Journal of Korea Multimedia Society Vol. 21, No. 8, August 2018(pp. 952-959)
https://doi.org/10.9717/kmms.2018.21.8.952

다차원 데이터에 대한 심층 군집 네트워크의 성능향상 방법

이 현 진†

Performance Improvement of Deep Clustering Networks for Multi Dimensional Data

Hyunjin Lee†

ABSTRACT

Clustering is one of the most fundamental algorithms in machine learning. The performance of clustering is affected by the distribution of data, and when there are more data or more dimensions, the performance is degraded. For this reason, we use a stacked auto encoder, one of the deep learning algorithms, to reduce the dimension of data which generate a feature vector that best represents the input data. We use k-means, which is a famous algorithm, as a clustering. Sine the feature vector which reduced dimensions are also multi dimensional, we use the Euclidean distance as well as the cosine similarity to increase the performance which calculating the similarity between the center of the cluster and the data as a vector. A deep clustering networks combining a stacked auto encoder and k-means re-trains the networks when the k-means result changes. When re-training the networks, the loss function of the stacked auto encoder and the loss function of the k-means are combined to improve the performance and the stability of the network. Experiments of benchmark image ad document dataset empirically validated the power of the proposed algorithm.

Key words: Deep Learning, Clustering, Auto Encoder, K-means, Dimension Reduction, Deep Clustering Networks

1. 서 론

군집화(Clustering)는 빅데이터 시대에 생성되는 방대한 양의 데이터를 이용하여 시청자나 쇼핑몰 고객의 구매 내역, 유사 구매상품, 시간대등의 성향을 분석하여 개인화된 추천 서비스를 제공하거나, 뉴스를 추천하거나, 소셜 네트워크에서 유사 사용자들에게 맞춤형 서비스 제공 등 다양한 분야에서 활용되고 있다. 군집화는 유사한 데이터들이 서로 모이도록 데이터를 구분하는 가장 대표적인 비교사 학습(Unsupervised Learning)방법이다. 하지만 다른 알고리즘과 결합하여 교사 학습(Supervised Learning)을 위한 자동 데이터 라벨(Label) 생성이나, 데이터 시각화나 분석을 위한 전 처리 단계, 또는 연속적인 숫자 데이터를 이산 데이터로 변경시키는 작업들에 활용된다.

군집화 알고리즘은 입력 데이터의 형태에 따라서 성능차이가 크게 발생한다. 그렇기 때문에 문제와 데이터의 형태에 따라 다양한 유사도 평가 방법과 군집화 알고리즘들을 적용해야 한다. 데이터의 차원이 클 때는 군집화 알고리즘의 변경이나 유사도 평가 방법의 변경으로는 우수한 성능을 기대할 수 없다. 따라

※ Corresponding Author : Hyunjin Lee, Address: (03132) #321, Jongno Biz-well, 23 Samil-daero 30-gil, Jongno-gu, Seoul, Korea, TEL : +82-2-708-7863, FAX : +82-2-708-7749, E-mail : hjlee@mail.kcu.ac

Receipt date : Jul. 9, 2018, Approval date : Jul. 19, 2018
† Division of ICT Engineering, Korea Soongsil Cyber University

서 문제 영역 별 지식에 따라 특징 벡터를 추출한 후, 추출 된 특징을 대상으로 군집화 알고리즘이 적용하거나, 입력 데이터를 분리하기 쉬운 특징 공간(feature space)으로 매핑하기 위해서 차원 축소(Dimension Reduction)를 하는 방법이 많이 사용된다[1-5].

최근에는 데이터에 신경회로망 기법을 적용하여 차원을 축소하는 표현 학습(Representation Learning)을 사용한 후 군집화 알고리즘을 적용하여 두 문제를 동시에 최적화하는 연구들이 진행되고 있다. 이미지 데이터에 대해서는 교사 학습(Supervised Learning)인 컨볼루션 신경망 (Convolutional Neural Network, CNN)의 특징추출 층인 컨볼루션 층(Convolution Layer)을 사용한 표현 학습이 많이 사용된다[2,3]. 비이미지 데이터에 대해서는 비교사 학습(Unsupervised Learning) 방법인 심층신뢰신경망(Deep Belief Networks, DBN)이나 오토 인코더(Autoencoder)를 사용한 표현 학습이 많이 사용된다[4,5].

본 논문에서는 재학습시에 최소화하는 손실함수와 군집화시 최소화하는 유사도 함수의 개선을 통하여 다차원데이터에 대한 심층 군집화 네트워크의 성능을 향상시키고자 한다. 심층 군집화 네트워크는 오토 인코더를 적용하여 차원 축소를 수행하고, k-means 알고리즘을 적용하여 군집화를 수행한다. 이때 오토 인코더 손실함수와 소프트 할당(soft labeling) 확률에 대한 쿨백-라이블러 발산(Kullback-Leibler Divergence, KLD)과 군집 구조의 적절성을 평가하기 위한 k-means 손실을 결합한 손실함수를 최소화 하도록 네트워크를 재학습하여 군집화의 정확도와 안정성을 높이고자 한다. 데이터와 군집간의 유사도는 벡터 개념을 도입하여 데이터와 군집 중심 간의 거리인 유클리디안 거리(Euclidean Distance)와 데이터와 군집 중심 간의 방향인 코사인 계수(Cosine Coefficients)를 동시에 반영하여 다차원 데이터에 대한 성능을 높이고자 한다.

본 논문의 구성은 다음과 같이 진행된다. 2장에서는 관련 연구들에 대해서 살펴보고, 3장에서는 심층 군집 네트워크에 대해 제안하는 방법에 대해서 자세히 살펴본다. 4장에서는 다차원 데이터에 대한 실험을 통하여 제안하는 방법을 평가하고, 5장에서 결론을 맺는다.

2. 관련 연구

2.1 군집화(Clustering)

군집화는 계층적 군집화(hierarchical clustering)와 분할 군집화(partitional clustering)로 구분된다. 계층적 군집화는 개별 데이터에서 시작하여 가까운 데이터들을 묶음으로써 군집의 계층을 구성하는 반면, 분할 군집화는 군집의 중심을 결정하고, 각 데이터와 군집의 중심 간의 거리를 계산하여 가장 가까운 군집에 데이터를 할당하는 방법이다. 분할 군집화의 대표적인 방법은 k-means와 가우시안 혼합 모델(Gaussian Mixture Models, GMM)이다[6]. 이 방법들은 빠르며, 여러 문제에 적용할 수 있다는 장점이 있지만, 거리 계산 척도가 데이터 공간에 한정되어 있어서 입력의 차원이 높을 때에 비효율적이라는 단점이 있다.

다차원 입력 공간에서 k-means를 수행하기 위한 다양한 시도가 있었다. Ye등은 k-means를 수행한 후 군집간의 거리를 최대화하기 위하여 저차원으로 사상하는 방법을 사용하였다[7]. Luxburg는 유연한 거리 계산 척도와 높은 성능을 나타내는 스펙트럴 군집화(spectral clustering)를 제안하였다[8]. 일반적인 스펙트럴 군집화 알고리즘들은 라플라시안(Laplacian) 매트릭스를 계산하기 때문에 데이터 수의 제곱 이상의 계산 복잡도로 데이터수가 많아질수록 메모리와 계산 시간이 증가의 부담이 커진다.

데이터 시각화와 차원 축소에서 데이터 분포와 군집 내에서의 분포 사이의 쿨백-라이블러 발산을 최소화하는 방법이 사용되고 있다[9]. Maaten이 제안한 t-SNE는 심층 신경망을 사용하여 파라미터의 최적화를 수행하였으며, 계산의 복잡도는 $O(n^2)$이다. Xie 등은 분할 군집화 알고리즘에 쿨백-라이블러 발산을 적용하여, 군집 할당과 특징추출의 성능을 점진적으로 증가시키는 심층 임베디드 군집화(Deep Embedded Clustering, DEC) 알고리즘을 제안하였다[5]. 분할 군집화 알고리즘으로 k-means를 사용하였고, 계산의 복잡도는 O(nk)이며, k는 군집의 수이다.

Yang등은 k-means 기반의 심층 군집 네트워크(Deep Clustering Network, DCN)를 제안하였다[10]. 오토 인코더의 오류(Loss)와 k-means의 오류를 수학적으로 결합하여 우수한 성능을 보였다. Yang 등은 컨볼루션 신경망을 표현 학습에 사용하고, 계층적

군집화를 적용한 JULE(Joint Unsupervised Learning of Deep Representations and Image Clusters)을 제안하였다[2]. 학습할 때 군집 오류만을 사용하고 학습과 군집화의 수에 차등을 두어 이미지 데이터에 대한 군집화에서 좋은 성능을 보였다.

2.2 오토 인코더(Autoencoder)

오토 인코더는 인공 신경망(Artificial Neural Network)의 한 종류로 비지도 학습(Unsupervised Learning)으로 데이터의 표현(Representation)을 학습하는 방법이다. 오토 인코더는 입력 데이터를 입력 데이터의 차원보다 작은 코드로 압축하는 인코더(Encoder)와 압축된 코드를 원본 데이터로 복원하는 디코더(Decoder)로 구성된다. 오토 인코더는 네트워크의 학습목표로 입력 데이터가 출력인 자가 학습(Self Learning)이며, 입력에 대한 정답이 존재하지 않는 비교사 학습이다. 1개의 은닉층(Hidden Layer)을 가지는 가장 간단한 오토 인코더의 구성은 (Fig. 1)과 같다.

입력은 x이고, 목표 출력은 입력과 같기 때문에 $x = x'$가 된다. 은닉층의 값인 인코더의 출력 z는

$$z = f(\theta) = S(Wx + b) \tag{1}$$

로 정의 된다. W는 가중치 행렬이며, b는 바이어스 벡터이다. 인코더의 출력 z는 잠재 표현 (Latent Representation)이며 표현 코드(Representation Code)라고도 불린다. z는 디코더에 의해 원형 입력인 $x = x'$를 다시 출력하며, 이 때 수식은

$$x' = g(\theta) = S'(W'z + b') \tag{2}$$

로 정의된다. S와 S'은 활성 함수(Activation Function)이다. 손실 함수 L_r은

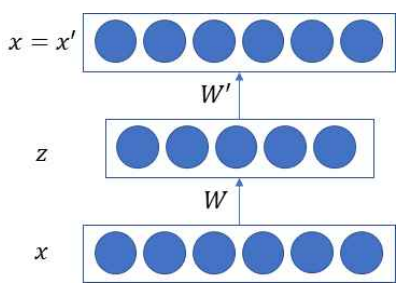


Fig. 1. The architecture of Autoencoder with 1 hidden layer.

$$L_r = \sum_{i=1}^{N} \| x_i - x_i' \|^2 \tag{3}$$

으로 정의된다. 입력 데이터 x_i와 오토 인코더에 의해 재생성된 출력 x_i'사이의 거리이며, 일반적으로 평균 제곱 오차(means squared error)를 사용한다. 오토 인코더의 학습은 오류 역전파(Error Backpropagation)와 통계적 경사 하강법(Stochastic Gradient Descent, SGD)이 사용된다.

계층적 오토 인코더는 오토 인코더를 층을 쌓아서 구성한 것이다. 오토 인코더는 오류 역전파를 이용하여 학습하기 때문에 은닉층이 많아지거나, 노드의 수가 많아질수록 가중치를 계산하는 시간이 오래 걸리고, 지역 최소(Local Minima)에 빠질 가능성이 증가한다. 또한, 계속 작은 값들이 수정되는 과정에서 가중치의 오차의 기울기가 사라지는(Vanishing Gradient) 문제가 발생할 수 있다. 이 문제를 해결한 것이 계층적 오토 인코더(Stacked Autoencoder, SAE)이다[11]. 계층적 오토 인코더는 모든 층을 한 번에 학습하는 것이 아니라, 심층신뢰신경망의 구조와 같이 층별로 학습 한 후 쌓아가는 방식으로 학습을 수행한다. 계층적 오토 인코더의 구성은 (Fig. 2)와 같다.

1 단계(phase 1)에서는 1개의 은닉층으로 오토 인코더를 학습 한 후, 2단계에서는 학습된 은닉 층을 입력으로 하는 1개의 은닉층을 가지는 오토 인코더를 구성한 후 학습한다. 2단계에서 학습된 오토 인코더를 1단계의 은닉층을 대치하여 입력 데이터를 사용한 3개의 은닉층을 가지는 오토 인코더를 구성한 후 파인 튜닝(Fine Tuning)을 수행한다. 이 과정을 목표한 은닉층의 수가 될 때까지 반복한다.

3. 제안하는 방법

3.1 전체적인 시스템의 구성

제안하는 시스템의 전체적인 구성은 (Fig. 3)과 같다. 표현 학습으로는 오토 인코더를 사용하며, 군집화에는 k-means를 사용한다. 오토 인코더를 이용하여 학습 한 후 인코더의 출력인 표현 벡터(Latent Representation)를 입력으로 사용하여 k-means를 학습한다. k-means의 학습이 이루어지면, 오토 인코더를 재학습하는데 이때 사용하는 손실 함수는 k-means의 손실 함수와 오토 인코더의 손실 함수를 결합하여 사용한다. 이와 같은 오토 인코더의 학습과

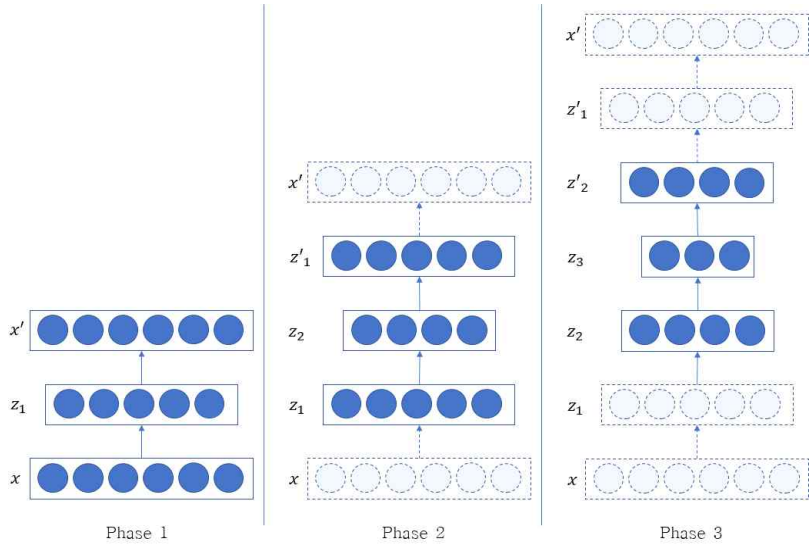


Fig. 2. The Architecture of Stacked Autoencoder.

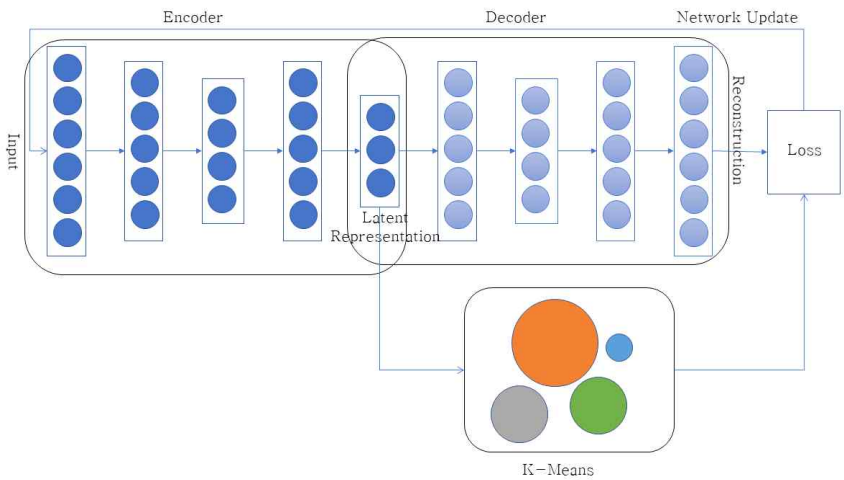


Fig. 3. The structure of the proposed method.

k-means의 학습을 학습 결과가 안정화될 때까지 반복하여 진행한다.

3.2 표현 학습(Representation Learning)

표현 학습의 대표적인 방법인 심층신뢰신경망은 이미지데이터에 특화되어 있다. 범용적으로 적용하기 위해서 계층적 오토 인코더(Stacked Autoencoder, SAE)를 적용하였다. 군집화는 분할 군집 알고리즘인 k-means로 하였다. 계층적 군집화는 데이터가 많아지면, 성능이 떨어지기 때문에 분할 군집 알고리즘을 적용하였다.

네트워크의 학습과정은 다음과 같다. 먼저, 계층적 오토 인코더가 수식 (3)의 계층적 오토 인코더의 손실 함수 L_r에 의해서 학습된다. 이는 사전 학습(pre-training) 단계로 계층적 오토 인코더가 학습되어 입력 데이터에 대한 높은 품질의 표현 코드를 얻을 수 있다. 계층적 오토 인코더가 충분히 학습되고 난 다음에 k-means 군집화 알고리즘에 의해서 입력 데이터의 표현 코드인 z에 소속 군집을 라벨(label)로 할당한다. 군집화 알고리즘에서 사용하는 손실 함수는

데이터를 군집에 할당했을 때의 k-means 손실(k-means Loss, L_k)과 소프트 할당(soft assignments) 한 q_i와 보조 분포 p_i사이의 쿨백-라이블러 발산으로 계산한 손실 함수(L_d)를 결합하여 계산한다. 이 후 계층적 오토 인코더를 다시 학습할 때 사용하는 손실 함수는 다음과 같다.

$$L = L_r + \beta L_k + \gamma L_d \tag{4}$$

L_r은 계층적 오토 인코더의 손실 함수이고, L_k는 k-means의 손실 함수이며, L_d는 쿨백-라이블러 발산 함수이다. $\beta > 0$, $\gamma > 0$은 공간의 왜곡 정도를 제어하는 계수이다. 쿨백-라이블러 발산은 데이터가 특정 군집에 속할 확률을 계산한 것이고, k-means 손실은 데이터와 속한 군집 중심 간의 유사도를 계산한 것으로 두 손실을 결합하여 최소화하면 군집의 형성의 안정성이 높아진다. 이러한 학습 단계가 완료되면 데이터가 군집이 잘 이루어질 수 있는 표현 코드로 표상(mapping) 된다.

3.3 군집화(Clustering)와 군집 손실(Clustering Loss)

군집화 알고리즘은 k-means 알고리즘을 활용하였다. 일반적으로 k-means에서 데이터와 군집간의 거리는 다음과 같은 유클리디안 거리를 사용한다.

$$D_{ij} = \sqrt{\sum_{l=1}^{n} (x_{il} - \mu_{jl})^2} \tag{5}$$

여기서, x_i는 i번째 데이터이고, μ_j는 j번째 군집의 중심이며, n은 데이터를 구성하는 특징 벡터의 노드 수이다. 유클리디안 거리는 방향성이 고려되지 않고, 거리만을 계산한다는 단점이 있다. 유사도를 계산하는 다른 방법 중의 하나로 코사인 유사도(Cosine Similarity)가 있다. 코사인 유사도는 두 벡터 사이 각도의 코사인 값을 이용하여 벡터간의 유사한 정도를 계산한다. 코사인 유사도는 방향의 유사도를 판단하는 목적으로 사용되며, 벡터의 방향이 완전히 같으면 1, 90도의 각을 이루면 0, 180도의 각을 이루면 -1의 값을 가진다. 이 때 벡터의 크기는 코사인 유사도에 영향을 미치지 않는다. 코사인 유사도는

$$Sc_{ij} = \cos(\theta) = \frac{x_i \mu_j}{\| x_i \| \| \mu_j \|} = \sum_{l=1}^{n} \frac{x_{il} \mu_{jl}}{\sqrt{\sum_{l=1}^{n} x_{il}^2} \sqrt{\sum_{l=1}^{n} \mu x_{jl}^2}} \tag{6}$$

으로 계산된다.

코사인 유사도는 -1 과 1 사이의 값을 가지며, 1이 가장 유사한 것, -1이 가장 유사하지 않은 것이다. 유클리디안 거리는 0이 가장 가까운 것이기 때문에 척도를 일치시키기 위하여 코사인 유사도는 $\frac{1 - Sc_{ij}}{2}$로 0 과 1 사이의 값을 가지도록 하여, 0에 가까울수록 유사한 것, 1에 가까울수록 유사하지 않도록 표현하였다. 또한, 유클리디안 거리는 최댓값이 존재하지 않기 때문에 코사인 유사도와 값의 범위를 동일하게 하기 하여 최댓값으로 나누어 0 과 1 사이의 값으로 나타내었다. α는 계수로 0 과 1 사이의 값을 가지며, 0으로 설정하면 유클리디안 거리만이 계산된다. 본 연구에서 유사도는 식(7)과 같이 계산하였다.

$$S_{ij} = \frac{D_{ij}}{\max(D)} + \alpha \frac{(1 - Sc_{ij})}{2} \tag{7}$$

군집 손실은 쿨백-라이블러 발산과 k-means 손실을 결합하여 사용한다. 군집화는 군집 중심을 학습 가능한 가중치로 유지하면서 계층적 오토 인코더의 표현 벡터인 z_i를 소프트 라벨인 q_i 로 매핑한다[13].

$$q_{ij} = \frac{(1 + \| z_i - \mu_j \|^2)^{-1}}{\sum_{j=1}^{k} (1 + \| z_i - \mu_j \|^2)^{-1}} \tag{8}$$

여기서, q_{ij}는 q_i의 j번째 값을 의미하는데, 이것은 데이터의 표현 벡터 z_i가 군집 μ_j에 속할 확률을 의미한다. 군집의 손실 함수는 쿨백-라이블러 발산에 의해

$$L_d = KL(P \| Q) = \sum_{i=1}^{N} \sum_{j=1}^{k} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{9}$$

과 같이 정의된다. 여기서 P는 목표 분포이며, N은 입력 데이터의 수이고, p_{ij}는

$$p_{ij} = \frac{q_{ij}^2 / \sum_{i=1}^{N} q_{ij}}{\sum_{j=1}^{k} (q_{ij}^2 / \sum_{i=1}^{N} q_{ij})} \tag{10}$$

와 같이 정의된다. K-means 손실은

$$L_k = \sum_{i=1}^{N} \sum_{j=1}^{K} s_{ij} S_{ij}(z_i, \mu_j) \tag{11}$$

으로 정의된다. S_{ij}는 수식 (7)에서 정의한 표현 벡터 z_i와 군집 μ_j 사이의 유사도이며, s_{ij}는 표현 벡터 z_i가 군집 μ_j에 할당되는지를 확인하는 0과 1을 가지는 값이다.

3.4 재학습의 손실 함수

계층적 오토 인코더가 학습되고, 계층적 오토 인코더의 표현 벡터를 통해 군집화가 이루어지고 난 후, 이 결과를 이용하여 계층적 오토 인코더는 다시 파인 튜닝을 수행하는 재학습을 하며, 이 때 사용하는 손실 함수는 수식 (4)와 같다. 오토 인코더는 데이터의 지역 구조(local structure)를 보존할 수 있기 때문에, 손실 함수에 군집 손실과 k-means 손실을 사용하여 미세하게 조정하는 것은 오토 인코더의 구조에 손상을 발생시키지 않는다. 따라서 계수 '베타'와 '감마'는 1보다 작을수록 좋으며, 실험을 통해 0.1로 고정한다.

4. 실험 및 결과

4.1 실험 데이터 집합 및 실험 환경

실험을 위한 데이터로 이미지 데이터 집합과 비이미지 데이터로 문서 데이터 집합을 사용하였다. 이미지 데이터는 70,000개의 필기체 숫자로 구성된 MNIST 데이터 집합이다[14]. MNIST-full은 MNIST 학습 집합 전체를 의미하고, MNIST-test는 MNIST 테스트 집합으로 10,000개의 이미지를 가지고 있다. 문서 데이터는 REUTERS로 810,000개의 Reuters의 영어 신문 기사로 이루어져 있다[15]. 각 기사는 여러 개의 분야가 할당되어 있는데 이 중 분류 개수가 많은 회사/산업 (corporate/industrial), 정부/사회(government/social), 시장(markets), 경제(economics) 분야의 문서 685,071개를 데이터로 사용하였다. 구현은 R과 케라스(Keras) 그리고, 케라스 인터페이스 라이브러리로 하였다.

제안하는 알고리즘의 성능을 k-means, 오토 인코더 + k-means와 비교하였다. 오토 인코더 + k-means는 오토 인코더를 학습한 후 추출한 표현 벡터에 k-means를 적용한 것으로 재학습 과정은 없는 방법이다. 오토 인코더 + k-means와 제안하는 방법의 오토 인코더의 구조는 d-500-500-2000-10으로, 디코더는 인코더의 역의 구조인 10-2000-500-500-d의 다층 퍼셉트론 (Multi Layer Perceptron, MLP)을 구성하였다. 여기서, d는 입력 데이터의 차원이다. 활성 함수는 ReLU 함수를 사용하였고, 오토 인코더는 400 에포크(epochs)로 학습률 0.01, 모멘텀 0.9로 학습하였다.

군집화의 성능 평가는 군집 정확도(clustering accuracy, ACC)로 하였다[5].

$$ACC = \max \frac{\sum_{i=1}^{n} 1\{l_i = m(\mu_i)\}}{n} \quad (12)$$

여기서, l_i는 검증 값이고, μ_i는 알고리즘에 의해 생성된 할당된 군집이며, m은 군집과 검증 값 간의 일대일 매핑 범위이다. 비지도 학습으로 데이터에 군집을 할당한 것과 검증 값 사이에 가장 일치하는 것을 찾는다.

4.2 실험 결과

Table 1은 모델의 학습에 사용하는 손실 함수에 따른 군집의 정확도의 성능을 비교한 것이다. AE_kmeans는 오토 인코더를 수행한 후 k-means를 한 번 수행한 것이다. AE_kmeans2는 모델 구조는 제안하는 방법과 동일 계층적 오토 인코더 손실 함수 (수식 (3), L_r)와 군집 손실로 쿨백-라이블러 발산 (수식 (8), L_d)을 결합한 손실함수로 계산하였다. 또한, AE_kmeans3는 모델의 구조는 동일하지만, 손실 함수는 계층적 오토 인코더 손실 (L_r)과 k-means 손실 (수식 (10), L_k)를 결합한 함수이고, k-means의 거리는

Table 1. Comparison of clustering accuracy on loss function (ACC, %)

	MNIST-full	MNIST-test	REUTERS
AE_kmeans	78.09	66.08	61.84
AE_kmeans2	84.27	81.14	74.31
AE_kmeans3	84.78	81.32	76.08
AE_kmeans4	83.96	79.51	75.60
Proposed	87.21	81.45	78.62

Fig. 4. Clustering examples of MNIST.

유클리디안 거리로 계산한 것이다. AE_kmeans4는 AE_kmeans3와 동일하고, k-means의 거리를 계산할 때 코사인 유사도를 사용한 것이다. k-means 손실함수를 같이 계산한 AE_kmeans3가 AE_kmeans2보다 최소 0.18%에서 최대 1.77% 정도 정확도가 향상되었으며, 평균 0.82%정도 정확도가 향상되었다. 코사인 유사도만 계산한 AE_kmeans4보다 군집 손실과 k-means 손실을 같이 계산한 AE_kmeans3의 정확도는 최소 0.48%에서 최대 1.81% 정도 향상되었으며, 평균 1% 정도 정확도가 향상되었다. 표현 벡터가 다차원 벡터이지만, 입력 데이터의 차원을 충분히 줄였기 때문에 거리를 계산한 유클리디안 거리가 각도를 계산한 코사인 유사도보다 성능이 우수하였고, 거리와 각도를 같이 고려한 유클리디안 거리와 코사인 유사도를 같이 사용하는 제안하는 방법은 다른 알고리즘과 비교하여 최소 0.13%에서 최대 4.31% 정도 정확도가 향상되었다.

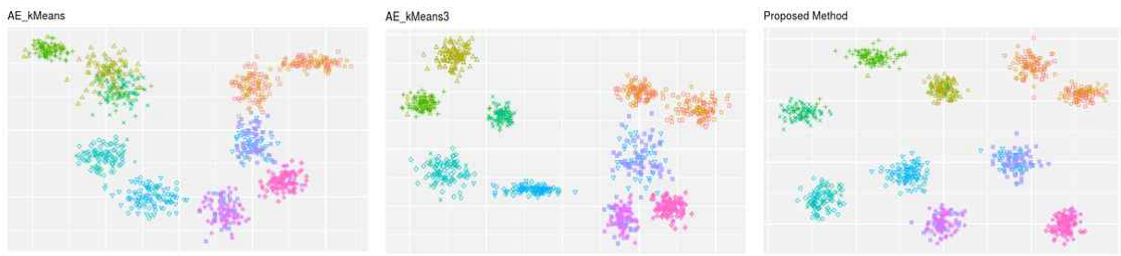
Fig. 4는 MNIST 데이터 집합에서 각 군집에 속할 가능성이 가장 높은 10개의 이미지를 각각의 군집으로 모아서 표현한 것이다. 각 행은 군집이며, 이미지들은 군집 중심까지의 유사도에 따라 정렬되어 있다. 4와 9의 경우에는 서로 섞여 있으므로 군집화 알고리즘으로는 잘 구별할 수 없지만, 다른 숫자의 경우에는 잘 구분하였다.

Fig. 5는 제안하는 방법이 군집의 구조를 유지하고 있는지를 실험한 결과이다. MNIST-full 데이터 집합에서 1,000개를 선택해서 t-SNE[13] 시각화 기법으로 결과를 표현하였다. AE_kmeans는 군집 2개가 거의 붙어서 9개의 군집만 있는 것처럼 표현된다. AE_kmeans3는 10개의 군집이 생성되지만, 오른쪽에 있는 군집들의 데이터의 분포가 퍼져있고, 군집간의 거리가 가까워서 명확하게 군집구분이 되지 않았으며, 제안하는 방법은 군집 간의 구분이 우수하였다.

5. 결 론

본 논문에서는 계층적 오토 인코더와 군집화 알고리즘을 결합하여 대용량 및 다차원 데이터에 대한 군집 알고리즘을 제안하였다. 계층적 오토 인코더로 입력 데이터의 형태에 제한이 없이 표현 벡터를 생성할 수 있고, 다차원 데이터에 대한 군집화를 위해서 유클리디안 거리와 코사인 유사도를 함께 계산하여 군집과 데이터 사이의 거리를 계산하였다. 네트워크는 계층적 오토 인코더의 손실 함수와 군집화 알고리즘의 손실 함수를 결합하여 재학습을 하였다. 실험을 통해 이미지 데이터와 문서 데이터 등 다양한 종류의 데이터에 대해 좋은 성능을 보였다.

군집화 성능을 높이기 위해서는 대량의 데이터와 데이터 속성에 대한 처리 방법뿐만 아니라 분산 군집화와 최적의 군집의 개수를 결정하는 방법도 중요한 연구 분야이다. 계층적 오토 인코더 나 k-means, 컨볼루션 신경망은 모두 학습할 데이터가 같은 시스템에 있을 때 최적의 성능을 나타낸다. 하지만, 데이터의 양이 점점 많아지는 빅데이터 시대에 한 컴퓨터에서 처리 가능한 용량 만큼만의 학습 데이터가 주어질 수는 없다. 따라서 분산 컴퓨팅 환경에서 군집화가 수행되는 분산 군집화와 심층 군집 네트워크를 조화시키는 방안에 대한 연구가 필요하다. 또한, 최적의 군집 개수를 결정하는 방법도 필요하다. 기존 연구에서는 보통 군집의 개수를 늘려가면서 실험하여 군집의 성능 평가 지표가 최적일 때를 최적의 군집 개수로 결정하였다. 하지만, 모든 경우를 실험하는 방법은 계산 시간이 증가하고, 초기 군집 중심에 영향을 받기 때문에 학습 시간의 증가는 최소화 하고 동적으로 군집의 개수를 결정하는 방법에 대한 연구



(a) (b) (c)

Fig. 5. Visualization of clustering results. (a)AE_kmeans, (b)AE_kmeans3 and (c)Proposed Method.

가 필요하다.

REFERENCE

[1] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, "A Deep Semi-nmf Model for Learning Hidden Representations," *Proceeding of the 31st International Conference on Machine Learning*, Vol. 46, pp. 1692-1700, 2014.

[2] J. Yang, D. Parikh, and D. Batra, "Joint Unsupervised Learning of Deep Representations and Image Clusters," *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147-5156, 2016.

[3] F. Li, H. Qiao, B. Zhang, and X. Xi, "Discriminatively Boosted Image Clustering with Fully Convolutional Auto-encoders," *Pattern Recognition*, Vol. 83, pp. 161-173, 2018.

[4] H.J. Lee, "Hierarchical Deep Belief Network for Activity Recognition Using Smartphone Sensor," *Journal of Korea Multimedia Society*, Vol. 20, No. 8, pp. 1421-1429, 2017.

[5] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," *Proceeding of the 33rd International Conference on Machine Learning*, Vol. 48, pp. 478-487, 2016.

[6] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2016.

[7] J. Ye, Z. Zhao, and M. Wu, "Discriminative K-means for Clustering," *Proceeding of the 21st Annual Conference on Neural Information Processing Systems*, arXiv:1306.2102, 2009.

[8] U.V. Luxburg, "A Tutorial on Spectral Clustering," *Statistics and Computing*, Vol. 17, No. 4, pp. 395-416, 2007.

[9] L. Van Der Maaten, "Accelerating t-SNE Using Tree-based Algorithms," *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 3221-3245, 2014.

[10] B. Yang, X. Fu, N.D. Sidiropoulos, and M. Hong, "Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering," *Proceeding of the 34th International Conference on Machine Learning*, arXiv:1610.04794, 2017.

[11] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World,* Basic Books, New York, 2015.

[12] G. Xifeng, L. Xinwang, Z. En, and Y. Jianping, "Deep Clustering with Convolutional Autoencoders," *Lecture Notes in Computer Science*, Vol. 10635, pp. 373-382, 2017.

[13] L.V.D. Maaten and G. Hinton, "Visualizing Data Using Accelerating t-SNE Using Tree-based Algorithms,," *The Journal of Machine Learning Research*, Vol. 9, pp. 2579-2605, 2008.

[14] Y. LeCun, C. Cortes, and C.J. Burges, http://yann.lecun.com/exdb/mnist/ (accessed Mar., 20, 2018).

[15] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *The Journal of Machine Learning Research*, Vol. 5, pp. 361-397, 2004.

이 현 진

1996년 순천향대학교 전산학과 공학사
1998년 연세대학교 대학원 컴퓨터과학과 공학석사
2002년 연세대학교 대학원 컴퓨터과학과 공학박사

2003년~현재 숭실사이버대학교 ICT공학부 부교수
관심분야 : 이러닝, 머신러닝, 빅데이터 처리