Research Article Open Access

Samy Blusseau*, Bastien Ponchon, Santiago Velasco-Forero*, Jesús Angulo, and Isabelle Bloch

Approximating morphological operators with part-based representations learned by asymmetric auto-encoders

https://doi.org/10.1515/mathm-2020-0102
Received November 11, 2019; accepted July 17, 2020

Abstract: This paper addresses the issue of building a part-based representation of a dataset of images. More precisely, we look for a non-negative, sparse decomposition of the images on a reduced set of atoms, in order to unveil a morphological and explainable structure of the data. Additionally, we want this decomposition to be computed online for any new sample that is not part of the initial dataset. Therefore, our solution relies on a sparse, non-negative auto-encoder, where the encoder is deep (for accuracy) and the decoder shallow (for explainability). This method compares favorably to the state-of-the-art online methods on two benchmark datasets (MNIST and Fashion MNIST) and on a hyperspectral image, according to classical evaluation measures and to a new one we introduce, based on the equivariance of the representation to morphological operators.

Keywords: Non-negative sparse coding, Auto-encoders, Mathematical Morphology, Morphological invariance, Representation Learning, XAI.

MSC: 06-08, 15A23, 5A80, 68T07, 68T09, 68U10

1 Introduction

Mathematical morphology is strongly related to the problem of data representation. Applying a morphological filter can be seen as a test on how well the analyzed element is represented by the set of invariants of the filter. For example, applying an opening by a structuring element B tells how well a shape can be represented by the supremum of translations of B. The morphological skeleton [18, 24] is a typical example of description of shapes by a family of building blocks, classically homothetic spheres. It provides a disjunctive decomposition where components - for example, the spheres - can only contribute positively as they are combined by supremum. A natural question is the optimality of this additive decomposition according to a given criterion, for example its sparsity - the number of components needed to represent an object. Finding a sparse disjunctive (or part-based) representation has at least two important features: first, it allows *saving resources* such as memory and computation time in the processing of the represented object; secondly, it provides a *better*

Bastien Ponchon: Centre for Mathematical Morphology, Mines ParisTech, PSL Research University, France, LTCI, Télécom Paris, Institut Polytechnique de Paris, France, E-mail: bastien.ponchon@gmail.com
***Corresponding Author: Samy Blusseau:** Centre for Mathematical Morphology, Mines ParisTech, PSL Research University, France, E-mail: samy.blusseau@mines-paristech.fr
***Corresponding Author: Santiago Velasco-Forero:** Centre for Mathematical Morphology, Mines ParisTech, PSL Research University, France, E-mail: santiago.velasco@mines-paristech.fr
Jesús Angulo: Centre for Mathematical Morphology, Mines ParisTech, PSL Research University, France, E-mail: jesus.angulo@mines-paristech.fr
Isabelle Bloch: LTCI, Télécom Paris, Institut Polytechnique de Paris, France, E-mail: isabelle.bloch@telecom-paris.fr

understanding of this object, as it reveals its most elementary components, hence operating a dimensionality reduction that can alleviate the issue of model over-fitting. Such representations are also believed to be the ones at stake in human object recognition [25].

Similarly, the question of finding a sparse disjunctive representation of a whole database is also of great interest and will be the main focus of the present paper. More precisely, we will approximate such a representation by a non-negative, sparse linear combination of non-negative components, and we will call *additive* this representation. Given a large set of images, our concern is then to find a smaller set of non-negative image components, called dictionary, such that any image of the database can be expressed as an additive combination of the dictionary components. As we will review in the next section, this question lies at the crossroad of two broader topics known as sparse coding and dictionary learning [17].

Besides a better understanding of the data structure, our approach is also more specifically linked to mathematical morphology applications. Inspired by recent work [1, 28], we look for image representations that can be used to efficiently calculate approximations to morphological operators. The main goal is to be able to apply morphological operators to massive sets of images by applying them only to the reduced set of dictionary images. This is especially relevant in the analysis of remote sensing hyperspectral images where different kinds of morphological decomposition, such as morphological profiles [19] are widely used. For reasons that will be explained later, sparsity and non-negativity are sound requirements to achieve this goal. What is more, whereas the representation process can be learned offline on a training dataset, we need to compute the decomposition of any new sample *online*. Hence, we take advantage of the recent advances in deep, sparse and non-negative auto-encoders to design a new framework able to learn part-based representations of an image database, compatible with morphological processing. To that extent, this work is part of the resurgent research line investigating interactions between deep learning and mathematical morphology [9, 22, 23, 27, 32]. However with respect to these studies, focusing mainly on introducing morphological operators in neural networks, the present paper addresses a different question.

The existing work on non-negative sparse representations of images is reviewed in Section 2, that stands as a baseline and motivation of the present study. Then we present in Section 3 new results about part-based approximations of morphological operators. The proposed model for part-based representation learning is described in Section 4, a preliminary version of which can be found in [20]. Results on two image datasets (MNIST [13] and Fashion MNIST [29]) are discussed in Section 5, and we show how the proposed model compares to other deep part-based representations. An example on hyperspectral images is illutrated as well. We finally draw conclusions and suggest several tracks for future work. The code for reproducing our experiments is available online[1].

2 Related work

2.1 Non-negative sparse mathematical morphology

The present work finds its original motivation in [28], where the authors set the problem of learning a representation of a large image dataset to quickly compute approximations of morphological operators on the images. They find a good representation in the sparse variant of Non-negative Matrix Factorization (sparse NMF) [11], that we present hereafter.

Consider a family of M images (binary or gray-scale) $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(M)}$ of N pixels each, aggregated into a $M \times N$ data matrix $\mathbf{X} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(M)})^T$ (the i^{th} row of $\mathbf{X}$ is the transpose of $\mathbf{x}^{(i)}$ seen as a vector). Given a feature dimension $k \in \mathbb{N}^*$ and two numbers s_H and s_W in $[0, 1]$, a sparse NMF of $\mathbf{X}$ with dimension k, as

[1] For code release, visit https://gitlab.telecom-paristech.fr/images-public/asymae_morpho

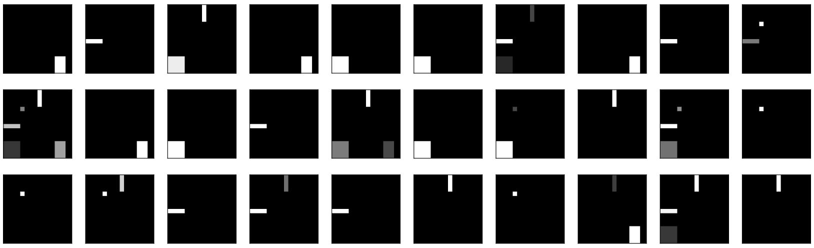


Figure 1: A subset of 30 images, extracted from a larger synthetic dataset of 1000 images, built as non-negative linear combinations of the five atom images of Figure 2(a). Although some images may look identical they are not, as the gray levels slightly differ.

defined in [11], is any solution $(\mathbf{H}, \mathbf{W})$ of the problem

$$\min ||\mathbf{X} - \mathbf{HW}||_2^2 \quad \text{s.t.} \quad \begin{cases} \mathbf{H} \in \mathbb{R}^{M\times k}, \mathbf{W} \in \mathbb{R}^{k\times N} \\ \mathbf{H} \geq 0,\ \mathbf{W} \geq 0 \\ \sigma(\mathbf{H}_{:,j}) = s_H,\ \sigma(\mathbf{W}_{j,:}) = s_W,\ 1 \leq j \leq k \end{cases} \tag{1}$$

where the second constraint means that both **H** and **W** have non-negative coefficients, and the third constraint imposes the degree of sparsity of the columns of **H** and lines of **W** respectively, with σ the function defined by

$$\forall \mathbf{v} \in \mathbb{R}^p, \quad \sigma(\mathbf{v}) = \frac{\sqrt{p} - ||\mathbf{v}||_1/||\mathbf{v}||_2}{\sqrt{p} - 1}. \tag{2}$$

Note that σ takes values in $[0, 1]$. The value $\sigma(\mathbf{v}) = 1$ characterizes vectors $\mathbf{v}$ having a unique non-zero coefficient, therefore the sparsest ones, and $\sigma(\mathbf{v}) = 0$ the vectors whose coefficients all have the same absolute value. Hoyer [11] designed an algorithm to find at least a local minimizer for the problem (1), and it was shown that under fairly general conditions (and provided the L_2 norms of **H** and **W** are fixed) the solution is unique [26].

In representation learning, each row $\mathbf{h}^{(i)}$ of **H** is called the *encoding* or *latent features* of the input image $\mathbf{x}^{(i)}$, and **W** holds in its rows a set of k images called the *dictionary*. In the following, we will refer to the images $\mathbf{w}_j = \mathbf{W}_{j,:}$ of the dictionary as *atom images* or *atoms*. As stated by Equation (1), the atoms are combined to approximate each image $\mathbf{x}^{(\mathbf{i})} := \mathbf{X}_{i,:}$ of the dataset by an estimate $\hat{\mathbf{x}}^{(i)}$, which writes as follows:

$$\forall i \in \{1, \ldots, M\},\ \hat{\mathbf{x}}^{(i)} = \mathbf{H}_{i,:}\mathbf{W} = \mathbf{h}^{(i)}\mathbf{W} = \sum_{j=1}^{k} h_{i,j}\mathbf{w}_j, \tag{3}$$

where $h_{i,j}$ is the coefficient at row i and column j in matrix **H** (see Figures 3 and 4 for illustration). The assumption behind this decomposition is that the more similar the images of the set, the smaller the required dimension to accurately approximate this set. Note that only $k(N + M)$ values need to be stored or handled when using the previous approximation to represent the data, against the NM values composing the original data.

For illustration purposes, we propose a toy example. We generated a dataset of 1000 images of size 32×32 pixels, as non negative linear combinations of the five atom images shown on Figure 2 (a). We call this dataset the *Rectangles dataset* and show 30 samples of it in Figure 1. Here the matrix **X** counts $M = 1000$ rows and $N = 32 \times 32 = 1024$ columns. We apply the sparse NMF algorithm to recover five atoms (stored in a non-negative matrix $\mathbf{W} \in \mathbb{R}_+^{5\times 1024}$) and 1000 encodings (stored in a non-negative matrix $\mathbf{H} \in \mathbb{R}_+^{1000\times 5}$) such that $\hat{\mathbf{X}} = \mathbf{HW}$ approximates well **X**. The five recovered atoms are shown in Figure 2 (b), and Figure 3 shows two examples of approximate non-negative reconstructions. Note that the excellent results here are due to the 1000 images of the Rectangles dataset being created precisely as sparse, non-negative combinations of only five, pairwise disjoint, atoms. As such, it is close to verify the hypothesis for which the NMF yields a unique and accurate part-based representation of data [6]. In the remaining of the paper we will no longer work with this dataset and focus on more realistic data.

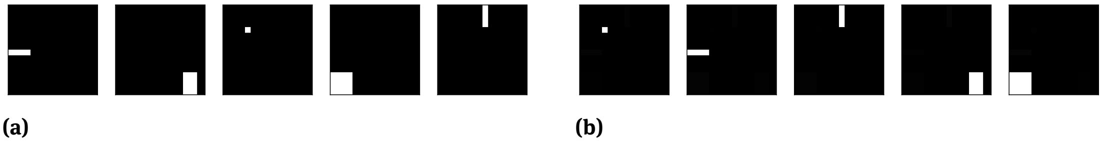


Figure 2: (a) The five atom images used to build a dataset of 1000 images such as those of Figure 1. (b) Computed atoms by the sparse NMF of the latter dataset. Up to a permutation in indexing, the computed atoms are very similar (but not strictly identical) to the original ones.

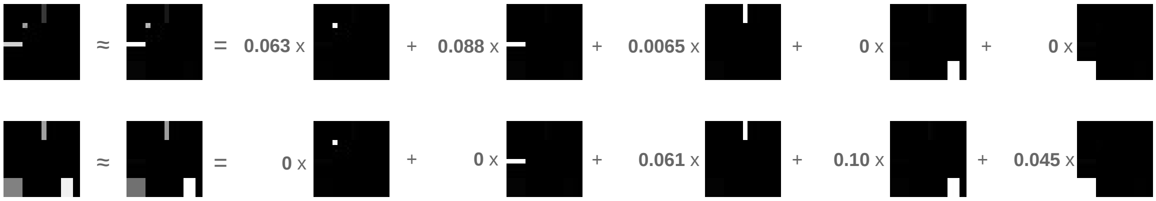


Figure 3: Approximation of two images of the Rectangles dataset by a sparse, non-negative matrix factorization. This illustrates Equation (3) for two different indices i and for $k = 5$. The leftmost images are $\mathbf{x}^{(\mathbf{i})}$s, the second column images are their approximations $\hat{\mathbf{x}}^{(i)}$s, the gray coefficients are the $h_{i,j}$ and the other images are the five computed atoms $\mathbf{w}_j$, $1 \leq j \leq 5$, also shown in Figure 2 (b).

By choosing the sparse NMF representation, the authors of [28] aim at approximating a morphological operator ϕ on the data $\mathbf{X}$ by applying it to the atom images $\mathbf{W}$ only, before projecting back into the input image space. That is, they want $\phi(\mathbf{x}^{(i)}) \approx \Phi(\mathbf{x}^{(i)})$, with $\Phi(\mathbf{x}^{(i)})$ defined by

$$\Phi(\mathbf{x}^{(i)}) := \sum_{j=1}^{k} h_{i,j}\phi(\mathbf{w}_j), \tag{4}$$

where the $h_{i,j}$ and $\mathbf{w}_j$ are the same as in Equation (3). The operator Φ in Equation (4) is called a **part-based approximation** to ϕ. To understand why non-negativity and sparsity help this approximation to be a good one, we can point out a few key arguments. First, sparsity favors the support of the weighted atom images to have little pairwise overlap. Secondly, a sum of images with disjoint supports is equal to their (pixel-wise) supremum. Finally, dilations commute with the supremum and, under certain conditions that are favored by sparsity, this also holds for the erosions. This will be developed in more details in Section 3. For now, Figure 4 illustrates the part-based approximation D_B of the dilation δ_B by a structuring element B, expressed as:

$$D_B(\mathbf{x}^{(i)}) := \sum_{j=1}^{k} h_{i,j}\delta_B(\mathbf{w}_j). \tag{5}$$

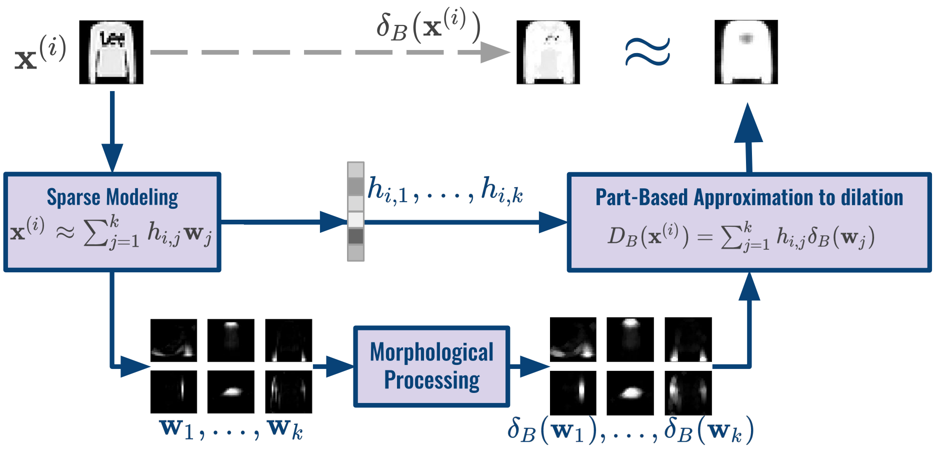

$\mathbf{x}^{(i)}$
$\delta_B(\mathbf{x}^{(i)})$
$\approx$
Sparse Modeling
$\mathbf{x}^{(i)} \approx \sum_{j=1}^{k} h_{i,j}\mathbf{w}_j$
$h_{i,1}, \ldots, h_{i,k}$
Part-Based Approximation to dilation
$D_B(\mathbf{x}^{(i)}) = \sum_{j=1}^{k} h_{i,j}\delta_B(\mathbf{w}_j)$
Morphological Processing
$\mathbf{w}_1, \ldots, \mathbf{w}_k$
$\delta_B(\mathbf{w}_1), \ldots, \delta_B(\mathbf{w}_k)$

Figure 4: Process for computing the part-based approximation to dilation, based on Equations (3) and (5).

2.2 Deep auto-encoders approaches

The main drawback of the NMF algorithm is that it is an *offline* process, and the encoding of any new sample with regards to the previously learned basis $\mathbf{W}$ requires either to solve a computationally extensive constrained optimization problem, or to relax the non-negativity constraint by using the pseudo-inverse $\mathbf{W}^+$ of the basis. Some approaches proposed to overcome this shortcoming rely on Deep Learning, and especially on deep auto-encoders, which are widely used in the representation learning field, and offer an *online* representation process [8, 10, 15].

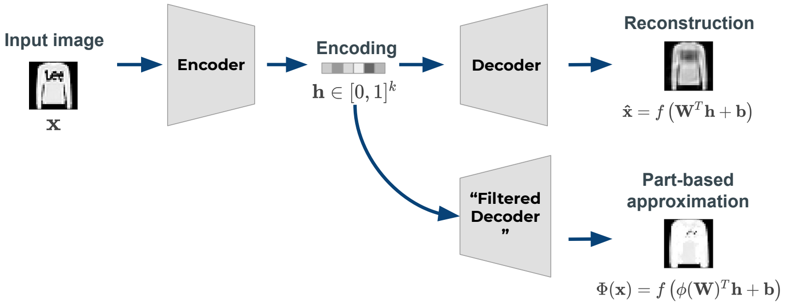


Figure 5: The auto-encoding process and the definition of part-based approximation to a morphological operator ϕ in this framework.

An auto-encoder, as represented in Figure 5, is a model composed of two stacked neural networks, an encoder and a decoder whose parameters are trained by minimizing a loss function. A common example of loss function is the mean square error (MSE) between the input images $\mathbf{x}^{(i)}$ and their reconstructions by the decoder $\hat{\mathbf{x}}^{(i)}$:

$$L_{AE} = \frac{1}{M}\sum_{i=1}^{M} L(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)}) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{N}||\hat{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}||_2^2. \tag{6}$$

In this framework, and when the decoder is composed of a single linear layer (possibly followed by a non-linear activation function), the model approximates the input images as:

$$\hat{\mathbf{x}}^{(i)} \quad = f\left(\mathbf{b} + \mathbf{h}^{(i)}\mathbf{W}\right) \quad = f\left(\mathbf{b} + \sum_{j=1}^{k} h_{i,j}\mathbf{w}_j\right) \tag{7}$$

where $\mathbf{h}^{(i)}$ is the encoding of the input image by the encoder network, $\mathbf{b}$ and $\mathbf{W}$ respectively the bias and weights of the linear layer of the decoder, and f the (possibly non-linear) activation function, that is applied pixel-wise to the output of the linear layer. The output $\hat{\mathbf{x}}^{(i)}$ is called the *reconstruction* of the input image $\mathbf{x}^{(i)}$ by the auto-encoder. It can be considered as a linear combination of atom images, up to the addition of an offset image $\mathbf{b}$ and to the application of the activation function f. The images of our learned dictionary are hence the columns of the weight matrix $\mathbf{W}$ of the decoder. We can extend the definition of part-based approximation, described in Section 2.1, to our deep learning architectures, by applying the morphological operator to these atoms $\mathbf{w}_1, ..., \mathbf{w}_k$, as pictured by Figure 5. Note that a central question lies in how to set the size k of the latent space. This question is beyond the scope of this study and the value of k will be arbitrarily fixed (we take $k = 100$) in the following.

The NNSAE architecture (for Non-Negative Sparse Autoencoder), from Lemme *et al.* [15], proposes a very simple and shallow architecture for online part-based representations using linear encoder and decoder with tied weights (the weight matrix of the decoder is the transpose of the weight matrix of the encoder). Both the NCAE architectures (Nonnegativity-Constrained Autoencoder), from Hosseini-Asl *et al.* [10] and the work from Ayinde *et al.* [2], that aims at extending it, drop this transpose relationship between the weights of the encoder and of the decoder, increasing the capacity of the model. Those three networks enforce the non-negativity of the elements of the representation, as well as the sparsity of the image encodings using various techniques.

2.2.1 Enforcing sparsity of the encoding

The most prevalent idea to enforce sparsity of the encoding in a neural network can be traced back to the work of H. Lee *et al.* [14]. This variant penalizes, through the loss function, a deviation S of the expected activation of each hidden unit (*i.e.* the output units of the encoder) from a low fixed level p. Intuitively, this should ensure that each of the units of the encoding is activated only for a limited number of images. The resulting loss function of the sparse auto-encoder is then:

$$L_{AE} = \frac{1}{M}\sum_{i=1}^{M} L(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)}) + \beta \sum_{j=1}^{k} S(p, \sum_{i=1}^{M} h_j^{(i)}), \tag{8}$$

where the parameter p sets the expected activation objective of each of the hidden neurons, and the parameter β controls the strength of the regularization. The function S can be of various forms, which were empirically surveyed in [31]. The approach adopted by the NCAE [10] and its extension [2] both rely on a penalty function based on the KL-divergence between two Bernoulli distributions, whose parameters are the expected activation and p respectively, as used in [10]:

$$S(p, t_j) = KL(p, t_j) = p \log\frac{p}{t_j} + (1-p)\log\frac{1-p}{1-t_j} \quad \text{with } t_j = \sum_{i=1}^{M} h_j^{(i)} \tag{9}$$

The NNSAE architecture [15] introduces a slightly different way of enforcing the sparsity of the encoding, based on a parametric logistic activation function at the output of the encoder, whose parameters are trained along with the other parameters of the network.

2.2.2 Enforcing non-negativity of the decoder weights

For the NMF (Section 2.1) and for the decoder, non-negativity results in a part-based representation of the input images. In the case of neural networks, enforcing the non-negativity of the weights of a layer eliminates cancellations of input signals. In all the aforementioned works, the encoding is non-negative since the activation function at the output of the encoder is a sigmoid. In the literature, various approaches have been designed to enforce weight positivity. A popular approach is to use an asymmetric weight decay, added to the loss function of the network, to enact more decay on the negative weights that on the positive ones. However this approach, used in both the NNSAE [15] and NCAE [10] architectures, does not ensure that all weights will be non-negative. This issue motivated the variant of the NCAE architecture [2, 15], which uses either the L_1 rather than the L_2 norm, or a smooth version of the decay using both the L_1 and the L_2 norms. The source code of this method being unavailable at the time the present work was done, we did not use this more recent version as a baseline for our study.

Another type of approaches consists in initializing the decoder weights with non-negative values and ensure they remain so after each update during the optimization process. The simplest strategy, as implemented in the projected gradient descent [5], is to project the weights onto the positive orthant by setting negative components to zero. More recently, the exponentiated gradient descent was proposed as an alternative to the projected gradient descent [8]. The idea is to update the weights by multiplying them by a positive coefficient, which is an exponentially decreasing function of the partial derivative of the loss with respect to the weights. Although promising, the latter proposition does not include any sparsity constraint and the authors provide no quantitative measure on image reconstruction errors.

As far as non-negativity of weights is concerned, we may also mention [30], which uses an optimization process inspired by the NMF to satisfy the non-negative probability constraints of Random Neural Networks stacked in auto-encoders.

We will present in Section 4 our own auto-encoder solution for an online, non-negative and sparse representation of data, compatible with the approximation of morphological operators. In the next section we provide some mathematical insights on how non-negativity and sparsity are connected to such an approximation.

3 Equivariance of morphological operators to non-negative linear combinations

In this section we precise the intuitions sketched in Section 2.1 about the part-based approximation of morphological operators. Let $\mathcal{L}$ be the complete lattice of images with N pixels and with values in $[0, +\infty]$ ordered by the Pareto ordering ($\mathbf{x} \le \mathbf{y}$ iff for any q, $1 \le q \le N$, $\mathbf{x}_q \le \mathbf{y}_q$). Consider a flat, extensive dilation δ_B on $\mathcal{L}$ and its adjoint anti-extensive erosion ε_B, B being a flat structuring element. Let $\mathbf{x} \in \mathcal{L}$ be an image approximated by the non-negative combination $\hat{\mathbf{x}} = \sum_{j=1}^{k} h_j \mathbf{w}_j$ of k atom images $\mathbf{w}_1, \dots, \mathbf{w}_k \in \mathcal{L}$. Following Equation (4), we define the part based approximations of the four operators δ_B, ε_B, $\gamma_B = \delta_B \varepsilon_B$ and $\varphi_B = \varepsilon_B \delta_B$ as:

$$\begin{array}{ll} D_B(\mathbf{x}) := \sum_{j=1}^{k} h_j \delta_B(\mathbf{w}_j), & E_B(\mathbf{x}) := \sum_{j=1}^{k} h_j \varepsilon_B(\mathbf{w}_j) \\ G_B(\mathbf{x}) := \sum_{j=1}^{k} h_j \gamma_B(\mathbf{w}_j), & F_B(\mathbf{x}) := \sum_{j=1}^{k} h_j \varphi_B(\mathbf{w}_j). \end{array} \tag{10}$$

We focus on establishing whether these expressions approximate well their exact counterparts $\delta_B(\mathbf{x})$, $\varepsilon_B(\mathbf{x})$, $\gamma_B(\mathbf{x})$ and $\varphi_B(\mathbf{x})$, assuming $\mathbf{x}$ is well approximated by $\hat{\mathbf{x}} = \sum_{j=1}^{k} h_j \mathbf{w}_j = \mathbf{W}\mathbf{h}$. It is likely to be so as soon as $D_B(\mathbf{x}) = \delta_B(\hat{\mathbf{x}})$, $E_B(\mathbf{x}) = \varepsilon_B(\hat{\mathbf{x}})$, $G_B(\mathbf{x}) = \gamma_B(\hat{\mathbf{x}})$ and $F_B(\mathbf{x}) = \varphi_B(\hat{\mathbf{x}})$, which is to say as soon as the four operators commute with the non-negative linear application $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k] \mapsto \mathbf{W}\mathbf{h} = \sum_{j=1}^{k} h_j \mathbf{w}_j$. As sketched earlier, sums can be identified to suprema if the involved images have disjoint supports, and this also favors the commutation of the erosion with the supremum. This is why we introduce the following hypothesis that characterizes the disjunction of supports (i.e. the regions where the image is non-zero) of the $h_j \mathbf{w}_j$.

Let H_1 denote the hypothesis:

$$H_1\text{: "For any } 1 \le i \le k,\ 1 \le j \le k,\ i \ne j,\ \delta_B(h_i\mathbf{w}_i) \bigwedge \delta_B(h_j\mathbf{w}_j) = 0\text{"}$$

where 0 denotes an image equal to zero everywhere (i.e. with empty support), and more generally, for an integer n,

$$H_n\text{: "For any } 1 \le i \le k,\ 1 \le j \le k,\ i \ne j,\ \delta_B^n(h_i\mathbf{w}_i) \bigwedge \delta_B^n(h_j\mathbf{w}_j) = 0\text{",}$$

where $\delta_B^n = \delta_B \circ \cdots \circ \delta_B = \delta_{nB}$, denoting by nB the n-terms Minkowski sum $B \oplus B \oplus \cdots \oplus B$ for $n > 0$, and δ_B^n is the identity for $n = 0$. Note that, since δ_B is extensive, H_n implies any H_p with $p \le n$. In particular, any H_n implies H_0, which simply states the disjunction of the supports of any two images $h_i\mathbf{w}_i$ and $h_j\mathbf{w}_j$, $i \ne j$. We can now state the following result:

Proposition 1. *If H_1 holds for the representation $\hat{\mathbf{x}} = \sum_{j=1}^k h_j\mathbf{w}_j$, then:*

$$D_B(\mathbf{x}) = \delta_B(\hat{\mathbf{x}}), \quad E_B(\mathbf{x}) = \varepsilon_B(\hat{\mathbf{x}}), \quad G_B(\mathbf{x}) = \delta_B\big(\varepsilon_B(\hat{\mathbf{x}})\big) = \gamma_B(\hat{\mathbf{x}}).$$

If additionally H_2 holds, then we also have:

$$F_B(\mathbf{x}) = \varepsilon_B\big(\delta_B(\hat{\mathbf{x}})\big) = \varphi_B(\hat{\mathbf{x}}).$$

A proof of this result is detailed in Appendix A. Proposition 1 implies that under the H_n hypothesis the error $||\phi_B(\mathbf{x}) - \Phi_B(\mathbf{x})||^2$ between the actual transformed image and its part-based approximation only depends on the quality of the reconstruction, that is to say on the error $||\mathbf{x} - \hat{\mathbf{x}}||^2$. Indeed, if $\mathbf{x} = \hat{\mathbf{x}}$ then $D_B(\mathbf{x}) = \delta_B(\mathbf{x})$, $E_B(\mathbf{x}) = \varepsilon_B(\mathbf{x})$ and so on. Obviously, the more constrained the representation, the smaller the class of images that can be accurately represented. The non-negativity and sparsity constraints are therefore likely to increase the representation error $||\mathbf{x} - \hat{\mathbf{x}}||^2$. Hence, unless the data can be perfectly represented by non-negative combinations of atoms complying with a hypothesis H_n, a trade-off needs to be found to achieve a good approximation of morphological operators. This is the target of our asymmetric auto-encoder presented in Section 4.

We shall now generalize Proposition 1 by applying it to the representation that we note $\hat{\mathbf{x}}^{(\mathbf{n-1})} = \sum_{j=1}^k h_j\delta_{(n-1)B}(\mathbf{w}_j)$. Notice that H_1 holds for $\hat{\mathbf{x}}^{(\mathbf{n-1})}$ if and only if H_n holds for $\hat{\mathbf{x}}$. This yields the following corollary.

Corollary 1. *If H_n holds for the representation $\hat{\mathbf{x}} = \sum_{j=1}^k h_j\mathbf{w}_j$, then for any integer $p \le n$:*

$$D_{pB}(\mathbf{x}) = \delta_{pB}(\hat{\mathbf{x}}), \quad E_{pB}(\mathbf{x}) = \varepsilon_{pB}(\hat{\mathbf{x}}), \quad G_{pB}(\mathbf{x}) = \delta_{pB}\big(\varepsilon_{pB}(\hat{\mathbf{x}})\big) = \gamma_{pB}(\hat{\mathbf{x}}),$$

and for any integer $p \le n - 1$

$$F_{pB}(\mathbf{x}) = \varepsilon_{pB}\big(\delta_{pB}(\hat{\mathbf{x}})\big) = \varphi_{pB}(\hat{\mathbf{x}}).$$

Remarks

Choice of the complete lattice $\mathcal{L}$. At the beginning of this section we chose $\mathcal{L}$ as the complete lattice of images with N pixels and with values in $[0, +\infty]$ ordered by the Pareto ordering. However, in practice we deal more commonly with images whose values are in a bounded interval such as $[0, 1]$. The previous results still hold in the latter case, provided we add the hypothesis $h_j \in [0, 1]$. More generally, we only need to make sure that $\mathbf{w} \in \mathcal{L} \Rightarrow h\mathbf{w} \in \mathcal{L}$.

Interpretation of H_n. The hypothesis H_n, $n \ge 0$, characterizes the degree of disjunction of the supports of the $h_j\mathbf{w}_j$ involved in the part-based approximation of an image $\mathbf{x}$. The dilation δ_B being extensive, the degree of disjunction, intended as distance between supports of the initial images, "increases" with n. Note that no assumption is made on the disjunction of the whole set of atom images $\mathbf{w}_j$, but only on those atoms that are used in the approximation of $\mathbf{x}$, in other words the $\mathbf{w}_j$ weighted by a positive h_j. This helps realize that the number of atoms used to approximate an image matters. In the limit case where only one atom is used,

H_n is verified for any n. By contrast, if as many as N atoms contribute to the approximation, then even H_1 becomes impossible. In the context of the representation of a large dataset, the ideal case seems to be when every image is well approximated by few atoms, as disjoint as possible. This indicates that the H_n are not unrealistic hypotheses in practice, provided a sparse part-based representation approximates well the data, and nB is small enough compared to the supports of the atoms.

How necessary is H_n? The proof of Proposition 1 mainly stands on points 3 and 5 (see Appendix A). Therefore, we may ask whether the hypothesis $\delta_B(\mathbf{x}) \bigwedge \delta_B(\mathbf{y}) = 0$ is necessary to have $\delta_B(\mathbf{x}+\mathbf{y}) = \delta_B(\mathbf{x}) + \delta_B(\mathbf{y})$ and $\varepsilon_B(\mathbf{x}+\mathbf{y}) = \varepsilon_B(\mathbf{x}) + \varepsilon_B(\mathbf{y})$, which comes down to questioning the necessity of H_1 and H_2 in Proposition 1, or H_n in the corollary. The answer is they are *not* necessary in general. For example, for any increasing function $g : \mathbb{R}^+ \to \mathbb{R}^+$ and $\mathbf{y} = [g(\mathbf{x}_1), \dots, g(\mathbf{x}_N)]$ such that $\mathbf{x}+\mathbf{y} \in \mathcal{L}$, we do have $\delta_B(\mathbf{x}+\mathbf{y}) = \delta_B(\mathbf{x}) + \delta_B(\mathbf{y})$ and $\varepsilon_B(\mathbf{x}+\mathbf{y}) = \varepsilon_B(\mathbf{x}) + \varepsilon_B(\mathbf{y})$. However, if we consider rather "independent" components, it is easy to build fairly general configurations where a certain degree of disjunction is necessary. In particular, as shown in the examples of Figures 6 and 7, a simple disjunction (corresponding to H_0) is not sufficient in general.

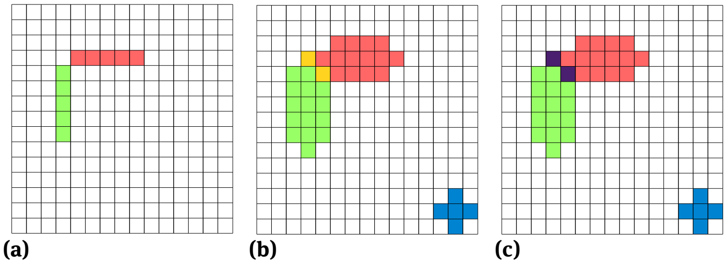


Figure 6: An example of non-equivariance of the dilation to non-negative linear combination. (a) The components h_1W_1 and h_2W_2 are piece-wise constant, equal to $h_1 > 0$ (in green) and $h_2 > 0$ (in red) respectively, where they are non-zero. (b) Dilation of the sum $\delta_B(h_1W_1 + h_2W_2)$, where B is the cross structuring element shown in blue. The color yellow represents the value $h_1 \vee h_2$. (c) Sum of the dilations $\delta_B(h_1W_1) + \delta_B(h_2W_2)$. The color purple represents the value $h_1 + h_2$ which is larger than $h_1 \vee h_2$. Thus although the two components do not overlap (H_0 holds), (b) and (c) are not equal.

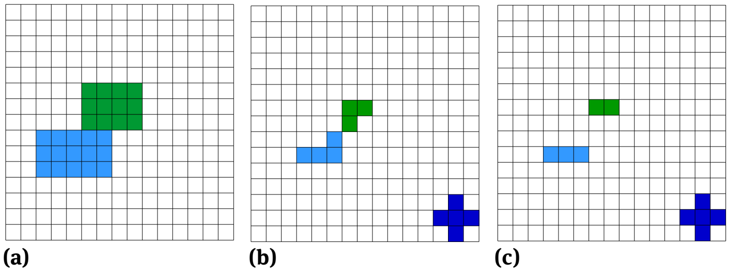


Figure 7: An example of non-equivariance of the erosion to the sum. (a) Two non-overlapping components $\mathbf{x}$ and $\mathbf{y}$.(b) Erosion of the sum $\varepsilon_B(\mathbf{x}+\mathbf{y})$ where B is the cross structuring element shown in dark blue. (c) Sum of the erosions $\varepsilon_B(\mathbf{x})+\varepsilon_B(\mathbf{y})$ by the same structuring element. Again, although the two components do not overlap (H_0 holds), (b) and (c) are not equal.

This section was meant to precise mathematically the role played by sparsity and non-negativity in the part-based approximation of morphological operators. Motivated by previous approaches described in Section 2.2, we present in the next section our proposed auto-encoder, designed to achieve the desired trade-off between explainability, accuracy of the data reconstruction and accuracy of the approximation of morphological operators.

4 Proposed model

We propose an online part-based representation learning model, using an asymmetric auto-encoder with sparsity and non-negativity constraints. As pictured in Figure 8, our architecture is composed of two networks: a deep encoder and a shallow decoder (hence the asymmetry and the name of AsymAE we chose for our architecture). The encoder network is based on the discriminator of the infoGAN architecture introduced in [4], which was chosen for its average depth, its use of widely adopted deep learning components such as batch-normalization [12], 2D-convolutional layers [7] and leaky-RELU activation function [16]. It has been designed specifically to perform interpretable representation learning on datasets such as MNIST and Fashion-MNIST. The network can be adapted to fit larger images. The decoder network is similar to the one presented in Figure 5. A Leaky-ReLU activation has been chosen after the linear layer. Its behavior is the same as the identity for positive entries, while it multiplies the negative ones by a fixed coefficient $\alpha_{lReLU} = 0.1$. This activation function has shown the best performances in similar architectures [16]. The sparsity of the encoding is achieved using the same approach as in [2, 10], that consists in adding to the previous loss function the regularization term described in Equations (8) and (9).

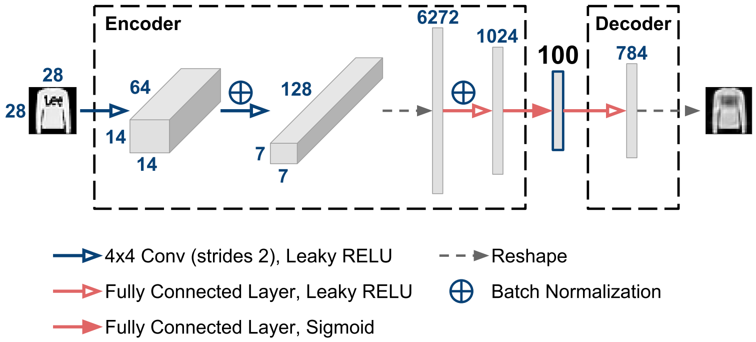

Encoder
6272
1024
100
Decoder
784
28
28
64
14
14
128
7
7
4x4 Conv (strides 2), Leaky RELU
Reshape
Fully Connected Layer, Leaky RELU
Batch Normalization
Fully Connected Layer, Sigmoid

Figure 8: Our proposed auto-encoder architecture.

We only enforced the non-negativity of the weights of the decoder, as they define the dictionary of images of our learned representation and as enforcing the non-negativity of the encoder weights would bring nothing but more constraints to the network and lower its capacity. Similarly to [5], we enforced this non-negativity constraint explicitly by projecting our weights on the nearest points of the positive orthant after each update of the optimization algorithm (such as the stochastic gradient descent). The main asset of this other method that does not use any additional penalty functions, and which is quite similar to the way the NMF enforces non-negativity, is that it ensures positivity of all weights without the cumbersome search for good values of the parameters of the various regularization terms in the loss function.

5 Experiments

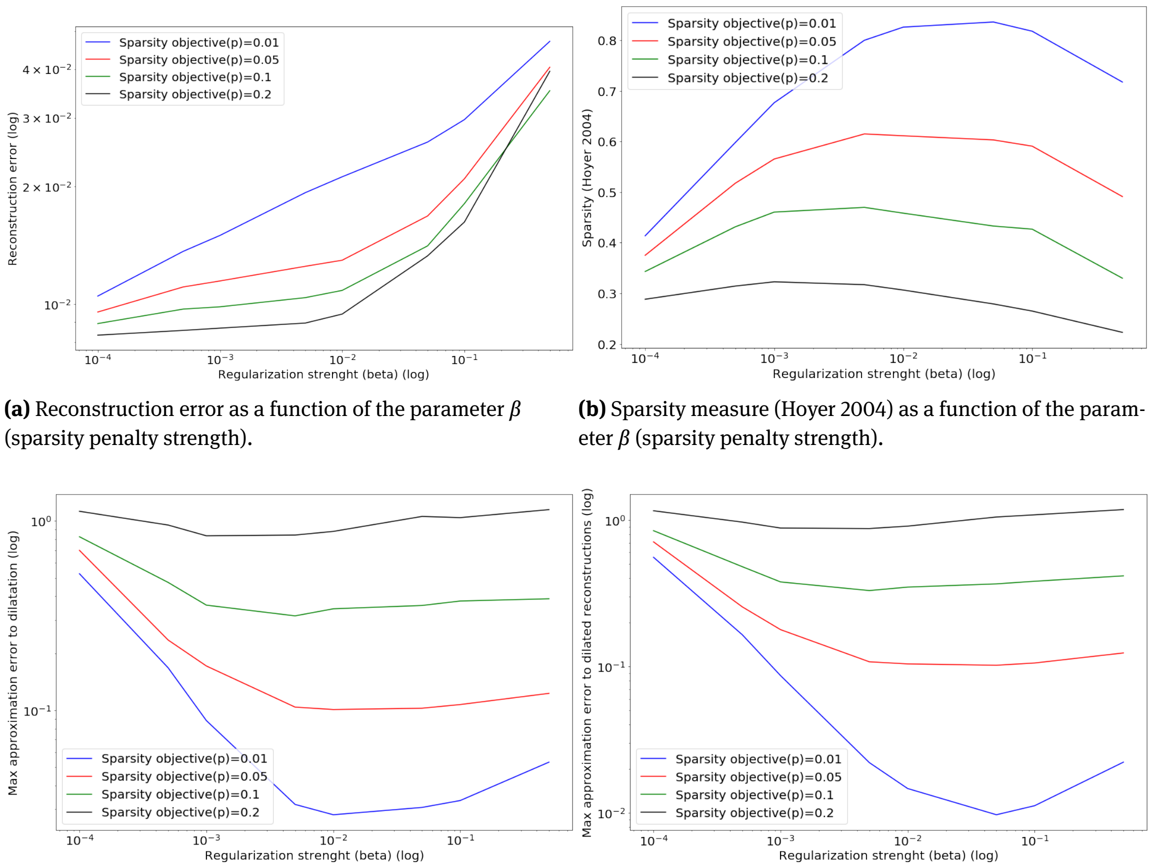
5.1 Experiment 1 on MNIST and Fashion MNIST

To demonstrate the goodness and drawbacks of our method, we have conducted experiments on two well-known datasets MNIST [13] and Fashion MNIST [29]. These two datasets share common features, such as the size of the images (28×28), the number of classes represented (10), and the total number of images (70000), divided into a training set of 60000 images and a test set of 10000 images.

5.1.1 Setting the parameters

For our AsymAE algorithm, we studied the effect of the sparsity objective p and regularization weight β in the loss function in Equation (8). In Figure 9 we present the results of the proposed approach on the Fashion-MNIST dataset. The maximum of the sparsity measure was reached with the sparsity parameters $p = 0.01$ and $\beta = 0.01$, whose atoms are shown in Figure 10e. It appears that these atoms are closer to full clothes shapes than parts. A possible interpretation is that, as the sparsity constraint gets stronger, the model is pushed to the limit where an atom should be involved in the reconstruction of a proportion p of the training images, that is approximately $p \cdot M$ images. When the number k of atoms is much smaller than $p \cdot M$ (which is the case for $k = 100$, $p = 0.01$ and $M = 60000$), each atom needs to be shared by a whole subset of images as their unique (or almost) representative. The model is therefore performing some sort of k-means clustering, each atom being a barycenter of a subgroup of the training set.

In Figure 10 we show examples of atom images for other values of sparsity parameters. The representations shown in Figures 10b, 10c and 10d are quite close to a part-based representation, even though the supports of the atom images are less disjoint as they would be in an ideal part-based representation, such as the sparse NMF, whose atom images are very neat. From this visual inspection as well as the plots of Figure 9, we found that a better trade-off seems to be reached for the values $p = 0.05$ and $\beta = 0.0005$ in the case of the Fashion-MNIST dataset. A similar study led to choose $p = 0.05$ and $\beta = 0.001$ with the MNIST dataset.



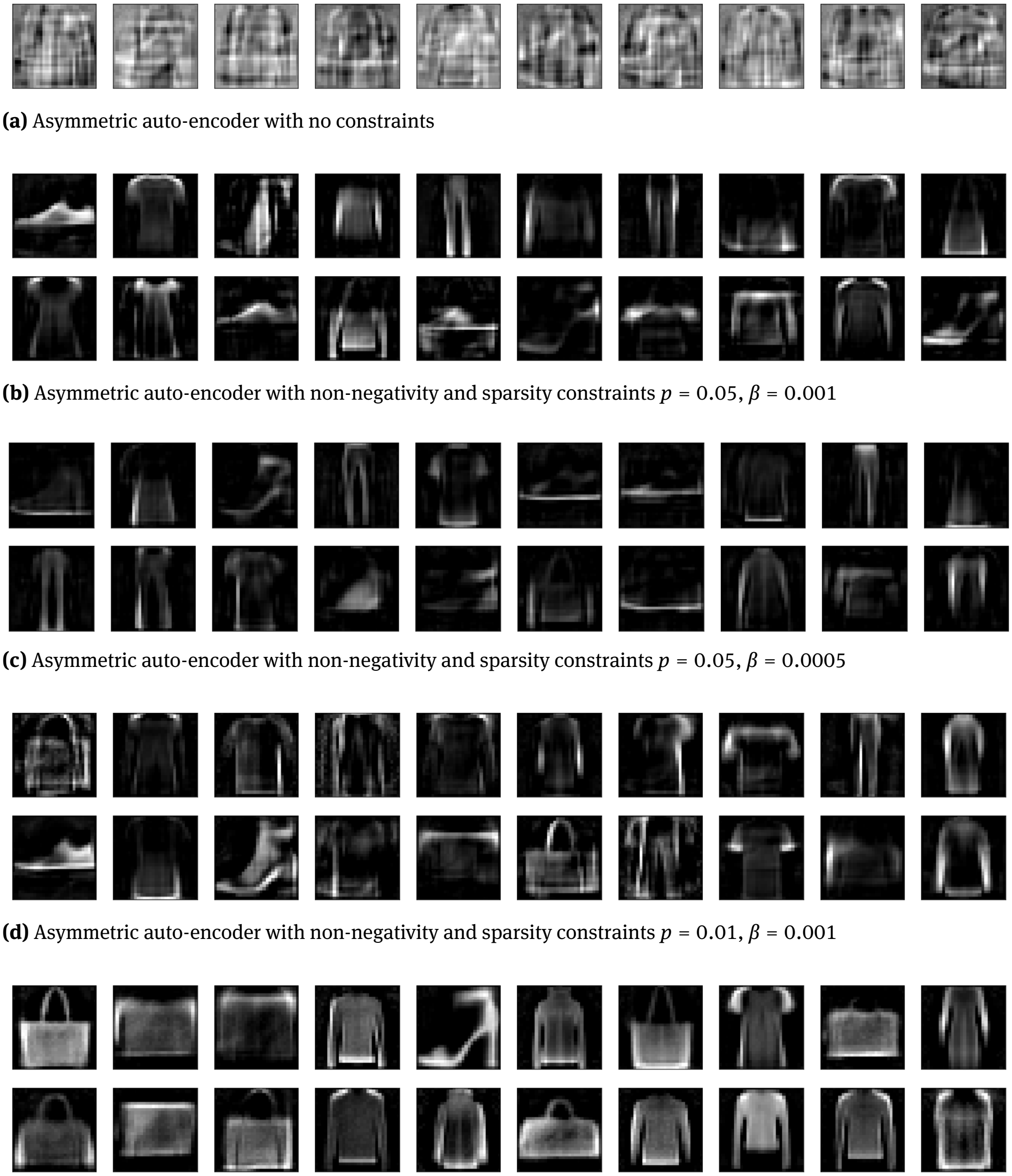
(a) Reconstruction error as a function of the parameter β (sparsity penalty strength).

(b) Sparsity measure (Hoyer 2004) as a function of the parameter β (sparsity penalty strength).

(c) Max-approximation error to dilation (of the original images) as a function of the parameter β (sparsity penalty strength).

(d) Max-approximation error to dilation (of the reconstructed images) as a function of the parameter β (sparsity penalty strength).

Figure 9: Some evaluation measures for sparse non-negative asymmetric auto-encoders for various parameters of the sparsity regularization, using a test set not used to train the network.



(a) Asymmetric auto-encoder with no constraints

(b) Asymmetric auto-encoder with non-negativity and sparsity constraints $p = 0.05, \beta = 0.001$

(c) Asymmetric auto-encoder with non-negativity and sparsity constraints $p = 0.05, \beta = 0.0005$

(d) Asymmetric auto-encoder with non-negativity and sparsity constraints $p = 0.01, \beta = 0.001$

(e) Asymmetric auto-encoder with non-negativity and sparsity constraints $p = 0.01, \beta = 0.01$

Figure 10: Some atoms (out of the 100 atoms) of various versions of the proposed asymmetric auto-encoder.

5.1.2 Comparison to state of the art methods

We compared our method to three baselines: the sparse-NMF [11], the NNSAE [15], and the NCAE [10]. The three deep-learning models (the proposed AsymAE, NNSAE and NCAE) were trained until convergence on the training set, and evaluated on the test set. The sparse-NMF algorithm was ran and evaluated on the test set. Note that all models but the NCAE may produce reconstructions that do not fully belong to the interval $[0, 1]$. In order to compare the reconstructions and the part-based approximations produced by the various algorithms, their outputs will be clipped between 0 and 1. There is no need to apply this operation to the output of NCAE as a sigmoid activation enforces the output of its decoder to belong to [0, 1]. We used three measures to conduct this comparison:

- the reconstruction error, that is the pixel-wise mean squared error between the input images $\mathbf{x}^{(i)}$ of the test dataset and their reconstruction/approximation $\hat{\mathbf{x}}^{(i)}$: $\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(\mathbf{x}_j^{(i)} - \hat{\mathbf{x}}_j^{(i)})^2$;
- the sparsity of the encoding, measured using the mean on all test images of the sparsity measure σ in Equation 2: $\frac{1}{M}\sum_{i=1}^{M}\sigma(\mathbf{h}^{(i)})$;
- the approximation error to dilation by a disk of radius one, obtained by computing the pixel-wise mean squared error between the dilation δ_B by a disk of radius one of the original image and the part-based approximation D_B to the same dilation, using the learned representation: $\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(D_B(\mathbf{x}^{(i)})_j - \delta_B(\mathbf{x}^{(i)})_j)^2$.

The parameter settings used for NCAE and the NNSAE algorithms are the ones provided in [10, 15]. For the sparse-NMF, a sparsity constraint of $S_h = 0.6$ was applied to the encodings and no sparsity constraint was applied on the atoms of the representation.

Table 1: Comparison of the reconstruction error, sparsity of encoding and part-based approximation error to dilation produced by the sparse-NMF, the NNSAE, the NCAE and the AsymAE, for both MNIST and Fashion-MNIST datasets.

Model	Reconstruction error	Sparsity of code	Part-based approximation error to dilation
		MNIST	
Sparse-NMF	0.011	**0.66**	**0.012**
NNSAE	0.015	0.31	0.028
NCAE	0.010	0.35	0.18
AsymAE	**0.007**	0.54	0.069
		Fashion MNIST	
Sparse-NMF	0.011	**0.65**	**0.022**
NNSAE	0.029	0.22	0.058
NCAE	0.017	0.60	0.030
AsymAE	**0.010**	0.52	0.066

Both the quantitative results (Table 1) and the reconstruction images (Figure 11) demonstrate the capacity of our model to reach a better trade-off between the accuracy of the reconstruction and the sparsity of the encoding (that usually comes at the expense of the former criteria), than the other neural architectures. Indeed, in all conducted experiments, varying the parameters of the NCAE and the NNSAE as an attempt to increase the sparsity of the encoding came with a dramatic increase of the reconstruction error of the model. We failed however to reach a trade-off as good as the sparse-NMF algorithm that manages to match a high sparsity of the encoding with a low reconstruction error, especially on the Fashion-MNIST dataset. The major difference between the algorithms can be seen in Figure 12 that pictures 16 of the 100 atoms of each of the four learned representations. While sparse-NMF manages, for both datasets, to build highly explainable and clean part-based representations, the two deep baselines build representations that picture either too local shapes, in the case of the NNSAE, or too global ones, in the case of the NCAE. Our method suffers from quite the same issues as the NCAE, as almost full shapes are recognizable in the atoms. We noticed through experiments that increasing the sparsity of the encoding leads to less and less local features in the atoms. It has to be noted that the L_2 Asymmetric Weight Decay regularization used by the NCAE and NNSAE models allows for a certain proportion of negative weights. As an example, up to 32.2% of the pixels of the atoms of the NCAE model trained on the Fashion-MNIST dataset are negative, although their amplitude is lower than the average amplitude of the positive weights. The amount of negative weights can be reduced by increasing the corresponding regularization, which comes at the price of an increased reconstruction error and less sparse encodings. Finally Figure 13 pictures the part-based approximation to dilation by a structuring element of size one, computed using the four different approaches on ten images from the test set. Although the quantitative results state otherwise, we can note that our approach yields an interesting part-based approximation, thanks to a good balance between a low overlapping of atoms (and dilated atoms) and a good reconstruction capability.

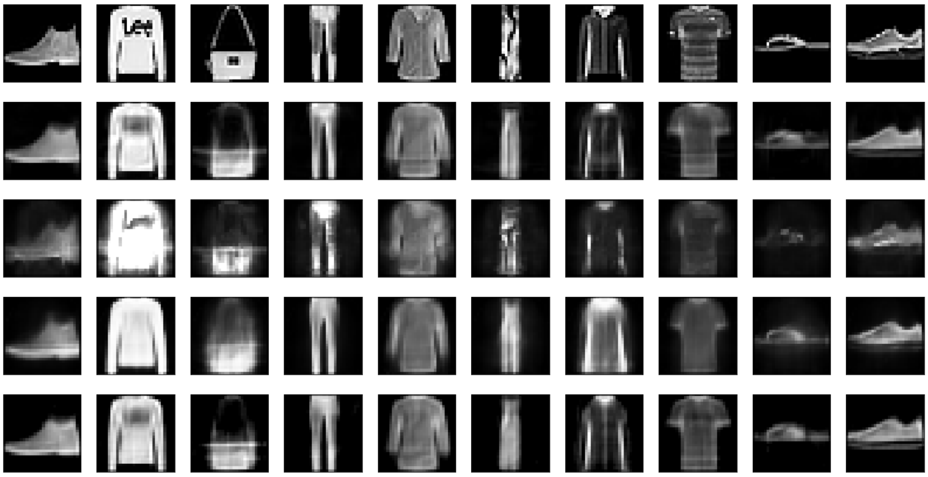
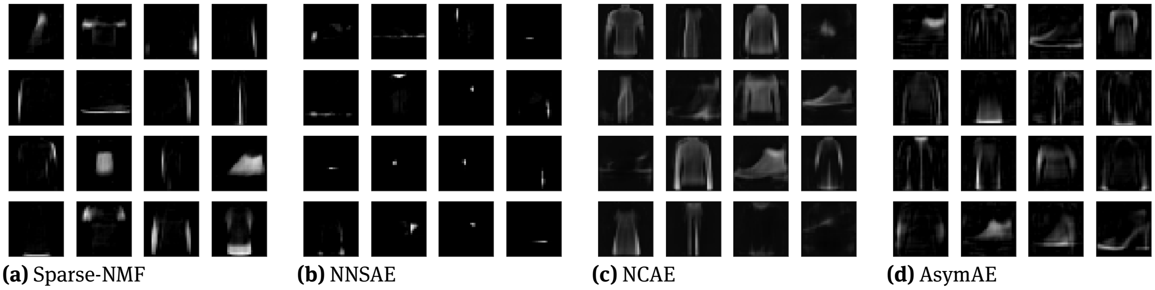


Figure 11: Reconstruction of the Fashion-MNIST dataset (first row) by the sparse-NMF, the NNSAE, the NCAE and the AsymAE.



(a) Sparse-NMF **(b)** NNSAE **(c)** NCAE **(d)** AsymAE

Figure 12: 16 of the 100 atom images of the four compared representations of Fashion-MNIST dataset.

5.2 Experiment 2: the Pavia University hyperspectral image

In order to test our approach on more realistic and complex data, we carried an experiment on the Pavia University hyperspectral image[2], of spatial size 610 × 340 pixels and containing $M = 103$ spectral bands (Figure 14). For memory issues and in order to take advantage of the previous experiment, we divided each channel image into $9 \times 5 = 45$ non-overlapping 64×64 patches, covering 576×320 pixels starting from the top left hand corner. The database thus counted $45 \times 103 = 4635$ patches, that we split into a training set and a test set by dedicating a fixed proportion $\rho \in [0, 1]$ of the spectral bands to the training. This means the patches of a given spectral band were all assigned to the training set or all to the test set. What is more, the spectral bands assigned to the test set were sampled regularly (not randomly).

[2] The Pavia University hyperspectral image is available here: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Pavia_University_scene

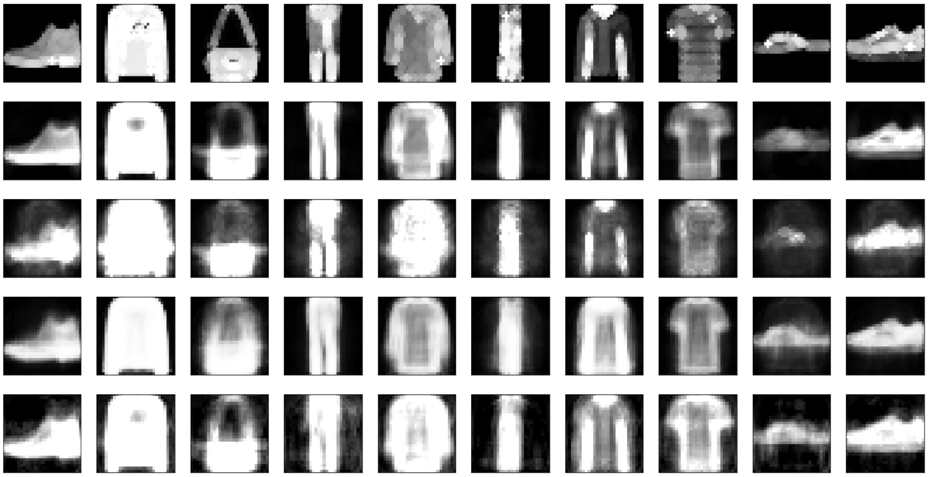


Figure 13: Part-based approximation of the dilation by a structuring element of size one (first row), computed using the sparse-NMF, the NNSAE, the NCAE and the AsymAE.

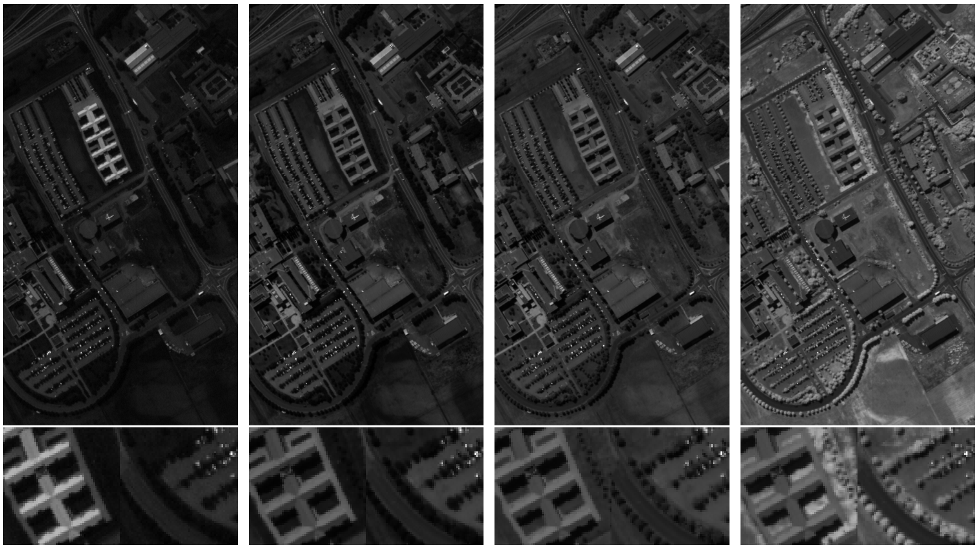


Figure 14: Four bands of the Pavia University hyperspectral image and two examples of patch per band.

We trained on these data the asymmetric auto-encoder presented earlier, with the same latent dimension ($k = 100$), same parameter $p = 0.05$ but larger $\beta = 0.005$. For comparison, as before, we also trained the sparse-NMF [11] and the NCAE [10] model, with the same parameters as before (those suggested by the authors). Despite all our attempts, we did not succeed in training the NNSAE [15] model to achieve sufficiently good performances so as to be interestingly compared to the other models. This might be a limitation of the model but could also be a misunderstanding on our part on how to set its parameters properly. We decided anyway no to report the obtained results, which were well below those presented hereafter. The two others deep-learning models were trained until they reached a reconstruction error of approximately 10^{-3} on the test. Regarding the sparse NMF, we observed that both the reconstruction error and the sparsity of the encoding could be easily controlled, and high quality results could be achieved that were out of reach for the online methods - at least during the tests we ran. Therefore, the sparse NMF shall be considered as a reference for the online methods, and this is why here we decided to apply it *a posteriori* to the whole dataset (training set and test set) targeting the best performance of the online models: a reconstruction error of approximately 10^{-3} and a sparsity of the encoding of approximately 0.7 (we set $S_h = 0.7$). In this comparison, the training set represented $\rho = 6/7$ of the whole set of patches.

Since the present experiment applies to richer data, the methods are compared on the four basic morphological operators (dilation, erosion, opening and closing) with several sizes of structuring elements. In order to enhance the differences across methods, we present the quality of the morphological approximations through the Peak Signal to Noise Ratio (PSNR), defined here by

$$PSNR = -10 \log_{10}(MSE), \tag{11}$$

where MSE is the pixel-wise mean squared error between the actual morphological operator and its part-based approximation. We recall that this comparison was made among models achieving a similar reconstruction error of the original images ($\approx 10^{-3}$) and a similar level of sparsity of the encoding (0.72 for our AsymAE, 0.75 for NCAE and the Sparse NMF). The plots of Figure 15 and Table 2 sum up the results, whereas Figures 16-20 provide visual examples for a structuring element of size three.

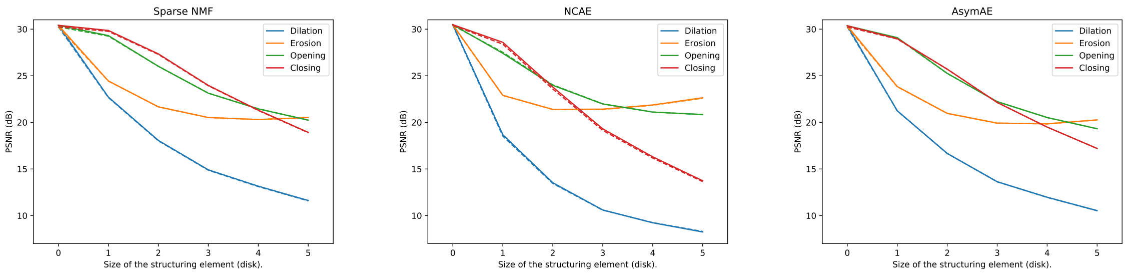


Figure 15: Quality of the approximation of different morphological operators on the test set (full lines) and training set (dashed lines), depending on the size of the structuring element (always a discrete disk). The quality of the approximation is expressed by the peak signal to noise ratio (PSNR, as defined by Eq. (11)). Higher is better. The size 0 corresponds the identity operator, showing therefore the reconstruction PSNR of the corresponding method. The results on training and test sets are almost identical. Here the proportion of the training set is $\rho = 6/7$. The figures are also shown in Table 2.

In generel, the Sparse NMF achieves the best part-based approximations and our model (AsymAE) is the best online method. This is the case except for the erosions of sizes two and onward, and the openings of sizes three and onward, where NCAE achieves better PSNRs than the AsymAE (and sometimes even than the Sparse NMF). This seems surprising as the visual examples for a structuring element of size three (Figures 16-20) do not show a better accuracy for the NCAE. These exceptions might have the same cause as the U-shape of the erosion curve, that we observe for all methods: for darker images, such as the eroded and openings of large structuring elements, the PSNR tends to favor over-dark approximations. In the limit case, it seems that

Table 2: Peak signal to noise ratio (PSNR, as defined by Eq. (11)) for the approximation of morphological operators on the test set for different models and different sizes of structuring elements (disks). Higher is better. The size zero corresponds to the identity operator, showing therefore the reconstruction PSNR of the model. The figures can also be visualized in the plots of Figure 15.

	Dilation				Erosion		
Size	AsymAE	NCAE	Sparse NMF	Size	AsymAE	NCAE	Sparse NMF
0	30.36	30.48	30.41	0	30.36	30.48	30.41
1	21.24	18.68	22.71	1	23.84	22.89	24.44
2	16.67	13.53	18.07	2	20.97	21.39	21.67
3	13.64	10.61	14.91	3	19.92	21.42	20.52
4	11.97	9.23	13.16	4	19.84	21.87	20.31
5	10.54	8.24	11.62	5	20.26	22.64	20.52

	Opening				Closing		
Size	AsymAE	NCAE	Sparse NMF	Size	AsymAE	NCAE	Sparse NMF
0	30.36	30.48	30.41	0	30.36	30.48	30.41
1	29.09	27.41	29.31	1	28.98	28.59	29.86
2	25.26	23.97	26.02	2	25.71	23.79	27.35
3	22.22	21.97	23.13	3	22.13	19.31	23.97
4	20.52	21.10	21.45	4	19.49	16.31	21.31
5	19.32	20.84	20.24	5	17.20	13.74	18.93

approximating such dark images by a constant zero-valued image yields a better PSNR than an approximation which would try to keep some structure.

As for the atom images, shown in Figures 21-23, they might not correspond to the intuition of a part-based representation, as their supports are quite extended. However there seems to be approximately one scale represented per atom, as in a granulometry decomposition, which is also a possible approximation of a part-based representation. Furthermore, we note that NCAE's atoms are the noisiest whereas the Sparse NMF's are the least noisy.

Another important remark is that the Sparse NMF could achieve even better results, but it still has the drawback of an offline method. By contrast, it is remarkable that both NCAE and AsymAE maintain almost exactly the same performance when we reduce the relative size of the training set down to $\rho = 0.5$. We do not report the results here as the difference with the presented ones is negligible. This shows the great interest of having a good online model when the training set is statistically representative of the whole data.

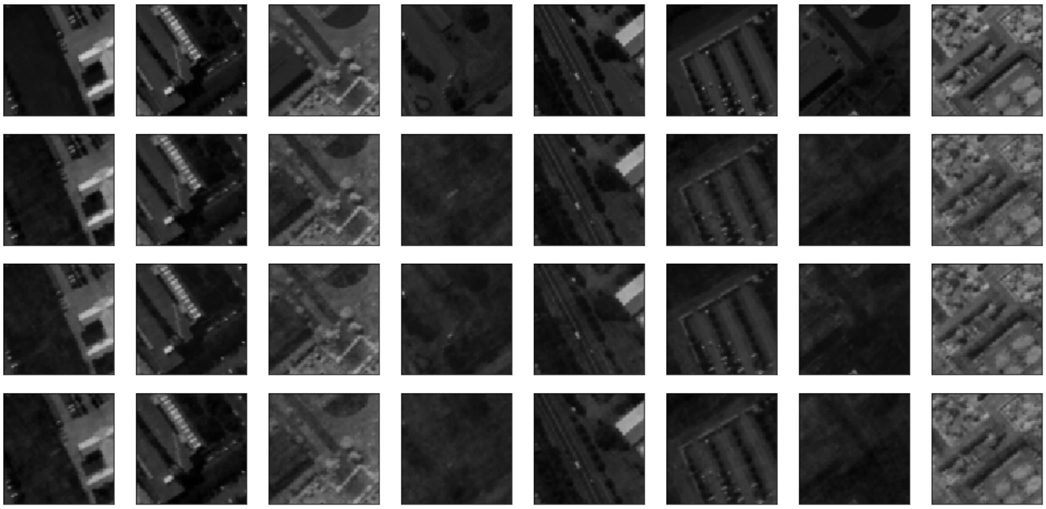


Figure 16: Examples of test patches (first row) and their reconstructions computed using the sparse-NMF, the NCAE and the AsymAE (from top to bottom).

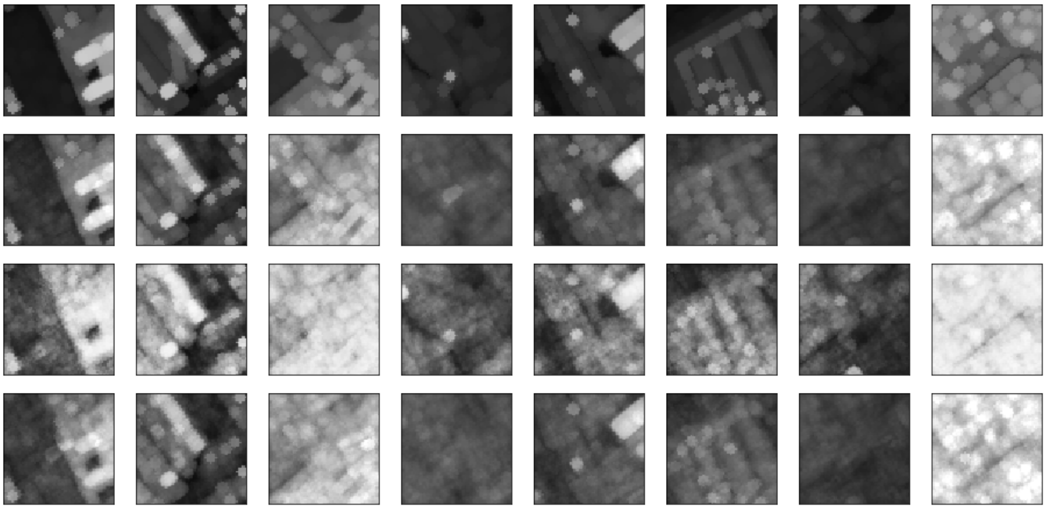


Figure 17: Part-based approximation of the dilation by a structuring element of size three. First row: dilation by a disc B of radius three(same patches as in Figure 16); following rows: approximation using the sparse-NMF, the NCAE and the AsymAE (from top to bottom).

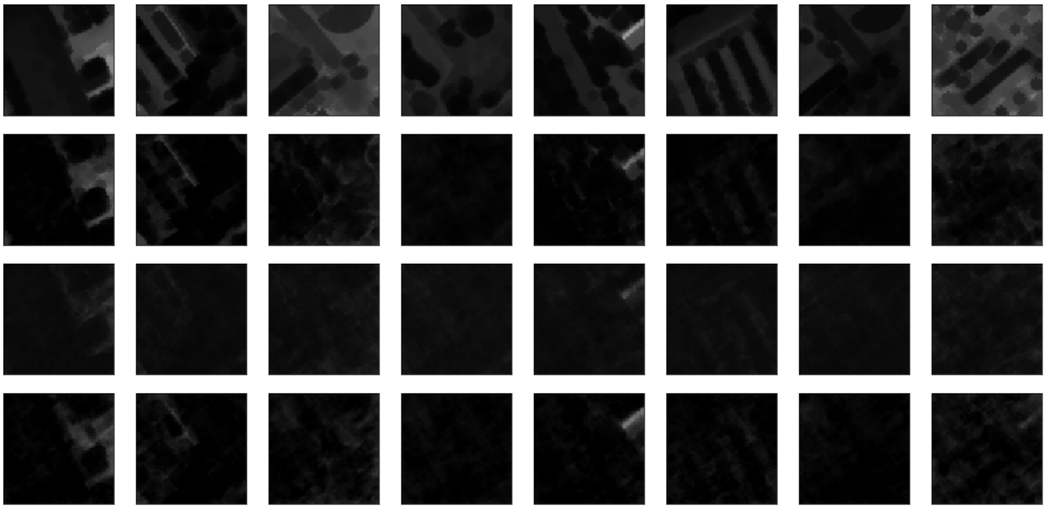


Figure 18: Part-based approximation of the erosion by a structuring element of size three. First row: erosion by a disc B of radius three; following rows: approximation using the sparse-NMF, the NCAE and the AsymAE (from top to bottom).

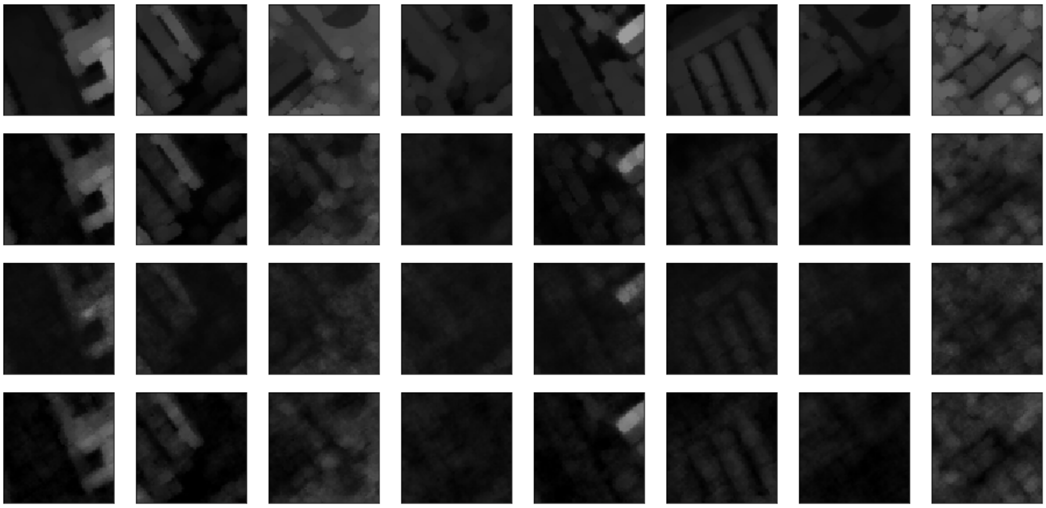


Figure 19: Part-based approximation of the opening by a structuring element of size three. First row: opening by a disc B of radius three; following rows: approximation using the sparse-NMF, the NCAE and the AsymAE (from top to bottom).

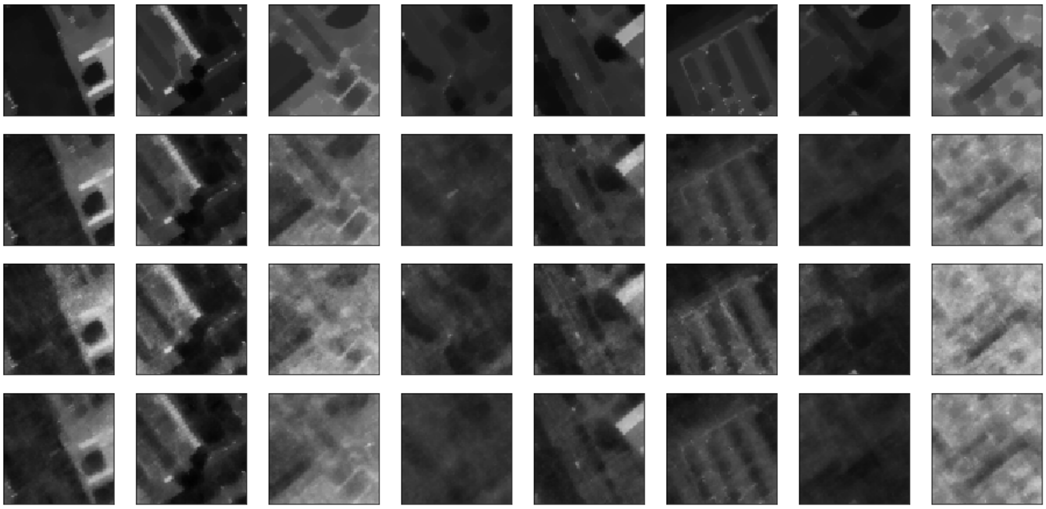


Figure 20: Part-based approximation of the closing by a structuring element of size three. First row: closing by a disc B of radius three; following rows: approximation using the sparse-NMF, the NCAE and the AsymAE (from top to bottom).

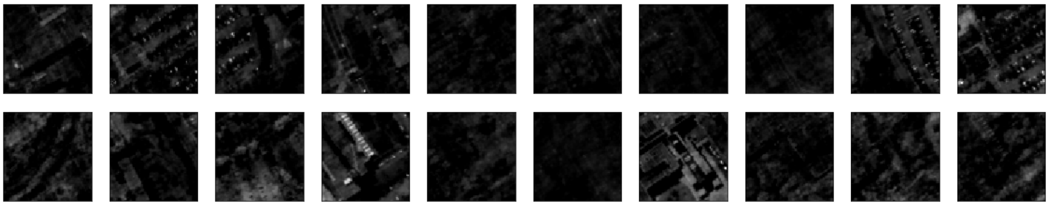


Figure 21: Examples of atoms for the sparse NMF in the experiment on the Pavia University image.

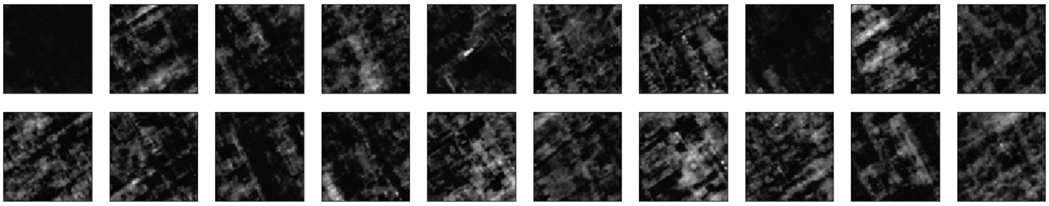


Figure 22: Examples of atoms for the NCAE auto-encoder in the experiment on the Pavia University image (proportion of the training set: $\rho = 6/7$).

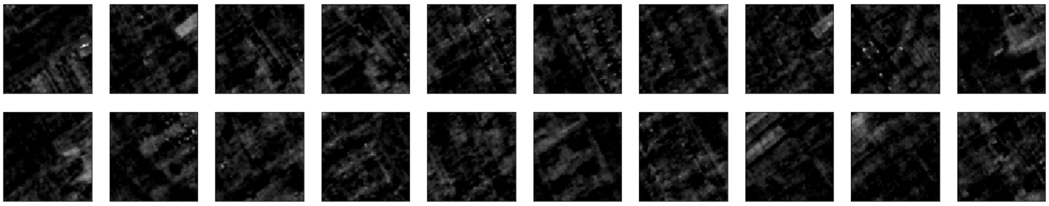


Figure 23: Examples of atoms for the AsymAE auto-encoder in the experiment on the Pavia University image (proportion of the training set: $\rho = 6/7$).

6 Conclusions and future works

We have presented an online method to learn a part-based dictionary representation of an image dataset, designed for accurate and efficient approximations of morphological operators. This method relies on auto-encoder networks, with a deep encoder for a higher reconstruction capability and a shallow linear decoder for a better interpretation of the representation. Among the online part-based methods using auto-encoders, it achieves the state-of-the-art trade-off between the accuracy of reconstructions and the sparsity of image encodings. Moreover, it ensures a strict (that is, non approximated) non-negativity of the learned representation. These results would need to be confirmed on color images, as the proposed model is scalable, but the illustration on the hyperspectral image already shows the potential use of the proposed approach in real applications. We especially evaluated the learned representation on an additional criterion, that is the commutation of the representation with morphological operators, and noted that all online methods perform worse than the offline sparse-NMF algorithm. A possible improvement would be to impose a major sparsity to the dictionary images with an appropriate regularization. Additionally, using a morphological layer [3, 21, 32] as a decoder may be more consistent with our definition of part-based approximation, since a representation in the (max, +) algebra would commute with the morphological dilation by essence.

Acknowledgments:
This work was partially funded by a grant from Institut Mines-Telecom and MINES ParisTech.

A Proof of Proposition 1

Proof. Let us first recall the definitions of δ_B and ε_B. The structuring element B is seen as a finite subset of $\mathbb{Z}^2$. We denote by $\check{B}$ its symmetric with respect to $(0,0)$ ($b \in \check{B} \iff -b \in B$) and B_u its translation by $u \in \mathbb{Z}^2$ ($b \in B_u \iff b - u \in B$). In the following we will identify the index i of a pixel with its coordinates in the image, and therefore denote by B_i the structuring element centered in pixel i. The adjoint dilation and erosion δ_B and ε_B are defined on any image $\mathbf{x} \in \mathcal{L}$ by

$$\forall i \in \{1, \dots, N\}, \quad \delta_B(\mathbf{x})_i = \bigvee_{j \in \check{B}_i} \mathbf{x}_j, \quad \varepsilon_B(\mathbf{x})_i = \bigwedge_{j \in B_i} \mathbf{x}_j. \tag{12}$$

Note that the extensivity of δ_B implies that $i \in B_i$ for any pixel i.

The conclusion of Proposition 1 is a consequence of the five points below:

1. For $h \geq 0$, $\delta_B(h\mathbf{w}) = h\delta_B(\mathbf{w})$ and $\varepsilon_B(h\mathbf{w}) = h\varepsilon_B(\mathbf{w})$.
   This is straightforward from the definitions in Equations (12). We shall remark however that this makes sense only because here $\mathbf{w} \in \mathcal{L}$ implies $h\mathbf{w} \in \mathcal{L}$.
2. If $\mathbf{x} \wedge \mathbf{y} = 0$ then $\mathbf{x} + \mathbf{y} = \mathbf{x} \vee \mathbf{y}$.
   Indeed, $\mathbf{x} \wedge \mathbf{y} = 0 \Rightarrow \forall i, \ \ \mathbf{x}_i = 0 \leq \mathbf{y}_i$ or $\mathbf{y}_i = 0 \leq \mathbf{x}_i \Rightarrow \forall i, \ \ \mathbf{x}_i + \mathbf{y}_i = \mathbf{x}_i \vee \mathbf{y}_i$.
3. If $\delta_B(\mathbf{x}) \wedge \delta_B(\mathbf{y}) = 0$ then $\mathbf{x} \wedge \mathbf{y} = 0$ as well, and therefore

   $$\delta_B(\mathbf{x} + \mathbf{y}) = \delta_B(\mathbf{x} \vee \mathbf{y}) = \delta_B(\mathbf{x}) \vee \delta_B(\mathbf{y}) = \delta_B(\mathbf{x}) + \delta_B(\mathbf{y}).$$

   The first implication is due to the extensivity of δ_B: $\mathbf{x} \leq \delta_B(\mathbf{x})$ and $\mathbf{y} \leq \delta_B(\mathbf{y})$ so $0 \leq \mathbf{x} \wedge \mathbf{y} \leq \delta_B(\mathbf{x}) \wedge \delta_B(\mathbf{y}) = 0$. Hence $\mathbf{x} \wedge \mathbf{y} = 0$ and point 2 yield the leftmost equality. The central equality is the defining property of dilations (as operations that commute with the supremum, the one in Equation 12 being a particular case). The rightmost equality is again point 2 applied to $\delta_B(\mathbf{x})$ and $\delta_B(\mathbf{y})$.
4. If $\delta_B(\mathbf{x}) \wedge \delta_B(\mathbf{y}) = 0$ then $\varepsilon_B(\mathbf{x} \vee \mathbf{y}) = \varepsilon_B(\mathbf{x}) \vee \varepsilon_B(\mathbf{y})$.
   This becomes clear by considering a pixel i and distinguishing three cases:
   Case 1: $\mathbf{x}_i = 0$ and $\mathbf{y}_i = 0$.

   Then $\varepsilon_B(\mathbf{x} \vee \mathbf{y})_i = 0 = \varepsilon_B(\mathbf{x})_i = \varepsilon_B(\mathbf{y})_i = \big(\varepsilon_B(\mathbf{x}) \vee \varepsilon_B(\mathbf{y})\big)_i$.
   Case 2: $\mathbf{x}_i > 0$.
   Then for any $j \in B_i$, $\mathbf{y}_j = 0$, otherwise there would be $j_0 \in B_i$ such that $\mathbf{y}_{j_0} > 0$ and therefore $\delta_B(\mathbf{y})_{j_0} \geq \mathbf{y}_{j_0} > 0$; since $i \in \check{B}_{j_0}$ we would also have $\delta_B(\mathbf{x})_{j_0} \geq \mathbf{x}_i > 0$ yielding $\delta_B(\mathbf{x})_{j_0} \wedge \delta_B(\mathbf{y})_{j_0} > 0$ which contradicts the initial hypothesis. We just showed $\mathbf{x}_i > 0 \Rightarrow \forall j \in B_i,\ \mathbf{y}_j = 0$, which also implies $\mathbf{y}_i = 0$ and $\varepsilon_B(\mathbf{y})_i = 0$. As a consequence, $\forall j \in B_i,\ \mathbf{x}_j \geq \mathbf{y}_j$ which leads to $\varepsilon_B(\mathbf{x} \vee \mathbf{y})_i = \bigwedge_{j \in B_i}(\mathbf{x}_j \vee \mathbf{y}_j) = \bigwedge_{j \in B_i} \mathbf{x}_j = \varepsilon_B(\mathbf{x})_i = \varepsilon_B(\mathbf{x})_i \vee \varepsilon_B(\mathbf{y})_i = \big(\varepsilon_B(\mathbf{x}) \vee \varepsilon_B(\mathbf{y})\big)_i$.
   Case 3: $\mathbf{y}_i > 0$.
   Then by symmetry the reasoning of Case 2 applies, and again $\varepsilon_B(\mathbf{x} \vee \mathbf{y})_i = \big(\varepsilon_B(\mathbf{x}) \vee \varepsilon_B(\mathbf{y})\big)_i$.
   Finally, in all cases we have $\varepsilon_B(\mathbf{x} \vee \mathbf{y})_i = \big(\varepsilon_B(\mathbf{x}) \vee \varepsilon_B(\mathbf{y})\big)_i$ and this is true for any pixel i, which achieves the proof of point 4.

5. If $\delta_B(\mathbf{x}) \wedge \delta_B(\mathbf{y}) = 0$ then $\varepsilon_B(\mathbf{x} + \mathbf{y}) = \varepsilon_B(\mathbf{x}) + \varepsilon_B(\mathbf{y})$.
   Indeed, like in point 3, $\delta_B(\mathbf{x}) \wedge \delta_B(\mathbf{y}) = 0$ implies $\mathbf{x} \wedge \mathbf{y} = 0$ thus point 2 applies: $\varepsilon_B(\mathbf{x} + \mathbf{y}) = \varepsilon_B(\mathbf{x} \wedge \mathbf{y})$, and applying point 4 yields $\varepsilon_B(\mathbf{x} + \mathbf{y}) = \varepsilon_B(\mathbf{x}) + \varepsilon_B(\mathbf{y})$.

With the five points listed here above, the conclusions of Proposition 1 are straightforward. Assuming H_1 is true:

- $D_B(\mathbf{x}) = \sum_{j=1}^{k} h_j \delta_B(\mathbf{w}_j) = \sum_{j=1}^{k} \delta_B(h_j \mathbf{w}_j) = \delta_B(\sum_{j=1}^{k} h_j \mathbf{w}_j) = \delta_B(\hat{\mathbf{x}})$, where point 1 was applied in the second equality and point 3 in the third equality. The first and last equalities are definitions.
- $E_B(\mathbf{x}) = \sum_{j=1}^{k} h_j \varepsilon_B(\mathbf{w}_j) = \sum_{j=1}^{k} \varepsilon_B(h_j \mathbf{w}_j) = \varepsilon_B(\sum_{j=1}^{k} h_j \mathbf{w}_j) = \varepsilon_B(\hat{\mathbf{x}})$, where point 1 was applied in the second equality, point 5 in the third equality. The first and last equalities are definitions.
- $G_B(\mathbf{x}) = \sum_{j=1}^{k} h_j \gamma_B(\mathbf{w}_j) = \sum_{j=1}^{k} h_j \delta_B\big(\varepsilon_B(\mathbf{w}_j)\big) = \sum_{j=1}^{k} \delta_B\big(\varepsilon_B(h_j \mathbf{w}_j)\big) = \delta_B\big(\sum_{j=1}^{k} \varepsilon_B(h_j \mathbf{w}_j)\big) = \delta_B\big(\varepsilon_B(\sum_{j=1}^{k} h_j \mathbf{w}_j)\big) = \gamma_B(\hat{\mathbf{x}})$, where point 1 was applied (twice) in the third equality, point 3 was applied to the $\varepsilon_B(h_j \mathbf{w}_j)$ in the fourth equality, since the $\varepsilon_B(h_j \mathbf{w}_j)$ verify H_1 as the $h_j \mathbf{w}_j$ do and $\varepsilon_B(h_j \mathbf{w}_j) \leq h_j \mathbf{w}_j$; and the fifth equality is given by point 5. The other equalities are definitions.
- $F_B(\mathbf{x}) = \sum_{j=1}^{k} h_j \varphi_B(\mathbf{w}_j) = \sum_{j=1}^{k} h_j \varepsilon_B\big(\delta_B(\mathbf{w}_j)\big) = \sum_{j=1}^{k} \varepsilon_B\big(\delta_B(h_j \mathbf{w}_j)\big)$, where point 1 was applied (twice) in the third equality.
  If the $\delta_B(h_j \mathbf{w}_j)$ comply with H_1, or equivalently if H_2 is true, then point 5 applies and we get $F_B(\mathbf{x}) = \varepsilon_B\big(\sum_{j=1}^{k} \delta_B(h_j \mathbf{w}_j)\big) = \varepsilon_B\big(\delta_B(\sum_{j=1}^{k} h_j \mathbf{w}_j)\big) = \varphi_B(\hat{\mathbf{x}})$.

□

References

[1] Jesús Angulo and Santiago Velasco-Forero. Sparse mathematical morphology using non-negative matrix factorization. In Pierre Soille, Martino Pesaresi, and Georgios K. Ouzounis, editors, *10th International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing (ISMM)*, volume LNCS 6671, pages 1–12, 2011.

[2] Babajide O. Ayinde and Jacek M. Zurada. Deep learning of constrained autoencoders for enhanced understanding of data. *CoRR*, abs/1802.00003, 2018.

[3] Vasileios Charisopoulos and Petros Maragos. Morphological perceptrons: Geometry and training algorithms. In Jesús Angulo, Santiago Velasco-Forero, and Fernand Meyer, editors, *13th International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing (ISMM)*, volume LNCS 10225, pages 3–15. Springer International Publishing, 2017.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016.

[5] Jan Chorowski and Jacek M. Zurada. Learning understandable neural networks with nonnegative weight constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 26(1):62–69, 2015.

[6] David Donoho and Victoria Stodden. When does non-negative matrix factorization give correct decomposition into parts? In *Advances in Neural Information Processing Systems 16 (NIPS)*, pages 1141–1148, 2004.

[7] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. *CoRR*, abs/1411.5928, 2014.

[8] Alaa El Khatib, Shimeng Huang, Ali Ghodsi, and Fakhri Karray. Nonnegative matrix factorization using autoencoders and exponentiated gradient descent. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

[9] Gianni Franchi, Amin Fehri, and Angela Yao. Deep morphological networks. *Pattern Recognition*, 102:107246, 2020.

[10] Ehsan Hosseini-Asl, Jacek M. Zurada, and Olfa Nasraoui. Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12):2486–2498, 2016.

[11] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *CoRR*, cs.LG/0408058, 2004.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[13] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[14] Honglak Lee, Chaitanya Ekanadham, and Andrew Y. Ng. Sparse deep belief net model for visual area v2. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 873–880. Curran Associates, Inc., 2008.

[15] Andre Lemme, René Felix Reinhart, and Jochen J. Steil. Online learning and generalization of parts-based image representations by non-negative sparse autoencoders. *Neural Networks*, 33:194–203, 2012.

[16] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*, page 3, 2013.

[17] Julien Mairal, Francis R. Bach, and Jean Ponce. Sparse modeling for image and vision processing. *CoRR*, abs/1411.3230, 2014.

[18] Petros Maragos and R Schafer. Morphological skeleton representation and coding of binary images. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(5):1228–1244, 1986.

[19] Martino Pesaresi and Jón A. Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(2):309–320, 2001.

[20] Bastien Ponchon, Santiago Velasco-Forero, Samy Blusseau, Jesús Angulo, and Isabelle Bloch. Part-based approximations for morphological operators using asymmetric auto-encoders. In Bernhard Burgeth, Andreas Kleefeld, Benoît Naegel, Nicolas Passat, and Benjamin Perret, editors, *14th International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing (ISMM)*, volume LNCS 11564, pages 323–334. Springer International Publishing, 2019.

[21] Gerhard X. Ritter and Peter Sussner. An introduction to morphological neural networks. In *13th International Conference on Pattern Recognition*, volume 4, pages 709 – 717, 1996.

[22] Yucong Shen, Xin Zhong, and Frank Y Shih. Deep morphological neural networks. *arXiv preprint arXiv:1909.01532*, 2019.

[23] Frank Y Shih, Yucong Shen, and Xin Zhong. Development of deep learning framework for mathematical morphology. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(06):1954024, 2019.

[24] Pierre Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[25] Keiji Tanaka. Columns for complex visual object features in the inferotemporal cortex: clustering of cells with similar but slightly different stimulus selectivities. *Cerebral Cortex*, 13 1:90–9, 2003.

[26] Fabian J Theis, Kurt Stadlthanner, and Toshihisa Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *13th IEEE European Signal Processing Conference*, pages 1–4, 2005.

[27] Marcos Eduardo Valle. Reduced dilation-erosion perceptron for binary classification. *Mathematics*, 8(4):512, 2020.

[28] Santiago Velasco-Forero and Jesús Angulo. Non-negative sparse mathematical morphology. In *Advances in Imaging and Electron Physics*, volume 202, chapter 1, pages 1 – 37. Elsevier Inc.Academic Press, 2017.

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

[30] Yonghua Yin and Erol Gelenbe. Non-negative autoencoder with simplified random neural network. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6, 2019.

[31] Li Zhang and Yaping Lu. Comparison of auto-encoders with different sparsity regularizers. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5, 2015.

[32] Yunxiang Zhang, Samy Blusseau, Santiago Velasco-Forero, Isabelle Bloch, and Jesús Angulo. Max-plus operators applied to filter selection and model pruning in neural networks. In Bernhard Burgeth, Andreas Kleefeld, Benoît Naegel, Nicolas Passat, and Benjamin Perret, editors, *14th International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing (ISMM)*, volume LNCS 11564, pages 310–322. Springer International Publishing, 2019.