Wavelets and Applications

Kévin Polisano
kevin.polisano@univ-grenoble-alpes.fr

M2 MSIAM & Ensimag 3A MMIS

January 21, 2022

The Scattering Transform

Understanding deep convolutional networks

Supervised learning against high dimension

- Data in high dimension $x \in \mathbb{R}^d$ with $d \approx 10^6$
- $f(x)$ represents a label of a class (whose can be also big, e.g $2 \cdot 10^3$ for ImageNet) for **classification** tasks, or a real for **regression**.
- Training set of n samples $\{x_i, y_i = f(x_i)\}_{i \leq n}$ (few samples per class)
- Supervised learning aims at generalizing from the samples to predict $f(x)$ for new datas.

Intuitively, to do an **interpolation** in x we need somehow to average among known samples $\{x_i, y_i\}$ in the neighborhood of x, saying:

$$\forall x \in [0,1]^d, \exists x_i \in [0,1]^d, \quad \|x - x_i\| \leqslant \epsilon$$

then if the x_i's are uniformly distributed, it would require ϵ^{-d} points to cover $[0,1]^d$ entirely!

Points are far away in high dimension $\Rightarrow$ Curse of dimensionality

Understanding deep convolutional networks

Kernel learning

1. **Representation**. Change of variable $\Phi(x) = \{\phi_k(x)\}_{k \leqslant d'}$ (*features*) in order to nearly linearize class bounderies:

$$x = (v_1, \dots, v_d) \xrightarrow{\Phi} \Phi(x) = (v'_1, \dots, v'_d)$$

2. **Classifier**. Find an hyperplan (that is an vector w orthogonal to the hyperplan) which seperates the transformed data:

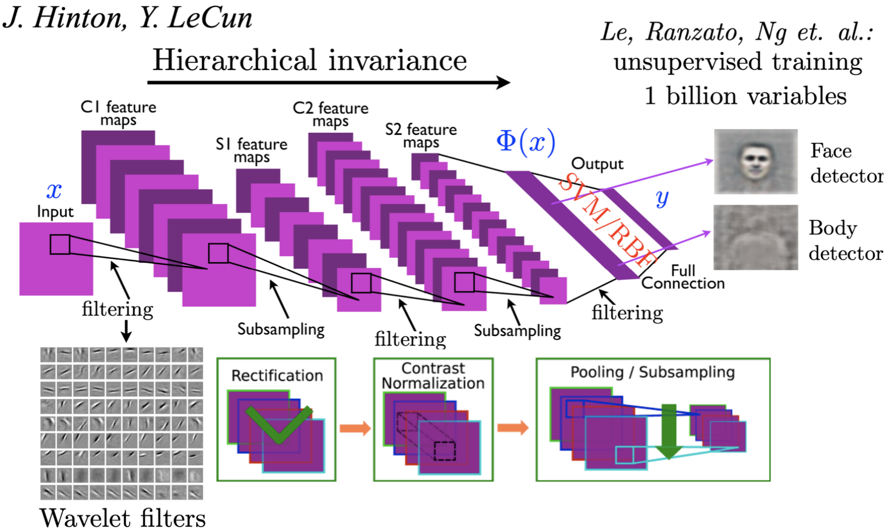
$$\tilde{f}(x) = \text{sign}(\langle \Phi(x), w \rangle + b) = \text{sign}\left(\sum_k w_k v'_k + b\right)$$

Questions:

- How to construct such a representation Φ?
- What regularity is needed?
- Can wavelets be useful to understand and draw CNN architectures?

Understanding deep convolutional networks

CNN architecture



Credits: S. Mallat

Understanding deep convolutional networks

CNN architecture: why are they so efficient for images classification?

- Why convolutions? Which filters?
- Why pooling? Why multi-stage and how deep?
- Why and which non-linearities?
- Why normalization?
- What is the role of sparsity?

$\Rightarrow$ what are the mathematical operators behind such architectures?

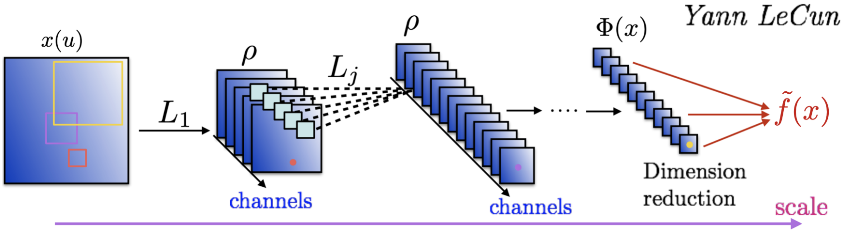


Figure: L_j: sum of spatial convolutions across channels, subsampling. ρ: scalar non-linearity ($\max(u, 0)$, $|u|$, ...)

Credits: S. Mallat

Understanding deep convolutional networks

The "3S" ingredients for reducing the dimensionality problem

1. Separability: variables separation can reduce the dimensionality from d to K problems of dimension $q \ll d$ (e.g decomposing an image $10^3 \times 10^3$ in small independant patches 8×8, whose interactions between pixels are essentially local $\Rightarrow$ SIFT). It is important to make **scales separation** but also to capture their interaction: deeper neurons can "see" greater portion of the image.
2. Symmetry: spatial symmetries produce **translation/rotation/flip invariance** (e.g convolution filters induce translation invariance) and reduce the dimensionality by eliminating some variables.
3. Sparsity: pattern recognition consists on decomposing the problem on sparse **elementary structures** in dictionaries (cat's hears, human's eyes, ...) in particular through the activation functions.

$\Rightarrow$ take advantage both of *a priori* information hard-coded in the network architecture and learning to design Φ.

Symmetry group

To know the regularity of f one can study it through local but also global transformation such that symmetry group of f:

$$G = \{g : \forall x \in \Omega, \quad f(g.x) = f(x)\}$$

- The functions g preserve the level sets $\Omega_t = \{x : f(x) = t\}$, that is if $x \in \Omega_t$ and $g \in G$ then $g.x \in \Omega_t$. So it is easy to verify the solutions of a level set has a structure of group.
- **Information *a priori***, a symmetry subgroup $H \subset G$. If $g \in H$ then x and $g.x$ have the same label $f(g.x) = f(x)$, so belong to the same class of equivalence. The quotient of Ω by H is denoted by $\Omega \backslash H$, for $x_0 \in \Omega \backslash H$ then it defines a class of equivalence:

  $$H_{x_0} = \{x \in \Omega : g \in H \text{ s.t } g.x = x_0\}$$

  Example: if x_0 is an image and $f(x_0)$ its label (cat/dog), then by translating $x = g.x_0 \in H_{x_0}$ the label remains the same $f(x) = f(x_0)$.
- One can then reduce the number of variables (variability) within the class of equivalence (reduction of dimensionality).

Symmetry group

Lie group: infinitely small generators

Reduction of dimensionality in the continuous case:

$$\dim(\Omega \backslash H) = \dim(\Omega) - \dim(H)$$

Diffeomorphisms group

Let $g : [0,1]^2 \to [0,1]^2$ be a $\mathcal{C}^1$ function acting on the underlying variable of x, namely u which is a low-dimensionnal quantity:

$$g.(x(u)) = x(g(u))$$

Examples

- Translation: $g.x(u) = x(u - g)$ with $g \in \mathbb{R}^2$
- Rotation: $g.x(u) = x(\mathbf{R}_g u)$ with $g \in [0, 2\pi]$
- Globally invariant to the translation group $\Rightarrow$ small
- Locally invariant to small diffeomorphisms $\Rightarrow$ **HUGE**

Continuous transports by successive action of generators $f(x_i) = f(x_0)$

$$O_x = \{g.x\}_{g \in G} \quad \text{(orbit = differentiable surface of iso-label)}$$

Understanding deep convolutional networks

Using the information *a priori* on the symmetry group of f to define the representation Φ for the final classification/regression (last layer):

$$\tilde{f}(x) = \langle \Phi(x), w \rangle = \sum_k w_k \phi_k$$

In order that $\tilde{f}$ is a good approximation of f, we impose that it has the **same invariants** $g \in G$ that is G is a symmetry group of Φ.

Two possibilities:

1. G **known and low dimension** (translation, rotation, ...)
   $\Rightarrow$ constructing directly Φ
2. G **unknown and high dimension** (diffeomorphisms)
   $\Rightarrow$ linearization + learning invariant through the classifier.

$$\tilde{f}(x) = \tilde{f}(g.x) \Rightarrow \langle \Phi(x), w \rangle = \langle \Phi(g.x), w \rangle \Rightarrow \langle \Phi(x) - \Phi(g.x), w \rangle = 0$$

$$\Phi(x) - \Phi(g.x) \in V \perp w$$

$\rightsquigarrow$ If V is a hyperplan it implies to linearize transformations, by considering small deformations g.

Linearization of small deformations

- **Linearize group actions**: $g.x = x + \tau.x$ so locally the tangent hyperplan to the orbit O_x is given by τ (Lie algebra).
- **For small deformations** $g.x(u) = x(u - \tau(u))$ we can write the action τ as a "global" action (the translation) and a small "local" action (the deformation), since $\tau(u) \approx \tau(u_0) + \nabla\tau(u_0)(u - u_0)$ then

$$x(u - \tau(u)) = x\Big(\underbrace{(\mathbb{I} - \nabla\tau(u_0))(u - u_0)}_{\text{local deformation}} + \underbrace{u_0 - \tau(u_0)}_{\text{global translation}} \Big)$$

- **Distance** for small deformations: $|g|_G = \|\tau\|_\infty + \|\nabla\tau\|_\infty$
- We do not know in advance what is the local range of diffeomorphism symmetries.
  Example: to classify images x of handwritten digits, certain deformations of x will preserve a digit class but modify the class of another digit.

Linearization of small deformations

- We shall linearize small diffeomorphims g via the change of variable $\Phi(x)$, which is say Lipschitz-continuous if

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

- The Radon–Nikodim property proves that the map that transforms g into $\Phi(g.x)$ is almost everywhere differentiable in the sense of Gâteaux. If $|g|_G$ is small, then $\Phi(g.x) - \Phi(x)$ is closely approximated by a bounded linear operator of g, which is the Gâteaux derivative. **Locally, it thus nearly remains in a linear space**.

$\Rightarrow$ The Lipschitz property of Φ is difficult to be obtained. Indeed, a local deformation is a dilation, so **the representation will have to be based on dilations**, that is we will need to separate scales with the wavelet transform.

Stable invariants

Fourier is not relevant

If $\Phi(x) = \{|\widehat{x}(\omega)|\}_\omega$ then:

- **Invariance to translations** $x_c(t) = x(t-c)$

$$\forall c \in \mathbb{R}, \quad \Phi(x_c) = \Phi(x)$$

- Not Lipschitz stable to small deformation $x_\tau(t) = x(t - \tau(t))$ where $\tau(t) = \epsilon t$ for example. The Fourier transform of $x(t-\tau(t)) = x((1-\epsilon)t)$ is $\widehat{x}(\omega(1+\epsilon))$, so two "bumps" centered in $\omega = \pm\omega_0$ will be "shifted" toward low frequencies by a quantity $\epsilon\omega_0$, such that they are not superposed anymore and then

$$\|\Phi(x_\tau) - \Phi(x)\| \neq \epsilon$$

.

$\Rightarrow$ Wavelets are localized waveforms and are thus stable to deformations, as opposed to Fourier sinusoidal waves

Stable invariants

Why wavelets?

- Wavelets are uniformly stable to deformations:
  If $\psi_{\lambda,\tau}(t) = \psi_\lambda(t - \tau(t))$ then

$$\|\psi_\lambda - \psi_{\lambda,\tau}\| \leq C \sup_t |\nabla \tau(t)|$$

- Wavelet separate multiscale information
- Wavelets provide sparse representation

Multiscale Wavelet Transform

- Complex wavelet $\psi(u) = \psi^a(u) + i\psi^b(u)$
- Dilated 1D wavelet: $\psi_\lambda(u) = 2^{-j/Q}\psi(2^{-j/Q}u)$ with $\lambda = 2^{-j/Q}$
- For images with two variables $u = (u_1, u_2)$ add a rotation $r \in G$ of angles $2k\pi/K$ for $0 \leq k < K$:

$$\psi_\lambda(u) = 2^{-2j}\psi(2^{-j}r^{-1}u), \quad \lambda = (2^{-j}, r)$$

- Wavelet transform:

$$Wx = \begin{pmatrix} x \star \phi(u) \\ x \star \psi_\lambda(u) \end{pmatrix}_{u,\lambda}$$

- If $|\widehat{\phi}(\omega)|^2 + \sum_\lambda |\widehat{\psi}_\lambda(\omega)|^2 = 1$ then W is unitary: $\|Wx\|^2 = \|x\|^2$

Stable translation invariance

- $x \star \psi_\lambda$ is translation covariant, not invariant and

$$\int x \star \psi_\lambda(u)\,\mathrm{d}u = 0$$

- Translation invariant representation: $\int M(x \star \psi_\lambda)(u)\,\mathrm{d}u$
- Diffeomorphism stability: M commutes with diffeomorphisms
- L^2 stability: $\|Mh\| = \|h\|$ and $\|Mg - Mh\| \leq \|g - h\|$

$$\Rightarrow M(h)(u) = |h(u)| = \sqrt{|h^a(u)|^2 + |h^b(u)|^2}$$

Wavelet translation invariance

- The modulus $|x \star \psi_{\lambda_1}| = \sqrt{|x \star \psi^a_{\lambda_1}|^2 + |x \star \psi^b_{\lambda_1}|^2}$ (*pooling*) is a regular envelop
- The average $|x \star \psi_{\lambda_1}| \star \phi(t)$ is invariant to small translations relatively to the support of ϕ
- Full translation invariance at the limit:

$$\lim_{\phi \to 1} |x \star \psi_{\lambda_1}| = \int |x \star \psi_{\lambda_1}(u)| \mathrm{d}u = \|x \star \psi_{\lambda_1}\|_1$$

- First Wavelet transform modulus:

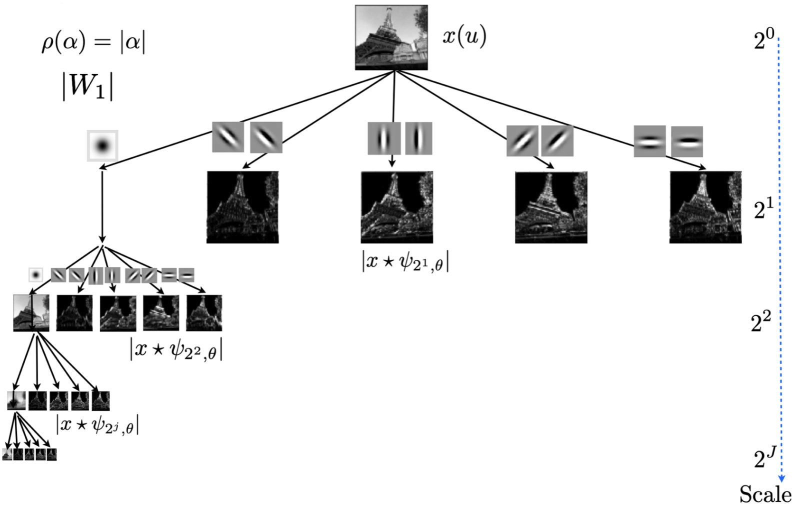
$$\rho W_1 = |W_1| x = \begin{pmatrix} x \star \phi_{2^J} \\ |x \star \psi_{\lambda_1}| \end{pmatrix}_{\lambda_1}$$

- Second Wavelet transform modulus (for recovering high freq. lost):

$$|W_2| |x \star \psi_{\lambda_1}| = \begin{pmatrix} |x \star \psi_{\lambda_1}| \star \phi_{2^J} \\ ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \end{pmatrix}_{\lambda_2}$$

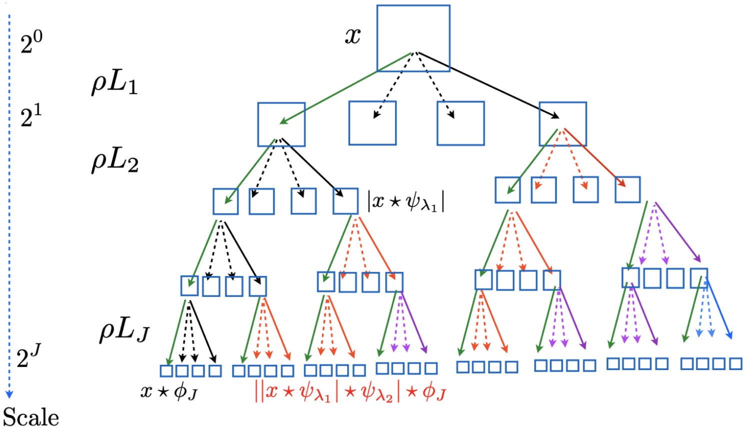
- Translation invariance by averaging $||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi_{2^J}, \quad \forall \lambda_1, \lambda_2$

Scattering Network



Credits: S. Mallat

Scattering Network



$$S_J = \rho W_1 \;\; \rho W_2 \; \cdots \; \rho W_J$$

$$\rho(\alpha) = |\alpha| \qquad S_J x = \Big\{ |||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2} \star \ldots| \star \psi_{\lambda_m}| \star \phi_J \Big\}_{\lambda_k}$$

Interactions across scales

Credits: S. Mallat

Scattering Properties

$$S_J x = \begin{pmatrix} x \star \phi_{2^J} \\ |x \star \psi_{\lambda_1}| \star \phi_{2^J} \\ ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi_{2^J} \\ |||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \psi_{\lambda_3}| \star \phi_{2^J} \\ \vdots \end{pmatrix}_{\lambda_1, \lambda_2, \lambda_1, \ldots} = \cdots |W_3||W_2||W_1|x$$

Lemma: $\|W_k D_\tau - D_\tau W - k\| \leq C \|\nabla \tau\|_\infty$ where $D_\tau x(u) = x(u - \tau(u))$

Theorem (Mallat et al.)

For appropriate wavelets, a scattering is contractive

$$\|S_J x - S_J y\| \leq \|x - y\|,$$

translations invariance and deformation stability:

$$\lim_{J \to +\infty} \|S_J D_\tau x - S_J x\| \leq C \|\nabla \tau\|_\infty \|x\|$$

Credits: S. Mallat

Scattering Network

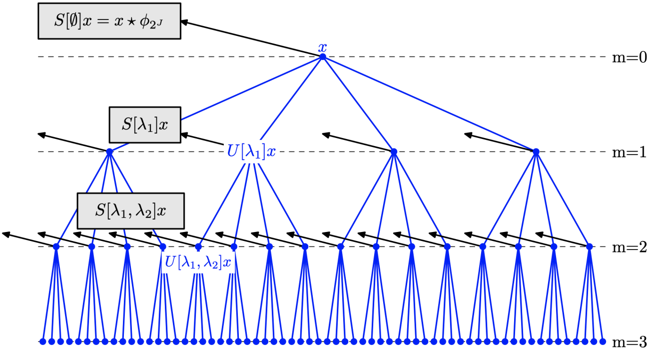


Fig. 2. A scattering propagator $\widetilde{W}$ applied to x computes the first layer of wavelet coefficients modulus $U[\lambda_1]x = |x \star \psi_{\lambda_1}|$ and outputs its local average $S[\emptyset]x = x \star \phi_{2^J}$ (black arrow). Applying $\widetilde{W}$ to the first layer signals $U[\lambda_1]x$ outputs first order scattering coefficients $S[\lambda_1] = U[\lambda_1] \star \phi_{2^J}$ (black arrows) and computes the propagated signal $U[\lambda_1, \lambda_2]x$ of the second layer. Applying $\widetilde{W}$ to each propagated signal $U[p]x$ outputs $S[p]x = U[p]x \star \phi_{2^J}$ (black arrows) and computes a next layer of propagated signals.

Credits: S. Mallat

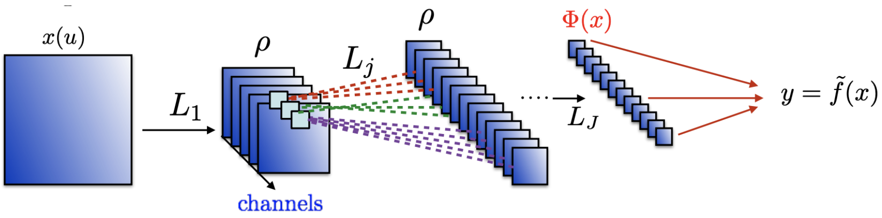
Understanding deep convolutional networks

Simplified architecture: Deep Convolutional Trees

Architecture

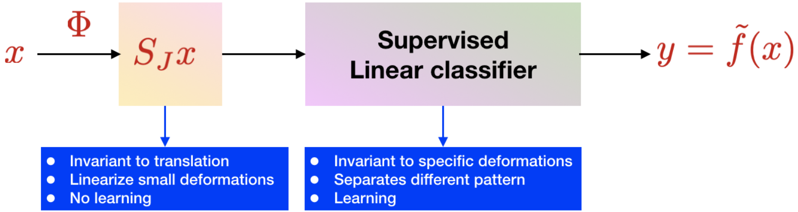
- Convolutional filters L_j: band-limited wavelets
- Pooling: L^1 norm as averaging
- Nonlinear activation ρ: modulus

$$\Phi(x) = S_J x \text{ (scattering vector)}$$



Credits: S. Mallat

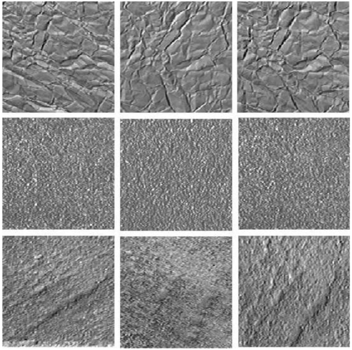
Experiments and results



- MNIST dataset for **digit classification**: for a training of 50,000 digits the classification error of the Scattering Network was similar to the Convolutional Network's (0.4 %)

Experiments and results

- CUReT dataset for **textures classification**: for a small training set of textures 200×200 in 61 classes (46 per class), the classification error with the Scattering Network achieves 0.2 %, far better than Fourier transform's one (1 %)



Experiments and results

Scattering coefficients

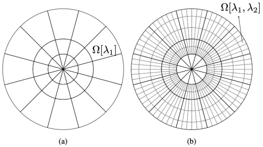


Fig. 3. To display scattering coefficients, the disk covering the image frequency support is partitioned into sectors $\Omega[p]$, which depend upon the path p. (a): For $m = 1$, each $\Omega[\lambda_1]$ is a sector rotated by r_1 which approximates the frequency support of $\hat{\psi}_{\lambda_1}$. (b): For $m = 2$, all $\Omega[\lambda_1, \lambda_2]$ are obtained by subdividing each $\Omega[\lambda_1]$.

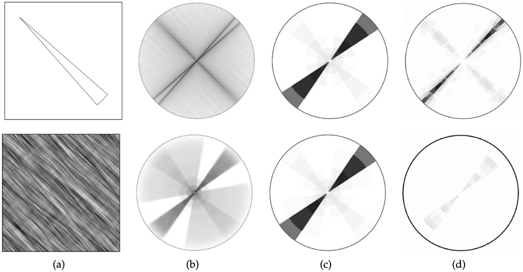
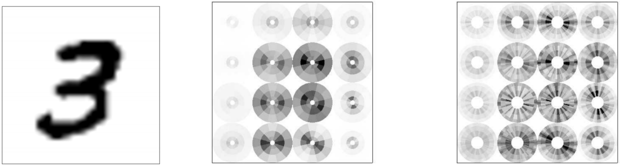


Fig. 4. (a) Two images $x(u)$. (b) Fourier modulus $|\hat{x}(\omega)|$. (c) First order scattering coefficients $Sx[\lambda_1]$ displayed over the frequency sectors of Figure 3(a). They are the same for both images. (d) Second order scattering coefficients $Sx[\lambda_1, \lambda_2]$ over the frequency sectors of Figure 3(b). They are different for each image.

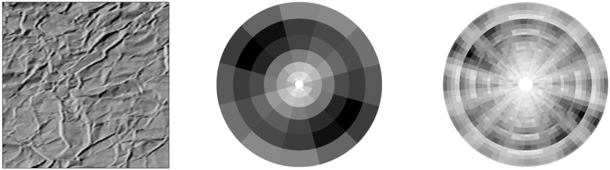
Experiments and results

Scattering coefficients



(a) (b) (c)

Fig. 7. (a): Image $X(u)$ of a digit '3'. (b): Arrays of windowed scattering coefficients $S[p]X(u)$ of order $m = 1$, with u sampled at intervals of $2^J = 8$ pixels. (c): Windowed scattering coefficients $S[p]X(u)$ of order $m = 2$.



(a) (b) (c)

Figure 4.3: (a): Example of CureT texture $X(u)$. (b): Scattering coefficients $S_J[p]X$, for $m = 1$ and 2^J equal to the image width. (c): Scattering coefficients $S_J[p]X(u)$, for $m = 2$.

Experiments and results

Scattering coefficients

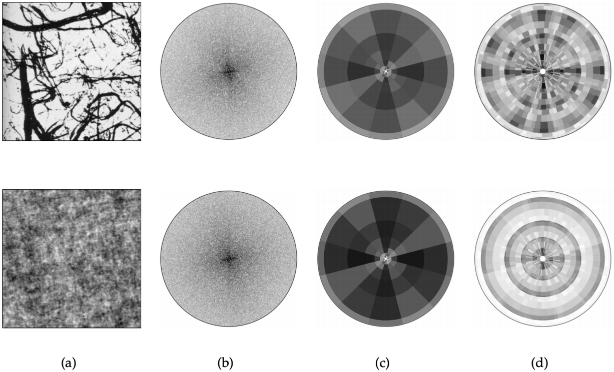


Fig. 5. (a) Realizations of two stationary processes $X(u)$. Top: Brodatz texture. Bottom: Gaussian process. (b) The power spectrum estimated from each realization is nearly the same. (c) First order scattering coefficients $S[p]X$ are nearly the same, for 2^J equal to the image width. (d) Second order scattering coefficients $S[p]X$ are clearly different.

Take home message

Interpretation of convolutional networks

- Deep convolutional network are really efficients to approximate functions in very high dimension
- Compute multiscale **invariants** of complex symmetries and learn sparse patterns
- Many mathematical questions still open (notion of regularity, complexity, approximation theorems, ...)

References

- J. Bruna & S. Mallat, Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence (2013)
- S. Mallat, Understanding deep convolutional networks. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences (2016)