Scattering representation

Geert Kapteijns

December 1, 2017

Abstract

Inspired by [2], I attempt to use the 3D scattering representation to classify baseline NCCT images of brain hemispheres of ischemic stroke patients into the classes "affected" and "unaffected" by stroke. I achieve 73% precision using a Gaussian radial basis function kernel SVM.

I introduce a method to discretize $SO(3)$, based on a Fibonacci lattice, to construct a filter bank that covers the frequency domain with as few filters as possible. I show that I am able to satisfy the Littlewood-Paley condition with $\epsilon = 0.92$ for filters of size $(128, 256, 128)$. Some pointers are given how this bound might be improved. Addionally, I explictly describe how to construct an appropriate mother wavelet and give an heuristic to decide on its hyperparameters. In the appendix, I discuss some steps to optimize the implementation of the scattering transform

1 Introduction

Let me start off by saying this is an informal document meant to make my research efforts of the last three months accessible. I have made no huge effort to cite the first paper to establish a concept or even to cite well-established results at all.

This research is part of a broader effort to use image data to aid radiologists in the process of deciding, in the earliest possible stage, whether to use intra-arterial therapy (a relatively new endovascular, catheter-based treatment) to treat ischemic stroke patients. I have limited my study to classifying brain hemispheres as being affected by stroke or not, thereby finding image features in 3D NCCT scans that are relevant for this decision.

I believe that, despite popularity and exceptional engineering results in many areas (easily outperforming "hand-crafted" algorithms), deep learning is not the best tool for this problem (and I suspect many other problems in the medical domain), because medical images are high-dimensional, training data is scarce and there is not always a clearly defined ground truth.

Following [2], what I have tried here is to use a different approach, namely one where no model is trained for feature extraction, but which uses an image representation called the scattering transform, that is known to have certain desirable properties, e.g. it preserves high-frequency information, is translation invariant over a tunable window and stable to deformations. This image representation may, in association with a simple classifier like an SVM, be used in classification problems.

While detecting which brain hemisphere is affected by stroke is not clinically a very relevant problem (it can, in most cases, be deducted from the symptoms of the patient) it serves as a test for the scattering representation in 3D, as at least there is an unequivocal ground truth. Though not done in this work, the learned features may be visualised to aid radiologists in giving an ASPECTS score or otherwise diagnosing the seriousness of the stroke. It is also possible, if a good delineation of the ASPECTS regions is available, to directly train a classifier on the scattering representation of these regions (though expert's opinion on whether a region is affected or not is not unequivocal, making it difficult to attain super-expert performance).

2 Scattering transform

I will in the briefest possible way state what the scattering transform is. For details refer to [3–5].

A wavelet transform is defined by convolving a d-dimensional signal $y(\boldsymbol{x})$ with scaled and rotated versions of a mother wavelet $\psi_{a^j,r}(\boldsymbol{x})$, with $j \in \mathbb{Z}$ and rotations $r \in SO(d)$ (rotation group in d dimensions). $d = 3$ in our case, since we are dealing with 3D images. $a = 2$ is common for image analysis. We will describe in the next section which mother wavelet we use in practice, and how to correctly choose a finite number of length scales and rotations r (which is not trivial in 3D).

A wavelet of dilation a^j and orientation r looks like

$$\psi_{a^j,r}(\boldsymbol{x}) = a^{-dj}\psi(a^{-j}r\boldsymbol{x}) \tag{1}$$

where the normalisation a^{-dj} is chosen such that the energy of the mother wavelet is conserved

$$\int_{\mathbb{R}^d} d\boldsymbol{x}|\psi_{a^j,r}(\boldsymbol{x})| = \int_{\mathbb{R}^d} d\boldsymbol{x}|\psi(\boldsymbol{x})|. \tag{2}$$

Translationally invariant coefficients (called scattering coefficients) of $y(\boldsymbol{x})$ that are stable to small deformations are obtained by taking the modulus and taking a spatial average:

$$\left\|y \star \psi_{a^j,r}\right\|_1 = \int d\boldsymbol{x}|y \star \psi_{a^j,r}|. \tag{3}$$

The signals $|y\star\psi_{a^j,r}|$ are themselves unstable, and in averaging (or equivalently, removing all non-zero frequencies) of $|y\star\psi_{a^j,r}|$ information is lost. To remedy this, we can perform a second wavelet transform on all first-order transforms, yielding second-order scattering coefficients

$$\left\||y \star \psi_{a^{j_1},r_1}| \star \psi_{a^{j_2},r_2}\right\| = \int d\boldsymbol{x}||y \star \psi_{a^{j_1},r_1}| \star \psi_{a^{j_2},r_2}| \tag{4}$$

for all j_1, j_2, r_1, r_2. One can keep iterating this transform to recover more lost information in the form of higher-order coefficients, but in practice two layers is sufficient for most tasks (luckily so, because the computational resources required scales exponentially in the number of layers).

To make notation easier, we define $\lambda = (j, r)$ and

$$U[\lambda_1, \ldots \lambda_m]y = |||y \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \ldots \psi_{\lambda_m}| \tag{5}$$

i.e. an unaveraged signal at the mth layer. The scattering coefficients at the mth layer are then written

$$\left\{\bar{S}y(\lambda_1, \ldots \lambda_m) = \int d\boldsymbol{x}U[p]y\right\}_{\lambda_i\in\Lambda} \tag{6}$$

In practice it is often better not to average over the entire signals $U[p]$ (where we write $p = \lambda_1, \ldots \lambda_m$ a *path*), but compute coefficients that are approximately translation invariant over lengths a^J. This is achieved by computing the wavelet transforms only at scales $j \leq J$ and averaging with a low-pass (blurring) filter (in practice a Gaussian) of support a^J, denoted by ϕ_{a^J}

$$S[p]y(\boldsymbol{x}) = (U[p] \star \phi_{a^J})(\boldsymbol{x}). \tag{7}$$

$\{S[\lambda_1, \ldots, \lambda_m]\}_{\lambda_i\in\Lambda}$ are called the windowed scattering coefficients at layer m. The correct maximum length scale a^J should be chosen based on knowledge about the input data or by cross-validation.

3 Implementation details

What follows are some considerations for choosing the mother wavelet and rotations r.

3.1 Construction of the mother wavelet

For the scattering representation to be stable to additive noise and contain all high frequency information, it must satisfy the Littlewood-Paley condition

$$(1 - \epsilon) \leq A(\boldsymbol{\omega}) \leq 1 \qquad \forall\boldsymbol{\omega} \in \mathbb{R}^d \tag{8}$$

with

$$A(\boldsymbol{\omega}) = \left|\hat{\phi}_{a^J}(\boldsymbol{\omega})\right|^2 + \frac{1}{2}\sum_{j\leq J}\sum_{r\in R}\left(\left|\hat{\psi}_{a^j,r}(\boldsymbol{\omega})\right|^2 + \left|\hat{\psi}_{a^j,r}(-\boldsymbol{\omega})\right|^2\right) \tag{9}$$

and ϵ small. Since the low-pass filter ϕ_{a^J} is normalized ($\int d\boldsymbol{x}\phi_{a^J}(\boldsymbol{x}) = 0$ or equivalently $\hat{\phi}_{a^J}(\omega = 0) = 1$), the above condition implies

$$\hat{\psi}_{a^j,r}(0) = 0 \tag{10}$$

or equivalently: all wavelets should average to zero. Furthermore, if we define a mother wavelet, called the Morlet wavelet, as follows

$$\psi(\boldsymbol{x}) = \mathcal{N} g_\sigma(\boldsymbol{x}) \left(e^{i\xi\boldsymbol{x}} - \kappa_\sigma\right) \tag{11}$$

where $g(\boldsymbol{x})_\sigma$ is a Gaussian and $\kappa_\sigma \ll 1$ has to be chosen to satisfy Equation 10, it will have the property that its Fourier transform is real, hence that if the input signal y is real, $U[j,r]y = U[j,-r]y$, allowing us to only consider positive rotations.

Instead of labeling a wavelet by its standard deviation in the spatial domain σ, it is in this case more insightful to label it by its bandwidth b in the Fourier domain, defined by

$$\hat{g}_\sigma\left(\pm\left(\frac{b}{2}, 0, 0\right)\right) = \exp\left(-\frac{1}{2}\sigma^2\left(\frac{b}{2}\right)^2\right) = \frac{1}{\sqrt{2}} \tag{12}$$

leading to

$$b^2 = \frac{4\ln 2}{\sigma^2}. \tag{13}$$

The Fourier transform of the Morlet is

$$\hat{\psi}(\boldsymbol{\omega}) = \mathcal{N}\left(\hat{g}_\sigma(\boldsymbol{\omega} - \boldsymbol{\xi}) - \kappa_\sigma \hat{g}_\sigma(\boldsymbol{\omega})\right). \tag{14}$$

It is, apart from the small factor κ_σ, centered at $\boldsymbol{\xi} = (\xi, 0, 0)$ with bandwidth b. The requirement $\hat{\psi}(\mathbf{0}) = 0$ leads to

$$\kappa_\sigma = \frac{\hat{g}_\sigma(-\boldsymbol{\xi})}{\hat{g}_\sigma(\mathbf{0})}. \tag{15}$$

The dilated and scaled wavelet becomes in the Fourier domain

$$\hat{\psi}_{a^j,r}(\boldsymbol{\omega}) = a^{-dj}\mathcal{N}\left(\hat{g}_\sigma(a^j r^{-1}\boldsymbol{\omega} - \boldsymbol{\xi}) - \kappa_\sigma \hat{g}_\sigma(a^j r^{-1}\boldsymbol{\omega})\right) \tag{16}$$

so that it is (apart from the small corrective term κ_σ) centered at frequency $\boldsymbol{\omega_c} = a^{-j} r \boldsymbol{\xi}$ with bandwidth $b_{a^j} = a^{-j}b$. Note that the corrective factor κ_σ is invariant under dilation and rotation.

We choose the normalisation factor of the mother wavelet $\mathcal{N}$, in order to be able satisfy Equation 8, as the inverse of the maximum of the "raw" Littlewood-Paley sum (which excludes the contribution of the low-pass filter)

$$\mathcal{N}^{-1} = \max_{\boldsymbol{\omega}} \frac{1}{2} \sum_{j \le J} \sum_{r \in R} \left(\left|\hat{\psi}_{a^j,r}(\boldsymbol{\omega})\right|^2 + \left|\hat{\psi}_{a^j,r}(-\boldsymbol{\omega})\right|^2 \right). \tag{17}$$

3.2 Discretizing $SO(3)$

We still need to specify which rotations we will use to build our filter bank, and, consequently, what the correct values of ξ and b are to satisfy Equation 8. Thanks to the Morlet wavelet having a real Fourier transform and our CT images being real, we only have to choose a finite number of positive rotations so that our scaled and rotated wavelets cover (are non-zero) on as much of the frequencies

$$(\omega_x, \omega_y, \omega_z) \qquad 0 < \omega_x < \pi, -\pi < \omega_y, \omega_z < \pi \tag{18}$$

as possible, where π is the Nyquist frequency in radians (equal to $N/2$ for an $N \times N \times N$ image). In practice, we can impose the additional constraint

$$\omega_x^2 + \omega_y^2 + \omega_z^2 \le \pi^2 \tag{19}$$

since we don't care so much about the highest possible frequencies in the signal (they are very small anyway).

A scaled and rotated wavelet has Fourier support around $\boldsymbol{\omega_c} = a^{-j} r \boldsymbol{\xi}$ with bandwidth $b_{a^j} = a^{-j}b$. Hence, because $\boldsymbol{\xi} \propto (1,0,0)$ and we essentially want to cover the half-ball described by Equation 18 and Equation 19, we should choose our rotations r such that they map the unit vector $(1,0,0)$ onto n points on the hemisphere with north pole $(1,0,0)$ such that the pairwise distance in $\mathbb{R}^3$ is maximal (i.e. they are "evenly spread out" across the hemisphere).

Placing our points on the corners of a platonic solid is ideal, but only works for at most 10 points (the regular dodecahedron). For a solution for general n, I have used the Fibonacci lattice, which distributes n points to maximize their distance *along the sphere* in an approximately optimal way, but this should be a good approximation to the problem at hand when to number of points is large.

3.3 Choosing ξ and b

I initially followed the reasoning in [7, Appendix A] adapted to symmetric wavelets to obtain $\xi = \frac{4\pi}{5}$ (which I used in the experiment, see below), but have since come to believe that, since we only care about frequencies described by a half-ball in Fourier space, we should require that

$$b = \pi - \xi, \tag{20}$$

$$b + \frac{b}{a} = \xi - \frac{\xi}{a}. \tag{21}$$

The first equation is obtained by requiring that frequencies up until $|\boldsymbol{\omega}| = \pi$ should be covered, whereas the second equation says that there should be no gap (in support) between the topmost wavelet, which is centered at $\boldsymbol{\xi}$ with bandwidth b, and the first scaled wavelet, which is centered at $\boldsymbol{\xi}a^{-1}$ with bandwidth ba^{-1}.

This yields

$$\xi = \frac{\pi(a+1)}{2a}, \tag{22}$$

$$b = \pi\left(1 - \frac{a+1}{2a}\right), \tag{23}$$

which is equal to

$$\xi = \frac{3\pi}{4}, \tag{24}$$

$$b = \frac{\pi}{4}, \tag{25}$$

for $a = 2$. This is the same value for ξ that is used in [5], but a much smaller bandwidth, showing that practical limitations (i.e. how many rotations are feasible) are more important than this heuristic. The final arbiter should always be the Littlewood-Paley condition given by Equation 8 and classification results.

Satisfying the Littlewood-Paley condition

For the classification experiment, I used $\xi = \frac{4\pi}{5}$ and $\sigma = 0.0129$ for the mother wavelet of size $(128, 256, 128)$, and constructed a filter bank for $J = 4$ and using rotations that map the vector $(1, 0, 0)$ onto 20 upper-hemisphere points of the Fibonacci lattice. This satisfies the Littlewood-Paley condition Equation 8 with $\epsilon = 0.92$ and an average deficiency $\epsilon_{\text{avg}} = 0.29$.

This value of σ seems too small to be reasonable, but it is likely necessary to compensate for the poor coverage of the Fourier domain by only 20 points of topmost frequency. The value of ϵ is correspondingly a lot worse than [4] obtains for experiments in 2D. More optimized code (to allow for more orientations) and further research is necessary to conclude whether it is feasible to construct a signal representation of 3D images of such size. See Appendix A for measures that might be taken to make the implementation fast.

4 Experiment on NCCT baseline scans

I used NCCT data from the MRCLEAN registry. To obtain 3D pictures of the brain hemispheres, I performed a rigid transform onto an “atlas” (a perfectly aligned model image) using the Elastix library [1], then used a region-growing algorithm to select only the pixels that belonged to brain tissue, then separated the hemisperes by cutting over the y-axis and finally resampled the image to dimensions $(128, 256, 128)$, to avoid having to zero-pad before doing the scattering transform.

I manually checked the center slices of the registered images and removed images for which rigid registration failed for some reason. I also removed patients with signs of earlier stroke. In this way, I selected 150 brain scans, yielding a dataset with 150 samples in each class. See Figure 1 for an example of an input affected hemisphere.

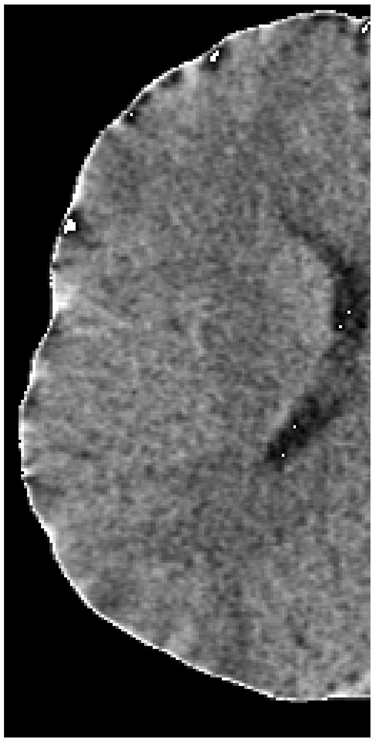

Figure 1: Axial slice of an input image of an affected brain.

4.1 Support vector machine classifier

I selected the SVM that had the highest precision on 80% of the dataset (with equal number of samples per class) with five-fold cross validation, from a set of linear SVMs with regularisation paramater $C \in \{1, 10, 100, 1000\}$ and Gaussian radial basis function kernel SVMS with the same C and $\gamma \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. The Gaussian kernel SVM with $C = 10$ and $\gamma = 10^{-5}$ performed the best with 80% precision on the training set. It obtained a **73% precision** on the remaining 20% of the dataset.

5 Discussion

The results are comparable to [2], and given the difficulty that professionals have in telling which brain hemispere is affected from image data alone, the result is not too bad. However, a first indication of classifying a downsampled version of the input data by raw pixels gives a precision of 65%-70% (also comparable to the experiments in [2]), so the scattering transform with current parameters does not give a significant improvement.

5.1 Further research

The MRCLEAN trial includes metadata indicating whether the radiologist could perform an ASPECTS scoring based on a comparison between brain hemispheres. It is interesting to compare the cases where the professional failed to distinguish between hemispheres and where the classifier failed to do so.

As indicated in section 3.3, further research needs to be done on choosing the number of filters in the filter bank, ξ and corresponding b. I assume that when some measures are taken to make the implementation more efficient, it might be possible to satisfy the Littlewood-Paley condition with smaller ϵ. Another simpler measure which might be undertaken to improve classification results is to perform cross-validation on the maximum length scale a^J.

When the scattering representation is somewhat under control, it might be worthwhile to use it as input of a deep neural network [9], or even as an initialisation of the first learnable layers.

As separate research questions, it might be asked whether visualising the features (as is done for example in [2]) can improve ASPECTS scoring by radiologists, perhaps in conjunction with directly classifying ASPECTS regions using the scattering representation.

5.2 Acknowledgements

I am grateful to Henk Marquering and Renan Sales Barros for giving me the opportunity to work on this project. Thanks to them and Ádám Hilbert and Elena Ponomareva for valuable discussions – I hope you guys find time to continue with this project, perhaps on Friday afternoons. Finally, I thank Vincent

Lostanlen for his great patience in sharing some of his expertise with me over the mail.

References

[1] URL: `http://elastix.isi.uu.nl/`.

[2] Tameem Adel et al. "3D scattering transforms for disease classification in neuroimaging". In: *NeuroImage: Clinical* 14.Supplement C (2017), pp. 506–517. ISSN: 2213-1582. DOI: `https://doi.org/10.1016/j.nicl.2017.02.004`. URL: `http://www.sciencedirect.com/science/article/pii/S2213158217300384`.

[3] J. Andén and S. Mallat. "Deep Scattering Spectrum". In: *IEEE Transactions on Signal Processing* 62.16 (Aug. 2014), pp. 4114–4128. ISSN: 1053-587X. DOI: `10.1109/TSP.2014.2326991`.

[4] J. Bruna. "Scattering representations for recognition". PhD thesis. Ecole Polytechnique X, 2013.

[5] J. Bruna and S. Mallat. "Invariant Scattering Convolution Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (Aug. 2013), pp. 1872–1886. ISSN: 0162-8828. DOI: `10.1109/TPAMI.2012.230`.

[6] V. Lostanlen. [Online; accessed 30 November 2017]. URL: `https://github.com/lostanlen/scattering.m/blob/master/lib/convolution/ifft_multiply.m`.

[7] V. Lostanlen. "Convolutional operators in the time-frequency domain". PhD thesis. PSL Research University, 2017.

[8] Stéphane Mallat. "Group Invariant Scattering". In: *Communications on Pure and Applied Mathematics* 65.10 (2012), pp. 1331–1398. ISSN: 1097-0312. DOI: `10.1002/cpa.21413`. URL: `http://dx.doi.org/10.1002/cpa.21413`.

[9] Edouard Oyallon, Eugene Belilovsky, and Sergey Zagoruyko. "Scaling the Scattering Transform: Deep Hybrid Networks". In: *preprint arXiv:1703.08961* (2017).

A Fast implementation

In this appendix, I will outline some straightforward steps to speed up the numerical implementation of the scattering transform, which is especially necessary in 3D.

Most of the energy is contained in the first two layers of the scattering transform, and especially in the subset of propagated signals with $j_1 < j_2$ [5, 8] (this is a typo in [**bruna2013**] under the header "Fast Scattering Computations", which says $j_1 \leq j_2$). I therefore only compute these paths in the implementation. Furthermore, signals that have been convolved with a filter at scale a^j, contain mostly frequencies below πa^{-j}, so can be safely downsampled by a factor a^j. (This also is true for the scattering coefficients $U[p] \star \phi_{a^J}$, which can be downsampled by a factor a^J at all layers.)

To implement the actual convolutions, it is of vital importance to perform them as a product in the Fourier domain (per the convolution theorem). It is therefore efficient to contstruct the filter bank in the Fourier domain and to keep downsampled versions in memory. Because we only consider the subset of paths with $j_1 < j_2$, we need a wavelet of length scale a^j downsampled at scales $a, a^2, \dots a^{j-1}$. See algorithm 1.

For pseudocode for the scattering transform, see [5]. To compute signals $|y \star \psi_{a^j,r}|$, it is more efficient to downsample in the Fourier domain, since that saves time in the inverse Fourier transformation, see algorithm 2. The equivalent of downsampling a spatial signal by a factor D, i.e.

$$x_D[n] = x[Dn] \tag{26}$$

corresponds in the Fourier domain to

$$X_D[\omega] = \frac{1}{D} \sum_{k=0}^{D-1} X(\frac{\omega - 2\pi k}{D}). \tag{27}$$

Note that both the construction of the filter bank and the scattering operations can be sped up by performing them with a GPU.

A final thing that might be attempted is to speed up elementwise multiplication in the Fourier domain by cropping the filters $\hat{\psi}_{a^j,r}$, see [6] for a code example in 1D.

Algorithm 1: Constructing the wavelet filter bank.

```
for $0 \leq j \leq J$ and $r \in R$ do
    construct $\hat{\psi}_{a^j,r}(\boldsymbol{\omega})$;
    for $j' \in \{1, \ldots, j-1\}$ do
        downsample $\hat{\psi}_{a^j,r}(\boldsymbol{\omega})$ by factor $a^{j'}$;
    end
end
construct $\hat{\phi}_{a^J}(\boldsymbol{\omega})$;
for $j \in \{1, \ldots, J\}$ do
    downsample $\hat{\phi}_{a^J}(\boldsymbol{\omega})$ by factor $a^j$;
end
```

Algorithm 2: Computing $|y \star \psi_{a^j,r}|$.

```
Input   : Fourier transforms $\hat{y}$ and $\hat{\psi}_{a^j,r}$.
Compute elementwise product $\hat{y} \cdot \hat{\psi}_{a^j,r}$;
Downsample $\hat{y} \cdot \hat{\psi}_{a^j,r}$ by factor $a^j$;
$y \star \psi_{a^j,r} \leftarrow \mathrm{ifft}(\hat{y} \cdot \hat{\psi}_{a^j,r})$;
Compute $|y \star \psi_{a^j,r}|$;
Compute $\mathrm{fft}(|y \star \psi_{a^j,r}|)$ for extracting
  scattering coefficients and computing next
  layer's transform;
```