Accepted ICASSP 2021 Toronto Canada

LEARNING A SPARSE GENERATIVE NON-PARAMETRIC SUPERVISED AUTOENCODER

Michel Barlaud, Fellow IEEE and Frederic Guyard

Laboratoire I3S CNRS, Cote d'Azur University Sophia Antipolis, France
Orange Labs Sophia Antipolis, France

ABSTRACT

This paper concerns the supervised generative non parametric autoencoder. Classical methods are based on variational non supervised autoencoders (VAE). Variational autoencoders encourage the latent space to fit a prior distribution, like a Gaussian. However, they tend to draw stronger assumptions for the data, often leading to higher asymptotic bias when the model is wrong.
In this paper, we relax the parametric distribution assumption in the latent space and we propose to learn a non-parametric data distribution of the clusters in the latent space. The network encourages the latent space to fit a distribution learned with the labels instead of the parametric prior assumptions. We have built a network architecture that uses the labels to compute the latent space. Thus we define a global criterion combining classification and reconstruction loss. In addition, we have proposed a $\ell_{1,1}$ regularization which has the advantage of sparsifying the network and improving the clustering. Finally we propose a tailored algorithm to minimize the criterion with constraint. We demonstrate the effectiveness of our method using the popular image dataset MNIST and two biological datasets.

1. RELATED WORKS

In many applications (Image analysis and biomedical research), the objective is to design algorithms to classify, generate data and select features to decrypt high-dimensional data. Autoencoders were introduced in the field of neural networks decades ago and their most efficient application was dimensionality reduction [1, 2]. A discriminative model maps feature points of a high dimensional space in $\mathbb{R}^d$ to labels in a low dimensional latent space in $\mathbb{R}^l$. Generative models map feature points of a low dimensional space $\in \mathbb{R}^l$ to a high dimensional latent space in $\mathbb{R}^d$. Recently, deep generative models have been used to learn generator functions that map points from a low-dimensional latent space, to a high-dimensional data space. These generative models, which include variational autoencoders (VAEs) [3] and generative adversarial networks (GANs) [4, 5], can generate high-fidelity output samples that look like real-world data.
Generative modeling is attractive for many reasons: i) Modelization of the latent space: Generative models express causal relations. ii) Generative models were used in semi-supervised learning settings, to improve classification [3, 6, 7, 8, 9].

Let's recall that VAE networks encourage the latent space to fit a prior distribution, like a Gaussian. These classical priors in the latent space are chosen for their computational simplicity rather than their compatibility with the latent structure and thus can lead to inaccurate latent low-dimensional representations of data. The classical VAE mixes the points of the clusters because the Gaussian prior encourages all the points to be centered at the origin. In order to cope with this issue some recent papers have proposed latent spaces with more complex distributions (e.g., hyperspheres [10], and mixtures of Gaussians [11]) on the latent vectors, but they are non-adaptive and unfortunately may not match the specific data distribution.
Contractive autoencoders add an explicit regularizer in their objective loss function that forces the model to learn a function that is robust to noisy variations of input values. A popular regularization method which sparsifies the weights of the neural network is the Absolute Shrinkage and Selection Operator (LASSO) formulation [12]. This classical ℓ_1 penalization ensures regularization and sparsity. Various structured constraints such as "group LASSO" and "exclusive LASSO" have been proposed in the framework of LASSO for inducing structured sparsity.
In this work, we relax the parametric distribution assumption in the latent space to learn a non-parametric data distribution of clusters. Our network encourages the latent space to fit a distribution learned with the clustering labels rather than a parametric prior distribution.
Moreover, we propose a constrained regularization approach that takes advantage of a available efficient projection algorithms for the ℓ_1 constraint [13], convex constraints [14] and structured constraints $\ell_{2,1}$ [15, 16] and $\ell_{1,2}$ [16].
We point out the following specific contributions:

- We create a network architecture that incorporates the labels into an autoencoder latent space. This enables us to compute a latent space structured distribution instead of a prior Gaussian distribution.
- We define a global criterion combining classification and reconstruction loss. In addition, we propose a $\ell_{1,1}$ regularization for which advantages are sparsity induction and an improvement in clustering.

- We propose a tailored algorithm to minimize the criterion with constraint.
- We propose a generative model using the real distribution of the data in the latent space.

2. PROPOSED APPROACH: NON-PARAMETRIC SUPERVISED AUTOENCODER FRAMEWORK

2.1. Criterion

Let X be the dataset in $\mathbb{R}^d$, as a $m \times d$ data matrix made of m line samples and d features $x_1, \ldots, x_m$. Let $y_i = j, j \in \{1, \cdots, k\}$ be the label, indicating that the sample x_i belongs to the j-th cluster. Projecting the data in the lower dimension latent space in $\mathbb{R}^l$ is crucial to be able to separate them accurately. In this paper we propose to use a deep neural network autoencoder framework.
Let's recall that the encoder (or discriminative part) of the autoencoder map features points of a high dimensional space in $\mathbb{R}^d$ to a low dimensional latent space in $\mathbb{R}^l$ and that the decoder maps feature points of a low dimensional space $\in \mathbb{R}^l$ to a high dimensional latent space in $\mathbb{R}^d$.
Figure 1 depicts the main constituent blocks of our proposed approach. We have added to our autoencoder block a "soft max" block to calculate the classification loss.
Let Z $\in \mathbb{R}^l$, the latent space, $\widehat{X} \in \mathbb{R}^d$ the reconstructed data (Fig 1) and W the weights of the neural network.
The goal is to compute the weights W minimizing the total loss which depends on both the classification loss and the reconstruction loss. Hence our strategy for training the various encoders and decoders is based on following requirements.

1. First, we want to classify data in the latent space

$$Loss(W) = \phi(Z, Y) \tag{1}$$

2. Second, we also want to minimize the difference between the reconstructed and the original data

$$Loss(W) = \psi(\widehat{X} - X) \tag{2}$$

3. Third, we want a sparse autoencoder network. To this end we also introduce a $\ell_{1,1}$ constrained regularization loss.

$$\|W\|_1{}^1 \leq \eta \tag{3}$$

Thus we propose to minimize the following criterion to design the auto-encoder:

$$Loss(W) = \phi(Z, Y) + \lambda\psi(\widehat{X} - X) \text{ s.t. } \|W\|_1 \leq \eta. \tag{4}$$

Where the classification loss ϕ is a function of the latent variable and labels. We use the Cross Entropy Loss for the classification loss function. We use the robust Smooth ℓ_1 (Huber) Loss [17] as reconstruction loss function ψ. Note that the dimension of the latent space is the number of clusters.

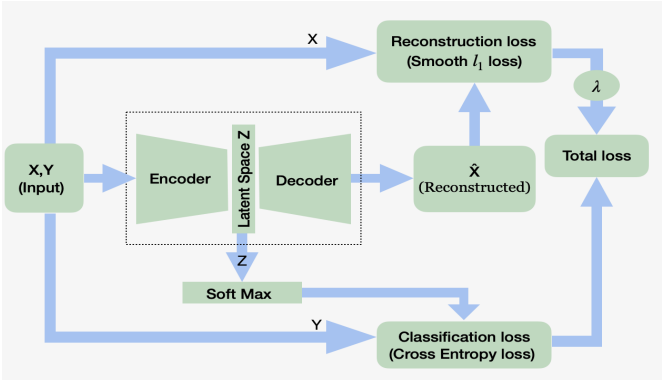

Fig. 1. Autoencoder framework

We use Markov chain Monte Carlo $(MCMC)$ methods for obtaining a sequence of random samples from a probability distribution in the latent space. Among the $MCMC$ methods we refer to the classical Metropolis–Hastings algorithms or the Gibbs sampling method [18, 19]. The density of points is estimated using a kernel method. A step is rejected if the density of the candidate location falls under a threshold. Then we use the decoder as a generative model. Thus we fit the real distribution in the latent space instead of making a random draw with a classical Gaussian assumption as in VAE.

2.2. Algorithms

We propose the following algorithm: we first compute the radius t_i and then project the rows using the ℓ_1 adaptive constraint t_i (See [20] for more details):
Following the work by Frankle and Carbin [21] further developed by [22] which proposed a double descent algorithm as follows: after training a network, set all weights smaller than some threshold to zero, rewind the rest of the weights to their initial configuration, and then retrain the network from this starting configuration but keeping the zero weights frozen (untrained). We replace the thresholding by our $\ell_{1,1}$ projection and devise the following algorithm:

Algorithm 1 Projection on the $\ell_{1,1}$ norm—$proj_{\ell_1}(V, \eta)$ is the projection on the ℓ_1-ball of radius η

Input: V, η
$t := \text{proj}_{\ell_1}((\|v_i\|_1)_{i=1}^d, \eta)$
for $i = 1, \ldots, d$ **do**
  $w_i := \text{proj}_{\ell_1}(v_i, t_i)$
end for
Output: W

Algorithm 2 Projection on the $\ell_{1,1}$ norm—$\text{proj}_{\ell_1}(V, \eta)$ is the projection on the ℓ_1-ball of radius η, $\nabla\phi(W, M_0)$ is the masked gradient with binary mask M_0, and f is the ADAM optimizer, γ is the learning rate

Input: $W*, \gamma, \eta$
for $n = 1, \ldots, N(epochs)$ **do**
  $V \leftarrow f(W, \gamma, \nabla\phi(W))$
end for
$t := \text{proj}_{\ell_1}((\|v_i\|_1)_{i=1}^d, \eta)$
for $i = 1, \ldots, d$ **do**
  $w_i := \text{proj}_{\ell_1}(v_i, t_i)$
end for
Output: W, M_0
Input: $W*$
for $n = 1, \ldots, N(epoch)$ **do**
  $W \leftarrow f(W, \gamma, \nabla\phi(W, M_0))$
end for
Output: W

3. EXPERIMENTAL RESULTS

We have modified the PyTorch framework to implement our sparse learning method using a constraint approach. The losses are averaged across observations for each mini-batch. We chose the ADAM optimizer [23], as the standard optimizer in PyTorch. We used the Cross Entropy Loss for the classification loss and the Smooth ℓ_1 Loss (Huber Loss) for the reconstruction loss. We evaluated clustering in the latent space using the silhouette criterion [24].
We used a linear fully connected network (LFC) with an input layer of d neurons, 4 hidden layers followed by a ReLU activation function and a latent layer of dimension k.
We evaluated our method on the popular MNIST dataset and two biological datasets.

3.1. MNIST dataset

MNIST dataset [25] contains 28×28 grey-scale images of handwritten digits of 10 classes (from 0 to 9). This dataset consists on a training set of 60,000 instances and a test set of 10,000 instances. We provide a visual evaluation of the data in the latent space for MNIST. The latent dimension is $k > 2$ so we project the data on a 2D plot using PCA.

Figure 2 illustrates that the distribution in the latent space is not gaussian for MNIST. Figure 3 shows the images decoded (output of the autoencoder). We computed 10 random samples in the latent space using Metropolis-Hastings algorithm and generated the corresponding virtual images using the decoder as shown in Figure 4.

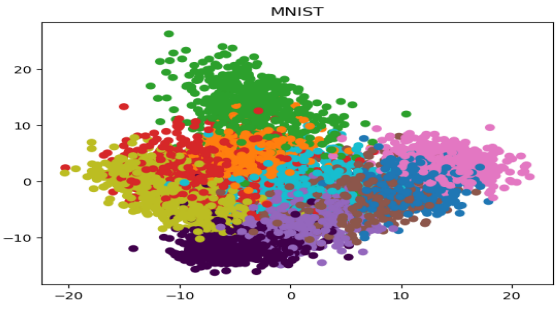

Fig. 2. MNIST dataset Clustering in the latent space.

Fig. 3. MNIST dataset: Output of the autoencoder

Fig. 4. MNIST dataset : Reconstructed datas using the Metropolis-Hastings algorithm in the latent space

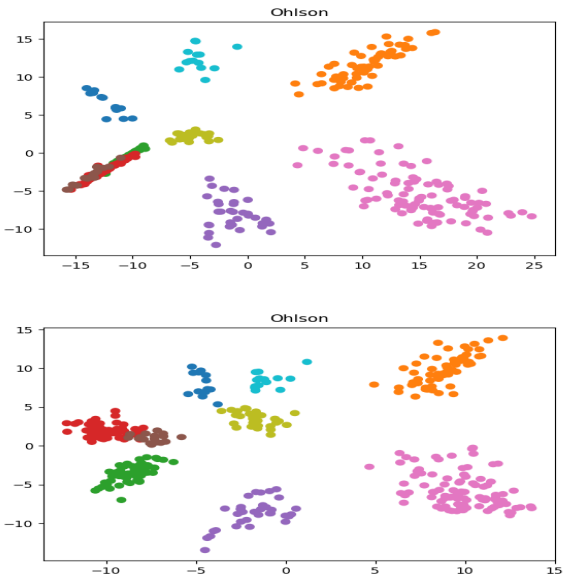

Fig. 5. Ohlson dataset d=532, k=9. Our new autoencoder. Top : without regularisation, Bottom : with regularization $\eta = 1000$.

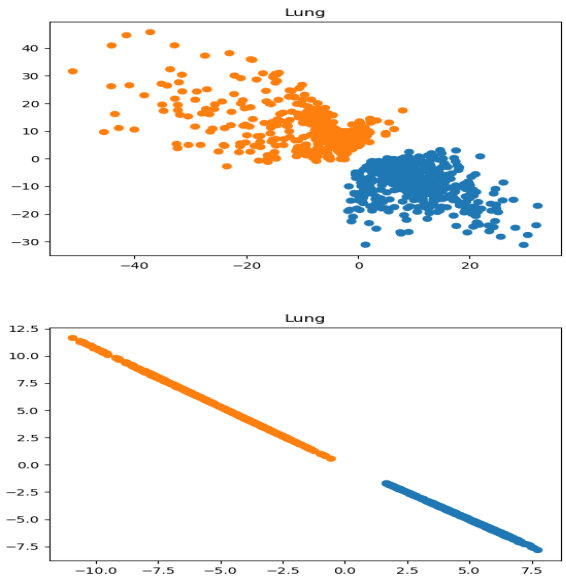

Fig. 6. Lung dataset m=1005, d=2900, k=2. Our new autoencoder. Top : without regularisation, Bottom : with regularization $\eta = 300$.

3.2. Biomedical dataset

The Ohlson dataset is a single cell RNA seq dataset used by [26] for clustering evaluation with m=382 samples, d=532 features and k=9 clusters. The lung dataset [27] is a metabolomic dataset with 1005 samples, 2944 features and 2 clusters. These are urine samples obtained from two groups of patients, one group has a lung cancer, the other is a control group.

η	200	400	1000	2000	10000
ℓ_1	0.67	0.718	0.738	0.718	0.619
$\ell_{1,1}$	0.608	0.705	0.741	0.723	0.618

Table 1. Ohlson dataset: Clustering evaluation using silhouette criterion.

η	20	100	200	300	400	10000
ℓ_1	0.75	0.761	0.744	0.696	0.674	0.588
$\ell_{1,1}$		0.593	0.723	0.735	0.713	0.554

Table 2. Lung dataset Clustering evaluation using silhouette criterion.

Figures 5 and 6 show that the distribution in the latent space for the Ohlson dataset and Lung dataset are not Gaussian. The Figures 5 and 6 and Tables 1 and 2 show that clustering using regularization ℓ_1 or $\ell_{1,1}$ outperforms clustering without regularization on both dataset Ohlson and Lung Dataset. Silhouette with $\ell_{1,1}$ regularization is slightly better than ℓ_1 on Ohlson data set while ℓ_1 is slightly better than $\ell_{1,1}$ on Lung. However the main benefit of $\ell_{1,1}$ is to reduce significantly the computational task [20].

4. CONCLUSION

In this paper, we propose a network architecture that use the labels to compute the latent space. This enables us to compute a latent space structured distribution instead of a prior gaussian distribution and devise a generative model using the real distribution of the data in the latent space. We define a global loss criterion combining classification and reconstruction loss and propose a tailored algorithm to minimize this global loss criterion with constraint. In addition, we propose an $\ell_{1,1}$ regularization who has two main advantages : a structured sparsity induction and an improvement of the clustering. We have illustrated our generative model using Metropolis–Hastings algorithm in the latent space. Experiments demonstrate the effectiveness of our method on MNIST dataset and two biological datasets.

The authors would thank internships Zhiyun Xu and Axel Gustovic for processing simulations and Professor Thierry Pourcher for providing the Lung dataset.

5. REFERENCES

[1] G. Hinton and R. Zemel, “Autoencoders, minimum description length and helmholtz free energy.” in *Advances in neural information processing systems*, 1994, pp. 3–10.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016, vol. 1.

[3] D. Kingma and M. Welling, "Auto-encoding variational bayes," *International Conference on Learning Representation*, 2014.

[4] 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*, 2014, pp. 2672–2680.

[5] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *arXiv cs.LG /1606.03498*, 2016.

[6] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," *arXiv stat.ML /1602.02282*, 2016.

[7] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[8] J. Snoek, R. Adams, and H. Larochelle, "On non parametric guidance for learning autoencoder representations," ser. Proceedings of Machine Learning Research, vol. 22. PMLR, 2012, pp. 1073–1080.

[9] S. Jadon and A. A. Srinivasan, "Improving siamese networks for one-shot learning using kernel-based activation functions," in *Data Management, Analytics and Innovation*, N. Sharma, A. Chakrabarti, V. E. Balas, and J. Martinovic, Eds. Springer Singapore, 2021, pp. 353–367.

[10] T. R. Davidson, L. Falorsi, N. D. Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," *arXiv stat.ML /1804.00891*, 2018.

[11] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv cs.LG 1611.02648*, 2016.

[12] T. Hastie, R. Tibshirani, and M. Wainwright, "Statistcal learning with sparsity: The lasso and generalizations," *CRC Press*, 2015.

[13] L. Condat, "Fast projection onto the simplex and the l1 ball," *Mathematical Programming Series A*, vol. 158, no. 1, pp. 575–585, 2016.

[14] M. Barlaud, W. Belhajali, P. Combettes, and L. Fillatre, "Classification and regression using an outer approximation projection-gradient method," vol. 65, no. 17, 2017, pp. 4635–4643.

[15] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l2, 1-norm minimization," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, ser. UAI '09. AUAI Press, 2009, pp. 339–348.

[16] M. Barlaud, A. Chambolle, and J.-B. Caillau, "Classification and feature selection using a primal-dual method and projection on structured constraints," *International Conference on Pattern Recognition, Milan*, pp. 6538–6545, 2020.

[17] P. J. Huber, "Robust statistics. 1981."

[18] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[19] L. Martino, J. Read, and D. Luengo, "Independent doubly adaptive rejection metropolis sampling within gibbs sampling," *IEEE Transactions on Signal Processing*, vol. 63, no. 12, pp. 3123–3138, june 2015.

[20] M. Barlaud and F. Guyard, "Learning sparse deep neural networks using efficient structured projections on convex constraints for green ai," *International Conference on Pattern Recognition, Milan*, pp. 1566–1573, 2020.

[21] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019.

[22] H. Zhou, J. Lan, R. Liu, and J. Yosinski, "Deconstructing lottery tickets: Zeros, signs, and the supermask," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 3597–3607.

[23] D. Kingma and J. Ba, "a method for stochastic optimization." *International Conference on Learning Representations*, pp. 1–13, 2015.

[24] P. Rousseuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, November 1987.

[25] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*.

[26] A. Klimovskaia, D. Lopez-Paz, L. Bottou, and M. Nickel, "Poincaré maps for analyzing complex hierarchies in single-cell data," *bioRxiv*, 2019.

[27] E. Mathé *et al*, "Noninvasive urinary metabolomic profiling identifies diagnostic and prognostic markers in lung cancer," *Cancer research*, vol. 74, no. 12, p. 3259—3270, June 2014.