PHÂN CỤM DỮ LIỆU DỰA TRÊN MẠNG NƠ RON HỌC SÂU

Phạm Thế Anh[1], Nguyễn Hoàng Long[1], Nguyễn Văn Cường[1], Hoàng Anh Công[2]

TÓM TẮT

Phân cụm dữ liệu (clustering) là bài toán cơ bản của lĩnh vực khoa học máy tính và có nhiều ứng dụng trong thực tiễn, đặc biệt là phân tích dữ liệu lớn và khai phá dữ liệu. Các thuật toán phân cụm truyền thống như K-means, MeanShift đã được ứng dụng trong nhiều năm qua nhưng vẫn tồn tại nhiều hạn chế liên quan đến độ chính xác phân cụm. Bài báo này nghiên cứu các mô hình mạng nơ ron học sâu, cụ thể là mạng AutoEncoder để giải quyết bài toán phân cụm. Kết quả thực nghiệm trên cơ sở dữ liệu chuẩn cho hệ thống có độ chính xác phân cụm cao vượt trội so với các phương pháp truyền thống.

Từ khoá: *Mạng AutoEncoder, phân cụm dữ liệu, mạng nơ ron học sâu.*

1. ĐẶT VẤN ĐỀ

Cho trước một tập dữ liệu bất kỳ, bài toán phân cụm dữ liệu thực hiện chia tập dữ liệu cho trước thành các cụm sao cho các mẫu dữ liệu trong mỗi cụm sẽ có cùng các đặc trưng nào đó. Một cách lý tưởng, các mẫu dữ liệu trong mỗi cụm sẽ giống nhau nhất có thể về mặt các đặc trưng tiềm ẩn (latent features). Bài toán phân cụm dữ liệu có rất nhiều ứng dụng trong thực tiễn như phân tích dữ liệu lớn, khai phá dữ liệu, phân vùng ảnh, hỗ trợ ra quyết định. Đặc biệt, phân vùng ảnh là một ứng dụng cụ thể của phân cụm dữ liệu và được ứng dụng rất hiệu quả trong các bài toán như tra cứu ảnh dựa trên nội dung, phân tích nội dung ảnh, hiểu ảnh, mô tả đặc trưng, nhận dạng đối tượng. Phân vùng ảnh yêu cầu các điểm ảnh trong mỗi vùng phải có cùng tính chất đặc trưng (ví dụ dựa trên màu ảnh, kết cấu, hình dáng, hay chuyển động,…) và có tính liên thông với nhau. Tuy nhiên, bài toán phân cụm dữ liệu thì không yêu cầu tính chất liên thông đó. Các điểm dữ liệu trong mỗi cụm thường có chung các tính chất hoặc tương tự nhau.

Các thuật toán truyền thống cho bài toán phân cụm được đề xuất từ rất sớm và tiêu biểu là K-means [1], MeanShift [2]. Mỗi thuật toán có ưu điểm, thế mạnh và hạn chế riêng. Chẳng hạn, K-means yêu cầu tham số đầu vào là số lượng các cụm. Ngoài ra, K-means chỉ thích hợp với dữ liệu có phân bố dạng hình khối cầu do hàm mục tiêu sử dụng độ đo Euclidean. MeanShift không yêu cầu biết trước số lượng các cụm nhưng có hạn chế về độ chính xác và tốc độ xử lý cao.

Trong thời gian gần đây, với sự phát triển và tiến bộ mạnh mẽ của công nghệ mạng nơ ron nhân tạo học sâu, các giải pháp hiện đại cho bài toán phân cụm đã được nghiên cứu và phát triển dựa trên kiến trúc mô hình AutoEncoder. Kết quả thử nghiệm ban đầu cho

[1] *Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Hồng Đức, Email: phamtheanh@hdu.edu.vn*
[2] *Trường Đại học Văn hóa, Thể thao và Du lịch Thanh Hóa*

thấy tiềm năng của lĩnh vực nghiên cứu mới này là rất lớn và thuyết phục. Do vậy, trong bài báo chúng tôi nghiên cứu về mạng nơ ron học sâu, cụ thể là các mạng AutoEncoder, và ứng dụng để xây dựng kiến trúc mạng phù hợp cho bài toán phân cụm dữ liệu.

2. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU

2.1. Phương pháp phân cụm truyền thống

Thuật toán K-means [1] là một thuật toán phân cụm phổ biến nhất trong nhiều lĩnh vực khác nhau như xử lý tín hiệu, xử lý ảnh, thị giác máy và máy học. Trong nhiều năm qua, K-means vẫn là thuật toán phân cụm phổ biến và mặc định được sử dụng trong nhiều ứng dụng khác nhau. Tư tưởng chính của K-means đó là phân cụm các *phần tử* (sample points) thành K cụm khác nhau sao cho các phần tử trong mỗi cụm sẽ giống nhau nhất có thể. Về lý thuyết, các phần tử có thể là bất kỳ đối tượng gì miễn là tồn tại một độ đo khoảng cách trong không gian vector (vector space) giữa hai đối tượng. Trong thực tế, chúng ta thường làm việc với các phần tử thuộc không gian thực và thường được biểu diễn bởi *một điểm* (point) trong không gian này. Mỗi cụm trong thuật toán K-means sẽ đại diện cho một nhóm các phần tử và được đặc trưng bởi phần tử tâm của cụm. Tâm của cụm được tính bởi phần tử trung bình của các phần tử thuộc cụm đó. Các phần tử thuộc một cụm sẽ có cùng một tính chất nào đó và được cụ thể hóa bằng độ đo khoảng cách từ phần tử đó đến tâm cụm. Mỗi phần tử sẽ được gán vào cụm có tâm gần hơn bất kỳ cụm khác. Thuật toán K-means dựa trên tư tưởng tối ưu hóa kỳ vọng EM (Expectation-Maximization) là một thủ tục lặp gồm 2 pha chính: tính toán các tâm và cập nhật các cụm. K-means có ưu điểm hoạt động khá hiệu quả, nhưng có thể rơi vào điểm tối ưu hóa cục bộ và cho kết quả không tốt. Ngoài ra, thuật toán yêu cầu phải có một tham số đầu vào là số lượng tâm cụm.

MeanShift là một thuật toán phân cụm cho kết quả khá tốt, được đề xuất từ năm 1975 bởi Fukunaga và Hostetler [2] nhưng không được sử dụng rộng rãi như K-means. MeanShift có nhiều ứng dụng thực tế trong lĩnh vực thị giác máy như phân vùng ảnh, theo vết đối tượng (object tracking), tách đối tượng... Mục đích ban đầu của thuật toán MeanShift là dò tìm các vùng có mật độ dày trong một tập dữ liệu. Nói một cách chính xác, MeanShift dò tìm các điểm có tần suất xuất hiện cao (còn gọi là các "modes hay "local maxima") trong một hàm phân bố xác suất (PDF - probability density function). MeanShift không yêu cầu tham số truyền vào là số lượng tâm cụm, cho kết quả phân cụm khá tốt. Tuy nhiên thuật toán có hai nhược điểm liên quan đến việc chọn kích thước cửa sổ phân cụm và độ phức tạp tính toán cao.

2.2. Phương pháp phân cụm dựa trên mạng nơ ron học sâu

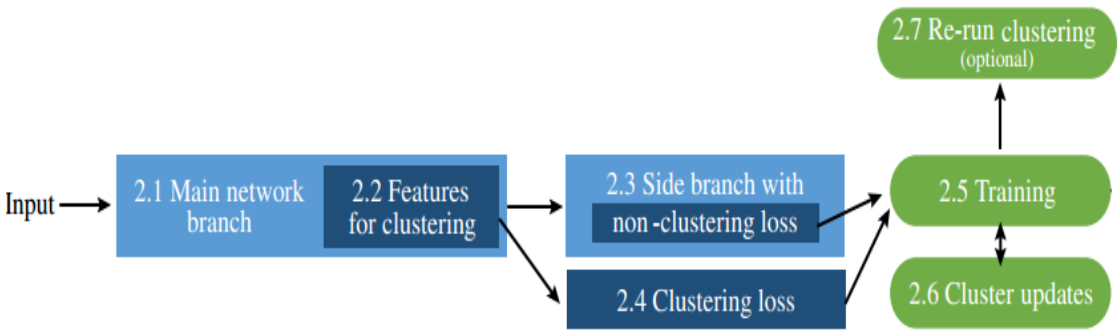
2.2.1. Kiến trúc tổng quát của hệ thống phân cụm

Theo các tác giả trong [3], kiến trúc tổng quan một hệ thống phân cụm dựa trên mạng nơ ron học sâu có thể được mô tả trên hình 1 gồm các thành phần như sau:

Thành phần mạng chính hay mạng cơ sở (main/baselinenetwork branch);

Thành phần các hàm mục tiêu;

Thành phần huấn luyện và cập nhật cụm.



Hình 1. Kiến trúc tổng quát hệ thống phân cụm dựa trên mạng nơ ron học sâu [3]

Các thành phần trên được trình bày chi tiết như sau.

i) Mạng cơ sở: được tạo thành từ mạng nơ ron học sâu có tác dụng biến đổi dữ liệu đầu vào sang không gian mới (Embedded Feature Space) thường có số chiều nhỏ hơn nhiều so với số chiều trong không gian gốc. Đầu ra của mạng này sẽ là vector đặc trưng có khả năng biểu diễn, tóm tắt, mô tả cơ bản các thông tin quan trọng ẩn chứa trong dữ liệu đầu vào.

ii) Hàm mục tiêu: hàm mục tiêu đóng vai trò quan trọng của các hệ thống học sâu. Các hàm mục tiêu có nhiệm vụ điều khiển, định hướng quá trình học để giải quyết một bài toán cụ thể nào đó. Trong bài toán phân cụm, các hàm mục tiêu thường được chia làm hai loại sau:

Non-clustering Loss (NCL): Như tên gọi của nó, các hàm mục tiêu NCL không cần phải có mối liên hệ với mục tiêu phân cụm. Trong hầu hết trường hợp, hàm NCL phổ biến được sử dụng là hàm tái tạo (Reconstruction Loss) dùng trong các mô hình AutoEncoder.

Clustering Loss (CL): Hàm mục tiêu này được sử dụng để giúp định hướng mô hình trong quá trình học sẽ khai thác các đặc trưng có lợi cho bài toán phân cụm nhất có thể. Các hàm mục tiêu phổ biến thường được sử dụng bao gồm:

Hàm mục tiêu K-means [4]:

$$L(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} s_{ik} \|z_i - \mu_k\|^2$$

Trong đó: K là số cụm dữ liệu, N là tổng số điểm dữ liệu, z_i là vector đặc trưng trong không gian mới của điểm dữ liệu x_i, μ_k là tâm cụm thứ k, s_{ik} là biến logic nhận giá trị 0 hoặc 1 nhằm thể hiện điểm dữ liệu z_i có thuộc tâm μ_k hay không.

Hàm mục tiêu CAH (Cluster Assignment Hardening loss): Các tác giả trong [5] đề xuất việc dùng giá trị mềm để gán các điểm về tâm cụm (Soft Cluster Assignments). Nghĩa là s_{ik} sẽ không nhận giá trị 0 và 1 mà là một giá trị trong đoạn [0, 1].

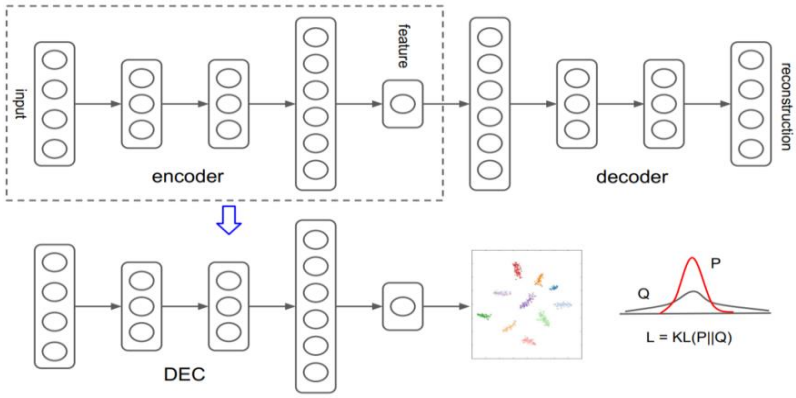
2.2.2. Mô hình phân cụm DEC (Deep Embedded Clustering):

DEC [4] gồm hai giai đoạn như sau:

Giai đoạn 1: Khởi tạo các tham số bằng cách sử dụng mạng Deep Autoencoder.

Giai đoạn 2: Tối ưu các tham số (phân cụm).

Mô hình mạng DEC được đề xuất có kiến trúc như sau:



Hình 2. Kiến trúc mạng DEC [5]

Mô hình Autoencoder trong kiến trúc mạng DEC sử dụng hàm mục tiêu có tên gọi là tái tạo (Reconstruction Loss) kí hiệu là L_r dùng để đo sự sai khác giữa dữ liệu gốc đầu vào và dữ liệu sau khi qua Decoder tái tạo lại. Công thức phổ biến của hàm mục tiêu này thường là dùng độ đo khoảng cách L2 như sau:

$$L = d_{\mathrm{AE}}\big(x_i, f(x_i)\big) = \sum_i \|x_i - f(x_i)\|^2$$

Trong đó: x_i là một điểm dữ liệu đầu vào, $f(x_i)$ là điểm dữ liệu tái tạo từ thành phần Decoder của x_i.

Thông thường mô hình Autoencoder sử dụng hàm L_r chỉ có chức năng dùng để khử nhiễu cho dữ liệu. Để thực hiện phân cụm dữ liệu, các tác giả sử dụng thêm một hàm mục tiêu định hướng phân cụm dựa trên độ đo khoảng cách KL (Kullback-Leibler) để đo khoảng cách giữa phân bố xác suất phân cụm mềm với một phân bố xác suất đích để điều hướng quá trình học. Cụ thể, phân cụm mềm giữa các điểm dữ liệu z_i và tâm cụm μ_j, được tính theo công thức sau:

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{j'}\left(1 + \|z_i - \mu_{j'}\|^2/\alpha\right)^{-\frac{\alpha+1}{2}}}$$

Trong đó: q_{ij} được hiểu là xác suất để điểm dữ liệu z_i thuộc tâm cụm μ_j, q_{ij} nhận giá trị trong đoạn [0,1].

Nhóm tác giả đề xuất cần tinh chỉnh việc lặp đi lặp lại phân cụm bằng cách học từ một phân bố xác suất đích p_{ij} để điều hướng q_{ij} tiến gần giống với p_{ij}. Bản chất là cực tiểu hoá hàm L:

$$L = \mathrm{KL}(P \parallel Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Điều quan trọng là cần xác định được p_{ij}. Các tác giả mong muốn xác định p_{ij} để có những thuộc tính sau:

Tăng cường khả năng dự đoán (cải thiện độ tinh khiết của cụm): Xác suất của q_{ij} cao, giá trị q_{ij} gần 0 hoặc 1 để xác định rõ điểm z_i có được gán về tâm μ_j hay không?

Ưu tiên học từ các điểm có độ tự tin cao (qij cao) để tăng khả năng phân lớp các điểm có độ tự tin thấp (các điểm khó).

Chuẩn hoá giá trị đóng góp của các điểm để giảm sự ảnh hưởng của các cụm lớn.

Các tác giả đề xuất công thức tính p_{ij} như sau:

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij'}^2/f_{j'}}$$

Mô hình DEC cho kết quả phân cụm khá tốt nhưng quá trình huấn luyện phức tạp do áp dụng phương pháp huấn luyện thủ công từng tầng. Ngoài ra, mạng AutoEncoder sử dụng các tầng kết nối đầy đủ nên độ phức tạp tính toán lớn.

3. PHÂN CỤM DỰA TRÊN MẠNG NƠ RON HỌC SÂU

Trong phần này chúng tôi nghiên cứu và đánh giá thực nghiệm một số mô hình học sâu tiên tiến cho bài toán phân cụm. Cụ thể, chúng tôi nghiên cứu hai mô hình DEC [5] và IDEC [6]. Khác với các nghiên cứu gốc [5][6], đóng góp chính trong phần này là thực hiện các nghiên cứu nhằm đánh giá hiệu năng của các mô hình trên bằng cách phân tích hiệu năng của từng thành phần mạng. Ngoài ra, chúng tôi cũng thay thế các mạng kết nối đầy đủ FC (Fully Connected) bằng các mạng nhân chập CNN (Convolutional Neural Network) nhằm giảm độ phức tạp mô hình và khai thác tính tương quan của các điểm ảnh. Dữ liệu thử nghiệm là các tập STL-10 [7] và Cifar-10 [8]. Cụ thể, chúng tôi thực hiện các nghiên cứu hiệu năng sau cho 2 mạng trên:

Đầu vào của DEC, IDEC là các vector đặc trưng (4096 chiều) được trích chọn từ mạng VGG16 (đã được tiền huấn luyện trên tập dữ liệu ImageNet [9]).

Đầu vào của DEC, IDEC là dữ liệu ảnh không qua tiền xử lý.

Thay thế các mạng kết nối đầy đủ trong DEC bằng mạng nhân chập CNN.

Dùng trực tiếp đầu ra của VGG16 và làm mịn bởi hàm mục tiêu định hướng phân cụm.

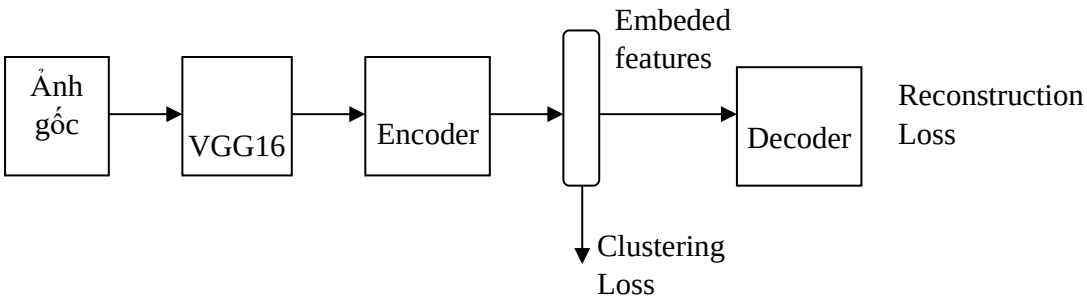
3.1. DEC, IDEC với đầu vào được tiền xử lý bởi VGG16

Cả DEC và IDEC đều dựa trên mô hình AutoEncoder để biến đổi đầu vào sang không gian đặc trưng mới (embeded feature space). Về cơ bản, một mô hình AutoEncoder gồm 2 phần: Encoder và Decoder. Phần Encoder biến đổi dữ liệu đầu vào sang không gian mới (thường có số chiều nhỏ). Phần Decoder có tác dụng tái tạo dữ liệu ngược lại từ không gian mới về không gian gốc sao cho dữ liệu tái tạo giống nhất có thể với dữ liệu gốc. Do vậy, hệ thống sử dụng hàm mục tiêu có tên gọi là tái tạo (Reconstruction Loss), bản chất là đo sự sai khác giữa tín hiệu tái tạo và tín hiệu gốc. Công thức phổ biến của hàm mục tiêu này thường là dùng độ đo khoảng cách L2 như sau:

$$L = d_{AE}\big(x_i, f(x_i)\big) = \sum_i \|x_i - f(x_i)\|^2$$

Trong đó: x_i là một điểm dữ liệu đầu vào, $f(x_i)$ là điểm dữ liệu tái tạo từ thành phần Decoder của x_i.

Các tầng mạng của DEC và IDEC đều dựa trên mạng kết nối đầy đủ. Để tăng độ chính xác và giảm độ phức tạp tính toán, các tác giả có thể tiền xử lý ảnh đầu vào bằng cách cho dữ liệu ảnh vào mạng VGG16 [10] (đã được huấn luyện trên tập dữ liệu lớn ImageNet) để trích chọn vector đặc trưng (4096) chiều. Hình 3 minh họa kiến trúc tổng quát của cách tiếp cận này.



Hình 3. Kiến trúc hệ thống DEC, IDEC và VGG16

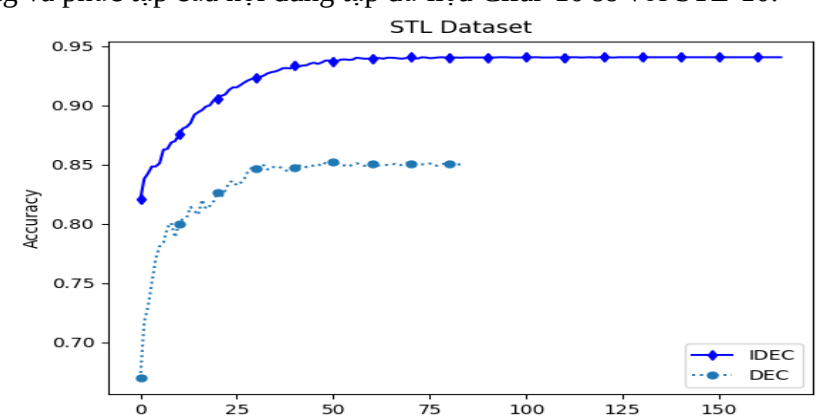
Cấu hình mạng và tham số mạng được giữ nguyên như trong các bài báo gốc của DEC, IDEC. Kết quả phân cụm được đo bằng độ chính xác ACC được tính như sau:

$$ACC = \max_{m} \frac{\sum_{i=1}^{n} \mathbf{1}\{l_i = m(c_i)\}}{n}$$

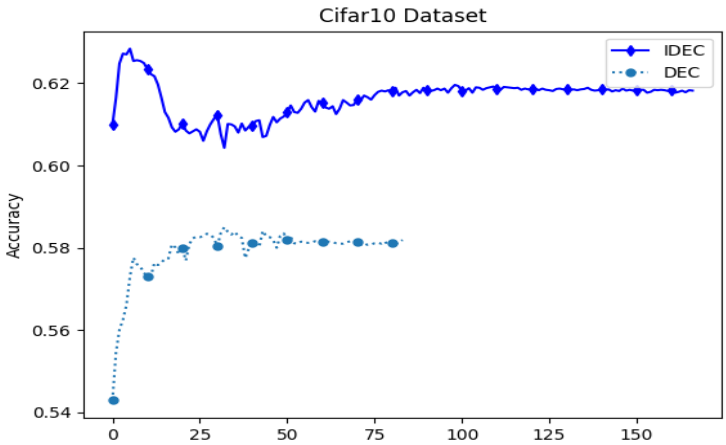
Trong đó l_i là nhãn đúng của điểm dữ liệu thứ i, c_i là nhãn được gán của điểm dữ liệu thứ i cho một cụm bởi thuật toán phân cụm, m đại diện cho tất cả các phép gán giữa các cụm và các nhãn đúng của dữ liệu. Mỗi phép gán sẽ gán một nhãn đúng cho một cụm tương ứng.

Độ chính xác ACC so sánh kết quả của thuật toán phân cụm không giám sát (unsupervised clustering) với kết quả của phân cụm đúng (ground truth) được xác định trước. ACC giúp đánh giá mức độ khớp giữa hai phân cụm và cho biết mức độ hiệu quả của thuật toán.

Kết quả thực nghiệm trên hai tập dữ liệu STL-10 và Cifar-10 được thể hiện trên hình 4 và hình 5. Trên cả hai tập dữ liệu IDEC kết hợp với VGG16 cho độ chính xác phân cụm tốt hơn DEC+VGG16. Khoảng cách khác biệt về độ chính xác khoảng 0.1-0.15. Điều này có thể được giải thích bởi IDEC đã tối ưu đồng thời cả hai hàm mục tiêu: hàm định hướng phân cụm và hàm huấn luyện mô hình AutoEncoder. Ngoài ra, cả hai hệ thống đều giảm hiệu năng (độ chính xác) khi làm việc trên tạp Cifar-10 khoảng 30%, nhiều khả năng là do tính đa dạng và phức tạp của nội dung tập dữ liệu Cifar-10 so với STL-10.



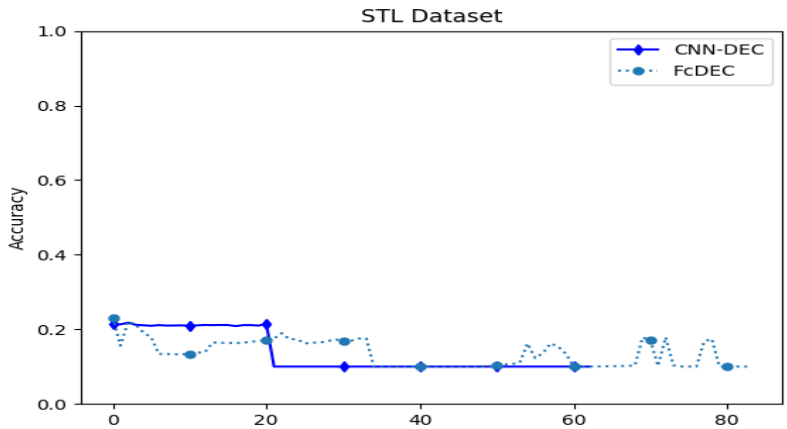
Hình 4. Độ chính xác của DEC và IDEC khi kết hợp VGG16 trên tập dữ liệu STL



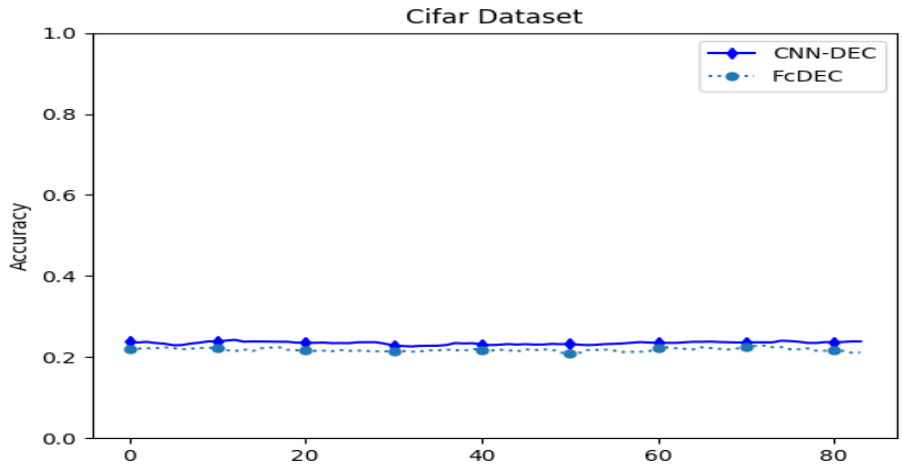
Hình 5. Độ chính xác của DEC và IDEC khi kết hợp VGG16 trên tập dữ liệu Cifar-10

3.2. DEC với đầu vào là dữ liệu ảnh gốc

Trong phần này, chúng tôi đánh giá hiệu năng của DEC với đầu vào là ảnh không qua tiền xử lý. Nghĩa là, hệ thống không dùng thành phần VGG16 như mô tả trên hình 3. Đồng thời, chúng tôi cũng so sánh hiệu năng khi dùng DEC với tầng kết nối đầy đủ FC và tầng nhân chập CNN. Kết quả thực nghiệm trên hai tập dữ liệu STL-10 và Cifar-10 được thể hiện trên hình 6 và hình 7. Cụ thể, kết quả trên hai tập dữ liệu khá giống nhau và thể hiện độ chính xác tương đương giữa việc dùng FC hay CNN. Tuy nhiên, khi dùng tầng CNN, chẳng hạn xét trên tập dữ liệu STL-10, thì mô hình sẽ có số lượng tham số (0.4 triệu tham số) nhỏ hơn so với khi dùng tầng FC (15 triệu tham số). Kết quả trên hình 6,7 cũng cho thấy độ chính xác phân cụm giảm mạnh khi mô hình hoạt động trực tiếp trên dữ liệu là ảnh đầu vào. So với kết quả trên hình 4,5 chúng ta có thể thấy độ chính xác ACC giảm khoảng 40%-60%, nguyên nhân là do mạng VGG16 đã được tiền huấn luyện trên tập dữ liệu ImageNet có số lượng ảnh rất lớn (hàng triệu ảnh). Do vậy, mô hình đã được học đầy đủ các đặc trưng cơ sở của các lớp đối tượng trong ImageNet. Nếu không dùng VGG16 để trích chọn vector đặc trưng cơ sở, các mô hình FcDEC hay CNN-DEC sẽ được huấn luyện trực tiếp trên tập dữ liệu STL-10 và Cifar-10 vốn có số lượng ảnh khá nhỏ (khoảng 5000 ảnh). Do vậy, mô hình sẽ bị giới hạn về độ chính xác do chưa được huấn luyện một cách đầy đủ.



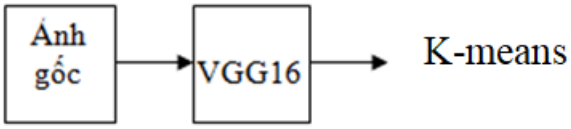
Hình 6. Độ chính xác của CNN-DEC và FcDEC trên tập dữ liệu STL



Hình 7. Độ chính xác của CNN-DEC và FcDEC trên tập dữ liệu Cifar-10

3.3. Dùng trực tiếp các đặc trưng của VGG16

Trong phần này, chúng tôi không sử dụng DEC và IDEC mà dùng trực tiếp đầu ra của VGG16 làm đầu vào của thuật toán phân cụm K-means. Kiến trúc của cách tiếp cận này được mô tả trên hình 8.



Hình 8. Kiến trúc hệ thống phân cụm chỉ dùng VGG16

Kết quả thực nghiệm của VGG16 trên hai tập dữ liệu STL-10 và Cifar-10 được thể hiện trên bảng sau.

Bảng 1. Độ chính xác phân cụm (ACC) khi sử dụng trực tiếp các đặc trưng trích chọn với mô hình tiền huấn luyện VGG16

Hệ thống	STL-10	Cifar-10
VGG16	0.77	0.58

Kết quả này cho thấy mặc dùng sử dụng trực tiếp đầu ra của VGG16 nhưng độ chính xác phân cụm vẫn khá ấn tượng (ACC = 0.77 trên STL-10 và ACC = 0.58 trên Cifar-10). Điều này một lần nữa khẳng định vai trò của dữ liệu trong huấn luyện các mạng nơ ron học sâu là khá quan trọng. VGG16 là một mạng cơ bản nhưng đủ sâu và có số lượng tham số lớn. Để huấn luyện đầy đủ mạng VGG16, chúng ta cần những tập dữ liệu lớn và phong phú như ImageNet. Khi đó, VGG16 có khả năng học những đặc trưng nổi bật của mỗi lớp đối tượng và có khả năng tổng quát hóa cao. Vì vậy, các đặc trưng mà VGG16 trích chọn trên hai tập dữ liệu STL-10 và Cifar-10 vẫn rất hữu ích để phục vụ các bài toán thị giác máy khác, trong trường hợp này là bài toán phân cụm dữ liệu.

4. KẾT LUẬN

Trong bài báo này, chúng tôi nghiên cứu về các mô hình mạng AutoEncoder và ứng dụng trong bài toán phân cụm dữ liệu. Các mô hình tiên tiến hiện nay là DEC, IDEC đều hoạt động dựa trên việc xây dựng các mô hình AutoEncoder để biến đổi dữ liệu đầu vào sang không gian đặc trưng tiềm ẩn mới. Sau đó, các đặc trưng này tiếp tục được làm mịn cho bài toán phân cụm bằng cách sử dụng một hàm mục tiêu định hướng phân cụm. Các hàm mục tiêu phổ biến được thiết kế dựa trên hàm KL hoặc K-means. Bài báo này cũng chỉ ra một điểm mới đó là việc dùng đơn lẻ các mô hình DEC hay IDEC trực tiếp trên dữ liệu đầu vào thì độ chính xác phân cụm khá thấp. Nghiên cứu này cũng chỉ ra rằng vai trò của các tầng FC và CNN là tương đồng về độ chính xác, ngoại trừ có sự khác biệt về độ phức tạp tính toán (CNN có lợi thế hơn). Do vậy, để cải tiến hiệu năng, việc dùng kết hợp DEC hay IDEC với một mạng nơ ron học sâu (chẳng hạn VGG16) đã được tiền huấn luyện trên một tập dữ liệu đủ lớn sẽ mang lại những kết quả rất ấn tượng. Cuối cùng, nghiên cứu này chỉ ra rằng, chúng ta có thể sử dụng trực tiếp các đặc trưng được trích chọn từ mạng VGG16 (đã được tiền huấn luyện trên ImageNet) để làm đầu vào cho một thuật toán phân cụm truyền thống (chẳng hạn K-means) và cũng đạt được độ chính xác khá ấn tượng. Trong các nghiên cứu tiếp theo, chúng tôi sẽ tìm hiểu tích hợp các kiến trúc mạng mới để cải tiến độ chính xác của bài toán phân cụm.

TÀI LIỆU THAM KHẢO

[1] J. B. MacQueen (1967), *Some Methods for classification and Analysis of Multivariate Observations*, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1: 281-297.

[2] Fukunaga, K., Hostetler, L. (1975), *The estimation of the gradient of a density function, with applications in pattern recognition*, IEEE Transactions on Information Theory, 21(1) 32-40.

[3] Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel, Daniel Cremers (2018), *Clustering with Deep Learning: Taxonomy and New Methods*, arXiv:1801.07648v2 [cs.LG].

[4] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, Mingyi Hong (2016), *Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering*, arXiv:1610.04794v2 [cs.LG].

[5] J. Xie, R. Girshick, A. Farhad (2016), *Unsupervised deep embedding for clustering analysis*, in Proceedings of the 33rd International Conference on Machine Learning.

[6] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin (2017), *Improved deep embedded clustering with local structure preservation*, In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), 1753-1759.

[7] Coates, Adam, Ng, Andrew Y, and Lee, Honglak (2011), *An analysis of single-layer networks in unsupervised feature learnin*g, In International Conference on Artificial Intelligence and Statistics, 215-223.

[8] Alex Krizhevsky (2009), *Learning multiple layers of features from tiny images*, Technical Report.

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei (2015), *ImageNet Large Scale Visual Recognition Challenge*, International Journal of Computer Vision (IJCV).

[10] Simonyan, K., Zisserman, A. (2015), *Very deep convolutional networks for large-scale image recognition*, 3rd International Conference on Learning Representations (ICLR 2015), 1-14.

[11] Rafael C. Gonzalez, Richard E. Woods (2007), *Digital Image Processing* (3rd Edition), Pearson, ISBN-13: 978-0131687288.

[12] LeCun, Yann, Bottou, Leon, Bengio, Yoshua, Haffner, Patrick (1998), *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86(11) 2278-2324.

DATA CLUSTERING BASED ON DEEP NEURAL NETWORKS

Pham The Anh, Nguyen Hoang Long, Nguyen Van Cuong, Hoang Anh Cong

ABSTRACT

Data clustering is a fundamental problem in the field of computer science and has many practical applications, especially in big data analysis and data mining. Traditional clustering algorithms such as K-means, MeanShift have been applied for many years but still have many limitations related to clustering accuracy. This paper investigates deep neural network models, specifically AutoEncoder networks, to address the clustering problem. Experimental results on standard datasets demonstrate that the system achieves significantly higher clustering accuracy compared to traditional methods.

Keywords: *AutoEncoder, data clustering, deep neural network.*

** Ngày nộp bài: 25/3/2023; Ngày gửi phản biện: 27/3/2023; Ngày duyệt đăng: 10/12/2023*