

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**



TIỂU LUẬN CUỐI KỲ

**Đề tài: Nhận diện hình ảnh thời trang áp dụng
Multinomial Logistic Regression và Convolutional Neural Network**

Môn học: Học máy

Giảng viên: TS. Cao Văn Chung

NHÓM 05

Họ và tên: Nguyễn Chí Dũng – MSV: 20002040

Họ và tên: Đinh Tiến Dũng – MSV: 20002039

Họ và tên: Nguyễn Hoàng Giang – MSV: 20002048

Lớp: K65A5 Khoa học dữ liệu – Khoa: Toán – Cơ – Tin học

Hà Nội, 06/2023

LỜI NÓI ĐẦU

1. Tổng quan về đề tài

Trong tiểu luận này, chúng ta sẽ đi tìm hiểu những khái niệm toán học cần thiết để trực quan hóa bộ dữ liệu theo phương pháp phân tích thành phần chính - PCA (Principal Component Analysis), xây dựng hai mô hình học có giám sát trong nhận diện hình ảnh và cách thức hoạt động của chúng gồm mạng thần kinh tích chập - CNN (Convolutional Neural Network) và mô hình hồi quy logistic đa thức - Multinomial Logistic Regression cùng với đó là phương pháp phân cụm không giám sát K-Means Clustering. Sau đó, sử dụng bộ dữ liệu Fashion-MNIST huấn luyện các mô hình và đưa ra mô hình có khả năng phân loại hình ảnh thời trang tốt nhất.

2. Luận chứng lựa chọn đề tài

Phân loại hình ảnh là bài toán rất được quan tâm trong lĩnh vực trí tuệ nhân tạo. Để giải quyết bài toán này, nhiều mô hình khác nhau đã được phát triển, trong đó có hai mô hình thường được sử dụng là Convolutional Neural Network và Multinomial Logistic Regression.

Các mô hình này rất phù hợp cho bài toán phân loại hình ảnh vì:

- Mô hình Convolutional Neural Network có khả năng trích xuất đặc trưng của bức ảnh cực kỳ tốt và phù hợp cho bài toán phân loại hình ảnh với độ chính xác cao.
- Mô hình Multinomial Logistic Regression cũng là một mô hình phân loại phổ biến, sử dụng cho bài toán phân loại đa lớp.

Đề tài sử dụng bộ dữ liệu Fashion-MNIST để thực hiện quá trình huấn luyện và dự đoán. Fashion-MNIST là một trong những bộ dữ liệu tiêu chuẩn được áp dụng trong rất nhiều mô hình học máy với mục đích nhận diện hình ảnh. Với việc áp dụng đồng thời cả hai mô hình trên bộ dữ liệu Fashion-MNIST, chúng ta sẽ so sánh được độ hiệu quả và chọn được mô hình phù hợp trong bài toán nhận diện hình ảnh thời trang.

MỤC LỤC

Chương 1. Cơ sở lý thuyết.....	3
1.1 Multinomial Logistic Regression	3
1.1.1 Khái niệm	3
1.1.2 Hàm Softmax.....	3
1.1.3 Hàm tổn thất Cross Entropy	4
1.1.4 Thuật toán tối ưu Stochastic Gradient Descent	5
1.2 Convolutional Neural Network	5
1.2.1 Khái niệm	5
1.2.2 Convolution Layer.....	6
1.2.3 Pooling Layer	8
1.2.4 Fully Connected Layer	9
1.2.5 Hàm kích hoạt	9
1.3 K-Means Clustering.....	10
1.3.1 Khái niệm	10
1.3.2 Thuật toán K-Means.....	10
1.3.3 Chọn số cụm trong K-Means	11
1.3.4 Phân cụm bộ dữ liệu Fashion-MNIST	12
Chương 2. Trực quan hóa dữ liệu	14
2.1 Bộ dữ liệu Fashion-MNIST.....	14
2.2 Trực quan hóa bộ dữ liệu Fashion-MNIST.....	15
2.2.1 Trực quan hóa sử dụng matplotlib	15
2.2.2 Trực quan hóa sử dụng PCA	16
2.2.3 Trực quan hóa sử dụng t-SNE	18
Chương 3. Xây dựng và thực thi chương trình.....	19
3.1 Mô hình Multinomial Logistic Regression	19
3.2 Mô hình Convolutional Neural Network.....	21
Chương 4: Kết luận và phương hướng phát triển.....	23
TÀI LIỆU THAM KHẢO	24

Chương 1. Cơ sở lý thuyết

1.1 Multinomial Logistic Regression

1.1.1 Khái niệm

Hồi quy logistic đa thức hay còn gọi là hồi quy softmax là trường hợp tổng quát của hồi quy logistic để xử lý trường hợp có nhiều hơn hai nhãn. Trong hồi quy logistic, nhãn được giả thuyết ở dạng nhị phân $y^{(i)} \in \{0,1\}$. Trong khi đó, hồi quy softmax cho phép giải quyết trường hợp nhãn $y^{(i)} \in \{1,\dots,k\}$ với k là số nhãn.

1.1.2 Hàm Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

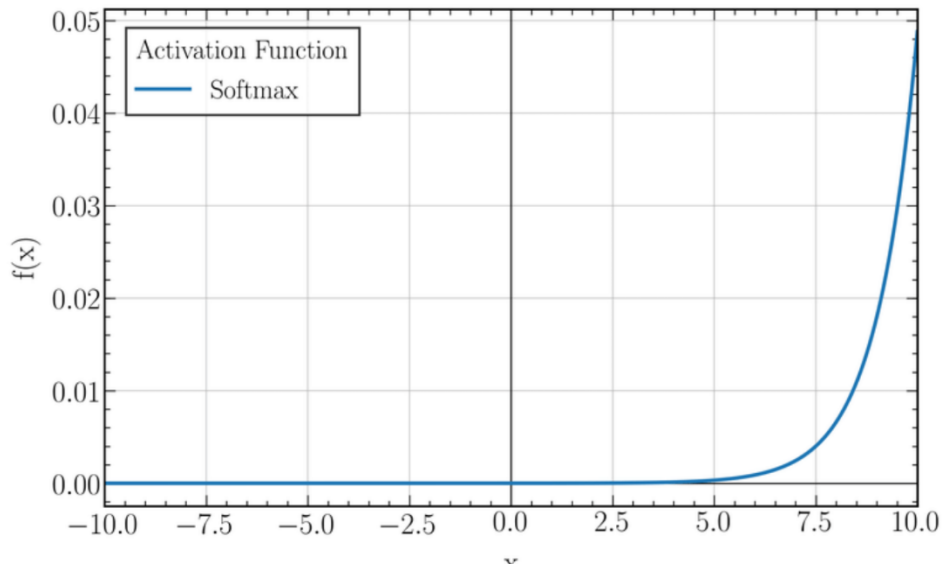
Trong đó:

\vec{z} : Giá trị vector nhập vào cho hàm softmax (Từ z_0 đến z_k)

z_i : Tất cả các giá trị z đều là giá trị vector nhập cho hàm softmax. Chúng có thể là bất cứ số thực nào, số dương, số âm hay số 0. Ví dụ, một mạng thần kinh nhân tạo có thể có giá trị vector ra là $(-0.62, 8.12, 2.53)$. Đây không phải là phân phối xác suất đúng. Đó là vì sao ta cần đến hàm softmax.

e^{z_i} : Hàm lũy thừa tiêu chuẩn được áp dụng cho mỗi giá trị nhập. Nó sẽ đưa ra một giá trị dương lớn hơn 0. Giá trị này sẽ rất nhỏ nếu giá trị nhập là âm, và rất lớn nếu giá trị nhập dương. Tuy nhiên, nó sẽ không cố định trong miền $(0,1]$. Đây là yêu cầu của một xác suất.

$\sum_{j=1}^K e^{z_j}$: Dòng phía dưới của công thức là một cụm chuẩn hóa. Nó đảm bảo rằng tổng của các giá trị ra sẽ luôn bằng 1 và nằm trong miền $(0,1]$. Như vậy, sẽ xuất hiện phân phối xác suất chính xác.



Đồ thị hàm Softmax

1.1.3 Hàm tổn thất Cross Entropy

Hàm Cross-Entropy (Negative Log Likelihood) được sử dụng để đo lường mức độ sai lệch giữa giá trị dự đoán của mô hình và giá trị thực tế. Đầu vào của hàm Cross-entropy là hai phân phối xác suất: Phân phối dự đoán của mô hình và phân phối thực tế của dữ liệu đích. Công thức của hàm Cross-entropy như sau:

$$J(W; x_i, y_i) = - \sum_{j=1}^c y_{ji} \log(a_{ji})$$

Với y_{ij} và a_{ij} lần lượt là phần tử thứ j của vector y_i và a_i .

1.1.4 Thuật toán tối ưu Stochastic Gradient Descent

Giải bài toán tối ưu $J(\theta) \rightarrow \min \theta$

Các bước thực hiện Stochastic Gradient Descent:

Bước 1: Tham số học $\eta > 0$; (x_n, y_n) . Khởi tạo: Chọn $\theta \in \mathbb{R}^{d \times c}$ bất kỳ ở bước $st = 0$

Bước 2: Tráo đổi ngẫu nhiên thứ tự các cặp dữ liệu (x_n, y_n) . Lần lượt với $n = 1, 2, \dots, N$, thực hiện:

Bước 3: Tính $\frac{\partial J_n(\theta)}{\partial \theta}$

Bước 4: Cập nhật tham số

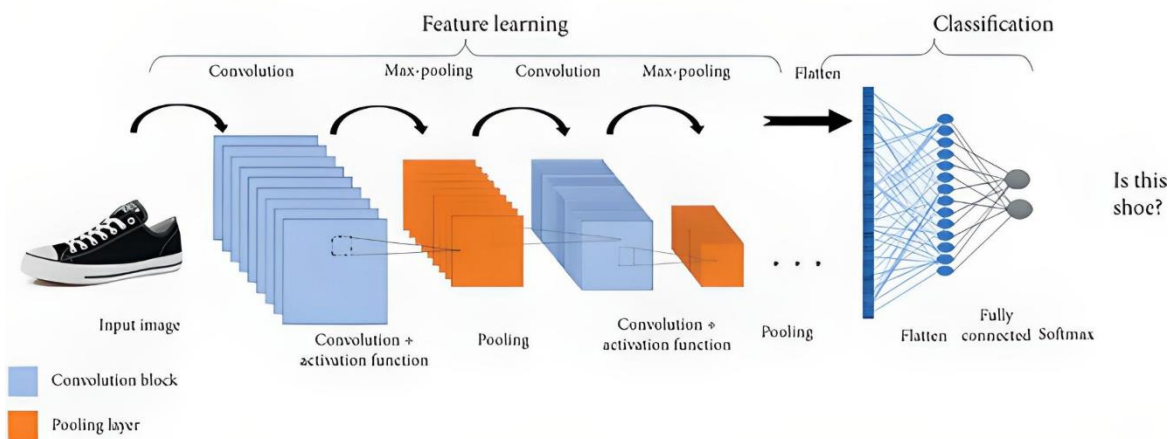
$$\theta = \theta - \eta \cdot \frac{\partial J_n(\theta)}{\partial \theta}$$

Bước 5: Nếu đạt điều kiện dừng, kết thúc. Ngược lại quay lại bước 2.

1.2 Convolutional Neural Network

1.2.1 Khái niệm

CNN (Convolutional Neural Network – Mạng nơ-ron tích chập) là một trong những mô hình học sâu tiên tiến. Nó giúp chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. CNN được sử dụng phổ biến trong các bài toán nhận dạng các đối tượng trong ảnh.



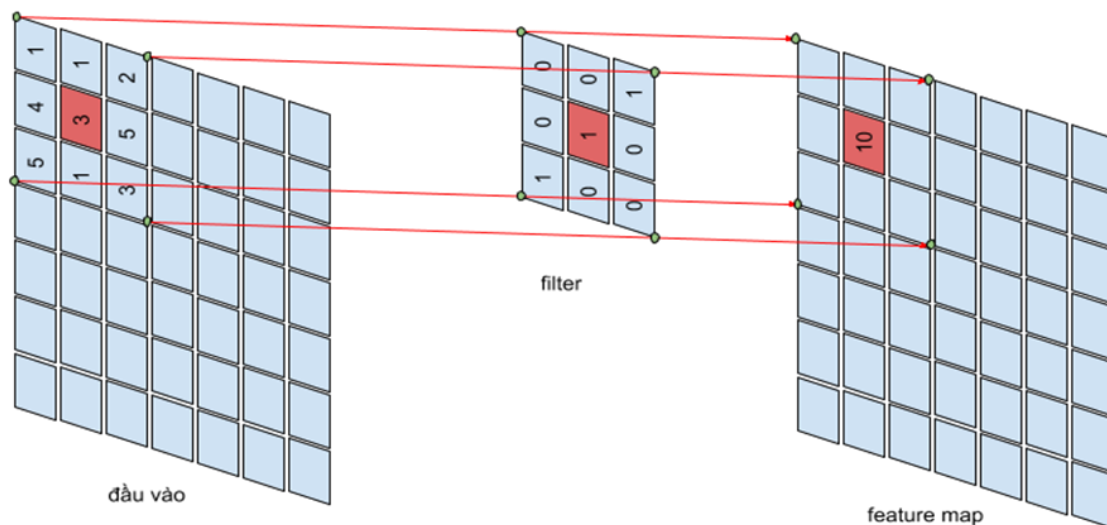
Mô hình hoạt động gồm các lớp cơ bản sau:

- Convolution layer
- Pooling layer
- Fully connected layer.

Các lớp này liên kết với nhau theo một thứ tự nhất định. Thông thường, một ảnh sẽ được lan truyền qua tầng convolution layer, sau đó các giá trị tính toán được sẽ lan truyền qua pooling layer, các lớp convolution layer, pooling layer có thể được lặp lại nhiều lần trong network. Và sau đó được lan truyền qua tầng fully connected layer và softmax để tính xác suất ảnh đó chứa vật thể gì.

1.2.2 Convolution Layer

Convolution layer là lớp quan trọng nhất và cũng là lớp đầu tiên của mô hình CNN. Lớp này có chức năng chính là phát hiện các đặc trưng của ảnh đầu vào. Trong tầng này có các đối tượng chính: Ma trận đầu vào, bộ filters và feature map.



Xét 1 ma trận 5x5 có giá trị pixel là 0 và 1. Ma trận bộ lọc 3x3 như hình dưới:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 – Image Matrix



1	0	1
0	1	0
1	0	1

3 x 3 – Filter Matrix

Sau đó, lớp tích chập của ma trận hình ảnh 5x5 nhân với ma trận bộ lọc 3x3 gọi là “Feature Map” như hình bên dưới.

1 _{x2}	1 _{x0}	1 _{x2}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x2}	0 _{x0}	1 _{x2}	1	1
0	0	1	1	0
0	1	1	0	0

Image

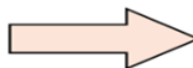
4		

Convolved Feature

Stride: Stride là sự dịch chuyển filter map theo pixel dựa trên giá trị từ trái sang phải. Khi stride là 1 thì ta di chuyển các kernel 1 pixel. Khi stride là 2 thì ta di chuyển các kernel đi 2 pixels và tiếp tục như vậy. Hình dưới là lớp tích chập hoạt động với stride là 2.

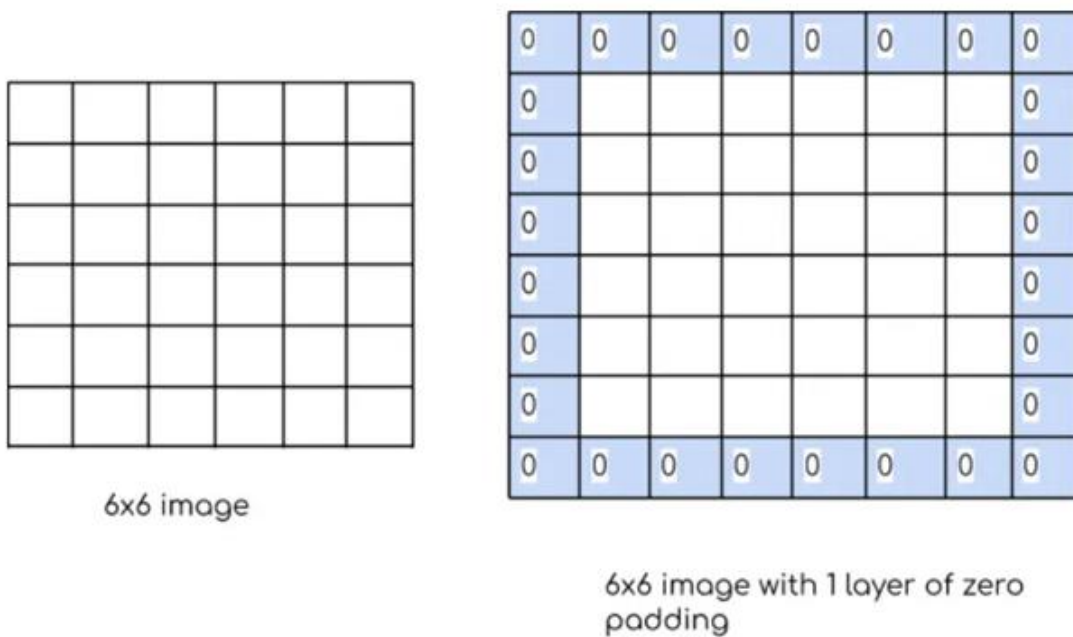
1	2	3	4	5	6	7
11	12	13	14	15	16	17
21	22	23	24	25	26	27
31	32	33	34	35	36	37
41	42	43	44	45	46	47
51	52	53	54	55	56	57
61	62	63	64	65	66	67
71	72	73	74	75	76	77

Convolve with 3x3 filters filled with ones



108	126	
288	306	

Padding: Là các giá trị 0 được thêm cùng lớp input.

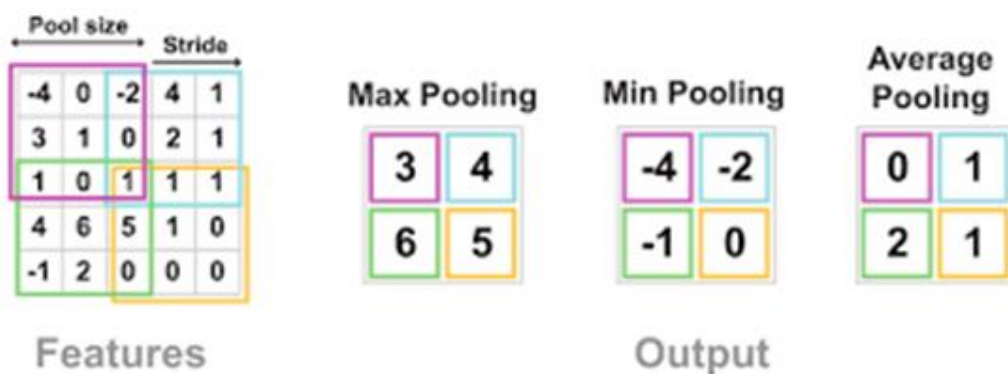


Đôi khi kernel không phù hợp với hình ảnh đầu vào. Ta có 2 lựa chọn:

- Chèn thêm các số 0 vào các đường biên của hình ảnh (padding).
- Cắt bớt hình ảnh tại những điểm không phù hợp với kernel.

1.2.3 Pooling Layer

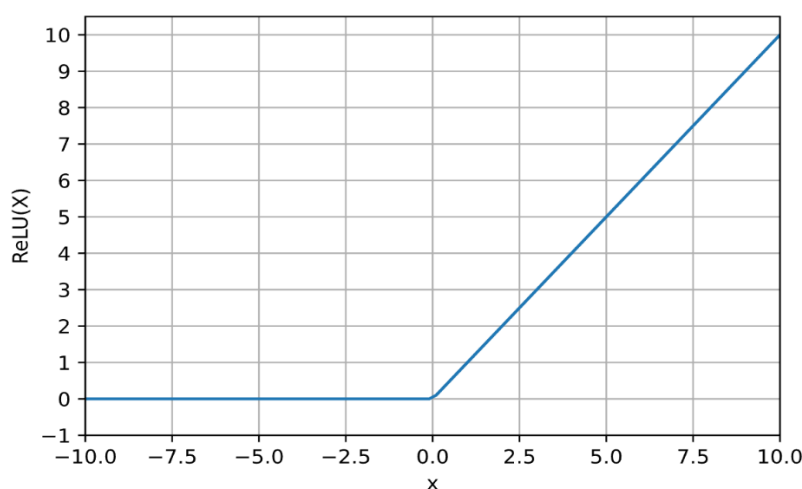
Sau hàm kích hoạt, thông thường chúng ta sử dụng tầng pooling. Một số loại pooling layer phổ biến như là max pooling, average pooling, với chức năng chính là giảm chiều của tầng trước đó. Với một pooling có kích thước 2x2, các bạn cần phải trượt filter 2x2 này trên những vùng ảnh có kích thước tương tự rồi sau đó tính max, hay average cho vùng ảnh đó.



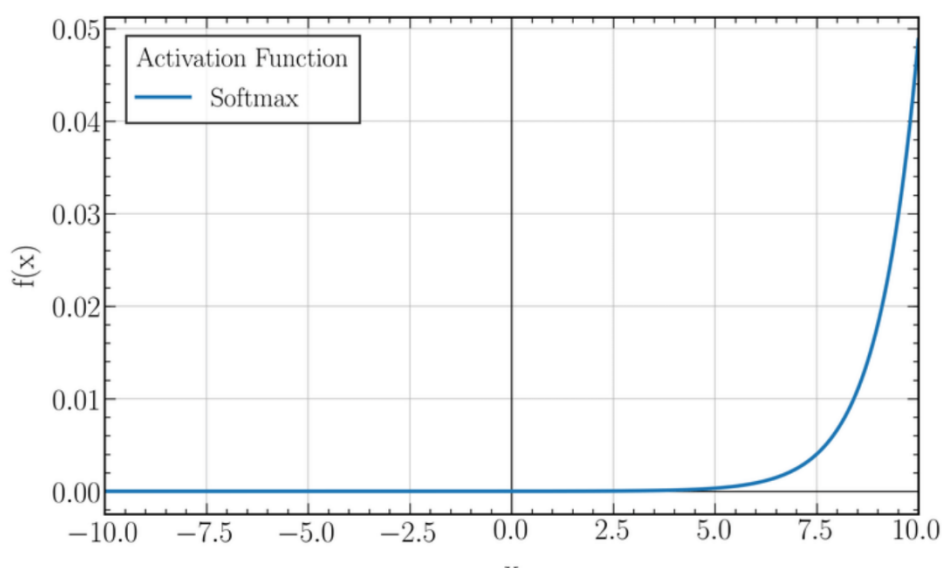
1.2.4 Fully Connected Layer

Tầng cuối cùng của mô hình CNN trong bài toán phân loại ảnh là tầng Fully Connected Layer. Tầng này có chức năng chuyển ma trận đặc trưng ở tầng trước thành vector chứa xác suất của các đối tượng cần được dự đoán. Trong bài toán phân loại hình ảnh thời trang Fashion-MNIST có 10 lớp tương ứng 10 số từ 0-1, tầng fully connected layer sẽ chuyển ma trận đặc trưng của tầng trước thành vector có 10 chiều thể hiện xác suất của 10 lớp tương ứng.

1.2.5 Hàm kích hoạt



$$\text{ReLU: } f(x) = \max(0, x)$$



$$\text{Softmax: } \sigma(\vec{Z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

1.3 K-Means Clustering

1.3.1 Khái niệm

K-Means là thuật toán phân cụm đơn giản thuộc loại học không giám sát (tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm. Ý tưởng của thuật toán phân cụm K-Means là phân chia một bộ dữ liệu thành các cụm khác nhau. Trong đó số lượng cụm được cho trước là k . Công việc phân cụm được xác lập dựa trên nguyên lý: Các điểm dữ liệu trong cùng một cụm phải có cùng một số tính chất nhất định. Tức là giữa các điểm trong cùng một cụm phải có sự liên quan lẫn nhau. Đối với máy tính thì các điểm trong một cụm đó sẽ là các điểm dữ liệu gần nhau.

1.3.2 Thuật toán K-Means

Cho tập dữ liệu $X = \{x_n\}_{n=1}^N$ và số cụm cần phân chia $0 < k < N$.

Khởi tạo: Lấy k điểm dữ liệu khác nhau c_1, \dots, c_k từ tập X - Các điểm trung tâm mỗi cụm.

Tại bước lặp thứ t :

+ Gán điểm dữ liệu vào cụm:

- Với k tâm cụm từ bước trước, với mỗi điểm dữ liệu x_n , tính các khoảng cách $d_{n,j} = \text{dist}(x_n, c_j)$ từ x_n đến tâm cụm hiện tại c_j .
- Giả sử l là chỉ số cụm có $d_{n,l} = \text{dist}(x_n, c_l)$ nhỏ nhất.
- Gán điểm x_n vào cụm thứ l , tức cụm có tâm hiện tại là c_l .
- Sau thao tác gán, dữ liệu X được chia vào các cụm S_1, S_2, \dots, S_k .
- Xác định lại tâm cụm: Tâm cụm c_j sẽ được xác định lại là trung bình cộng các điểm trong cụm S_j đó:

$$c_j^{\text{new}} := \frac{\sum_{x_n \in S_j} x_n}{|S_j|}$$

+ Thuật toán dừng khi tâm cụm tìm được ở bước $t + 1$ hoàn toàn trùng với tâm cụm tìm được ở bước t .

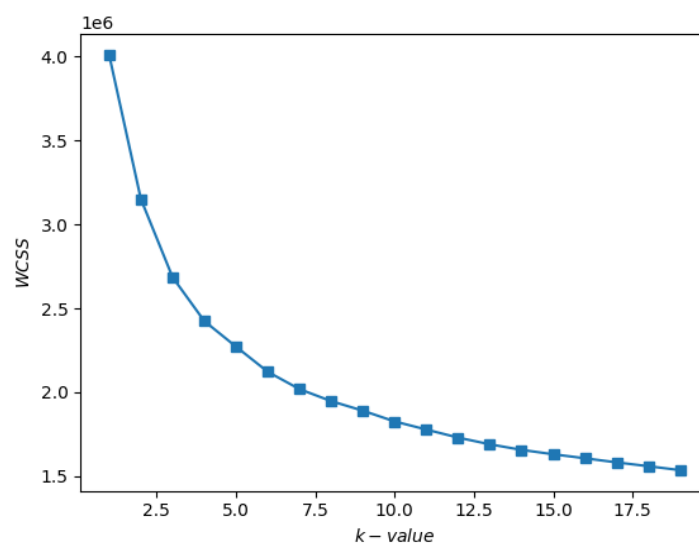
Một số lưu ý khi thực hiện thuật toán K-Means:

- Số cụm k phù hợp nhất: Tùy theo số điểm dữ liệu N. Số cụm k tối ưu thường được chọn bằng cách xét một dãy các giá trị của k và tìm giá trị phù hợp nhất.
- Chọn tâm cụm ở bước khởi tạo:
 - Không chọn các điểm khởi tạo quá gần nhau.
 - Chọn điểm c_1 bất kỳ
 - Chọn c_2 ở xa c_1 nhất có thể (theo khoảng cách Euclide).
 - Chọn c_3 sao cho $\text{dist}(c_1, c_3) + \text{dist}(c_2, c_3)$ lớn nhất có thể.
 - Tiếp tục như vậy cho đến tâm cụm chỉ số k.

1.3.3 Chọn số cụm trong K-Means

*Phương pháp Elbow:

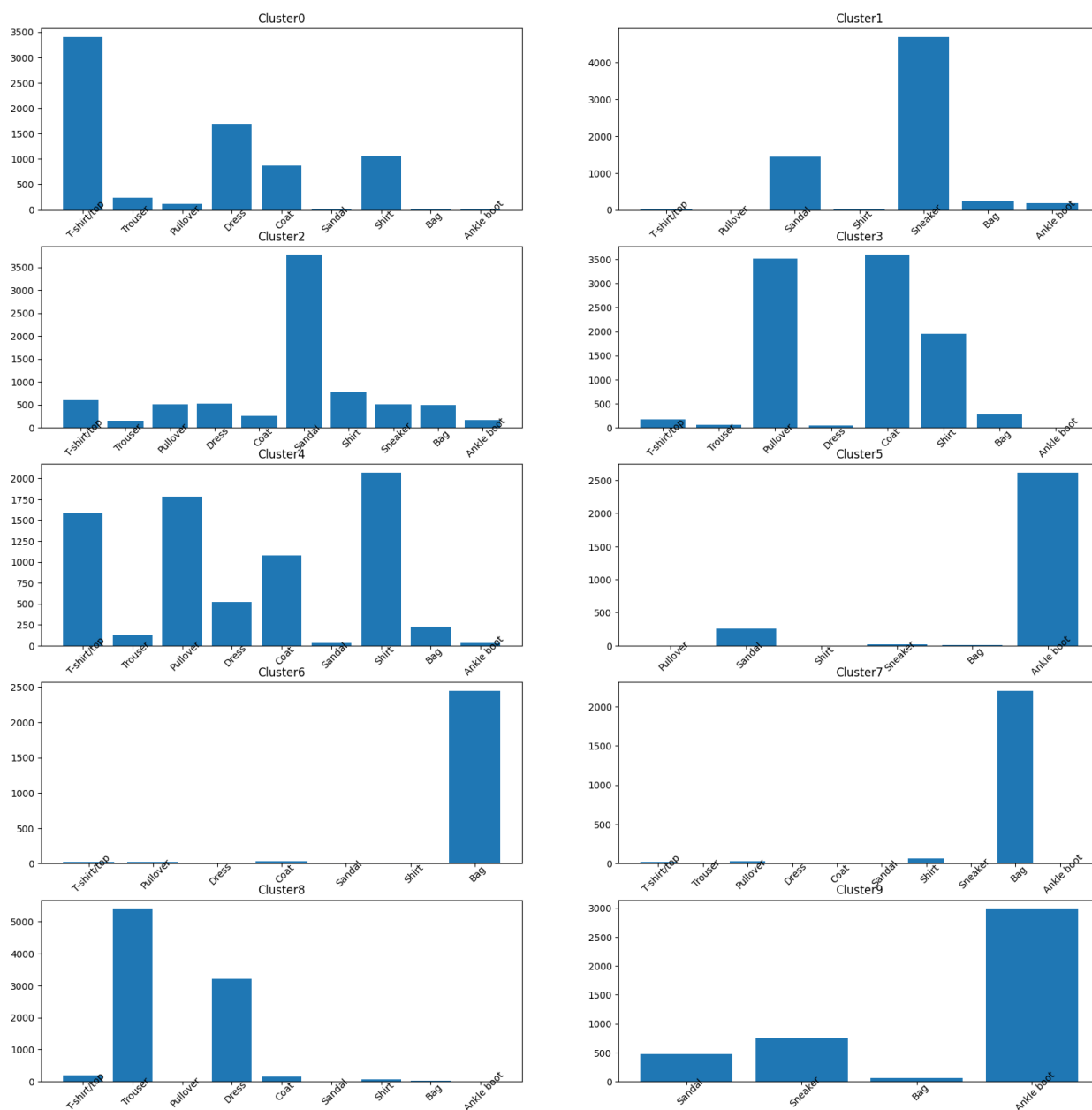
- Sử dụng thuật toán K-Means với các giá trị số lượng cụm khác nhau.
- Với mỗi số lượng cụm, tính tổng bình phương khoảng cách của các điểm đến tâm cụm tương ứng.
- Vẽ đường cong tổng bình phương khoảng cách (Within-Cluster Sum of Square - WCSS) với số lượng cụm trên trục x.
- Xác định số lượng cụm tối ưu bằng cách chọn điểm “khủy tay” của đường cong, tức là số lượng cụm mà sau đó việc tăng số lượng cụm không đem lại sự giảm thiểu đáng kể về tổng bình phương khoảng cách.



Đồ thị chọn số cụm cho bộ dữ liệu Fashion-MNIST

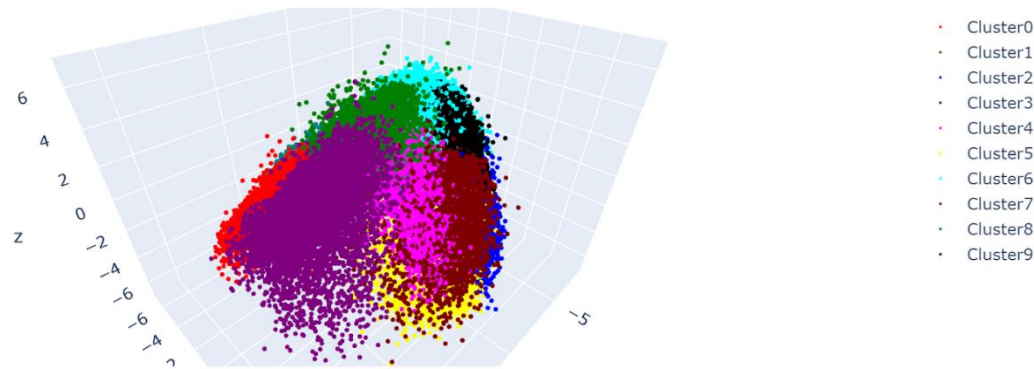
Nhận xét: Số cụm tối ưu nằm trong khoảng từ 7 đến 12.

1.3.4 Phân cụm bộ dữ liệu Fashion-MNIST

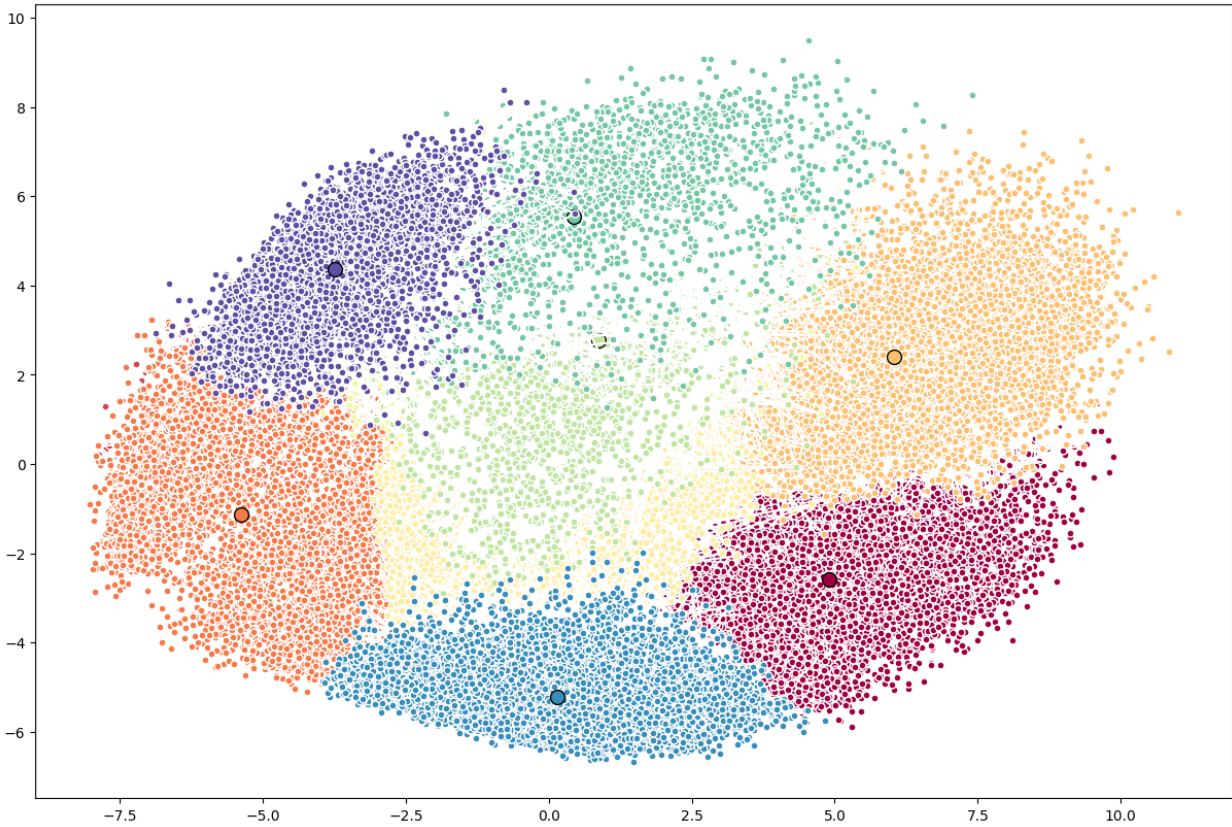


Đồ thị số lượng theo từng nhãn trong mỗi cụm

Cluster Visualisation



Đồ thị phân cụm 3D



Đồ thị phân cụm 2D

Chương 2. Trắc quan hóa dữ liệu

2.1 Bộ dữ liệu Fashion-MNIST

Fashion-MNIST (Fashion Modified National Institute of Standards and Technology) là bộ dữ liệu gồm các hình ảnh thời trang thuộc về công ty thương mại điện tử Zalando.

Zalando tạo ra bộ dữ liệu Fashion-MNIST nhằm đóng vai trò thay thế trực tiếp cho bộ dữ liệu MNIST ban đầu về chữ viết tay để đo điểm chuẩn cho các thuật toán học máy. Nó chia sẻ cùng kích thước hình ảnh và cấu trúc phân tách huấn luyện và kiểm tra.

Lý do ta dùng bộ dữ liệu Fashion-MNIST thay cho MNIST là do chúng ta muốn nhận biết những hình ảnh phức tạp hơn để đánh giá chương trình của ta khắt khe hơn.

Các hình ảnh có kích thước 28x28 pixels, với mỗi pixel mang một giá trị cho biết độ sáng hoặc độ tối, chỉ số càng cao thì càng tối. Giá trị pixel là một số nguyên từ 0 đến 255.

Bộ dữ liệu này bao gồm 785 cột, trong đó:

- Cột đầu tiên: Ứng với các nhãn (label) có giá trị từ 0-9.
- 784 cột còn lại: Chứa các giá trị pixel.

Các giá trị cột nhãn (label) tương ứng với các mặt hàng thời trang sau:

- 0: T-shirt/top (Áo thun/áo ba lỗ)
- 1: Trouser (Quần dài)
- 2: Pullover (Áo dài tay không có mũ trùm đầu)
- 3: Dress (Váy)
- 4: Coat (Áo choàng ngoài)
- 5: Sandal (Dép)
- 6: Shirt (Áo sơ mi)
- 7: Sneaker (Giày thể thao)
- 8: Bag (Túi)
- 9: Ankle boot (Giày cao cổ)

Số lượng hình ảnh tương ứng với mỗi nhãn thời trang trong bộ dữ liệu:

Pullover	:	6000	or	10.0%
Ankle Boot	:	6000	or	10.0%
Shirt	:	6000	or	10.0%
T-shirt/top	:	6000	or	10.0%
Dress	:	6000	or	10.0%
Coat	:	6000	or	10.0%
Sandal	:	6000	or	10.0%
Bag	:	6000	or	10.0%
Sneaker	:	6000	or	10.0%
Trouser	:	6000	or	10.0%

Với mỗi nhãn sẽ có 6000 hình ảnh cùng chiếm 10% trong bộ dữ liệu. Điều này cho chúng ta thấy được đây là một bộ dữ liệu cân bằng trong các nhãn.

2.2 Trực quan hóa bộ dữ liệu Fashion-MNIST

2.2.1 Trực quan hóa sử dụng matplotlib

Sử dụng thư viện matplotlib trực quan hóa 30 ảnh ngẫu nhiên trong bộ dữ liệu huấn luyện.



30 ảnh đầu tiên của bộ dữ liệu huấn luyện

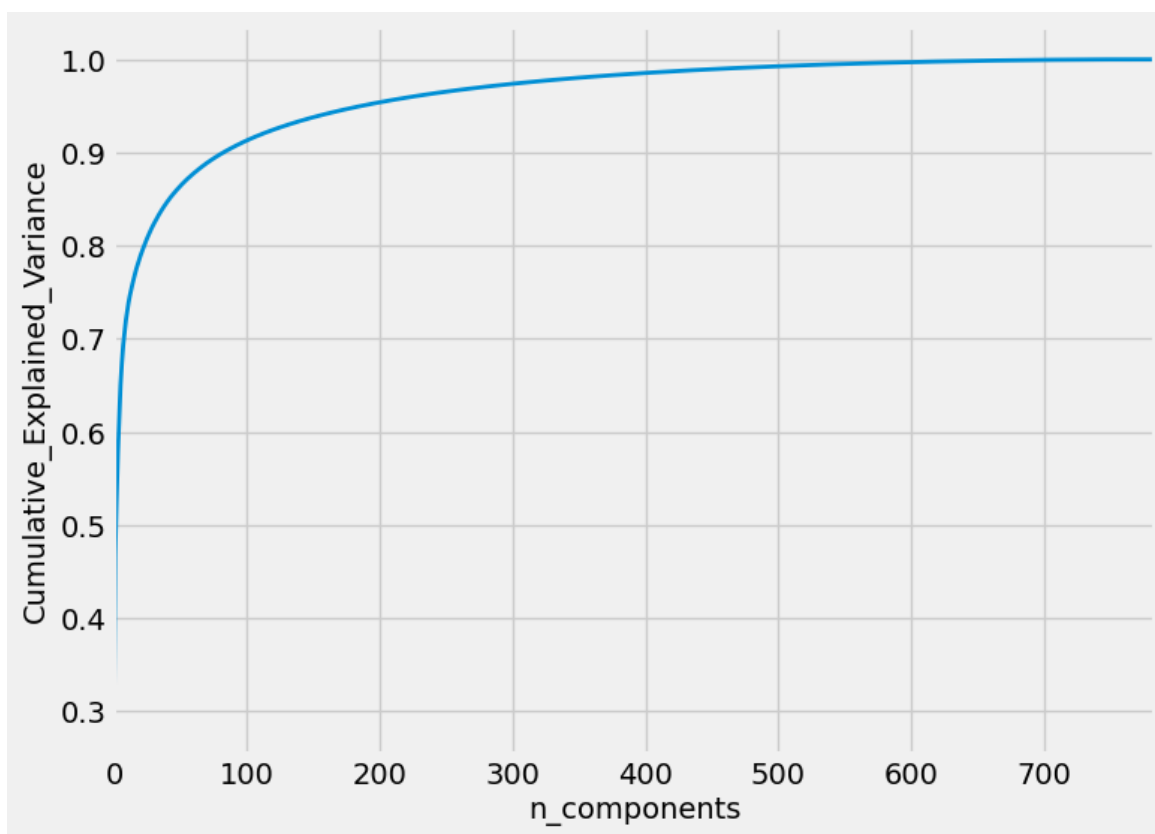
2.2.2 Trực quan hóa sử dụng PCA

Trong khi làm việc với dữ liệu đa chiều, các mô hình học máy thường có vẻ quá khớp và điều này làm giảm khả năng khái quát hóa qua các ví dụ của tập huấn luyện. Do đó, điều quan trọng là phải thực hiện các kỹ thuật giảm kích thước trước khi tạo mô hình. Một trong các phương pháp phổ biến trong học máy chính là phương pháp PCA mà chúng ta sẽ tìm hiểu ngay sau đây.

**PCA là gì ?*

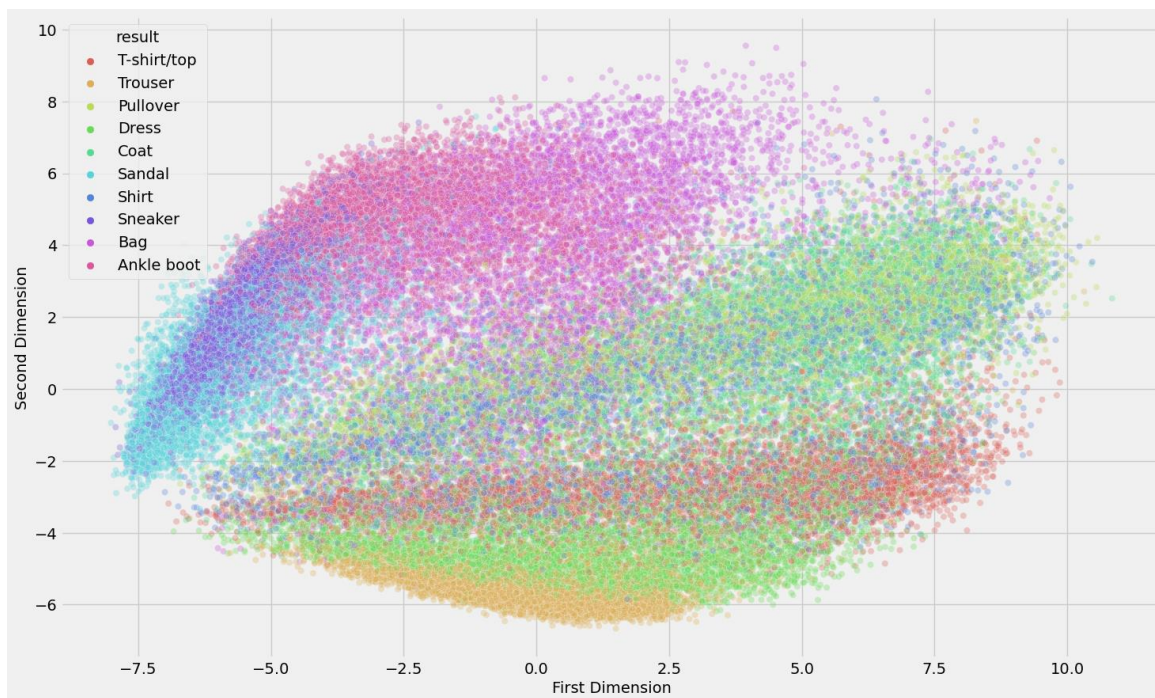
PCA (Principal Component Analysis): Thuật toán thống kê sử dụng phép biến đổi trực giao để biến đổi một tập hợp dữ liệu từ một không gian nhiều chiều sang một không gian mới ít chiều hơn (2 hoặc 3 chiều) nhằm tối ưu hóa việc thể hiện sự biến thiên của dữ liệu.

Sử dụng thư viện scikit-learn triển khai PCA với số thành phần chính là 2.



Đồ thị thể hiện sự hiệu quả khi giảm chiều với số chiều từ 0 đến 784

Trực quan hóa sự phân bố của các nhãn với 2 thành phần chính:



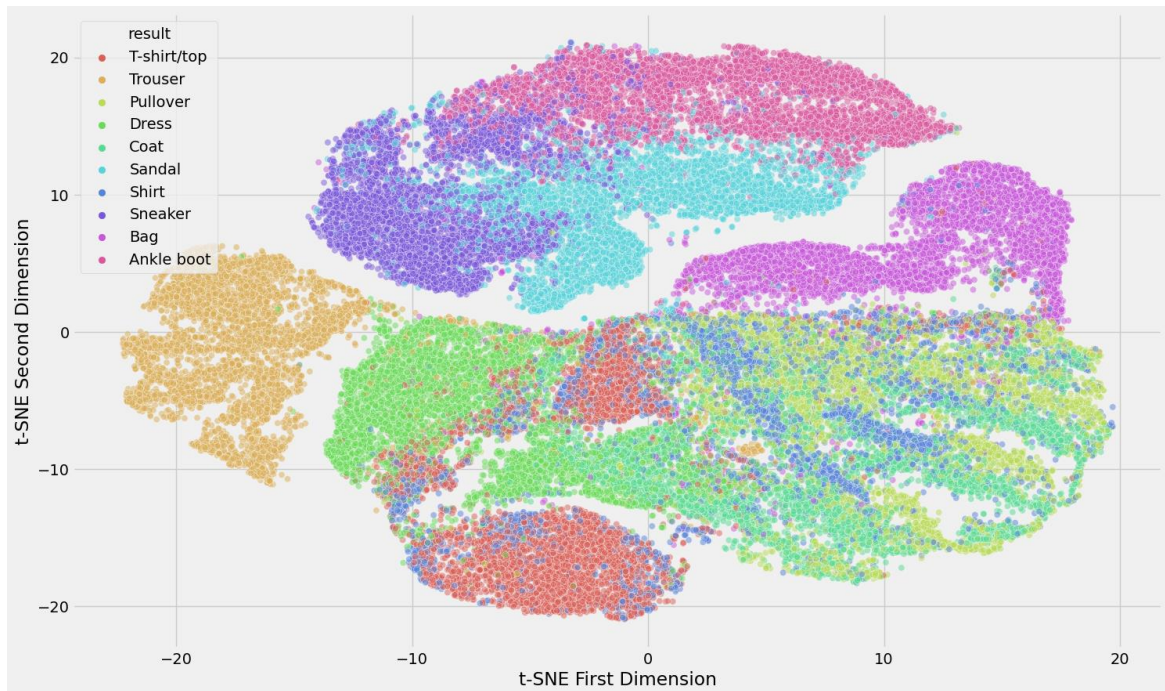
Đồ thị phân tán các nhãn

Nhận xét: Sau khi sử dụng kỹ thuật PCA giảm về 2 thành phần chính, đồ thị có độ tách biệt thấp giữa các nhãn. Ta cần sử dụng một kỹ thuật hiệu quả hơn.

2.2.3 Trực quan hóa sử dụng t-SNE

T-Distributed Stochastic Neighbor Embedding (t-SNE) là một kỹ thuật khác để giảm chiều dữ liệu và nó đặc biệt phù hợp để trực quan hóa các bộ dữ liệu đa chiều. Đây là một kỹ thuật dựa trên xác suất.

Sau khi sử dụng kỹ thuật t-SNE giảm số chiều của bộ dữ liệu về 2, trực quan hóa sự phân bố của các nhãn.



Đồ thị phân tán theo t-SNE

Nhận xét: Sau khi sử dụng kỹ thuật t-SNE, đồ thị với các nhãn được phân tách rõ ràng. Do đó, t-SNE rất phù hợp cho việc trực quan hóa bộ dữ liệu đa chiều.

Chương 3. Xây dựng và thực thi chương trình

Sử dụng thư viện: numpy, pandas, tensorflow, matplotlib, seaborn, sklearn, keras.

Bộ dữ liệu training: 80% bộ dữ liệu training gốc

Bộ dữ liệu validation: 20% bộ dữ liệu training gốc

Bộ dữ liệu testing: Toàn bộ dữ liệu thuộc bộ dữ liệu testing gốc

Đầu vào mô hình: Dữ liệu dưới dạng số, theo chuẩn min-max

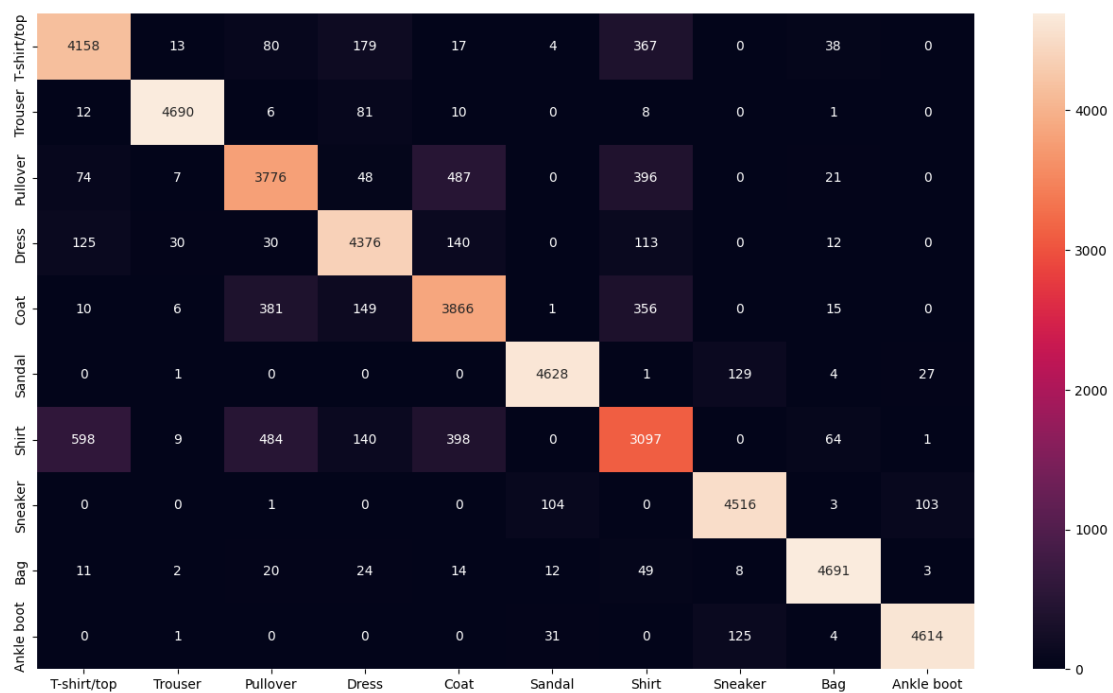
Đầu ra mô hình: Các nhãn được đánh số từ 0 đến 9

Chỉ số đánh giá: Accuracy.

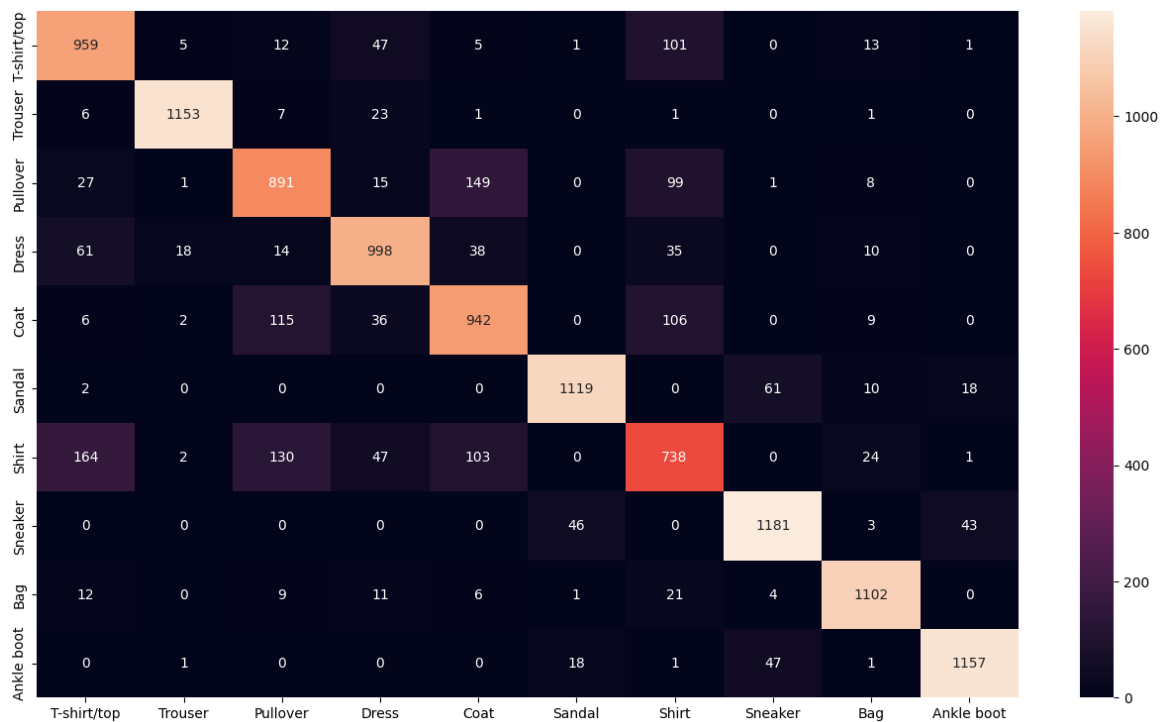
3.1 Mô hình Multinomial Logistic Regression

Kết quả chạy trên bộ dữ liệu training, validation:

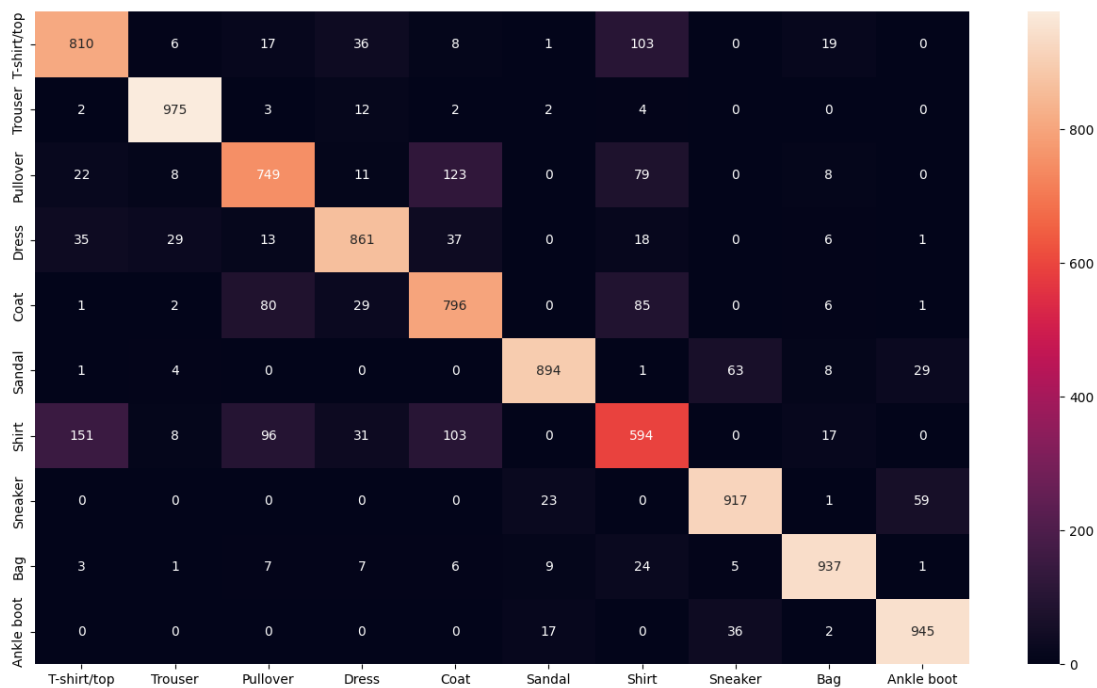
*** Kết quả chạy trên bộ dữ liệu Training: 88.36%**



***Kết quả chạy trên bộ dữ liệu Validation: 85.33%**

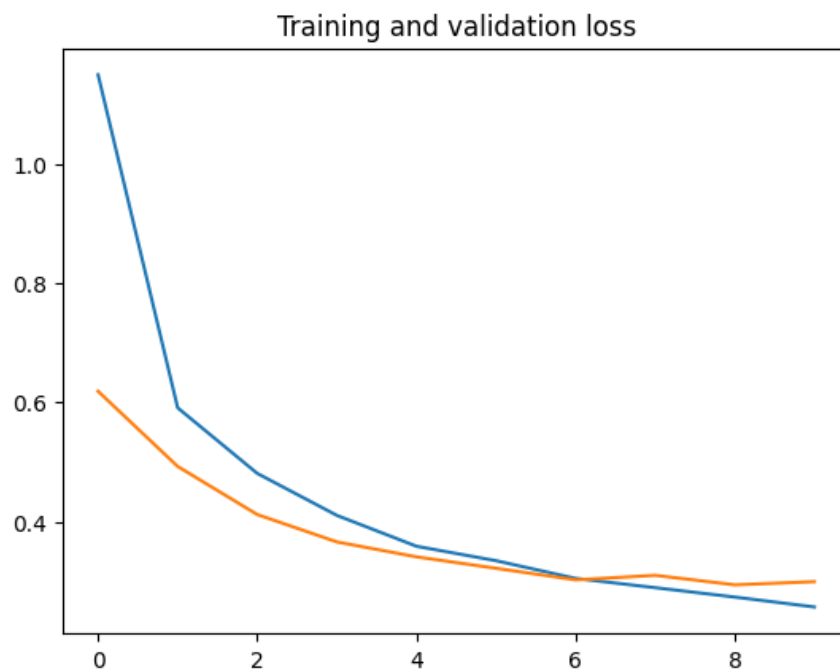
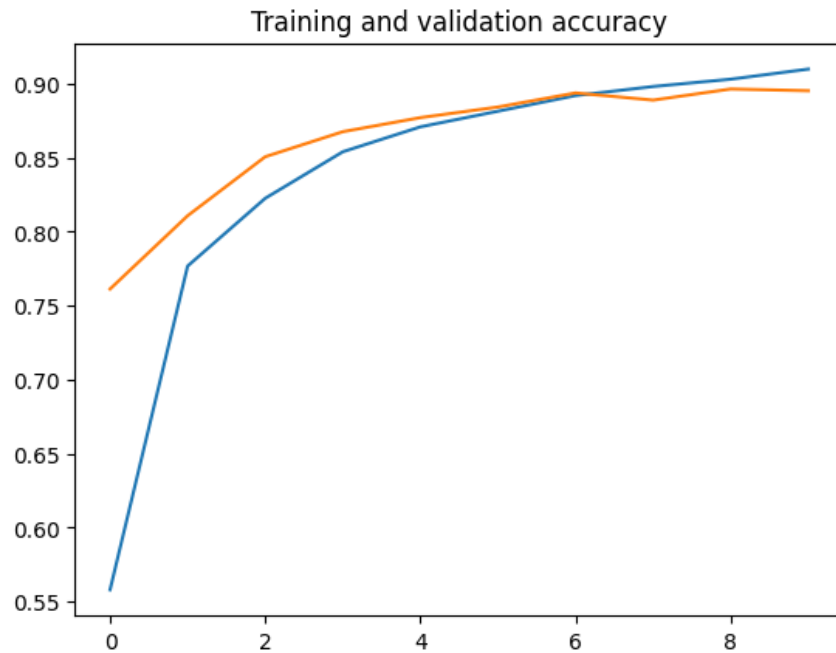


***Kết quả chạy trên bộ dữ liệu Testing: 84.59%**

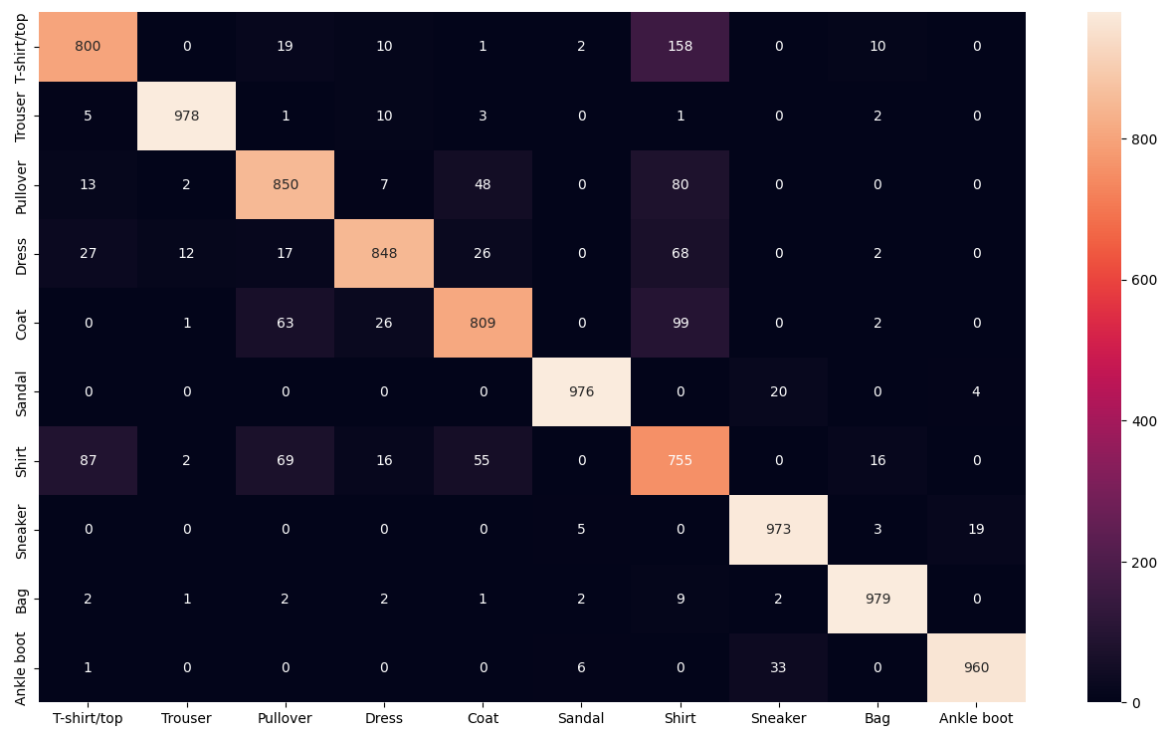


3.2 Mô hình Convolutional Neural Network

Kết quả chạy của mô hình trên tập training và validation với 10 epochs:



***Kết quả chạy trên bộ dữ liệu Test là: 89,27%**



Chương 4: Kết luận và phương hướng phát triển

Nhận diện hình ảnh thời trang là một đề tài nghiên cứu quan trọng được sử dụng rộng rãi trong các nền tảng thương mại điện tử, mạng xã hội và thiết kế thời trang. Ứng dụng của nó giúp việc tìm kiếm và đề xuất sản phẩm, xác định chủng loại quần áo trở nên dễ dàng hơn, góp phần tạo nên những sự đổi mới sáng tạo, tăng độ chính xác, hiệu quả trong phân tích xu hướng thời trang tương lai.

Trong đề tài này, chúng ta đã áp dụng các phương pháp nhận diện hình ảnh để giúp phân loại và nhận diện các sản phẩm thời trang trong bộ dữ liệu Fashion-MNIST. Cụ thể, chúng ta đã sử dụng hai phương pháp cơ bản là Convolutional Neural Network và Multinomial Logistic Regression. Kết quả cho thấy, độ chính xác của phương pháp Multinomial Logistic Regression thấp hơn so với Convolutional Neural Network.

Từ kết quả thu được, chúng ta đề xuất một số phương hướng phát triển trong tương lai để cải thiện độ chính xác của mô hình như: Sử dụng phương pháp kết hợp giữa Convolutional Neural Network và K-Means, kết hợp với các phương pháp giảm chiều dữ liệu khác để đạt được độ chính xác cao hơn và tăng hiệu quả khi nhận diện và phân loại hình ảnh trong bộ dữ liệu lớn hơn. Áp dụng các mô hình học sâu cao cấp: VGG, ResNet, để cải thiện độ chính xác và giảm chi phí tính toán cho bài toán nhận diện hình ảnh thời trang.

Nhóm 05 muốn gửi cảm ơn sâu sắc tới thầy Cao Văn Chung vì đã tận tình hướng dẫn chúng em hoàn thành báo cáo này. Chúng em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- [1] Mitchell, Tom M. Machine learning. New York: McGraw-Hill, 1997.
- [2] Rashid, Tariq. Make Your Own Neural Network: A Gentle Introduction to Neural Networks and Deep Learning with Python. O'Reilly Media, 2017.
- [3] Kaggle, <https://www.kaggle.com/zalando-research/fashionmnist>, 2019.
- [4] S. S. Kadam, A. C. Adamuthe, and A. B. Patil, “CNN model for image classification on MNIST and fashion-MNIST dataset”, Journal of Scientific Research, vol. 64, no. 2, pp. 374–384, 2020.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.
- [6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
- [7] Hilbe, Joseph M. Logistic Regression Models. Chapman & Hall/CRC, 2009.
- [8] Jolliffe, Ian T. Principal Component Analysis. Springer Science & Business Media, 2002.
- [9] Arthur, David, and Sergei Vassilvitskii. “The K-Means Clustering Algorithm”. Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2007. 445-54.
- [10] Maaten, Laurens van der, and Geoffrey Hinton. “t-Distributed Stochastic Neighbor Embedding”. Journal of Machine Learning Research 9.11 (2008): 2579-2605.