logarithmic range from 10^{-2} to 10^3 were tested, and the model with the highest classification accuracy, on a 15 folds cross-validation, was selected. A model with this value of the parameter was then re-trained and evaluated on a 25 folds cross-validation on left out data.

- The sparsity of the encoding, measured using the mean on all test images of the sparsity metric introduced by P.O. Hoyer in [13] (2004) and presented in section 2.2, based on the relationship between the L_1 and the L_2 norm:

$$S(\mathbf{h}^{(i)}) = \frac{\sqrt{k} - \frac{||\mathbf{h}^{(i)}||_1}{||\mathbf{h}^{(i)}||_2}}{\sqrt{k} - 1}$$

- The Max-Approximation error to dilatation by a disk of radius 1, obtained by computing the mean squared error between the dilatation by a disk of radius 1 of the original image and the max-approximation error to the same dilatation, using the learned representation.

The three first metrics are quite commonly used in the field of representation learning. In particular they were used by the related works mentioned in Section 1.5.

In the following sections, we will first expose in further details the Non-Negative Matrix Factorization, as well as its sparse variant introduced by P.O.Hoyer in [13] (2004), used by J.Angulo and S.Velasco-Forero in [22] and [1], and its performance on our data set. We will then present how we tried to implement a similar sparse part-based representation using auto-encoders, using firstly a shallow architecture, and then an asymmetric architecture with a deep encoder, along with the performance of the implemented architectures.

2 Non-Negative Matrix Factorization

2.1 General Presentation and interests

The first Non-Negative Matrix Factorization algorithm was introduced by D.D. Lee and H.S. Seung in 1999 [15] as a way to find a set of basis functions for representing non-negative data. They claimed that the notion is particularly applicable to image articulation libraries made up of images showing a composite object in many articulations and poses. They suggested (in the very title of the article) that when used in the analysis of such data, NMF would find the intrinsic 'parts' going to make up the object being pictured. They based their methods on the observation of psychological and physiological evidence for part-based representations in the brain, which can be conceptually tied to the non-negativity constraints.

The NMF is a factoriation matrix algorithm, just like Principal Component Analysis (PCA). But the latter, whose atoms are the eigen vectors associated with the largest eigen values of the covariance matrix of the data, and therefore are the directions along which the variance of the data is maximal, produces distorted versions of the input images. NMF on the opposite produces atoms corresponding to localized features that correspond better with the intuitive notion of parts, only additive combinations of the atoms being allowed. The variability of the data is generated by combining these different parts, without using all of them for each image, even though all atoms are used by at least one image. Hence this representation respects a latent sparsity constraint, which is explicitly constraint in the variant we used as a baseline in our study [13], as we will see in the following section.

The decomposition of the data set of images into $\hat{\mathbf{X}} = \mathbf{HW} \approx \mathbf{X}$ is obtained though very simple update rules of both the encoding and the atom matrices, that preserve their non-negativity and also constrain the norm of $\mathbf{W}$ as the decomposition into $\mathbf{HW}$ is invariant to scaling of both $\mathbf{H}$ and $\mathbf{W}$. This iterative algorithm is proven to monotonically converge to a local maximum of the objective function:

$$\sum_{i=1}^{M}\sum_{n=1}^{N}(\mathbf{X}_{i,n}\log(\mathbf{HW})_{i,n} - (\mathbf{HW})_{i,n})$$

A work from D.Donoho and V.Stodden in 2003 [6] explains how and when this Non-Negative Matrix Factorization gives a correct decomposition into parts. The paper shows that the NMF algorithm actually recovers the parts of the images if the data set of images is a *separable factorial articulation family*, that is follows the three following properties:

- Generative Model: each image is a actually generated by a linear combination of positive atom images associated with non-negative weights.
- Separability: for a given pixel, there is only one atom image whose value is not null at this pixel, that is the atom images all have separated supports.
- Complete Factorial Sampling: all different combinations of parts are exhaustively sampled.

[6] theoretically and empirically shows that in such Separable Factorial Articulation Families, non-negative factorization is effectively unique (up to scaling and permutation of the atom images). In such libraries, NMF will indeed successfully 'find the parts'.

Note however that our Fashion-MNIST data probably does not verify the properties of a Separable Factorial Articulation Families, as a set of $k = 100$ separable atom images cannot easily be explicitly constructed.

Many variants were built on top of the original algorithm, usually producing additional constraints and properties: local NMF (enforcing atom images containing localized features), sparse NMF (enforcing sparse encoding or atom images), non linear variants, binary variants, etc. In the following section, we will present one those variants, that was the one used by the authors of [22] when they presented the framework of max-approximation to morphological operators.

2.2 Addition of sparsity constraints (Hoyer 2004)

The idea of incorporating sparse constraints to NMF, is to construct a succinct representation of the image data as a combination of a few typical patterns (few atoms of the dictionary) learned from the data itself. By enforcing each images to be represented by a weighted combination of a small number of atoms, we expect those atoms to be closer to being pair-wise disjoint, and to verify equation (1), and as consequence to verify equation (2) which is sufficient for our max-approximations to be palatable approximations of the morphological operators. Moreover, there exist conceptual motivations behind sparsity, as stated in [20], [18] and [16], as attempts to reproduce the receptive fields found in the visual cortex of human brain.

In the case of NMF, the sparse variant we will use as a baseline is the one introduced by P.O.Hoyer in 2004 [13], and stems from the observation that NMF not always produced an intuitive decomposition into parts that would correspond to the human idea of the 'building blocks' of the data and is based on a new sparseness measure involving the relationship between the L_1 norm and the L_2 norm, defined, for some vector $\mathbf{v}$, as:

$$S(\mathbf{v}) = \frac{\sqrt{|\mathbf{v}|} - \frac{||\mathbf{v}||_1}{||\mathbf{v}||_2}}{\sqrt{|\mathbf{v}|} - 1} \tag{3}$$

where $|\mathbf{v}|$ is the dimensionality of the vector $\mathbf{v}$. This function evaluates to unity if and only if $\mathbf{v}$ contains only a single non-zero entry, and takes a value of zero if and only if all components are equal (up to signs), interpolating smoothly between the two extremes.

The algorithm proposed by [13] constrains the sparsity measure of each image encoding $S(\mathbf{h}^{(i)}), i \in [1, M]$ and/or of each atom $S(\mathbf{w}_j), j \in [1, k]$ to be equal to fixed values S_h and S_w respectively. This is done by reprojecting the weights (atoms) and encoding matrices on the corresponding space after each update.

2.3 Implementation and application to the data

I used the same Matlab implementation provided by [13] as the one used by the authors of [22]. I opted for not applying any sparsity constraint on the atom images (note that this is not equivalent to setting $S_w = 0$) , after trying several values, I chose to show the results for $S_h = 0.6$, for the fixed latent dimension $k = 100$ I will use throughout my study, in Figures 1 and 2. We note that despite sparsity and non-negativity constraints, leading to a really nice part-based decomposition, with each atom showing localized features, even though the support of the supports are not pair-wise separable, the approximation is quite close to the original images, in spite of some loss in the texture information.

As we can see in Figure 1c, most coefficients in the encoding of each image are close to 0 or very small. This leads to the quite accurate max-approximation to dilatation by a disk of radius 1 that can be seen in Figure 2.

3 Part-Based representation using Auto-Encoders

An **Auto-Encoder** is a Neural Network that performs representation learning by approximating the identity function, that is learning a function $h_{\mathbf{W},\mathbf{b}}(\mathbf{x}) = \hat{\mathbf{x}} \approx \mathbf{x}$ where $\mathbf{x} \in \mathcal{R}^N$ is an input of dimension N, $\hat{\mathbf{x}}$ its image by the auto-encoder, and $\mathbf{W}$ and $\mathbf{b}$ respectively the **weights** and **biases** of the network. In our case, the dimension of the input data points N will be equal $d \times d \times c$, as we will deal with images of size d by d with c channels (more precisely in the case of the Fashion-MNIST data set: $d = 28$, $c = 1$). Internally, the network has a hidden layer $\mathbf{h} \in \mathcal{R}^k$ that describes a code of fixed size k used to represent the input. It is composed of two sub-networks:

- An **encoder** $f_{\mathbf{W}_e,\mathbf{b}_e} : \mathbf{x} \longmapsto \mathbf{h} \in \mathbb{R}^k$, that learns a representation of the input of dimension k.
- A **decoder** $g_{\mathbf{W}_d,\mathbf{b}_d} : \mathbf{h} \longmapsto \hat{\mathbf{x}}$ that learns to reconstruct the input from an encoding of it.

We therefore aim at finding the parameters of the functions f and g, $\mathbf{W} = (\mathbf{W}_e, \mathbf{W}_d)$ and $\mathbf{b} = (\mathbf{b}_e, \mathbf{b}_d)$, that minimize the mean of the **Reconstruction Error** $L(\hat{\mathbf{x}}^{(i)}, \mathbf{x}^{(i)}) = L(h(\mathbf{x}^{(i)}), \mathbf{x}^{(i)}) = L(g(f(\mathbf{x}^{(i)})), \mathbf{x}^{(i)})$ over all the M points $\mathbf{x}^{(i)}$ of the data set (we omit the indexation of the function by their parameters for the sake of readability). The **objective function** (that we will call the **loss** of the auto-encoder) we try to minimize is thus:

$$L_{AE} = \frac{1}{m} \sum_{j=1}^{m} L(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)})$$

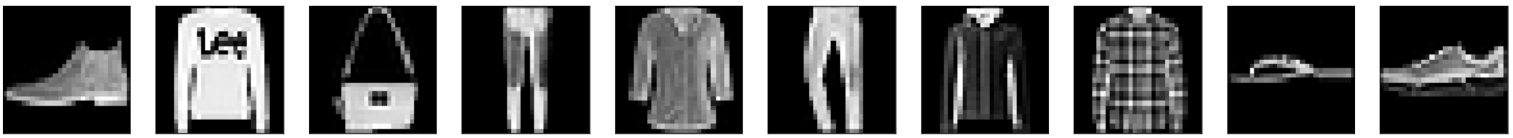
The reconstruction error function L can be of various types:

- **Mean-Squared Error**: $L(\hat{\mathbf{x}}^{(i)}, \mathbf{x}^{(i)}) = \frac{1}{N} ||\hat{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}||_2^2$
- **Binary-Cross Entropy**: $L(\hat{\mathbf{x}}^{(i)}, \mathbf{x}^{(i)}) = \frac{1}{N} \sum_{l=1}^{N} \left(x_l^{(i)} \log \hat{x}_l^{(i)} + (1 - x_l^{(i)}) \log(1 - \hat{x}_l^{(i)}) \right)$
- etc.

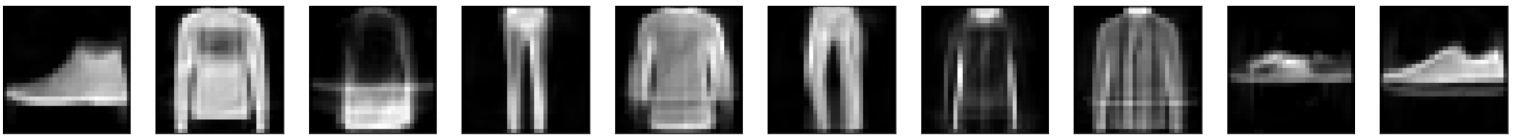
The Mean-Squared Error was our implementation choice for the design of our model.

The capacity of the model increases with the dimension of the encoding and with the depth of the auto-encoder, but when the size of the encoding is chosen to be smaller than the dimension of the input, the so-called **under-complete** auto-encoder performs a form of dimensionality reduction that can capture hidden structure in the input.

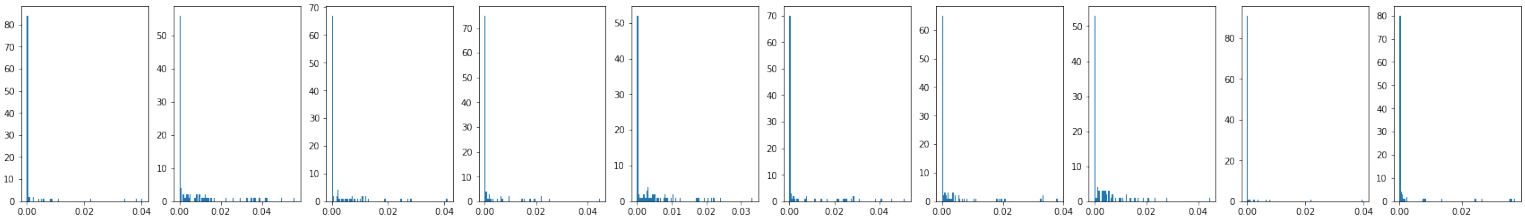
Another way to perform an efficient representation learning is to add other types of constraints on the model (sparsity of the representation, smallness of the derivatives of the representation, robustness to noise, robustness to adversarial examples, etc.)



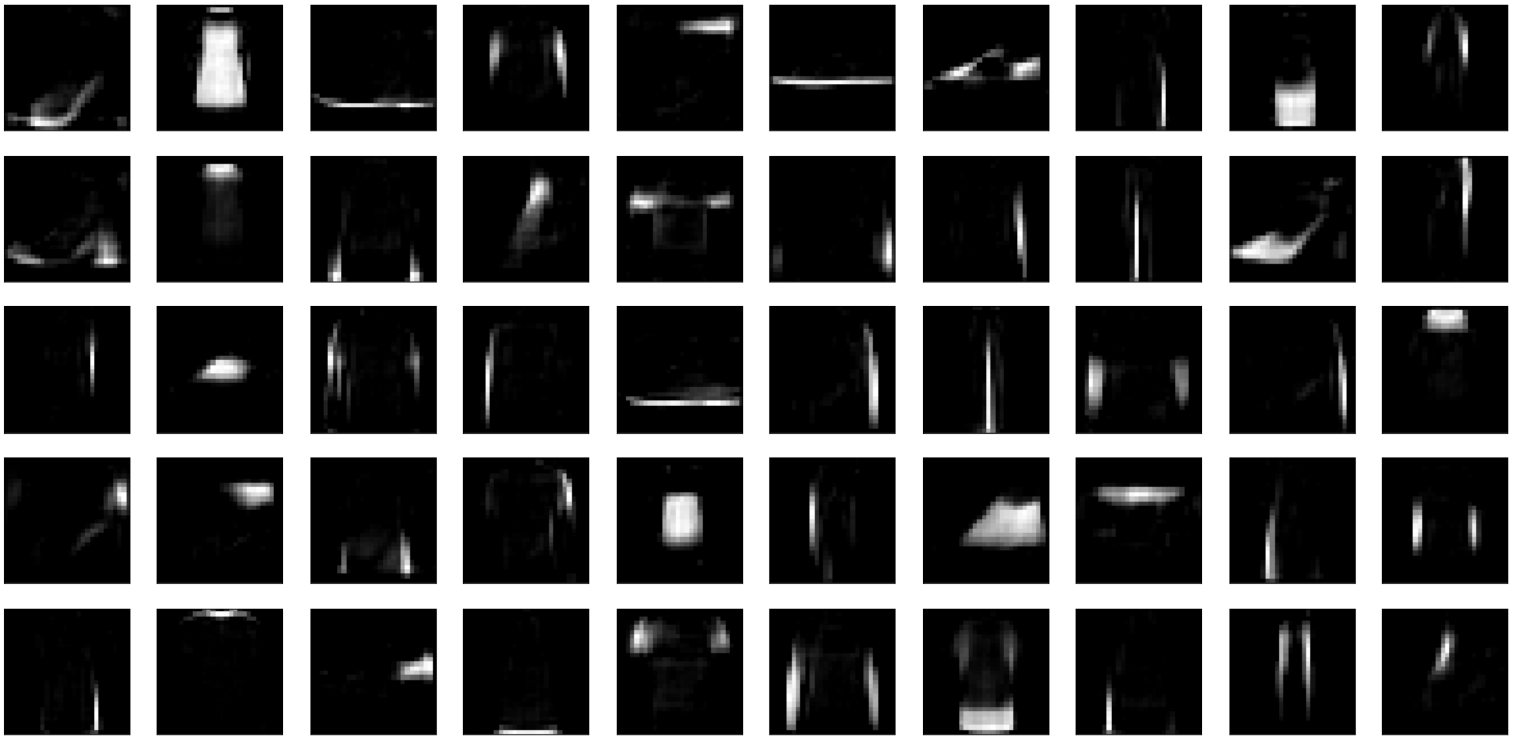
(a) Original Images



(b) Approximation (reconstruction) of the original images by the sparse NMF - *Reconstruction error: 0.0109*

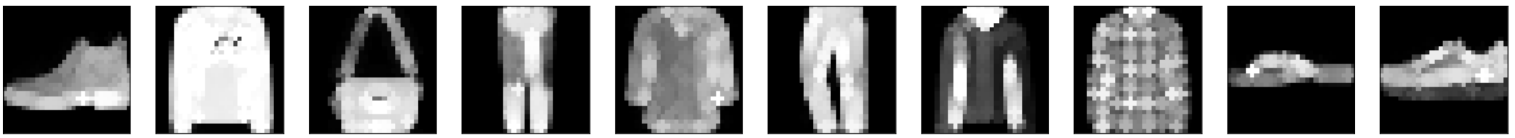


(c) Histogram of the encoding of each of the images - *Sparsity (Hoyer 2004): 0.650*

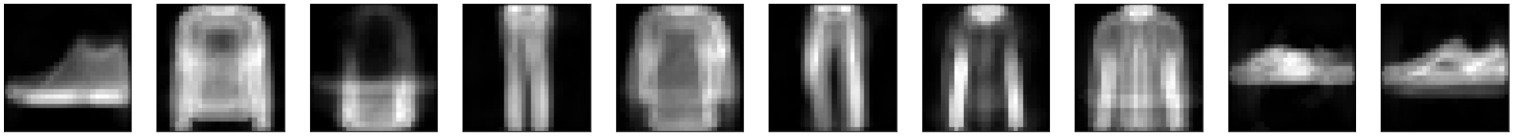


(d) 50 of the $k = 100$ learned dictionary images (atoms)

Figure 1: Part-based representation learned using the Hoyer NMF with sparsity constraint on the encoding: $S_h = 0.6$



(a) Dilatation of the original images by a disk of radius 1



(b) Max-approximation to a disk of radius 1 - *Max-approximation error: 0.107*

Figure 2: Max-approximation to dilatation of a disk of radius 1 using the representation learned with Hoyer NMF with sparsity constraint on the encoding: $S_h = 0.6$[13] [22]

3.1 Introduction using a shallow architecture

For the sake of simplicity and understandability of the method, we first tried to perform an efficient part-based representation using **shallow** Auto-Encoders, that is, encoders and decoders both composed of only two densely connected layers of neurons (input and output) as represented in Fig 3.

We hence have an auto-encoder of the form:

$$\begin{aligned}\hat{\mathbf{x}} &= h(\mathbf{x}) \\ &= g(f(\mathbf{x})) \\ &= g(\mathbf{h}) \\ &= \sigma_d\left(\mathbf{W}_d^T\mathbf{h} + \mathbf{b}_d\right) \\ \mathbf{h} &= f(\mathbf{x}) \\ &= \sigma_e\left(\mathbf{W}_e^T\mathbf{x} + \mathbf{b}_e\right)\end{aligned}$$

Where σ_e and σ_d are two **non-linear** functions applied element-wise to their input vectors. Once our network trained, we hence can represent each image of our database as the image by σ_d of a linear combination of k **atom images** $\mathbf{W}_{d,1}, \mathbf{W}_{d,2}, ..., \mathbf{W}_{d,k}$ (each row of $\mathbf{W}_d$ is an atom image flatten in a $(1, N)$ row vector), to which a bias image $\mathbf{b}_d$ is added (presented as flatten $(N, 1)$ column vecotr). Therefore the weights of the decoder constitute a dictionary of images, similarly to the learned representation in the NMF of section 2.

When applied to a batch of M points, represented as a design matrix $\mathbf{X}$ of size $M \times N$ (N is the input dimension), the computation performed by our feed-forward network can be written:

$$\begin{aligned}\hat{\mathbf{X}} &= h(\mathbf{X}) \\ &= g(f(\mathbf{X})) \\ &= g(\mathbf{H}) \\ &= \sigma_d\left(\mathbf{H}\mathbf{W}_d + \mathbf{b}_d^T\right) \\ \mathbf{H} &= f(\mathbf{X}) \\ &= \sigma_e\left(\mathbf{X}\mathbf{W}_e + \mathbf{b}_e^T\right)\end{aligned}$$

where the addition by the bias vectors $\mathbf{b}_e$ and $\mathbf{b}_d$, of size $k \times 1$ and $N \times 1$ respectively.

Note that if no further constraints are applied, if σ_e is the identity function, and if the reconstruction error is chosen to be the mean squared-error, then the under-complete auto-encoder learns to span the same space as PCA.