



Article submitted to journal

Subject Areas:

analysis, pattern recognition,
statistics

Keywords:

Deep convolutional neural networks,
learning, wavelets

Author for correspondence:

Stéphane Mallat

e-mail: stephane.mallat@ens.fr

Understanding Deep Convolutional Networks

Stéphane Mallat

École Normale Supérieure, CNRS, PSL

45 rue d'Ulm, Paris, France

Deep convolutional networks provide state of the art classifications and regressions results over many high-dimensional problems. We review their architecture, which scatters data with a cascade of linear filter weights and non-linearities. A mathematical framework is introduced to analyze their properties. Computations of invariants involve multiscale contractions with wavelets, the linearization of hierarchical symmetries, and sparse separations. Applications are discussed.

1. Introduction

Supervised learning is a high-dimensional interpolation problem. We approximate a function $f(x)$ from q training samples $\{x^i, f(x^i)\}_{i \leq q}$, where x is a data vector of very high dimension d . This dimension is often larger than 10^6 , for images or other large size signals. Deep convolutional neural networks have recently obtained remarkable experimental results [21]. They give state of the art performances for image classification with thousands of complex classes [19], speech recognition [17], bio-medical applications [22], natural language understanding [30], and in many other domains. They are also studied as neuro-physiological models of vision [4].

Multilayer neural networks are computational learning architectures which propagate the input data across a sequence of linear operators and simple non-linearities. The properties of shallow networks, with one hidden layer, are well understood as decompositions in families of ridge functions [10]. However, these approaches do not extend to networks with more layers. Deep convolutional neural networks, introduced by Le Cun [20], are implemented with linear convolutions followed by non-linearities, over typically more than 5 layers. These complex programmable machines, defined by potentially billions of filter weights, bring us to a different mathematical world.

Many researchers have pointed out that deep convolution networks are computing progressively more powerful invariants as depth increases [4,21], but relations with networks weights and non-linearities are complex. This paper aims at clarifying important principles which govern the properties of such networks, although their architecture and weights may differ with applications. We show that computations of invariants involve multiscale contractions, the linearization of hierarchical symmetries, and sparse separations.

Statistical estimation can ultimately be reduced to a bias versus variance problem. In high dimension, x has a considerable number of parameters and a huge variance, which is a dimensionality curse. Sampling uniformly a volume of dimension d requires a number of samples which grows exponentially with d . In most applications, the number q of training samples rather grows linearly with d . It is possible to approximate $f(x)$ with so few samples, only if f has some strong regularity properties allowing to ultimately reduce the dimension of the estimation. Any learning algorithm, including deep convolutional networks, thus relies on an underlying assumption of regularity, which introduces a small estimation bias if the model is appropriate.

We need to reduce the variance of the input data $x \in \Omega$ without sacrificing the ability to approximate $f(x)$. This can be done by defining a new variable $\Phi(x)$ where Φ is a *contractive* operator which reduces the variance of x , while *separating* different values of f : $\Phi(x) \neq \Phi(x')$ if $f(x) \neq f(x')$. The statistical bias-variance trade-off becomes a separation-contraction trade-off, which needs to be adjusted in high dimension.

Linearization is a strategy used in machine learning to reduce the dimension with a linear projector. A linear projection of x separates the values of f if this function remains constant in the direction of a high-dimensional linear space. This is rarely the case, but one can try to find $\Phi(x)$ which linearizes high-dimensional domains where $f(x)$ remains constant. The dimension is then reduced by applying a linear projector on $\Phi(x)$. Finding such a Φ is the dream of kernel learning algorithms, explained in Section 2.

Deep neural networks are more conservative. They progressively reduce the variance while controlling bias. They contract the space and linearize transformations along which f remains nearly constant, to avoid introducing a bias. Such directions are defined by linear operators which belong to groups of local symmetries, introduced in Section 3. To understand the difficulty to linearize the action of high-dimensional groups of operators, we begin with the groups of translations and diffeomorphisms, which deform signals. They capture essential mathematical properties that are extended to general deep network symmetries, in Section 7.

To linearize diffeomorphisms and preserve separability, Section 4 shows that we must separate the variations of x at different scales, with a wavelet transform. This is implemented with multiscale filter convolutions, which are building blocks of deep convolution filtering. General deep network architectures are introduced in Section 5. They iterate on linear operators which filter and linearly combine different channels in each network layer, followed by contractive non-linearities.

To understand how non-linear contractions interact with linear operators, Section 6 begins with simpler networks which do not recombine channels in each layer. It defines a non-linear scattering transform, introduced in [24], where wavelets have a separation and linearization role. The resulting contraction, linearization and separability properties are reviewed. We shall see that sparsity is important for separation.

Section 7 extends these ideas to more general deep convolutional networks, while showing that the same principles are in action. Channel combinations provide the flexibility needed to extend translations to larger groups of local symmetries adapted to f . The network can be structured by factorizing groups of symmetries, in which case all linear operators are generalized convolutions. Computations are ultimately performed on filter weights, which are learned. Their relation with groups of symmetries is explained. Controlling bias also requires to preserve a separation margin across classification frontiers. It can be implemented by separating network fibers, which are progressively more invariant and specialized. This gives rise to invariant grandmother type

neurons as depth increases. The paper studies architectures as opposed to computational learning of network weights, which is an outstanding optimization issue [21].

Notations $\|z\|$ is a Euclidean norm if z is a vector in a Euclidean space. If z is a function in \mathbf{L}^2 then $\|z\|^2 = \int |z(u)|^2 du$. If $z = \{z_k\}_k$ is a sequence of vectors or functions then $\|z\|^2 = \sum_k \|z_k\|^2$.

2. Linearization, Projection and Separability

Supervised learning computes an approximation $\tilde{f}(x)$ of a function $f(x)$ from q training samples $\{x^i, f(x^i)\}_{i \leq q}$, for $x = (x(1), \dots, x(d)) \in \Omega$. The domain Ω is a high dimensional open subset of \mathbb{R}^d , not a low-dimensional manifold. In a regression problem, $f(x)$ takes its values in \mathbb{R} , whereas in classification its values are class indices.

Ideally, we would like to reduce the dimension of x by computing a low dimensional vector $\Phi(x)$ such that one can write $f(x) = f_0(\Phi(x))$. It is equivalent to impose that if $f(x) \neq f(x')$ then $\Phi(x) \neq \Phi(x')$. We then say that Φ separates f . For regression problems, to guarantee that f_0 is regular, we further impose that the separation is Lipschitz:

$$\exists \epsilon > 0 \quad \forall (x, x') \in \Omega^2, \quad \|\Phi(x) - \Phi(x')\| \geq \epsilon |f(x) - f(x')|. \quad (2.1)$$

It implies that f_0 is Lipschitz continuous: $|f_0(z) - f_0(z')| \leq \epsilon^{-1}|z - z'|$, for $(z, z') \in \Phi(\Omega)^2$. In a classification problem, $f(x) \neq f(x')$ means that x and x' are not in the same class. The Lipschitz separation condition (2.1) becomes a margin condition specifying a minimum distance across classes:

$$\exists \epsilon > 0 \quad \forall (x, x') \in \Omega^2, \quad \|\Phi(x) - \Phi(x')\| \geq \epsilon \quad \text{if } f(x) \neq f(x'). \quad (2.2)$$

We can try to find a linear projection of x in some space \mathbf{V} of lower dimension k , which separates f . It requires that $f(x) = f(x + z)$ for all $z \in \mathbf{V}^\perp$, where \mathbf{V}^\perp is the orthogonal complement of \mathbf{V} in \mathbb{R}^d , of dimension $d - k$. In most cases, the final dimension k can not be much smaller than d .

An alternative strategy is to linearize the variations of f with a first change of variable $\Phi(x) = \{\phi_k(x)\}_{k \leq d'}$ of dimension d' potentially much larger than the dimension d of x . We can then optimize a low-dimensional linear projection along directions where f is constant. We say that Φ separates f linearly if $f(x)$ is well approximated by a one-dimensional projection:

$$\tilde{f}(x) = \langle \Phi(x), w \rangle = \sum_{k=1}^{d'} w_k \phi_k(x). \quad (2.3)$$

The regression vector w is optimized by minimizing a loss on the training data, which needs to be regularized if $d' > q$, for example by an \mathbf{I}^p norm of w with a regularization constant λ :

$$\sum_{i=1}^q \text{loss}(f(x^i) - \tilde{f}(x^i)) + \lambda \sum_{k=1}^{d'} |w_k|^p. \quad (2.4)$$

Sparse regressions are obtained with $p \leq 1$, whereas $p = 2$ defines kernel regressions [16].

Classification problems are addressed similarly, by approximating the frontiers between classes. For example, a classification with Q classes can be reduced to $Q - 1$ "one versus all" binary classifications. Each binary classification is specified by an $f(x)$ equal to 1 or -1 in each class. We approximate $f(x)$ by $\tilde{f}(x) = \text{sign}(\langle \Phi(x), w \rangle)$, where w minimizes the training error (2.4).

3. Invariants, Symmetries and Diffeomorphisms

We now study strategies to compute a change of variables Φ which linearizes f . Deep convolutional networks operate layer per layer and linearize f progressively, as depth increases. Classification and regression problems are addressed similarly by considering the level sets of f , defined by $\Omega_t = \{x : f(x) = t\}$ if f is continuous. For classification, each level set is a particular class. Linear separability means that one can find w such that $f(x) \approx \langle \Phi(x), w \rangle$. It implies that $\Phi(\Omega_t)$ linearizes all level sets in hyperplanes orthogonal to some w .

To linearize level sets, we need to find directions along which $f(x)$ does not vary locally, and then linearize these directions in order to map them in a linear space. It is tempting to try to do this with some local data analysis along x . This is not possible because the training set includes few close neighbors in high dimension. We thus consider simultaneously all points $x \in \Omega$ and look for common directions along which $f(x)$ does not vary. This is where groups of symmetries come in. Translations and diffeomorphisms will illustrate the difficulty to linearize high dimensional symmetries, and provide a first mathematical ground to analyze convolution networks architectures.

We look for invertible operators which preserve the value of f . The action of an operator g on x is written $g.x$. A global symmetry is an invertible and often non-linear operator g from Ω to Ω , such that $f(g.x) = f(x)$ for all $x \in \Omega$. If g_1 and g_2 are global symmetries then $g_1.g_2$ is also a global symmetry, so products define groups of symmetries. Global symmetries are usually hard to find. We shall first concentrate on local symmetries. We suppose that there is a metric $|g|_G$ which measures the distance between $g \in G$ and the identity. A function f is locally invariant to the action of G if

$$\forall x \in \Omega, \exists C_x > 0, \forall g \in G \text{ with } |g|_G < C_x, f(g.x) = f(x). \quad (3.1)$$

We then say that G is a group of local symmetries of f . The constant C_x is the local range of symmetries which preserve f . Since Ω is a continuous subset of \mathbb{R}^d , we consider groups of operators which transport vectors in Ω with a continuous parameter. They are called Lie groups if the group has a differential structure.

Let us interpolate the d samples of x and define $x(u)$ for all $u \in \mathbb{R}^n$, with $n = 1, 2, 3$ respectively for time-series, images and volumetric data. The translation group $G = \mathbb{R}^n$ is an example of Lie group. The action of $g \in G = \mathbb{R}^n$ over $x \in \Omega$ is $g.x(u) = x(u - g)$. The distance $|g|_G$ between g and the identity is the Euclidean norm of $g \in \mathbb{R}^n$. The function f is locally invariant to translations if sufficiently small translations of x do not change $f(x)$. Deep convolutional networks compute convolutions, because they assume that translations are local symmetries of f . The dimension of a group G is the number of generators which define all group elements by products. For $G = \mathbb{R}^n$ it is equal to n .

Translations are not powerful symmetries because they are defined by only n variables, and $n = 2$ for images. Many image classification problems are also locally invariant to small deformations, which provide much stronger constraints. It means that f is locally invariant to diffeomorphisms $G = \text{Diff}(\mathbb{R}^n)$, which transform $x(u)$ with a differential warping of $u \in \mathbb{R}^n$. We do not know in advance what is the local range of diffeomorphism symmetries. For example, to classify images x of hand-written digits, certain deformations of x will preserve a digit class but modify the class of another digit. We shall linearize small diffeomorphisms g . In a space where local symmetries are linearized, we can find global symmetries by optimizing linear projectors which preserve the values of $f(x)$, and thus reduce dimensionality.

Local symmetries are linearized by finding a change of variable $\Phi(x)$ which locally linearizes the action of $g \in G$. We say that Φ is Lipschitz continuous if

$$\exists C > 0, \forall (x, g) \in \Omega \times G, \|\Phi(g.x) - \Phi(x)\| \leq C |g|_G \|x\|. \quad (3.2)$$

The norm $\|x\|$ is just a normalization factor often set to 1. The Radon-Nikodim property proves that the map that transforms g into $\Phi(g.x)$ is almost everywhere differentiable in the sense of Gateaux. If $|g|_G$ is small then $\Phi(x) - \Phi(g.x)$ is closely approximated by a bounded linear operator of g , which is the Gateaux derivative. Locally, it thus nearly remains in a linear space.

Lipschitz continuity over diffeomorphisms is defined relatively to a metric, which is now defined. A small diffeomorphism acting on $x(u)$ can be written as a translation of u by a $g(u)$:

$$g.x(u) = x(u - g(u)) \text{ with } g \in \mathbf{C}^1(\mathbb{R}^n). \quad (3.3)$$

This diffeomorphism translates points by at most $\|g\|_\infty = \sup_{u \in \mathbb{R}^n} |g(u)|$. Let $|\nabla g(u)|$ be the matrix norm of the Jacobian matrix of g at u . Small diffeomorphisms correspond to $\|\nabla g\|_\infty =$

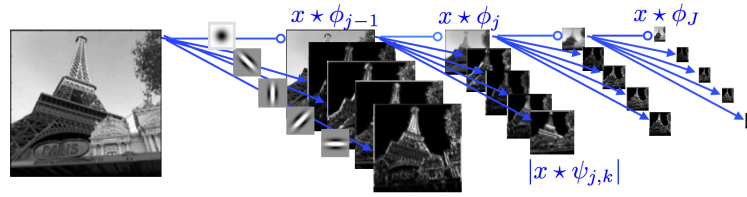


Figure 1. Wavelet transform of an image $x(u)$, computed with a cascade of convolutions with filters over $J = 4$ scales and $K = 4$ orientations. The low-pass and $K = 4$ band-pass filters are shown on the first arrows.

$\sup_u |\nabla g(u)| < 1$. Applying a diffeomorphism g transforms two points (u_1, u_2) into $(u_1 - g(u_1), u_2 - g(u_2))$. Their distance is thus multiplied by a scale factor, which is bounded above and below by $1 \pm \|\nabla g\|_\infty$. The distance of this diffeomorphism to the identity is defined by:

$$|g|_{\text{Diff}} = 2^{-J} \|\nabla g\|_\infty + \|\nabla g\|_\infty. \quad (3.4)$$

The factor 2^J is a local translation invariance scale. It gives the range of translations over which small diffeomorphisms are linearized. For $J = \infty$ the metric is globally invariant to translations.

4. Contractions and Scale Separation with Wavelets

Deep convolutional networks can linearize the action of very complex non-linear transformations in high dimensions, such as inserting glasses in images of faces [28]. A transformation of $x \in \Omega$ is a transport of x in Ω . To understand how to linearize any such transport, we shall begin with translations and diffeomorphisms. Deep network architectures are covariant to translations, because all linear operators are implemented with convolutions. To compute invariants to translations and linearize diffeomorphisms, we need to separate scales and apply a non-linearity. This is implemented with a cascade of filters computing a wavelet transform, and a pointwise contractive non-linearity. Section 7 extends these tools to general group actions.

To linearize the action of the translation group G with a linear operator, we must average x along the orbit $\{g.x\}_{g \in G}$, which are translations of x . This is done with a convolution by an averaging kernel $\phi_J(u) = 2^{-nJ} \phi(2^{-J}u)$ of size 2^J , with $\int \phi(u) du = 1$:

$$\Phi_J x(u) = x * \phi_J(u). \quad (4.1)$$

One can verify [24] that this averaging is Lipschitz continuous to diffeomorphisms for all $x \in \mathbf{L}^2(\mathbb{R}^n)$, over a translation range 2^J . However, it eliminates the variations of x above the frequency 2^{-J} . If $J = \infty$ then $\Phi_\infty x = \int x(u) du$, which eliminates nearly all information.

A diffeomorphism acts as a local translation and scaling of the variable u . If we let aside translations for now, to linearize small diffeomorphism actions we must linearize this scaling action. This is done by separating the variations of x at different scales with wavelets. We define K wavelets $\psi_k(u)$ for $u \in \mathbb{R}^n$. They are regular functions with a fast decay and a zero average $\int \psi_k(u) du = 0$. These K wavelets are dilated by 2^J : $\psi_{j,k}(u) = 2^{-Jn} \psi_k(2^{-J}u)$. A wavelet transform computes the local average of x at a scale 2^J , and variations at scales $2^j \geq 2^J$ with wavelet convolutions:

$$\mathcal{W}x = \{x * \phi_J(u), x * \psi_{j,k}(u)\}_{j \leq J, 1 \leq k \leq K}. \quad (4.2)$$

The parameter u is sampled on a grid such that intermediate sample values can be recovered by linear interpolations. The wavelets ψ_k are chosen so that \mathcal{W} is a contractive and invertible operator, and in order to obtain a sparse representation. This means that $x * \psi_{j,k}(u)$ is mostly zero besides few high amplitude coefficients corresponding to variations of $x(u)$ which “match” ψ_k at the scale 2^j . This sparsity plays an important role in non-linear contractions.

For audio signals, $n = 1$, sparse representations are usually obtained with at least $K = 12$ intermediate frequencies within each octave 2^j , which are similar to half-tone musical notes.

This is done by choosing a wavelet $\psi(u)$ having a frequency bandwidth of less than $1/12$ octave and $\psi_k(u) = 2^{k/K} \psi(2^{-k/K} u)$ for $1 \leq k \leq K$. For images, $n = 2$, we must discriminate image variations along different spatial orientation. It is obtained by separating angles $\pi k/K$, with an oriented wavelet which is rotated $\psi_k(u) = \psi(r_k^{-1} u)$. Intermediate rotated wavelets are approximated by linear interpolations of these K wavelets. Figure 1 shows the wavelet transform of an image, with $J = 4$ scales and $K = 4$ angles, where $x \star \psi_{j,k}(u)$ is subsampled at intervals 2^j . It has few large amplitude coefficients shown in white.

Wavelet transforms can be computed with a fast multiscale cascade of filters, which is at the core of deep network architectures. At each scale 2^j , we define a low-pass filter $w_{j,0}$ which increases the averaging scale from 2^{j-1} to 2^j , and band-pass filters $w_{j,k}$ which compute each wavelet:

$$\phi_j = w_{j,0} \star \phi_{j-1} \quad \text{and} \quad \psi_{j,k} = w_{j,k} \star \phi_{j-1}. \quad (4.3)$$

Let us write $x_j(u, 0) = x \star \phi_j(u)$ and $x_j(u, k) = x \star \psi_{j,k}(u)$ for $k \neq 0$. It results from (4.3) that for $0 < j \leq J$ and all $1 \leq k \leq K$:

$$x_j(u, k) = x_{j-1}(\cdot, 0) \star w_{j,k}(u). \quad (4.4)$$

These convolutions may be subsampled by 2 along u , in which case $x_j(u, k)$ is sampled at intervals 2^j along u .

Wavelet coefficients $x_j(u, k) = x \star \psi_{j,k}(u)$ oscillate at a scale 2^j . Translations of x smaller than 2^j modifies the complex phase of $x_j(u, k)$ if the wavelet is complex or its sign if it is real. Because of these oscillations, averaging x_j with ϕ_J outputs a zero signal. It is necessary to apply a non-linearity which removes oscillations. A modulus $\rho(\alpha) = |\alpha|$ computes such a positive envelop. Averaging $\rho(x \star \psi_{j,k}(u))$ by ϕ_J outputs non-zero coefficients which are locally invariant at a scale 2^J :

$$\Phi_J x(u, j, k) = \rho(x \star \psi_{j,k}) \star \phi_J(u). \quad (4.5)$$

Replacing the modulus by a rectifier $\rho(\alpha) = \max(0, \alpha)$ gives nearly the same result, up to a factor 2. One can prove [24] that this representation is Lipschitz continuous to actions of diffeomorphisms over $x \in \mathbf{L}^2(\mathbb{R}^n)$, and thus satisfies (3.2) for the metric (3.4). Indeed, the wavelet coefficients of x deformed by g can be written as the wavelet coefficients of x with deformed wavelets. Small deformations produce small modifications of wavelets in $\mathbf{L}^2(\mathbb{R}^n)$, because they are localized and regular. The resulting modifications of wavelet coefficients is of the order of the diffeomorphism metric $|g|_{\text{Diff}}$.

A modulus and a rectifier are contractive non-linear pointwise operators:

$$|\rho(\alpha) - \rho(\alpha')| \leq |\alpha - \alpha'|. \quad (4.6)$$

However, if $\alpha = 0$ or $\alpha' = 0$ then this inequality is an equality. Replacing α and α' by $x \star \psi_{j,k}(u)$ and $x' \star \psi_{j,k}(u)$ shows that distances are much less reduced if $x \star \psi_{j,k}(u)$ is sparse. Such contractions do not reduce as much the distance between sparse signals and other signals. This is illustrated by reconstruction examples in Section 6.

The local multiscale invariants in (4.5) have dominated pattern classification applications for music, speech and images, until 2010. It is called *Mel-spectrum* for audio [25] and SIFT type feature vectors [23] in images. Their limitations comes from the loss of information produced by the averaging by ϕ_J in (4.5). To reduce this loss, they are computed at short time scales $2^J \leq 50ms$ in audio signals, or over small image patches $2^{2J} = 16^2$ pixels. As a consequence, they do not capture large scale structures, which are important for classification and regression problems. To build a rich set of local invariants at a large scale 2^J , it is not sufficient to separate scales with wavelets, we must also capture scale interactions.

A similar issue appears in physics to characterize the interactions of complex systems. Multiscale separations are used to reduce the parametrization of classical many body systems, for example with multipole methods [11]. However, it does not apply to complex interactions, as in quantum systems. Interactions across scales, between small and larger structures, must be taken into account. Capturing these interactions with low-dimensional models is a major challenge.

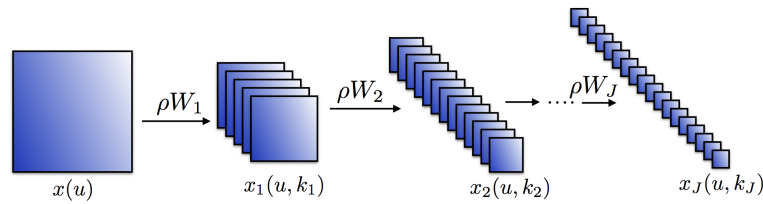


Figure 2. A convolution network iteratively computes each layer x_j by transforming the previous layer x_{j-1} , with a linear operator W_j and a pointwise non-linearity ρ .

We shall see that deep neural networks and scattering transforms provide high order coefficients which partly characterize multiscale interactions.

5. Deep Convolutional Neural Network Architectures

Deep convolutional networks are computational architectures introduced by Le Cun [20], providing remarkable regression and classification results in high dimension [17,19,21]. We describe these architectures illustrated by Figure 2. They iterate over linear operators W_j including convolutions, and predefined pointwise non-linearities.

A convolutional network takes in input a signal $x(u)$, which is here an image. An internal network layer $x_j(u, k_j)$ at a depth j is indexed by the same translation variable u , usually subsampled, and a channel index k_j . A layer x_j is computed from x_{j-1} by applying a linear operator W_j followed by a pointwise non-linearity ρ :

$$x_j = \rho W_j x_{j-1}.$$

The non-linearity ρ transforms each coefficient α of the array $W_j x_{j-1}$, and satisfies the contraction condition (4.6). A usual choice is the rectifier $\rho(\alpha) = \max(\alpha, 0)$ for $\alpha \in \mathbb{R}$, but it can also be a sigmoid, or a modulus $\rho(\alpha) = |\alpha|$ where α may be complex.

Since most classification and regression functions $f(x)$ are invariant or covariant to translations, the architecture imposes that W_j is covariant to translations. The output is translated if the input is translated. Since W_j is linear, it can thus be written as a sum of convolutions:

$$x_j(u, k_j) = \sum_k \sum_v x_{j-1}(v, k) w_{j,k_j}(u-v, k) = \sum_k \left(x_{j-1}(\cdot, k) \star w_{j,k_j}(\cdot, k) \right) (u). \quad (5.1)$$

The variable u is usually subsampled. For a fixed j , all filters $w_{j,k_j}(u, k)$ have the same support width along u , typically smaller than 10.

The operators ρW_j propagates the input signal $x_0 = x$ until the last layer x_J . This cascade of spatial convolutions defines translation covariant operators of progressively wider supports as the depth j increases. Each $x_j(u, k_j)$ is a non-linear function of $x(v)$, for v in a square centered at u , whose width Δ_j does not depend upon k_j . The width Δ_j is the spatial scale of a layer j . It is equal to $2^j \Delta$ if all filters w_{j,k_j} have a width Δ and the convolutions (5.1) are subsampled by 2.

Neural networks include many side tricks. They sometimes normalize the amplitude of $x_j(v, k)$, by dividing it by the norm of all coefficients $x_j(v, k)$ for v in a neighborhood of u . This eliminates multiplicative amplitude variabilities. Instead of subsampling (5.1) on a regular grid, a max pooling may select the largest coefficients over each sampling cell. Coefficients may also be modified by subtracting a constant adapted to each coefficient. When applying a rectifier ρ , this constant acts as a soft threshold, which increases sparsity. It is usually observed that inside network coefficients $x_j(u, k_j)$ have a sparse activation.

The deep network output $x_J = \Phi_J(x)$ is provided to a classifier, usually composed of fully connected neural network layers [21]. Supervised deep learning algorithms optimize the filter values $w_{j,k_j}(u, k)$ in order to minimize the average classification or regression error on the

training samples $\{x^i, f(x^i)\}_{i \leq q}$. There can be more than 10^8 variables in a network [21]. The filter update is done with a back-propagation algorithm, which may be computed with a stochastic gradient descent, with regularization procedures such as dropout. This high-dimensional optimization is non-convex, but despite the presence of many local minima, the regularized stochastic gradient descent converges to a local minimum providing good accuracy on test examples [12]. The rectifier non-linearity ρ is usually preferred because the resulting optimization has a better convergence. It however requires a large number of training examples. Several hundreds of examples per class are usually needed to reach a good performance.

Instabilities have been observed in some network architectures [31], where additions of small perturbations on x can produce large variations of x_J . It happens when the norms of the matrices W_j are larger than 1, and hence amplified when cascaded. However, deep network also have a strong form of stability illustrated by transfer learning [21]. A deep network layer x_J optimized on particular training databasis, can approximate different classification functions, if the final classification layers are trained on a new databasis. This means that it has learned stable structures, which can be transferred across similar learning problems.

6. Scattering on the Translation Group

A deep network alternates linear operators W_j and contractive non-linearities ρ . To analyze the properties of this cascade, we begin with a simpler architecture, where W_j does not combine multiple convolutions across channels in each layer. We show that such network coefficients are obtained through convolutions with a reduced number of equivalent wavelet filters. It defines a scattering transform [24] whose contraction and linearization properties are reviewed. Variance reduction and loss of information are studied with reconstructions of stationary processes.

Suppose that $x_j(u, k_j)$ is computed by convolving a single channel $x_{j-1}(u, k_{j-1})$ along u :

$$x_j(u, k_j) = \rho\left(x_{j-1}(\cdot, k_{j-1}) \star w_{j,h}(u)\right) \text{ with } k_j = (k_{j-1}, h). \quad (6.1)$$

It corresponds to a deep network filtering (5.1), where filters do not combine several channels. Iterating on j defines a convolution tree, as opposed to a full network. It results from (6.1) that

$$x_J(u, k_J) = \rho(\rho(\rho(x \star w_{1,h_1}) \star \dots) \star w_{J-1,h_{J-1}}) \star w_{J,h_J}). \quad (6.2)$$

If ρ is a rectifier $\rho(\alpha) = \max(\alpha, 0)$ or a modulus $\rho(\alpha) = |\alpha|$ then $\rho(\alpha) = \alpha$ if $\alpha \geq 0$. We can thus remove this non-linearity at the output of an averaging filter $w_{j,h}$. Indeed this averaging filter is applied to positive coefficients and thus computes positive coefficients, which are not affected by ρ . On the contrary, if $w_{j,h}$ is a band-pass filter then the convolution with $x_{j-1}(\cdot, k_{j-1})$ has alternating signs or a complex phase which varies. The non-linearity ρ removes the sign or the phase, which has a strong contraction effect.

Suppose that there are m band-pass filters $\{w_{j_n, h_{j_n}}\}_{1 \leq n \leq m}$ in the convolution cascade (6.2), and that all others are low-pass filters. If we remove ρ after each low-pass filter, we get m equivalent band-pass filters:

$$\psi_{j_n, k_n}(u) = w_{j_{n-1}+1, h_{j_{n-1}+1}} \star \dots \star w_{j_n, h_{j_n}}(u). \quad (6.3)$$

The cascade of J convolutions (6.2) is reduced to m convolutions with these equivalent filters

$$x_J(u, k_J) = \rho(\rho(\dots\rho(x \star \psi_{j_1, k_1}) \star \psi_{j_2, k_2}) \dots \star \psi_{j_{m-1}, k_{m-1}}) \star \psi_{J, k_J}(u)), \quad (6.4)$$

with $0 < j_1 < j_2 < \dots < j_{m-1} < J$. If the final filter w_{J, h_J} at the depth J is a low-pass filter then $\psi_{J, k_J} = \phi_J$ is an equivalent low-pass filter. In this case, the last non-linearity ρ can also be removed, which gives

$$x_J(u, k_J) = \rho(\rho(\dots\rho(x \star \psi_{j_1, k_1}) \star \psi_{j_2, k_2}) \dots \star \psi_{j_{m-1}, k_{m-1}}) \star \phi_J(u). \quad (6.5)$$

The operator $\Phi_J x = x_J$ is a wavelet scattering transform, introduced in [24]. Changing the network filters $w_{j,h}$ modifies the equivalent band-pass filters $\psi_{j,k}$. As in the fast wavelet

transform algorithm (4.4), if $w_{j,h}$ is a rotation of a dilated filter w_j then $\psi_{j,h}$ is a dilation and rotation of a single mother wavelet ψ .

The order $m = 1$ coefficients $x_J(u, k_J) = \rho(x \star \psi_{j_1, k_1}) \star \phi_J(u)$ are the wavelet coefficient computed in (4.5). The loss of information due to averaging is now compensated by higher order coefficient. For $m = 2$, $\rho(\rho(x \star \psi_{j_1, k_1}) \star \psi_{j_2, k_2}) \star \phi_J$ are complementary invariants. They measure interactions between variations of x at a scale 2^{j_1} , within a distance 2^{j_2} , and along orientation or frequency bands defined by k_1 and k_2 . These are scale interaction coefficients, missing from first order coefficients. Because ρ is strongly contracting, order m coefficients have an amplitude which decrease quickly as m increases [24,32]. For images and audio signals, the energy of scattering coefficients becomes negligible for $m \geq 3$. Let us emphasize that the convolution network depth is J , whereas m is the number of effective non-linearity of an output coefficient.

Section 4 explains that a wavelet transform defines representations which are Lipschitz continuous to actions of diffeomorphisms. Scattering coefficients up to the order m are computed by applying m wavelet transforms. One can prove [24] that it thus defines a representation which is Lipschitz continuous to the the action of diffeomorphisms. There exists $C > 0$ such that

$$\forall (g, x) \in \text{Diff}(\mathbb{R}^n) \times \mathbf{L}^2(\mathbb{R}^n), \quad \|\Phi_J(g.x) - \Phi_J x\| \leq C m \left(2^{-J} \|g\|_\infty + \|\nabla g\|_\infty \right) \|x\|,$$

plus a Hessian term which is neglected. This result is proved in [24] for $\rho(\alpha) = |\alpha|$, but it remains valid for any contractive pointwise operator such as rectifiers $\rho(\alpha) = \max(\alpha, 0)$. It relies on commutation properties of wavelet transforms and diffeomorphisms. It shows that the action of small diffeomorphisms is linearized over scattering coefficients.

Scattering vectors are restricted to coefficients of order $m \leq 2$, because their amplitude is negligible beyond. A translation scattering $\Phi_J x$ is well adapted to classification problems where the main source of intra-class variability are due to translations, to small deformations, or to ergodic stationary processes. For example, intra-class variabilities of hand-written digit images are essentially due to translations and deformations. On the MNIST digit data basis [6], applying a linear classifier to scattering coefficients $\Phi_J x$ gives state of the art classification errors. Music or speech classification over short time intervals of 100ms can be modeled by ergodic stationary processes. Good music and speech classification results are then obtained with a scattering transform [2]. Image texture classification are also problems where intra class variability can be modeled by ergodic stationary processes. Scattering transforms give state of the art results over a wide range of image texture databases [6,29], compared to other descriptors including power spectrum moments.

To analyze the information loss, we now study the reconstruction of x from its scattering coefficients, in a stochastic framework where x is a stationary process. This will raise variance and separation issues, where sparsity plays a role. It also demonstrates the importance of second order scale interaction terms, to capture non-Gaussian geometric properties of ergodic stationary processes. Let us consider scattering coefficients of order m

$$\Phi_J x(u, k) = \rho(\dots \rho(\rho(x \star \psi_{j_1, k_1}) \star \psi_{j_2, k_2}) \dots \star \psi_{j_m, k_m}) \star \phi_J(u), \quad (6.6)$$

with $\int \phi_J(u) du = 1$. If x is a stationary process then $\rho(\dots \rho(x \star \psi_{j_1, k_1}) \dots \star \psi_{j_m, k_m})$ remains stationary because convolutions and pointwise operators preserve stationarity. The spatial averaging by ϕ_J provides a non-biased estimator of the expected value of $\Phi_J x(u, k)$, which is a scattering moment:

$$\mathbb{E}(\Phi_J x(u, k)) = \mathbb{E} \left(\rho(\dots \rho(\rho(x \star \psi_{j_1, k_1}) \star \psi_{j_2, k_2}) \dots \star \psi_{j_m, k_m}) \right). \quad (6.7)$$

If x is a slow mixing process, which is a weak ergodicity assumption, then the estimation variance $\sigma_J^2 = \|\Phi_J x - \mathbb{E}(\Phi_J x)\|^2$ converges to zero [8] when J goes to ∞ . Indeed, Φ_J is computed by iterating on contractive operators, which average an ergodic stationary process x over progressively larger scales. One can prove that scattering moments characterize complex multiscale properties of fractals and multifractal processes, such as Brownian motions, Levi processes or Mandelbrot cascades [7].

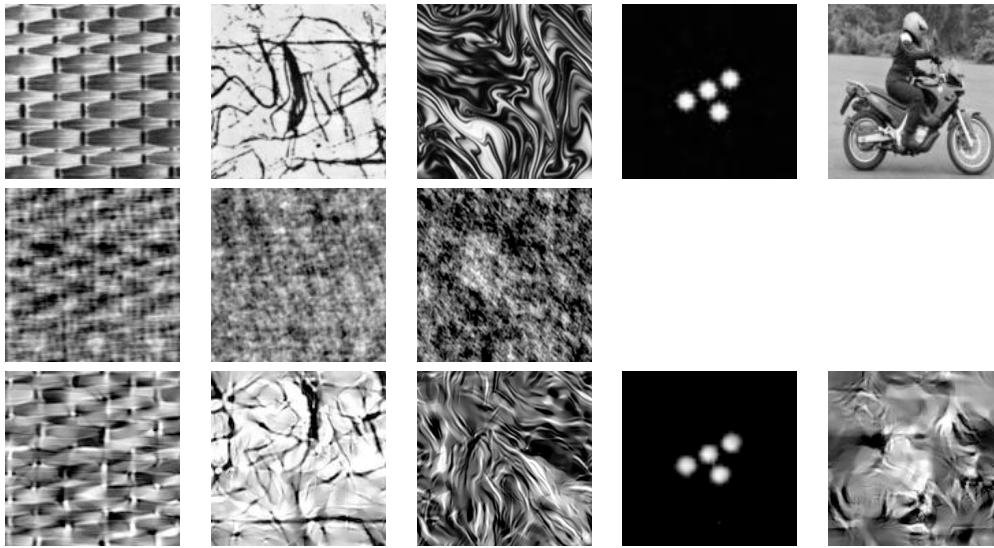


Figure 3. First row: original images. Second row: realization of a Gaussian process with same second covariance moments. Third row: reconstructions from first and second order scattering coefficients.

Scattering transforms are generally not invertible but given $\Phi_J(x)$ one can compute vectors \tilde{x} such that $\|\Phi_J(x) - \Phi_J(\tilde{x})\| \leq \sigma_J$. We initialize \tilde{x}_0 as a Gaussian white noise realization, and iteratively update \tilde{x}_n by reducing $\|\Phi_J(x) - \Phi_J(\tilde{x}_n)\|$ with a gradient descent method, until it reaches σ_J [8]. Since $\Phi_J(x)$ is not convex, there is no guaranteed convergence, but numerical reconstructions converge up to a sufficient precision. The recovered \tilde{x} is a stationary process having nearly the same scattering moments as x , whose properties are similar to a maximum entropy process for fixed scattering moments [8].

Figure 3 shows several examples of images x with N^2 pixels. The first three images are realizations of ergodic stationary textures. The second row gives realizations of stationary Gaussian processes having the same N^2 second order covariance moments as the top textures. The third column shows the vorticity field of a two-dimensional turbulent fluid. The Gaussian realization is thus a Kolmogorov type model, which does not restore the filament geometry. The third row gives reconstructions from scattering coefficients, limited to order $m \leq 2$. The scattering vector is computed at the maximum scale $2^J = N$, with wavelets having $K = 8$ different orientations. It is thus completely invariant to translations. The dimension of $\Phi_J x$ is about $(K \log_2 N)^2 / 2 \ll N^2$. Scattering moments restore better texture geometries than the Gaussian models obtained with N^2 covariance moments. This geometry is mostly captured by second order scattering coefficients, providing scale interaction terms. Indeed, first order scattering moments can only reconstruct images which are similar to realizations of Gaussian processes. First and second order scattering moments also provide good models of ergodic audio textures [8].

The fourth image has very sparse wavelet coefficients. In this case the image is nearly perfectly restored by its scattering coefficients, up to a random translation. The reconstruction is centered for comparison. Section 4 explains that if wavelet coefficients are sparse then a rectifier or an absolute value contractions ρ does not contract as much distances with other signals. Indeed, $|\rho(\alpha) - \rho(\alpha')| = |\alpha - \alpha'|$ if $\alpha = 0$ or $\alpha' = 0$. Inverting a scattering transform is a non-linear inverse problem, which requires to recover a lost phase information. Sparsity has an important role on such phase recovery problems [32]. Translating randomly the last motorcycle image defines a non-ergodic stationary process, whose wavelet coefficients are not as sparse. As a result, the reconstruction from a random initialization is very different, and does not preserve patterns which are important for most classification tasks. This is not surprising since there is much less scattering coefficients than image pixels. If we reduce 2^J so that the number of scattering

coefficients reaches the number of pixels then the reconstruction is of good quality, but there is little variance reduction.

Concentrating on the translation group is not so effective to reduce variance when the process is not translation ergodic. Applying wavelet filters can destroy important structures which are not sparse over wavelets. Next section addresses both issues. Impressive texture synthesis results have been obtained with deep convolutional networks trained on image data bases [14], but with much more output coefficients. Numerical reconstructions [13] also show that one can also recover complex patterns, such as birds, airplanes, cars, dogs, ships, if the network is trained to recognize the corresponding image classes. The network keeps some form of memory of important classification patterns.

7. Multiscale Hierarchical Convolution Networks

Scattering transforms on the translation group are restricted deep convolutional network architectures, which can suffer from variance and bias issues. We analyze a general class of convolutional network architectures with channels combinations, by extending the tools previously introduced. Contractions and invariants to translations are replaced by contractions along groups of local symmetries adapted to f , which are defined by parallel transports in each network layer. To regularize the network, we impose that groups of local symmetries are progressively growing as depth increases. It implies that all linear operators can be written as generalized convolutions across channels. The classification margin is preserved by replacing wavelets by adapted filter weights, which separate discriminative patterns in multiple network fibers.

Network layers $x_j = \rho W_j x_{j-1}$ are computed with operators ρW_j which contract and separate components of x_j , to control bias and variance. We shall see that W_j also needs to prepare x_j for the next transformation W_{j+1} , so consecutive operators W_j and W_{j+1} are strongly dependant. Each W_j is a contractive linear operator, $\|W_j z\| \leq \|z\|$ to reduce variance, and avoid instabilities when cascading such operators [31]. A layer x_{j-1} must separate f so that we can write $f(x) = f_{j-1}(x_{j-1})$ for some function $f_{j-1}(z)$. To simplify explanations, we concentrate on classification, where separation is an $\epsilon > 0$ margin condition:

$$\forall (x, x') \in \Omega^2, \|x_{j-1} - x'_{j-1}\| \geq \epsilon \text{ if } f(x) \neq f(x'). \quad (7.1)$$

The next layer $x_j = \rho W_j x_{j-1}$ has a smaller variance than x_{j-1} , but it must also satisfy

$$\forall (x, x') \in \Omega^2, \|\rho W_j x_{j-1} - \rho W_j x'_{j-1}\| \geq \epsilon \text{ if } f(x) \neq f(x'). \quad (7.2)$$

The operator W_j computes a linear projection which preserves this margin condition, but the resulting dimension reduction is limited. We can further contract the space non-linearly with ρ . To preserve the margin, it must reduce distances along non-linear displacements which transform any x_{j-1} into an x'_{j-1} which is in the same class. Such displacements are defined by local symmetries (3.1), which are non-linear transformations \bar{g} such that $f_{j-1}(x_{j-1}) = f_{j-1}(\bar{g}.x_{j-1})$.

To define a local invariant to a group of transformations G , we must process the orbit $\{\bar{g}.x_{j-1}\}_{\bar{g} \in G}$. However, W_j is applied to x_{j-1} not on the non-linear transformations $\bar{g}.x_{j-1}$ of x_{j-1} . The key idea is that a deep network can proceed in two steps. Let us write $x_j(u, k_j) = x_j(v)$ with $v \in P_j$. First, ρW_j computes an approximate mapping of such an orbit $\{\bar{g}.x_{j-1}\}_{\bar{g} \in G}$ into a parallel transport in P_j , which moves coefficients of x_j . Then W_{j+1} applied to x_j is filtering the orbits of this parallel transport. A parallel transport is defined by operators $g \in G_j$ acting on $v \in P_j$, and we write

$$\forall (g, v) \in G_j \times P_j, g.x_j(v) = x_j(g.v).$$

The operator W_j is defined so that G_j is a group of local symmetries: $f_j(g.x_j) = f_j(x_j)$ for small $|g|_{G_j}$. This is obtained if transports by $g \in G_j$ correspond approximatively to local symmetries \bar{g} of f_{j-1} . Indeed, if $g.x_j = g.[\rho W_j x_{j-1}] \approx \rho W_j[\bar{g}.x_{j-1}]$ then $f_j(g.x_j) = f_{j-1}(\bar{g}.x_{j-1}) = f_{j-1}(x_{j-1}) = f(x_j)$.

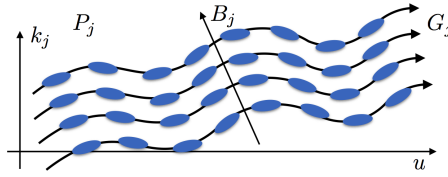


Figure 4. A multiscale hierarchical networks computes convolutions along the fibers of a parallel transport. It is defined by a group G_j of symmetries acting on the index set P_j of a layer x_j . Filter weights are transported along fibers.

The index space P_j is called a G_j -principal fiber bundle in differential geometry [26], illustrated by Figure 4. The orbits of G_j in P_j are fibers, indexed by the equivalence classes $B_j = P_j/G_j$. They are globally invariant to the action of G_j , and play an important role to separate f . Each fiber is indexing a continuous Lie group, but it is sampled along G_j at intervals such that values of x_j can be interpolated in between. As in the translation case, these sampling intervals depend upon the local invariance of x_j , which increases with j .

In a hierarchical convolution network, we further impose that local symmetry groups are growing with depth, and can be factorized:

$$\forall j \geq 0, \quad G_j = G_{j-1} \times H_j. \quad (7.3)$$

The hierarchy begins for $j = 0$ by the translation group $G_0 = \mathbb{R}^n$, which acts on $x(u)$ through the spatial variable $u \in \mathbb{R}^n$. The condition (7.3) is not necessarily satisfied by general deep networks, besides $j = 0$ for translations. It is used by joint scattering transforms [3,29] and has been proposed for unsupervised convolution network learning [9]. Proposition 7.1 proves that this hierarchical embedding implies that each W_j is a convolution on G_{j-1} .

Proposition 7.1. *The group embedding (7.3) implies that x_j can be indexed by $(g, h, b) \in G_{j-1} \times H_j \times B_j$ and there exists $w_{j,h,b} \in \mathbb{C}^{P_{j-1}}$ such that*

$$x_j(g, h, b) = \rho \left(\sum_{v' \in P_{j-1}} x_{j-1}(v') w_{j,h,b}(g^{-1} \cdot v') \right) = \rho \left(x_{j-1} \star^{j-1} w_{j,h,b}(g) \right), \quad (7.4)$$

where $h \cdot b$ transports $b \in B_j$ by $h \in H_j$ in P_j .

Proof. We write $x_j = \rho W_j x_{j-1}$ as inner products with row vectors:

$$\forall v \in P_j, \quad x_j(v) = \rho \left(\sum_{v' \in P_{j-1}} x_{j-1}(v') w_{j,v}(v') \right) = \rho \left(\langle x_{j-1}, w_{j,v} \rangle \right). \quad (7.5)$$

If $\bar{g} \in G_j$ then $\bar{g} \cdot x_j(v) = x_j(\bar{g} \cdot v) = \rho \left(\langle x_{j-1}, w_{j,\bar{g} \cdot v} \rangle \right)$. One can write $w_{j,v} = w_{j,\bar{g},b}$ with $\bar{g} \in G_j$ and $b \in B_j = P_j/G_j$. If $G_j = G_{j-1} \times H_j$ then $\bar{g} \in G_j$ can be decomposed into $\bar{g} = (g, h) \in G_{j-1} \times H_j$, where $g \cdot x_j = \rho \left(\langle g \cdot x_{j-1}, w_{j,b} \rangle \right)$. But $g \cdot x_{j-1}(v') = x_{j-1}(g \cdot v')$ so with a change of variable we get $w_{j,g,b}(v') = w_{j,b}(g^{-1} \cdot v')$. Hence $w_{j,\bar{g},b}(v') = w_{j,(g,t),b}(v) = h \cdot w_{j,h,b}(g^{-1} \cdot v')$. Inserting this filter expression in (7.5) proves (7.4). \square

This proposition proves that W_j is a convolution along the fibers of G_{j-1} in P_{j-1} . Each $w_{j,h,b}$ is a transformation of an elementary filter $w_{j,b}$ by a group of local symmetries $h \in H_j$ so that $f_j(x_j(g, h, b))$ remains constant when x_j is locally transported along h . We give below several examples of groups H_j and filters $w_{j,h,b}$. However, learning algorithms compute filters directly, with no prior knowledge on the group H_j . The filters $w_{j,h,b}$ can be optimized so that variations of $x_j(g, h, b)$ along h captures a large variance of x_{j-1} within each class. Indeed, this variance is then reduced by the next ρW_{j+1} . The generators of H_j can be interpreted as *principal symmetry generators*, by analogy with the principal directions of a PCA.

The scattering convolution along translations (6.1) is replaced in (7.4) by a convolution along G_{j-1} , which combines different layer channels. Results for translations can essentially be extended to the general case. If $w_{j,h,b}$ is an averaging filter then it computes positive coefficients, so the non-linearity ρ can be removed. If each filter $w_{j,h,b}$ has a support in a single fiber indexed by b , as in Figure 4, then $B_{j-1} \subset B_j$. It defines a *generalized scattering transform*, which is a structured multiscale hierarchical convolutional network such that $G_{j-1} \rtimes H_j = G_j$ and $B_{j-1} \subset B_j$. If $j = 0$ then $G_0 = P_0 = \mathbb{R}^n$ so B_0 is reduced to 1 fiber.

As in the translation case, we need to linearize small deformations in $\text{Diff}(G_{j-1})$, which include much more local symmetries than the low-dimensional group G_{j-1} . A small diffeomorphism $g \in \text{Diff}(G_{j-1})$ is a non-parallel transport along the fibers of G_{j-1} in P_{j-1} , which is a perturbation of a parallel transport. It modifies distances between pairs of points in P_{j-1} by scaling factors. To linearize such diffeomorphisms, we must use localized filters whose supports have different scales. Scale parameters are typically different along the different generators of $G_{j-1} = \mathbb{R}^n \rtimes H_1 \rtimes \dots \rtimes H_{j-1}$. Filters can be constructed with wavelets dilated at different scales, along the generators of each group H_k for $1 \leq k \leq j$. Linear dimension reduction mostly results from this filtering. Variations at fine scales may be eliminated, so that $x_j(g, h, b)$ can be coarsely sampled along g .

For small j , the local symmetry groups H_j may be associated to linear or non-linear physical phenomena such as rotations, scaling, colored illuminations or pitch frequency shifts. Let $SO(n)$ be the group of rotations. Rigid movements $SE(n) = \mathbb{R}^n \rtimes SO(n)$ is a non-commutative group, which often includes local symmetries. For images, $n = 2$, this group becomes a transport in P_1 with $H_1 = SO(n)$ which rotates a wavelet filter $w_{1,h}(u) = w_1(r_h^{-1}u)$. Such filters are often observed in the first layer of deep convolutional networks [13]. They map the action of $\bar{g} = (v, r_k) \in SE(n)$ on x to a parallel transport of $(u, h) \in P_1$ defined for $g \in G_1 = \mathbb{R}^2 \times SO(n)$ by $g.(u, h) = (v + r_k u, h + k)$. Small diffeomorphisms in $\text{Diff}(G_j)$ correspond to deformations along translations and rotations, which are sources of local symmetries. A roto-translation scattering [27,29] linearizes them with wavelet filters along translations and rotations, with $G_j = SE(n)$ for all $j > 1$. This roto-translation scattering can efficiently regress physical functionals which are often invariant to rigid movements, and Lipschitz continuous to deformations. For example, quantum molecular energies $f(x)$ are well estimated by sparse regressions over such scattering representations [18].

Pitch frequency shift is a more complex example of a non-linear symmetry for audio signals. Two different musical notes of a same instrument have a pitch shift. Their harmonic frequencies are multiplied by a factor 2^h , but it is not a dilation because the note duration is not changed. With narrow band-pass filters $w_{1,h}(u) = w_1(2^{-h}u)$, a pitch shift is approximatively mapped to a translation along $h \in H_1 = \mathbb{R}$ of $\rho(x \star w_{1,h}(u))$, with no modification along the time u . The action of $g = (v, k) \in G_1 = \mathbb{R} \times \mathbb{R} = \mathbb{R}^2$ over $(u, h) \in P_1$ is thus a two-dimensional translation $g.(u, h) = (u + v, h + k)$. A pitch shift also comes with deformations along time and log-frequencies, which define a much larger class of symmetries in $\text{Diff}(G_1)$. Two-dimensional wavelets along (u, h) can linearize these small time and log-frequency deformations. These define a joint time-frequency scattering applied to speech and music classifications [3]. Such transformations were first proposed as neurophysiological models of audition [25].

The group H_j is associated to complex transformations when j increases. It needs to capture large transformations between different patterns in a same class, for example chairs of different styles. Let us consider training samples $\{x^i\}_i$ of a same class. The iterated network contractions transforms them into vectors $\{x_{j-1}^i\}_i$ which are much closer. Their distances define weighted graphs which sample underlying continuous manifolds in the space. Such manifolds clearly appear in [5], for high-level patterns such as chairs or cars, together with poses and colors. As opposed to manifold learning, deep network filters result from a global optimization which can be computed in high dimension. The principal symmetry generators of H_j is associated to common transformations over all manifolds of examples x_{j-1}^i , which preserve the class while capturing large intra-class variance. They are approximatively mapped to a parallel transport in x_j by

the filters $w_{j,h,b}$. The diffeomorphisms in $\text{Diff}(G_j)$ are non-parallel transports corresponding to high-dimensional displacements on the manifolds of x_{j-1} . Linearizing $\text{Diff}(G_j)$ is equivalent to partially flatten simultaneously all these manifolds, which may explain why manifolds are progressively more regular as the network depth increases [5], but it involves open mathematical questions.

We have up to now been concentrated on variance reduction through contractions. The bias can be controlled, thanks to the existence of multiple fibers B_j in P_j , by adapting filters instead of using standard wavelets. The fibers indexed by $b \in B_j$ are separation instruments, which increase dimensionality to avoid reducing the classification margin. They prevent from collapsing vectors in different classes, which have a distance $\|x_{j-1} - x'_{j-1}\|$ close to the minimum margin ϵ . These vectors are close to classification frontiers. They are called *multiscale support vectors*, by analogy with support vector machines. To avoid further contracting their distance, they can be separated along different fibers indexed by b . The separation is achieved by filters $w_{j,h,b}$, which transform x_{j-1} and x'_{j-1} into $x_j(g, h, b)$ and $x'_j(g, h, b)$ having sparse supports on different fibers b . The next contraction ρW_{j+1} reduces distances along fibers indexed by $(g, h) \in G_j$, but not across $b \in B_j$, which preserves distances. The contraction increases with j so the number of support vectors close to frontiers also increases, which implies that more fibers are needed to separate them.

When j increases, the size of x_j is a balance between the dimension reduction along fibers, by subsampling $g \in G_j$, and an increasing number of fibers B_j which encode progressively more support vectors. Coefficients in these fibers become more specialized and invariants, as the grandmother neurons observed in deep layers of convolutional networks [1]. They have a strong response to particular patterns and are invariant to a large class of transformations. In this model, the choice of filters $w_{j,h,b}$ are adapted to produce sparse representations of multiscale support vectors. They provide a sparse distributed code, defining invariant pattern memorisation. This memorisation is numerically observed in deep network reconstructions [13], which can restore complex patterns within each class. Let us emphasize that groups and fibers are mathematical ghost behind filters, which are never computed. The learning optimization is directly performed on filters, which carry the trade-off between contractions to reduce variance and separation to preserve margin and thus to control bias.

8. Conclusion

This paper provides a mathematical framework to analyze the bias and variance properties of deep convolutional networks. In this model, network filters are guiding non-linear contractions, to reduce the variance in directions of local symmetries. The bias is controlled by progressive separations, which define an increasing number of network fibers. These fibers combine invariances along groups of symmetries and distributed pattern representations, which could be sufficiently stable to explain transfer learning of deep networks [21]. However, we need complexity measures, approximation theorems in spaces of high-dimensional functions, and guaranteed convergence of filter optimization, to fully understand the mathematics of these convolution networks.

Besides learning, there are striking similarities between these multiscale mathematical tools and the treatment of symmetries in particle and statistical physics [15]. One can expect a rich cross fertilization between high-dimensional learning and physics, through the development of a common mathematical language.

Data accessibility. Softwares reproducing experiments can be retrieved at www.di.ens.fr/data/software.

Acknowledgements. I would like to thank Carmine Emanuele Cella, Ivan Dokmaninc, Sira Ferradans, Edouard Oyallon and Irène Waldspurger for their helpful comments and suggestions.

Funding statement. This work was supported by the ERC grant InvariantClass 320959.

References

1. Agrawal A, Girshick R, Malik J, 2014, *Analyzing the Performance of Multilayer Neural Networks for Object Recognition*, Proc. of ECCV.
2. Andèn J, Mallat S. 2014 *Deep Scattering Spectrum*, IEEE Trans. on Signal Processing, **62**.
3. Andèn J, Lostanlen V, Mallat S. 2015, *Joint time-frequency scattering for audio classification*, Proc. of Machine Learn. for Signal Proc., Boston.
4. Anselmi F, Leibo J, Rosasco L, Mutch J, Tacchetti A, Poggio T. 2013 *Unsupervised Learning of Invariant Representations in Hierarchical Architectures* arXiv:1311.4158.
5. Aubry M, Russell B. 2015 *Understanding deep features with computer-generated imagery*, arXiv:1506.01151.
6. Bruna J, Mallat S. 2013, *Invariant Scattering Convolution Networks*, IEEE Trans. on PAMI **35**.
7. Bruna J, Mallat S, Bacry E, Muzy JF. 2015 *Intermittent process analysis with scattering moments* Annals of Stats. **43**.
8. Bruna J, Mallat S. 2015 *Stochastic scattering models* submit. IEEE Trans. Info. Theory.
9. Bruna J., Szlam A., Le Cun Y., 2014, *Learning Stable Group Invariant Representations with Convolutional Networks*, ICLR 2014.
10. Candès E, Donoho D. 1999, *Ridglets: a key to high-dimensional intermittency ?* Phil. Trans. Roy. S. A **357**.
11. Carrier J, Greengard L, Rokhlin V. 1988 *A Fast Adaptive Multipole Algorithm for Particle Simulations*, SIAM J. Sci. Stat. Comput. **9**.
12. Choromanska A, Henaff M, Mathieu M, Ben Arous G, Le Cun Y. 2014, *The loss surfaces of multilayer networks*, arXiv:1412.0233.
13. Denton E, Chintala S, Szlam A, Fergus R. 2015 *Deep generative image models using a Laplacian pyramid of adversarial networks*, NIPS 2015.
14. Gatys LA, Ecker AS, Bethge M. 2015 *Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks*, arXiv:1505.07376.
15. Glinsky M 2011 *A new perspective on renormalization: the scattering transformation*, arXiv:1106.4369.
16. Hastie T, Tibshirani R, Friedman J. 2009 *The elements of statistical learning*, Springer Series in Statistics.
17. Hinton G, Li D, Yu D, Dahl G, Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath T, Kingsbury B. 2012 *Deep neural networks for acoustic modeling in speech recognition* IEEE Signal Processing Magazine, **29**, 82-97.
18. Hirn M, Poilvert N, Mallat S. 2015 *Quantum energy regression using scattering transforms* arXiv:1502.02077.
19. Krizhevsky A, Sutskever I, Hinton G. 2012 *ImageNet classification with deep convolutional neural networks*, In Proc. of NIPS, p. 1090-1098, 2012.
20. Le Cun Y, Boser B, Denker J, Henderson D, Howard R, Hubbard W, Jackelt L. 1990 *Handwritten digit recognition with a back-propagation network*, In Proc. of NIPS. **3**.
21. Le Cun Y, Bengio Y, Hinton G. 2015 *Deep learning*, Nature, **521**.
22. Leung MK, Xiong HY, Lee LJ, Frey BJ. 2014 *Deep learning of the tissue regulated splicing code*, Bioinformatics, **30**.
23. Lowe DG. 2004 *SIFT: Scale Invariant Feature Transform*, J. of Computer Vision, **60**.
24. Mallat S. 2012, *Group Invariant Scattering*, Comm. in Pure and Applied Mathematics, **65**.
25. Mesgarani M, Slaney M, Shamma S. 2006 *Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations* IEEE Trans. Audio, Speech, Lang. Process., **14**.
26. J. Petitot, 2008 *Neurogéométrie de la vision*, Éditions de l'École Polytechnique.
27. Oyallon E, Mallat S. 2015 *Deep roto-translation scattering for object classification*, Proc. of CVPR.
28. Radford A, Metz L, Chintala S. 2016 *Unsupervised representation learning with deep convolutional generative adversarial networks*, ICLR 2016.
29. Sifre L, Mallat S. 2013 *Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination*, In Proc. of CVPR.
30. Sutskever I, Vinyals O, Le QV. 2015 *Sequence to sequence learning with neural networks* In Proc. of NIPS, **27**.
31. Szegedy C, Erhan D, Zaremba W, Sutskever I, Goodfellow I, Bruna J, Fergus R. 2014, *Intriguing properties of neural networks*, In Proc. of ICLR.
32. Waldspurger I. 2015 *Wavelet transform modulus: phase retrieval and scattering*, Ph.D Ecole Normale Supérieure.