Harmonic Analysis of Deep Convolutional Neural Networks

Helmut Bölcskei

ETH Zürich

Department of Information Technology and Electrical Engineering

October 2017

joint work with Thomas Wiatowski and Philipp Grohs
ImageNet
CNNs win the ImageNet 2015 challenge [He et al., 2015]
CNNs win the ImageNet 2015 challenge [He et al., 2015]
Describing the content of an image

CNNs generate sentences describing the content of an image [Vinyals et al., 2015]
Describing the content of an image

*CNNs generate sentences describing the content of an image* [Vinyals et al., 2015]

“Carlos.”
Describing the content of an image

CNNs generate sentences describing the content of an image [Vinyals et al., 2015]

“Carlos Kleiber”
Describing the content of an image

CNNs generate sentences describing the content of an image [Vinyals et al., 2015]

“Carlos Kleiber conducting the"
Describing the content of an image

CNNs generate sentences describing the content of an image [Vinyals et al., 2015]

“Carlos Kleiber conducting the Vienna Philharmonic’s.”
Describing the content of an image

*CNNs generate sentences describing the content of an image [Vinyals et al., 2015]*

“Carlos Kleiber conducting the Vienna Philharmonic’s New Year’s Concert.”
Describing the content of an image

CNNs generate sentences describing the content of an image [Vinyals et al., 2015]

“Carlos Kleiber conducting the Vienna Philharmonic’s New Year’s Concert 1989.”
Feature extraction and classification

input: \( f = \)

non-linear feature extraction

feature vector \( \Phi(f) \)

linear classifier

output: \[
\begin{align*}
\langle w, \Phi(f) \rangle > 0, \quad &\Rightarrow \text{Shannon} \\
\langle w, \Phi(f) \rangle < 0, \quad &\Rightarrow \text{von Neumann}
\end{align*}
\]
Why non-linear feature extractors?

**Task:** Separate two categories of data through a *linear* classifier

\[
\langle w, f \rangle > 0
\]

\[
\langle w, f \rangle < 0
\]

\[
\Phi(f) = \|f\|_1
\]

\[
\langle w, \Phi(f) \rangle > 0
\]

\[
\langle w, \Phi(f) \rangle < 0
\]

**Notes:**
- \( w \) is a weight vector.
- \( f \) represents the feature vector.
- The linear classifier separates the data into two categories based on the dot product of the weight vector and the feature vector, with positive and negative values indicating different classes.
Why non-linear feature extractors?

**Task**: Separate two categories of data through a **linear** classifier.

\[
\begin{align*}
\langle w, f \rangle > 0 & : \text{red points} \\
\langle w, f \rangle < 0 & : \text{blue points}
\end{align*}
\]
Why non-linear feature extractors?

**Task:** Separate two categories of data through a **linear** classifier

\[ \Phi(f) = \begin{bmatrix} \|f\| \\ 1 \end{bmatrix} \]

\[ \langle w, \Phi(f) \rangle > 0 \]

\[ \langle w, \Phi(f) \rangle < 0 \]

**not possible!**

possible with \( w = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \)
Why non-linear feature extractors?

**Task:** Separate two categories of data through a **linear** classifier

\[ \Phi(f) = \begin{bmatrix} \|f\| \\ 1 \end{bmatrix} \]

⇒ \( \Phi \) is **invariant** to angular component of the data
Why non-linear feature extractors?

**Task:** Separate two categories of data through a **linear** classifier

\[ \Phi(f) = \begin{bmatrix} \|f\| \\ 1 \end{bmatrix} \]

⇒ \( \Phi \) is **invariant** to angular component of the data

⇒ **Linear separability** in feature space!
Translation invariance

*Handwritten digits from the MNIST database [LeCun & Cortes, 1998]*

Feature vector should be invariant to spatial location

⇒ translation invariance
Deformation insensitivity

Feature vector should be independent of cameras (of different resolutions), and insensitive to small acquisition jitters
Scattering networks ([Mallat, 2012], [Wiatowski and HB, 2015])
Scattering networks ([Mallat, 2012], [Wiatowski and HB, 2015])
Scattering networks ([Mallat, 2012], [Wiatowski and HB, 2015])

General scattering networks guarantee [Wiatowski & HB, 2015]
- (vertical) **translation invariance**
- **small deformation sensitivity**

essentially irrespective of filters, non-linearities, and poolings!
Building blocks

Basic operations in the $n$-th network layer

$g_{\lambda_n}^{(k)}$ → non-lin. → pool. → $g_{\lambda_n}^{(r)}$ → non-lin. → pool.

$\mathbf{f}$

Filters: Semi-discrete frame $\Psi_n := \{\chi_n\} \cup \{g_{\lambda_n}\}_{\lambda_n \in \Lambda_n}$

$$A_n \| f \|_2^2 \leq \| f \ast \chi_n \|_2^2 + \sum_{\lambda_n \in \Lambda_n} \| f \ast g_{\lambda_n} \|_2^2 \leq B_n \| f \|_2^2, \quad \forall f \in L^2(\mathbb{R}^d)$$
Building blocks

Basic operations in the $n$-th network layer

Filters: Semi-discrete frame $\Psi_n := \{\chi_n\} \cup \{g_{\lambda_n}\}_{\lambda_n \in \Lambda_n}$

$$A_n \|f\|_2^2 \leq \|f \ast \chi_n\|_2^2 + \sum_{\lambda_n \in \Lambda_n} \|f \ast g_{\lambda_n}\|^2 \leq B_n \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d)$$

e.g.: Structured filters
Building blocks

Basic operations in the $n$-th network layer

![Diagram showing basic operations in the $n$-th network layer: $g_{\lambda_n}^{(k)}$, non-lin., pool., followed by $g_{\lambda_n}^{(r)}$, non-lin., pool.]

Filters: Semi-discrete frame $\Psi_n := \{\chi_n\} \cup \{g_{\lambda_n}\}_{\lambda_n \in \Lambda_n}$

$$A_n \|f\|_2^2 \leq \|f \ast \chi_n\|_2^2 + \sum_{\lambda_n \in \Lambda_n} \|f \ast g_{\lambda_n}\|_2^2 \leq B_n \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d)$$

e.g.: Unstructured filters
Building blocks

Basic operations in the $n$-th network layer

Filters: Semi-discrete frame $\Psi_n := \{\chi_n\} \cup \{g_{\lambda_n}\}_{\lambda_n \in \Lambda_n}$

\[ A_n \|f\|_2^2 \leq \|f \ast \chi_n\|_2^2 + \sum_{\lambda_n \in \Lambda_n} \|f \ast g_{\lambda_n}\|^2 \leq B_n \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d) \]

e.g.: Learned filters
**Building blocks**

**Basic operations in the \( n \)-th network layer**

\[
g_{\lambda_n}^{(k)} \rightarrow \text{non-lin.} \rightarrow \text{pool.} \quad g_{\lambda_n}^{(r)} \rightarrow \text{non-lin.} \rightarrow \text{pool.} \quad f
\]

**Non-linearities:** Point-wise and Lipschitz-continuous

\[
\| M_n(f) - M_n(h) \|_2 \leq L_n \| f - h \|_2, \quad \forall f, h \in L^2(\mathbb{R}^d)
\]
Building blocks

Basic operations in the \( n \)-th network layer

\[
g_{\lambda_n}^{(k)} \rightarrow \text{non-lin.} \rightarrow \text{pool.}
\]

\[
g_{\lambda_n}^{(r)} \rightarrow \text{non-lin.} \rightarrow \text{pool.}
\]

Non-linearities: Point-wise and Lipschitz-continuous

\[
\| M_n(f) - M_n(h) \|_2 \leq L_n \| f - h \|_2, \quad \forall f, h \in L^2(\mathbb{R}^d)
\]

⇒ Satisfied by virtually all non-linearities used in the deep learning literature!

ReLU: \( L_n = 1 \); modulus: \( L_n = 1 \); logistic sigmoid: \( L_n = \frac{1}{4} \); ...
Building blocks

Basic operations in the \( n \)-th network layer

\[
g_{\lambda_n}^{(k)} \xrightarrow{\text{non-lin.}} \text{pool.}
\]

\[
g_{\lambda_n}^{(r)} \xrightarrow{\text{non-lin.}} \text{pool.}
\]

\[
f \mapsto S_{n}^{d/2} P_n(f)(S_n \cdot),
\]

where \( S_n \geq 1 \) is the pooling factor and \( P_n : L^2(\mathbb{R}^d) \rightarrow L^2(\mathbb{R}^d) \) is \( R_n \)-Lipschitz-continuous
Building blocks

Basic operations in the $n$-th network layer

$g_{\lambda_n}^{(k)} \xrightarrow{\text{non-lin.}} \text{pool.}$

$g_{\lambda_n}^{(r)} \xrightarrow{\text{non-lin.}} \text{pool.}$

$\mathbf{Pooling:}$ In continuous-time according to

$$f \mapsto S_{n/2}^d P_n(f)(S_n \cdot),$$

where $S_n \geq 1$ is the **pooling factor** and $P_n : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ is $R_n$-Lipschitz-continuous

$\Rightarrow$ **Emulates** most poolings used in the deep learning literature!

e.g.: Pooling by **sub-sampling** $P_n(f) = f$ with $R_n = 1$
Building blocks

Basic operations in the $n$-th network layer

$$
g_{\lambda_n^{(k)}} \xrightarrow{\text{non-lin.}} \text{pool.}$$

$$
g_{\lambda_n^{(r)}} \xrightarrow{\text{non-lin.}} \text{pool.}$$

**Pooling:** In continuous-time according to

$$f \mapsto S_n^{d/2} P_n(f)(S_n \cdot),$$

where $S_n \geq 1$ is the **pooling factor** and $P_n : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ is $R_n$-Lipschitz-continuous

$\Rightarrow$ **Emulates** most **poolings** used in the **deep learning literature**!

* e.g.: Pooling by **averaging** $P_n(f) = f \ast \phi_n$ with $R_n = \|\phi_n\|_1$
Vertical translation invariance

Theorem (Wiatowski and HB, 2015)

Assume that the filters, non-linearities, and poolings satisfy

\[ B_n \leq \min\{1, L_n^{-2} R_n^{-2}\}, \quad \forall \, n \in \mathbb{N}. \]

Let the pooling factors be \( S_n \geq 1, \, n \in \mathbb{N} \). Then,

\[ \|\|\| \Phi^n(T_t f) - \Phi^n(f) \|\|\| = \mathcal{O}\left( \frac{\|t\|}{S_1 \ldots S_n} \right), \]

for all \( f \in L^2(\mathbb{R}^d) \), \( t \in \mathbb{R}^d \), \( n \in \mathbb{N} \).
Vertical translation invariance

Theorem (Wiatowski and HB, 2015)

Assume that the filters, non-linearities, and poolings satisfy

\[ B_n \leq \min\{1, L_n^{-2} R_n^{-2}\}, \quad \forall n \in \mathbb{N}. \]

Let the pooling factors be \( S_n \geq 1, n \in \mathbb{N} \). Then,

\[ ||| \Phi^n(T_t f) - \Phi^n(f) ||| = \mathcal{O}\left( \frac{||t||}{S_1 \ldots S_n} \right), \]

for all \( f \in L^2(\mathbb{R}^d), t \in \mathbb{R}^d, n \in \mathbb{N} \).

⇒ Features become more invariant with increasing network depth!
Vertical translation invariance

**Theorem (Wiatowski and HB, 2015)**

Assume that the filters, non-linearities, and poolings satisfy

\[ B_n \leq \min\{1, L_n^{-2} R_n^{-2}\}, \quad \forall \ n \in \mathbb{N}. \]

Let the pooling factors be \( S_n \geq 1, \ n \in \mathbb{N} \). Then,

\[ \|\|\| \Phi^n(T_t f) - \Phi^n(f) \|\| \| = \mathcal{O}\left(\frac{\|t\|}{S_1 \ldots S_n}\right), \]

for all \( f \in L^2(\mathbb{R}^d), \ t \in \mathbb{R}^d, \ n \in \mathbb{N} \).

**Full translation invariance:** If \( \lim_{n\to\infty} S_1 \cdot S_2 \cdot \ldots \cdot S_n = \infty \), then

\[ \lim_{n\to\infty} \|\|\| \Phi^n(T_t f) - \Phi^n(f) \|\| \| = 0 \]
Theorem (Wiatowski and HB, 2015)

Assume that the filters, non-linearities, and poolings satisfy

\[ B_n \leq \min\{1, L_n^{-2} R_n^{-2}\}, \quad \forall \ n \in \mathbb{N}. \]

Let the pooling factors be \( S_n \geq 1, \ n \in \mathbb{N}. \) Then,

\[ \|\|\Phi^n(T_t f) - \Phi^n(f)\|\| = O\left(\frac{\|t\|}{S_1 \ldots S_n}\right), \]

for all \( f \in L^2(\mathbb{R}^d), \ t \in \mathbb{R}^d, \ n \in \mathbb{N}. \)

The condition

\[ B_n \leq \min\{1, L_n^{-2} R_n^{-2}\}, \quad \forall \ n \in \mathbb{N}, \]

is easily satisfied by normalizing the filters \( \{g_{\lambda_n}\}_{\lambda_n \in \Lambda_n}. \)
Theorem (Wiatowski and HB, 2015)

Assume that the filters, non-linearities, and poolings satisfy

\[ B_n \leq \min\{1, L_n^{-2} R_n^{-2}\}, \quad \forall n \in \mathbb{N}. \]

Let the pooling factors be \( S_n \geq 1, n \in \mathbb{N}. \) Then,

\[ \|\|\Phi^n(T_t f) - \Phi^n(f)\|\| = \mathcal{O}\left(\frac{\|t\|}{S_1 \ldots S_n}\right), \]

for all \( f \in L^2(\mathbb{R}^d), t \in \mathbb{R}^d, n \in \mathbb{N}. \)

\( \Rightarrow \) applies to **general** filters, non-linearities, and poolings
Philosophy behind invariance results

Mallat’s “horizontal” translation invariance [Mallat, 2012]:

\[
\lim_{J \to \infty} \| \Phi_W(T_t f) - \Phi_W(f) \| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \ \forall t \in \mathbb{R}^d
\]

“Vertical” translation invariance:

\[
\lim_{n \to \infty} \| \Phi^n(T_t f) - \Phi^n(f) \| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \ \forall t \in \mathbb{R}^d
\]
Philosophy behind invariance results

Mallat’s “horizontal” translation invariance [Mallat, 2012]:

$$\lim_{J \to \infty} ||\Phi_W(T_t f) - \Phi_W(f)|| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \forall t \in \mathbb{R}^d$$

- features become invariant in every network layer, but needs $J \to \infty$

“Vertical” translation invariance:

$$\lim_{n \to \infty} ||\Phi^n(T_t f) - \Phi^n(f)|| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \forall t \in \mathbb{R}^d$$

- features become more invariant with increasing network depth
Philosophy behind invariance results

Mallat’s “horizontal” translation invariance [Mallat, 2012]:

\[
\lim_{J \to \infty} \| \Phi_W(T_t f) - \Phi_W(f) \| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \forall t \in \mathbb{R}^d
\]

- features become invariant in every network layer, but needs \( J \to \infty \)
- applies to wavelet transform and modulus non-linearity without pooling

“Vertical” translation invariance:

\[
\lim_{n \to \infty} \| \Phi^n(T_t f) - \Phi^n(f) \| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \forall t \in \mathbb{R}^d
\]

- features become more invariant with increasing network depth
- applies to general filters, general non-linearities, and general poolings
Non-linear deformation \((F_\tau f)(x) = f(x - \tau(x))\), where \(\tau : \mathbb{R}^d \rightarrow \mathbb{R}^d\)

For “small” \(\tau\):
Non-linear deformations

**Nonlinear** deformation \((F_{\tau}f)(x) = f(x - \tau(x))\), where \(\tau : \mathbb{R}^d \to \mathbb{R}^d\)

For “large” \(\tau\):

![Image 1](image1.png) ![Image 2](image2.png)
Deformation sensitivity for signal classes

Consider $(F_{\tau}f)(x) = f(x - \tau(x)) = f(x - e^{-x^2})$

For given $\tau$ the amount of deformation induced can depend drastically on $f \in L^2(\mathbb{R}^d)$
Mallat’s deformation stability bound [Mallat, 2012]:
\[ \|\| \Phi_W(F_{\tau} f) - \Phi_W(f)\|\| \leq C(2^{-J}\|\| \tau \|\|_{\infty} + J\|\| D\tau \|\|_{\infty} + \|\| D^2 \tau \|\|_{\infty}) \|\| f \|\|_W, \]
for all \( f \in H_W \subseteq L^2(\mathbb{R}^d) \)

- The signal class \( H_W \) and the corresponding norm \( \|\| \cdot \|\|_W \) depend on the mother wavelet (and hence the network)

Our deformation sensitivity bound:
\[ \|\| \Phi(F_{\tau} f) - \Phi(f)\|\| \leq C_{\mathcal{C}}\|\| \tau \|\|_{\infty}^{\alpha}, \quad \forall f \in \mathcal{C} \subseteq L^2(\mathbb{R}^d) \]

- The signal class \( \mathcal{C} \) (band-limited functions, cartoon functions, or Lipschitz functions) is independent of the network
Philosophy behind deformation stability/sensitivity bounds

Mallat’s deformation stability bound [Mallat, 2012]:
\[ \|\| \Phi_W(F_{\tau}f) - \Phi_W(f) \|\| \leq C\left(2^{-J}\|\tau\|_{\infty} + J\|D\tau\|_{\infty} + \|D^2\tau\|_{\infty}\right)\|f\|_W, \]
for all \( f \in H_W \subseteq L^2(\mathbb{R}^d) \)

- Signal class description complexity implicit via norm \( \| \cdot \|_W \)

Our deformation sensitivity bound:
\[ \|\| \Phi(F_{\tau}f) - \Phi(f) \|\| \leq C_{C}\|\tau\|_{\infty}^\alpha, \quad \forall f \in C \subseteq L^2(\mathbb{R}^d) \]

- Signal class description complexity explicit via \( C_C \)
  - \( L \)-band-limited functions: \( C_C = \mathcal{O}(L) \)
  - cartoon functions of size \( K \): \( C_C = \mathcal{O}(K^{3/2}) \)
  - \( M \)-Lipschitz functions \( C_C = \mathcal{O}(M) \)
Mallat’s deformation stability bound [Mallat, 2012]:

\[ \|\|\Phi_W(F_\tau f) - \Phi_W(f)\|\| \leq C \left( 2^J \|\tau\|_\infty + J \|D\tau\|_\infty + \|D^2\tau\|_\infty \right) \|f\|_W, \]

for all \( f \in H_W \subseteq L^2(\mathbb{R}^d) \)

Our deformation sensitivity bound:

\[ \|\|\Phi(F_\tau f) - \Phi(f)\|\| \leq C_c \|\tau\|_\infty^\alpha, \quad \forall f \in \mathcal{C} \subseteq L^2(\mathbb{R}^d) \]

- Decay rate \( \alpha > 0 \) of the deformation error is signal-class-specific (band-limited functions: \( \alpha = 1 \), cartoon functions: \( \alpha = \frac{1}{2} \), Lipschitz functions: \( \alpha = 1 \))
Mallat’s deformation stability bound [Mallat, 2012]:
\[ ||\Phi_W(F_\tau f) - \Phi_W(f)|| \leq C \left(2^{-J} ||\tau||_\infty + J||D\tau||_\infty + ||D^2\tau||_\infty \right) ||f||_W, \]
for all \( f \in H_W \subseteq L^2(\mathbb{R}^d) \)

- The bound depends explicitly on higher order derivatives of \( \tau \)

Our deformation sensitivity bound:
\[ ||\Phi(F_\tau f) - \Phi(f)|| \leq C_C ||\tau||_\infty^\alpha, \quad \forall f \in \mathcal{C} \subseteq L^2(\mathbb{R}^d) \]

- The bound implicitly depends on derivative of \( \tau \) via the condition \( ||D\tau||_\infty \leq \frac{1}{2d} \)
Philosophy behind deformation stability/sensitivity bounds

Mallat’s deformation stability bound [Mallat, 2012]:

\[
|||\Phi_W(F_\tau f) - \Phi_W(f)||| \leq C(2^{-J}\|\tau\|_\infty + J\|D\tau\|_\infty + \|D^2\tau\|_\infty)\|f\|_W,
\]

for all \( f \in H_W \subseteq L^2(\mathbb{R}^d) \)

- The bound is \textit{coupled} to horizontal translation invariance

\[
\lim_{J \to \infty} |||\Phi_W(T_t f) - \Phi_W(f)||| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \forall t \in \mathbb{R}^d
\]

Our deformation sensitivity bound:

\[
|||\Phi(F_\tau f) - \Phi(f)||| \leq C_C\|\tau\|_\infty^\alpha, \quad \forall f \in \mathcal{C} \subseteq L^2(\mathbb{R}^d)
\]

- The bound is \textit{decoupled} from vertical translation invariance

\[
\lim_{n \to \infty} |||\Phi^n(T_t f) - \Phi^n(f)||| = 0, \quad \forall f \in L^2(\mathbb{R}^d), \forall t \in \mathbb{R}^d
\]
CNNs in a nutshell

CNNs used in practice employ potentially hundreds of layers and 10,000s of nodes!
CNNs in a nutshell

CNNs used in practice employ potentially hundreds of layers and 10,000s of nodes!

e.g.: Winner of the ImageNet 2015 challenge [He et al., 2015]
- Network **depth**: 152 layers
- average # of **nodes** per layer: 472
- # of **FLOPS** for a single forward pass: 11.3 billion
CNNs in a nutshell

CNNs used in practice employ potentially hundreds of layers and 10,000s of nodes!

e.g.: Winner of the ImageNet 2015 challenge [He et al., 2015]
- Network **depth**: 152 layers
- average # of **nodes** per layer: 472
- # of **FLOPS** for a single forward pass: 11.3 billion

Such depths (and breadths) pose formidable computational challenges in **training** and **operating** the network!
Determine **how fast** the energy contained in the propagated signals (a.k.a. feature maps) decays across layers.
Determine **how fast** the energy contained in the propagated signals (a.k.a. feature maps) decays across layers.

Guarantee **trivial null-space** for feature extractor $\Phi$. 
Topology reduction

Determine **how fast** the energy contained in the propagated signals (a.k.a. feature maps) decays across layers.

Guarantee **trivial null-space** for feature extractor $\Phi$.

Specify the **number of layers** needed to have “most” of the input signal energy be contained in the feature vector.
Topology reduction

Determine **how fast** the energy contained in the propagated signals (a.k.a. feature maps) decays across layers.

Guarantee **trivial null-space** for feature extractor $\Phi$.

Specify the **number of layers** needed to have “most” of the input signal energy be contained in the feature vector.

For a fixed (possibly small) depth, **design CNNs** that capture “most” of the input signal energy.
Building blocks

Basic operations in the $n$-th network layer

Filters: Semi-discrete frame $\Psi_n := \{\chi_n\} \cup \{g_{\lambda_n}\}_{\lambda_n \in \Lambda_n}$

Non-linearity: Modulus $|\cdot|$ 

Pooling: Sub-sampling with pooling factor $S \geq 1$
Demodulation effect of modulus non-linearity

Components of feature vector given by $|f \ast g_{\lambda_n}| \ast \chi_{n+1}$
Demodulation effect of modulus non-linearity

Components of feature vector given by $|f * g_{\lambda n}| * \chi_{n+1}$
Demodulation effect of modulus non-linearity

Components of feature vector given by $|f \ast g_{\lambda n}| \ast \chi_{n+1}$

Modulus squared:

$$|f \ast g_{\lambda n}(x)|^2$$
Demodulation effect of modulus non-linearity

Components of feature vector given by $|f \ast g_{\lambda_n}| \ast \chi_{n+1}$
Do all non-linearities demodulate?

**High-pass** filtered signal:

\[ \mathcal{F}(f \ast g_{\lambda}) \]

![Diagram](image-url)
Do all non-linearities demodulate?

**High-pass** filtered signal:

$$\mathcal{F}(f \ast g_\lambda)$$

Modulus: **Yes**!

$$|\mathcal{F}(|f \ast g_\lambda|)|$$

... but (small) tails!
Do all non-linearities demodulate?

**High-pass** filtered signal:

\[ \mathcal{F}(f \ast g_{\lambda}) \]

\[ |\mathcal{F}(|f \ast g_{\lambda}|^2)| \]

**Modulus squared:** Yes, and sharply so!

... but not Lipschitz-continuous!
Do all non-linearities demodulate?

**High-pass** filtered signal:

\[ \mathcal{F}(f \ast g_\lambda) \]

**Rectified linear unit:** No!

\[ |\mathcal{F}({\text{ReLU}}(f \ast g_\lambda))| \]
First goal: Quantify feature map energy decay

\[ W_1(f) = \sum \left| f * g_{\lambda_1} \right| * \chi_1 \]

\[ W_2(f) = \sum \left| f * g_{\lambda_1} \right| * \chi_2 \]
Assumptions (on the filters)

i) **Analyticity**: For every filter $g_{\lambda_n}$ there exists a (not necessarily canonical) orthant $H_{\lambda_n} \subseteq \mathbb{R}^d$ such that

$$\text{supp}(\hat{g}_{\lambda_n}) \subseteq H_{\lambda_n}.$$ 

ii) **High-pass**: There exists $\delta > 0$ such that

$$\sum_{\lambda_n \in \Lambda_n} |\hat{g}_{\lambda_n}(\omega)|^2 = 0, \quad \text{a.e. } \omega \in B_\delta(0).$$
Assumptions (on the filters)

i) **Analyticity**: For every filter \( g_{\lambda_n} \) there exists a (not necessarily canonical) orthant \( H_{\lambda_n} \subseteq \mathbb{R}^d \) such that 
\[
\text{supp}(\hat{g}_{\lambda_n}) \subseteq H_{\lambda_n}.
\]

ii) **High-pass**: There exists \( \delta > 0 \) such that 
\[
\sum_{\lambda_n \in \Lambda_n} |\hat{g}_{\lambda_n}(\omega)|^2 = 0, \quad \text{a.e. } \omega \in B_{\delta}(0).
\]

⇒ Comprises various constructions of WH filters, wavelets, ridgelets, \((\alpha)\)-curvelets, shearlets

e.g.: analytic band-limited curvelets:
Sobolev functions of order $s \geq 0$:

$$H^s(\mathbb{R}^d) = \left\{ f \in L^2(\mathbb{R}^d) \mid \int_{\mathbb{R}^d} (1 + |\omega|^2)^s |\hat{f}(\omega)|^2 d\omega < \infty \right\}$$
Input signal classes

Sobolev functions of order $s \geq 0$:

$$H^s(\mathbb{R}^d) = \left\{ f \in L^2(\mathbb{R}^d) \mid \int_{\mathbb{R}^d} (1 + |\omega|^2)^s |\hat{f}(\omega)|^2 d\omega < \infty \right\}$$

$H^s(\mathbb{R}^d)$ contains a wide range of **practically relevant** signal classes.
Input signal classes

Sobolev functions of order $s \geq 0$:

$$H^s(\mathbb{R}^d) = \left\{ f \in L^2(\mathbb{R}^d) \mid \int_{\mathbb{R}^d} (1 + |\omega|^2)^s |\hat{f}(\omega)|^2 d\omega < \infty \right\}$$

$H^s(\mathbb{R}^d)$ contains a wide range of **practically relevant** signal classes

- square-integrable functions $L^2(\mathbb{R}^d) = H^0(\mathbb{R}^d)$
Input signal classes

Sobolev functions of order $s \geq 0$:

$$H^s(\mathbb{R}^d) = \left\{ f \in L^2(\mathbb{R}^d) \mid \int_{\mathbb{R}^d} (1 + |\omega|^2)^s |\hat{f}(\omega)|^2 d\omega < \infty \right\}$$

$H^s(\mathbb{R}^d)$ contains a wide range of **practically relevant** signal classes

- square-integrable functions $L^2(\mathbb{R}^d) = H^0(\mathbb{R}^d)$
- $L$-band-limited functions $L^2_L(\mathbb{R}^d) \subseteq H^s(\mathbb{R}^d), \forall L > 0, \forall s \geq 0$
Input signal classes

Sobolev functions of order \( s \geq 0 \):

\[
H^s(\mathbb{R}^d) = \left\{ f \in L^2(\mathbb{R}^d) \mid \int_{\mathbb{R}^d} (1 + |\omega|^2)^s |\hat{f}(\omega)|^2 d\omega < \infty \right\}
\]

\( H^s(\mathbb{R}^d) \) contains a wide range of \textbf{practically relevant} signal classes

- square-integrable functions \( L^2(\mathbb{R}^d) = H^0(\mathbb{R}^d) \)
- \( L \)-band-limited functions \( L^2_L(\mathbb{R}^d) \subseteq H^s(\mathbb{R}^d), \forall L > 0, \forall s \geq 0 \)
- cartoon functions \([\text{Donoho, 2001}]\) \( \mathcal{C}_{\text{CART}} \subseteq H^s(\mathbb{R}^d), \forall s \in [0, \frac{1}{2}) \)

Handwritten digits from MNIST database \([\text{LeCun & Cortes, 1998}]\)
Exponential energy decay

Theorem

Let the filters be **wavelets** with mother wavelet

\[ \text{supp}(\hat{\psi}) \subseteq [r^{-1}, r], \quad r > 1, \]

or **Weyl-Heisenberg (WH) filters** with prototype function

\[ \text{supp}(\hat{g}) \subseteq [-R, R], \quad R > 0. \]

Then, for every \( f \in H^s(\mathbb{R}^d) \), there exists \( \beta > 0 \) such that

\[ W_n(f) = \mathcal{O}\left(a^{-\frac{n(2s+\beta)}{2s+\beta+1}}\right), \]

where \( a = \frac{r^2+1}{r^2-1} \) in the wavelet case, and \( a = \frac{1}{2} + \frac{1}{R} \) in the WH case.
Exponential energy decay

Theorem

Let the filters be wavelets with mother wavelet

$$\text{supp}(\hat{\psi}) \subseteq [r^{-1}, r], \quad r > 1,$$

or Weyl-Heisenberg (WH) filters with prototype function

$$\text{supp}(\hat{g}) \subseteq [-R, R], \quad R > 0.$$

Then, for every \( f \in H^s(\mathbb{R}^d) \), there exists \( \beta > 0 \) such that

$$W_n(f) = O\left(a^{-\frac{n(2s+\beta)}{2s+\beta+1}}\right),$$

where \( a = \frac{r^2+1}{r^2-1} \) in the wavelet case, and \( a = \frac{1}{2} + \frac{1}{R} \) in the WH case.

\( \Rightarrow \) decay factor \( a \) is explicit and can be tuned via \( r, R \)
Exponential energy decay

**Exponential** energy decay:

\[ W_n(f) = O\left(a^{-\frac{n(2s+\beta)}{2s+\beta+1}}\right) \]
Exponential energy decay:

\[ W_n(f) = O\left( a^{-\frac{n(2s+\beta)}{2s+\beta+1}} \right) \]

- \( \beta > 0 \) determines the decay of \( \hat{f}(\omega) \) (as \( |\omega| \to \infty \)) according to

\[ |\hat{f}(\omega)| \leq \mu(1 + |\omega|^2)^{-\left(\frac{s}{2} + \frac{1}{4} + \frac{\beta}{4}\right)}, \quad \forall |\omega| \geq L, \]

for some \( \mu > 0 \), and \( L \) acts as an “effective bandwidth”
Exponential energy decay

Exponential energy decay:

$$W_n(f) = \mathcal{O}\left(a^{-\frac{n(2s+\beta)}{2s+\beta+1}}\right)$$

- $\beta > 0$ determines the decay of $\hat{f}(\omega)$ (as $|\omega| \to \infty$) according to

$$|\hat{f}(\omega)| \leq \mu(1 + |\omega|^2)^{-\left(\frac{s}{2} + \frac{1}{4} + \frac{\beta}{4}\right)}, \quad \forall |\omega| \geq L,$$

for some $\mu > 0$, and $L$ acts as an “effective bandwidth”

- smoother input signals (i.e., $s \uparrow$) lead to faster energy decay
Exponential energy decay

**Exponential** energy decay:

\[ W_n(f) = \mathcal{O}\left(a^{-\frac{n(2s+\beta)}{2s+\beta+1}}\right) \]

- \( \beta > 0 \) determines the **decay** of \( \hat{f}(\omega) \) (as \( |\omega| \to \infty \)) according to

\[ |\hat{f}(\omega)| \leq \mu \left(1 + |\omega|^2\right)^{-\left(\frac{s}{2} + \frac{1}{4} + \frac{\beta}{4}\right)}, \quad \forall |\omega| \geq L, \]

for some \( \mu > 0 \), and \( L \) acts as an “effective bandwidth”

- **smoother** input signals (i.e., \( s \uparrow \)) lead to faster energy decay

- **pooling** through sub-sampling \( f \mapsto S^{1/2} f(S \cdot) \) leads to decay factor \( \frac{a}{S} \)

What about general filters? \( \Rightarrow \) **polynomial energy decay!**
Exponential energy decay

**Exponential energy decay:**

\[ W_n(f) = \mathcal{O}\left(a^{-\frac{n(2s+\beta)}{2s+\beta+1}}\right) \]

- \( \beta > 0 \) determines the **decay** of \( \hat{f}(\omega) \) (as \( |\omega| \to \infty \)) according to

\[ |\hat{f}(\omega)| \leq \mu (1 + |\omega|^2)^{-\left(\frac{s}{2} + \frac{1}{4} + \frac{\beta}{4}\right)}, \quad \forall |\omega| \geq L, \]

for some \( \mu > 0 \), and \( L \) acts as an “effective bandwidth”

- **smoother** input signals (i.e., \( s \uparrow \)) lead to **faster** energy decay

- **pooling** through sub-sampling \( f \mapsto S^{1/2}f(S \cdot) \) leads to decay factor \( \frac{a}{S} \)

What about **general** filters? \( \Rightarrow \) **polynomial** energy decay!
... our second goal ... trivial null-space for $\Phi$

**Why trivial null-space?**

**Feature space**

![Feature space diagram]

- : $\langle w, \Phi(f) \rangle > 0$
- : $\langle w, \Phi(f) \rangle < 0$

Non-trivial null-space: $\exists f^* \neq 0 \text{ such that } \Phi(f^*) = 0 \Rightarrow \langle w, \Phi(f^*) \rangle = 0$ for all $w \neq 0 \Rightarrow$ these $f^*$ become unclassifiable!
... our second goal ... trivial null-space for $\Phi$

Why trivial null-space?

Feature space

\[ \langle w, \Phi(f) \rangle > 0 \]
\[ : \langle w, \Phi(f) \rangle < 0 \]

Non-trivial null-space: \( \exists f^* \neq 0 \) such that \( \Phi(f^*) = 0 \)

\( \Rightarrow \langle w, \Phi(f^*) \rangle = 0 \) for all \( w \)!

\( \Rightarrow \) these \( f^* \) become unclassifiable!
... our second goal ...

**Trivial null-space** for feature extractor:

\[
\left\{ f \in L^2(\mathbb{R}^d) \mid \Phi(f) = 0 \right\} = \{0\}
\]

**Feature extractor** \( \Phi(\cdot) = \bigcup_{n=0}^{\infty} \Phi^n(\cdot) \) shall satisfy

\[
A\|f\|_2^2 \leq \||\Phi(f)||^2 \leq B\|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d),
\]

for some \( A, B > 0 \).
Theorem

For the frame upper \( \{B_n\}_{n \in \mathbb{N}} \) and frame lower bounds \( \{A_n\}_{n \in \mathbb{N}} \), define \( B := \prod_{n=1}^{\infty} \max\{1, B_n\} \) and \( A := \prod_{n=1}^{\infty} \min\{1, A_n\} \). If \( 0 < A \leq B < \infty \), then

\[
A \|f\|_2^2 \leq \|\Phi(f)\|_2^2 \leq B \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d).
\]
**Theorem**

For the frame upper \( \{B_n\}_{n \in \mathbb{N}} \) and frame lower bounds \( \{A_n\}_{n \in \mathbb{N}} \), define 
\[
B := \prod_{n=1}^{\infty} \max\{1, B_n\} \quad \text{and} \quad A := \prod_{n=1}^{\infty} \min\{1, A_n\}. 
\]
If 
\[
0 < A \leq B < \infty, 
\]
then
\[
A \|f\|_2^2 \leq \|\Phi(f)\|_2^2 \leq B \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d).
\]

- For **Parseval** frames (i.e., \( A_n = B_n = 1, \ n \in \mathbb{N} \)), this yields
\[
\|\Phi(f)\|_2^2 = \|f\|_2^2
\]
Theorem

For the frame upper \( \{B_n\}_{n \in \mathbb{N}} \) and frame lower bounds \( \{A_n\}_{n \in \mathbb{N}} \), define \( B := \prod_{n=1}^{\infty} \max\{1, B_n\} \) and \( A := \prod_{n=1}^{\infty} \min\{1, A_n\} \). If \( 0 < A \leq B < \infty \), then

\[
A \|f\|_2^2 \leq \|\Phi(f)\|_2^2 \leq B \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d).
\]

- For Parseval frames (i.e., \( A_n = B_n = 1, n \in \mathbb{N} \)), this yields

\[
\|\Phi(f)\|_2^2 = \|f\|_2^2
\]

- Connection to energy decay:

\[
\|f\|_2^2 = \sum_{k=0}^{n-1} \|\Phi^k(f)\|_2^2 + W_n(f) \rightarrow 0
\]
... and our third goal ...

For a given CNN, specify the **number of layers** needed to capture “most” of the input signal energy.
For a given CNN, specify the **number of layers** needed to capture “most” of the input signal energy

How many layers $n$ are needed to have at least $((1 - \varepsilon) \cdot 100)\%$ of the input signal energy be contained in the **feature vector**, i.e.,

$$(1 - \varepsilon)\|f\|_2^2 \leq \sum_{k=0}^{n} |||\Phi^k(f)|||^2 \leq \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d).$$
Number of layers needed

**Theorem**

Let the frame bounds satisfy $A_n = B_n = 1$, $n \in \mathbb{N}$. Let the input signal $f$ be $L$-band-limited, and let $\varepsilon \in (0, 1)$. If

$$n \geq \left\lfloor \log_a \left( \frac{L}{(1 - \sqrt{1 - \varepsilon})} \right) \right\rfloor,$$

then

$$(1 - \varepsilon) \| f \|_2^2 \leq \sum_{k=0}^{n} \| \Phi^k (f) \|_2^2 \leq \| f \|_2^2.$$
Number of layers needed

**Theorem**

Let the frame bounds satisfy \( A_n = B_n = 1, \ n \in \mathbb{N} \). Let the input signal \( f \) be \( L \)-band-limited, and let \( \varepsilon \in (0, 1) \). If

\[
n \geq \left\lceil \log_a \left( \frac{L}{(1 - \sqrt{1 - \varepsilon})} \right) \right\rceil,
\]

then

\[
(1 - \varepsilon) \| f \|_2^2 \leq \sum_{k=0}^{n} \| \Phi^k(f) \|_2^2 \leq \| f \|_2^2.
\]

\[
\Rightarrow \ \text{also guarantees trivial null-space for} \ \bigcup_{k=0}^{n} \Phi^k(f)
\]
Number of layers needed

**Theorem**

Let the frame bounds satisfy \( A_n = B_n = 1, \ n \in \mathbb{N} \). Let the input signal \( f \) be \( L \)-band-limited, and let \( \varepsilon \in (0, 1) \). If

\[
n \geq \left\lceil \log_a \left( \frac{L}{(1 - \sqrt{1 - \varepsilon})} \right) \right\rceil
\]

then

\[
(1 - \varepsilon) \| f \|_2^2 \leq \sum_{k=0}^{n} \| \Phi_k(f) \|_2^2 \leq \| f \|_2^2.
\]

- lower bound depends on
  - **description complexity** of input signals (i.e., bandwidth \( L \))
  - **decay factor** (wavelets \( a = \frac{r^2 + 1}{r^2 - 1} \), WH filters \( a = \frac{1}{2} + \frac{1}{R} \))
Number of layers needed

**Theorem**

Let the frame bounds satisfy $A_n = B_n = 1$, $n \in \mathbb{N}$. Let the input signal $f$ be $L$-band-limited, and let $\varepsilon \in (0, 1)$. If

$$n \geq \left\lceil \log_a \left( \frac{L}{(1 - \sqrt{1 - \varepsilon})} \right) \right\rceil,$$

then

$$(1 - \varepsilon) \|f\|_2^2 \leq \sum_{k=0}^{n} \|\Phi^k(f)\|_2^2 \leq \|f\|_2^2.$$

- lower bound depends on
  - **description complexity** of input signals (i.e., bandwidth $L$)
  - **decay factor** (wavelets $a = \frac{r^2+1}{r^2-1}$, WH filters $a = \frac{1}{2} + \frac{1}{R}$)
- similar estimates for **Sobolev** input signals and for **general** filters (polynomial decay!)
Number of layers needed

Numerical example for bandwidth $L = 1$:

<table>
<thead>
<tr>
<th></th>
<th>$(1 - \varepsilon)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25  0.5  0.75  0.9  0.95  0.99</td>
</tr>
<tr>
<td>wavelets ($r = 2$)</td>
<td>2    3    4    6    8    11</td>
</tr>
<tr>
<td>WH filters ($R = 1$)</td>
<td>2    4    5    8    10   14</td>
</tr>
<tr>
<td>general filters</td>
<td>2    3    7   19   39   199</td>
</tr>
</tbody>
</table>
Number of layers needed

Numerical example for bandwidth $L = 1$:

<table>
<thead>
<tr>
<th></th>
<th>0.25</th>
<th>0.5</th>
<th>0.75</th>
<th>0.9</th>
<th>0.95</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavelets ($r = 2$)</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>WH filters ($R = 1$)</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>8</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>general filters</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>19</td>
<td>39</td>
<td>199</td>
</tr>
</tbody>
</table>

Recall: Winner of the ImageNet 2015 challenge [He et al., 2015]

- Network depth: 152 layers
- Average # of nodes per layer: 472
- # of FLOPS for a single forward pass: 11.3 billion
Number of layers needed

Numerical example for bandwidth $L = 1$:

<table>
<thead>
<tr>
<th></th>
<th>(1 − ε)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.25 0.5 0.75 0.9 0.95 0.99</td>
</tr>
<tr>
<td>wavelets ($r = 2$)</td>
<td>2 3 4 6 8 11</td>
</tr>
<tr>
<td>WH filters ($R = 1$)</td>
<td>2 4 5 8 10 14</td>
</tr>
<tr>
<td>general filters</td>
<td>2 3 7 19 39 199</td>
</tr>
</tbody>
</table>

**Recall:** Winner of the ImageNet 2015 challenge [He et al., 2015]

- Network **depth**: 152 layers
- average # of **nodes** per layer: 472
- # of **FLOPS** for a single forward pass: 11.3 billion
... our fourth and last goal ...

For a fixed (possibly small) depth $N$, design scattering networks that capture “most” of the input signal energy.
For a fixed (possibly small) depth $N$, design scattering networks that capture “most” of the input signal energy.

**Recall:** Let the filters be wavelets with mother wavelet

$$\text{supp}(\hat{\psi}) \subseteq [r^{-1}, r], \quad r > 1,$$

or Weyl-Heisenberg filters with prototype function

$$\text{supp}(\hat{g}) \subseteq [-R, R], \quad R > 0.$$
... our fourth and last goal ... 

For a fixed (possibly small) depth $N$, design scattering networks that capture “most” of the input signal energy.

For fixed depth $N$, want to choose $r$ in the wavelet and $R$ in the WH case so that

$$(1 - \varepsilon)\|f\|_2^2 \leq \sum_{k=0}^{N} \|\Phi^k(f)\|_2^2 \leq \|f\|_2^2, \quad \forall f \in L^2(\mathbb{R}^d).$$
Depth-constrained networks

**Theorem**

Let the frame bounds satisfy \( A_n = B_n = 1, \ n \in \mathbb{N} \). Let the input signal \( f \) be \( L \)-band-limited, and fix \( \varepsilon \in (0, 1) \) and \( N \in \mathbb{N} \). If, in the wavelet case,

\[
1 < r \leq \sqrt{\frac{\kappa + 1}{\kappa - 1}},
\]

or, in the WH case,

\[
0 < R \leq \sqrt{\frac{1}{\kappa - \frac{1}{2}}},
\]

where \( \kappa := \left( \frac{L}{(1 - \sqrt{1 - \varepsilon})} \right)^{\frac{1}{N}} \), then

\[
(1 - \varepsilon) \| f \|_2^2 \leq \sum_{k=0}^{N} \| \Phi_k^k(f) \|_2^2 \leq \| f \|_2^2.
\]
Depth-width tradeoff

Spectral supports of wavelet filters:
Spectral supports of wavelet filters:

Smaller depth $N \Rightarrow$ smaller “bandwidth” $r$ of mother wavelet
Spectral supports of wavelet filters:

Smaller depth $N \Rightarrow$ smaller “bandwidth” $r$ of mother wavelet

$\Rightarrow$ larger number of wavelets ($O(\log_r(L))$) to cover the spectral support $[-L, L]$ of input signal
**Spectral supports** of wavelet filters:

Smaller depth $N \Rightarrow$ smaller “bandwidth” $r$ of mother wavelet

$\Rightarrow$ larger number of wavelets ($O(\log_r(L))$) to cover the spectral support $[-L, L]$ of input signal

$\Rightarrow$ larger number of filters in the first layer
Depth-width tradeoff

Spectral supports of wavelet filters:

Smaller depth $N \Rightarrow$ smaller “bandwidth” $r$ of mother wavelet

$\Rightarrow$ larger number of wavelets $\mathcal{O}(\log_r(L))$ to cover the spectral support $[-L, L]$ of input signal

$\Rightarrow$ larger number of filters in the first layer

$\Rightarrow$ depth-width tradeoff
Yours truly
Experiment: Handwritten digit classification

- Dataset: MNIST database of handwritten digits [LeCun & Cortes, 1998]; 60,000 training and 10,000 test images
- Φ-network: $D = 3$ layers; same filters, non-linearities, and pooling operators in all layers
- Classifier: SVM with radial basis function kernel [Vapnik, 1995]
- Dimensionality reduction: Supervised orthogonal least squares scheme [Chen et al., 1991]
Experiment: Handwritten digit classification

 Classification error in percent:

<table>
<thead>
<tr>
<th></th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
<th></th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
</tr>
</thead>
<tbody>
<tr>
<td>n.p.</td>
<td>0.57</td>
<td>0.57</td>
<td>1.35</td>
<td>1.49</td>
<td></td>
<td>0.51</td>
<td>0.57</td>
<td>1.12</td>
<td>1.22</td>
</tr>
<tr>
<td>sub.</td>
<td>0.69</td>
<td>0.66</td>
<td>1.25</td>
<td>1.46</td>
<td></td>
<td>0.61</td>
<td>0.61</td>
<td>1.20</td>
<td>1.18</td>
</tr>
<tr>
<td>max.</td>
<td>0.58</td>
<td>0.65</td>
<td>0.75</td>
<td>0.74</td>
<td></td>
<td>0.52</td>
<td>0.64</td>
<td>0.78</td>
<td>0.73</td>
</tr>
<tr>
<td>avg.</td>
<td>0.55</td>
<td>0.60</td>
<td>1.27</td>
<td>1.35</td>
<td></td>
<td>0.58</td>
<td>0.59</td>
<td>1.07</td>
<td>1.26</td>
</tr>
</tbody>
</table>

- modulus and ReLU perform better than tanh and LogSig
- results with pooling ($S = 2$) are competitive with those without
- state-of-the-art: 0.43 [Bruna and Mallat, 2013]
- similar feature extraction network with directional, non-separable wavelets and no pooling
- significantly higher computational complexity
## Experiment: Handwritten digit classification

### Classification error in percent:

<table>
<thead>
<tr>
<th></th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
<th></th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>n.p.</strong></td>
<td>0.57</td>
<td>0.57</td>
<td>1.35</td>
<td>1.49</td>
<td></td>
<td>0.51</td>
<td>0.57</td>
<td>1.12</td>
<td>1.22</td>
</tr>
<tr>
<td><strong>sub.</strong></td>
<td>0.69</td>
<td>0.66</td>
<td>1.25</td>
<td>1.46</td>
<td></td>
<td>0.61</td>
<td>0.61</td>
<td>1.20</td>
<td>1.18</td>
</tr>
<tr>
<td><strong>max.</strong></td>
<td>0.58</td>
<td>0.65</td>
<td>0.75</td>
<td>0.74</td>
<td></td>
<td>0.52</td>
<td>0.64</td>
<td>0.78</td>
<td>0.73</td>
</tr>
<tr>
<td><strong>avg.</strong></td>
<td>0.55</td>
<td>0.60</td>
<td>1.27</td>
<td>1.35</td>
<td></td>
<td>0.58</td>
<td>0.59</td>
<td>1.07</td>
<td>1.26</td>
</tr>
</tbody>
</table>

- modulus and ReLU perform better than tanh and LogSig

- results with pooling (\(S=2\)) are competitive with those without pooling, at significantly lower computational cost

- state-of-the-art: 0.43 \[Bruna and Mallat, 2013\]

- similar feature extraction network with directional, non-separable wavelets and no pooling

- significantly higher computational complexity
## Experiment: Handwritten digit classification

### Classification error in percent:

<table>
<thead>
<tr>
<th></th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
</tr>
</thead>
<tbody>
<tr>
<td>n.p.</td>
<td>0.57</td>
<td>0.57</td>
<td>1.35</td>
<td>1.49</td>
<td>0.51</td>
<td>0.57</td>
<td>1.12</td>
<td>1.22</td>
</tr>
<tr>
<td>sub.</td>
<td>0.69</td>
<td>0.66</td>
<td>1.25</td>
<td>1.46</td>
<td>0.61</td>
<td>0.61</td>
<td>1.20</td>
<td>1.18</td>
</tr>
<tr>
<td>max.</td>
<td>0.58</td>
<td>0.65</td>
<td>0.75</td>
<td>0.74</td>
<td>0.52</td>
<td>0.64</td>
<td>0.78</td>
<td>0.73</td>
</tr>
<tr>
<td>avg.</td>
<td>0.55</td>
<td>0.60</td>
<td>1.27</td>
<td>1.35</td>
<td>0.58</td>
<td>0.59</td>
<td>1.07</td>
<td>1.26</td>
</tr>
</tbody>
</table>

- modulus and ReLU perform better than tanh and LogSig
- results with pooling ($S = 2$) are competitive with those without pooling, at significantly lower computational cost
### Classification error in percent:

<table>
<thead>
<tr>
<th></th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
<th>abs</th>
<th>ReLU</th>
<th>tanh</th>
<th>LogSig</th>
</tr>
</thead>
<tbody>
<tr>
<td>n.p.</td>
<td>0.57</td>
<td>0.57</td>
<td>1.35</td>
<td>1.49</td>
<td>0.51</td>
<td>0.57</td>
<td>1.12</td>
<td>1.22</td>
</tr>
<tr>
<td>sub.</td>
<td>0.69</td>
<td>0.66</td>
<td>1.25</td>
<td>1.46</td>
<td>0.61</td>
<td>0.61</td>
<td>1.20</td>
<td>1.18</td>
</tr>
<tr>
<td>max.</td>
<td>0.58</td>
<td>0.65</td>
<td>0.75</td>
<td>0.74</td>
<td>0.52</td>
<td>0.64</td>
<td>0.78</td>
<td>0.73</td>
</tr>
<tr>
<td>avg.</td>
<td>0.55</td>
<td>0.60</td>
<td>1.27</td>
<td>1.35</td>
<td>0.58</td>
<td>0.59</td>
<td>1.07</td>
<td>1.26</td>
</tr>
</tbody>
</table>

- modulus and ReLU perform better than tanh and LogSig
- results with pooling ($S = 2$) are competitive with those without pooling, at significantly lower computational cost
- state-of-the-art: 0.43 [Bruna and Mallat, 2013]
  - similar feature extraction network with directional, non-separable wavelets and no pooling
  - significantly higher computational complexity
Energy decay: Related work

[Waldspurger, 2017]: Exponential energy decay

\[ W_n(f) = O(a^{-n}), \]

for some unspecified \( a > 1 \).

- 1-D wavelet filters
  - every network layer equipped with the same set of wavelets
[Waldspurger, 2017]: Exponential energy decay

\[ W_n(f) = \mathcal{O}(a^{-n}), \]

for some unspecified \( a > 1 \).

- 1-D \textit{wavelet} filters
- \textit{every} network layer equipped with the same set of wavelets
- \textit{vanishing moments} condition on the mother wavelet
Energy decay: Related work

[Waldspurger, 2017]: Exponential energy decay

\[ W_n(f) = O(a^{-n}), \]

for some unspecified \( a > 1 \).

- 1-D wavelet filters
- every network layer equipped with the same set of wavelets
- vanishing moments condition on the mother wavelet
- applies to 1-D real-valued band-limited input signals \( f \in L^2(\mathbb{R}) \)
[Czaja and Li, 2016]: Exponential energy decay

\[ W_n(f) = \mathcal{O}(a^{-n}), \]

for some unspecified \( a > 1 \).

- \( d \)-dimensional uniform covering filters (similar to Weyl-Heisenberg filters), but does not cover multi-scale filters (e.g. wavelets, ridgedelets, curvelets etc.)
- every network layer equipped with the same set of filters
Energy decay: Related work

[Czaja and Li, 2016]: Exponential energy decay

\[ W_n(f) = \mathcal{O}(a^{-n}), \]

for some unspecified \( a > 1 \).

- \( d \)-dimensional uniform covering filters (similar to Weyl-Heisenberg filters), but does not cover multi-scale filters (e.g. wavelets, ridgedeleks, curvelets etc.)
- every network layer equipped with the same set of filters
- analyticity and vanishing moments conditions on the filters
Energy decay: Related work

[Czaja and Li, 2016]: Exponential energy decay

\[ W_n(f) = \mathcal{O}(a^{-n}), \]

for some unspecified \( a > 1 \).

- \( d \)-dimensional uniform covering filters (similar to Weyl-Heisenberg filters), but does not cover multi-scale filters (e.g. wavelets, ridgedelets, curvelets etc.)
- every network layer equipped with the same set of filters
- analyticity and vanishing moments conditions on the filters
- applies to \( d \)-dimensional complex-valued input signals \( f \in L^2(\mathbb{R}^d) \)