Spectral Representations for Convolutional Neural Networks

Oren Rippel, Jasper Snoek, Ryan P. Adams

Reading Group Presenter: David Carlson

July 17, 2015

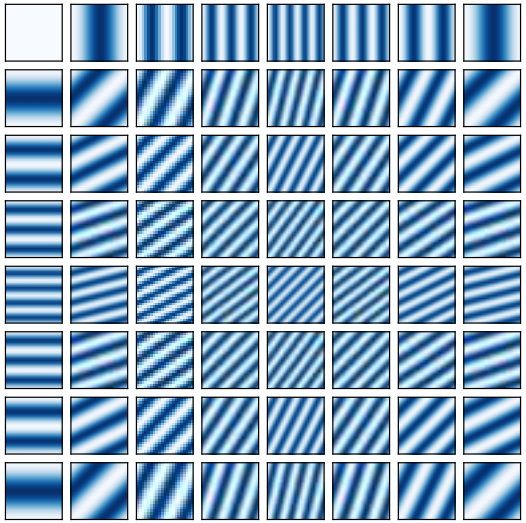
Introduction

- Convolutional Neural Nets (CNNs) have been wildly successful for image classification tasks
- However:
  - They are computationally expensive
  - Any pooling step reduces the dimensionality by at least 4
- Previous work suggests using FFTs to compute the convolutional mask–even for small filter sizes–to help with computational time
- This work suggests using FFTs, and then performing *pooling* and *learning* in the Fourier Transform domain
  - Introduced *spectral pooling* can reduce dimensionality by an user-defined amount (reduces *slower* than traditional pooling steps)
  - *Spectral parameterization* defines the CNN filters in the frequency domains, which empirically converges 2-5 times faster than the standard spatial representation with the same result

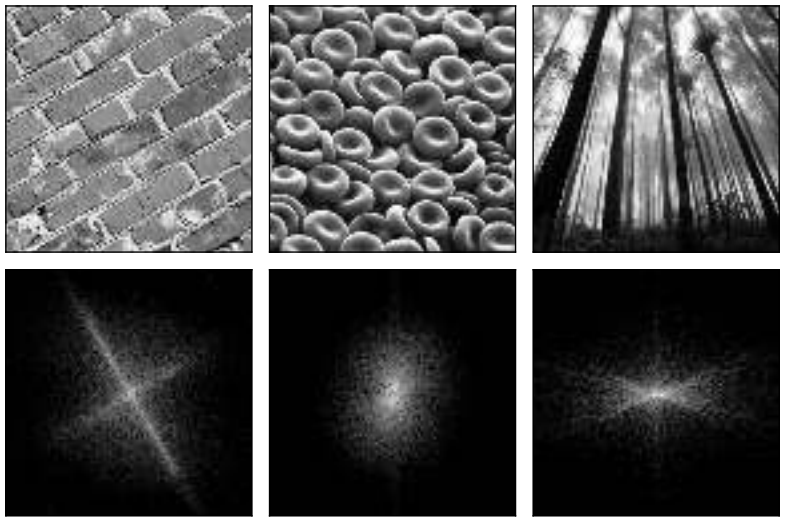
Reminder of Fourier Properties

- Convolution using DFT: $\mathcal{F}(\boldsymbol{x} * \boldsymbol{f}) = \mathcal{F}(\boldsymbol{x}) \odot \mathcal{F}(\boldsymbol{f})$
- Parseval's Theorem: $||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2 = ||\mathcal{F}(\boldsymbol{x}) - \mathcal{F}(\hat{\boldsymbol{x}})||_2^2$
- Conjugate symmetry forces $\mathbb{R}$ representation–if $\boldsymbol{x} \in \mathbb{R}^{M \times N}$, then $\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}) \in \mathbb{C}^{M \times N}$ has: $y_{mn} = y_{(M-m)\text{mod}(M),(N-n)\text{mod}(N)}$
  - This adds constraints on conjugate symmetry for filters
- Differentiation is straightforward because the Fourier transform is an (orthonormal) linear operator

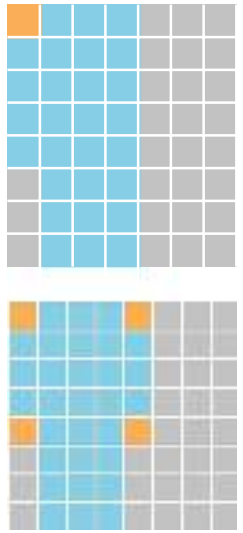
$$\frac{\delta R}{\delta \boldsymbol{x}} = \mathcal{F}^{-1}\left(\frac{\delta R}{\delta \boldsymbol{y}}\right)$$



(a) DFT basis functions.



(b) Examples of input-transform pairs.



(c) Conjugate Symm.

Figure 1: Properties of discrete Fourier transforms. **(a)** All discrete Fourier basis functions of map size 8×8. Note the equivalence of some of these due to conjugate symmetry. **(b)** Examples of input images and their frequency representations, presented as log-amplitudes. The frequency maps have been shifted to center the DC component. Rays in the frequency domain correspond to spatial domain edges aligned perpendicular to these. **(c)** Conjugate symmetry patterns for inputs with odd (top) and even (bottom) dimensionalities. **Orange**: real-valuedness constraint. **Blue**: no constraint. **Gray**: value fixed by conjugate symmetry.

Spectral Pooling

- The first proposed idea is *spectral pooling*:

Algorithm 1: Spectral pooling

Input: Map $\mathbf{x} \in \mathbb{R}^{M \times N}$, output size $H \times W$
Output: Pooled map $\hat{\mathbf{x}} \in \mathbb{R}^{H \times W}$

1: $\mathbf{y} \leftarrow \mathscr{F}(\mathbf{x})$
2: $\hat{\mathbf{y}} \leftarrow \textsc{CropSpectrum}(\mathbf{y}, H \times W)$
3: $\hat{\mathbf{y}} \leftarrow \textsc{TreatCornerCases}(\hat{\mathbf{y}})$
4: $\hat{\mathbf{x}} \leftarrow \mathscr{F}^{-1}(\hat{\mathbf{y}})$

- Very simple to understand, not as obvious why this is a good idea

Figure 2: Approximations for different pooling schemes, for different factors of dimensionality reduction. Spectral pooling projects onto the Fourier basis and truncates it as desired. This retains significantly more information and permits the selection of any arbitrary output map dimensionality.

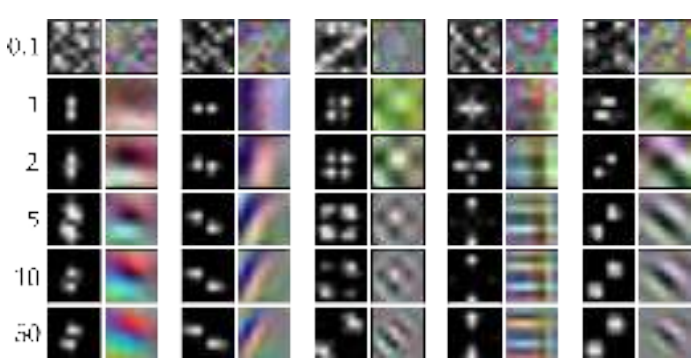
Spectral Parameterization of filters

- Seek to learn a filter $\boldsymbol{f} \in \mathbb{C}^{H \times W}$ that is parameterized in the frequency space
- If conjugate symmetry is upheld, then $\mathcal{F}^{-1}(\boldsymbol{f}) \in \mathbb{R}^{H \times W}$
- Because the Fourier transform is an (invertible) linear operator, the local minima and gradients are the same–only a change of basis
  - However, the recent optimization methods (ADAgrad, RMSprop, ADAM) use diagonal preconditioners, so a different basis can give vastly different performance

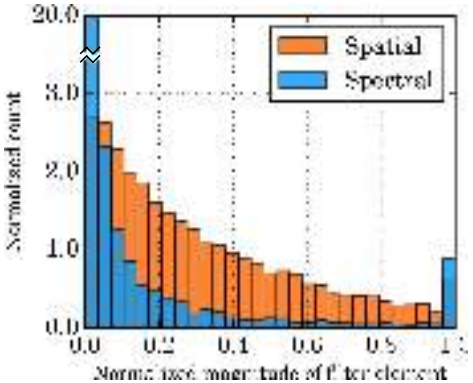
Algorithm 2: Spectral pooling back-propagation

Input: Gradient w.r.t output $\frac{\partial R}{\partial \hat{\mathbf{x}}}$
Output: Gradient w.r.t input $\frac{\partial R}{\partial \mathbf{x}}$

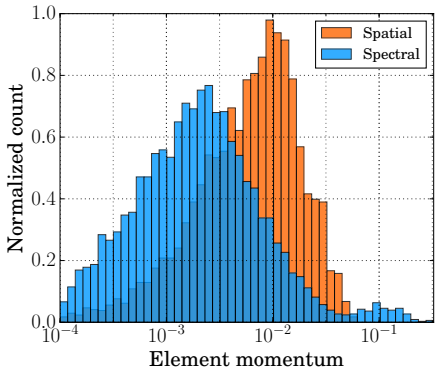
1: $\hat{\mathbf{z}} \leftarrow \mathscr{F}\left(\frac{\partial R}{\partial \hat{\mathbf{x}}}\right)$
2: $\hat{\mathbf{z}} \leftarrow \textsc{RemoveRedundancy}(\hat{\mathbf{z}})$
3: $\mathbf{z} \leftarrow \textsc{PadSpectrum}(\hat{\mathbf{z}}, M \times N)$
4: $\mathbf{z} \leftarrow \textsc{RecoverMap}(\mathbf{z})$
5: $\frac{\partial R}{\partial \mathbf{x}} \leftarrow \mathscr{F}^{-1}(\mathbf{z})$



(a) Filters over time.



(b) Sparsity patterns.



(c) Momenta distributions.

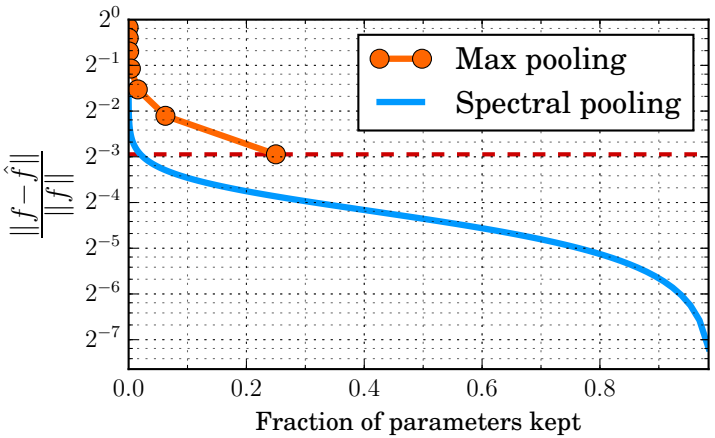
Figure 3: Learning dynamics of CNNs with spectral parametrization. The histograms have been produced after 10 epochs of training on CIFAR-10 by each method, but are similar throughout. **(a)** Progression over several epochs of filters parametrized in the frequency domain. Each pair of columns corresponds to the spectral parametrization of a filter and its inverse transform to the spatial domain. Filter representations tend to be more local in the Fourier basis. **(b)** Sparsity patterns for the different parametrizations. Spectral representations tend to be considerably sparser. **(c)** Distributions of momenta across parameters for CNNs trained with and without spectral parametrization. In the spectral parametrization considerably fewer parameters are updated.

Experiments

- Used CIFAR-10, CIFAR-100, and ImageNet
- Used the network:

$$\left(\mathrm{C}_{3\times3}^{96+32m} \rightarrow \mathrm{SP}_{\downarrow\lfloor\gamma H_m\rfloor\times\lfloor\gamma H_m\rfloor}\right)_{m=1}^{M} \rightarrow \mathrm{C}_{1\times1}^{96+32M} \rightarrow \mathrm{C}_{1\times1}^{10/100} \rightarrow \mathrm{GA} \rightarrow \mathrm{Softmax} \quad (5)$$

- SP is a spectral pooling layer and C_S^F has filters of size S with F filters
- Number of layers, penalization, nonlinearity type, and dimensionality reduction hyperparameters were tuned using
- Some other networks were used as well to show comparisons

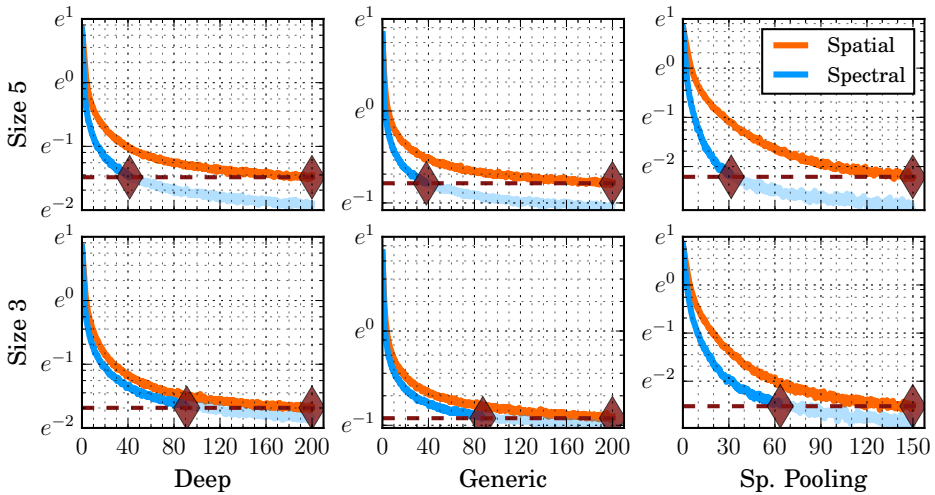


(a) Approximation loss for the ImageNet validation set.

Method	**CIFAR-10**	**CIFAR-100**
Stochastic pooling	15.13%	41.51%
Maxout	11.68%	38.57%
Network-in-network	10.41%	35.68%
Deeply supervised	9.78%	34.57%
Spectral pooling	**8.6%**	**31.6%**

(b) Classification rates.

Figure 4: **(a)** Average information dissipation for the ImageNet validation set as a function of fraction of parameters kept. This is measured in ℓ_2 error normalized by the input norm. The red horizontal line indicates the best error rate achievable by max pooling. **(b)** Test errors on CIFAR-10/100 without data augmentation of the optimal spectral pooling architecture, as compared to current state-of-the-art approaches: stochastic pooling (Zeiler & Fergus, 2013), Maxout (Goodfellow et al., 2013), network-in-network (Lin et al., 2013), and deeply-supervised nets (Lee et al., 2014).



(a) Training curves.

Architecture	Filter size	Speedup factor
Deep (7)	3×3	2.2
Deep (7)	5×5	4.8
Generic (6)	3×3	2.2
Generic (6)	5×5	5.1
Sp. Pooling (5)	3×3	2.4
Sp. Pooling (5)	5×5	4.8

(b) Speedup factors.

Figure 5: Optimization of CNNs via spectral parametrization. All experiments include data augmentation. **(a)** Training curves for the various experiments. The remainder of the optimization past the matching point is marked in light blue. The red diamonds indicate the relative epochs in which the asymptotic error rate of the spatial approach is achieved. **(b)** Speedup factors for different architectures and filter sizes. A non-negligible speedup is observed even for tiny 3×3 filters.