The Shattering Transform: formalizing convolutional networks to analyze few example raw sonar data

David Weber and Naoki Saito

Department of Mathematics
University of California, Davis

December 16, 2019

Why formalize?

- Fundamental understanding

Why formalize?

- Fundamental understanding
- Decrease costs: high data and compute costs

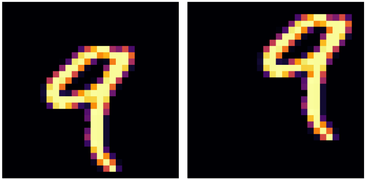
Why formalize?

- Fundamental understanding
- Decrease costs: high data and compute costs
- Avoid adversarial examples

Why formalize?

- Fundamental understanding
- Decrease costs: high data and compute costs
- Avoid adversarial examples

The fundamental claim of the scattering transform is that neural networks learn functions by finding their invariants



Convolutional Neural Networks

General Neural net $\quad a_i^j = \sigma\big(\vec{W}_i^{(j-1)} \cdot \vec{a}^{(j-1)}\big)$

Convolutional Neural net $\quad a_i^j(k) = R\Big[\sigma\big(\vec{W}_i^{(j-1)} \star \vec{a}^{(j-1)}(k)\big)\Big]$

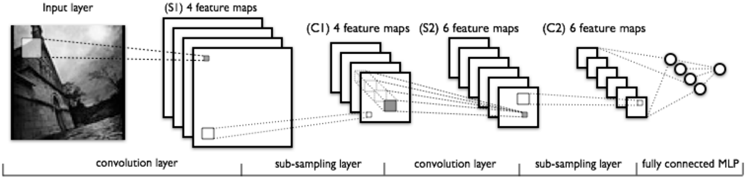


Figure: From http://deeplearning.net/tutorial/lenet.html

Here a^j is the set of coefficients in layer j, σ is a nonlinearity such as $|\cdot|$ or ReLU, and R is a subsampling operator.

Signal Invariants

How to construct a feature extractor that is invariant to task-irrelevant deformations in the data? Some examples:

Translation	$T_c[f] = f(x-c)$
Modulation	$M_\omega[f] = \mathrm{e}^{\mathrm{i}\omega t} f(x)$
Scaling	$\mathscr{S}_a[f] = f\left(x/a\right)$
Amplitude	$A_a[f] = a f(x)$

Signal Invariants

How to construct a feature extractor that is invariant to task-irrelevant deformations in the data? Some examples:

Translation	$T_c[f] = f(x-c)$
Modulation	$M_\omega[f] = \mathrm{e}^{\mathrm{i}\omega t} f(x)$
Scaling	$\mathscr{S}_a[f] = f\left(x/a\right)$
Amplitude	$A_a[f] = af(x)$

[**?**] and [**?**] demonstrated that only trivial linear features are absolutely invariant to even just translation, so they use *relative invariance* of feature extractor ρ:

$$\rho[T_c f] = \eta(c)\rho[f]$$

Signal Invariants

How to construct a feature extractor that is invariant to task-irrelevant deformations in the data? Some examples:

Translation	$T_c[f] = f(x-c)$
Modulation	$M_\omega[f] = e^{i\omega t} f(x)$
Scaling	$\mathscr{S}_a[f] = f\left(x/a\right)$
Amplitude	$A_a[f] = af(x)$

[?] and [?] demonstrated that only trivial linear features are absolutely invariant to even just translation, so they use *relative invariance* of feature extractor ρ:

$$\rho[T_c f] = \eta(c)\rho[f]$$

They establish that the only linear feature extractors $\rho[f] = \langle f, \rho\rangle$ that are relatively invariant w.r.t. both amplitude and translation deformations are Fourier-Laplace type, i.e. for some $z \in \mathbb{C}^d$

$$\int_{\mathbb{R}^d} f(x) c_1 e^{z\cdot x} \, dx$$

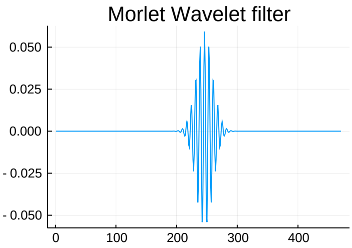
Edge detection & local features

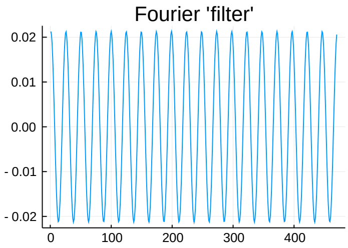
Fourier coefficients have a couple of problems:

Edge detection & local features

Fourier coefficients have a couple of problems:

- Detects global features, rather than local

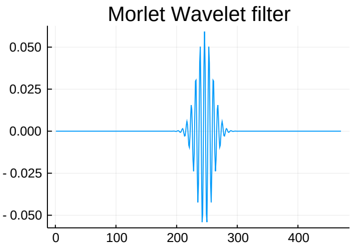


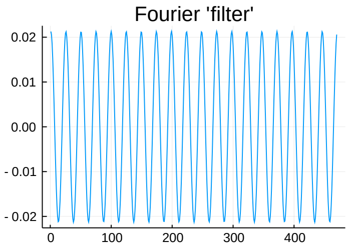


Edge detection & local features

Fourier coefficients have a couple of problems:

- Detects global features, rather than local
- Slow decay rate for signals with sharp edges, e.g. images



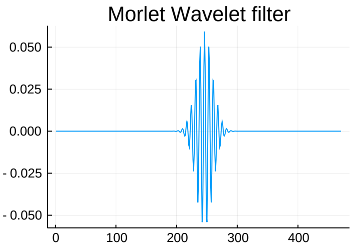


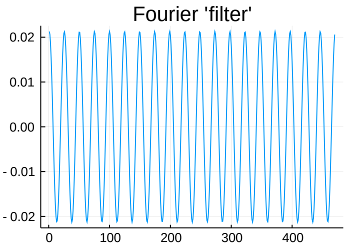
Edge detection & local features

Fourier coefficients have a couple of problems:

- Detects global features, rather than local
- Slow decay rate for signals with sharp edges, e.g. images

Loosening the translation invariance further, we have translation *covariance*: $\rho[T_c f] = T_c \rho[f]$, which implies a convolutional filter $\rho[T_c f] = g_\rho \star f$. Many examples adapted to address both of the above, e.g. Wavelets, Curvelets, Shearlets.





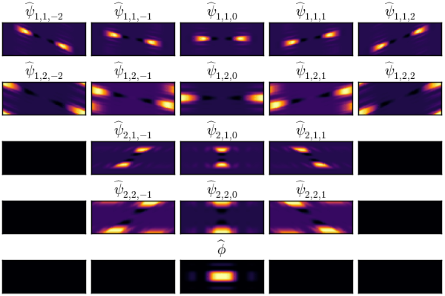
Shearlets

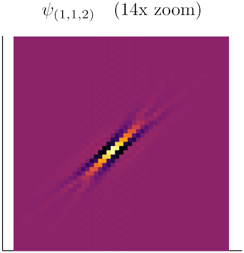
A particularly useful class of edge detectors for $\geq$2D data, the filters are indexed by shearing, scale, location and cone:

$$\mathscr{S}_1[f] = (f \star \phi, f \star \psi_{j,k})$$

$$\psi_{j,k}(x) = 2^{(2+\alpha)j/4}\psi\left(A_j^{-1} S_k^{-1} x\right)$$

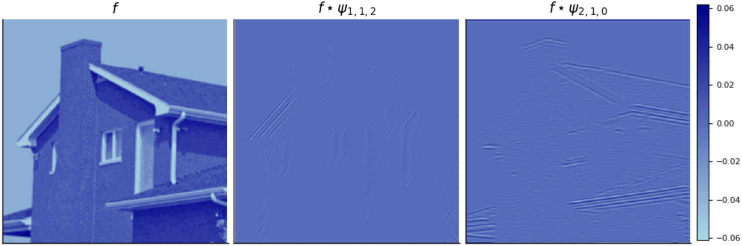
$$A_j = \begin{pmatrix} 2^j & 0 \\ 0 & 2^{j\alpha/2} \end{pmatrix} S_k = \begin{pmatrix} 1 & ck \\ 0 & 1 \end{pmatrix}$$





Shearlet Sparsity of cartoon-like images

Shearlets use few coefficients to represent cartoon-like images



Decay rate of order $\frac{\log(n)}{n^{3/2}}$, which is the optimal[1] decay rate across all recoverable linear transforms. See [?] for more details.

1. modulo the $\log n$

Generalized Scattering Transform

A single propagating layer $u[q_i]f$ of the generalized scattering transform is a vector consisting of semi-discrete shift invariant frame transforms $\psi_{\lambda_i^{(m)}} \star f$ indexed by $\lambda_i^{(m)} \in \Lambda_m$, a pointwise nonlinearity σ_m with Lipschitz constant γ_m, and a subsampling factor $r_m \geq 1$.

Generalized Scattering Transform

A single propagating layer $u[q_i]f$ of the generalized scattering transform is a vector consisting of semi-discrete shift invariant frame transforms $\psi_{\lambda_i^{(m)}} \star f$ indexed by $\lambda_i^{(m)} \in \Lambda_m$, a pointwise nonlinearity σ_m with Lipschitz constant γ_m, and a subsampling factor $r_m \geq 1$.Putting together k indices in a path $q = (\lambda_{i_m}^{(m)}, \ldots, \lambda_{i_1}^{(1)})$ gives the propagating value at layer m:

$$u[\lambda_i]f := \frac{1}{r_1^{d/2}} \sigma\big(\psi_{\lambda_i} \star f\big)(r_1 \cdot) \qquad m = 1$$

$$u[\lambda_i^{(2)}, \lambda_j^{(1)}]f := \frac{1}{r_2^{d/2}} \sigma\Big(\psi_{\lambda_i^{(2)}} \star \frac{1}{r_1^{d/2}} \sigma\big(\psi_{\lambda_j^{(1)}} \star f\big)(r_1 \cdot)\Big)(r_2 \cdot) \quad m = 2$$

And so on, until the desired depth.

Generalized Scattering Transform

A single propagating layer $u[q_i]f$ of the generalized scattering transform is a vector consisting of semi-discrete shift invariant frame transforms $\psi_{\lambda_i^{(m)}} \star f$ indexed by $\lambda_i^{(m)} \in \Lambda_m$, a pointwise nonlinearity σ_m with Lipschitz constant γ_m, and a subsampling factor $r_m \geq 1$.Putting together k indices in a path $q = (\lambda_{i_m}^{(m)}, \ldots, \lambda_{i_1}^{(1)})$ gives the propagating value at layer m:

$$u[\lambda_i]f := \frac{1}{r_1^{d/2}} \sigma\big(\psi_{\lambda_i} \star f\big)(r_1 \cdot) \qquad m = 1$$

$$u[\lambda_i^{(2)}, \lambda_j^{(1)}]f := \frac{1}{r_2^{d/2}} \sigma\Big(\psi_{\lambda_i^{(2)}} \star \frac{1}{r_1^{d/2}} \sigma\big(\psi_{\lambda_j^{(1)}} \star f\big)(r_1 \cdot)\Big)(r_2 \cdot) \quad m = 2$$

And so on, until the desired depth. The output $s^m[f]$ is taken by averaging $u[f]$ for every path q of depth m with one atom ϕ_m, and then subsampling:

$$s_m[\lambda_{i_m}^{(m)}, \ldots, \lambda_{i_1}^{(1)}]f := \phi_m \star u[\lambda_{i_m}^{(m)}, \ldots \lambda_{i_1}^{(1)}]f$$

Generalized Scattering Transform

A single propagating layer $u[q_i]f$ of the generalized scattering transform is a vector consisting of semi-discrete shift invariant frame transforms $\psi_{\lambda_i^{(m)}} \star f$ indexed by $\lambda_i^{(m)} \in \Lambda_m$, a pointwise nonlinearity σ_m with Lipschitz constant γ_m, and a subsampling factor $r_m \geq 1$.Putting together k indices in a path $q = (\lambda_{i_m}^{(m)}, \ldots, \lambda_{i_1}^{(1)})$ gives the propagating value at layer m:

$$u[\lambda_i]f := \frac{1}{r_1^{d/2}} \sigma\big(\psi_{\lambda_i} \star f\big)(r_1 \cdot) \qquad m = 1$$

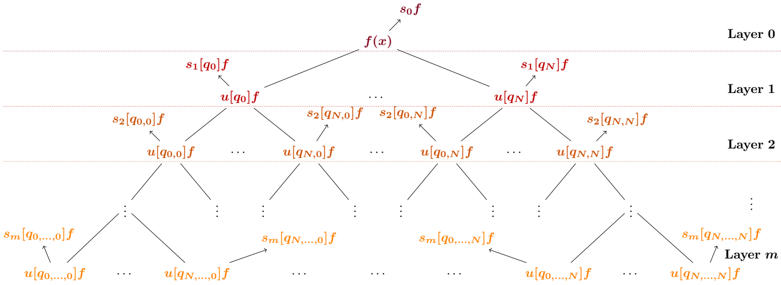
$$u[\lambda_i^{(2)}, \lambda_j^{(1)}]f := \frac{1}{r_2^{d/2}} \sigma\Big(\psi_{\lambda_i^{(2)}} \star \frac{1}{r_1^{d/2}} \sigma\big(\psi_{\lambda_j^{(1)}} \star f\big)(r_1 \cdot)\Big)(r_2 \cdot) \quad m = 2$$

And so on, until the desired depth. The output $s^m[f]$ is taken by averaging $u[f]$ for every path q of depth m with one atom ϕ_m, and then subsampling:

$$s_m[\lambda_{i_m}^{(m)}, \ldots, \lambda_{i_1}^{(1)}]f := \phi_m \star u[\lambda_{i_m}^{(m)}, \ldots \lambda_{i_1}^{(1)}]f$$

If we write s without an index, this is the collection of outputs at all layers up to some desired M: $0, 1, \ldots, M$.

Generalized Scattering Transform



The first results on Scattering transforms were from [?] and his group. A more recent generalization for systems like shearlets, relates depth m to translation:

Theorem (Translation invariance, [?])

As long as the frames have upper frame bounds b_m satisfying $\max\{b_m, \gamma_m b_m / r_m^d\} \leq 1$, the features at depth m satisfy:

$$S^m[T_c f] = T_{\frac{c}{r_1 \cdots r_{m-1}}} S^m[f]$$

Further if the output atoms satisfy $\widehat{\phi_m}|\omega| \leq K$, this implies a bound on the difference in norm:

$$\|S^m[f] - S^m[T_c f]\| \leq \frac{2\pi|c|K}{r_1 \cdots r_m} \|f\|_2$$

Previous Theory
Non-uniform Translation and Modulation

In addition to this quasi-translation invariance, this generalized scattering transform is stable under space and frequency modulations:

$$F_{\tau,\omega}[f](x) = \mathrm{e}^{\mathrm{i}\omega(x)} f\big(x - \tau(x)\big)$$

Theorem (Stability, [?])

If f is a band limited function, ω and τ are continuous, τ is once differentiable and $\|\nabla\tau\|_\infty \le \frac{1}{2d}$, there is a C independent of S so that

$$\Big\| S[f] - S\big[F_{\tau,\omega}[f]\big] \Big\|_2 \le C\|f\|_2\big(R\|\tau\|_\infty + \|\omega\|_\infty\big)$$

where the norm on S is just $\|\cdot\|_2$ on each output element

Shattering transform

using a shearlet scattering transform, in addition to the previous result, we have that the coefficients decay at the same rate as the linear coefficients:

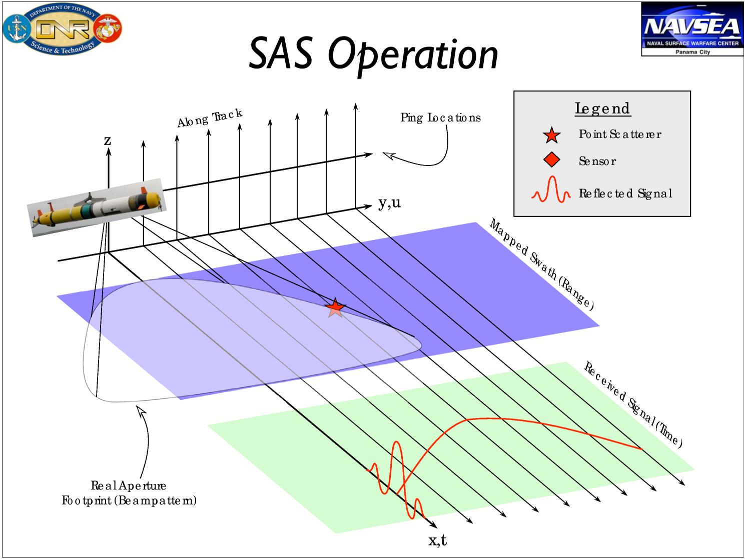
Theorem (Sparsity)

Assume that the input image f is a cartoon-like function, i.e. has the form $f = f_0 + f_1\chi_B$, where f_i is smooth, and ∂B is a C^2 function, and that σ is smooth at all but one point. If we denote the reordering of $S[f]$ by size as c_i, we have that

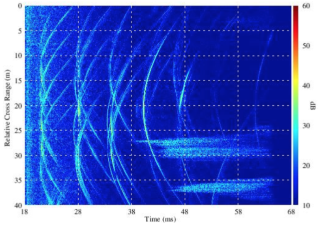
$$|c_i| \approx O\left(\frac{\log n}{n^{3/2}}\right)$$

We have implemented the Shattering Transform in Julia in the package COLLATINGTRANSFORM.JL (to be released in the next 6 months)

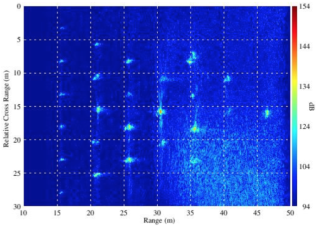
Sonar Scattering



Sonar Scattering



Original signal



Single input

1D Classifier

Consider each 2D wavefield as a set of 1D signals, perform a scattering transform using Morlet Wavelets, and use the resulting vectors as the input to a linear classifier, in our case Sparse logistic regression.

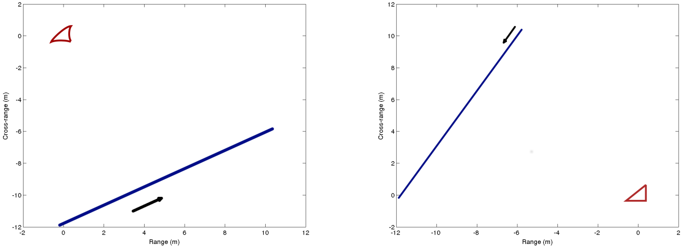


Figure: The triangle, the sharkfin and observation paths

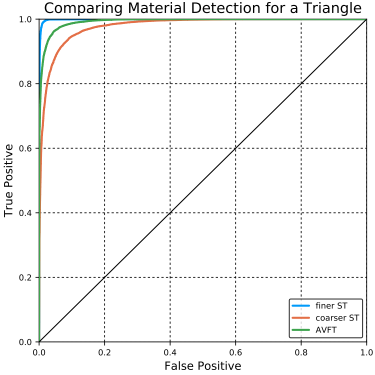


Figure: The ROC curve for detecting the material difference in a triangle, for speeds of sound $c_1 = 2000\mathrm{m/s}$ and $c_1 = 2500\mathrm{m/s}$.

Synthetic Experiments
Shape Discrimination

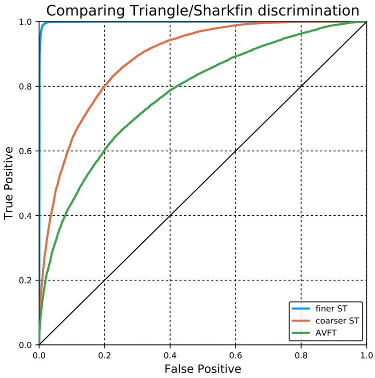


Figure: The ROC curve for discriminating a shark-fin from a triangle where both have a speed of sound fixed at $2000\mathrm{m/s}$.

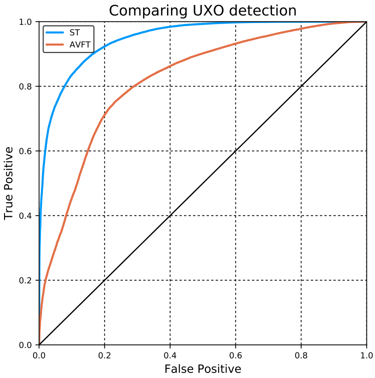


Figure: The ROC curve for detecting UXOs. The scattering transform has two layers, with quality factors $Q_1 = Q_2 = 8$

Acknowledgements

- ONR Grants: N00014-12-1-0177; N00014-16-1-2255
- Vincent Bodin (former intern with Saito, responsible for initial database code)
- Jim Bremer and Ian Sammis (UC Davis, wrote the fast Helmholtz solver)
- Bradley Marchand
- Frank Crosby, Julia Gazagnaire (NSWC-PCD, FL, real dataset)
- GLMNET (T. Hastie and his group)
  - Simon Kornblith for his Julia wrapper for the fortran GLMNET

References I

Amari, S. (1968).
Invariant structures of signal and feature space in pattern recognition problems.
RAAG Memoirs, 4(1-2):553–566.

Kutyniok, G. and Labate, D., editors (2012).
Shearlets - Multiscale Analysis for Multivariate Data.
Applied and Numerical Harmonic Analysis. Birkhäuser Basel.

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

Naoki Saito, D. W. (2017).
Underwater object classification using scattering transform of sonar signals.

References II

Otsu, O. (1973).
An invariant theory of linear functionals as linear feature extractors.
Bulletin of the Electrotechnical Laboratory, 37(10):893–913.

Wiatowski, T. and Bölcskei, H. (2015).
Deep convolutional neural networks based on semi-discrete frames.
IEEE Int. Symp. on Info. Theory, pages 1212–1216.

We use Mallat's framework with Morlet Wavelets, and compare with the absolute value of the Fourier transform (AVFT). We use two scattering transforms:

Type	Q_1	Q_2	Q_3
Finer	8	8	1
Coarser	8	4	4

- Each signal is normalized so the maximum amplitude is 1
- White Gaussian noise is added to get average SNR is about 5dB.
- Multiclass logistic regression with Lasso (via GLMNET) is used as a feature extractor and a classifier.
- Perform 10-fold cross validation, i.e., repeat the classification 10 times by randomly splitting the whole dataset into training and test sets with 50/50.

[?]