Clustering with Deep Neural Networks – An Overview of Recent Methods

Janik Schnellbach, Marton Kajo*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: janik.schnellbach@tum.de, kajo@net.in.tum.de

***Abstract*—The application of clustering has always been an important method for problem-solving. As technology advances, in particular the trend of Deep Learning enables new methods of clustering. This paper serves as an overview of recent methods that are based on Deep Neural Networks (DNNs). The approaches are categorized depending on the underlying architecture as well as their intended purpose. The classification highlights and explains the four categories of Feedforward Networks, Autoencoders (AEs) as well as the generative setups of Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs). Subsequently, a comparison of the concepts points out the advantages and disadvantages while evaluating their suitability in the area of image clustering.**

***Index Terms*—Deep Neural Networks, Deep Clustering, Variational Autoencoder, Generative Adversarial Net**

Figure 1: John Snow's Death Map [1]

1. Introduction

The basic idea of clustering is the analysis of data with the aim to categorize it into groups sharing certain similarities. The assessed data can range from a small number of characteristics to a huge multidimensional set. Because it is expected to derive certain trends from the input, clustering is a common method to solve practical problems.

A particular example is the application of clustering as performed by John Snow back in the 19th century. John Snow worked as a physician during the cholera epidemic in London. His idea was to mark the cholera deaths on a map of the city, as one can see in Figure (1). Since the deaths notably centered around water pumps, he discovered the correlation between the water supply and the epidemic.

While John Snow did his clustering task manually on a sheet of paper, nowadays methods allow clustering in an automated manner. The application of Artificial Intelligence enables to process big amounts of data while being way more effective. One can distinguish between Supervised and Unsupervised Learning. Supervised Learning assigns the data to prior defined classes of characteristics and qualities. This process is also called classification. In contrast, Unsupervised Learning, of which one category is clustering, can uncover those classes simply from the given set of data without preliminary definitions [2]. The methodology of clustering can either be generative or discriminative. The generative approach tries to work out the data distribution with statistical models such as a Gaussian Mixture Model (GMM) or the k-means algorithm. These models will be explained later in the paper. Discriminative Clustering on the other hand applies separation and classification techniques to map the data into categories without any detour. Regularized information maximization (RIM) is a famous example of this type and will also be discussed in the next section [3].

As both, the amount of data as well as the type of data can vary considerably, a steadily growing selection of methods is currently available. With an increasing amount of approaches, it can be difficult to maintain an overview of the various concepts. The recently published work of Technical University in Munich [4] discusses the current state of the art deep clustering algorithms in a taxonomy. The authors give an overview of the different approaches on a modular basis to provide a starting point for the creation of new methods. However, it lacks proper classification of currently available frameworks, as the authors rather have an eye for the composition of methods instead of the big picture. For this reason, our paper makes a further contribution towards this set of methods with a more detailed description of the concepts as well as a proper classification of them. As it has only been marginally included in the recent paper, special attention is given to novel trends in the area of Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs).

In the following, Section 2 describes the different categories for clustering with Deep Neural Networks (DNNs). For each category, several methods are illustrated. Subsequently, Section 3 does provide an evaluation of the aforementioned methods, with regard to the application area of images, followed by a summary in Section 4.

doi: 10.2313/NET-2020-04-1_08

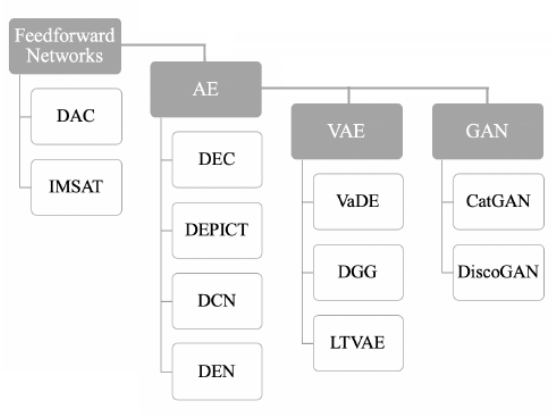


Figure 2: Overview of methods that are addressed in this paper. Feedforward Networks are the basic building block for AEs. VAEs and GANs then again consist of AEs themselves.

2. Deep Clustering

2.1. Feedforward Networks

As a standard setup of a Neural Network, one can define a group of Feedforward Network architectures that follow the same approach: the optimization of a specific clustering loss [5]. This category can be subdivided into Fully-Connected Neural Networks (FCNs) and Convolutional Neural Networks (CNNs).

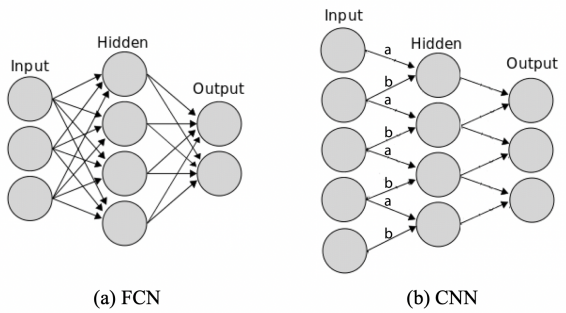


Figure 3: Layout of Feedforward Networks [6]

FCN is also frequently called Multilayer Perceptron (MLP). This architecture has a topology where each neuron of a layer is connected with every neuron on the subjacent layer. The links between neurons have their own weight, regardless of the other connections. CNNs, on the other hand, are rather inspired by the biological layout of neurons, which means that a neuron is only connected to a few others of the overlying layer [5]. In contrast to FCN, a consistent pattern of weightings is used between the neurons of two layers. Figure (3) illustrates the layouts and their weighting described above.

Deep Adaptive Clustering (DAC) is an approach for image clustering, developed by the University of Chinese Academy of Sciences. Due to the area of application, it is also called Deep Adaptive Image Clustering. DAC handles the relationship of two pictures as a binary relationship. By doing this, it decides whether an image matches a certain cluster or not. The pictures are compared by the cosine distance of previously calculated label features, that are extracted from the images by a CNN. Based on the results, the framework decides whether the pictures belong to the same or different clusters. However, this method requires a good initial distribution of clusters, which can be hard to initialize [7].

Information Maximizing Self-Augmented Training (IMSAT) The Previously described feedforward method is based on CNNs. However, this paper seeks to provide a broad overview of the different approaches pending on the network architecture. An example for the application of FCNs is IMSAT. This method is based and advanced from the method of Regularized Information Maximization (RIM) [8].

The basic idea is to handle both the class balance as well as the class separation, meaning that RIM has the objective to balance the amount of data entities inside the clusters. The underlying FCN applies a function that maps data dependent upon the similarity into similar or dissimilar discrete representations. Additionally, Self Augmentation is applied to the data set. This is done, in order to impose the invariance on the data representations [9].

2.2. Autoencoder (AE)

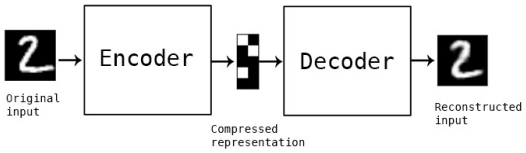


Figure 4: Basic layout of an AE [10]

The above described Feedforward Networks can be used to assemble the network of an AE, which is shown in Figure (4). It consists of an encoder and a decoder [11]. Both have different tasks during their training phase. While the encoder maps the input data according to an encode function within a latent space, the decoder reconstructs the initial input data with the objective of a minimal loss on the reconstruction [12]. The encoder, as well as the decoder, can either be constructed as FCN or CNN. The setup can be trained according to a certain data set [5].

Training can be divided into two phases. While one can separate the two phases in a logical way, both are generally realised simultaneously. During the first phase, the AE performs a pretraining while focusing on the minimization of the basic reconstruction loss. The optimization of this parameter is carried out by any type of AE. The second phase can be seen as a finetuning of the network. The approaches for this step can differ substantially, as various kinds of clustering parameters can be used to optimize the result. The different finetuning strategies are described as part of the approaches presented in the following paragraphs [4].

Deep Embedded Clustering (DEC) is possibly the

doi: 10.2313/NET-2020-04-1_08

most significant contribution in the area of clustering with AEs. For the second phase, the so-called cluster assignment hardening loss is optimized. The framework targets to minimize the Kullback–Leibler divergence between an initially computed soft assignment and an auxiliary target distribution. This is done iteratively, with an accompanied improvement of the clustering assignment [13]. It is often used as a starting point, as well as a comparison tool for other approaches [14].

Deep Embedded Regularized Clustering (DEPICT) This approach is based on DEC and is particularly suited for image datasets. It mitigates the risk of reaching degenerative solutions by the addition of a balanced assignment loss [4].

Deep Clustering Network (DCN) extends the previously described AE with the k-means algorithm. The k-means optimization tries to cluster the data around so-called cluster centers to enable an easier representation of the data. DCN optimizes k-means along with the reconstruction loss in the second phase [4].

Deep Embedding Network (DEN) The DEN approach has the objective to improve the clustering towards an effective representation. This is done by an additional locality-preserving loss as well as a group sparsity loss that are jointly optimized in the second phase [14].

2.3. VAEs

While the two aforementioned types can result in high-quality clustering, they are not able to point out the actual coherence of the analyzed data set. Knowledge about that enables to synthesize sample data from the existing dataset. This can be particularly impressive for pictures. In a nutshell, VAE is a refined variant of the traditional AE that forces the AE cluster to impose a certain distribution. It optimizes the lower bound of a data log-likelihood function [15].

Variational Deep Embedding (VaDE) VaDE uses a GMM as the predefined distribution. The GMM selects a fitting cluster that is subsequently transposed towards an observable embedding by a DNN [15].

Deep clustering via GMM VAE with graph embedding (DGG) extends the GMM with stochastic graph embedding in order to address a scattered and complex spread of data points. Graph embedding is applied to the pairs of vertexes in a similarity graph. The objective is to retain information about the relationship of the pairs while mapping each node as a vector with preferably low dimension [16]. The relationship and similarity among pairs are calculated by a minimization of the weighted distance, using their posterior distributions. In summary, DGG optimizes a combination of the loss of the previously described graph embedding with the already known GMM distributive function [17].

Latent tree VAE (LTVAE) has been published by researchers from Hong Kong earlier this year. Their framework takes a particular account of the multidimensionality and the associated range of differentiating structures concerning the data. A tree structure is used, built by multiple latent variables, each including a partition of data. During a learning phase, the tree updates itself, using the relationships among the different facets of the data. Figure (5) shows four different facets as the outcome of clustering applied to the STL-10 dataset. It can be observed that (b) Facet 2 has an emphasis on the front of the cars, compared to the other facets. In general, Facet 2 seems to have a relation to the eyes and lights of the objects. Also, when comparing the deers of facet 2 and 3, one can recognize a pattern in facet 2 with an emphasis on the antlers of the animals [18].

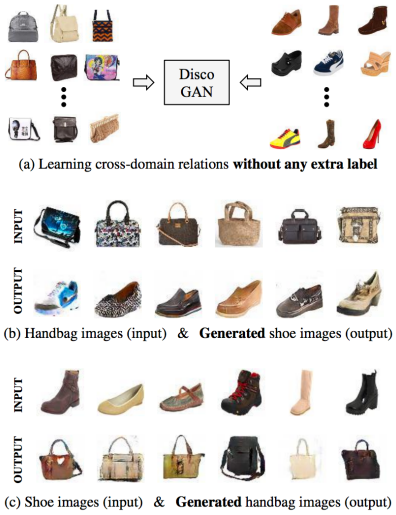
Figure 5: Results for application of LTVAE to STL-10 [19]

2.4. GANs

Next to VAEs, we take a closer look at GANs. A GAN is constructed from a generator and a discriminator. Those two operate in a minimax game. The generator is trained towards a distribution of a certain data set. The discriminator has the task to verify whether a sample from this distribution is a real one or a fake one. Based on this verification, feedback is given to the generator which is used to further improve the sample quality [20].

Categorical GAN (CatGAN) A popular modification of the common GANs are the CatGANs. In simple terms, the discriminator no longer decides whether the samples are real or not. Instead, samples are assigned to appropriate categories. CatGANs use a combination of generative and discriminative approaches. This novel approach requires the generator to spread the samples across the different categories in a balanced way and, most importantly, the generated samples need to be clearly classifiable for the discriminator [3, Section 3.2].

Discover relations between different domains (DiscoGAN) DiscoGANs are based on the idea of cross-domain relations. Human beings are able to understand correlations among different entities. For instance, one can discover the relationship between shoes and handbags that share a resemblance in their color



(a) Learning cross-domain relations **without any extra label**

(b) Handbag images (input) & **Generated** shoe images (output)

(c) Shoe images (input) & **Generated** handbag images (output)

Figure 6: Application of DiscoGAN [22]

sample. Figure (6) presents the application of DiscoGANs on this particular example. Mutually independent image sets of shoes on the one hand and bags, on the other hand, are subject to this picture. Depending on the input, the GAN finds a visually appropriate match.

DiscoGANs can associate an entity from a given pool of entities to a fitting entity from a different pool of entities. This is achieved by coupling two different GANs, which are able to map each entity to the opposite entity [21]. This technique enables to discover links between different clusters and therefore DiscoGANs may create new clusters by combining existing ones.

3. Discussion

After the previous section pointed out the different categories with the different types, this part focuses on the application as well as the advantages and disadvantages of the frameworks. The comparison is made on the level of categories, focusing on the application area of images. Since FCNs are fully connected, they are less suited for image processing. For high-resolution images, FCNs quickly find themselves reaching the limits of feasibility in terms of trainability and depth. Therefore, CNNs are rather suited for images. Depending on the requirements, the depth of Feedforward Networks and in particular of CNNs can be adapted.

The depth of AEs is rather limited since the opposing layout of decoder and encoder requires the depth on both sides. Instead, AEs offer the usage of different clustering parameters, which can be jointly optimized. Conventional Feedforward Networks solely optimize clustering loss.

In contrast to the previous methods, VAEs and GANs feature the ability of sample generation. In general, the optimization process of both can be expected to require a larger extent of computing power than Feedforward Networks and AEs [5]. Considering images once more, GANs usually score better than VAEs in terms of image quality, as the usage of the maximum likelihood approach tends to deliver blurry images. With a more rapid generation and better quality through a generative model, GANs usually score better. It can be said that the general setup allows a more extensive and rather flexible usage in comparison to VAEs [23].

This paper does offer a large extent of recent approaches and methods. In addition, we want to provide further food for thoughts in the area of deep clustering.

Deep Believe Networks (DBNs) As briefly mentioned in the context of DGG, there is a group of generative graphical models that have not been mentioned yet. DBNs are assembled by multiple stacked Restricted Boltzmann machines (RBMs). The starting paper [4] provides Nonparametric Maximum Margin Clustering (NMMC) as an example for DBNs.

Further types of GANs do also apply adversarial nets with the objective of clustering. Information Maximizing Generative Adversarial Nets (InfoGANs) learn the disentangled representation of the data and are particularly suited for scaling of complex datasets [5]. Other types may not have an immediate link to the task of clustering. However, the fundamentals of those might be useful for future research. Stacked GANs (StackGANs), for instance, address the task of image generation based on textual descriptions. It is based on a divide and conquer approach that splits up the problem into smaller subproblems [24].

VAE-GANs combine the two approaches of sample generating methods. As described in [25], the idea is to replace the decoder of a VAE with a GAN. This tries to deal with the blurry images that were mentioned earlier in this section. The idea behind its design is to cope with the VAE's reconstruction task by utilizing the detected feature representation from the discriminator of the GAN. However, as mentioned before, both require much computing power, which applies all the more for a combination as described above.

4. Conclusion

In this paper, we have emphasized the opportunities for clustering, which emerge through the recent advancements in the area of Deep Learning. Based on the network layout we derived different categories. For each of them, several frameworks are described in detail, featuring information about a preferred application area. In addition, we provided a comparison of the categories which included a specific focus on image clustering with special attention to the respective advantages and disadvantages. Finally, we give a further reference to different technologies that haven't been mentioned in this paper.

Overall, our paper has provided a general overview of the existing clustering frameworks and can further be used to get deeper into either the general topic of Deep Clustering or a specific type of category.

References

[1] [Online]. Available: http://blog.rtwilson.com/wp-content/uploads/2012/01/SnowMap_Points-1024x724.png

[2] R. Sathya and A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification," *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, 2013. [Online]. Available: http://dx.doi.org/10.14569/IJARAI.2013.020206

[3] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," 2015.

[4] E. Aljalbout, V. Golkov, Y. Siddiqui, and D. Cremers, "Clustering with deep learning: Taxonomy and new methods," *CoRR*, vol. abs/1801.07648, 2018. [Online]. Available: http://arxiv.org/abs/1801.07648

[5] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39 501–39 514, 2018.

[6] [Adjusted]. [Online]. Available: https://www.researchgate.net/profile/Eftim_Zdravevski/publication/327765620/figure/fig3/AS:672852214812688@1537431877977/Fully-connected-neural-network-vs-convolutional-neural-network-with-filter-size-1-2.ppm

[7] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 5880–5888.

[8] A. Krause, P. Perona, and R. G. Gomes, "Discriminative clustering by regularized information maximization," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 775–783. [Online]. Available: http://papers.nips.cc/paper/4154-discriminative-clustering-by-regularized-information-maximization.pdf

[9] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama, "Learning discrete representations via information maximizing self-augmented training," 2017.

[10] [Online]. Available: https://i.stack.imgur.com/zzzp7.jpg

[11] N. Mrabah, N. M. Khan, and R. Ksantini, "Deep clustering with a dynamic autoencoder," *CoRR*, vol. abs/1901.07752, 2019. [Online]. Available: http://arxiv.org/abs/1901.07752

[12] D. Berthelot, C. Raffel, A. Roy, and I. J. Goodfellow, "Understanding and improving interpolation in autoencoders via an adversarial regularizer," *CoRR*, vol. abs/1807.07543, 2018. [Online]. Available: http://arxiv.org/abs/1807.07543

[13] J. Xie, R. B. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," *CoRR*, vol. abs/1511.06335, 2015. [Online]. Available: http://arxiv.org/abs/1511.06335

[14] T. Yang, G. Arvanitidis, D. Fu, X. Li, and S. Hauberg, "Geodesic clustering in deep generative models," *CoRR*, vol. abs/1809.04747, 2018. [Online]. Available: http://arxiv.org/abs/1809.04747

[15] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: A generative approach to clustering," *CoRR*, vol. abs/1611.05148, 2016. [Online]. Available: http://arxiv.org/abs/1611.05148

[16] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, Jan 2007.

[17] L. Yang, N.-M. Cheung, J. Li, and J. Fang, "Deep clustering by gaussian mixture variational autoencoders with graph embedding," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[18] X. Li, Z. Chen, and N. L. Zhang, "Latent tree variational autoencoder for joint representation learning and multidimensional clustering," *CoRR*, vol. abs/1803.05206, 2018. [Online]. Available: http://arxiv.org/abs/1803.05206

[19] [Online]. Available: https://d3i71xaburhd42.cloudfront.net/7b85357834e398437a291906aded59caff5151eb/9-Figure6-1.png

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[21] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," *CoRR*, vol. abs/1703.05192, 2017. [Online]. Available: http://arxiv.org/abs/1703.05192

[22] [Online]. Available: https://ieee.nitk.ac.in/blog/assets/img/GAN/discogan.png

[23] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," 2016.

[24] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *CoRR*, vol. abs/1612.03242, 2016. [Online]. Available: http://arxiv.org/abs/1612.03242

[25] A. B. L. Larsen, S. K. Sønderby, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *CoRR*, vol. abs/1512.09300, 2015. [Online]. Available: http://arxiv.org/abs/1512.09300