The wavelet scattering network for classification

Introduction to wavelet scattering networks

Hubert Leterme

Introduction

Introduction

- One of the main tasks of image classification is to measure the level of similarity between two images.
- This is usually done by computing a representation for each image and then comparing the outputs, rather than directly comparing the original images.
- In order to build relevant classifiers, the representations must be stable under certain types of variability: translation, scaling, noise, deformations...
- In this presentation, we will show how to build such a representation using a wavelet scattering network, and highlight similarities with convolution neural networks.

Seeking invariant or stable representations

Notations

- Let $\mathscr{I} \subset L^2(\mathbb{R}^2)$ denote a set of input images and $\mathscr{J} \subset L^2(\mathbb{R}^2)$ a set of output image representations.
  For any $x \in \mathscr{I}$ and $u = (u_1, u_2) \in \mathbb{R}^2$, $x(u)$ is the gray-scaled value corresponding to the position u.
  N.B.: x is null outside a certain frame.
- Let $\Phi : \mathscr{I} \to \mathscr{J}$ denote a transformation which, for any $x \in \mathscr{I}$, outputs a representation of x denoted Φx.
  Φx can have its values in $\mathbb{R}$ or in $\mathbb{C}$ (complex representations).

Different types of variability

Global translations

For any image $x \in \mathscr{I}$ and any point $c = (c_1, c_2) \in \mathbb{R}^2$, let's denote x_c the translated image with respect to c, i.e.:

$$\forall u \in \mathbb{R}^2, x_c(u) = x(u - c)$$

We want: $\Phi x_c = \Phi x$.

Different types of variability

Additive noise

For any image $x \in \mathscr{I}$ and any noise $\varepsilon \in \mathscr{I}$,
let's denote x' such that:

$$\forall u \in \mathbb{R}^2, x'(u) = x(u) + \varepsilon(u)$$

We want Φ to be stable to additive noise:
There exists $C > 0$ such that:

$$\forall x, x' \in \mathscr{I}, \|\Phi x' - \Phi x\| \leq C\|x' - x\|$$

(Lipschitz continuity to additive noise)
N.B.: $\|.\|$ denotes the L^2-norm on $\mathscr{I}$ and $\mathscr{J}$.

Different types of variability

Deformations

For any image $x \in \mathscr{I}$ and any displacement field $\tau : \mathbb{R}^2 \to \mathbb{R}^2$, let's denote x_τ the deformed image with respect to τ, i.e.:

$$\forall u \in \mathbb{R}^2, x_\tau(u) = x(u - \tau(u))$$

We want Φ to be stable to deformations:

There exists $C > 0$ such that for any $x \in \mathscr{I}$ and any $\tau : \mathbb{R}^2 \to \mathbb{R}^2$:

$$\|\Phi x_\tau - \Phi x\| \leq C \times \|x\| \times \sup_u |\nabla \tau(u)|$$

(Lipschitz continuity to deformations)

Different types of variability

Scaling: a special kind of deformation

For any image $x \in \mathscr{I}$ and any scale factor $\alpha \in \mathbb{R}_+^*$, let's denote x_α the scaled image with respect to α, i.e.:

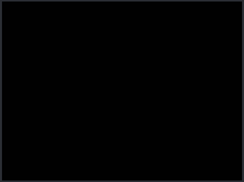
$$\forall u \in \mathbb{R}^2, x_\alpha(u) = x(\alpha^{-1}u)$$

Different types of variability

Why stable and not invariant representations?

Additive noise

$\Downarrow$



Deformations

$\Downarrow$

The wavelet transform

Among other representations

Representation	Formula	T	N	D
Canonical	$\Phi x(u) = x(u - a(x))$	✓	?	X
Fourier	$\Phi x(\omega) = \left\lvert \iint_{\mathbb{R}^2} x(v) e^{-2i\pi\langle\omega,v\rangle} dv \right\rvert$	✓	✓	X
Autocorrelation	$\Phi x(u) = \iint_{\mathbb{R}^2} x(v) x(v-u) dv$	✓	✓	X
Wavelet	$\Phi_\lambda x(u) = \iint_{\mathbb{R}^2} x(v) \overline{\psi_\lambda(u-v)} dv$	X	✓	✓

- Under certain conditions, the wavelet transform is stable to deformations, unlike the others. For instance, the Fourier transform becomes unstable at high frequencies, even for small deformations.
- Problem: the wavelet transform isn't translation-invariant. This can be solved by introducing some non-linear transformation.

The wavelet transform

Definition

A function $\psi \in L^1(\mathbb{R}^2) \cap L^2(\mathbb{R}^2)$ is a wavelet ("small wave") if it satisfies the *admissibility condition*:

$$C_\psi = \iint_{\mathbb{R}^2} \frac{\left|\widehat{\psi}(\omega)\right|^2}{|\omega|} d\omega < +\infty$$

where $\widehat{\psi}$ is the 2-dimensional Fourier transform of ψ.

- Important property:

$$\iint_{\mathbb{R}^2} \psi(u) du = 0$$

- Example: complex Morlet wavelet:

$$\psi(u) = \alpha(e^{iu.\xi} - \beta)e^{-|u|^2/(2\sigma^2)}$$

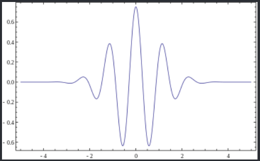


Figure: 1D real Morlet wavelet (source: Wikipedia)

The wavelet transform

Dilated and rotated wavelets

From a primary wavelet ψ as previously defined, we will build a family of wavelets that are rotated and dilated.

Let G be a group of rotations in $\mathbb{R}^2$. Then, for any $\lambda = 2^{-j}r$ such that $r \in G$ (2D-rotation matrix) and $j \in \mathbb{Z}$ (2^{-j} is a dilating coefficient), we denote:

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

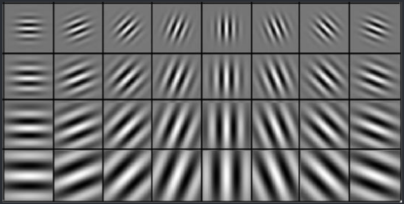


Figure: Dilated and rotated wavelets (source: Naver Labs)

The wavelet transform

Application to image processing

Let $x \in \mathscr{I}$. For any parameter λ such as described before, we define the wavelet transform $\Phi_\lambda x$ which is the *convolution product* of x and ψ_λ:

$$\Phi_\lambda x(u) = \iint_{\mathbb{R}^2} x(v)\overline{\psi_\lambda(u-v)}dv = (x * \psi_\lambda)(u)$$

- One can show that this representation is stable and invertible if the wavelet filters $\widehat{\psi}(\omega)$ cover the whole frequency plane (see next slide).
- Since the wavelet quickly decreases to 0, in the output representation, the value at each point is only influenced by its vicinity. For this reason, the wavelet transform is not translation-invariant.

The wavelet transform

The wavelets in the Fourier domain

This figure displays the support of dilated (3 dilation factors) and rotated (12 rotations) wavelets in the Fourier domain.
The wavelet transforms act like band-pass filters.
N.B.: The smaller the support, the more dilated the wavelet ("uncertainty principle").

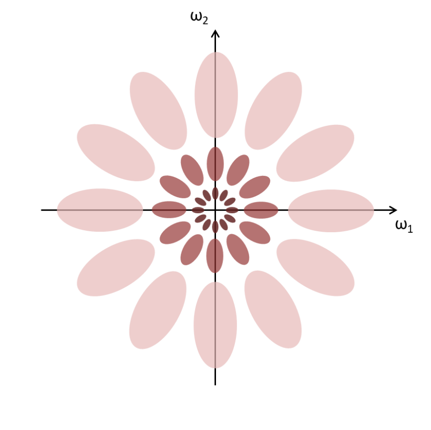


Figure: Support of $\widehat{\psi}_{2^{-j},r}(\omega)$

The wavelet transform

Illustration of a convolution product

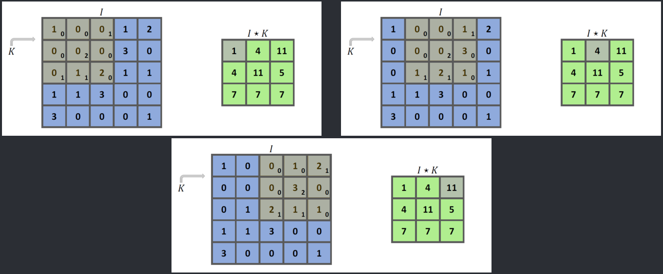


Figure: Discrete convolution product (source: Naver Labs)

In this example, the blue matrix I can be seen as the input image and the grey matrix K can be seen as the wavelet.

Introducing non-linearity

Requirements

If an operator Q (not necessarily linear) commutes with translations, we can prove that $\left(x \longmapsto \int Qx(u)du\right)$ is translation-invariant.
Since the wavelet transform commutes with translation, we could use this result to build a translation-invariant operator. Unfortunately in that case, $\int \Phi_\lambda x(u)du$ always outputs 0, which is totally useless for classification.

$\Rightarrow$ The idea is to seek a non-linear operator M such that:

- $M \circ \Phi_\lambda$ also commutes with translations
- $x \longmapsto \int (M \circ \Phi_\lambda)x(u)du$ is stable to deformations and additive noise
- M preserves the signal energy

Introducing non-linearity

The L^1 norm

In order to meet these requirements, we can choose M equal to the modulus operator. For any $x \in \mathscr{I}$, $u \in \mathbb{R}^2$:

$$(M \circ \Phi_\lambda)x(u) = M(x * \psi_\lambda)(u) = |x * \psi_\lambda(u)|$$

By integrating this quantity, we get the L^1-norm:

$$\|x * \psi_\lambda\|_1 = \int |x * \psi_\lambda(u)| du$$

$\Rightarrow$ The L^1-norm of a wavelet transform is a representation which is translation-invariant and stable to additive noise and deformations.

Toward a convolution network

Definition

$\Longrightarrow$ Sebastian's presentation

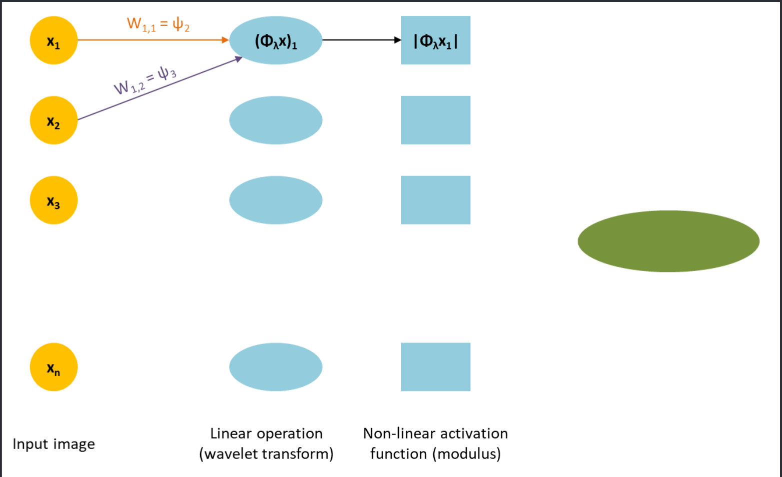
The wavelet scattering network

First convolution

In the previous section, we have implicitly built a convolutional neural network with 2 layers (providing images are discretized). Given a parameter λ, we have the following CNN:

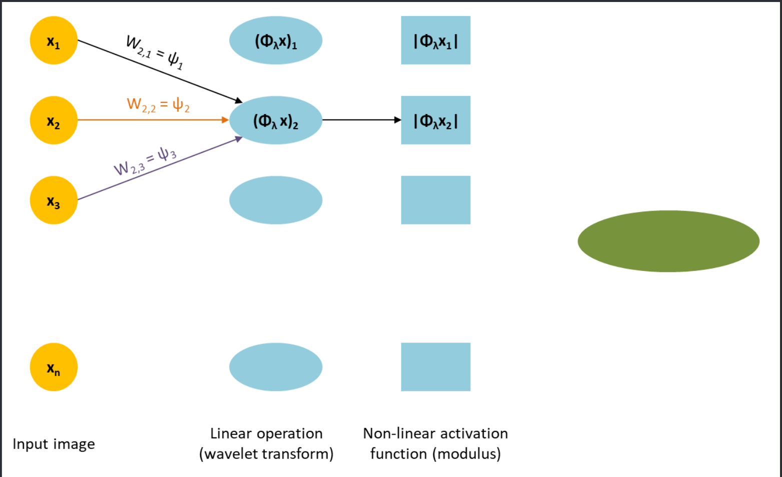
The wavelet scattering network

First convolution



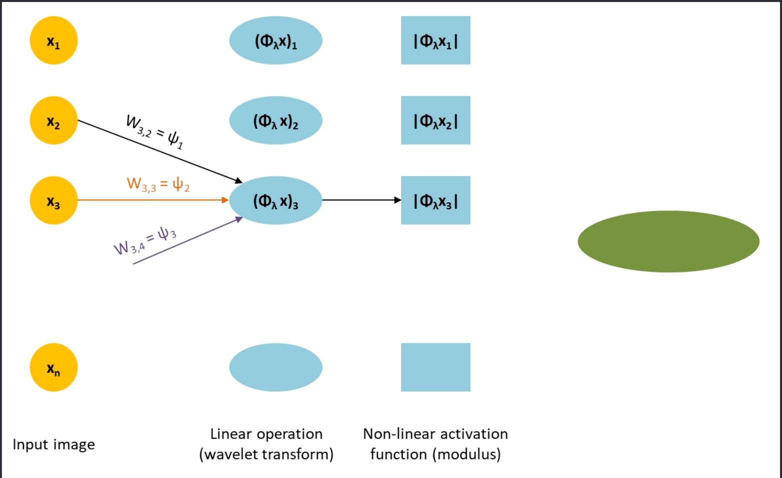
The wavelet scattering network

First convolution



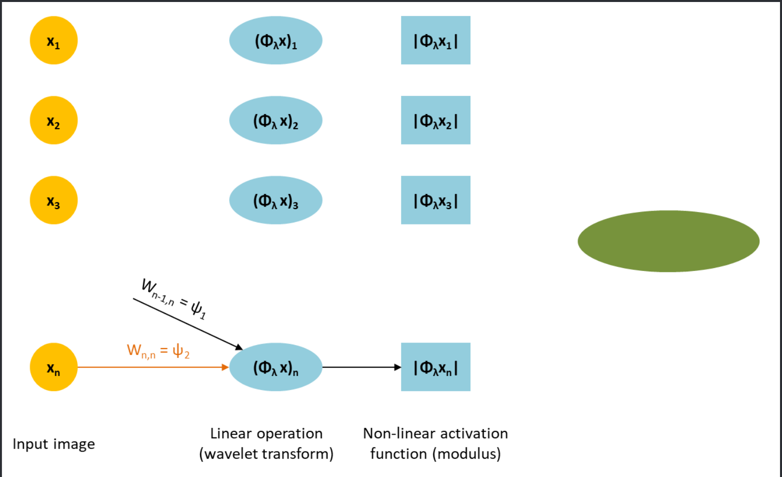
The wavelet scattering network

First convolution



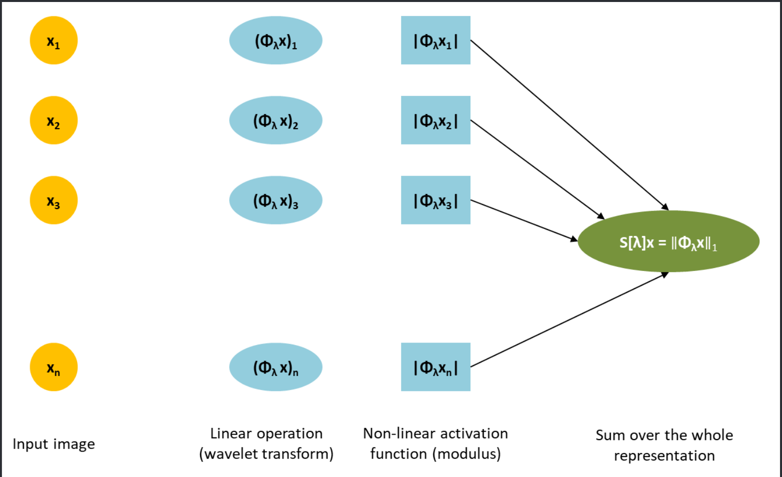
The wavelet scattering network

First convolution



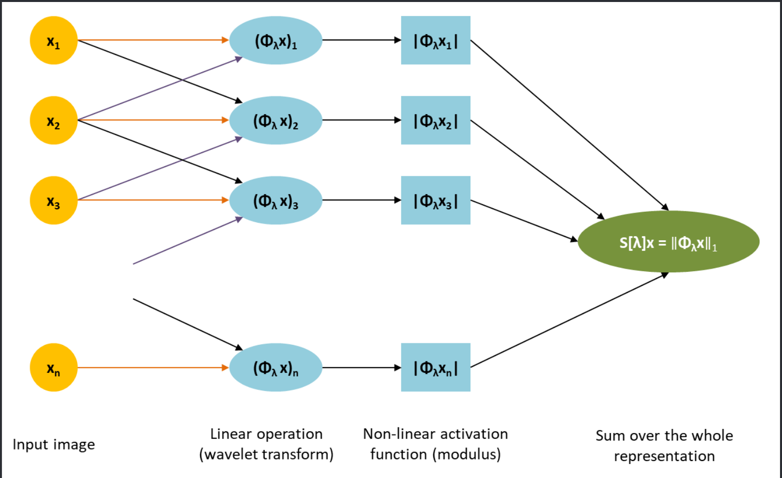
The wavelet scattering network

First convolution



The wavelet scattering network

First convolution



Building a wavelet scattering network

- Depth = 1: we can compute as many invariants as parameters $\lambda = 2^{-j}r$, with $j \in \mathbb{Z}$ and r being rotations of angles $2k\pi/K$, with $0 \leq k < K$:

$$x \longrightarrow \underbrace{|x * \psi_{\lambda_1}|}_{\textit{representation}} \longrightarrow \underbrace{\||x * \psi_{\lambda_1}|\|_1}_{\textit{invariant}}$$

- Depth > 1: we can iterate the above procedure to build a deep convolutional neural network and compute powerful invariants:

$$\begin{aligned} x \longrightarrow |x * \psi_{\lambda_1}| &\longrightarrow ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| \\ &\longrightarrow ||||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \ldots| * \psi_{\lambda_m}| \\ &\longrightarrow \text{something comparable to } \|.\|_1 \end{aligned} \tag{1}$$

Building a wavelet scattering network

- For any depth m, we can define a path: $p = (\lambda_1, \lambda_2, ..., \lambda_m)$, along which iterative (non-linear) modulus wavelet transforms are computed:

$$\begin{aligned} U[p]x &= ||||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * ...| * \psi_{\lambda_m}| \\ &= U[\lambda_m]...U[\lambda_2]U[\lambda_1]x \end{aligned} \tag{2}$$

- We compute a wavelet scattering transform along the path p, which can be compared to the $L1$-norm in the case of depth = 1:

$$\overline{S}x(p) = \mu_p^{-1} \int U[p]x(u)du$$

with $\mu_p = \int U[p]\delta(u)du$ (normalization factor)

Building a wavelet scattering network

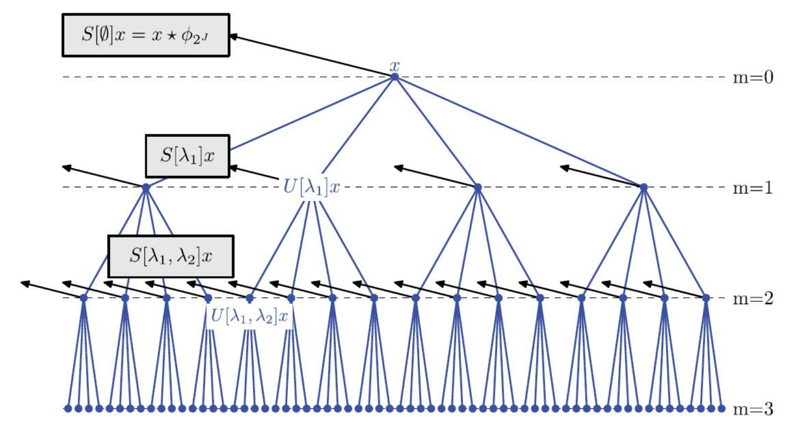


Figure: Representation of a wavelet scattering network with a depth equal to $m = 3$.
In this example, $1 + 4 + 16 + 64 = 85$ invariants are computed.
Source: S. Mallat

Scattering properties