HAL open science

A Non-Parametric Supervised Autoencoder for discriminative and generative modeling

Michel Barlaud, Frederic Guyard

To cite this version:

Michel Barlaud, Frederic Guyard. A Non-Parametric Supervised Autoencoder for discriminative and generative modeling. ICASSP, 2022, Toronto, Canada. hal-02937643

HAL Id: hal-02937643

https://hal.science/hal-02937643v1

Submitted on 14 Sep 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.

A Non-Parametric Supervised Autoencoder for discriminative and generative modeling

Michel Barlaud
Laboratoire I3S
CNRS, Cote d'Azur University
Sophia Antipolis, France
michel.barlaud@i3s.unice.fr

Frederic Guyard
Orange Labs
Sophia Antipolis, France
frederic.guyard@orange.com

Zhiyun Xu
Laboratoire I3S and TIRO
Cote d'Azur University
Sophia Antipolis, France
zhiyun.xu@etu.univ-cotedazur.fr

***Abstract*—This paper deals with supervised discriminative and generative modeling. Classical methods are based on variational autoencoders or supervised variational autoencoders encourage the latent space to fit a prior distribution, like a Gaussian. However, they tend to make stronger assumptions on 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 incorporates the labels into an autoencoder latent space to create discriminative and generative models. Thus we define a global criterion combining classification and reconstruction loss. In addition, we have proposed a $\ell_{1,1}$ regularization which advantages are a faster convergence of the algorithm and an improvement of the clustering. Finally we propose a tailored algorithm to minimize the criterion with constraint. We demonstrate the effectiveness of our method on two popular image datasets (MNIST and Fashion MNIST) and two biological datasets.**

I. 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. Classification of high-dimensional data is known to suffer from the curse of dimensionality [42]. In order to overcome this issue, the main idea of early works on *Linear Discriminant Analysis* (LDA) [12], [14], [2] was to project data into a lower dimensional space. Many techniques have been proposed in the machine learning literature for dimensionality reduction to discover various aspects of structure from data [47], [44] [7]. Deep neural networks were used for dimensionality reduction[25].

Autoencoders were introduced in the field of neural networks decades ago and their most efficient application were dimensionality reduction [26], [20]. Autoencoders were successfully used for denoising [51] to extract useful features and lossy image coding [48].

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 $\mathbb{R}^d$. Note that they tend to make stronger assumptions on the data often leading to higher asymptotic bias when the model is wrong [3].

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) [30], [43] and generative adversarial networks (GANs) [21], [45], can generate high-fidelity output samples that look like real-world data.

Generative modeling is attractive for many reasons:

1) Modelization of the latent space: Generative models express causal relations,
2) Generative models were used in semi-supervised learning settings, to improve classification [30], [46], [31],
3) Generative models are a potential solution for privacy issues concerning biomedical data [9], [40], [39].

Generative models offer unique opportunities in domains where either data collection is technically difficult or expensive or where personal data privacy is critical. From these perspectives, their ability to provide synthetic data is highly relevant to the biomedical and healthcare domains. GANs have been used since their creation to provide various types of synthetic data in these domains. In [53] GANs are used to generate synthetic laboratory test time series for the prediction of drug effects. MedGAN [9] uses GANs to generate realistic discrete medical patient records. In [16], VAE are used to automatically optimize molecules in order to improve their physico-chemical properties. Conditioned GANs [38] are used to automatically design molecules having a high probability of inducing a given transcriptomic profile. DermGan [19] proposed a GAN based tool to create synthetic dermoscopic images.

Supervised autoencoders are autoencoders whose loss function are augmented with the loss of a task (typically a classification) performed using the representation layer. Relatively few studies have been devoted to these autoencoders. Let us mention however the work of [57] who uses an autoencoder whose goal is to maximize the distances of classification centroids on the representation layer. Let us mention also the generalization result for supervised linear auto-encoders, with unsupervised regularizers [33].

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 [11], and mixtures of Gaussians [13]) on the latent vectors, but they are non-adaptive and unfortunately may not match the specific data distribution.

All these methods result in a problem of criteria optimization. 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. Moreover this regularizer takes into account the over-parametrization of the neural network (in practice, relatively few network weights are necessary to accurately learn data features). A popular regularization method which sparsifies the weights of the neural network is the Absolute Shrinkage and Selection Operator (LASSO) formulation [49], [18], [24]. 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. A proximal gradient has been used with an ℓ_1 constraint [27] while the “Group LASSO” constraint has been used in [55] and in [1]. Neuron sparsification [54], is realized combining the “group LASSO” ($\ell_{2,1}$) constraint and an additional “Exclusive LASSO” ($\ell_{1,2}$) constraint enforces neurons to fit disjoint sets of features. However, the computational time needed for the processing of the corresponding hyper-parameter is expensive (see [23]).

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 an available efficient projection algorithms for the ℓ_1 constraint [10], [41], convex constraints [4] and structured constraints $\ell_{2,1}$ [35], [5] and $\ell_{1,2}$ [5].

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 propose a generative model using the real distribution of the data in the latent space.
- We develop an autoencoder model based on discriminative and generative models. Thus we define a global criterion combining classification and reconstruction loss. In addition, we propose a $\ell_{1,1}$ regularization whose advantages are a faster convergence of the algorithm and an improvement of the clustering.
- We propose a tailored algorithm to minimize the criterion with constraint.

II. Proposed approach: Non-Parametric supervised Autoencoder framework

Modelisation

Let X be the dataset in $\mathbb{R}^d$, as a $m \times d$ data matrix made of m line samples $x_1, \ldots, x_m$. Let $y_i = j, j \in .[1...k]$ 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. Note that the soft max is a projection on the simplex (a positive part of the ℓ_1 ball).

Let XL $\in \mathbb{R}^l$, the latent space, XR $\in \mathbb{R}^d$ 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 various requirements.

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

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

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

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

Note that these individual losses by themselves do not entirely promote reasonable stable results and therefore, we also introduce a constrained regularization loss. Thus we propose to minimize the following criterion to design the auto-encoder:

$$Loss(W) = \phi(XL, Y) + \lambda\psi(XR - X) \text{ s.t. } \|W\|_1 \leq \eta. \tag{3}$$

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 [28] as reconstruction loss function ψ. The size of the latent space is the number of clusters.

We can compute the center of the clusters and generate them in the high dimensional space using the decoder. We can 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 and Metropolis–Hastings algorithms or the Gibbs sampling method [8], [36]. Then we use the decoder as a generative model. Thus we fit

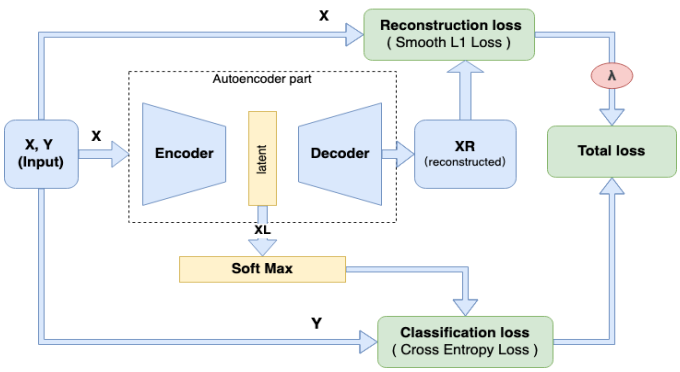

Fig. 1. Autoencoder framework

the real distribution in the latent space instead of making a random draw with a classical Gaussian assumption as in VAE.

A tailored Algorithm

Let's recall that the classical $\ell_{2,2}$ induced operator norm constraint of $A - B$ in the ℓ_2 domain with the ℓ_2 co-domain does not induce any sparsity.

$$\| A - B \|_{2,2} = \left(\sup_{\|\mathbf{x}\|_2 = 1} \| (A - B) \cdot \mathbf{x} \|_2 \right) \tag{4}$$

Thus, we propose a constrained regularization approach using a new projection on the $\ell_{1,1}$ constraint instead of a group LASSO $\ell_{2,1}$ or en exclusive Lasso $\ell_{1,2}$ constraint. The induced operator norm of $A - B$ in the ℓ_1 domain with the ℓ_1 co-domain is given by:

$$\| A - B \|_{1,1} = \left(\sup_{\|\mathbf{x}\|_1 = 1} \| (A - B) \cdot \mathbf{x} \|_1 \right) \tag{5}$$

We recall that $\ell_{1,1}$ is computed as the maximum ℓ_1 norm of a columns of $A - B$. Using this property, 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: Following

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

the work by Frankle and Carbin [17] further developed by [56] 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 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

III. 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 [29], 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 compared our method to Variational Autoencoder (VAE) and classical TSNE for the biomedical datasets [50].

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.

In our approach, we provide a visual evaluation of the data and of the cluster centers in the latent space. If the latent dimension $k > 2$, we project the data and the cluster centers on a 2D plot using PCA. We compute the matrix distance between centers which we can visualize using PCA. We provide two kinds of synthetic generated data i) of the centroid of the cluster and ii) of random using the Metropolis algorithm.

We evaluated our method on two classical image datasets and two biological datasets.

We selected the popular MNIST dataset [34] containing 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. Fashion-MNIST [52] is a dataset of Zalando's article images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Fashion-MNIST is to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking of machine learning algorithms. Fashion-MNIST and MNIST share the same image size and structure of training and testing splits.

Image dataset

We first study the two image datasets.

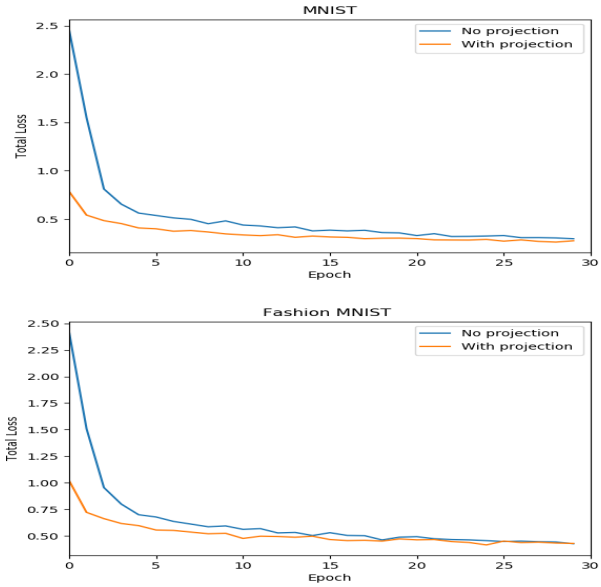

Fig. 2. MNIST dataset: Convergences of our algorithm on MNIST (TOP) and Fashion MNIST (Bottom) are very similar.

Fig 2, Fig 15 and Fig 21 show that regularization using projections on the $\ell_{1,1}$ constraint improves the convergence of the algorithm. Figure 3 and Figure 4 illustrate that the distributions in the latent space are not gaussian, for MNIST and Fashion MNIST respectively. Figure 5 and Figure 10 shows that the regularization increases the distance between the centers.

It can be noted that the reconstructed images (Figure 7 and Figure 12) are less noisy than original images (Figure 6 and Figure 11).

We computed the centroids of the clusters in the latent space and then we generated the corresponding virtual images using the decoder as shown in Figures 8 and 13. We computed 10 random samples in the latent space using Metropolis algorithm

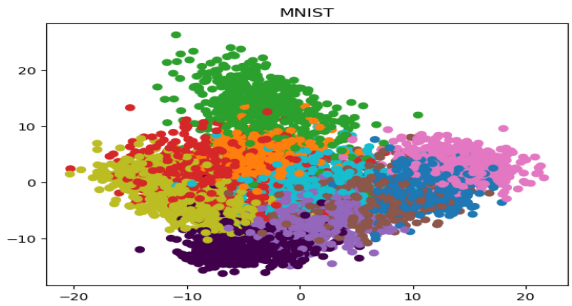

Fig. 3. MNIST dataset Clustering in the latent space using Huber loss.

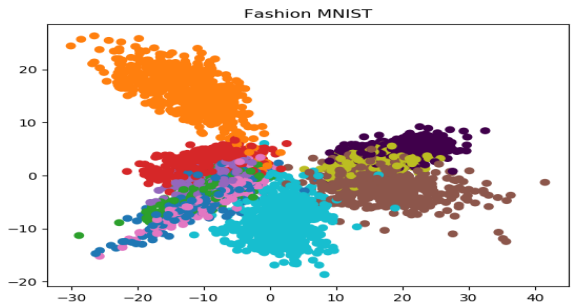

Fig. 4. Fashion MNIST dataset Clustering in the latent space using Huber loss.

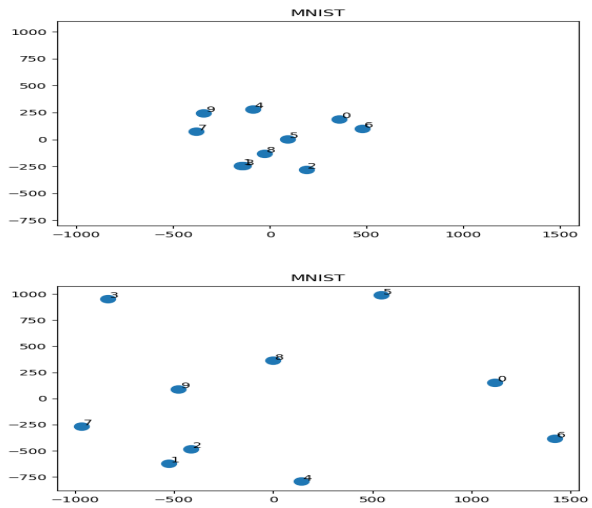

Fig. 5. MNIST dataset: Top : Centers without regularization, Bottom : Centers with projections on the $\ell_{1,1}$ constraint

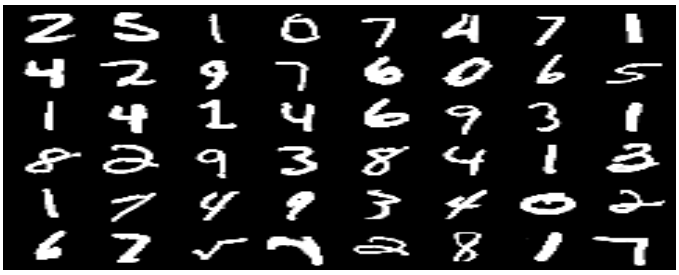

Fig. 6. Original MNIST dataset.

Fig. 7. Reconstructed MNIST dataset.

Fig. 8. Reconstructed MNIST dataset : cluster centers using the decoder.

Fig. 9. MNIST dataset : Reconstructed using the Metropolis algorithm in the latent space

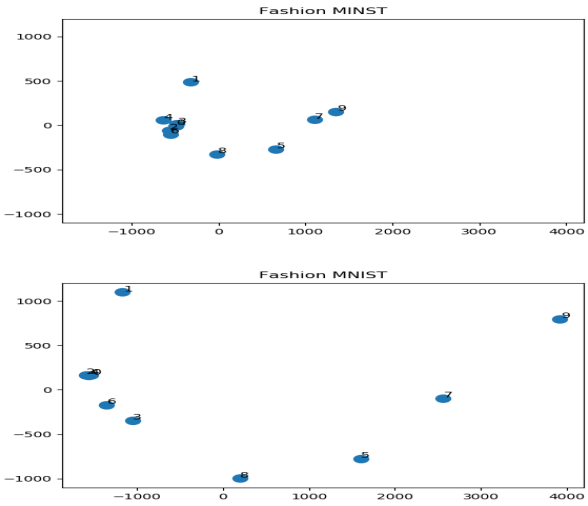

Fig. 10. Fashion MNIST dataset: Top : Centers without regularization, Bottom : Centers with projections on the $\ell_{1,1}$ constraint

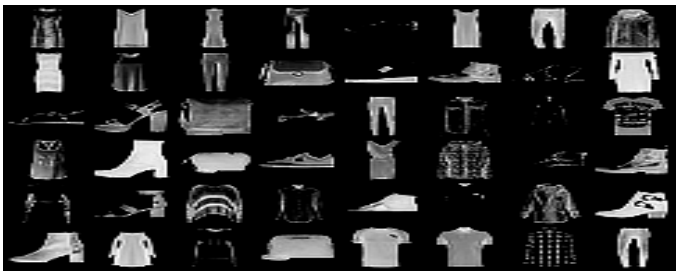

Fig. 11. Original Fashion MNIST dataset.

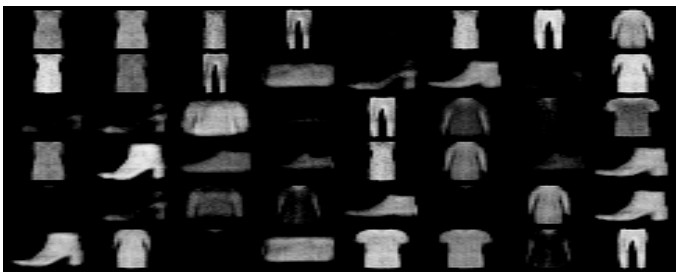

Fig. 12. Reconstructed Fashion MNIST dataset.

Fig. 13. Reconstructed Fashion MNIST dataset : cluster centers using the decoder.

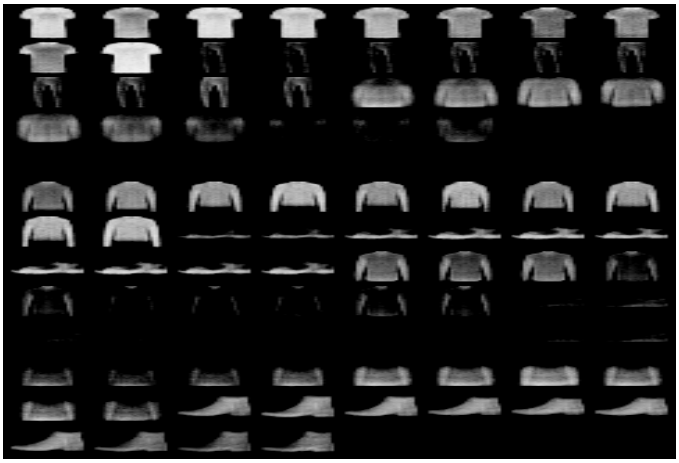

Fig. 14. Fashion MNIST dataset : Reconstructed using the Metropolis algorithm in the latent space

and generated the corresponding virtual images using the decoder as shown in Figure 9 and Figure 14.

Biomedical dataset: The lung dataset [37] 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. The Ohlson dataset [15] is a single cell RNA seq dataset used by [32] for clustering evaluation with m=382 samples, d=532 features and k=9 clusters.

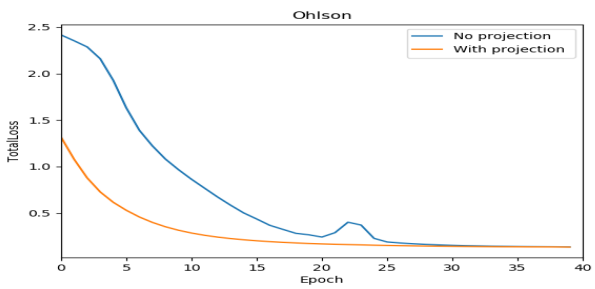

Fig. 15. Ohlson dataset m=382, d=532, k=9. Convergence without projections and Convergence with projections

Figure 17 and Figure 16 show that the distribution in the latent space for the Ohlson dataset is not Gaussian and that our NPautoencoder outperforms classical VAE and TSNE. Figure 17 shows that by using Huber loss the clustering is improved. Figure 18 shows that regularization increases the distance between the cluster centers.

On the Lung dataset, Figures 19 and 20 show that our NPautoencoder outperforms classical TSNE method. Fig 20 shows that $\ell_{1,1}$ regularization improves cluster separability.

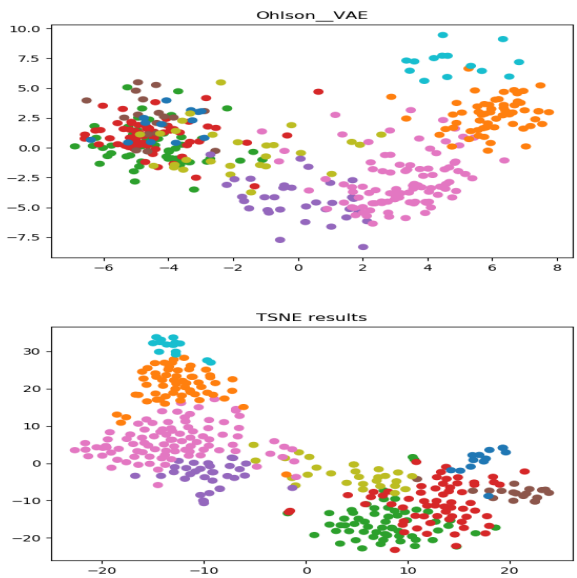

Fig. 16. Ohlson dataset d=532, k=9. Top : VAE with BCE Loss, Bottom : TSNE

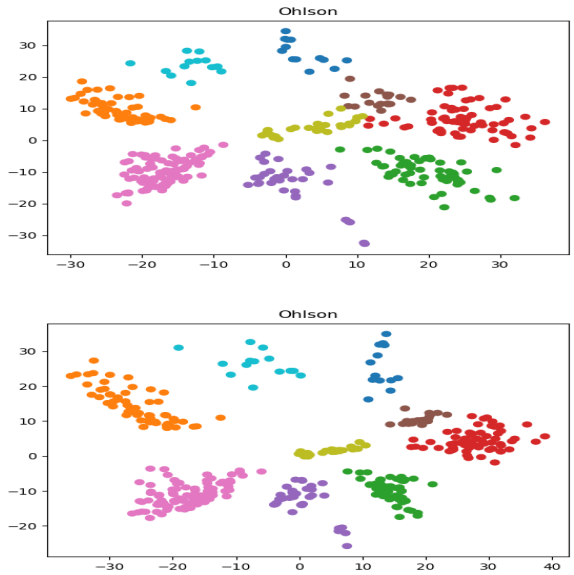

Fig. 17. Ohlson dataset d=532, k=9, Our new NPautoencoder. Top : with MSE loss, Bottom : with Huber loss.

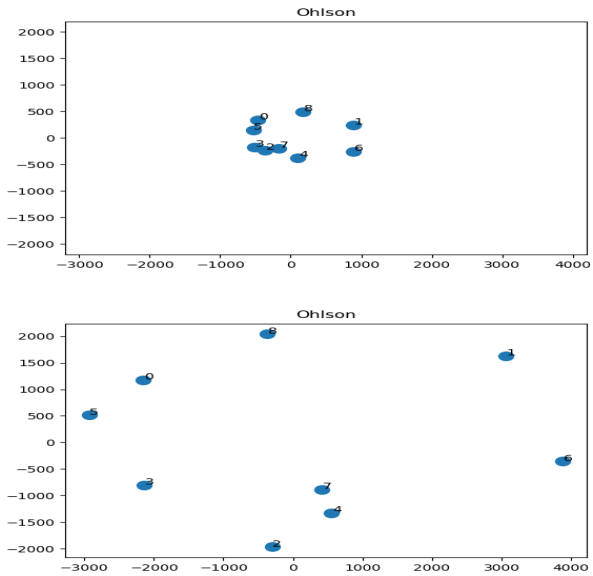

Fig. 18. Ohlson dataset: Top : Centers without regularization, Bottom : Centers with projections on the $\ell_{1,1}$ constraint.

IV. Discussion

In this paper we have limited the choice of the classification loss and reconstruction loss to cross entropy and Huber loss because they are considered to be the best of the state of the art.

An exhaustive study of various projection methods has shown that the $\ell_{1,1}$ projection method is the best in terms of memory and calculation cost [6]. Thus, we have only considered the $\ell_{1,1}$ regularization.

We have illustrated our generative model using

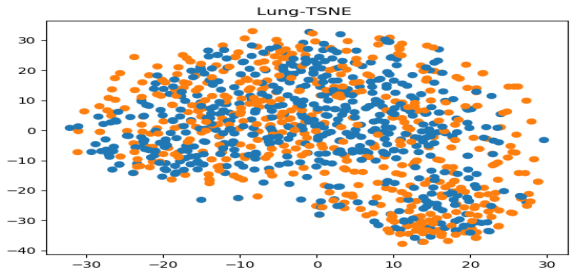

Fig. 19. Lung dataset m=1005 d=2900, k=2. TSNE.

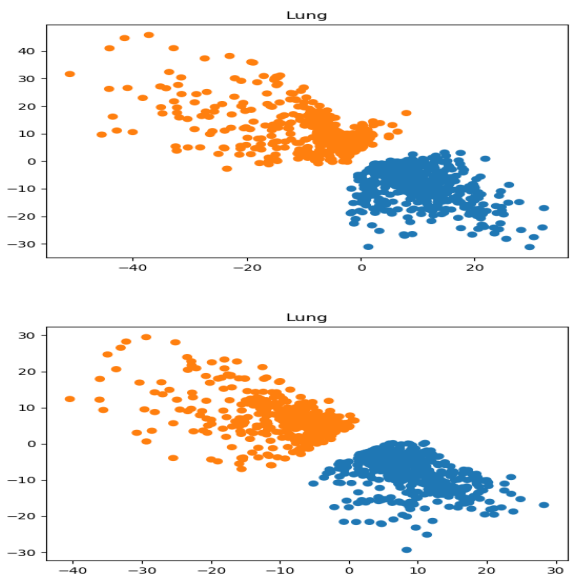

Fig. 20. Lung dataset m=1005 d=2900, k=2. Our new NPautoencoder. Top : without regularisation, Bottom : with regularization

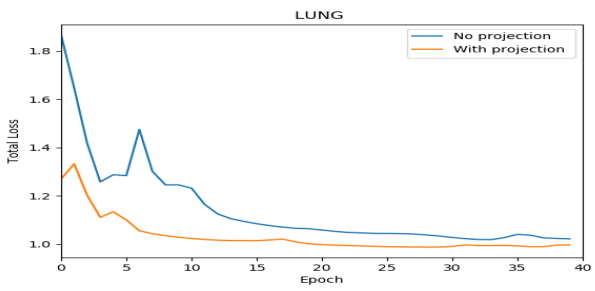

Fig. 21. Lung dataset m=1005, d=2900, k=2. Convergence without projections and Convergence with projections

Metropolis–Hastings algorithm in the latent space. Contrary to the VAE approach, our method perfectly fits the distribution in the latent space.

Note that for all experiments the $\ell_{1,1}$ regularization improves the convergence and separability of the clusters. Algorithm 1 for projecting a matrix on the constraint $\ell_{1,1}$ can be extended to the projection of a tensor on the constraint $\ell_{1,1,1}$. Thus extension of our method to image processing using convolutional Neural network architecture such as simplenet [22] is straightforward. Note that extending our method to discrete clinical data is straightforward.

V. Conclusion

In this paper, we propose 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 and devise a generative model using the real distribution of the data in the latent space. We develop a supervised auto-encoder model based on disciminative and generative models. 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 faster convergence of the algorithm and an improvement of the clustering. Experiments demonstrate the effectiveness of our method on two popular image datasets (MNIST and Fashion MNIST) and two biological datasets.

References

[1] Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, pages 2270–2278, 2016.

[2] Francis R Bach and Zaïd Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In *Advances in Neural Information Processing Systems*, pages 49–56, 2008.

[3] Arindam Banerjee. An analysis of logistic models: Exponential family connections and online performance. *Proceedings of the 7th SIAM International Conference on Data Mining*, pages 204–215, 2007.

[4] Michel Barlaud, Wafa Belhajali, Patrick Combettes, and Lionel Fillatre. Classification and regression using an outer approximation projection-gradient method. volume 65, pages 4635–4643, 2017.

[5] Michel Barlaud, Antonin Chambolle, and Jean-Baptiste Caillau. Robust supervised classification and feature selection using a primal-dual method. *arXiv cs.LG/1902.01600*, 2019.

[6] Michel Barlaud and Frédéric Guyard. Learning sparse deep neural networks using efficient structured projections on convex constraints for green ai. *HAL Id: hal-02556382*, 2020.

[7] Yoshua Bengio and Martin Monperrus. Non-local manifold tangent learning. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 129–136. MIT Press, 2005.

[8] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[9] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter E. Stewart, and Jimeng Sun. Generating multi-labal discrete patient records using generative adversarial networks. *arXiv, cs.LG, 1703.06490v3*, january 2018.

[10] Laurent Condat. Fast projection onto the simplex and the l1 ball. *Mathematical Programming Series A*, 158(1):575–585, 2016.

[11] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *arXiv stat.ML /1804.00891*, 2018.

[12] Fernando de la Torre and Takeo Kanade. Discriminative cluster analysis. *ICML 06 Proceedings of the 23rd international conference on Machine learning, Pittsburgh, Pennsylvania, USA*, 2006.

[13] Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. 2016.

[14] Chris Ding and Tao Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 521–528, 2007.

[15] Ohlson et al. Single-cell analysis of mixed-lineage states leading to a binary cell fate choice. *Nature 537,698*, 2016.

[16] Rafael Gómez-Bombarelli et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.*, (4):268–276, 2018.

[17] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

[18] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization path for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–122, 2010.

[19] Amirata Ghorbani, Vivek Natarajan, David Coz, and Yuan Liu. Dermgan: Synthetic generation of clinical skin images with pathology. pages 155–170, 2019.

[20] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*, volume 1. MIT press, 2016.

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[22] Seyyed Hossein Hasanpour, Mohammad Rouhani, Mohsen Fayyaz, and Mohammad Sabokrou. Lets keep it simple, using simple architectures to outperform deeper and more complex architectures. *arXiv preprint arXiv:1608.06037*, 2016.

[23] Trevor Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[24] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistcal learning with sparsity: The lasso and generalizations. *CRC Press*, 2015.

[25] Geoffrey. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[26] Zemel Richard Hinton, Geoffrey. Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10. 1994.

[27] Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 304–320, 2018.

[28] Peter J Huber. Robust statistics. 1981.

[29] Durk Kingma and Jimmy Ba. a method for stochastic optimization. *International Conference on Learning Representations , pages=1–13, year=2015,*.

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

[31] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3581–3589. Curran Associates, Inc., 2014.

[32] Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *bioRxiv*, 2019.

[33] Lei Le, Andrew Patterson, and Martha White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In *Advances in Neural Information Processing Systems 31*, pages 107–117. Curran Associates, Inc., 2018.

[34] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

[35] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient l2, 1-norm minimization. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 339–348. AUAI Press, 2009.

[36] Luca Martino, Jesse Read, and David Luengo. Independent doubly adaptive rejection metropolis sampling within gibbs sampling. *IEEE Transactions on Signal Processing*, 63(12):3123–3138, june 2015.

[37] E. Mathé *et al*. Noninvasive urinary metabolomic profiling identifies diagnostic and prognostic markers in lung cancer. *Cancer research*, 74(12):3259—3270, June 2014.

[38] Oscar Méndez-Lucio, Benoit Naillif, Djork-Arné Clevert, David Rouquié, and Joerg Wichard. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature Communication*, 11(10), 2020.

[39] Malekzadeh Mohammad, Clegg Richard G., and Haddadi Hamed. Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis. *arXiv / 1710.06564*, 2018.

[40] Phan NhatHai, Wang Yue, Wu Xintao, and Dou Dejing. Differential privacy preservation for deep auto-encoders: An application of human behavior prediction. *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[41] Guillaume Perez, Michel Barlaud, Lionel Fillatre, and Jean-Charles Régin. A filtered bucket-clustering method for projection onto the simplex and the ℓ_1-ball. *Mathematical Programming*, May 2019.

[42] Milos Radovanovic, Alexandros Nanopoulos, and Mirjina Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.

[43] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1278–1286. PMLR, 22–24 Jun 2014.

[44] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[45] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *arXiv cs.LG /1606.03498*, 2016.

[46] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *arXiv stat.ML /1602.02282*, 2016.

[47] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[48] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv stat.ML /1703.00395*, 2017.

[49] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[51] Pascal Vincent and Hugo Larochelle. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11, 2010.

[52] Han Xiao, K Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv cs.LG/1708.07747*, 2017.

[53] Alexandre Yahi, Rami Vanguri, Noémie Elhadad, and Nicholas P. Tatonetti. Generative adversarial networks for electronic health records: A framework for exploring and evaluating methods for predicting drug-induced laboratory test trajectories. *arXiv, cs.LG, 1712.00164*, 2017.

[54] Jaehong Yoon and Sung Ju Hwang. Combined group and exclusive sparsity for deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3958–3966. JMLR. org, 2017.

[55] Hao Zhou, Jose M Alvarez, and Fatih Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision*, pages 662–677. Springer, 2016.

[56] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3597–3607. Curran Associates, Inc., 2019.

[57] Qiuyu Zhu and Ruixin Zhang. A classification supervised auto-encoder based on predefined evenly-distributed class centroids. *arXiv, cs.CV, 1902.00220v3*, january 2020.