IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received May 12, 2018, accepted July 2, 2018, date of publication July 17, 2018, date of current version August 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2855437

A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture

ERXUE MIN, XIFENG GUO, QIANG LIU, (Member, IEEE), GEN ZHANG, JIANJING CUI, AND JUN LONG
College of Computer, National University of Defense Technology, Changsha 410073, China

Corresponding authors: Erxue Min (minerxue12@nudt.edu.cn) and Qiang Liu (qiangliu06@nudt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 60970034, Grant 61105050, Grant 61702539, and Grant 6097003.

ABSTRACT Clustering is a fundamental problem in many data-driven application domains, and clustering performance highly depends on the quality of data representation. Hence, linear or non-linear feature transformations have been extensively used to learn a better data representation for clustering. In recent years, a lot of works focused on using deep neural networks to learn a clustering-friendly representation, resulting in a significant increase of clustering performance. In this paper, we give a systematic survey of clustering with deep learning in views of architecture. Specifically, we first introduce the preliminary knowledge for better understanding of this field. Then, a taxonomy of clustering with deep learning is proposed and some representative methods are introduced. Finally, we propose some interesting future opportunities of clustering with deep learning and give some conclusion remarks.

INDEX TERMS Clustering, deep learning, data representation, network architecture.

I. INTRODUCTION

Data clustering is a basic problem in many areas, such as machine learning, pattern recognition, computer vision, data compression. The goal of clustering is to categorize similar data into one cluster based on some similarity measures (e.g., Euclidean distance). Although a large number of data clustering methods have been proposed [1]–[5], conventional clustering methods usually have poor performance on high-dimensional data, due to the inefficiency of similarity measures used in these methods. Furthermore, these methods generally suffer from high computational complexity on large-scale datasets. For this reason, dimensionality reduction and feature transformation methods have been extensively studied to map the raw data into a new feature space, where the generated data are easier to be separated by existing classifiers. Generally speaking, existing data transformation methods include linear transformation like Principal component analysis (PCA) [6] and non-linear transformation such as kernel methods [7] and spectral methods [8]. Nevertheless, a highly complex latent structure of data is still challenging the effectiveness of existing clustering methods. Owing to the development of deep learning [9], deep neural networks (DNNs) can be used to transform the data into more clustering-friendly representations due to its inherent property of highly non-linear transformation. For the simplicity of description, we call clustering methods with deep learning as *deep clustering*[1] in this paper.

Basically, previous work mainly focuses on feature transformation or clustering independently. Data are usually mapped into a feature space and then directly fed into a clustering algorithm. In recent years, deep embedding clustering (DEC) [11] was proposed and followed by other novel methods [12]–[18], making deep clustering become a popular research field. Recently, an overview of deep clustering was proposed in [19] to review most remarkable algorithms in this field. Specifically, it presented some key elements of deep clustering and introduce related methods. However, this paper mainly focuses on methods based on autoencoder [20], and it was incapable of generalizing many other important methods, e.g., clustering based on deep generative model. What is worse, some up-to-date progress is also missing. Therefore, it is meaningful to conduct a more systematic survey covering the advanced methods in deep clustering.

Classical clustering methods are usually categorized as partition-based methods [21], density-based methods [22],

[1] The concept of "deep clustering" was firstly introduced in a deep learning framework for acoustic source separation [10], and gradually became popular among general clustering tasks.

hierarchical methods [23] and so on. However, since the essence of deep clustering is to learning a clustering-oriented representation, it is not suitable to classify methods according to the clustering loss, instead, we should focus on the network architecture used for clustering. In this paper, we make a survey of deep clustering from the perspective of network architecture. The first category uses the autoencoder (AE) to obtain a feasible feature space. An autoencoder network provides a non-linear mapping function through learning an encoder and a decoder, where the encoder is a mapping function to be trained, and the decoder is required to be capable to reconstruct the original data from those features generated by the encoder. The second category is based on feed-forward networks trained only by specific clustering loss, thus we refer to this type of DNN as Clustering DNN (CDNN). The network architecture of this category can be very deep and networks pre-trained on large-scale image datasets can further boost its clustering performance. The third and fourth categories are based on Generative Adversarial Network (GAN) [24] and Variational Autoencoder (VAE) [25] respectively, which are the most popular deep generative models in recent years. They can not only perform clustering task, but also can generate new samples from the obtained clusters. To be more detailed, we present a taxonomy of existing deep clustering methods based on the network architecture. We introduce the representative deep clustering methods and compare the advantages and disadvantages of different architectures and methods. Finally, some directions are suggested for future development of this field.

The rest of this paper is organized as follows: Section II reviews some preliminaries of deep clustering. Section III presents a taxonomy of existing deep clustering algorithms and introduces some representative methods. Finally, Section IV provides some notable trends of deep clustering and gives conclusion remarks.

II. PRELIMINARIES

In this section, we introduce some preliminary knowledge of deep clustering. It includes the related network architectures for feature representation, loss functions of standard clustering methods, and the performance evaluation metrics for deep clustering.

A. NEURAL NETWORK ARCHITECTURE FOR DEEP CLUSTERING

In this part, we introduce some neural network architectures, which have been extensively used to transform inputs to a new feature representation.

1) FEEDFORWARD FULLY-CONNECTED NEURAL NETWORK

A fully-connected network (FCN) consists of multiple layers of neurons, each neuron is connected to every neuron in the previous layer, and each connection has its own weight. The FCN is also known as multi-layer perceptron (MLP). It is a totally general purpose connection pattern and makes no assumptions about the features in the data. It is usually used in supervised learning when labels are provided. However, for clustering, a good initialization of parameters of network is necessary because a naive FC network tends to obtain a trivial solution when all data points are simply mapped to tight clusters, which will lead to a small value of clustering loss, but be far from being desired [13].

2) FEEDFORWARD CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks (CNNs) [26] were inspired by biological process, in which the connectivity pattern between neurons is inspired by the organization of the animal visual cortex. Likewise, each neuron in a convolutional layer is only connected to a few nearby neurons in the previous layer, and the same set of weights is used for every neuron. It is widely applied to image datasets when locality and shift-invariance of feature extraction are required. It can be trained with a specific clustering loss directly without any requirements on initialization, and a good initialization would significantly boost the clustering performance. To the best of our knowledge, no theoretical explanation is given in any existing papers, but extensive work shows its feasibility for clustering.

3) DEEP BELIEF NETWORK

Deep Belief Networks (DBNs) [27] are generative graphical models which learn to extract a deep hierarchical representation of the input data. A DBN is composed of several stacked Restricted Boltzmann machines (RBMs) [28]. The greedy layer-wise unsupervised training is applied to DBNs with RBMs as the building blocks for each layer. Then, all (or part) of the parameters of DBN are fine-tuned with respect to certain criterion (loss function), e.g., a proxy for the DBN log-likelihood, a supervised training criterion, or a clustering loss.

4) AUTOENCODER

Autoencoder (AE) is one of the most significant algorithms in unsupervised representation learning. It is a powerful method to train a mapping function, which ensures the minimum reconstruction error between coder layer and data layer. Since the hidden layer usually has smaller dimensionality than the data layer, it can help find the most salient features of data. Although autoencoder is mostly applied to find a better initialization for parameters in supervised learning, it is also natural to combine it with unsupervised clustering. More details and formulations will be introduced in Section III-A.

5) GAN & VAE

Generative Adversarial Network (GAN) and Variational Autoencoder (VAE) are the most powerful frameworks for deep generative learning. GAN aims to achieve an equilibrium between a generator and a discriminator, while VAE attempts to maximizing a lower bound of the data log-likelihood. A series of model extensions have been developed for both GAN and VAE. Moreover, they have also been applied to handle clustering tasks. The details of the two models will be elaborated in Section III-C and Section III-D, respectively.

B. LOSS FUNCTIONS RELATED TO CLUSTERING

This part introduces some clustering loss functions, which guides the networks to learn clustering-friendly representations. Generally, there are two kinds of clustering loss. We name them as *principal clustering loss* and *auxiliary clustering loss*.

- **Principal Clustering Loss**: This category of clustering loss functions contain the cluster centroids and cluster assignments of samples. In other words, after the training of network guided by the clustering loss, the clusters can be obtained directly. It includes k-means loss [13], cluster assignment hardening loss [11], agglomerative clustering loss [29], nonparametric maximum margin clustering [30] and so on.
- **Auxiliary Clustering Loss**: The second category solely plays the role of guiding the network to learn a more feasible representation for clustering, but cannot output clusters straightforwardly. It means deep clustering methods with merely auxiliary clustering loss require to run a clustering method after the training of network to obtain the clusters. There are many auxiliary clustering losses used in deep clustering, such as locality-preserving loss [31], which enforces the network to preserve the local property of data embedding; group sparsity loss [31], which exploits block diagonal similarity matrix for representation learning; sparse subspace clustering loss [32], which aims at learning a sparse code of data.

C. PERFORMANCE EVALUATION METRICS FOR DEEP CLUSTERING

Two standard unsupervised evaluation metrics are extensively used in many deep clustering papers. For all algorithms, the number of clusters are set to the number of ground-truth categories. The first metric is *unsupervised clustering accuracy (ACC)*:

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

where y_i is the ground-truth label, c_i is the cluster assignment generated by the algorithm, and m is a mapping function which ranges over all possible one-to-one mappings between assignments and labels. It is obvious that this metric finds the best matching between cluster assignments from a clustering method and the ground truth. The optimal mapping function can be efficiently computed by Hungarian algorithm [33].

The second one is *Normalized Mutual Information (NMI)* [34]:

$$NMI(Y, C) = \frac{I(Y, C)}{\frac{1}{2}[H(Y) + H(C)]}$$

where Y denotes the ground-truth labels, C denotes the clusters labels, I is the mutual information metric and H is entropy.

III. TAXONOMY OF DEEP CLUSTERING

Deep clustering is a family of clustering methods that adopt deep neural networks to learn clustering-friendly representations. The loss function (optimizing objective) of deep clustering methods are typically composed of two parts: network loss L_n and clustering loss L_c, thus the loss function can be formulated as follows:

$$L = \lambda L_n + (1 - \lambda) L_c \tag{1}$$

where $\lambda \in [0, 1]$ is a hype-parameter to balance L_n and L_c. The network loss L_n is used to learn feasible features and avoid trivial solutions, and the clustering loss L_c encourages the feature points to form groups or become more discriminative. The network loss can be the reconstruction loss of an autoencoder (AE), the variational loss of a variational encoder (VAE) or the adversarial loss of a generative adversarial network (GAN). As described in Section II-B, the clustering loss can be k-means loss, agglomerative clustering loss, locality-preserving loss and so on. For deep clustering methods based on AE network, the network loss is essential. But some other work designs a specific clustering loss to guide the optimization of networks, in which case the network loss can be removed. As mentioned in Section I, we refer this type of networks trained only by L_c as clustering DNN (CDNN). For GAN-based or VAE-based deep clustering, the network loss and the clustering loss are usually incorporated together. In this section, from the perspective of DNN architecture, we divide deep clustering algorithms into four categories: AE-based, CDNN-based, VAE-based, and GAN-based deep clustering. Characteristics of each category are revealed and related algorithms are introduced. Some notations frequently used in the paper and their meanings are presented in Table 1. The components of representative algorithms are illutrated in Table 2 and their contributions are described briefly in Table 3.

A. AE-BASED DEEP CLUSTERING

Autoencoder is a kind of neural network designed for unsupervised data representation. It aims at minimizing the reconstruction loss. An autoencoder may be viewed as consisting of two parts: an encoder function $\boldsymbol{h} = f_\phi(\boldsymbol{x})$ which maps original data $\boldsymbol{x}$ into a latent representation $\boldsymbol{h}$, and a decoder that produces a reconstruction $\boldsymbol{r} = g_\theta(\boldsymbol{h})$. The reconstructed representation $\boldsymbol{r}$ is required to be as similar to $\boldsymbol{x}$ as possible. Note that both encoder and decoder can be constructed by fully-connected neural network or convolutional neural network. When the distance measure of the two variables is mean square error, given a set of data samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$, its optimizing objective is formulated as follows:

$$\min_{\phi,\theta} L_{rec} = \min \frac{1}{n} \sum_{i=1}^{n} \parallel \boldsymbol{x}_i - g_\theta(f_\phi(\boldsymbol{x}_i)) \parallel^2 . \tag{2}$$

where ϕ and θ denote the parameters of encoder and decoder respectively. Many variants of autoencoder have

TABLE 1. Notations and their meanings.

Notations	Meanings
L_n	the network loss
L_c	the clustering loss
L_{rec}	the reconstruction loss (a specific type of network loss)
λ	the hype-parameter to balance L_n and L_c
$\boldsymbol{x}$	the vector of an original data sample
$\boldsymbol{z}$	the vector of the embedding representation of $\boldsymbol{x}$, or the prior vector for GAN
c	the obtained class label of sample $\boldsymbol{x}$
i	the counter variable
$\|\| \cdot \|\|$	the 2-norm of a vector
$f_\phi(\cdot)$	the encoder part of the autoencoder
$g_\theta(\cdot)$	the decoder part of the autoencoder
$\mathbb{E}$	the expectation
$Cat(\cdot)$	the categorical distribution
$\mathcal{N}(\cdot)$	multivariate Gaussian distribution
$\mathcal{B}(\cdot)$	multivariate Bernoulli distrubution
$\boldsymbol{\mu}$	the mean of the Gaussian distribution
$\boldsymbol{\sigma}$	the variance of the Gaussian distribution
$G(\cdot)$	the generative network of GAN
$D(\cdot)$	the discriminative network of GAN

been proposed and applied to deep clustering. The performance of autoencoder can be improved from the following perspectives: (1)

1) **Architecture**: The original autoencoder is comprised of multiple layer perceptions. For the sake of handling data with spatial invariance, e.g., image data, convolutional and pooling layers can be used to construct a convolutional autoencoder (CAE).
2) **Robustness**: To avoid overfitting and to improve robustness, it is natural to add noise to the input. Denoising autoencoder [35] attempts to reconstruct $\boldsymbol{x}$ from $\tilde{\boldsymbol{x}}$, which is a corrupted version of $\boldsymbol{x}$ through some form of noise. Additionally, noise can also be added to the inputs of each layer [14].
3) **Restrictions on latent features**: Under-complete autoencoder constrains the dimension of latent coder $\boldsymbol{z}$ lower than that of input $\boldsymbol{x}$, enforcing the encoder to extract the most salient features from original space. Other restrictions can also be adopted, e.g., sparse autoencoder [36] imposes a sparsity constraint on latent coder to obtain a sparse representation.
4) **Reconstruction loss**: Commonly the reconstruction loss of an autoencoder consists of only the discrepancy between input and output layer, but the reconstruction losses of all layers can also be optimized jointly [14].

The optimizing objective of AE-based deep clustering is thus formulated as follows:

$$L = \lambda L_{rec} + (1 - \lambda)L_c \tag{3}$$

The reconstruction loss enforce the network to learn a feasible representation and avoid trivial solutions. The general architecture of AE-based deep clustering algorithms is illustrated in Figure 1, and some representative methods are introduced as follows:

- **Deep Clustering Network (DCN)**:
DCN [13] is one of the most remarkable methods in this field, which combines autoencoder with the k-means algorithm. In the first step, it pre-trains an autoencoder. Then, it jointly optimizes the reconstruction loss and k-means loss. Since k-means uses discrete cluster assignments, the method requires an alternative optimization algorithm. The objective of DCN is simple compared with other methods and the computational complexity is relatively low.
- **Deep Embedding Network (DEN)**:
DEN [31] proposes a deep embedding network to extract effective representations for clustering. It first utilizes a deep autoencoder to learn reduced representation from the raw data. Secondly, in order to preserve the local structure property of the original data, a locality-preserving constraint is applied. Furthermore, it also incorporates a group sparsity constraint to diagonalize the affinity of representations. Together with the reconstruction loss, the three losses are jointly optimized to fine-tune the network for a clustering-oriented representation. The locality-preserving and group sparsity constraints serve as the auxiliary clustering loss (see Section II-B), thus, as the last step, k-means is required to cluster the learned representations.
- **Deep Subspace Clustering Networks (DSC-Nets)**:
DSC-Nets [37] introduces a novel autoencoder architecture to learn an explicit non-linear mapping that is

TABLE 2. Comparison of algorithms based on network architecture and loss function.

Categories	Algorithms	Network Architecture	Network loss	Clustering loss	
				Principal	Auxiliary
AE	DCN	AE	reconstruction loss	k-means loss	N
	DEN	AE	reconstruction loss	N	1) locality-preserving constraint 2) group sparsity constraint
	DSC-Nets	CAE	reconstruction loss	N	self-expressiveness term
	DMC	AE	reconstruction loss	proximity penalty term	locality-preserving loss
	DEPICT	CAE (Denoising)	reconstruction loss	unsupervised cross entropy loss	N
	DCC	AE/CAE	reconstruction loss	robust continuous clustering loss	N
CDNN	DNC	RBM	N	nonparametric maximum margin clustering loss	N
	DEC	FCN	N	cluster assignment hardening loss	N
	DBC	CNN	N	cluster assignment hardening loss	N
	CCNN	CNN	N	k-means	N
	IMSAT	FCN	N	1) regularized information maximization, 2) self-augmented training loss	N
	JULE	CNN	N	agglomerative clustering	N
	DAC[1]	CNN	N	pairwise-classification loss	N
VAE	VaDE	VAE	variational lower bound on the marginal likelihood, with a GMM priori		
	GMVAE	VAE	variational lower bound on the marginal likelihood, with a GMM priori		
GAN	DAC[2]	Adversarial autoencoder	reconstruction loss	1) GMM likelihood, 2) adversarial objective	N
	CatGAN	GAN	adversarial objective with a multi-classes priori		
	InfoGAN	GAN	adversarial objective with a multi-classes priori		

[1] Deep Adaptive Clustering
[2] Deep Adversarial Clustering

TABLE 3. Main contributions of the representative algorithms.

Categories	Algorithms	Main contributions to clustering
AE	DCN	perform k-means clustering and feature learning simultaneously, simple but effective
	DEN	learn a clustering-friendly representation
	DSC-Nets	improve the classical subspace clustering by AE
	DMC	improve the classical multi-manifold clustering by AE
	DEPICT	computational efficient, robust, perform well on image datasets
	DCC	avoid alternative optimization, require no prior knowledge of cluster number
CDNN	DNC	improve the classical NMMC clustering by DBN
	DEC	the first well-known deep clustering method, making this field popular
	DBC	improve DEC using CNN
	CCNN	computational efficient, deal with large-scale image datasets
	IMSAT	introduce self-augment training to deep clustering
	JULE	perform well on image datasets, but have high computational and memory cost
	DAC	well-designed clustering loss, achieve the-state-of-art performance on several datasets
VAE	VaDE	combine VAE with clustering
	GMVAE	combine VAE with clustering
GAN	DAC	combine AAE with clusteirng
	CatGAN	combine GAN with clustering
	InfoGAN	learn disentangled representations

friendly to *subspace clustering* [38]. The key contribution is introducing a novel self-expressive layer, which is a fully connected layer without bias and non-linear activation and inserted to the junction between the encoder and the decoder. This layer aims at encoding the self-expressiveness property [39] [40] of data drawn from a union of subspaces. Mathematically, its optimizing objective is a subspace clustering loss combined

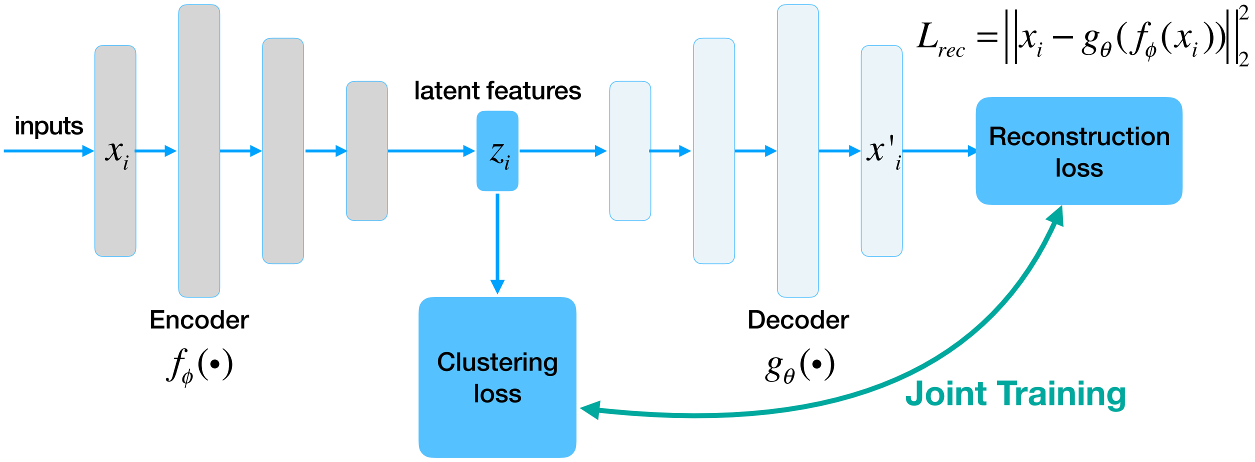


FIGURE 1. Architecture of clustering based on autoencoder. The network is trained by both clustering loss and reconstruction loss.

with a reconstruction loss. Although it has superior performance on several small-scale datasets, it is really memory-consuming and time-consuming and thus can not be applied to large-scale datasets. The reason is that its parameter number is $O(n^2)$ for n samples, and it can only be optimized by gradient descent.

- **Deep Multi-Manifold Clustering (DMC)**:
DMC [41] is deep learning based framework for *multi-manifold clustering* (MMC). It optimizes a joint loss function comprised of two parts: the locality preserving objective and the clustering-oriented objective. The first part makes the learned representations meaningful and embedded into their intrinsic manifold. It includes the autoencoder reconstruction loss and locality preserving loss. The second part penalizes representations based on their proximity to each cluster centroids, making the representation cluster-friendly and discriminative. Experimental results show that DMC has a better performance than the state-of-the-art multi-manifold clustering methods.
- **Deep Embedded Regularized Clustering (DEPICT)**:
DEPICT [14] is a sophisticated method consisting of multiple striking tricks. It consists of a softmax layer stacked on top of a multi-layer convolutional autoencoder. It minimizes a relative entropy loss function with a regularization term for clustering. The regularization term encourages balanced cluster assignments and avoids allocating clusters to outlier samples. Furthermore, the reconstruction loss of autoencoder is also employed to prevent corrupted feature representation. Note that each layer in both encoder and decoder contributes to the reconstruction loss, rather than only the input and output layer. Another highlight of this method is that it employs a noisy encoder to enhance the robustness of the algorithm. Experimental results show that DEPICT achieves superior clustering performance while having a high computational efficiency.
- **Deep Continuous Clustering (DCC)**:
DCC [42] is also an AE-based deep clustering algorithm. It aims at solving two limitations of deep clustering. Since most deep clustering algorithms are based on classical center-based, divergence-based or hierarchical clustering formulations, they have some inherent limitations. For one thing, they require setting the number of clusters in priori. For another, the optimization procedures of these methods involve discrete reconfigurations of the objective, which require updating the clustering parameters and network parameters alternatively. DCC is rooted in Robust Continuous Clustering (RCC) [43], a formulation having a clear continuous objective and no prior knowledge of clusters number. Similar to many other methods, the representation learning and clustering is optimized jointly.

B. CDNN-BASED DEEP CLUSTERING

CDNN-based algorithms only use the clustering loss to train the network, where the network can be FCN, CNN or DBN. The optimizing objective of CDNN-based algorithms can be formulated as follows:

$$L = L_c \tag{4}$$

Without the reconstruction loss, CDNN-based algorithms suffer from the risk of obtaining corrupted feature space, when all data points are simply mapped to tight clusters, resulting in a small value of clustering loss but meaningless. Consequently, the clustering loss should be designed carefully and network initialization is important for certain clustering loss. For this reason, we divide CDNN-based deep clustering algorithms into three categories according to the ways of network initialization, i.e., unsupervised pre-trained, supervised pre-trained and randomly initialized (non-pre-trained).

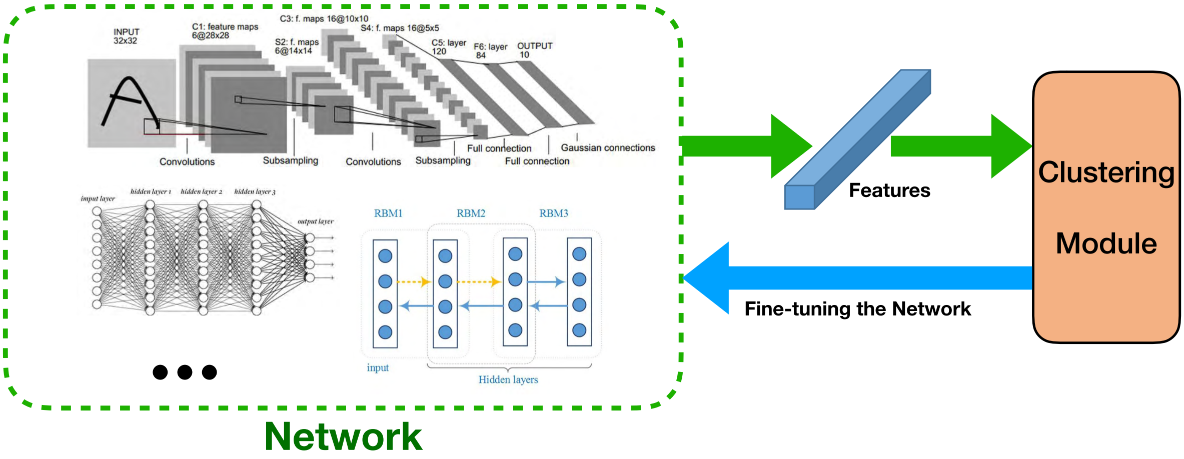


FIGURE 2. Architecture of CDNN-based deep clustering algorithms. The network is only adjusted by the clustering loss. The network architecture can be FCN, CNN, DBN and so on.

1) UNSUPERVISED PRE-TRAINED NETWORK

RBMs and autoencoders have been applied to CDNN-based clustering. These algorithms firstly train a RBM or an autoencoder in an unsupervised manner, then fine-tune the network (only encoder part for the autoencoder) by the clustering loss. Several representative algorithms are introduced as below.

- **Deep Nonparametric Clustering (DNC)**:
  DNC [30] leverages unsupervised feature learning with DBN for clustering analysis. It first trains a DBN to map original training data into the embedding codes. Then, it runs the nonparametric maximum margin clustering (NMMC) algorithm to obtain the number of clusters and labels for all training data. After that, it takes the fine-tuning process to refine the parameters of the top layer of the DBN. The experimental results show advantages over classical clustering algorithms.
- **Deep Embedded Clustering (DEC)**:
  DEC [11] is one of the most representative methods of deep clustering and attracts lots of attention into this field. It uses autoencoder as the network architecture and uses cluster assignment hardening loss as a regularization. It first trains an autoencoder by using the reconstruction loss and then drops the decoder part. The features extracted by the encoder network serve as the input of clustering module. After that, the network is fine-tuned using the cluster assignment hardening loss. Meanwhile, the clusters are iteratively refined by minimizing the KL-divergence between the distribution of soft labels and the auxiliary target distribution. As a result, the algorithm obtains a good result and become a reference to compare the performances of new deep clustering algorithms.
- **Discriminatively Boosted Clustering (DBC)**:
  DBC [12] has almost the same architecture with DEC and the only improvement is that it use convolutional autoencoder. In other words, it also first pre-trains an autoencoder and then uses the cluster assignment hardening loss to fine-tune the network, along with refining the clustering parameters. It outperforms DEC on image datasets on account of the use of the convolutional network.

2) SUPERVISED PRE-TRAINED NETWORK

Although unsupervised pre-training provides a better initialization of networks, it is still challenging to extract feasible features from complex image data. Guérin *et al.* [44] conduct extensive experiments by testing the performance of combinations of different popular CNN architectures pre-trained on ImageNet [45] and different classical clustering algorithms. The experimental results show that feature extracted from deep CNN trained on large and diverse labeled datasets, combined with classical clustering algorithms, can outperform the state-of-the-art image clustering methods. To this effect, when the clustering objective is complex image data, it is natural to make use of the most popular network architectures like VGG [46], ResNet [47] or Inception [48] models, which are pre-trained on large-scale image datasets like ImageNet, to speed up the convergence of iterations and to boost the clustering quality. The most remarkable method of this type is introduced as follows:

- **Clustering Convolutional Neural Network (CCNN)**:
  CCNN [17] is an efficient and reliable deep clustering algorithm which can deal with large-scale image datasets. It proposes a CNN-based framework to solve clustering and representation learning iteratively. It first randomly picks k samples and uses an initial model pre-trained on the ImageNet dataset to extract their features as the initial cluster centroids. In each step, mini-batch k-means is performed to update assignments of samples and cluster centroids, while stochastic gradient descent is used to update the parameters of the proposed CNN. The mini-batch k-means significantly reduces computation and memory costs, enabling CCNN to be adapted to large-scale datasets. Moreover, it

also includes a novel iterative centroid updating method that avoids drift error induced by the feature inconsistency between two successive iterations. At the same time, only top-k_m samples with the smallest distances to their corresponding centroids are chosen to update the network parameters, in order to enhance the reliability of updates. All these techniques improve the clustering performance. To the best of our knowledge, it is the only deep clustering method which can deal with the task of clustering millions of images.

3) NON-PRE-TRAINED NETWORK

Despite the fact that a pre-trained network can significantly boost the clustering performance, under the guidance of a well-designed clustering loss, the networks can also be trained to extract discriminative features.

- **Information Maximizing Self-Augmented Training (IMSAT)**:
  IMSAT [49] is an unsupervised discrete representation learning algorithm, the task of which is to obtain a function mapping data into discrete representations. Clustering is a special case of the task. It combines FCN and regularized Information Maximization (RIM) [50], which learns a probabilistic classifier such that mutual information between inputs and cluster assignments is maximized. Besides, the complexity of the classifier is regularized. At the same time, an flexible and useful regularization objective termed Self-Augmented Training (SAT) is proposed to impose the intended invariance on the data representations. This data augmentation technique significantly improves the performance of standard deep RIM. IMSAT shows state-of-the-art results on MNIST and REUTERS datasets.
- **Joint Unsupervised Learning (JULE)**:
  JULE [16] is proposed to learn feature representations and cluster images jointly. A convolutional neural network is used for representation learning and a hierarchical clustering (to be specific, agglomerative clustering) is used for clustering. It optimizes the objective iteratively in a recurrent process. Hierarchical image clustering is performed in the forward pass while feature representation is learned in the backward pass. In the forward pass, the representations of images are regarded as initial samples, and then label information is generated from an undirected affinity matrix based on the deep representations of images. After that, two clusters are merged according to a predefined loss metric. In the backward pass, the network parameters are iteratively updated towards obtaining a better feature representation by optimizing the already merged clusters. In experiments, the method shows excellent results on image datasets and indicates that the learned representations can be transferred across different datasets. Nevertheless, the computational cost and memory complexity are extremely high when datasets is large as it requires to construct an undirected affinity matrix. What is worse, the cost can hardly be optimized since it is a dense matrix.
- **Deep Adaptive Image Clustering (DAC)**:
  DAC [51] is a single-stage convolutional-network-based method to cluster images. The method is motivated from a basic assumption that the relationship between pairwise images is binary and its optimizing objective is the binary pairwise-classification problem. The images are represented by label features extracted by a convolutional neural network, and the pairwise similarities are measured by the cosine distance between label features. Furthermore, DAC introduces a constraint to make the learned label features tend to be one-hot vectors. Moreover, since the ground-truth similarities are unknown, it adopts an adaptive learning algorithm [52], an alternating iterative method to optimize the model. In each iteration, pairwise images with the estimated similarities are selected based on the fixed network, then the network is trained by the selected labeled samples. DAC converges when all instances are used for training and the objective can not be improved further. Finally, images are clustered according to the largest response of label features. DAC achieves superior performance on five challenging datasets.

C. VAE-BASED DEEP CLUSTERING

AE-based and CDNN-based deep clustering have made impressive improvements compared to classical clustering method. However, they are designed specifically for clustering and fail to uncover the real underlying structure of data, which prevent them from being extended to other tasks beyond clustering, e.g., generating samples. Worse still, the assumptions underlying the dimensionality reduction techniques are generally independent of the assumptions of the clustering techniques, thus there is no theoretical guarantee that the network would learn feasible representations. In recent years, Variational Autoencoder (VAE), a kind of deep generative model, has attracted extensive attention and motivated a large number of variants. In this section, we introduce the deep clustering algorithms based on VAE.

VAE can be considered as a generative variant of AE, as it enforces the latent code of AE to follow a predefined distribution. VAE combines variational bayesian methods with the flexibility and scalability of neural networks. It introduces neural networks to fit the conditional posterior and thus can optimize the variational inference objective via stochastic gradient descent [53] and standard backpropagation [54]. To be specific, it uses the reparameterization of the variational lower bound to yield a simple differentiable unbiased estimator of the lower bound. This estimator can be used for efficient approximate posterior inference in almost any model with continuous latent variables. Mathematically, it aims at minimizing the (variational) lower bound on the marginal likelihood of the dataset $X = \{x^{(i)}\}_{i=1}^{N}$, its objective function

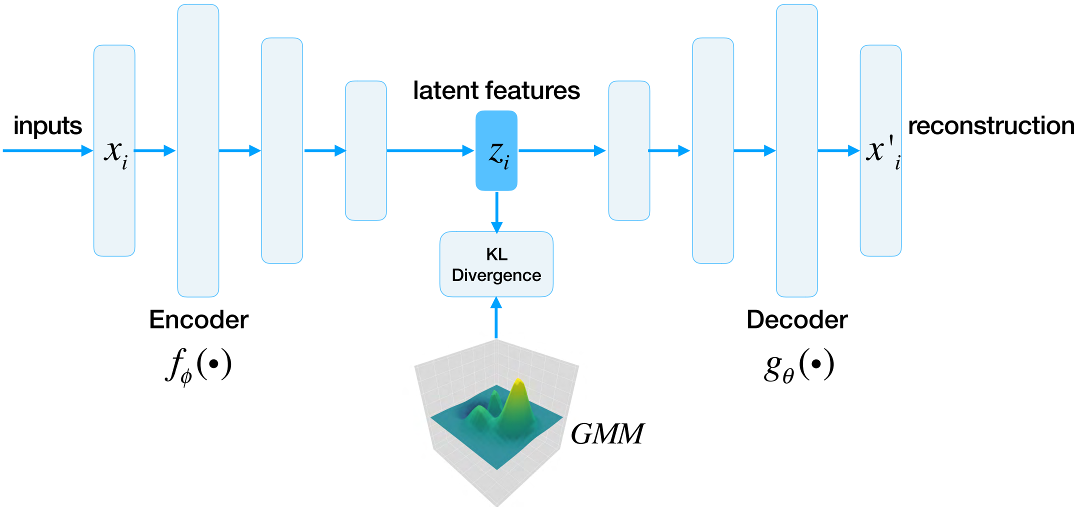


FIGURE 3. Architecture of VAE-based deep clustering algorithms. They impose a GMM priori over the latent code.

can be formulated as follows:

$$L(\theta, \phi; X) = \sum_{i}^{N}(-D_{KL}(q_\phi(z|\boldsymbol{x}^{(i)}) \parallel p(z)) + \mathbb{E}_{q_\phi(z|\boldsymbol{x}^{(i)})}[logp_\theta(\boldsymbol{x}^{(i)}|z)]) \qquad (5)$$

$p(z)$ is the priori over the latent variables. $q_\phi(z|\boldsymbol{x}^{(i)})$ is the variational approximation to the intractable true posterior $p_\phi(z|\boldsymbol{x}^{(i)})$ and $p_\theta(\boldsymbol{x}^{(i)}|z)$ is the likelihood function. From a coding theory perspective, the unobservable variables z can be interpreted as a latent representation, thus $q_\phi(z|\boldsymbol{x})$ is a probabilistic encoder and $p_\theta(\boldsymbol{x}|z)$ is a probabilistic decoder. In summary, the most significant difference between standard autoencoder and VAE is that VAE impose a probabilistic prior distribution over the latent representation z. In regular VAEs, the prior distribution $p(z)$ is commonly an isotropic Gaussian. But in the context of clustering, we should choose a distribution which can describe the cluster structure. As illustrated in Figure 3, existing algorithms choose a mixture of Gaussians as a priori. In other words, they assume that the observed data is generated from a mixture of Gaussians, inferring the class of a data point is equivalent to inferring which mode of the latent distribution the data point was generated from. After maximizing the evidence lower bound, the cluster assignment can be inferred by the learned GMM model. This kind of algorithms are able to generate images in addition to outputting clustering results, but they usually suffer from high computational complexity. Two related algorithms are presented as follows.

- **Variational Deep Embedding (VaDE)**:
  VaDE [15] consider the generative model $p(\boldsymbol{x}, z, c) = p(\boldsymbol{x}|z)p(z|c)p(c)$,In this model, an observed sample $\boldsymbol{x}$ is generated by the following process:

$$c \sim Cat(1/K), z \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2 I)$$
$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x(z), \boldsymbol{\sigma}_x^2(z)I) \text{ or } \mathcal{B}(\boldsymbol{\mu}_x(z))$$

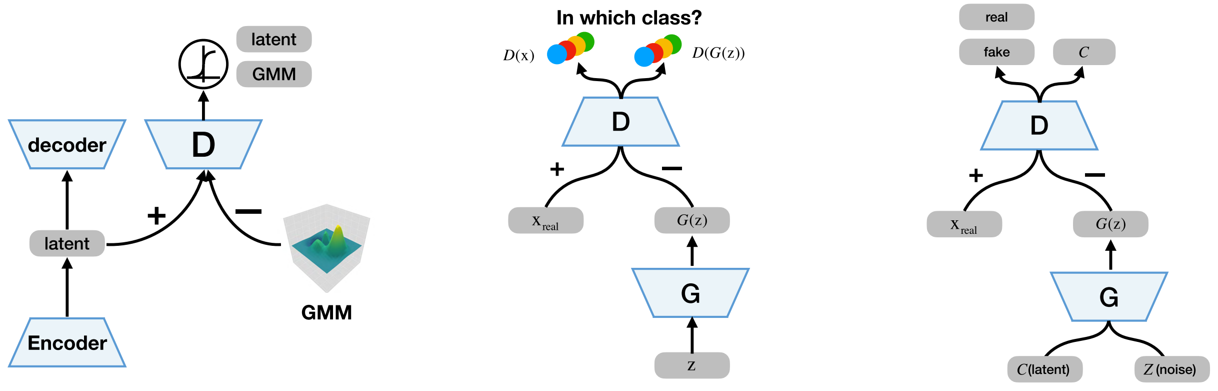
where $Cat(\cdot)$ is the categorical distribution, K is the predefined number of clusters, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and the variance of the Gaussian distribution corresponding to cluster c or parameterized by a given vector. $\mathcal{N}(\cdot)$ and $\mathcal{B}(\cdot)$ are multivariate Gaussian distribution and Bernoulli distribution parameterized by $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\mu}$ respectively. A VaDE instance is tuned to maximize the likelihood of the given samples. The log-likelihood of VaDE can be formulated as:

$$\begin{aligned} &\log p(\boldsymbol{x}) \\ &= \log \int_z \sum_c p(\boldsymbol{x}, z, c)dz \\ &\geq E_{q(z,c|\boldsymbol{x})}[\log \frac{p(\boldsymbol{x}, z, c)}{q(z, c|\boldsymbol{x})}] \\ &= L_{ELBO}(\boldsymbol{x}) \\ &= E_{q(z,c|\boldsymbol{x})}[\log p(\boldsymbol{x}|z)] - D_{KL}(q(z, c|\boldsymbol{x})||p(z, c)) \qquad (6) \end{aligned}$$

where L_{ELBO} is the evidence lower bound (ELBO), $q(z, c|\boldsymbol{x})$ is the variational posterior to approximate the true posterior $p(z, c|\boldsymbol{x})$. The first term in Equation 6 is the reconstruction loss (network loss L_n), and the second term, which is the Kullback-Leibler divergence from the Mixture-of-Gaussians (MoG) prior $p(z, c)$ to the variational posterior $q(z, c|\boldsymbol{x})$, can be consider as the clustering loss L_c. After the maximization of the lower bound, the cluster assignments can be inferred directly from the MoG prior.

- **Gaussian Mixture VAE (GMVAE)**:
  GMVAE [55] proposes a similar formulation. It considers the generative model $p(\boldsymbol{x}, z, \boldsymbol{n}, c) = p(\boldsymbol{x}|z)p_\beta(z|c, \boldsymbol{n})p(\boldsymbol{n})p(c)$. In this model, an observed sample $\boldsymbol{x}$ is generated as the following process:

$$c \sim Cat(1/K), \boldsymbol{n} \sim \mathcal{N}(0, I)$$
$$z \sim \mathcal{N}(\boldsymbol{\mu}_c(\boldsymbol{n}), \boldsymbol{\sigma}_c^2(\boldsymbol{n}))$$
$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x(z), \boldsymbol{\sigma}_x(z)I) \text{ or } \mathcal{B}(\boldsymbol{\mu}_x(z))$$



(a) (b) (c)

FIGURE 4. GAN-based deep clustering. (a) DAC. (b) CatGAN. (c) InfoGAN.

Note that GMVAE is a little complex than VaDE and has worse results empirically.

D. GAN-BASED DEEP CLUSTERING

Then Generative Adversarial Network (GAN) is another popular deep generative model in recent years. The (GAN) framework establishes a min-max adversarial game between two neural networks: a generative network, G, and a discriminative network, D. The generative network tries to map a sample z from a prior distribution $p(z)$ to the data space, while the discriminative network tries to compute the probability that a input is a real sample from the data distribution, rather than a sample generated by the generative network. The objective to this game can be formulated as follows:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D(\boldsymbol{x})] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \tag{7}$$

The generator G and the discriminator D can be optimized alternatively using SGD. The idea of GAN is interesting as it provides an adversarial solution to match the distribution of data or its representations with an arbitrary prior distribution. In recent years, many GAN-based algorithms have been proposed for clustering, some of which are specific to clustering task, while others just take clustering as a special case. GAN-based deep clustering algorithms have the same problems of GAN, e.g., hard to converge and mode collapse. The noticeable works are presented as follows:

- **Deep Adversarial Clustering (DAC)**:
DAC [56] is a generative model specific to clustering. It applies the adversarial autoencoder (AAE) [57] to clustering. AAE is similar to VAE as VAE uses a KL divergence penalty to impose a prior distribution on the latent representation, while AAE uses an adversarial training procedure to match the aggregated posterior of the latent representation with the prior distribution. Inspired by the success of VaDE, Harchaoui *et al.* [56] match the aggregated posterior of the latent representation with a Gaussian Mixture Distribution. The network architecture is illustrated in Figure 4(a). Its optimizing objective is comprised of three terms: the traditional auto-encoder reconstruction objective, the Gaussian mixture model likelihood, and the adversarial objective, where the reconstruction objective can be considered as the network loss, and the other two terms are the clustering loss. Experiment in [57] illustrates that it has a comparable result with VaDE on the MNIST dataset.
- **Categorial Generative Adversarial Network (CatGAN)**:
CatGAN [58] generalizes the GAN framework to multiple classes. As illustrated in Figure 4(b), it considers the problem of unsupervisedly learning a discriminative classifier D from dataset, which classifies the data points into a priori chosen number of categories instead of only two categories (fake or real). CatGAN introduces a new two player game based on GAN framework: Instead of requiring D to predict the probability of $\boldsymbol{x}$ belonging to real dataset, it enforces D to classify all data points into k classes, while being uncertain of class assignments for samples generated by G. On the other hand, it requires G to generate samples belonging to precisely one out of k classes, instead of generating samples belonging to the dataset. Mathematically, the goal of CatGAN is maximizing $H[p(c|\boldsymbol{x}, D)]$ and $H[p(c|D)]$, and minimizing $H[p(c|G(z), D)]$, where $H[\cdot]$ denotes the empirical entropy, $\boldsymbol{x}$ is the real sample, $\boldsymbol{x}$ is the random noise, and c is the class label. The objective function of the discriminator, which we refer to with $\mathcal{L}_D$, and the generator, which we refer to with $\mathcal{L}_G$ can be defined as follows:

$$\begin{aligned}\mathcal{L}_D &= \max_D H_{\mathcal{X}}[p(c|D)] - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}}[H[p(c|\boldsymbol{x}, D)]] \\ &\quad + \mathbb{E}_{z\sim P(z)}[H[p(c|G(z), D)]] \\ \mathcal{L}_G &= \min_G -H_G[p(c|D)] + \mathbb{E}_{z\sim P(z)}[H[p(c|G(z), D)]]\end{aligned} \tag{8}$$

TABLE 4. Comparison of different categories of deep clustering algorithms.

Categories	L_c	L_n	Description	Advantages	Disadvantages	Computational Complexity
AE-based DC	Yes	Yes (AE loss)	Joint optimize an AE and clustering parameters	1) Not obtain trivial solutions 2) Easy to implement	1) Introduce a hyper-parameter to balance the two losses 2) Limited network depth	Clustering loss specific
CDNN-based DC	Yes	No	Optimize the network only by clustering loss	1) Simple and graceful objective 2) Extended to large-scale tasks	1) Have the risk of obtaining corrupted feature space 2) Require well-designed clustering loss	Clustering loss specific
VAE-based DC	Yes	Yes	Impose a GMM priori on VAE	1) Capable to generate samples 2) Decent theoretical guarantee	High-computional complexity	High
GAN-based DC	Yes	Yes	Impose a multi-class priori on GAN	1) Capable to generate samples 2) Flexible	1) Hard to converge 2) Mode collapse	High

where $\mathcal{X}$ is the distribution of dataset. Empirical evaluation shows that CatGAN is superior to k-means and RIM [50] on a "circles" dataset.

- **Information Maximizing Generative Adversarial Network (InfoGAN)**:
  InfoGAN [59] is an unsupervised method that learns disentangled representations, and it can also be used for clustering. It can disentangle both discrete and continuous latent factors, scale to complicated datasets. The idea of InfoGAN is maximizing the mutual information [60] between a fixed small subset of the GAN's noise variables and the observation, which is relatively straightforward but surprisingly effective. To be specific, as illustrated in Figure 4(c), it decomposes the input noise vector into two parts: incompressible noise $\boldsymbol{z}$ and latent code $\boldsymbol{c}$, so the form of the generator becomes $G(\boldsymbol{z}, \boldsymbol{c})$. To avoid trivial codes, it uses an information-theoretic regularization to ensure that the mutual information between latent codes $\boldsymbol{c}$ and generator distribution $G(\boldsymbol{z}, \boldsymbol{c})$ should be high. The optimizing objective of InfoGAN become the following information-regularized minimax game:

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(\boldsymbol{c}; G(\boldsymbol{z}, \boldsymbol{c})) \quad (9)$$

  where V(D,G) denotes the object of standard GAN, and $I(\boldsymbol{c}; G(\boldsymbol{z}, \boldsymbol{c}))$ is the information-theoretic regularization. When choosing to model the latent codes with one categorical code having k values, and several continuous codes, it has the function of clustering data points into k clusters. Experiments in [59] shows that it can achieve 0.95 accuracy on the MNIST dataset.

E. SUMMARY OF DEEP CLUSTERING ALGORITHMS

In this part, we present an overall scope of deep clustering algorithms. Specifically, we compare the four categories of algorithms in terms of loss function, advantages, disadvantages and computational complexity. as shown in Table 4. In regard to the loss function, apart from CDNN-based algorithms, the other three categories of algorithms jointly optimize both clustering loss L_c and network loss L_n. The difference is that the network loss of AE-based algorithms is explicitly reconstruction loss, while the two losses of VAE-based and GAN-based algorithms are usually incorporated together.

AE-based DC algorithms are most common as autoencoder can be combined with almost all clustering algorithms. The reconstruction loss of autoencoder ensures the network learns a feasible representation and avoid obtaining trivial solutions. However, due to the symmetry architecture, the network depth is limited for computational feasibility. Besides, the hyper-parameter to balance the two losses requires extra fine-tuning. In contrast to AE-based DC algorithms, CDNN-based DC algorithms only optimize the clustering loss. Therefore, the depth of network is unlimited and supervisedly pre-trained architectures can be used to extract more discriminative features, thus they are capable to cluster large-scale image datasets. However, without the reconstruction loss, they have the risk of learning a corrupted feature representation, thus the clustering loss should be well-designed. VAE-based and GAN-based DC algorithms are generative DC techniques, as they are capable to generate samples from the finally obtained clusters. VAE-based algorithms have a good theoretical guarantee because they minimizes the variational lower bound on the marginal likelihood of data, but it suffer from high-computational complexity. GAN-based algorithms impose a multi-class priori on general GAN framework. They are more flexible and diverse than VAE-based ones. Some of them aim at learning interpretable representations and just take clustering task as a specific case. The shortcomings of GAN-based algorithms are similar to GANs, e.g, mode collapse and converge slowly.

The computational complexity of deep clustering varies a lot. For AE-based and CDNN-based algorithms, the computational cost is highly related to the clustering loss. For example, k-means loss results in a relatively low overhead while the cost of agglomerative clustering is extremely high. At the same time, the network architectures also influence the computational complexity significantly, as a deep CNN requires a long time to train. For VAE and GAN, due to the difficulty to optimize, they usually have a higher computational complexity than efficient methods in the AE-based and CDNN-based categories, e.g., DEC, DCN, DEPICT and so on.

IV. FUTURE OPPORTUNITIES AND CONCLUSIONS

A. FUTURE OPPORTUNITIES OF DEEP CLUSTERING

Based on the aforementioned literature review and analysis, we argue that the following perspectives of deep clustering are worth being studied further:

1) **Theoretical exploration**
Although jointly optimizing networks and clustering models significantly boost the clustering performance, there is no theoretical analysis explaining why it works and how to further improve the performance. Therefore, it is meaningful to explore the theoretical guarantee of deep clustering, in order to guide further researches in this area.

2) **Other network architectures**
Existing deep clustering algorithms mostly focus on image datasets, while few attempts have been made on sequential data, e.g., documents. To this effect, it is recommended to explore the feasibility of combining other network architectures with clustering, e.g., recurrent neural network [61].

3) **Tricks in deep learning**
It is viable to introduce some tricks or techniques used in supervised deep learning to deep clustering, e.g. data augmentation and specific regularizations. A concrete example is augmenting data with noise to improve the robustness of clustering methods.

4) **Other clustering tasks**
Combining deep neural networks with diverse clustering tasks, e.g. multi-task clustering, self-taught clustering (transfer clustering) [62], is another interesting research direction. To the best of our knowledge, these tasks have not exploited the powerful non-linear transformation of neural networks.

B. CONCLUSION REMARKS

As deep clustering is widely used in many practical applications for its powerful ability of feature extraction, it is natural to combine clustering algorithms with deep learning for better clustering results. In this paper, we give a systematic survey of deep clustering, which is a popular research field of clustering in recent years. A taxonomy of deep clustering is proposed from the perspective of network architectures, and the representative algorithms are presented in detail. The taxonomy explicitly shows the characteristics, advantages and disadvantages of different deep clustering algorithms. Furthermore, we provide several interesting future directions of deep clustering. We hope this work can serve as a valuable reference for researchers who are interested in both deep learning and clustering.

REFERENCES

[1] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, nos. 1–3, pp. 1–6, 1998.
[2] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*. Springer, 2015, pp. 827–832.
[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, pp. 226–231.
[4] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.
[5] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. Roy. Stat. Soc. C, Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.
[6] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[7] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Stat.*, vol. 36, no. 3, pp. 1171–1220, 2008.
[8] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 849–856.
[9] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[10] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 31–35.
[11] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.
[12] F. Li, H. Qiao, B. Zhang, and X. Xi. (2017). "Discriminatively boosted image clustering with fully convolutional auto-encoders." [Online]. Available: https://arxiv.org/abs/1703.07980
[13] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. (2016). "Towards K-means-friendly spaces: Simultaneous deep learning and clustering." [Online]. Available: https://arxiv.org/abs/1610.04794
[14] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5747–5756.
[15] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. (2016). "Variational deep embedding: An unsupervised and generative approach to clustering." [Online]. Available: https://arxiv.org/abs/1611.05148
[16] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5147–5156.
[17] C.-C. Hsu and C.-W. Lin, "CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 421–429, Feb. 2018.
[18] Z. Wang, S. Chang, J. Zhou, M. Wang, and T. S. Huang, "Learning a task-specific deep architecture for clustering," in *Proc. SIAM Int. Conf. Data Mining*, 2016, pp. 369–377.
[19] E. Aljalbout, V. Golkov, Y. Siddiqui, and D. Cremers. (2018). "Clustering with deep learning: Taxonomy and new methods." [Online]. Available: https://arxiv.org/abs/1801.07648
[20] A. Ng, "Sparse autoencoder," CS294A Lecture Notes, 2011.
[21] M. D. Boomija and M. Phil, "Comparison of partition based clustering algorithms," *J. Comput. Appl.*, vol. 1, no. 4, pp. 18–21, 2008.
[22] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 231–240, 2011.
[23] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
[24] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[25] D. P. Kingma and M. Welling. (2013). "Auto-encoding variational Bayes." [Online]. Available: https://arxiv.org/abs/1312.6114
[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[27] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, ''Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,'' in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 609–616.

[28] G. E. Hinton, ''A practical guide to training restricted Boltzmann machines,'' *Momentum*, vol. 9, no. 1, pp. 599–619, 2010.

[29] D. Beeferman and A. Berger, ''Agglomerative clustering of a search engine query log,'' in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2000, pp. 407–416.

[30] G. Chen. (2015). ''Deep learning with nonparametric clustering.'' [Online]. Available: https://arxiv.org/abs/1501.03084

[31] P. Huang, Y. Huang, W. Wang, and L. Wang, ''Deep embedding network for clustering,'' in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 1532–1537.

[32] X. Peng, J. Feng, S. Xiao, J. Lu, Z. Yi, and S. Yan. (2017). ''Deep sparse subspace clustering.'' [Online]. Available: https://arxiv.org/abs/1709.08374

[33] H. W. Kuhn, ''The Hungarian method for the assignment problem,'' *Nav. Res. Logistics*, vol. 52, no. 1, pp. 7–21, 2005.

[34] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, ''Normalized mutual information feature selection,'' *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[35] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, ''Extracting and composing robust features with denoising autoencoders,'' in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[36] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, ''Sparse autoencoder-based feature transfer learning for speech emotion recognition,'' in *Proc. Hum. Assoc. Conf. Affect. Comput. Intell. Interact. (ACII)*, Sep. 2013, pp. 511–516.

[37] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, ''Deep subspace clustering networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 23–32.

[38] R. Vidal, ''Subspace clustering,'' *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.

[39] E. Elhamifar and R. Vidal, ''Sparse subspace clustering,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2790–2797.

[40] S. R. Rao, R. Tron, R. Vidal, and Y. Ma, ''Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[41] D. Chen, J. Lv, and Z. Yi, ''Unsupervised multi-manifold clustering by learning deep representation,'' in *Proc. Workshops 31st AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 385–391.

[42] S. A. Shah and V. Koltun. (2018). ''Deep continuous clustering.'' [Online]. Available: https://arxiv.org/abs/1803.01449

[43] S. A. Shah and V. Koltun, ''Robust continuous clustering,'' *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 37, pp. 9814–9819, 2017.

[44] J. Guérin, O. Gibaru, S. Thiery, and E. Nyiri. (2017). ''CNN features are also great at unsupervised classification.'' [Online]. Available: https://arxiv.org/abs/1707.01700

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ''ImageNet: A large-scale hierarchical image database,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.

[46] K. Simonyan and A. Zisserman. (2014). ''Very deep convolutional networks for large-scale image recognition.'' [Online]. Available: https://arxiv.org/abs/1409.1556

[47] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ''Rethinking the inception architecture for computer vision,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.

[49] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. (2017). ''Learning discrete representations via information maximizing self-augmented training.'' [Online]. Available: https://arxiv.org/abs/1702.08720

[50] A. Krause, P. Perona, and R. G. Gomes, ''Discriminative clustering by regularized information maximization,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 775–783.

[51] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, ''Deep adaptive image clustering,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 5879–5887.

[52] M. P. Kumar, B. Packer, and D. Koller, ''Self-paced learning for latent variable models,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.

[53] L. Bottou, ''Large-scale machine learning with stochastic gradient descent,'' in *Proc. COMPSTAT*, 2010, pp. 177–186.

[54] R. Hecht-Nielsen, ''Theory of the backpropagation neural network,'' *Neural Netw.*, vol. 1, no. 1, pp. 445–448, 1988.

[55] N. Dilokthanakul *et al.* (2016). ''Deep unsupervised clustering with Gaussian mixture variational autoencoders.'' [Online]. Available: https://arxiv.org/abs/1611.02648

[56] W. Harchaoui, P. A. Mattei, and C. Bouveyron, ''Deep adversarial Gaussian mixture auto-encoder for clustering,'' in *Proc. ICLR*, 2017, pp. 1–5.

[57] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. (2015). ''Adversarial autoencoders.'' [Online]. Available: https://arxiv.org/abs/1511.05644

[58] J. T. Springenberg. (2015). ''Unsupervised and semi-supervised learning with categorical generative adversarial networks.'' [Online]. Available: https://arxiv.org/abs/1511.06390

[59] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, ''InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[60] A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger, ''Hierarchical clustering using mutual information,'' *Europhys. Lett.*, vol. 70, no. 2, p. 278, 2005.

[61] L. Medsker and L. Jain, ''Recurrent neural networks: Design and applications,'' in *Proc. Joint Conf. Neural Netw.*, vol. 5, 1999.

[62] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, ''Self-taught clustering,'' in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 200–207.

ERXUE MIN received the B.S. degree from the National University of Defense Technology, Changsha, China, in 2016, where he is currently pursuing the master's degree with the School of Computer. His research interests include machine learning, data mining, optimization, and intrusion detection.

XIFENG GUO received the M.S. degree in computer science from the National University of Defense Technology, China, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include machine learning, deep learning, transfer learning, unsupervised learning, and computer vision.

QIANG LIU (M'14) received the Ph.D. degree in computer science and technology from the National University of Defense Technology (NUDT) in 2014. He is currently an Assistant Professor at NUDT. He has contributed over 50 archived journals and international conference papers, such as the *IEEE Network Magazine*, TKDE, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON CYBERNETICS, *Pattern Recognition*, the IEEE COMMUNICATIONS LETTERS, *Neurocomputing*, *Neural Computing and Applications*, *Mobile Information Systems*, ICC'18, EDBT'17, WCNC'17, ICANN'17, and SmartMM'17. His research interests include 5G network, Internet of Things, wireless network security, and machine learning. He is a member of the China Computer Federation. He currently serves on the Editorial Review Board of *Artificial Intelligence Research* journal.

GEN ZHANG received the B.S. degree from the National University of Defense Technology, Changsha, China, in 2016, where he is currently pursuing the master's degree with the College of Computer. His research interests include binary analysis, grey-box fuzzing, and deep learning.

JIANJING CUI received the B.S. degree from the University of Science and Technology Beijing in 2016. He is currently pursuing the master's degree with the School of Computer, NUDT. His research interests include machine learning, intrusion detection, and data mining.

JUN LONG received the Ph.D. degree from the National University of Defense Technology (NUDT), Changsha, China, in 2009. He is currently an Associate Professor with the School of computer, NUDT. His research interests include machine learning, intrusion detection, and data visualization.