1 Introduction

1.1 Context

The aim of this internship was to study the interactions between two widely used but very different non-linear methods for image processing: deep-learning and mathematical morphology. We chose to focus our work on learning efficient representations of images, and more specifically representations mimicking human cognition of natural images, and that could be used to approximate the application of morphological operators to a large set of images at a lesser cost. Finally we tried to study the impact of feeding our neural network with morphological multi-scaled decompositions of images.

1.2 Representation Learning and Part based representation

Representation Learning [3] stems from the need to find an underlying structure/process explaining the data, that can somehow be represented as a set of latent features. In the case of probabilistic models, a good representation is often one that captures the posterior distribution of the underlying explanatory factors for the observed input. The assumptions usually being that the data points live on a manifold of lesser dimension than the original space, these explanatory factors are hence points in a space of smaller dimension than the input data. Finding a good representation therefore usually comes with reduced storage and computation costs, along with a solution to the curse of dimensionality, that causes most learning models to over-fit when their input data are points in a space of too great dimension. It thus comes with no surprise that representation learning has become in the past year a major field of study in the machine learning community.

Sparse coding and dictionary learning [18] are branches of representation learning where data is assumed to be well represented as a linear combination of a few elements from a dictionary. It is an active research active that leads to state-of-the-art results in image processing applications, such as image denoising, inpainting, demosaicking or compression. If we represent by $\mathbf{X} \in E^{M \times N}$ our data set of M images of N pixels, it boils down to the matrix factorization:

$$\mathbf{X} \approx \mathbf{HW}$$

where $\mathbf{H} = (h_{i,j})_{i,j} \in \mathbb{R}^{M \times k}$ are the matrix holding the encoding of each of the input images, that is the latent features and each row of $\mathbf{W} \in \mathcal{R}^{k \times N}$ is a dictionary images. In other words each image $\mathbf{x^{(i)}} = \mathbf{X}_{i,:}$ of the set can be written as a weighted linear combination of the atoms $\mathbf{w}_j = \mathbf{W}_{j,:}$ of the dictionary:

$$\begin{aligned}\forall i \in [1, M], \mathbf{x}^{(i)} &= \mathbf{H}_{i,:}\mathbf{W} \\ &= \mathbf{h}^{(i)}\mathbf{W} \\ &= \sum_{j=1}^{k} h_{i,j}\mathbf{w}_j\end{aligned}$$

In term of representation it means that each image $\mathbf{x}^{(i)}$ of $\mathbf{X}$, of N pixels, can be represented as a point $\mathbf{h}^{(i)}$ in a space of dimension k. In addition, the sparsity of the encoding in enforced, which means that we enforce most of coefficients of $\mathbf{h}^{(i)}$ to be zero (or close to zero), so that only a few of the atoms of the image are really needed to approximate $\mathbf{x}^{(i)}$. The dictionary of images is fixed for all the M images of our set. The assumption behind this decomposition is that the more similar the images of the set, the smaller the required dimension to accurately approximate it. 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.

Finally, the notion of "part" based representation [21] was introduced by Daniel D.Lee and H. Sebastian Seung in a 1999 *Nature* article ([15]) where they proposed the first Non-Negative Matrix

Factorization algorithm (NMF). The aim was to perform dictionary learning in such a way that the learned atom images would represent localized features corresponding with intuitive notions of the parts of the input image family, such as parts of faces, in the case of the face database that the authors used for the first experiment of their algorithm. We see more about NMF and one of its variant in Section 2.

The aim of my internship was to use deep auto-encoder models to learn similar part-based representation of our input images, and to see how this representation could be used to approximate morphological operators, by applying them on the basis images, as we will see in the newt subsections.

1.3 Some reminders on Mathematical Morphology

Mathematical morphology (Serra, 1982, 1988; Heijmans, 1994) [4] is a nonlinear image processing methodology based on the application of lattice theory to spatial structures. It can be applied to spaces with a complete lattice structure (that is partially ordered with definition of infimum and supremum): sets (example: binary images), functions (example: grey-scale images). We will here only focus on the definition of morphological operators on space of functions, and more precisely on grey-scale images. Note that we could have also work only with binary images, as numerical grey-scale images (whose pixel intensities are quantified in a number of grey-scale) can be also considered as level set decompositions made of binary images. Morphological operators are defined using very intuitive geometrical notions which allows us the perceptual development and interpretation of complex algorithms by combination of various operators.

The two fundamental morphological operators are the **dilatation** and the **erosion**, which are defined respectively as the operators that commutes with the supremum and the infimum. We note $\mathbf{x}$ a grey-scale image, which can be seen as a function from a subset E of the discrete space $\mathbb{Z}^2$ (a grid of pixels, considered as the support space of the 2D image) to $\mathbb{R}^+$. Note that in practice, the value of the intensity of the pixels of a numerical images is not a continuous space but is quantified into a $[l_0, ..., l_L]$ discrete space, where L is the number of grey scales l_i in the numerical encoding of images, however both types of function spaces are complete lattices and assuming the pixel intensities to be in $\mathbb{R}^+$ does not change what follows. Given a symmetric structuring element $SE \subset E$ (we place ourselves in the case of **flat** mathematical operators on functions, which means that structuring elements are chosen to be subset of E, that can be seen as well as binary images), the dilatation δ_{SE} of the image $\mathbf{x}$ by the structuring elements SE can be defined as the image:

$$\delta_{SE}(\mathbf{x}) : i \in E \longmapsto \bigvee_{z \in SE(i)} \mathbf{x}(z)$$

Where $SE(i)$ denotes the structuring element translated by i, that is, $SE(i) = \{z \in E, z - i \in SE\}$ (centered at pixel i if SE is centered on the original). $\bigvee$ and $\bigwedge$ are respectively the supremum and infimum operations. Dilatation is the counterpart of convolution in max-plus algebra.

Similarly, we define the erosion ϵ_{SE} of the image $\mathbf{x}$ by the structuring element SE as the image:

$$\epsilon_{SE}(\mathbf{x}) : i \in E \longmapsto \bigwedge_{z \in SE(i)} \mathbf{x}(z)$$

Erosion and dilatation by a same symmetric structural elements are dual by inversion, that is:

$$\delta_{SE}(\mathbf{x}) = n\left(\epsilon_{SE}(n(\mathbf{x}))\right)$$

where n is an inversion: a bijection from the image space to the image space, such that if $\mathbf{x}$ and $\mathbf{y}$ are two images, $\mathbf{x} < \mathbf{y} \iff n(\mathbf{x}) > n(\mathbf{y})$. This property can be used to some results applying to the dilatation also are true for the erosion.

Now let us define the **opening** and **closing** by a structuring element SE, which are respectively the composition of an erosion and a dilatation, and the composition of a dilatation and an erosion:

$$\gamma_{SE}(\mathbf{x}) = \epsilon_{SE}\left(\delta_{SE}(\mathbf{x})\right)$$
$$\phi_{SE}(\mathbf{x}) = \delta_{SE}\left(\epsilon_{SE}(\mathbf{x})\right)$$

These two idempotent operators are dual by inversion just like (dilatation and erosion). Qualitatively, opening darkens narrow bright zones of the image (like small object protuberance), while closing brightens narrow dark zones (like "holes" and thin cavities).

Morphological filters and transformations are useful for various image processing tasks, such as denoising, contrast enhancement, multi-scale decomposition, feature extraction and object segmentation.

There exists many other morphological operators, some of them have been considered within this study (skeleton), but are not included in this report, others (reconstructions by dilatation/by erosion, and opening/closing by reconstruction) will be presented later.

1.4 Max-Approximation to morphological operators

One of the core ideas of this internship rises from two works from J.Angulo and S.Velasco-Forero ([22] and [1]), whose aim was to explore how image sparse representations can be useful to efficiently calculate approximations to morphological operators. The main motivation is to be able to apply morphological operators to massive sets of images by applying them only to the reduced set of images of the representation, in such a way that the original processed images are approximately obtained by projecting back to the initial space. This framework gets more efficiency and interest if the number of images to process is larger than the number of atoms and if many operators are to be applied.

In this context, the authors base their work on a sparse variant of the NMF decomposition introduced by Hoyer in 2004 [13], which we will explain in more depths in Section 2.2. Let us consider a family of M images (binary or gray-scale) $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, ..., $\mathbf{x}^{(M)}$, whose linearized form are aggregated in a $M \times N$ data matrix $\mathbf{X}^T = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M)$ (the i^{th} row of $\mathbf{X}$ is the transpose of the linearized version of $\mathbf{x}^{(i)}$, and $|\mathbf{x}^{(i)}| = N$ is the number of pixels in all the images). The sparse NMF algorithm gives an linear non-negative approximate data decomposition of $\mathbf{X}$ into two matrices $\mathbf{H} \in \mathbb{R}_+^{M\times k}$ and $\mathbf{W} \in \mathbb{R}_+^{k\times N}$:

$$\mathbf{X} \approx \mathbf{H}\mathbf{W} = \hat{\mathbf{X}}$$

Each of the lines of $\mathbf{W}$ contains a basis vector $\mathbf{w}_j$ (which is a linearized image of N pixels), and each row of $\mathbf{H}$ contains the coefficient vector $\mathbf{h}^{(i)}$ corresponding to image $\mathbf{x}^{(i)}$. We say that the matrix $\mathbf{W}$ contains the dictionary, and $\mathbf{H}$ the encoding of the decomposition.

Recall from Section 1.2 that, each image $\mathbf{x}^{(\mathbf{i})}$ of the set can be written as a weighted linear combination of the atoms $\mathbf{w}_j$ of the dictionary:

$$\forall i \in [1, M], \mathbf{x}^{(i)} = \sum_{j=1}^{k} h_{i,j}\mathbf{w}_j$$

The authors then define the **Sparse Max-Approximation to gray-level dilatation and ero-**

sion respectively as:

$$\begin{aligned} D_{SE}(\mathbf{x}^{(i)}) &= \sum_{j=1}^{k} h_{i,j}\delta_{SE}(\mathbf{w}_j) \\ E_{SE}(\mathbf{x}^{(i)}) &= \sum_{j=1}^{k} h_{i,j}\epsilon_{SE}(\mathbf{w}_j) \\ &= \sum_{j=1}^{k} h_{i,j} n\left(\delta_{SE}(n(\mathbf{w}_j))\right) \end{aligned}$$

where the erosion and dilatation are applied to the atom gray-scaled images (in 2D, the same notation is kept for the images and their linearized column-vector versions, for the sake of simplicity).

Note that these approximate non-linear operators do not satisfy the standard properties of gray-level dilatation and erosion.

Similarly, the authors define the **Sparse Max-Approximation to gray-level opening and closing**, respectively as:

$$\begin{aligned} G_{SE}(\mathbf{x}^{(i)}) &= \sum_{j=1}^{k} h_{i,j}\gamma_{SE}(\mathbf{w}_j) \\ F_{SE}(\mathbf{x}^{(i)}) &= \sum_{j=1}^{k} h_{i,j}\phi_{SE}(\mathbf{w}_j) \\ &= \sum_{j=1}^{k} h_{i,j} n\left(\gamma_{SE}(n(\mathbf{w}_j))\right) \end{aligned}$$

As the dilatation commutes with the supremum of functions, and as the support of our positive functions being pairwise-disjoint is a sufficient condition for the commutation of opening with the supremum, the authors state and empirically show that a good non-redundant sparse-NMF decomposition yields a good max-approximation to the dilatation and to the opening. Indeed, this is when the part-based decomposition, along with the sparsity and non-negativity of the learned representation comes into play. As we will see in Section 2, the non-negativity prevents mutual cancellations between parts of the atom images, leading to part-based representations. Moreover the sparsity constraint enforces the decomposition of each image to use fewer atoms, and therefore enforces the atoms not to overlap in their supports. As a result, weighted atoms of the decomposition of an image can be considered almost pair-wise disjoint, that is:

$$\forall i \in [1, M], \forall (j,l) \in [1,k]^2, h_{i,j}\mathbf{w}_j \bigwedge h_{i,l}\mathbf{w}_l \approx 0$$

as sparsity is enforced on the encoding, and as NMF ensures a part based decomposition. The supremum of pair-wise disjoint functions is equal to their sum, which implies that the decomposition by sum performed by the NMF is close to a decomposition by max (the atoms does not need to be pair-wise disjoint as long as the weighted atoms of the decomposition of an image are):

$$\forall i \in [1, M], \bigvee_{j \in [1,k]} h_{i,j}\mathbf{w}_j \approx \sum_{j=1}^{k} h_{i,j}\mathbf{w}_j$$

This motivates the framework proposed by the authors:

$$\begin{aligned}
\delta_{SE}(\mathbf{x}^{(i)}) &= \delta_{SE}\left(\sum_{j=1}^{k} h_{i,j}\mathbf{w}_j\right) \\
&\approx \delta_{SE}\left(\bigvee_{j\in[1,k]} h_{i,j}\mathbf{w}_j\right) \\
&= \bigvee_{j\in[1,k]} \delta_{SE}(h_{i,j}\mathbf{w}_j) \\
&= \bigvee_{j\in[1,k]} h_{i,j}\delta_{SE}(\mathbf{w}_j) \text{ as } h_{i,j} \geq 0, \forall (i,j) \in [1,M]\times[1,k] \\
&\approx \sum_{j=1}^{k} h_{i,j}\delta_{SE}(\mathbf{w}_j) \\
&= D_{SE}(\mathbf{x}^{(i)})
\end{aligned}$$

With a similar results for the max-approximation to openings, as well as for the max-approximation to erosions and closings by duality.

The framework is shown to provide empirically encouraging results by the authors when experimenting on data sets of gray-scale images.

1.5 Interests of Deep Learning

Even though NMF have the very nice properties that will be listed in Section 2, it also has one major drawback. Indeed, the dictionary of the representation is learned for a specific data set of images, and the algorithm provides no way to represent, using the same atom images, images that were not included in the training set on which the NMF algorithm was run. Unlike in PCA decomposition, where the orthogonality of the components is enforced, and therefore a projection on the learned space is tractable, there is no closed form giving the weights $\mathbf{h}^{(M+1)}$ of the linear combination of the atom images $\mathbf{w}_1, ..., \mathbf{w}_k$ approximating a new data point $x^{(M+1)}$ with the proper sparsity constraint, when the atom images $\mathbf{W}$ have been learned on the images $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ...\mathbf{x}^{(M)}$. If we could find a neural network performing the same king of sparse and non-negative part-based representation, then the framework proposed in Section 1.4 using this decomposition would enable to compute max-approximation to morphological operators on new similar but previously unseen images, without the need to re-train the model and alter the atom images. Applying the morphological operator to a limited number of k fixed images would hence theoretically enable to compute the max-approximation of it on an unlimited number of images showing similar content (as variance of the content showed by the images increases, so would do the size of the latent representation space).

Moreover, deep neural networks are believed to have the ability to capture relationships much more complex than linear ones, the universal approximator theorem guaranteeing that a feed-forward neural network with at least one hidden layer can represent an approximation of any function (within a broad class) to an arbitrary degree of accuracy, provided that it has enough hidden units. The last few years indeed have seen significant interest in deep learning algorithms that learn layered, hierarchical representations of high-dimensional data. Much of this work appears to have been motivated by the hierarchical organization of the human's brain visual cortex, and indeed authors frequently compare their algorithms' output to the oriented simple cell receptive fields found in visual area V1 or V2. Indeed, some of these models are often viewed as first attempts to elucidate what learning algorithm (if any) the cortex may be using to model natural image statistics [16]. Given the nature of the decomposition we want to learn, it seems that auto-encoders are the most intuitive architectures answering to problem at stakes, as we will see in Section 3.

Throughout my research work, I encountered several attempts at applying "deep" methods with sparsity and non-Negativity constraints to perform part-based decompositions, similar to the very efficient one presented by the NMF algorithms. However none of them considered the max-approximation to morphological operators framework as an application of the learned part-based representations, and usually focused on reconstruction quality and/or classification accuracy. This works are namely:

- Lemme *et al.* (2011) [17], who presented a shallow auto-encoder architecture, with tied weights between the encoder and the decoder (that is the decoder weight matrix being the transpose of the encoder one) yielding higher sparsity and lower reconstruction errors than related algorithms based on matrix factorization. The authors qualified their algorithm as *online* as it generalizes to new inputs both accurately and without costly computations, which is fundamentally different from the classical matrix factorization approaches, that they relate to as *offline* algorithms.

- Hosseini-Asl *et al.* (2016) [12], that introduces a deep learning autoencoder network, trained by a non-negativity and sparsity constrained algorithm that appears to learn features which show part-based representation of data. The authors assess the performance of their network using classification and reconstruction error.

- Guo and Zang (2017) [9] which introduces a deep Non-Negative Factorization framework with a specifically designed optimization algorithm, not really using tools from the Deep Learning paradigm such as back-propagation.

- Ayinde and Zurada (2018) [2], which is an improvement proposal to the algorithms proposed in [12] to train an understandable classifier initialized using auto-encoders with non-negativity and sparsity constraints.

1.6 Evaluation and Data

Many different kinds of image families could be used in the context of part-based representation: binary shapes, database of registered gray images, patches of large images, time series, Hyper-spectral images, etc. While, we were at first planning to use multi-modal PET/MRI images of the brain, we finally opted for the use of the fashion MNIST data set for the sake of interpretability and simplicity. Fashion-MNIST ([23]) is a dataset of Zalando's article (clothes) images consisting of a balanced training set of 60,000 examples and a balanced test set of 10,000 examples. Just like the regular MNIST data set of handwritten digits, the images are 28x28 grayscale images, each associated with a label from 10 classes. It is intened to serve as a drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. The latter is indeed considered by the authors of [23] as too easy (as many models easily achieve 97%), overused (even suspecting an over-fitting of the machine learning community to this data set) and not suitable to modern day computer vision tasks. For our concerns, we also chose Fashion-MNIST as it represented shapes that can easily be interpreted as union of easily identifiable parts (such as sleeves in shirts), unlike medical images.

I implemented my models and experiments in Python 3.6, with the Keras v.2.2.0 deep learning framework on top of Tensorflow 1.9.

In order to evaluate and compare my various approaches to part, the following metrics will be used:

- The reconstruction error, that is the Mean-Squared Error (L_2, eventually normalized by the number of pixels in the images) between the original input images and their approximation by the learned representation.

- The best Support-Vector Machine (SVM) classification accuracy obtained using the encoding of the test images. A SVM with a linear kernel is trained and evaluated using the encodings of the images obtained with the various methods. 30 values of the C penalty parameter, in a 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.