



HAL
open science

Carte SOM profonde : Apprentissage joint de représentations et auto-organisation

Florent Forest, Mustapha Lebbah, Hanene Azzag, Jérôme Lacaille

► To cite this version:

Florent Forest, Mustapha Lebbah, Hanene Azzag, Jérôme Lacaille. Carte SOM profonde : Apprentissage joint de représentations et auto-organisation. CAp2020: Conférence d'Apprentissage, Jun 2020, Vannes, France. hal-02859997

HAL Id: hal-02859997

<https://hal.science/hal-02859997v1>

Submitted on 8 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Carte SOM profonde : Apprentissage joint de représentations et auto-organisation

Florent Forest^{*1,2}, Mustapha Lebbah¹, Hanene Azzag¹, et Jérôme Lacaille²

¹LIPN (CNRS UMR 7030), Université Sorbonne Paris Nord

²Safran Aircraft Engines

Résumé

Dans la lignée des récentes avancées en apprentissage profond de représentations pour le clustering, ce travail (précédemment publié en anglais) présente le modèle DESOM (Deep Embedded SOM), combinant l'apprentissage non supervisé de représentations et d'une carte auto-organisée de Kohonen (SOM). Le modèle, composé d'un auto-encodeur et d'une couche SOM, est optimisé conjointement, afin de régulariser l'espace latent et améliorer la performance de la carte SOM. Nous évaluons les performances de classification et de visualisation ainsi que les bénéfices de l'apprentissage joint.

Mots-clef : carte auto-organisée, clustering, apprentissage profond, auto-encodeur.

1 Introduction

Après les succès des réseaux de neurones en apprentissage supervisé, de récents travaux se sont tournés vers l'apprentissage de représentations pour des tâches non supervisées, en particulier le clustering. Les algorithmes classiques sont basés sur des mesures de similarité qui se révèlent inefficaces en grande dimension. Une solution est de réduire d'abord la dimensionalité des données, puis d'effectuer le clustering dans un espace de basse dimension. La réduction est obtenue par des méthodes linéaires comme l'ACP ou non linéaires comme les auto-encodeurs profonds. Cette approche consiste, dans un premier temps, à optimiser un critère de perte d'information (en général via une erreur de reconstruction), puis dans un deuxième temps, optimiser un critère de partitionnement des données via un algorithme de clustering. À l'inverse, les approches dites de deep clustering [SHL⁺14, XGF16, GGLY17, YFSH16, JZT⁺17, HMAB18] considèrent ces deux étapes comme

une tâche jointe, et tentent d'apprendre un espace latent intégrant la structure de clusters.

La carte auto-organisée (SOM) [Koh82] permet de classifier et visualiser des données multivariées sur un grille de faible dimension, constituée de neurones chacun associé à un vecteur prototype de l'espace d'origine. Par contrainte topologique de l'apprentissage, des neurones voisins sur la carte correspondent à des prototypes proches dans l'espace de grande dimension.

Ce travail présente DESOM (Deep Embedded SOM) [FLAL19], un modèle apprenant conjointement une SOM et un espace latent de projection. La transformation entre les espaces d'origine et latent est représentée par un auto-encodeur (AE). Les prototypes sont situés dans l'espace latent et décodés au moment de la visualisation et interprétation. Cette approche possède certains avantages : (1) les représentations riches apprises via les auto-encodeurs améliorent les performances de SOM, (2) l'auto-organisation et l'apprentissage de représentations s'effectue comme une tâche jointe, améliorant la qualité de la classification et réduisant la durée d'entraînement. À notre connaissance, le seul travail similaire est SOM-VAE [FHL⁺19], avec d'importantes différences au niveau de l'espace latent (discret, alors que celui de DESOM est continu), du voisinage (fixe alors que nous utilisons un voisinage gaussien avec décroissance du rayon) et de l'auto-encodeur (variationnel au lieu de déterministe).

2 Proposition

L'architecture proposée est illustrée figure 1. La carte SOM est composée de K neurones, associés aux prototypes $\{\mathbf{m}_k\}_{1 \leq k \leq K}$. Soit $\delta(\cdot, \cdot)$ la distance topologique entre deux neurones sur la carte. Nous adoptons une fonction voisinage gaussienne $\mathcal{K}^T(d) = e^{-d^2/T^2}$, dont le rayon est contrôlé par un paramètre de tempéra-

*forest@lipn.univ-paris13.fr

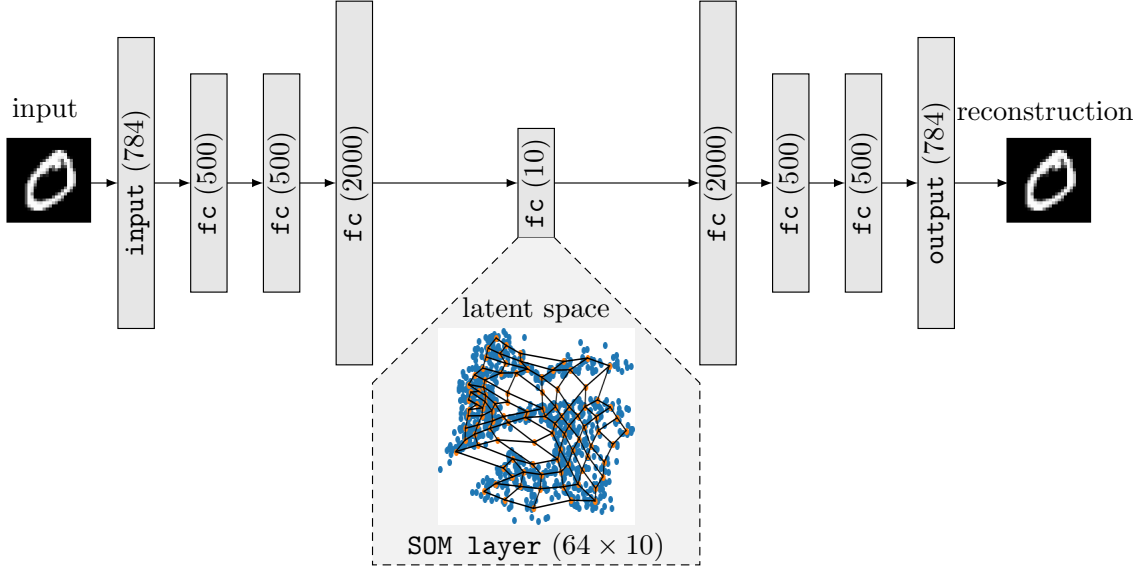


FIGURE 1 – Architecture DESOM avec une carte 8×8 .

ture T . Ce dernier décroît exponentiellement durant l'apprentissage. Les paramètres des réseaux encodeur et décodeur sont notés respectivement \mathbf{W}_e et \mathbf{W}_d . $\mathbf{z}_i = \mathbf{f}_{\mathbf{W}_e}(\mathbf{x}_i)$ représente la projection d'un point \mathbf{x}_i sur l'espace latent, et sa reconstruction via le décodeur est notée $\tilde{\mathbf{x}}_i = \mathbf{g}_{\mathbf{W}_d}(\mathbf{z}_i)$. Nous définissons une fonction coût composée de deux termes :

$$\mathcal{L}(\mathbf{W}_e, \mathbf{W}_d, \mathbf{m}_1, \dots, \mathbf{m}_K, \chi) = \mathcal{L}_r(\mathbf{W}_e, \mathbf{W}_d) + \gamma \mathcal{L}_{som}(\mathbf{W}_e, \mathbf{m}_1, \dots, \mathbf{m}_K, \chi)$$

Le premier terme \mathcal{L}_r est une erreur quadratique moyenne de reconstruction. Le second terme \mathcal{L}_{som} est la fonction coût de SOM. Celle-ci dépend des $\{\mathbf{m}_k\}$ et de la fonction d'affectation $\chi(\mathbf{z}) = \operatorname{argmin}_k \|\mathbf{z} - \mathbf{m}_k\|^2$.

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

Le coefficient γ définit le poids relatif du coût de reconstruction et de SOM. Notre procédure jointe fixe l'affectation χ (non différentiable) entre chaque étape d'optimization. Ainsi, il est possible de définir des coefficients constants $w_{i,k} \equiv \mathcal{K}^T(\delta(\chi(\mathbf{f}_{\mathbf{W}_e}(\mathbf{x}_i)), k))$, grâce auxquels les dérivées partielles de la fonction coût s'expriment simplement. Le parcours des gradients est illustré figure 2.

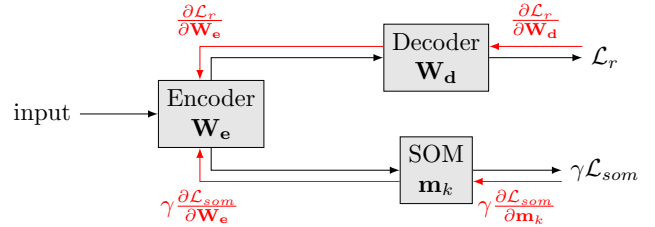


FIGURE 2 – Parcours des gradients de DESOM.

3 Implémentation

DESOM est implémenté sous le framework Keras et le code est open-source¹. La principale nouveauté est une couche SOM, paramétrée par une matrice $K \times L$, avec K le nombre de neurones et L la dimensionnalité latente. Cette couche calcule les distances euclidiennes entre les données en entrée et les prototypes, ce qui permet d'exprimer la fonction coût SOM comme une somme pondérée par les poids $w_{i,k}$. La procédure d'entraînement est détaillée dans l'algorithme 1.

4 Expériences

Les expériences utilisent des jeux de données classiques en classification automatique : MNIST, Fashion-MNIST (images) et Reuters-10k (texte). Les clusterings obtenus sont évalués de manière quantitative par

1. <https://github.com/FlorentF9/DESOM>

Input: jeu d'apprentissage; topologie SOM;
 T_{max} ; T_{min} ; $iterations$; $batchSize$;
Output: poids de l'AE \mathbf{W}_e , \mathbf{W}_d ; vecteurs
référents SOM $\{\mathbf{m}_k\}$
Initialiser AE (Glorot uniform) et SOM
(échantillon aléatoire);
for $iter = 1, \dots, iterations$ **do**
 $T \leftarrow T_{max} (T_{min}/T_{max})^{iter/iterations}$;
 Charger le prochain batch d'entraînement ;
 Calculer les distances à la carte SOM et les
 poids $w_{i,k}$;
 Effectuer une étape de descente de gradient
 sur le batch ;
end
Algorithm 1: Procédure d'entraînement de DESOM

la pureté et le NMI (information mutuelle normalisée). La discrimination des classes est évaluée en classifiant les prototypes par k -means, puis en mesurant la précision de classification non supervisée par rapport aux classes cibles. Qualitativement, nous vérifions l'organisation topologique et la qualité visuelle de la carte.

Nous comparons **minisom**, une implémentation standard de SOM²; **kerasom**, notre implémentation de SOM sous Keras (i.e. DESOM sans auto-encodeur); **AE+minisom** et **AE+kerasom**, avec AE and SOM entraînés séparément; finalement, **DESOM**. Nous incluons aussi SOM-VAE.

4.1 Paramètres d'entraînement

Chaque modèle est entraîné 10000 itérations avec une taille de batch de 256 et l'optimiseur Adam. Les températures initiale et finale sont $T_{max} = 8.0$ and $T_{min} = 0.1$. L'architecture de l'auto-encodeur ([500, 500, 2000, 10]) et la taille de la carte (8×8) sont choisis pour être comparables avec de précédents travaux. L'hyper-paramètre γ est fixé empiriquement à 0.001, sans validation croisée pour rester dans un cadre non supervisé. Une valeur trop élevée mène à des solutions dégénérées pour l'encodeur, dû au fait que le coût SOM est plus facilement optimisé que l'erreur de reconstruction. De plus, le modèle est peu sensible à la valeur de γ tant qu'elle reste dans cet ordre de grandeur. Un pré-entraînement de l'AE est bénéfique dans la plupart des approches de clustering profond. Cependant, la fonction coût SOM produit de forts gradients en début d'entraînement, perturbant l'encodeur. Par conséquent, aucun pré-entraînement n'est utilisé ici.

2. <https://github.com/JustGlowing/minisom>

4.2 Résultats quantitatifs et qualitatifs

Les résultats (table 1) confirment l'intérêt de la réduction de dimension par AE. Les modèles les plus performants sont AE+kerasom et DESOM. L'apprentissage joint n'améliore pas systématiquement la pureté et le NMI mais reste à minima comparable, et plus rapide à entraîner. Étonnamment, kerasom obtient de meilleurs résultats que minisom, suggérant que l'optimisation par descente de gradient stochastique améliore l'entraînement de SOM, découverte faite aussi par [FHL⁺19]. Enfin, DESOM surclasse son concurrent direct SOM-VAE. En terme de discrimination des classes (table 2), DESOM obtient les meilleurs scores, montrant que l'apprentissage joint a permis d'apprendre un espace latent régularisé plus adapté pour la classification des prototypes de la carte.

Les visualisations des prototypes décodés de DESOM (figure 3) forment des régions continues et bien organisées correspondant aux différentes classes. De plus, les images prototypes apprises par une SOM standard sont floues, à cause du calcul de moyennes dans l'espace original, ce qui n'est pas le cas pour DESOM.

5 Conclusion et travaux futurs

DESOM est la première approche qui entraîne conjointement un auto-encodeur et une carte SOM dans un espace latent continu. La carte obtenue est organisée, compétitive en terme de clustering, et ne nécessite pas de pré-entraînement. Des travaux sont encore nécessaires pour étudier les hyper-paramètres, et le modèle pourrait être étendu en une variante variationnelle afin d'obtenir un modèle génératif.

Références

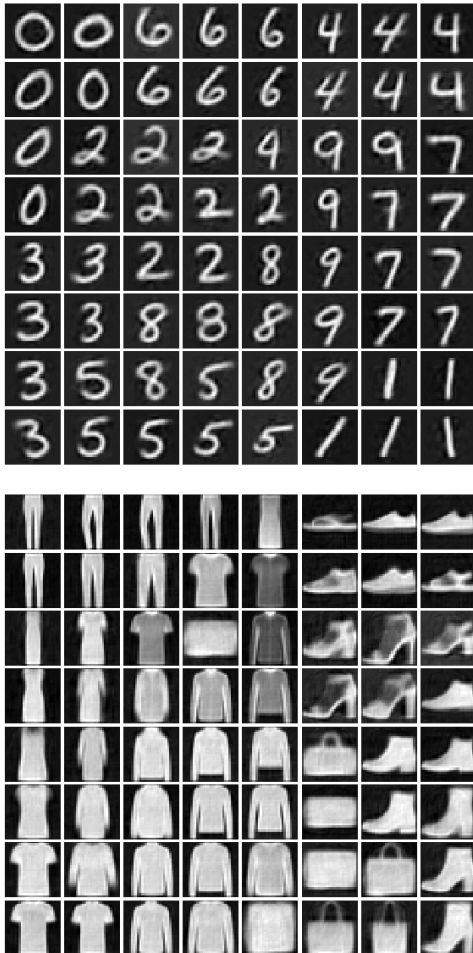
- [FHL⁺19] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. Deep Self-Organization : Interpretable Discrete Representation Learning on Time Series. In *ICLR 2019*, 2019.
- [FLAL19] Florent Forest, Mustapha Lebbah, Hanane Azzag, and Jérôme Lacaïlle. Deep Embedded SOM : Joint Representation Learning and Self-Organization. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2019)*, pages 1–6, 2019.

Méthode	MNIST		Fashion-MNIST		REUTERS-10k	
	pur	nmi	pur	nmi	pur	nmi
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 1 – Purité et NMI (moyenne sur 10 entraînements). Meilleur résultat et résultats statistiquement équivalents (p -value > 0.05) en gras.

Méthode	MNIST	Fashion-MNIST	REUTERS-10k
AE+kerasom (8×8) + k -means	76.06	44.87	36.61
DESOM (8×8) + k -means	76.11	56.02	57.18

TABLE 2 – Précision de classification non supervisée (%) (moyenne sur 10 entraînements).



- [GGLY17] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.
- [HMAB18] Warith Harchaoui, Pierre-Alexandre Mattei, Andrés Alamansa, and Charles Bouveyron. Wasserstein Adversarial Mixture Clustering. 2018.
- [JZT+17] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational Deep Embedding : An Unsupervised and Generative Approach to Clustering. In *IJCAI*, pages 1965–1972, 2017.
- [Koh82] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1) :59–69, 1982.
- [SHL+14] Chunfeng Song, Yongzhen Huang, Feng Liu, Zhenyu Wang, and Liang Wang. Deep auto-encoder based clustering. *Intelligent Data Analysis*, 18(6) :S65–S76, 2014.
- [XGF16] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised Deep Embedding for Clustering Analysis. In *ICML 2016*, volume 48, 2016.
- [YFSH16] Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards K-means-friendly Spaces : Simultaneous Deep Learning and Clustering. In *ICML*, 2016.

FIGURE 3 – Cartes DESOM de MNIST (gauche) et Fashion-MNIST (droite).