Deep Network Classification by Scattering and Homotopy Dictionary Learning

Anonymous authors
Paper under double-blind review

Abstract

We introduce a sparse scattering deep convolutional neural network, which provides a simple model to analyze properties of deep representation learning for classification. Learning a single dictionary matrix with a classifier yields a higher classification accuracy than AlexNet over the ImageNet ILSVRC2012 dataset. The network first applies a scattering transform which linearizes variabilities due to geometric transformations such as translations and small deformations. A sparse l^1 dictionary coding reduces intra-class variability while preserving class separation through projections over unions of linear spaces. It is implemented in a deep convolutional network with a homotopy algorithm having an exponential convergence. A convergence proof is given in a general framework including ALISTA. Classification results are analyzed over ImageNet.

1 Introduction

Deep convolutional networks have spectacular applications to classification and regression (LeCun et al., 2015), but they are a black box which are hard to analyze mathematically because of their architecture complexity. We introduce a simplified convolutional neural network illustrated in Figure 1, whose learning can be reduced to a single dictionary matrix and a classifier. Despite its simplicity, it applies to complex image classification and reaches a higher accuracy than AlexNet (Krizhevsky et al., 2012) over ImageNet ILSVRC2012. It is a cascade of well understood mathematical operators, and thus provides a simplified mathematical framework to analyze classification performances.
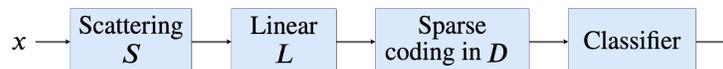
Figure 1: A sparse scattering network is composed of a scattering transform S followed by an optional linear operator L which reduces its dimensionality. A sparse code approximation of scattering coefficients is computed in a dictionary D. The dictionary D and the classifier are jointly learned by minimizing the classification loss with a stochastic gradient descent.

Intra-class variabilities due to geometric image transformations such as translations or small deformations are linearized by a scattering transform (Bruna & Mallat, 2013) which is invertible. Scattering transforms include no learning. They are effective representations to classify relatively simple images such as digits in MNIST, textures (Bruna & Mallat, 2013) or small CIFAR images (Oyallon & Mallat, 2014). Learning deep convolutional networks however gives a much higher accuracy over complex databases such as ImageNet. A fundamental issue is to understand the source of this improvement. This paper shows that it can be captured by a sparse l^1 code in a dictionary D optimized by supervised learning. It is implemented with a deep convolutional network architecture. The sparse code eliminates non-informative image components and

projects each class in unions of linear spaces. The classification accuracy is considerably improved and goes beyond AlexNet over ImageNet 2012.

Dictionary learning for classification was introduced in Mairal et al. (2009) and implemented with deep convolutional neural network architectures by several authors (Sulam et al., 2018; Mahdizadehaghdam et al., 2019; Sun et al., 2018). These algorithms have been applied to simpler image classification problems such as MNIST or CIFAR but no results were published on large datasets such as ImageNet on which they do not seem to scale. This is due to their complexity and the need to cascade several sparse codes, which leads to complex structures. We show that a single dictionary learning is sufficient if applied to scattering coefficients as opposed to raw data. A major issue is to compute the sparse code with a small network. We introduce a new architecture based on homotopy continuation, which leads to exponential convergence. It is thus implemented in a small convolutional network. The ALISTA (Liu et al., 2019) sparse code is incorporated in this framework. The main contributions of the paper are summarized below:

- A Sparse Scattering network architecture, illustrated in Figure 1, where the classification is performed over a sparse code in a learned dictionary of scattering coefficients. It outperforms AlexNet over ImageNet 2012.
- A new dictionary learning algorithm with homotopy sparse coding, optimized by gradient descent in a deep convolutional network.
- A proof of exponential convergence of ALISTA (Liu et al., 2019) in presence of noise.

We explain the implementation and mathematical properties of each element of the sparse scattering network. Section 2 briefly reviews multiscale scattering transforms. Section 3 introduces homotopy dictionary learning for classification, with a proof of exponential convergence under appropriate assumptions. Section 4 analyzes image classification results of sparse scattering networks on ImageNet 2012.

2 SCATTERING TRANSFORM

A scattering transform is a cascade of wavelet transforms and ReLU or modulus non-linearities. It can be interpreted as a deep convolutional network with predefined wavelet filters (Mallat, 2016). For images, wavelet filters are calculated from a mother complex wavelet ψ whose average is zero. It is rotated by $r_{-\theta}$, dilated by 2^j and its phase is shifted by α:

$$\psi_{j,\theta}(u) = 2^{-2j}\psi(2^{-j}r_{-\theta}u) \ \text{ and } \ \psi_{j,\theta,\alpha} = \text{Real}(e^{-i\alpha}\,\psi_{j,\theta}(u)).$$

We choose a Morlet wavelet as in Bruna & Mallat (2013) to produce a sparse set of non-negligible wavelet coefficients. A ReLU is written $\rho(a) = \max(a, 0)$.

Scattering coefficients of order $m = 1$ are computed by averaging rectified wavelet coefficients with a subsampling stride of 2^J:

$$Sx(u, k, \alpha) = \rho(x \star \psi_{j,\theta,\alpha}) \star \phi_J(2^J u) \ \text{ with } \ k = (j, \theta),$$

where ϕ_J is a Gaussian dilated by 2^J (Bruna & Mallat, 2013).

The averaging by ϕ_J eliminates the variations of $\rho(x \star \psi_{j,\theta,\alpha})$ at scales smaller than 2^J. This information is recovered by computing their variations at all scales $2^{j'} < 2^J$, with a second wavelet transform. Scattering coefficients of order two are:

$$Sx(u, k, k', \alpha, \alpha') = \rho(\rho(x \star \psi_{j,\theta,\alpha}) \star \psi_{j',\theta',\alpha'}) \star \phi_J(2^J u) \ \text{ with } \ k, k' = (j, \theta), (j', \theta').$$

To reduce the dimension of scattering vectors, we define phase invariant second order scattering coefficients with a complex modulus instead of a phase sensitive ReLU:

$$Sx(u, k, k') = ||x \star \psi_{j,\theta}| \star \psi_{j',\theta'}| \star \phi_J(2^J u) \text{ for } j' > j.$$

The scattering representation includes order 1 coefficients and order 2 phase invariant coefficients. In this paper, we choose $J = 4$ and hence 4 scales $1 \leq j \leq J$, 8 angles θ and 4 phases α on $[0, 2\pi]$. Scattering coefficients are computed with the software package Kymatio (Andreux et al., 2018). They preserve the image information and x can be recovered from Sx (Oyallon et al., 2019). For computational efficiency, the dimension of scattering vectors can be reduced by a factor 6 with a linear operator L which preserves the ability to recover a close approximation of x from $LS(x)$. The dimension reduction operator L of Figure 1 is computed by preserving the principal directions of a PCA calculated on the training image databasis, or is optimized by gradient descent together with the other network parameters.

The scattering transform is Lipschitz continuous to translations and deformations (Mallat, 2012). Intra-class variablities due to translations and deformations smaller than 2^J are linearized. Good classification accuracies are obtained with a linear classifier over scattering coefficients in image databases where intra-class variabilities are dominated by translations and deformations. This is the case for digits in MNIST or texture images (Bruna & Mallat, 2013). However it does not take into account variabilities of pattern structures and clutter which dominate complex image databases. To remove this clutter while preserving class separation requires some form of supervised learning as in deep convolutional networks. When applied to raw image data, dictionary learning often computes wavelet-like filters as in the first layer of deep neural networks (Krizhevsky et al., 2012). This is not sufficient to obtain high classification accuracy over complex image databases. The sparse scattering network of Figure 1 computes a sparse code of scattering representation $\beta = LS(x)$, in a dictionary D optimized by minimizing the classification loss. For this purpose, the next section introduces a homotopy dictionary learning algorithm, implemented in a small convolutional network.

3 HOMOTOPY DICTIONARY LEARNING FOR CLASSIFICATION

Task-driven dictionary learning for classification with sparse coding was proposed in Mairal et al. (2011). We introduce a small convolutional network architecture to implement a sparse l^1 code and learn the dictionary with a homotopy continuation on thresholds. ALISTA (Liu et al., 2019) is also shown to be a homotopy sparse coding whose exponential convergence is proved under more general conditions. Next section reviews dictionary learning for classification. Homotopy sparse coding algorithms are studied in Section 3.2.

3.1 DICTIONARY LEARNING

Unless specified, all norms are Euclidean norms. A sparse code approximates a vector β with a linear combination of a minimum number of columns D_m of a dictionary matrix D, which are normalized $\|D_m\| = 1$. A sparse code is a vector α^0 of minimum support which has a bounded error $\|D\alpha^0 - \beta\| \leq \sigma$. Such sparse codes have been used to optimize signal compression and to remove noise, to solve inverse problems in compressive sensing (Candes et al., 2006), and for classification (Mairal et al., 2011).

Minimizing the support of a code α amounts to minimizing its l^0 "norm" which is not convex. This non-convex optimization is convexified by replacing the l^0 norm by an l^1 norm $\|\alpha\|_1 = \sum_m |\alpha(m)|$. It is solved by minimizing a convex Lagrangian with a multiplier λ_* which depends on the error bound $\|D\alpha - \beta\| \leq \sigma$:

$$\alpha^1(\beta) = \arg\min_\alpha \frac{1}{2}\|D\alpha - \beta\|^2 + \lambda_* \|\alpha\|_1 . \tag{1}$$

The sparse code $\alpha^1(\beta, D, \lambda_*)$ also depends upon the dictionary D and λ_*, we omit these two last variables in the equation above for readability. One can prove (Donoho & Elad, 2006) that α^1 has the same support as the minimum support sparse code α^0 if the support size s and the dictionary coherence satisfy:

$$s\,\mu(D) < 1/2 \text{ where } \mu(D) = \max_{m \neq m'} |D_m^t D_{m'}| . \tag{2}$$

Sparse approximation versus sparse code   Sparse coding was first introduced for denoising (Donoho & Elad, 2006). The sparse approximation $D\alpha^1(\beta)$ is a non-linear filtering which preserves the "signal" components of β represented by few large amplitude coefficients. It eliminates the "noise" corresponding to incoherent components of β whose correlations with all dictionary vectors D_m are below λ_*. It can also be interpreted as a projection in a union of linear spaces, each of which corresponding to a sparse code support.

For classification, we need to reduce intra-class variabilities and preserve or increase class separability. Intra-class variabilites may be interpreted as "noise" for the classification whereas image transformations from one class to another correspond to the "signal" we want to preserve. By defining sparse representations of training vectors $\beta_i = LS(x_i)$ with different supports for different classes, it projects each class in different unions of linear spaces, which reduces intra-class variabilites while preserving separation. The dictionary learning optimizes the choice of D to obtain sparse codes with discriminative supports.

The classification is usually performed from the sparse code $\alpha^1(\beta)$. We will see that a classification applied on the reconstructed sparse approximation $D\alpha^1(\beta)$ has nearly the same accuracy. Indeed, the linear operator D can preserve separated linear spaces.

Dictionary learning by gradient descent   Given a set of inputs and labels $\{x_i, y_i\}$, task-driven dictionary learning minimizes a classification loss $\ell(\alpha^1(x_i, D, \lambda_*), y_i, W)$ that takes as input the sparse code $\alpha^1(x_i, D, \lambda_*)$ of the input x_i, the label y_i and the classification parameters W. Thus, the loss depends upon the dictionary D, the Lagrange multiplier λ_* which adjusts the sparsity level, and the classification parameters W. All these parameters can be jointly optimized by stochastic gradient descent to minimize the loss. This requires to compute the sparse code $\alpha^1(x_i, D, \lambda_*)$ and its derivatives w.r.t D and λ_*, which can be done by implementing the sparse coding in a deep convolutional network where the sparse code α^1 is computed in the forward pass and the derivatives of α^1 w.r.t D and λ_* are computed in the backward pass. For this purpose, next section introduces a homotopy iterated soft thresholding network architecture.

3.2   HOMOTOPY ITERATED SOFT THRESHOLDING NETWORK

This section introduces an efficient convolutional network architecture to compute sparse codes and learn dictionaries. Iterative Soft-Thresholding Algorithms (ISTA) (Daubechies et al., 2004), and FISTA (Beck & Teboulle, 2009) can be implemented with deep neural networks but they require many layers because of their slow convergence. LISTA algorithm (Gregor & LeCun, 2010) and its more recent version ALISTA (Liu et al., 2019) accelerate this convergence by introducing an auxiliary matrix which is adapted to the statistics of the input and to the properties of the dictionary. For ALISTA, it leads to exponential convergence under appropriate hypotheses. However, we shall see that this auxiliary matrix prevents from using this approach to learn a dictionary which minimizes a classification loss with a sparse l^1 code. We introduce a dictionary learning based on a homotopy Iterated Soft Thresholding Continuation (Jiao et al., 2017), which has the same exponential convergence without an auxiliary matrix. We shall see that ALISTA can also be considered as a homotopy continuation algorithm. We give a proof of exponential convergence for non-zero Lagrange multipliers λ_* in this general framework.

Iterated Soft Thresholding   ISTA alternates a gradient step on the quadratic term of the l^1 Lagrangian (1) and a soft-thresholding $T_\lambda(a) = \mathrm{sign}(a) \max(|a| - \lambda, 0)$:

$$\alpha_{n+1} = T_{\epsilon\lambda_*}(\alpha_n + \epsilon D^t(\beta - D\alpha_n)) \ \text{ with } \ \epsilon < \frac{1}{\|D^t D\|_{2,2}}, \tag{3}$$

where $\|.\|_{2,2}$ is the spectral norm and $\alpha_0 = 0$. The first iteration computes a non-sparse code $\alpha_1 = T_{\epsilon\lambda_*}(\epsilon D^t\beta) = \epsilon T_{\lambda_*}(D^t\beta)$ which is progressively sparsified through iterated thresholdings. After n iterations, the sparse code α_n has an error in $O(n^{-1})$. FISTA (Beck & Teboulle, 2009) accelerates the error decay to $O(n^{-2})$, which remains slow. Each iteration of ISTA and FISTA is computed with linear operators and a soft
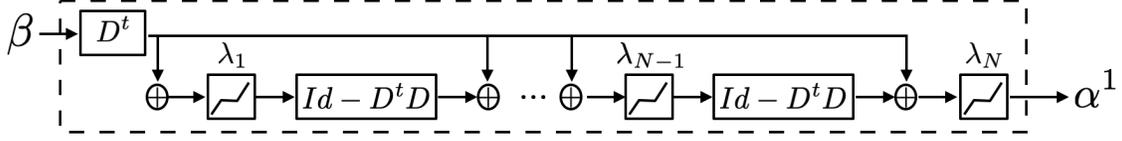
Figure 2: A homotopy iterated soft thresholding network computes an l^1 sparse code with exponentially decreasing thresholds $\lambda_n = \lambda_{\max}\gamma^{-n}$ with $\gamma = (\lambda_{\max}/\lambda_*)^{1/N}$.

thresholding and can thus be implemented with one layer in a deep network (Papyan et al., 2017). However, the total number N of layers must be large to achieve a small error, and it requires to compute spectral norms during training, which is slow.

Homotopy Iterated Thresholding and ALISTA Homotopy continuation algorithms introduced in Osborne et al. (2000), minimize the l^1 Lagrangian (1) by progressively decreasing the Lagrange multiplier. This optimization path is opposite to ISTA and FISTA since it goes from a very sparse initial solution towards a less sparse but optimal one, similarly to matching pursuit algorithms (Davis et al., 1997; Donoho & Tsaig, 2008). Homotopy algorithms are particularly efficient if the final Lagrange multiplier λ_* is large so that the optimal solution is very sparse. We shall see that it is the case for classification.

The homotopy Iterative Soft-Thresholding Continuation (ISTC) of Jiao, Jin and Lu (Jiao et al., 2017) algorithm adjusts the decay rate of an exponentially decreasing sequence of Lagrange multipliers λ_n for $n \leq N$:

$$\alpha_{n+1} = T_{\lambda_{n+1}}(\alpha_n + D^t(\beta - D\alpha_n)) \ \text{ with } \ \lambda_n = \lambda_{\max}\,\gamma^{-n} \ \text{ and } \ \gamma = \left(\frac{\lambda_{\max}}{\lambda_*}\right)^{1/N}. \tag{4}$$

After N iterations, they prove that α_N has the same support as the optimal sparse code α^0, if $\lambda_{\max} \geq \|D^t\beta\|_\infty$, if the dictionary coherence condition (2) is satisfied, and if γ is sufficiently close to 1. Figure 2 illustrates the implementation of this sparse coding algorithm in a deep network of depth N, with side connections. For image classification we use a convolutional translation invariant dictionary, which defines a deep convolutional network. This convolutional network is used to compute the sparse code of scattering coefficients $LS(x)$ in Figure 1.

ALISTA can be considered as a generalization of the homotopy ISTC algorithm, which replaces D^t by an auxiliary matrix W^t. We shall also study whether this flexibility can improve results. Each column W_m of W is normalized by $|W_m^t D_m| = 1$. The iteration (4) is thus rewritten

$$\alpha_{n+1} = T_{\lambda_{n+1}}(\alpha_n + W^t(\beta - D\alpha_n)) \ \text{ with } \ \lambda_n = \lambda_{\max}\,\gamma^{-n} \ \text{ and } \ \gamma = \left(\frac{\lambda_{\max}}{\lambda_*}\right)^{1/N}. \tag{5}$$

The following theorem extends the convergence result of homotopy ISTC algorithm, by replacing the coherence of D by the mutual coherence of W and D

$$\widetilde{\mu} = \max_{m \neq m'} |W_{m'}^t D_m|.$$

This theorem also extends the ALISTA exponential convergence result in the general setting where the sparse code introduces a reconstruction error, which may be interpreted as a noise removal. We will see that this error can be large for image classification applications because it corresponds to non-informative clutter removal.

Theorem 3.1. *Let α^0 be the l^0 sparse code of β with error $\|\beta - D\alpha^0\| \leq \sigma$. If its support s satisfies*

$$s\,\widetilde{\mu} < 1/2 \tag{6}$$

then soft-thresholding iterations (5) with thresholds

$$\lambda_n = \lambda_{\max} \, \gamma^{-n} \geq \lambda_* = \frac{\|W^t(\beta - D\alpha^0)\|_\infty}{1 - 2\gamma\widetilde{\mu}s} \; \text{ with } 1 < \gamma < (2\widetilde{\mu}s)^{-1} \text{ and } \lambda_{\max} \geq \|W^t\beta\|_\infty \quad (7)$$

define a sparse code α_n, whose support is included in the support of α^0 and

$$\|\alpha_n - \alpha^0\|_\infty \leq 2\,\lambda_{\max}\,\gamma^{-n} \,. \tag{8}$$

The proof is in Appendix A of the supplementary material. It adapts the convergence proof of ISTC to the more general ALISTA framework. When $W = D$, we recover the convergence result of the homotopy ISTC, and when $\lambda_* = 0$ we recover the ALISTA exponential convergence result. However, one should not get too impressed by this exponential convergence rate because the condition $s\widetilde{\mu} < 1/2$ only applies to very sparse codes in highly incoherent dictionaries. ALISTA optimizes W in order to minimize the mutual coherence $\widetilde{\mu}$, but it is usually not possible to reach $s\widetilde{\mu} < 1/2$. It thus restricts the set of possible signals and dictionaries, as opposed to ISTA and FISTA algorithms whose convergence is guaranteed for any signal and dictionary. However, the condition $s\widetilde{\mu} < 1/2$ is based on a brutal upper bound calculation in the proof, and it is not necessary for convergence. Next section shows that for image classification over ImageNet, by setting $W = D$ we learn a dictionary where the homotopy ISTC algorithm converges exponentially although the theorem hypothesis is not satisfied. By learning simultaneously W and D, we shall see that we can reduce the classification loss but the resulting algorithm does not converge to a sparse $\mathbf{l}^1$ code anymore.

4 Image Classification

The goal of this work is to construct a deep neural network model which is sufficiently simple to be interpreted mathematically, while reaching a level of accuracy of more complex deep convolutional networks on complex classification problems. This is why we concentrate on ImageNet as opposed to MNIST or CIFAR. Next section compares its performance to state of the art deep networks, and analyzes the influence of different architecture components. Section 4.2 studies the exponential convergence of the homotopy ISTC sparse coding network in comparison with ISTA, FISTA and a flexible ALISTA.

4.1 Image Classification on ImageNet

We show that a sparse dictionary learning on scattering coefficients considerably improves the classification performance on $S(x)$ and can outperform AlexNet accuracy.

ImageNet ILSVRC2012 is a challenging color image dataset of 1.2 million training images and 50,000 validation images, divided into 1000 classes. Prior to convolutional networks, SIFT representations combined with Fisher vector encoding reached a Top 5 classification accuracy of 74.3% with multiple model averaging (Sánchez & Perronnin, 2011). In their PyTorch implementation, the Top 5 accuracy of AlexNet and ResNet-152 is 79.1% and 94.1% respectively[1] .

The scattering transform $S(x)$ at a scale $2^J = 16$ of an ImageNet color image is a spatial array of 14×14 of 1539 channels. Applying to $S(x)$ an MLP classifier with 2 hidden layers of size 4096, ReLU and dropout like in AlexNet gives a 60.7% Top 5 accuracy. Applying to $S(x)$ a 3-layer SLE network of 1x1 convolutions with ReLU with the same MLP reaches AlexNet performance (Oyallon et al., 2017). However, there is no mathematical understanding of the operations performed by these three layers, and the origin of the improvements.

The sparse scattering architecture is described in Figure 3. The convolutional operator L is applied on a standardized scattering transform and reduces the number of scattering channels from 1539 to 256. The

---

[1] Accuracies from https://pytorch.org/docs/master/torchvision/models.html
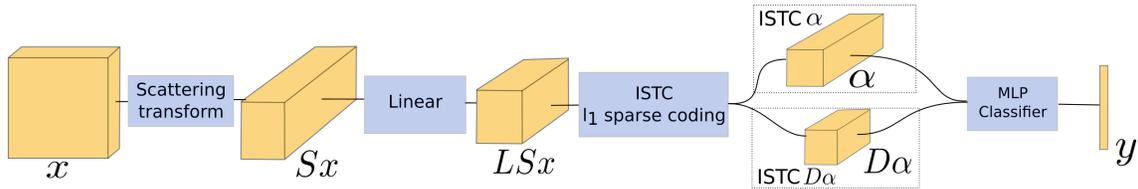
Figure 3: Two variants of the image classification architecture: ISTC α inputs the sparse code α into the classifier, ISTC $D\alpha$ inputs the reconstruction $D\alpha$ into the classifier.

sparse code is calculated with a 1×1 convolutional dictionary D having 2048 vectors. It takes as input an array $LS(x)$ of $14 \times 14 \times 256$ which has been normalized and outputs a code α^1 of size $14 \times 14 \times 2048$ or a sparse approximation $D\alpha^1$ of size $14 \times 14 \times 256$. Either is provided as input to the MLP classifier. The ISTC network illustrated in Figure 2 has $N = 12$ layers with softshrink non-linearities and no batch normalization. Before the classifier, there is a batch normalization and a 5×5 average pooling. The MLP classifier has 2 hidden layers of size 4096, ReLU and dropout rate of 0.3. The supervised learning jointly optimizes L, the dictionary D with the Lagrange multiplier λ_* and the MLP classifier. It is done with a stochastic gradient descent during 120 epochs using an initial learning rate of 0.01 with a decay of 0.1 at epochs 50 and 100. With a sparse code in input of the MLP, it has a Top 5 accuracy of 80.9%, outperforming AlexNet. If we replace the ISTC network by an ALISTA network, the accuracy improves to 83.7%. However, next section shows that contrarily to ISTC, an ALISTA network optimized for classification does not compute a sparse l^1 code and is therefore not mathematically interpretable. In the following we thus concentrate on the homotopy ISTC network.

Table 1: Top 1 and Top 5 accuracy on ImageNet for Fisher Vectors (Perronnin & Larlus, 2015), AlexNet (Krizhevsky et al., 2012), ResNet 152 (He et al., 2016), Scattering with SLE (Oyallon et al., 2019), Scattering alone, with a sparse code α^1 or with a classification on $D\alpha^1$.

	Fisher Vectors	AlexNet	ResNet 152	Scat. + SLE	Scat. alone	Scat.+ ISTC α	Scat.+ ISTC $D\alpha$	Scat.+ ALISTA α
Top1	55.6	56.5	78.3	57.0	37.5	59.0	54.8	62.6
Top5	78.4	79.1	94.1	79.6	60.7	80.9	77.8	83.7

The dimension reduction operator L has a marginal effect in terms of performance. If we eliminate it or if we replace it by an unsupervised PCA dimension reduction, the performance drops by less than 2%, whereas the accuracy drops by 20% if we eliminate the sparse coding. The considerable improvement brought by the sparse code is further amplified if the MLP classifier is replaced by a linear classifier. A linear classifier on a scattering vector has a (Top 1, Top 5) accuracy of $(23.4\%, 41.8\%)$. With an ISTC sparse code in a learned dictionary the accuracy jumps to $(51.5\%, 73.4\%)$ and hence improves by more than 30%.

If the MLP classification is applied to the sparse approximation $D\alpha^1$ as opposed to the sparse code α^1 then the accuracy drops only by 3%. The sparse approximation $D\alpha^1$ of $LS(x)$ has a small dimension $14 \times 14 \times 256$ similar to AlexNet last convolutional layer output and is not sparse. This indicates that it is not the individual sparse outputs of the sparse code α^1 which are important but the linear space defined by their support, which are mapped to other linear spaces by D.

The optimization learns a large factor λ_* which yields a large approximation error $\|LS(x) - D\alpha^1\|/\|LS(x)\| \approx 0.5$. The resulting code α^1 is very sparse with about 3% non-zero coefficients. The sparse approximation $D\alpha^1$ thus eliminates nearly half of the energy of $LS(x)$ which can be interpreted as
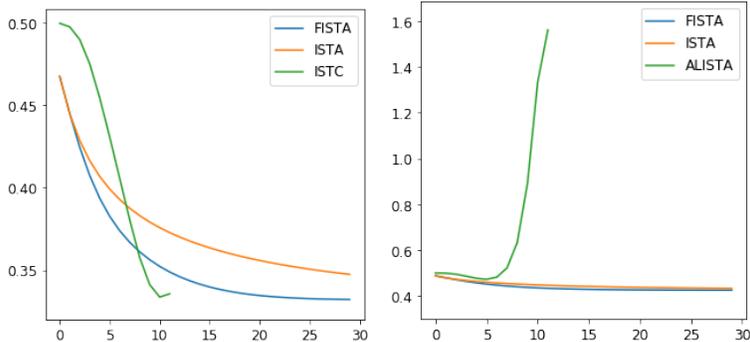
Figure 4: Value of $\mathcal{L}(\alpha_n) = \frac{1}{2}\|D\alpha_n - \beta\|^2 + \lambda_* \|\alpha_n\|_1$ versus the number of iterations n, for ISTC, ISTA and FISTA on the left, and for ALISTA, ISTA and FISTA on the right.

non-informative "clutter" removal. The sparse code α^1 is a projection of $LS(x)$ over a linear space defined by the support of α^1. If a column D_m is interpreted as a "scattering space feature" then this linear space is a conjunction of a particular set of such features. The high classification accuracy indicates that different linear spaces correspond mostly to different classes. These linear spaces are mapped by D into lower dimensional linear spaces which remain separated. It thus indicates that D is optimized to preserves discriminative directions which transform a vector of one class into a vector of another one.

4.2 Convergence of Homotopy Algorithms

To guarantee that the network is mathematically interpretable we verify numerically that the homotopy ISTC algorithm computes an accurate approximation of the optimal l^1 sparse code in (1), with a small number of iterations (typically 12).

The Theorem 3.1 guarantees an exponential convergence if $s\,\mu(D) < 1/2$. In our classification setting, the theorem hypothesis is clearly not satisfied : $s\mu(D) \approx 60$, which is well above $1/2$. However, this condition is not necessary and based on a relatively crude upper bound.

Figure 4 left shows numerically that ISTC algorithm minimizes the l^1 Lagrangian $\mathcal{L}(\alpha) = \frac{1}{2}\|D\alpha - \beta\|^2 + \lambda_* \|\alpha\|_1$, with an exponential convergence which is faster than ISTA and FISTA over the dictionary that it learns. On the contrary, Figure 4 right shows that ALISTA does not minimize the l^1 Lagrangian at all. This comes from the fact that contrarily to standard ALISTA (Liu et al., 2019), we do not impose that the auxiliary matrix W has a minimum joint coherence with the dictionary D. It would require too much computation and the matrix W is rather optimized to minimize the classification loss. This is why it improves the classification accuracy but does not compute a sparse l^1 code.

To further compare the convergence speed of ISTC versus ISTA and FISTA, we compute the relative mean square error $\mathrm{MSE}(x, y) = \|x - y\|^2/\|x\|^2$ between the optimal sparse code α^1 and the sparse code output of 12 iterations of each of these three algorithms. The MSE is 0.02 for ISTC, 0.25 for FISTA and 0.46 for ISTA, which shows that ISTC reduces the error by a factor 10 compared to ISTA and FISTA after 12 iterations.

5 Conclusion

The first goal of this work is to define a deep neural network having a good accuracy for complex image classification and which can be analyzed mathematically. This sparse scattering network learns the repre-

sentation by optimizing a sparse code computed with a dictionary learned over scattering coefficients. The dictionary learning is implemented with a new homotopy ISTC network having an exponential convergence. The sparse dictionary learning improves accuracy by more than 20% over a scattering representation alone, and has a higher accuracy than AlexNet. The dictionary seems to be optimized in order to build separated sparse codes for each class, which belong to unions of linear spaces. Because the network operators are mathematically well specified, the analysis of its properties is simpler than for standard deep convolutional networks. However, more work is needed to understand the dictionary optimization and how it relates to image and class properties.

REFERENCES

M. Andreux, T. Angles, G. Exarchakis, R. Leonarduzzi, G. Rochette, L. Thiry, J. Zarka, S. Mallat, J. Andén, E. Belilovsky, J. Bruna, V. Lostanlen, M. J. Hirn, E. Oyallon, S. Zhang, C. E. Cella, and M. Eickenberg. Kymatio: Scattering transforms in python. *CoRR*, 2018. URL `http://arxiv.org/abs/1812.11214`.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1872–1886, 2013.

E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constr. Approx.*, 13(1):57–98, 1997.

D.L. Donoho and M. Elad. On the stability of the basis pursuit in the presence of noise. *Signal Processing*, 86(3):511–532, 2006.

D.L. Donoho and Y. Tsaig. Fast solution of l_1-norm minimization problems when the solution may be sparse. *IEEE Trans. Information Theory*, 54(11):4789–4812, 2008.

K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *ICML*, pp. 399–406, 2010.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Y. Jiao, B. Jin, and X. Lu. Iterative soft/hard thresholding with homotopy continuation for sparse recovery. *IEEE Signal Processing Letters*, 24(6):784–788, June 2017.

A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1106–1114, 2012.

Y. LeCun, Y. Bengio, and G.E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

J. Liu, X. Chen, Z. Wang, and W. Yin. ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations*, 2019.

S. Mahdizadehaghdam, A. Panahi, H. Krim, and L. Dai. Deep dictionary learning: A parametric network approach. *IEEE Transactions on Image Processing*, 28(10):4790–4802, Oct 2019.

J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pp. 1033–1040, 2009.

J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):791–804, 2011.

S. Mallat. Group invariant scattering. *Comm. Pure Appl. Math.*, 65(10):1331–1398, 2012.

S. Mallat. Understanding deep convolutional networks. *Phil. Trans. of Royal Society A*, 374(2065), 2016.

M.R. Osborne, B. Presnell, and B.A. Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389, 2000.

E. Oyallon and S. Mallat. Deep roto-translation scattering for object classification. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2865–2873, 2014.

E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5618–5627, 2017.

E. Oyallon, S. Zagoruyko, G. Huang, N. Komodakis, S. Lacoste-Julien, M. Blaschko, and E. Belilovsky. Scattering networks for hybrid representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2208–2221, Sep. 2019.

V. Papyan, Y. Romano, and M. Elad. Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research*, 18:83:1–83:52, 2017.

F. Perronnin and D. Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3743–3752, 2015.

J. Sulam, V. Papyan, Y. Romano, and M. Elad. Multilayer convolutional sparse modeling: Pursuit and dictionary learning. *IEEE Transactions on Signal Processing*, 66(15):4090–4104, 2018.

X. Sun, N. M. Nasrabadi, and T. D. Tran. Supervised deep sparse coding networks. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 346–350, 2018.

J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *CVPR*, pp. 1665–1672. IEEE Computer Society, 2011.

A APPENDIX

A.1 PROOF OF THEOREM 3.1

Let α^0 be the optimal l^0 sparse code. We denote by $\mathcal{S}(\alpha)$ the support of any α. We are going to prove by induction on n that for any $n \geq 0$ we have $\mathcal{S}(\alpha_n) \subset \mathcal{S}(\alpha^0)$ and $\|\alpha_n - \alpha^0\|_\infty \leq 2\lambda_n$ if $\lambda_n \geq \lambda_*$.

For $n = 0$, $\alpha_0 = 0$ so $\mathcal{S}(\alpha_0) = \emptyset$ is indeed included in the support of α^0 and $\|\alpha_0 - \alpha^0\|_\infty = \|\alpha^0\|_\infty$. To verify the induction hypothesis for $\lambda_0 = \lambda_{\max} \geq \lambda_*$, we shall prove that $\|\alpha^0\|_\infty \leq 2\lambda_{\max}$.

Let us write the error $w = \beta - D\alpha^0$. For all m

$$\alpha^0(m)W_m^t D_m = W_m^t \beta - W_m^t w - \sum_{m \neq m'} \alpha^0(m')\, W_m^t D_{m'}.$$

Since the support of α^0 is smaller than s, $W_m^t D_m = 1$ and $\widetilde{\mu} = \max_{m \neq m'} |W_m^t D_{m'}|$

$$|\alpha^0(m)| \leq |W_m^t \beta| + |W_m^t w| + s\,\widetilde{\mu}\,\|\alpha^0\|_\infty$$

so taking the max on m gives:

$$\|\alpha^0\|_\infty (1 - \widetilde{\mu}s) \leq \|W^t \beta\|_\infty + \|W^t w\|_\infty$$

But given the inequalities

$$
\begin{aligned}
\|W^t \beta\|_\infty &\leq& \lambda_{\max} \\
\|W^t w\|_\infty &\leq& \lambda_{\max}(1 - 2\gamma\widetilde{\mu}s) \\
\frac{(1 - \gamma\widetilde{\mu}s)}{(1 - \widetilde{\mu}s)} &\leq& 1 \quad \text{since } \gamma \geq 1 \text{ and } (1 - \widetilde{\mu}s) > 0
\end{aligned}
$$

we get

$$\|\alpha^0\|_\infty \leq 2\lambda_{\max} = 2\lambda_0$$

Let us now suppose that the property is valid for n and let us prove it for $n+1$. We denote by $D_{\mathcal{A}}$ the restriction of D to vectors indexed by $\mathcal{A}$. We begin by showing that $\mathcal{S}(\alpha_{n+1}) \subset \mathcal{S}(\alpha^0)$. For any $m \in \mathcal{S}(\alpha_{n+1})$, since $\beta = D\alpha^0 + w$ and $W_m^t D_m = 1$ we have

$$
\begin{aligned}
\alpha_{n+1}(m) &=& T_{\lambda_{n+1}}(\alpha_n(m) + W_m^t(\beta - D\alpha_n)) \\
&=& T_{\lambda_{n+1}}(\alpha^0(m) + W_m^t(D_{\mathcal{S}(\alpha^0) \cup \mathcal{S}(\alpha_n) - \{m\}}(\alpha^0 - \alpha_n)_{\mathcal{S}(\alpha^0) \cup \mathcal{S}(\alpha_n) - \{m\}} + w))
\end{aligned}
$$

For any m not in $\mathcal{S}(\alpha^0)$, let us prove that $\alpha_{n+1}(m) = 0$. The induction hypothesis assumes that $\mathcal{S}(\alpha_n) \subset \mathcal{S}(\alpha^0)$ and $\|\alpha^0 - \alpha_n\|_\infty \leq 2\lambda_n$ with $\lambda_n \geq \lambda_*$ so:

$$
\begin{aligned}
I &=& |\alpha^0(m) + W_m^t(D_{\mathcal{S}(\alpha^0) \cup \mathcal{S}(\alpha_n) - \{m\}}(\alpha^0 - \alpha_n)_{\mathcal{S}(\alpha^0) \cup \mathcal{S}(\alpha_n) - \{m\}} + w)| \\
&\leq& |W_m^t(D_{\mathcal{S}(\alpha^0)}(\alpha^0 - \alpha_n)_{\mathcal{S}(\alpha^0)})| + |W_m^t w| \quad \text{since } \mathcal{S}(\alpha_n) \subset \mathcal{S}(\alpha^0) \text{ and } \alpha^0(m) = 0 \text{ by assumption.} \\
&\leq& \widetilde{\mu}s\|\alpha^0 - \alpha_n\|_\infty + \|W^t w\|_\infty
\end{aligned}
$$

Since we assume that $\lambda_{n+1} \geq \lambda_*$, we have

$$\|W^t w\|_\infty \leq (1 - 2\gamma\widetilde{\mu}s)\lambda_{n+1}$$

and thus

$$I \leq \widetilde{\mu}s\|\alpha^0 - \alpha_n\|_\infty + \|W^t w\|_\infty \leq \widetilde{\mu}s 2\lambda_n + \lambda_{n+1}(1 - 2\gamma\widetilde{\mu}s) \leq \lambda_{n+1}$$

since $\lambda_n = \gamma\lambda_{n+1}$.

Because of the thresholding $T_{\lambda_{n+1}}$, it proves that $\alpha_{n+1}(m) = 0$ and hence that $\mathcal{S}(\alpha_{n+1}) \subset \mathcal{S}(\alpha^0)$.

Let us now evaluate $\|\alpha^0 - \alpha_{n+1}\|_\infty$. For any $(\alpha_1, \alpha_2, \lambda)$, a soft thresholding satisfies

$$|T_\lambda(\alpha_1 + \alpha_2) - \alpha_1| \leq \lambda + |\alpha_2|$$

so:

$$
\begin{aligned}
|\alpha_{n+1}(m) - \alpha^0(m)| &\leq& \lambda_{n+1} + |W_m^t(D_{\mathcal{S}(\alpha^0) \cup \mathcal{S}(\alpha_n) - \{m\}}(\alpha^0 - \alpha_n)_{\mathcal{S}(\alpha^0) \cup \mathcal{S}(\alpha_n) - \{m\}})| + |W_m^t w| \\
&\leq& \lambda_{n+1} + \widetilde{\mu}s\|\alpha^0 - \alpha_n\|_\infty + \|W^t w\|_\infty \\
&\leq& \lambda_{n+1} + \widetilde{\mu}s 2\lambda_n + \lambda_{n+1}(1 - 2\gamma\widetilde{\mu}s) = 2\lambda_{n+1}
\end{aligned}
$$

Taking a max over m proves the induction hypothesis.