Wavelets

A gentle introduction for the wary researcher

Ben Cardoen

November 27th, 2021

Examples/Figures reused from [5, 1]

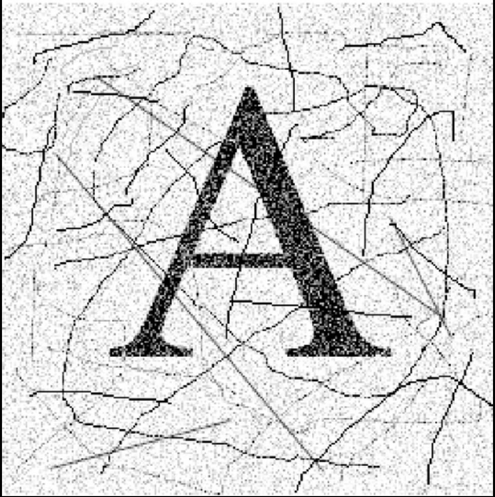
Why do you need Wavelets?

To encode, compress, detect, reconstruct a signal f(x) by another f'(x)

you need to be able to

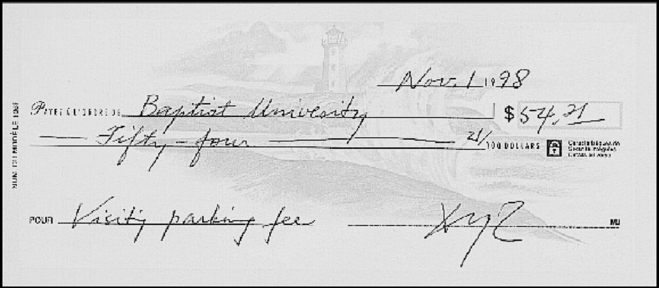
- Capture transient signal
- Have guaranteed convergence
- Converge fast, uniform (on range) ($||f'|| \rightarrow \infty \Rightarrow \; ||f, f'|| \rightarrow 0$)
- Interpretable: f' translates into usable parts (enable compression, weights location of signal, ...)
- Adaptively compute f' fast (nice to have)

Transient signals ?



Edge of letter A is a transient signal –> Capture

Transient signals ?

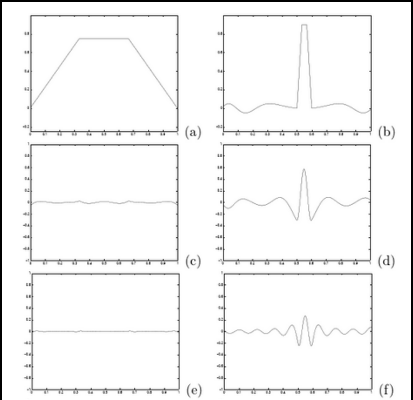


Lines of the cheque form are transient –> Filter

Let's try Fourier?

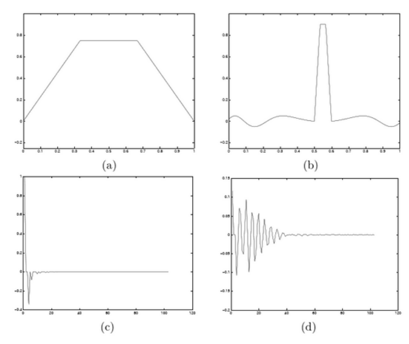
We know Fourier decomposition can approximate a signal arbitrarily close, so why need something else?

$$f'(x) = \frac{a_0}{2} + \sum_{k=1}^{N} (a_k \cos(kx) + b_k \sin(kx) \int_0^{2\pi} |f(x) - f'(x)|^2 \rightarrow 0 \text{ if } N \rightarrow \infty$$



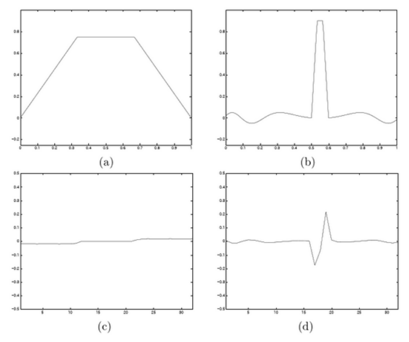
Top row: f(x). Row 2, 3 : error of Fourier at N=8, 30. The error for a transient signal is extremely large while the non-transient signal converges quickly.

Fourier decomposition model is not informative



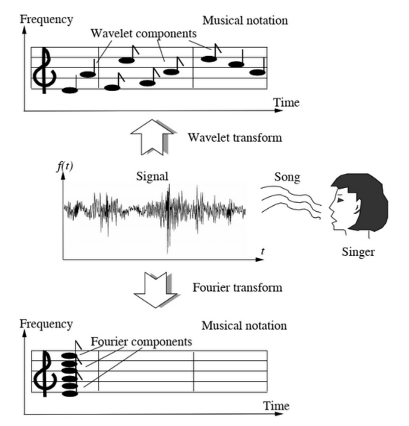
Top row: f(x). Lower row: a_i, b_i of Fourier series. Frequency based models tells us nothing about **where** in the signal the pattern of interest occurred. (learning a density function == frequency decomp.

Wavelet decomposition is informative



Top row: f(x). Bottom row : Haar coefficients. Note that the coefficient values align in velocity with the signal.

Motivation



Difference between frequency and wavelet analysis. Wavelet decomposition captures frequency AND position/location, and converges fast on transient signal.

Motivation

Why should you care ?

- Transient signals are either annoying noise (acq noise)
- Or the hard to extract signal you care about (tumor)
- Improved convergence
- Wavelet coefficients are interpretable (where in my encoding is the decisive information ?)
- Compress a signal with location sensitivity (e.g. fingerprinting, attention, ..)
- VERY fast
- ... if you have the wavelet basis

Between 1908 (first wavelet) - 1984 (MRA) it was seriously doubted useful wavelets existed.

Wavelets – what is it?

Wave

- Oscillates
- Infinite Support: $(-\infty, +\infty) \rightarrow < \infty$: wave

Wavelet

- Oscillates
- Finite Support: $f : [a, b) \rightarrow < \infty$: wavelet

Wavelet : dampened wave. Can a piecewise constant function be a wave? Why is finite support important ?

The loneliest wavelet: Haar-family

Code demo

Definition

Not all wavelets are useful Given a function f you want to learn/capture/compress/reconstruct"

Given finite energy signal space $L^2(\mathbb{R})$

- Given function $f : \mathbb{R} \to \mathbb{R} : \int_a^b |f(t)dt| < \infty \wedge -\infty < a < b < \infty$
- Let ψ be a function s.t. : $\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \quad (j, k \in \mathbb{Z})$
- then $f(x) = \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} c_{j,k}\psi_{j,k}(x)$
- If, and only if $\langle\psi_{j,k}\psi_{i,l}\rangle = 0, \quad \forall(i, j) \neq (j, k)$

If ψ exists, then any finite function on finite support can be written as an approximation of a recursive **scaling** (j) and **translation** (k).
For close to 80 years, majority of the field considered existence of such a function (except 1) as **unlikely**. Wavelets need finite/compact support to be useful. Why $\mathbb{Z}$?

Definition – missing pieces

A signal is ...

- S_n ordered string of exactly n finite real numbers $\mathbb{R}^n$
- If addition and scalar multiplication are defined, S_n is a vector space
- If inner product <a, b> is defined, S_n = inner product space.
- $S_m \subset S_n$: a subspace S_m of S_n is defined by : $s_i \in S_m$, where s_i[m..n] = 0.
- $S_{m\perp} \subset S_n$: Residual subspace of S_m s.t. $\forall s_p \in S_{m\perp}$: s_p[1..m]=0.
- $s_m \in S_m, s_p \in S_{m\perp} \Rightarrow < s_m, s_p >= 0$

Definition – missing pieces

Wavelets are part of inner product space V_n

- $\langle f, g \rangle = \int_a^b f(t)g(t)dt$
- $||f|| = \sqrt{\langle f \rangle}$
- $d(f, g) = \sqrt{\int_a^b (f(t)g(t))^2 dt}$
- Wavelet basis in V_n if all wavelets j=1..n^2 are orthonormal, e.g. $||f|| = 1$, $\langle g, f \rangle = 0$

Any finite function on finite support can be written as an approximation of a recursive scaling (j) and translation (k) function, if ψ forms an **orthonormal basis** in V_n.
Not all bases are equal(ly useful).

Orthogonal decomposition theorem

Given subspace W

- $dim(W) < \infty$, $W \subset V$, V inner product space
- $\forall v \in V, \exists w \in W, w_{\perp} \in W_{\perp} : v = w + w_{\perp}$
- Given basis $W_b = w_1, .., w_m$
- Find the w vector by $w = \sum_{i=1}^{m} \frac{\langle v, w_i \rangle}{\langle w_i, w_i \rangle} w_i$

Consequence: Multiresolution Analysis

- $V_0 \subseteq V_1 \subseteq ... V_n$
- Inner product spaces form a nested hierarchy of subspaces

Multiresolution Analysis

Multiresolution Analysis is a

- Nested subspace sequence $\ldots V_{-1} \subseteq V_0 \subseteq V_1 \subseteq \ldots V_n$
- of $L^2\mathbb{R}$ with ϕ scaling function s.t.
  - $\cup_{n\in\mathbb{Z}} V_n$ dense in $L^2\mathbb{R}$
  - $\cap_{n\in\mathbb{Z}} V_n = 0$ (separation property, n+ : interval $\rightarrow 0$
  - $f(t) \in V_n$ iff $f(2^{-n}) \in V_0$
  - $\{\phi(t-k)\}_{k\in\mathbb{Z}}$ is orthonormal basis for V_0
  - $\phi(t) = \sum_k c_k \phi(2t-k)$ (dilation equation, for Haar $c_{0,1} = 1$, rest 0.

Given sets A, B: B is **dense** in A if $\forall a\, inA$, $\exists b_1 \ldots b_n$, $b_i \in B$, s.t. $b_n rightarrow a$. In other words, B is dense if elements of B approximate a arbitrarily close.
A wavelet space of V_n is dense when it approximates a given function space when n increases.
c_k are 'refinement coefficients'

Invariant Scattering Convolution Networks [2]

A major difficulty of image classification comes from the considerable variability within image classes and the inability of Euclidean distances to measure image similarities. Part of this variability is due to rigid translations, rotations or scaling. This variability is often uninformative for classification and should thus be eliminated. In the framework of kernel classifiers, metrics are defined as a Euclidean distance applied on a representation ϕ(x) of signals x. The operator ϕ must therefore be invariant to these rigid transformations.

A wavelet scattering is thus a deep convolution network which cascades wavelet transforms and modulus operators. *Deep convolution networks which learn filters from the data have the flexibility to adapt to such variability, but learning translation in-variant filters is not necessary. A wavelet scattering transform can be used on the first two network layers, while learning the next layer filters applied to scattering coefficients.*

Invariant Scattering Convolution Networks [2]

5

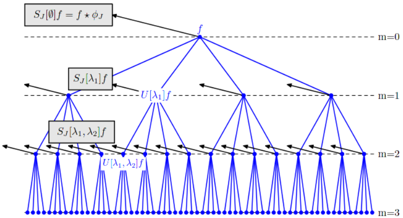


Fig. 2. A scattering propagator U_J applied to x computes each $U[\lambda_1]x = |x \star \psi_{\lambda_1}|$ and outputs $S_J[\emptyset]x = x \star \phi_{2^J}$ (black arrow). Applying U_J to each $U[\lambda_1]x$ computes all $U[\lambda_1, \lambda_2]x$ and outputs $S_J[\lambda_1] = U[\lambda_1] \star \phi_{2^J}$ (black arrows). Applying U_J iteratively to each $U[p]x$ outputs $S_J[p]x = U[p]x \star \phi_{2^J}$ (black arrows) and computes the next path layer.

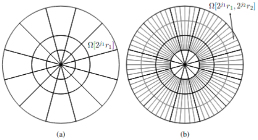


Fig. 3. For $m = 1$ and $m = 2$, a scattering is displayed as piecewise constant functions equal to $S_J[p]x(u)$ over each frequency subset $\Omega[p]$. (a): For $m = 1$, each $\Omega[2^{j_1}r_1]$ is a rotated quadrant of surface proportional to 2^{j_1}. (b): For $m = 2$, each $\Omega[2^{j_1}r_1]$ is subdivided into a partition of subsets $\Omega[2^{j_1}r_1, 2^{j_2}r_2]$.

Invariant Scattering Convolution Networks [2]

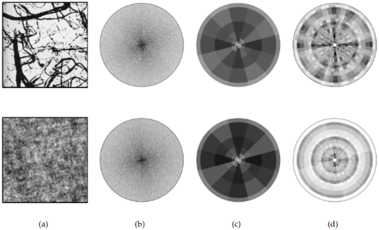


Fig. 5. Two different textures having the same Fourier power spectrum. (a) Textures $X(u)$. Top: Brodatz texture. Bottom: Gaussian process. (b) Same estimated power spectrum $\widehat{R}X(\omega)$. (c) Nearly same scattering coefficients $S_J[p]X$ for $m = 1$ and 2^J equal to the image width. (d) Different scattering coefficients $S_J[p]X$ for $m = 2$.

Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction [3]

With the rise of artificial intelligence, many recent studies attempted to use deep neural networks to extract comprehensive features from traffic networks to enhance prediction performance, given the volume and variety of traffic data has been greatly increased. Considering that traffic status on a road segment is highly influenced by the upstream/downstream segments and nearby bottlenecks in the traffic network, e**xtracting well-localized features from these neighboring segments is essential for a traffic prediction model.**

Classical wavelet transform can detect sudden changes and peaks in temporal signals. Analogously, when extending to the graph/spectral domain, graph wavelet can concentrate more on key vertices in the graph and discriminatively extract localized features. In this study, to capture the complex spatial-temporal dependencies in network-wide traffic data, we learn the traffic network as a graph and propose a **graph wavelet gated recurrent (GWGR) neural network.**

In addition, experiments show that the sparsity of graph wavelet weight matrices greatly increases the interpretability of GWGR.

Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction [3]

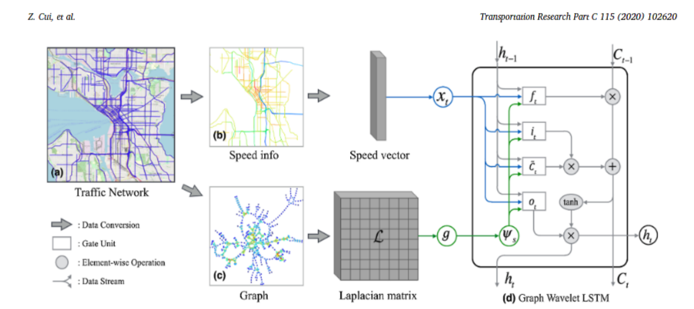


Fig. 1. Demonstration of model framework. (a) Urban traffic network in downtown Seattle. (b) Speed information of roadway segments illustrated by various colors. (c) Graph structure converted from the traffic network. (d) Structure of a graph wavelet LSTM unit at time t, in which $\mathbf{g}$ is the kernel function and Ψ_s is the graph wavelet matrix.

Wavelets on graphs via deep learning [4]

An increasing number of applications require processing of signals defined on weighted graphs. While wavelets provide a flexible tool for signal processing in the classical setting of regular domains, the existing graph wavelet constructions are less flexible—they are guided solely by the structure of the underlying graph and do not take directly into consideration the particular class of signals to be processed.

This chapter introduces a machine learning framework for constructing graph wavelets that can sparsely represent a given class of signals. Our construction uses the lifting scheme, and is based on the observation that the recurrent nature of the lifting scheme gives rise to a structure resembling a deep auto-encoder network.

The training is unsupervised, and is conducted similarly to the greedy pre-training of a stack of auto-encoders. After training is completed, we obtain a linear wavelet transform that can be applied to any graph signal in time and memory linear in the size of the graph. Improved sparsity of our wavelet transform for the test signals is confirmed via experiments both on synthetic and real data.

Graph Wavelet Neural Network [6]

For each vertex, convolution is defined as a weighted average function over all vertices located in its neigh- borhood, with the weighting function characterizing the influence exerting to the target vertex by its neighbors (Monti et al., 2017). The main challenge is to define a convolution operator that can han- dle neighborhood with different sizes and maintain the weight sharing property of CNN.—Spectral methods define convolution via graph Fourier transform and convolution theorem. Spectral methods leverage graph Fourier transform to convert signals defined in vertex domain into spectral domain, e.g., the space spanned by the eigenvectors of the graph Laplacian matrix, and then filter is defined in spectral domain, maintaining the weight sharing property of CNN.

Different from graph Fourier transform, graph wavelet transform can be obtained via a fast algorithm without requiring matrix eigendecomposition with high computational cost. Moreover, graph wavelets are sparse and localized in vertex domain, offering high efficiency and good interpretability for graph con- volution.

GWNN significantly outperforms previous spectral graph CNNs in the task of graph-based semi-supervised classification

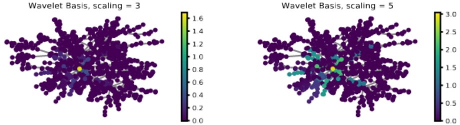
Graph Wavelet Neural Network [6]

Compared to graph Fourier transform, graph wavelet transform has the following benefits when being used to define graph convolution:

1. **High efficiency:** graph wavelets can be obtained via a fast algorithm without requiring the eigendecomposition of Laplacian matrix. In Hammond et al. (2011), a method is proposed to use Chebyshev polynomials to efficiently approximate ψ_s and ψ_s^{-1}, with the computational complexity $O(m \times |\mathbb{E}|)$, where $|\mathbb{E}|$ is the number of edges and m is the order of Chebyshev polynomials.

2. **High spareness:** the matrix ψ_s and ψ_s^{-1} are both sparse for real world networks, given that these networks are usually sparse. Therefore, graph wavelet transform is much more computationally efficient than graph Fourier transform. For example, in the Cora dataset, more than 97% elements in ψ_s^{-1} are zero while only less than 1% elements in $\boldsymbol{U}^\top$ are zero (Table 4).

3. **Localized convolution:** each wavelet corresponds to a signal on graph diffused away from a centered node, highly localized in vertex domain. As a result, the graph convolution defined in Equation (4) is localized in vertex domain. We show the localization property of graph convolution in Appendix A. It is the localization property that explains why graph wavelet transform outperforms Fourier transform in defining graph convolution and the associated tasks like graph-based semi-supervised learning.



(a) (b)

Figure 1: Wavelets on an example graph at (a) small scale and (b) large scale.

References I

[1] Edward Aboufadel and Steven Schlicker. **Discovering wavelets**. John Wiley & Sons, 2011.

[2] Joan Bruna and Stéphane Mallat. "Invariant Scattering Convolution Networks". In: **CoRR** abs/1203.1513 (2012). arXiv: 1203.1513. URL: http://arxiv.org/abs/1203.1513.

[3] Zhiyong Cui et al. "Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction". In: **Transportation Research Part C: Emerging Technologies** 115 (2020), p. 102620.

[4] Raif M Rustamov and Leonidas J Guibas. "Wavelets on graphs via deep learning". In: **Vertex-Frequency Analysis of Graph Signals**. Springer, 2019, pp. 207–222.

[5] Yuan Yan Tang. **Wavelet theory approach to pattern recognition**. Vol. 74. World Scientific, 2009.

[6] Bingbing Xu et al. "Graph wavelet neural network". In: **arXiv preprint arXiv:1904.07785** (2019).