Roto-translation scattering for texture classification.

Joakim Anden, Joan Bruna, Stéphane Mallat, Laurent Sifre, Irene Waldspurger
Understanding deep network is hard

Krizhevsky’s net
Understanding deep network is hard

Krizhevsky's net

You there! Yes you!
Neuron 6x3x5 in layer 4:
Understanding deep network is hard

Krizhevsky’s net

You there! Yes you!

Neuron 6x3x5 in layer 4:

• What exactly are you trying to do?
• What should I consider as good or bad behavior?
• If it is bad, how can I fix you (and your coworkers)?
Scattering deep network
Scattering deep network

- Deep network where the weights are handcrafted: we know exactly who does what.
Scattering deep network

• Deep network where the weights are handcrafted: we know exactly who does what.

• Invariance to transformation group (e.g. translation, rotation).
Scattering deep network

- Deep network where the weights are handcrafted: we know exactly who does what.

- **Invariance to transformation group** (e.g. translation, rotation).

- Complementary **information preservation**.
Scattering deep network

- Deep network where the weights are handcrafted: we know exactly who does what.

- **Invariance to transformation group** (e.g. translation, rotation).

- Complementary **information preservation**.

- **State-of-the-art classification** results on particular tasks (e.g. MNIST, texture).
Scattering deep network

- Deep network where the weights are handcrafted: we know exactly who does what.

- **Invariance to transformation group** (e.g. translation, rotation).

- Complementary information preservation.

- **State-of-the-art classification** results on particular tasks (e.g. MNIST, texture).

- Understanding scattering may **help you tune deep networks**.
Texture classification

25 classes

40 images per class

UIUC Texture Dataset
Translation

\[ x(u) \quad x(u - v) \]
Rotation

\[ x(u) \quad \text{and} \quad x(r_\theta u) \]
Scaling

\[ x(u) \]

\[ x(s^{-1}u) \]
Shear

\[ x(u) \quad x(r_\theta (1, s^{-1})^T r_{-\theta} u) \]
Elastic deformation

\[ x(u) \quad x(u - \tau(u)) \]
Modulus of Fourier transform

- Translation invariant
- Retains the information
Modulus of Fourier transform

- Translation invariant
- Retains the information

- Unstable to deformation
Local averaging

- Translation invariant (up to the window width)
- Stable to deformation
Local averaging

- Translation invariant (up to the window width)
- Stable to deformation
- Discards most of the information
Complex oriented and dilated Wavelets

\[ \theta \]

\[ \psi_{j,\theta} \]
Wavelet Transform

The wavelet transform decomposes an image in the averaging and wavelet coefficients:

\[
W x = (x \ast \phi, \ x \ast \psi_{j,\theta})
\]
The Wavelet Transform preserves the norm

The norm of the wavelet transform is:

$$\|Wx\|^2 = \|x \ast \phi\|^2 + \left\| \sum_{j, \theta} x \ast \psi_{j, \theta} \right\|^2$$

If the Littlewood-Paley is bounded:

$$\forall \omega, \quad 1 - \epsilon \leq \left| \hat{\phi}(\omega) \right|^2 + \sum_{j, \theta} \left| \hat{\psi}_{j, \theta}(\omega) \right|^2 \leq 1$$

Then the wavelet transform almost preserves the norm

$$(1 - \epsilon)\|x\|^2 \leq \|Wx\|^2 \leq \|x\|^2$$

Littlewood–Paley function for the family $$(\phi, \psi_{j, \theta})_{j, \theta}$$
The Wavelet Transform is invertible

One can reconstruct the original image from its wavelet transform:

$$x = x \ast \phi \ast \tilde{\phi} + \sum_{j, \theta} x \ast \psi_{j, \theta} \ast \tilde{\psi}_{j, \theta}$$
Wavelet Transform is stable to deformation
The Wavelet Transform is not invariant to translation

If you translate the image:

- The wavelet coefficients are also translated.
- Their amplitude stays the same.

Covariance
<table>
<thead>
<tr>
<th></th>
<th>Fourier Modulus</th>
<th>Local Averaging</th>
<th>Wavelet Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation invariance</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Retain information</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Stable to deformation</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Scattering combines averaging and wavelet transform and offers the **best of both**.
Wavelet Modulus

\[ x \star \psi_{j, \theta} \]

covariant

\[ x \star \phi \]

invariant
Wavelet Modulus

\[ x \xrightarrow{\text{covariant}} x \ast \psi_{j,\theta} \]

\[ x \ast \phi \quad \text{invariant} \]

\[ \text{invariant...} \]
Wavelet Modulus

\[ x \rightarrow x \ast \psi_{j, \theta} \]

invariant

covariant

\[ x \ast \phi \rightarrow 0 \]

invariant…

but not very useful
Wavelet Modulus

\[ x \star \psi_{j, \theta} \]

invariant

covariant
Wavelet Modulus

\[ x \rightarrow x \ast \psi_{j,\theta} \rightarrow |x \ast \psi_{j,\theta}| \]

covariant

\[ x \ast \phi \]

invariant
Wavelet Modulus

\[ x \rightarrow x \ast \psi_{j,\theta} \rightarrow |x \ast \psi_{j,\theta}| \]

\[ x \ast \phi \rightarrow |x \ast \psi_{j,\theta}| \ast \phi \]
Wavelet Modulus

\[ x \rightarrow x * \psi_{j,\theta} \rightarrow |x * \psi_{j,\theta}| \rightarrow |x * \psi_{j,\theta}| * \phi \]

- Covariant
- Non-linear
- Invariant
Wavelet Modulus

\[ x \xrightarrow{\star \phi} x \star \psi_{j,\theta} \xrightarrow{|\cdot|} |x \star \psi_{j,\theta}| \star \phi \]

- Covariant
- Non linear
- Invariant

SIFT

Image gradients

Keypoint descriptor
Why Modulus?

- Has a regularizing effect for complex analytic filters.
- Preserves the norm.
Scattering of order 1

\( x \)
Scattering of order 1

\[ x \xrightarrow{S_0} S_0x \]
Scattering of order 1

$x \rightarrow S_0 x \rightarrow |.| \rightarrow U_1 x$
Scattering of order 1

\[ x \rightarrow S_0 x \rightarrow S_1 x \rightarrow U_1 x \]
Scattering of order 2

\[ U_1 x \]

\[ S_1 x \]
Scattering of order 2

\[ S_1 x \]

\[ U_1 x \]

\[ U_2 x \]
Scattering of order 2

$S_1 x$ \rightarrow $U_1 x$ \rightarrow $\theta_2$ \rightarrow $\theta_1$ \rightarrow $S_2 x$
Translation Scattering

- Scattering of order 0: local average
  \[ S_0 x = x \ast \phi \]
- Scattering of order 1: SIFT-like
  \[ S_1 x(j, \theta) = |x \ast \psi_{j, \theta}| \ast \phi \]
- Scattering of order 2: deep networks
  \[ S_2 x(j_1, \theta_1, j_2, \theta_2) = ||x \ast \psi_{j_1, \theta_1}\ast \psi_{j_2, \theta_2}| \ast \phi \]
- Scattering vector: concatenation of all orders
  \[ Sx = (S_0 x, S_1 x, S_2 x, ... ) \]
Translation Scattering properties

- Almost preserves the norm.
- Stable to deformation.
- Invariant to translation

Good representation for classification when there is not significant rotation, scaling etc…
MNIST classification with Translation Scattering (Joan Bruna)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>5.6</td>
<td>7.18</td>
</tr>
<tr>
<td>1k</td>
<td>2.6</td>
<td>3.21</td>
</tr>
<tr>
<td>10k</td>
<td>1</td>
<td>0.85</td>
</tr>
<tr>
<td>60k</td>
<td>0.43</td>
<td>0.53</td>
</tr>
</tbody>
</table>

[1] Invariant Scattering Convolution Networks, PAMI AUG2013
Joan Bruna, Stephane Mallat

[2] Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition, CVPR07
Marc’Aurelio Ranzato, Fu-Jie Huang, Y-Lan Boureau, Yann LeCun
Rotation and translation
Scattering

For texture, we need a representation that is:
• invariant to translation and rotation (and more)
• does not lose too much information
• stable to deformation

This is the same problem as previously but with a different group.

Let’s apply the same solution on that group.
Roto-translation group

Translation \( v \in \mathbb{R}^2 \)

Rotation \( \theta \in SO(2) \)

Roto-translation \( g = (v, \theta) \in \mathbb{R}^2 \times SO(2) \)

A roto-translation can act on an image position \( u \):
\[
gu = v + r_{\theta} u
\]

The product of two roto-translations \( g \) and \( h = (v', \theta') \) is:
\[
gh = (v + r_{\theta} v', \theta + \theta')
\]

Non-commutative
Fixing scattering for rotation invariance
Fixing scattering for rotation invariance

\[ S_0 x \]
Fixing scattering for rotation invariance

This kernel happens to be rotation invariant. Don’t touch it.

\[ S_0 x \]
Fixing scattering for rotation invariance

This kernel happens to be rotation invariant. Don’t touch it.
Fixing scattering for rotation invariance

This kernel happens to be rotation invariant. Don’t touch it.

Those clearly are not.
Fixing scattering for rotation invariance

This kernel happens to be rotation invariant. Don’t touch it.

Those clearly are not.

We need to process $U_1x$ along position and orientation.

$S_0x$
Orbit extraction

We extract 3D signals called « orbits »

Spatial position  \( v = (v_1, v_2) \)
Roto-translation convolution

Generic group convolution:

\[
Y \odot Z(g) = \sum_{h \in G} Y(h) Z(h^{-1}g)
\]

Translation convolution is a particular case!

\[
x \ast \phi(u) = \sum_{v} x(v) \phi(u - v)
\]

Roto-translation convolution:

\[
Y \odot Z(u, \theta) = \sum_{u', \theta'} Y(u', \theta') Z(r_{-\theta'}(u - u'), \theta - \theta')
\]
How do you make 2d wavelets from 1d wavelets?
How do you make 2d wavelets from 1d wavelets?

\[ \Phi(u_1, u_2) = \phi(u_1)\phi(u_2) \]

\[ \Psi^{(1)}(u_1, u_2) = \psi(u_1)\phi(u_2) \]

\[ \Psi^{(2)}(u_1, u_2) = \phi(u_1)\psi(u_2) \]

\[ \Psi^{(3)}(u_1, u_2) = \psi(u_1)\psi(u_2) \]

Separable Products!
Roto-translation separable wavelets

Separable wavelets along position and orientations:

\[
\Phi_J(u, \theta) = \phi(u)\overline{\phi}(\theta)
\]

\[
\Psi_{j_2, \theta_2, 0}(u, \theta) = \psi_{j_2, \theta_2}(u)\overline{\phi}(\theta)
\]

\[
\Psi_{0,0,k_2}(u, \theta) = \phi(u)\overline{\psi}_{k_2}(\theta)
\]

\[
\Psi_{j_2, \theta_2 k_2}(u, \theta) = \psi_{j_2, \theta_2}(u)\overline{\psi}_{k_2}(\theta)
\]
Fast group convolutions

\[ Y \odot Z(u, \theta) = \sum_{\theta', \theta''} Y(u', \theta') Z(r_{-\theta'}(u - u'), \theta - \theta') \]

Stupid way in \( O(#u^2#\theta^2) \)

But if the filters are separable:

\[ \psi_{j_2, \theta_2 + \theta'}(u - u') \]

\[ Y \odot \Psi_{j_2, \theta_2 k_2}(u, \theta) = \sum_{\theta'} \left( \sum_{u'} Y(u', \theta') \psi_{j_2, \theta_2}(r_{-\theta'}(u - u')) \right) \bar{\psi}_{j_2}(\theta - \theta') \]

Fast way in \( O(#uC(\theta) + #\theta C(u)) \)

\[ C(u) = \text{cost of convolution along } u \]
\[ C(\theta) = \text{cost of convolution along } \theta \]
Fast group convolutions

\[ Y \otimes \Psi_{j_2, \theta_2 k_2}(u, \theta) = \sum_{\theta'} \left( \sum_{u'} Y(u', \theta') \psi_{j_2, \theta_2}(r_{-\theta'}(u - u')) \right) \bar{\psi}_{j_2}(\theta - \theta') \]
Roto-translation Scattering
Roto-translation Scattering

- Almost preserves the norm.
- Stable to deformation.
- Invariant to translation and rotation.

Generic method for invariance. Can be applied to other groups.
What you might miss if you don’t mix orientations.

If you process orientations independently, you won’t see the difference.
UIUC Texture classification

<table>
<thead>
<tr>
<th>Training size / error rate</th>
<th>State-of-the-art [1, 2]</th>
<th>Translation Scattering</th>
<th>Roto-trans Scattering</th>
<th>+ log</th>
<th>+ scale processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7.6</td>
<td>50</td>
<td>23</td>
<td>16</td>
<td>7.7</td>
</tr>
<tr>
<td>10</td>
<td>3.0</td>
<td>35</td>
<td>10</td>
<td>6</td>
<td>2.2</td>
</tr>
<tr>
<td>20</td>
<td>1.2</td>
<td>20</td>
<td>3.3</td>
<td>1.8</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Each « dealt with » group dramatically improves results.
Use separable convolution in standard deep network

End of my internship at Google:
Replace full 3D convolutions with separable convolutions in Zeiler Fergus 2013 net.
Much less weights: seems to learn faster on ImageNet.
Thank you!

Questions?