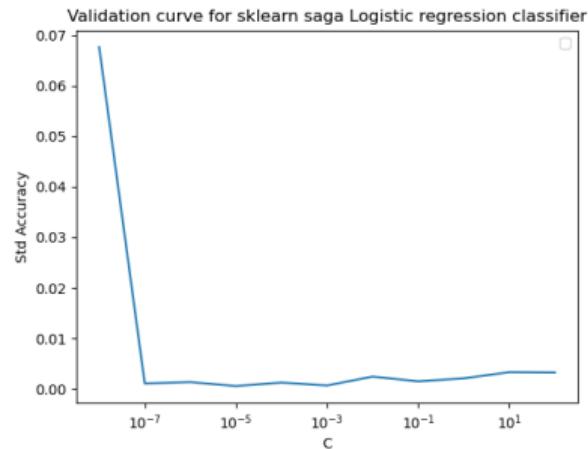
Thực nghiệm CNN

Kết luận

■ Với solver = 'saga':



(a) Hệ số C và chỉ số mean accuracy (saga).



(b) Hệ số C và chỉ số std accuracy (saga).

**Figure:** So sánh mối quan hệ giữa hệ số C với mean accuracy và std accuracy.

# Tìm tham số C phù hợp

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

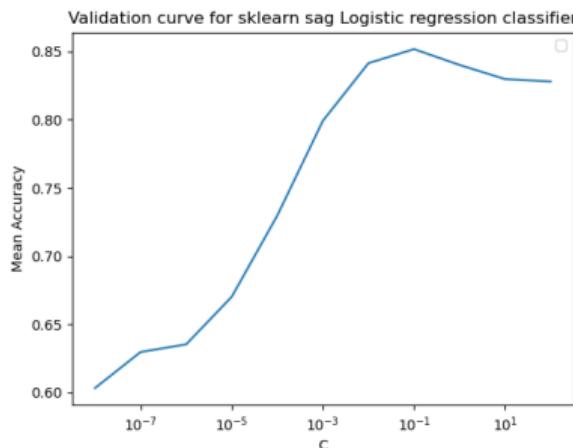
Thực nghiệm PCA

Thực nghiệm Logistic Regression

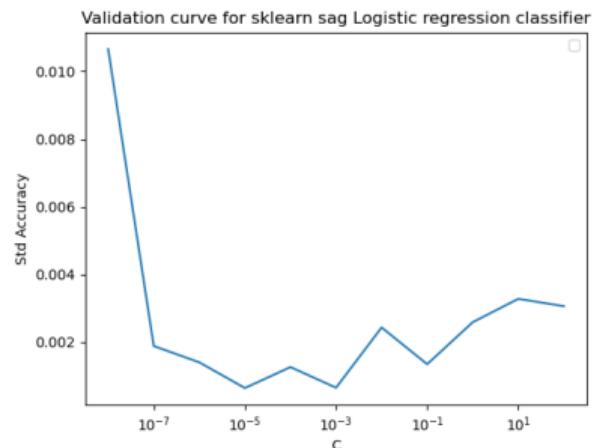
Thực nghiệm CNN

Kết luận

■ Với solver = 'sag':



(a) Hệ số C và chỉ số mean accuracy (sag).



(b) Hệ số C và chỉ số std accuracy (sag).

Figure: So sánh mối quan hệ giữa hệ số C với mean accuracy và std accuracy.

# Áp dụng hiệu chỉnh L2 với bộ dữ liệu chưa giảm chiều

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

STT	train/validation	Bộ tham số	Accuracy	Precision	Recall
1	4:1	solver='lbfgs'	0.888 / 0.850	0.887 / 0.848	0.888 / 0.850
2	4:1	solver='saga'	0.885 / 0.859	0.884 / 0.858	0.885 / 0.859
3	4:1	solver='sag'	0.887 / 0.858	0.886 / 0.857	0.887 / 0.858
4	7:3	solver='lbfgs', C=0.1	0.873 / 0.861	0.872 / 0.860	0.873 / 0.861
5	7:3	solver='saga'	0.887 / 0.847	0.886 / 0.846	0.887 / 0.847
6	7:3	solver='sag'	0.889 / 0.851	0.888 / 0.850	0.889 / 0.851
7	6:4	solver='lbfgs', C=0.1	0.874 / 0.860	0.873 / 0.860	0.874 / 0.860
8	6:4	solver='saga', C=0.1	0.874 / 0.861	0.873 / 0.860	0.874 / 0.861
9	6:4	solver='sag', C=0.1	0.874 / 0.861	0.873 / 0.860	0.874 / 0.861

Table: Thực nghiệm với các bộ dữ liệu sau khi dùng hiệu chỉnh L2.

- Mô hình 4 đảm bảo: chỉ số trên hai tập đủ cao và chênh lệch nhau ít nhất.

⇒ Chọn mô hình 4 là mô hình phù hợp với dữ liệu.

# Với tập dữ liệu đã giảm chiều

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

STT	train/validation	Bộ tham số	Accuracy	Precision	Recall
1	4:1	solver='lbfgs'	0.841 / 0.846	0.839 / 0.845	0.841 / 0.846
2	4:1	solver='saga'	0.840 / 0.845	0.839 / 0.844	0.840 / 0.845
3	4:1	solver='sag'	0.840 / 0.846	0.839 / 0.845	0.840 / 0.846
4	7:3	solver='lbfgs'	0.841 / 0.843	0.839 / 0.842	0.841 / 0.843
5	7:3	solver='saga'	0.840 / 0.843	0.839 / 0.842	0.840 / 0.843
6	7:3	solver='sag'	0.841 / 0.843	0.839 / 0.842	0.841 / 0.843
7	6:4	solver='lbfgs'	0.840 / 0.843	0.839 / 0.842	0.840 / 0.843
8	6:4	solver='saga'	0.840 / 0.843	0.838 / 0.842	0.840 / 0.843
9	6:4	solver='sag'	0.840 / 0.843	0.893 / 0.842	0.840 / 0.843

Table: Thực nghiệm với các bộ dữ liệu và không dùng hiệu chỉnh L2.

- Các mô hình đều không bị overfit nhưng độ chính xác thấp hơn từ 3% đến 5% so với khi chưa giảm chiều.

# Lựa chọn mô hình và đánh giá

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận

- Chọn bộ dữ liệu có tỷ lệ train:validation = 7:3
- Các tham số phù hợp của mô hình: solver = 'lbfgs', penalty='l2', C = 0.1
- Tiến hành huấn luyện trên tập train và đánh giá mô hình trên tập test.

## ■ Kết quả huấn luyện:

- Accuracy: 84.5%
- Precision: 84.4%
- Recall: 84.5%.



# Ma trận nhầm lẫn

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

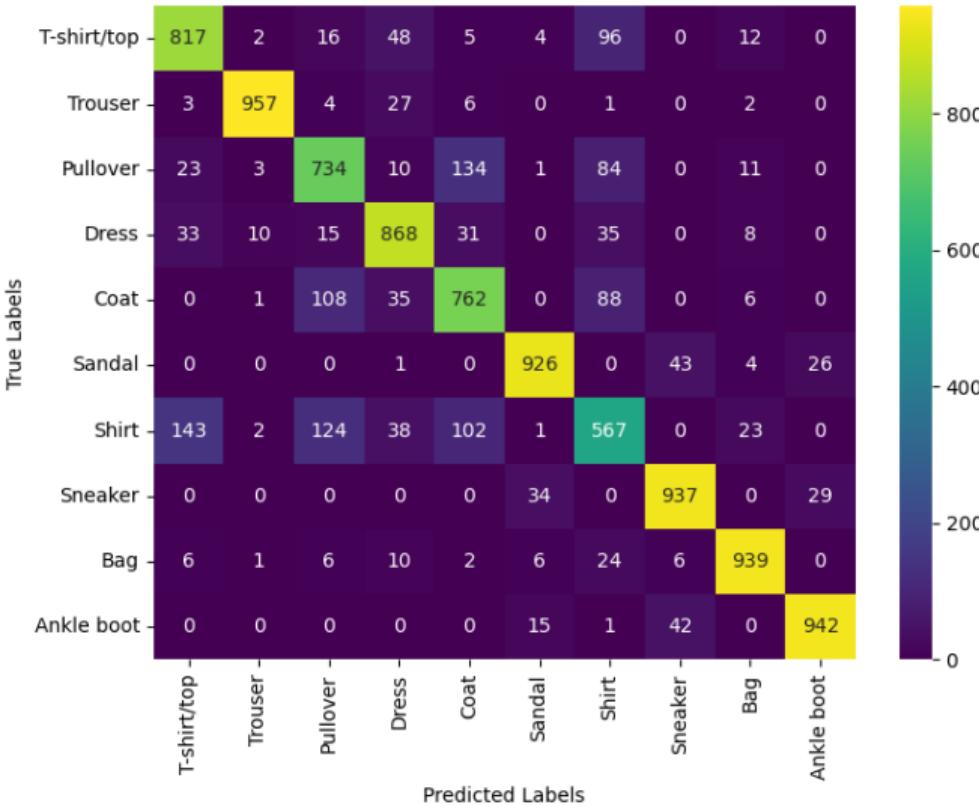
Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



Confusion Matrix (Softmax Regression)



Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



**Đầu vào:** Ảnh có kích thước  $28 \times 28 \times 1$ .

**Phần một:**

■ **Tầng Conv2D:**

- Số bộ lọc: 32
- Kích thước bộ lọc: (3, 3)
- Strides: 1
- Padding: "same"

■ **Tầng BatchNormalization**

- Tầng kích hoạt giá trị sử dụng hàm **ReLU**.
- Tầng **Dropout**:  $p = 0.25$ .

**Phần hai:**

■ Cấu trúc tương tự như phần một.

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Phần ba:

### ■ Tầng Conv2D:

- Số bộ lọc: 64
- Kích thước bộ lọc: (3, 3)
- Strides: 1
- Padding: "same"

### ■ Tầng BatchNormalization

### ■ Tầng kích hoạt giá trị sử dụng hàm ReLU.

### ■ Tầng MaxPooling:

- Kích thước bộ lọc: (2, 2)

### ■ Tầng Dropout: $p = 0.25$ .

# Cấu trúc CNN sử dụng

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Phần bốn:

### ■ Tầng Conv2D:

- Số bộ lọc: 128
- Kích thước bộ lọc: (3, 3)
- Strides: 1
- Padding: "same"

### ■ Tầng BatchNormalization

- Tầng kích hoạt giá trị sử dụng hàm **ReLU**.
- Tầng **Dropout**:  $p = 0.25$ .

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Phần cuối:

### ■ Tầng Flatten

### ■ Tầng Fully-connected:

#### ■ Hidden layer 1:

- Tầng Dense(512, activation="relu").
- Tầng BatchNormalization.
- Tầng Dropout:  $p = 0.5$ .

#### ■ Hidden layer 2:

- Tầng Dense(128, activation="relu").
- Tầng BatchNormalization.
- Tầng Dropout:  $p = 0.5$ .

#### ■ Output layer:

- Tầng Dense(10, activation="softmax"): Dự đoán 10 lớp đầu ra.

Giới thiệu

Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

Thực nghiệm và đánh  
giá

Dữ liệu

Thực nghiệm t-SNE

Thực nghiệm PCA

Thực nghiệm Logistic Regression

Thực nghiệm CNN

Kết luận



## Tham số cho huấn luyện

- Hàm mất mát: SparseCategoricalEntropy
- Hàm tối ưu: Adam với learning\_rate = 0.001.
- Áp dụng giảm learning\_rate với công thức:  $10^{-3} \times 0.9^{\text{learning\_rate}}$ .
- 40 epochs.
- Batch size = 128.

## Kết quả huấn luyện

- Accuracy = 93.83%.
- Precision = 93.83%.
- Recall = 93.83%.

# Ma trận Confusion

## Giới thiệu

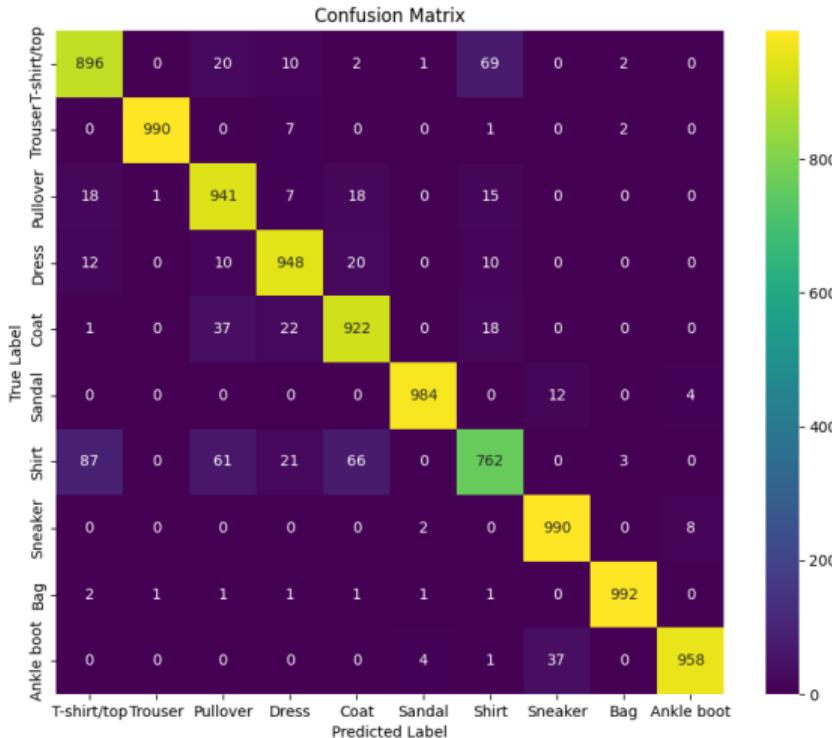
## Cơ sở lý thuyết

PCA  
TSNE  
Hàm Softmax  
Logistic Regression  
CNN  
Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu  
Thực nghiệm t-SNE  
Thực nghiệm PCA  
Thực nghiệm Logistic Regression  
Thực nghiệm CNN

## Kết luận



# Quá trình Training và Validation

## Giới thiệu

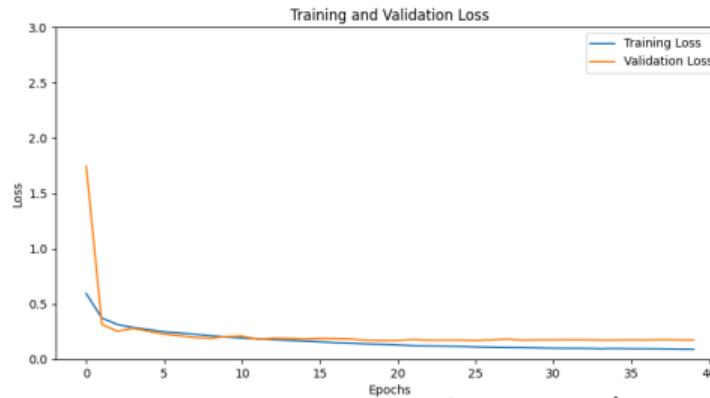
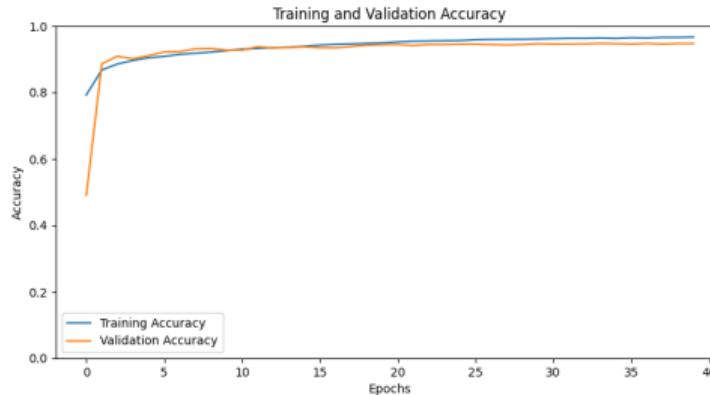
## Cơ sở lý thuyết

PCA  
TSNE  
Hàm Softmax  
Logistic Regression  
CNN  
Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu  
Thực nghiệm t-SNE  
Thực nghiệm PCA  
Thực nghiệm Logistic Regression  
Thực nghiệm CNN

## Kết luận



# So sánh PCA và t-SNE

## Giới thiệu

## Cơ sở lý thuyết

PCA

TSNE

Hàm Softmax

Logistic Regression

CNN

Độ đo đánh giá

## Thực nghiệm và đánh giá

Dữ liệu

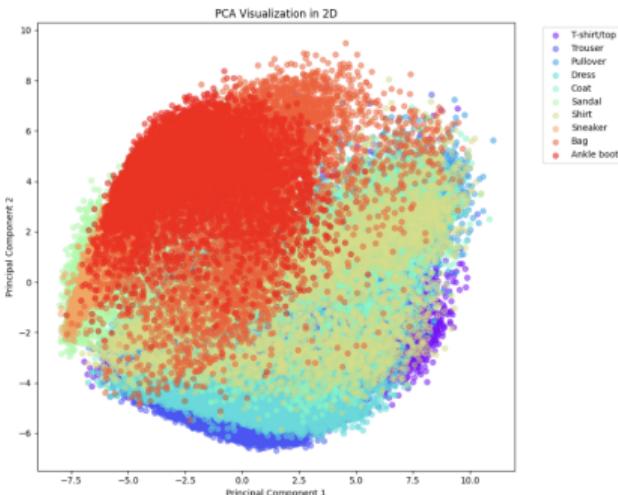
Thực nghiệm t-SNE

Thực nghiệm PCA

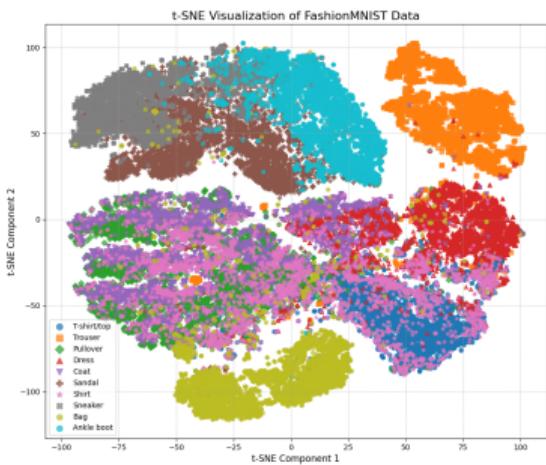
Thực nghiệm Logistic Regression

Thực nghiệm CNN

## Kết luận



(a) PCA 2D Trực quan hóa dữ liệu



(b) t-SNE 2D Trực quan hóa dữ liệu

# So sánh Softmax Regression và CNN

## Giới thiệu

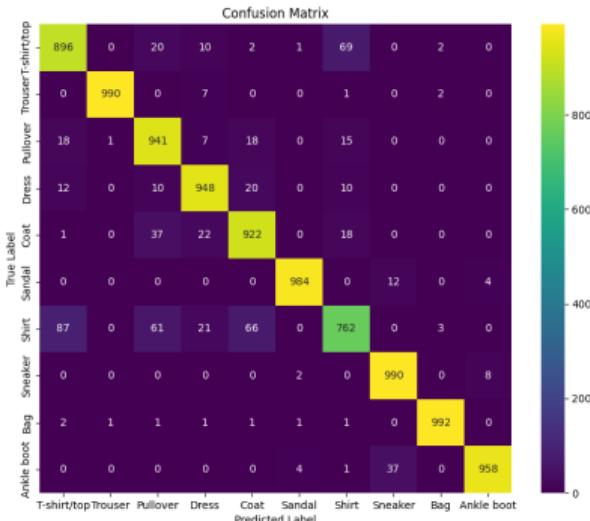
## Cơ sở lý thuyết

PCA  
TSNE  
Hàm Softmax  
Logistic Regression  
CNN  
Độ đo đánh giá

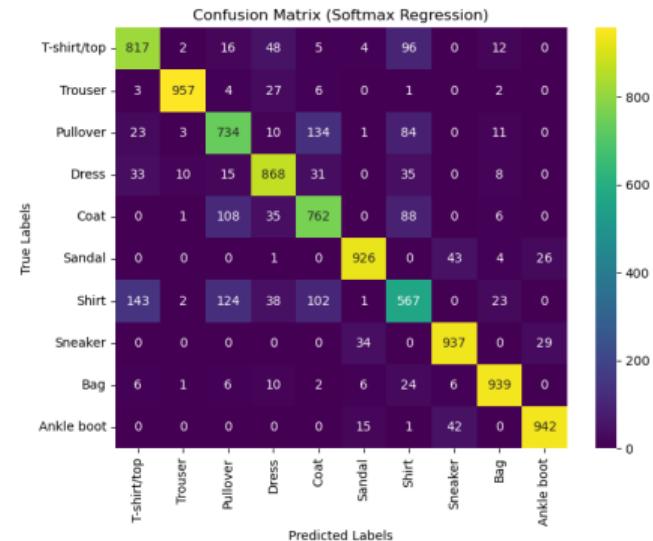
## Thực nghiệm và đánh giá

Dữ liệu  
Thực nghiệm t-SNE  
Thực nghiệm PCA  
Thực nghiệm Logistic Regression  
Thực nghiệm CNN

## Kết luận



(a) Confusion Matrix cho t-SNE



(b) Confusion Matrix cho Softmax Regression



Thank You

