Deep Clustering Self-Organizing Maps with Relevance Learning

Heitor R. Medeiros [1] **Pedro H. M. Braga** [1] **Hansenclever F. Bassani** [1]

1. Introduction

Clustering is one of the most natural ways of summarizing and organizing data. In particular, the main objective of clustering is to separate data into groups of similar data points. Although there exists multiple successful approaches for clustering data with high-level features, clustering high-dimensional raw data such as image and sound is a hard task. Techniques based on *Deep Learning* (DL) have been very successful in yielding good high-level representations for such type of data (Bengio et al., 2013; Aljalbout et al., 2018). Some of the most efficient approaches for producing representations from unlabeled data are *Autoencoder* (AE), *Variational Autoencoder* (VAE) and *Generative Adversarial Networks* (GAN). Moreover, those techniques can be applied in different ways, such as with self-labeling (Asano et al., 2019), or *Deep Clustering* (DC) (Nutakki et al., 2019). This work focuses on DC, that differs from conventional approaches according to the algorithms structure, network architectures, loss functions, and optimization methods for training (Nutakki et al., 2019).

In this work, we investigate the Joint DC, a task that aims at learning a good representation of the input data as well as clustering prototypes by minimizing a loss that combines clustering and reconstruction errors. Our preliminary results show that an AE combined with state-of-art clustering methods based on *Self-Organizing Map* (SOM) with relevance learning produce a meaningful topology in the latent space and clustering prototypes that represent well the existing data categories for MNIST dataset.

2. Research Problem and Motivation

DC approaches treat representation learning and clustering as a joint task and focus on learning representations that are clustering-friendly, i.e., that preserve the prior knowledge of cluster structure. It is typically performed by optimizing a loss function ($\mathcal{L}_c$), which can be seen as a clustering loss, jointly with a regular loss ($\mathcal{L}_n$), such as the reconstruction loss ($\mathcal{L}_{rec}$) of an AE (Ji et al., 2017), or the variational loss of a VAE (Jiang et al., 2016).

In this work, we particularly focus on a family of clustering algorithms called SOM (Kohonen, 1990). SOM is a biologically inspired unsupervised learning method that maps data from a higher-dimensional input space to a lower-dimensional output space, while preserving the similarities and the topological relations found between points in the input space. Each map unit is associated with a prototype vector from the original data space. State-of-the-art SOM-based models are suitable for clustering high-level features, such Braga & Bassani (2019); Bassani & Araujo (2015).

Recent works use the SOM loss, achieving good clustering results. For instance, *Deep Embedded Self-Organizing Map* (DESOM) uses an AE combined with a classical bidimensional SOM grid (Fortuin et al., 2018), and *Self Organizing Map Variational Autoencoder* (SOM-VAE) (Forest et al., 2019), uses a discrete latent space and combines the SOM loss with the *Vector Quantised-Variational AutoEncoder* (VQ-VAE) loss (van den Oord et al., 2017).

3. Technical Contribution

The *Deep Clustering Self-Organizing Map with Relevance Learning* (DC-SOMRL) is proposed as a viable option of SOM-based model that can support batches of samples during training, and be easily integrated into *Deep Neural Networks* (DNN). We also added a neighborhood that works as a radial basis function with an exponential decay (γ) according to the level of activation of each node to an input pattern. It was developed based on some ideas from Bassani & Araujo (2015) and Braga & Bassani (2018).

In this work, we combined DC-SOMRL with an AE. The AE can minimize the reconstruction error by ensuring the hidden units capture the most relevant aspects of the data (Murphy, 2012). The AE network consists of two parts: an encoder function $\mathbf{h} = \mathbf{f}(\mathbf{x})$ and a decoder that produces a reconstruction $\mathbf{r} = \mathbf{g}(\mathbf{h})$. If the AE converges, it learns to set $\mathbf{g}(\mathbf{f}(\mathbf{x})) = \hat{\mathbf{x}}$. The AE learning process is described as minimizing Equation (1):

$$\mathcal{L}_{rec}(x, \hat{x}) = \|x - \hat{x}\| \tag{1}$$

*Equal contribution [1]Universidade Federal de Pernambuco, Brazil. Correspondence to: Heitor R. Medeiros <hrm@cin.ufpe.br>.

Proceedings of the LatinX in AI Research Workshopat at the 37^{th} International Conference on Machine Learning (ICML), 2020.

The complete forward pass consists of feeding the encoder with the input x, and then the latent representation is controlled by a sigmoid function. Finally, it is sent to DC-SOMRL and the decoder. The DC-SOMRL procedure consists of calculating the winner's prototype for the latent representations. To do so, an activation is computed as a radial basis function of a weighted distance. All prototypes with an activation above a defined threshold are used to compute the DC-SOMRL loss. It is expressed by Equation (2), which tries to minimize the weighted distance between the encoded feature ($h = f(x) = x_z$) and the winner prototype (c). Notice that the weights represent the relevance of each dimension (ω) on each prototype automatically learned by DC-SOMRL:

$$\mathcal{L}_{\text{DC-SOMRL}}(x_z) = \omega * \|x_z - c\| \tag{2}$$

With this information at hand, we combine both loss functions in order to backpropagate the error to the AE. The following Equation (3) illustrates the total loss that adds part of the Equation (2) to Equation (1) weighted by α:

$$\mathcal{L}_{Total}(x, \hat{x}) = \mathcal{L}_{rec}(x, \hat{x}) + \alpha \mathcal{L}_{\text{DC-SOMRL}}(x) \tag{3}$$

4. Experiments

The experiments evaluated the results from both quantitative and qualitative perspectives. For the first, two common metrics were used on MNIST dataset: *Normalized Mutual Information* (NMI) and Purity. For the latter, the quality of the latent representations was explored using the relations between the encoded features of the dataset and the prototypes of DC-SOMRL, and the top ten samples closest to each DC-SOMRL prototype.

The setup was based on Fortuin et al. (2018) to permit one-to-one comparisons. The models were trained for 1,000 iterations with a batch size of 256. The AE is consistent with the architecture proposed by Xie et al. (2016). The features in the latent space pass through a sigmoid function before being fed as input of ten features to DC-SOMRL. The maximum number of prototypes is defined to be 64 to be consistent with the 8x8 grids of previous SOM works. The alpha parameter varies between 0 and 1, but in this evaluation, it was fixed at 0.001 to better comparisons.

4.1. Quantitative Analysis

Table 1 shows the clustering quality in terms of NMI and purity of DC-SOMRL in comparison with DESOM and SOM-VAE. The results show that the proposed model achieves competitive results, while reducing the dimensionality from 784 (input space) to 10 (latent space). The model does not outperform DESOM. However, the results are close, showing a promising path to follow. It is important mentioning that the main idea is not necessarily to outperform in terms of metric value, but to build a solid representations which with meaningful topological properties.

Table 1. Evaluation metrics on MNIST dataset

METHOD	PUR	NMI
SOM-VAE	0.868	0.595
DESOM	0.939	0.657
DC-SOMRL	0.921	0.615

4.2. Qualitative Analysis

We analyzed the reconstructed image of each prototype and its top ten closest samples. In Figure 1, notice that the prototypes shown represent two ways of writing the number 2, and two ways of writing the number 7, respectively. Then, we evaluate a plot in t-SNE space (Maaten & Hinton, 2008) of the datapoints in relation to the cluster prototypes for the test samples (Figure 2). The edges between prototypes represent the topological neighborhood found. Notice that the model was able to create at least one cluster for each class, and the connections between nodes of different class regions make sense in a semantic perspective (e.g., the number 9 shares similarities with 4 and 7). These interesting results allows to observe characteristics of prototypes found and features shared by prototypes of different categories.

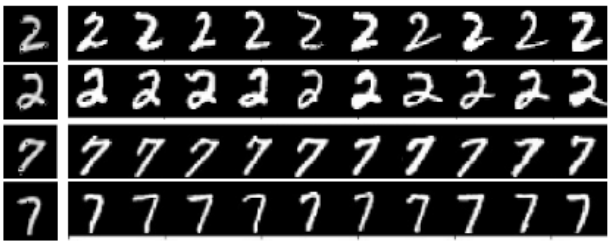

Figure 1. On the left, a column with the prototypes. On the right, top ten samples closest to the prototypes.

Acknowledgements

The authors would like to thank CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brazil, for supporting this research study, and FACEPE (Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco), Brazil, for financial support on the project #APQ-0880-1.03/14. Moreover, the authors also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.