ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Deep Embedded SOM: Joint Representation Learning and Self-Organization

Florent Forest[1,2], Mustapha Lebbah[1], Hanene Azzag[1] and Jérôme Lacaille[2]

1- Université Paris 13 - LIPN-UMR CNRS 7030 - France

2- Safran Aircraft Engines - France

Abstract. In the wake of recent advances in joint clustering and deep learning, we introduce the Deep Embedded Self-Organizing Map, a model that jointly learns representations and the code vectors of a self-organizing map. Our model is composed of an autoencoder and a custom SOM layer that are optimized in a joint training procedure, motivated by the idea that the SOM prior could help learning *SOM-friendly* representations. We evaluate SOM-based models in terms of clustering quality and unsupervised clustering accuracy, and study the benefits of joint training.

1 Introduction

After the successes of neural networks in supervised learning, recent research has focused on learning representations for unsupervised tasks, and cluster analysis in particular. Traditional algorithms tend to be ineffective on high-dimensional data where similarity metrics become meaningless. A solution is to first reduce dimensionality, then cluster in a low-dimensional space. This can be achieved with linear techniques as Principal Component Analysis, or more expressive models such as deep autoencoders. In this two-stage approach, we (1) optimize a pure information loss criterion between data points and their embeddings (generally via a reconstruction loss) (2) optimize a pure clustering criterion using a clustering algorithm. In contrast, *deep clustering* approaches [1–7] treat representation learning and clustering as a joint task and learn a *clustering-friendly* space preserving prior knowledge of cluster structure. See [8] for a review.

The self-organizing map (SOM) [9] achieves simultaneous clustering and visualization by projecting high-dimensional data onto a low-dimensional grid. The grid is composed of *units*, each one associated with a *prototype vector* from the original data space (also called *code vector*). The learning algorithm enforces a topology constraint, so that neighboring map units correspond to prototypes that are close in the original space, according to euclidean distance. We introduce the Deep Embedded SOM (DESOM), a model that jointly learns a SOM and a latent space that is more adapted to the SOM algorithm, according to some quality metric. Using the term coined by [4], we seek a *SOM-friendly* space. We represent the mappings between original and latent space by an autoencoder (AE). The prototypes lie in its intermediate space and are reconstructed for visualization and interpretation purposes. This approach resembles joint representation learning and clustering, but with an additional topology constraint, and has clear advantages: (1) Autoencoders yield meaningful low-dimensional representations that improve general performance of SOM. (2) Self-organization

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

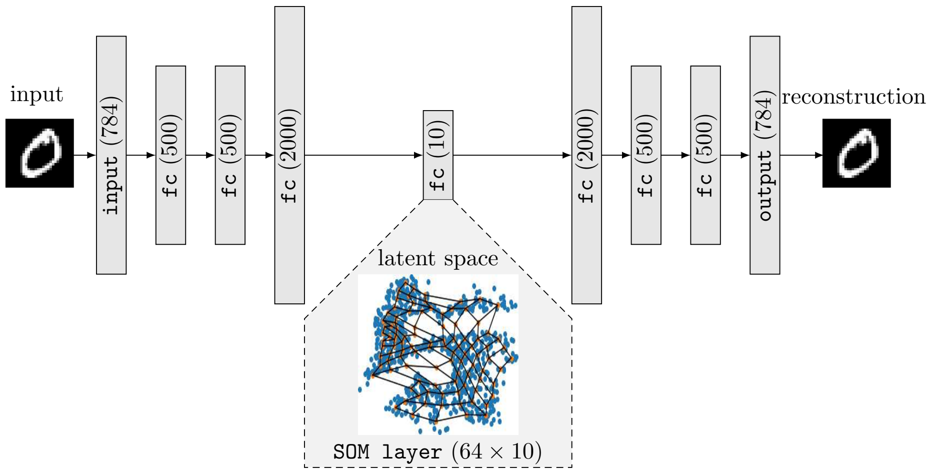

Fig. 1: DESOM architecture with an 8×8 map.

and representation learning can be achieved as a joint task, improving classification performance and cutting down training time.

To the best of our knowledge, the only similar work is the SOM-VAE [10]. SOM-VAE add a topology constraint to the VQ-VAE [11] loss function. However, SOM-VAE uses a discrete latent space, whereas in DESOM, the SOM is learned in a continuous latent space. Second, they use a fixed window neighborhood function, whereas we use a gaussian neighborhood with exponential radius decay. Finally, DESOM is based on a deterministic AE instead of a VAE.

2 Proposition

The proposed architecture is illustrated in Fig. 1. The self-organizing map is composed of K units, corresponding to prototype vectors $\{\mathbf{m}_k\}_{1\leq k\leq K}$. $\delta(\cdot,\cdot)$ is the topographic distance between two units on the map. We adopt a gaussian neighborhood function $\mathcal{K}^T(d) = e^{-d^2/T^2}$, depending on a temperature parameter T, controlling the radius of the neighborhood. Temperature decays exponentially at each training iteration. The encoder and decoder parameters are respectively noted $\mathbf{W_e}$ and $\mathbf{W_d}$. $\mathbf{z}_i = \mathbf{f}_{\mathbf{W_e}}(\mathbf{x}_i)$ is the embedding of a data point $\mathbf{x}_i$ in the intermediate latent space, and $\tilde{\mathbf{x}}_i = \mathbf{g}_{\mathbf{W_d}}(\mathbf{z}_i)$ is its reconstruction by the decoder. We define a loss function composed of two terms:

$$\mathcal{L}(\mathbf{W_e}, \mathbf{W_d}, \mathbf{m}_1, \ldots, \mathbf{m}_K, \chi) = \mathcal{L}_r(\mathbf{W_e}, \mathbf{W_d}) + \gamma \mathcal{L}_{som}(\mathbf{W_e}, \mathbf{m}_1, \ldots, \mathbf{m}_K, \chi) \quad (1)$$

The first term $\mathcal{L}_r$ is a least squares reconstruction loss. The second term $\mathcal{L}_{som}$ is the self-organizing map loss. It depends on the parameters $\{\mathbf{m}_k\}$ and the assignment function $\chi(\mathbf{z}) = \text{argmin}_k ||\mathbf{z} - \mathbf{m}_k||^2$. It is defined as follows:

$$\mathcal{L}_{som} = \sum_i \sum_{k=1}^{K} \mathcal{K}^T \left(\delta(\chi(\mathbf{f}_{\mathbf{W_e}}(\mathbf{x}_i)), k)\right) ||\mathbf{f}_{\mathbf{W_e}}(\mathbf{x}_i) - \mathbf{m}_k||^2 \tag{2}$$

Note that when the temperature approaches zero, the SOM loss becomes identical to a k-means loss, and our model thus converges towards DCN [4] or DKM [5] (at the end of their hyperparameter annealing):

$$\lim_{T \to 0} \mathcal{L}_{som} = \sum_i ||\mathbf{f}_{\mathbf{W_e}}(\mathbf{x}_i) - \mathbf{m}_{\chi(\mathbf{f}_{\mathbf{W_e}}(\mathbf{x}_i))}||^2 \tag{3}$$

The coefficient γ trades off between reconstruction loss and SOM loss. Our joint training procedure fixes χ between each optimization step, as it is non-differentiable. Thus, we can define constant weights $w_{i,k} \equiv \mathcal{K}^T \left(\delta(\chi(\mathbf{f}_{\mathbf{W_e}}(\mathbf{x}_i)), k)\right)$. Under this assumption, the partial derivatives of the loss function are easy to derive. The path of the gradients is illustrated on Fig. 2.

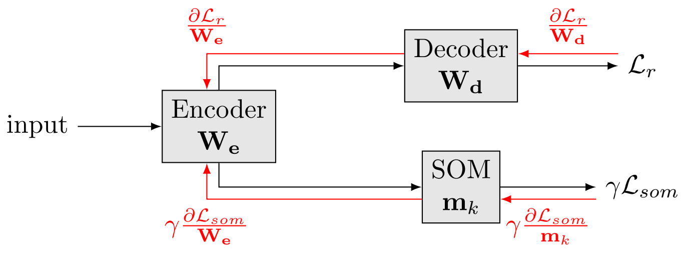
$\frac{\partial \mathcal{L}_r}{\mathbf{W_e}}$
$\frac{\partial \mathcal{L}_r}{\mathbf{W_d}}$
Decoder $\mathbf{W_d}$
$\mathcal{L}_r$
input
Encoder $\mathbf{W_e}$
SOM $\mathbf{m}_k$
$\gamma\mathcal{L}_{som}$
$\gamma\frac{\partial \mathcal{L}_{som}}{\mathbf{W_e}}$
$\gamma\frac{\partial \mathcal{L}_{som}}{\mathbf{m}_k}$

Fig. 2: Path of DESOM gradients.

3 Implementation

The code for DESOM[1] was implemented in Keras and partly inspired by IDEC[2]. The main novelty is a custom SOM layer, parameterized by a $K \times L$ matrix where K is the number of units and L is the latent dimensionality. The outputs are the pairwise squared euclidean distances between the input batch and the prototypes: this allows to express the SOM loss as a weighted sum, using the weight terms $w_{i,k}$. The whole training procedure is detailed in algorithm 1.

4 Experiments

We conducted experiments on classification benchmark datasets described in Tab. 1. SOM-based models are evaluated in two ways: (1) Quantitative assessment of clustering quality using purity and NMI metrics. We also evaluate their classifying power (without removing the topology constraint) by performing k-means clustering on the resulting prototypes, and measure unsupervised

[1]`https://github.com/FlorentF9/DESOM`
[2]`https://github.com/XifengGuo/IDEC`

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

```
Input: training set; SOM topology; $T_{max}$; $T_{min}$; iterations; batchSize
Output: AE weights $\mathbf{W_e}$, $\mathbf{W_d}$; SOM code vectors $\{\mathbf{m}_k\}$
Initialize AE (Glorot uniform) and SOM parameters (random samples) ;
for iter = 1, ..., iterations do
    $T \leftarrow T_{max}\,(T_{min}/T_{max})^{iter/iterations}$ ;
    Load next training batch ;
    Predict SOM pairwise distances on batch and compute weights $w_{i,k}$ ;
    Train DESOM on batch ;
end
```

Algorithm 1: DESOM training procedure

Dataset	description	examples	classes	dimension
MNIST [12]	images (digits)	70000	10	784
Fashion-MNIST [13]	images (clothing)	70000	10	784
REUTERS-10k [14]	text (TF-IDF)	10000	4	2000

Table 1: Dataset statistics

clustering accuracy. (2) Qualitative assessment of the self-organization of the resulting map, as our model must be topology-preserving.

4.1 Baselines and compared models

We evaluate **minisom**, a standard SOM[3]; **kerasom**, our Keras implementation of a SOM (DESOM with identity encoder); **AE+minisom** and **AE+kerasom**, with AE and SOM trained separately; and finally **DESOM**. We also include SOM-VAE and k-means (keeping in mind that it lacks self-organization).

4.2 Training parameters

All models are trained for 10000 iterations with a batch size of 256 using the Adam optimizer [15]. Initial and final temperatures are $T_{max} = 10.0$ and $T_{min} = 0.1$. The AE is symmetric with a $[500, 500, 2000, 10]$ encoder architecture, and the map has 8×8 units (to compare with previous work). Empirically, we fixed $\gamma = 0.001$ across all experiments, without cross-validation to remain in a fully unsupervised setting. Large values of γ lead to degenerate solutions for the autoencoder, due to the SOM loss being easier to optimize than the reconstruction loss. Moreover, the model is not very sensitive to the value of γ as long as it stays in this order of magnitude. Pretraining is beneficial in most deep clustering approaches, either layer-wise [2,4] [6], RBM [1] or end-to-end [5]. However, initializing DESOM with pretrained AE weights does not lead to any improvement, because the SOM loss produces strong gradients at the beginning of training that disturb encoder weights and cancel out pretraining. Thus, we use no pretraining.

[3]`https://github.com/JustGlowing/minisom`

	MNIST		Fashion-MNIST		REUTERS-10k	
Method	pur	nmi	pur	nmi	pur	nmi
k-means ($k = 64$)	0.842	0.571	0.716	0.512	0.892	0.427
minisom (8×8)	0.637	0.430	0.646	0.494	0.690	0.230
kerasom (8×8)	0.826	0.565	0.717	0.512	0.697	0.324
AE+minisom (8×8)	0.871	0.616	0.734	0.531	0.690	0.235
AE+kerasom (8×8)	**0.939**	**0.661**	**0.764**	**0.539**	0.777	0.306
SOM-VAE (8×8)	0.868	0.595	0.739	0.520	-	-
DESOM (8×8)	**0.939**	0.657	0.752	**0.538**	**0.849**	**0.381**

Table 2: Purity and NMI (average on 10 runs). Best result and results with no significant difference (p-value > 0.05) in bold.

Method	MNIST	Fashion-MNIST	REUTERS-10k
k-means ($k = \#$classes)	58.34	56.45	59.37
AE+kerasom (8×8) + km	**76.06**	44.87	36.61
DESOM (8×8) + km	**76.11**	**56.02**	**57.18**

Table 3: Unsupervised clustering accuracy (%) (average on 10 runs).

4.3 Quantitative and qualitative results

Clustering quality results (Tab. 2) confirm the benefits of reducing dimensionality with an AE. The overall best-performing models are AE+kerasom and DESOM. Joint training does not consistently improve purity and NMI on the first two datasets but remains competitive, performs better on REUTERS-10k by a fair margin and is faster to train. Interestingly, kerasom achieves better than minisom. The same discovery was made by [10], suggesting that Adam improves SOM training. Finally, DESOM consistently outperforms its direct competitor, SOM-VAE. On the classification task (Tab. 3), DESOM consistently achieves the best performance, demonstrating that joint training with a SOM prior has enabled to learn a *SOM-friendly* representation for subsequent classification.

Visualizations of decoded DESOM prototypes (Fig. 3) display well-organized regions corresponding to different classes and smooth transitions between them. In addition, code images learned by the standard SOM algorithm are blurred because of vector averaging in original space, which is not the case in DESOM.

5 Conclusion and future work

DESOM is the first approach that jointly trains an autoencoder and a SOM in a continuous latent space. The learned map is self-organized, competitive in terms of clustering quality and requires no pretraining. On the classification task, it outperforms similar methods. Future work will include a more thorough investigation of hyperparameters, and extensions to the variational or adversarial frameworks to learner richer representations and provide a generative model.

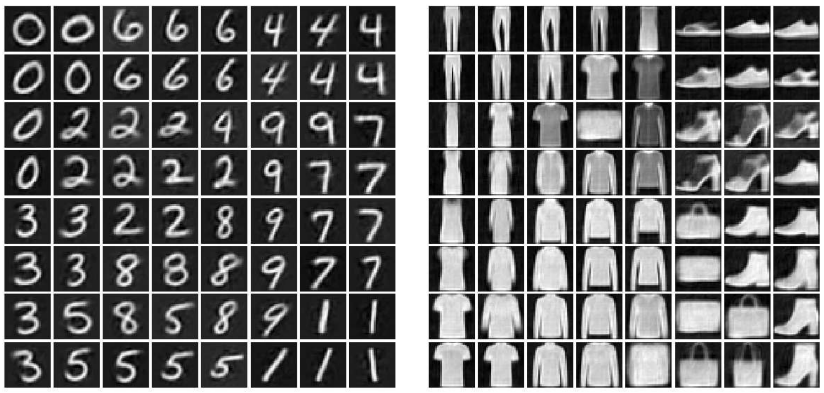
Fig. 3: DESOM map of MNIST (left) and Fashion-MNIST (right).

References

[1] C Song, Y Huang, F Liu, Z Wang, and L Wang. Deep auto-encoder based clustering. *Intelligent Data Analysis*, 18(6):S65–S76, 2014.

[2] J Xie, R Girshick, and A Farhadi. Unsupervised Deep Embedding for Clustering Analysis. In *ICML*, volume 48, 2015.

[3] X Guo, L Gao, X Liu, and J Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[4] B Yang, X Fu, N D Sidiropoulos, and M Hong. Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. In *ICML*, 2016.

[5] M M Fard, T Thonet, and E Gaussier. Deep k-Means: Jointly Clustering with k-Means and Learning Representations. 2018.

[6] Z Jiang, Y Zheng, H Tan, B Tang, and H Zhou. Variational Deep Embedding : An Unsupervised and Generative Approach to Clustering. In *IJCAI*, pages 1965–1972, 2017.

[7] W Harchaoui, P Mattei, A Alamansa, and C Bouveyron. Wasserstein Adversarial Mixture Clustering. 2018.

[8] E Aljalbout, V Golkov, Y Siddiqui, and D Cremers. Clustering with Deep Learning: Taxonomy and New Methods. 2018.

[9] T Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[10] V Fortuin, M Hüser, F Locatello, H Strathmann, and G Rätsch. Deep Self-Organization: Interpretable Discrete Representation Learning on Time Series. 2018.

[11] A van den Oord, O Vinyals, and K Kavukcuoglu. Neural Discrete Representation Learning. In *NIPS*, 2017.

[12] Y LeCun, L Bottou, Y Bengio, and P Haffner. Gradient-based Learning Applied to Document Recognition. In *Proceedings of the IEEE*, 1998.

[13] H Xiao, K Rasul, and R Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017.

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

[15] D P Kingma and J L Ba. Adam: A Method For Stochastic Optimization. In *ICLR*, 2015.