Invariance and Stability to Deformations of Deep Convolutional Representations

Julien Mairal
Inria Grenoble

Theory of Deep Learning Workshop, ICML, 2018
This is mostly the work of Alberto Bietti


Objectives

Deep convolutional signal representations

- Are they stable to deformations?
- How can we achieve invariance to transformation groups?
- Do they preserve signal information?

Learning aspects

- Building a functional space for CNNs (or similar objects).
- Deriving a measure of model complexity.
A kernel perspective

Recipe

- Map data $x$ to **high-dimensional space**, $\Phi(x)$ in $\mathcal{H}$ (RKHS), with Hilbertian geometry (projections, barycenters, angles, . . . , exist!).
- Non-linear function $f$ in $\mathcal{H}$ becomes linear: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$.
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$.

\[\text{[Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004].}…\]
A kernel perspective

Recipe

- Map data $x$ to **high-dimensional space**, $\Phi(x)$ in $\mathcal{H}$ (RKHS), with Hilbertian geometry (projections, barycenters, angles, . . ., exist!).
- Non-linear function $f$ in $\mathcal{H}$ becomes linear: $f(x) = \langle f, \Phi(x) \rangle_\mathcal{H}$.
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_\mathcal{H}$.

What is the relation with deep neural networks?

- It is possible to design a RKHS $\mathcal{H}$ where a large class of deep neural networks live [Mairal, 2016].

$$f(x) = \sigma_k(W_k\sigma_{k-1}(W_{k-1} \ldots \sigma_2(W_2\sigma_1(W_1x)) \ldots)) = \langle f, \Phi(x) \rangle_\mathcal{H}.$$  

- This is the construction of “**convolutional kernel networks**”.

[Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004]…
A kernel perspective

Recipe

- Map data $x$ to **high-dimensional space**, $\Phi(x)$ in $\mathcal{H}$ (RKHS), with Hilbertian geometry (projections, barycenters, angles, . . . , exist!).
- Non-linear function $f$ in $\mathcal{H}$ becomes linear: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$.
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$.

Why do we care?

- $\Phi(x)$ is related to the **network architecture** and is **independent of training data**. Is it stable? Does it lose signal information?
- $f$ is a **predictive model**. Can we control its stability?

$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\Phi(x) - \Phi(x')\|_{\mathcal{H}}.$$  

- $\|f\|_{\mathcal{H}}$ controls both **stability and generalization**!

[Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004]...
A signal processing perspective
plus a bit of harmonic analysis

- Consider images defined on a **continuous** domain $\Omega = \mathbb{R}^d$.
- $\tau : \Omega \to \Omega$: $C^1$-diffeomorphism.
- $L_\tau x(u) = x(u - \tau(u))$: action operator.
- Much richer group of transformations than translations.

A signal processing perspective
plus a bit of harmonic analysis

- Consider images defined on a \textbf{continuous} domain \( \Omega = \mathbb{R}^d \).
- \( \tau : \Omega \rightarrow \Omega \): \( C^1 \)-diffeomorphism.
- \( L_\tau x(u) = x(u - \tau(u)) \): action operator.
- Much richer group of transformations than translations.

Relation with deep convolutional representations
Stability to deformations studied for wavelet-based scattering transform.

A signal processing perspective
plus a bit of harmonic analysis

- Consider images defined on a **continuous** domain $\Omega = \mathbb{R}^d$.
- $\tau : \Omega \to \Omega$: $C^1$-diffeomorphism.
- $L_\tau x(u) = x(u - \tau(u))$: action operator.
- Much richer group of transformations than translations.

**Definition of stability**

- Representation $\Phi(\cdot)$ is **stable** [Mallat, 2012] if:

\[
\|\Phi(L_\tau x) - \Phi(x)\| \leq (C_1\|\nabla \tau\|_\infty + C_2\|\tau\|_\infty)\|x\|.
\]

- $\|\nabla \tau\|_\infty = \sup_u \|\nabla \tau(u)\|$ controls deformation.
- $\|\tau\|_\infty = \sup_u |\tau(u)|$ controls translation.
- $C_2 \to 0$: translation invariance.

Summary of our results

Multi-layer construction of the RKHS $\mathcal{H}$

- Contains CNNs with smooth homogeneous activations functions.
- CKNs provide approximation of the kernel mapping $\Phi$. 
Summary of our results

Multi-layer construction of the RKHS $\mathcal{H}$
- Contains CNNs with smooth homogeneous activations functions.
- CKNs provide approximation of the kernel mapping $\Phi$.

Signal representation
- **Signal preservation** of the multi-layer kernel mapping $\Phi$.
- Conditions of **non-trivial stability** for $\Phi$.
- Constructions to achieve **group invariance**.
Summary of our results

Multi-layer construction of the RKHS $\mathcal{H}$
- Contains CNNs with smooth homogeneous activations functions.
- CKNs provide approximation of the kernel mapping $\Phi$.

Signal representation
- **Signal preservation** of the multi-layer kernel mapping $\Phi$.
- Conditions of **non-trivial stability** for $\Phi$.
- Constructions to achieve **group invariance**.

On learning
- Bounds on the RKHS norm $\| . \|_{\mathcal{H}}$ to control **stability and generalization** of a predictive model $f$.

\[ |f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\Phi(x) - \Phi(x')\|_{\mathcal{H}}. \]
Outline

1. Construction of the multi-layer convolutional representation

2. Invariance and stability

3. Learning aspects: model complexity
A generic deep convolutional representation

Initial map $x_0$ in $L^2(\Omega, \mathcal{H}_0)$

- $x_0 : \Omega \to \mathcal{H}_0$: continuous input signal
- $u \in \Omega = \mathbb{R}^d$: location ($d = 2$ for images).
- $x_0(u) \in \mathcal{H}_0$: input value at location $u$ ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images).
A generic deep convolutional representation

Initial map $x_0$ in $L^2(\Omega, \mathcal{H}_0)$

$x_0 : \Omega \to \mathcal{H}_0$: continuous input signal

- $u \in \Omega = \mathbb{R}^d$: location
- $x_0(u) \in \mathcal{H}_0$: input value at location $u$ ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images).

Building map $x_k$ in $L^2(\Omega, \mathcal{H}_k)$ from $x_{k-1}$ in $L^2(\Omega, \mathcal{H}_{k-1})$

$x_k : \Omega \to \mathcal{H}_k$: feature map at layer $k$

$$P_k x_{k-1}.$$ 

- $P_k$: patch extraction operator, extract small patch of feature map $x_{k-1}$ around each point $u$ ($P_k x_{k-1}(u)$ is a patch centered at $u$).
A generic deep convolutional representation

Initial map $x_0$ in $L^2(\Omega, \mathcal{H}_0)$

$x_0 : \Omega \to \mathcal{H}_0$: **continuous** input signal

- $u \in \Omega = \mathbb{R}^d$: location
- $x_0(u) \in \mathcal{H}_0$: input value at location $u$ ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images).

Building map $x_k$ in $L^2(\Omega, \mathcal{H}_k)$ from $x_{k-1}$ in $L^2(\Omega, \mathcal{H}_{k-1})$

$x_k : \Omega \to \mathcal{H}_k$: **feature map** at layer $k$

$$M_k P_k x_{k-1}.$$

- $P_k$: **patch extraction** operator, extract small patch of feature map $x_{k-1}$ around each point $u$ ($P_k x_{k-1}(u)$ is a patch centered at $u$).
- $M_k$: **non-linear mapping** operator, maps each patch to a new Hilbert space $\mathcal{H}_k$ with a **pointwise** non-linear function $\varphi_k(\cdot)$. 
A generic deep convolutional representation

Initial map $x_0$ in $L^2(\Omega, \mathcal{H}_0)$

$x_0 : \Omega \to \mathcal{H}_0$: continuous input signal

- $u \in \Omega = \mathbb{R}^d$: location
- $x_0(u) \in \mathcal{H}_0$: input value at location $u$ ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images).

Building map $x_k$ in $L^2(\Omega, \mathcal{H}_k)$ from $x_{k-1}$ in $L^2(\Omega, \mathcal{H}_{k-1})$

$x_k : \Omega \to \mathcal{H}_k$: feature map at layer $k$

$$x_k = A_k M_k P_k x_{k-1}.$$ 

- $P_k$: patch extraction operator, extract small patch of feature map $x_{k-1}$ around each point $u$ ($P_k x_{k-1}(u)$ is a patch centered at $u$).
- $M_k$: non-linear mapping operator, maps each patch to a new Hilbert space $\mathcal{H}_k$ with a pointwise non-linear function $\varphi_k(\cdot)$.
- $A_k$: (linear) pooling operator at scale $\sigma_k$. 
A generic deep convolutional representation

\[ x_k := A_k M_k P_k x_{k-1} : \Omega \to \mathcal{H}_k \]

\[ x_k(w) = A_k M_k P_k x_{k-1}(w) \in \mathcal{H}_k \]

linear pooling

\[ M_k P_k x_{k-1} : \Omega \to \mathcal{H}_k \]

kernel mapping

\[ M_k P_k x_{k-1}(v) = \varphi_k(P_k x_{k-1}(v)) \in \mathcal{H}_k \]

\[ P_k x_{k-1}(v) \in \mathcal{P}_k \text{ (patch extraction)} \]

\[ x_{k-1}(u) \in \mathcal{H}_{k-1} \]

\[ x_{k-1} : \Omega \to \mathcal{H}_{k-1} \]
Patch extraction operator $P_k$

$$P_k x_{k-1}(u) := (v \in S_k \mapsto x_{k-1}(u + v)) \in P_k = \mathcal{H}^{S_k}_{k-1}.$$ 

$S_k$: patch shape, e.g. box.

$P_k$ is **linear**, and **preserves the norm**: $\|P_k x_{k-1}\| = \|x_{k-1}\|$.

Norm of a map: $\|x\|^2 = \int_\Omega \|x(u)\|^2 du < \infty$ for $x$ in $L^2(\Omega, \mathcal{H})$. 
Non-linear pointwise mapping operator $M_k$

$$M_k P_k x_{k-1}(u) := \varphi_k(P_k x_{k-1}(u)) \in \mathcal{H}_k.$$
Non-linear pointwise mapping operator $M_k$

\[ M_k P_k x_{k-1}(u) := \varphi_k(P_k x_{k-1}(u)) \in \mathcal{H}_k. \]

- $\varphi_k : \mathcal{P}_k \to \mathcal{H}_k$ pointwise non-linearity on patches.
- We assume non-expansivity: for $z, z' \in \mathcal{P}_k$

\[ \| \varphi_k(z) \| \leq \| z \| \quad \text{and} \quad \| \varphi_k(z) - \varphi_k(z') \| \leq \| z - z' \|. \]

- $M_k$ then satisfies, for $x, x' \in L^2(\Omega, \mathcal{P}_k)$

\[ \| M_k x \| \leq \| x \| \quad \text{and} \quad \| M_k x - M_k x' \| \leq \| x - x' \|. \]
Non-linear pointwise mapping operator $M_k$

$$M_k P_k x_{k-1}(u) := \varphi_k(P_k x_{k-1}(u)) \in \mathcal{H}_k.$$ 

- $\varphi_k : \mathcal{P}_k \rightarrow \mathcal{H}_k$ pointwise non-linearity on patches.
- or instead: for $z, z' \in \mathcal{P}_k$
  $$\|\varphi_k(z)\| \leq \rho_k \|z\| \quad \text{and} \quad \|\varphi_k(z) - \varphi_k(z')\| \leq \rho_k \|z - z'\|.$$ 

- $M_k$ then satisfies, for $x, x' \in L^2(\Omega, \mathcal{P}_k)$
  $$\|M_k x\| \leq \rho_k \|x\| \quad \text{and} \quad \|M_k x - M_k x'\| \leq \rho_k \|x - x'\|.$$ 

- but at some point, we pay a “price” in $\Pi_{i=1}^k \rho_i$. 

$\varphi_k$ from kernels

- Kernel mapping of **homogeneous dot-product kernels**:

  $$K_k(z, z') = \|z\|\|z'\| \kappa_k \left( \frac{\langle z, z' \rangle}{\|z\|\|z'\|} \right) = \langle \varphi_k(z), \varphi_k(z') \rangle.$$ 

- $\kappa_k(u) = \sum_{j=0}^{\infty} b_j u^j$ with $b_j \geq 0$, $\kappa_k(1) = 1$.
- $\|\varphi_k(z)\| = K_k(z, z)^{1/2} = \|z\|$ (norm preservation).
- $\|\varphi_k(z) - \varphi_k(z')\| \leq \|z - z'\|$ if $\kappa'_k(1) \leq 1$ (non-expansiveness).
$\varphi_k$ from kernels

- Kernel mapping of homogeneous dot-product kernels:

$$K_k(z, z') = \|z\|\|z'\|\kappa_k\left(\frac{\langle z, z' \rangle}{\|z\|\|z'\|}\right) = \langle \varphi_k(z), \varphi_k(z') \rangle.$$  

- $\kappa_k(u) = \sum_{j=0}^{\infty} b_j u^j$ with $b_j \geq 0$, $\kappa_k(1) = 1$.
- $\|\varphi_k(z)\| = K_k(z, z)^{1/2} = \|z\|$  \hspace{2cm} (norm preservation).
- $\|\varphi_k(z) - \varphi_k(z')\| \leq \|z - z'\|$ if $\kappa_k'(1) \leq 1$  \hspace{2cm} (non-expansiveness).

Examples

- $\kappa_{\text{exp}}(\langle z, z' \rangle) = e^{\langle z, z' \rangle} - 1 = e^{-\frac{1}{2}\|z-z'\|^2}$  \hspace{2cm} (if $\|z\| = \|z'\| = 1$).
- $\kappa_{\text{inv-poly}}(\langle z, z' \rangle) = \frac{1}{2-\langle z, z' \rangle}$.

Pooling operator $A_k$

$$x_k(u) = A_k M_k P_k x_{k-1}(u) = \int_{\mathbb{R}^d} h_{\sigma_k}(u - v) M_k P_k x_{k-1}(v) dv \in \mathcal{H}_k.$$

$x_k := A_k M_k P_k x_{k-1} : \Omega \to \mathcal{H}_k$

$x_k(w) = A_k M_k P_k x_{k-1}(w) \in \mathcal{H}_k$

linear pooling

$M_k P_k x_{k-1} : \Omega \to \mathcal{H}_k$

$x_{k-1} : \Omega \to \mathcal{H}_{k-1}$
Pooling operator $A_k$

\[ x_k(u) = A_k M_k P_k x_{k-1}(u) = \int_{\mathbb{R}^d} h_{\sigma_k}(u - v) M_k P_k x_{k-1}(v) dv \in \mathcal{H}_k. \]

- $h_{\sigma_k}$: pooling filter at scale $\sigma_k$.
- $h_{\sigma_k}(u) := \sigma_k^{-d} h(u/\sigma_k)$ with $h(u)$ **Gaussian**.
- **linear, non-expansive operator**: $\| A_k \| \leq 1$ (operator norm).
Recap: $P_k$, $M_k$, $A_k$

$x_k := A_k M_k P_k x_{k-1} : \Omega \rightarrow \mathcal{H}_k$

$M_k P_k x_{k-1} : \Omega \rightarrow \mathcal{H}_k$

$M_k P_k x_{k-1}(v) = \varphi_k(P_k x_{k-1}(v)) \in \mathcal{H}_k$

$P_k x_{k-1}(v) \in \mathcal{P}_k$ (patch extraction)

$x_{k-1}(u) \in \mathcal{H}_{k-1}$

$x_{k-1} : \Omega \rightarrow \mathcal{H}_{k-1}$

linear pooling

kernel mapping

Julien Mairal

Invariance and stability of DL
Multilayer construction

Assumption on $x_0$

- $x_0$ is typically a **discrete** signal acquired with physical device.
- Natural assumption: $x_0 = A_0 x$, with $x$ the original continuous signal, $A_0$ local integrator with scale $\sigma_0$ (**anti-aliasing**).
Multilayer construction

Assumption on $x_0$

- $x_0$ is typically a **discrete** signal acquired with physical device.
- Natural assumption: $x_0 = A_0 x$, with $x$ the original continuous signal, $A_0$ local integrator with scale $\sigma_0$ (**anti-aliasing**).

Multilayer representation

$$\Phi_n(x) = A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 x_0 \in L^2(\Omega, \mathcal{H}_n).$$

- $S_k, \sigma_k$ grow exponentially in practice (i.e., fixed with subsampling).
Multilayer construction

Assumption on $x_0$

- $x_0$ is typically a discrete signal acquired with physical device.
- Natural assumption: $x_0 = A_0 x$, with $x$ the original continuous signal, $A_0$ local integrator with scale $\sigma_0$ (anti-aliasing).

Multilayer representation

$$
\Phi_n(x) = A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 x_0 \in L^2(\Omega, \mathcal{H}_n).
$$

- $S_k$, $\sigma_k$ grow exponentially in practice (i.e., fixed with subsampling).

Prediction layer

- e.g., linear $f(x) = \langle w, \Phi_n(x) \rangle$.
- “linear kernel” $K(x, x') = \langle \Phi_n(x), \Phi_n(x') \rangle = \int_\Omega \langle x_n(u), x'_n(u) \rangle du$. 
Discretization and signal preservation: example in 1D

- Discrete signal $\bar{x}_k$ in $\ell^2(\mathbb{Z}, \mathcal{H}_k)$ vs continuous ones $x_k$ in $L^2(\mathbb{R}, \mathcal{H}_k)$.
- $\bar{x}_k$: subsampling factor $s_k$ after pooling with scale $\sigma_k \approx s_k$:

  $$\bar{x}_k[n] = \bar{A}_k \bar{M}_k \bar{P}_k \bar{x}_{k-1}[ns_k].$$

Warning: no claim that recovery is practical and/or stable.
Discretization and signal preservation: example in 1D

- Discrete signal $\bar{x}_k$ in $\ell^2(\mathbb{Z}, \bar{H}_k)$ vs continuous ones $x_k$ in $L^2(\mathbb{R}, \mathcal{H}_k)$.
- $\bar{x}_k$: subsampling factor $s_k$ after pooling with scale $\sigma_k \approx s_k$:
  $$\bar{x}_k[n] = \bar{A}_k \bar{M}_k \bar{P}_k \bar{x}_{k-1}[ns_k].$$

- **Claim**: We can recover $\bar{x}_{k-1}$ from $\bar{x}_k$ if factor $s_k \leq \text{patch size}$.  

Warning: no claim that recovery is practical and/or stable.
Discretization and signal preservation: example in 1D

- Discrete signal $\bar{x}_k$ in $\ell^2(\mathbb{Z}, \mathcal{H}_k)$ vs continuous ones $x_k$ in $L^2(\mathbb{R}, \mathcal{H}_k)$.
- $\bar{x}_k$: subsampling factor $s_k$ after pooling with scale $\sigma_k \approx s_k$:
  \[
  \bar{x}_k[n] = A_k M_k P_k \bar{x}_{k-1}[ns_k].
  \]

- **Claim**: We can recover $\bar{x}_{k-1}$ from $\bar{x}_k$ if factor $s_k \leq \text{patch size}$.
- **How**? Recover patches with **linear functions** (contained in $\mathcal{H}_k$)
  \[
  \langle f_w, M_k P_k \bar{x}_{k-1}(u) \rangle = f_w(P_k \bar{x}_{k-1}(u)) = \langle w, P_k \bar{x}_{k-1}(u) \rangle,
  \]
  and
  \[
  P_k \bar{x}_{k-1}(u) = \sum_{w \in B} \langle f_w, M_k P_k \bar{x}_{k-1}(u) \rangle w.
  \]

---

Warning: no claim that recovery is practical and/or stable.
Discretization and signal preservation: example in 1D

- Discrete signal \( \bar{x}_k \) in \( \ell^2(\mathbb{Z}, \mathcal{H}_k) \) vs continuous ones \( x_k \) in \( L^2(\mathbb{R}, \mathcal{H}_k) \).
- \( \bar{x}_k \): subsampling factor \( s_k \) after pooling with scale \( \sigma_k \approx s_k \):

\[
\bar{x}_k[n] = A_k \bar{M}_k \bar{P}_k \bar{x}_{k-1}[ns_k].
\]

- **Claim:** We can recover \( \bar{x}_{k-1} \) from \( \bar{x}_k \) if factor \( s_k \leq \text{patch size} \).
- **How?** Recover patches with **linear functions** (contained in \( \mathcal{H}_k \))

\[
\langle f_w, \bar{M}_k \bar{P}_k \bar{x}_{k-1}(u) \rangle = f_w(\bar{P}_k \bar{x}_{k-1}(u)) = \langle w, \bar{P}_k \bar{x}_{k-1}(u) \rangle,
\]

and

\[
\bar{P}_k \bar{x}_{k-1}(u) = \sum_{w \in B} \langle f_w, \bar{M}_k \bar{P}_k \bar{x}_{k-1}(u) \rangle w.
\]

**Warning:** no claim that recovery is practical and/or stable.
Discretization and signal preservation: example in 1D

\[ \bar{x}_{k-1} \]

\[ \bar{A}_k \bar{x}_{k-1} \]

\[ \bar{x}_k \]

\[ \bar{A}_k \bar{M}_k \bar{P}_k \bar{x}_{k-1} \]

\[ \bar{M}_k \bar{P}_k \bar{x}_{k-1} \]

\[ \bar{x}_{k-1} \]

\[ \bar{P}_k \bar{x}_{k-1}(u) \in \mathcal{P}_k \]

deconvolution

recovery with linear measurements

downsampling

linear pooling

dot-product kernel
Outline

1. Construction of the multi-layer convolutional representation

2. Invariance and stability

3. Learning aspects: model complexity
Invariance, definitions

- $\tau : \Omega \to \Omega$: $C^1$-diffeomorphism with $\Omega = \mathbb{R}^d$.
- $L_\tau x(u) = x(u - \tau(u))$: action operator.
- Much richer group of transformations than translations.

Invariance, definitions

- \( \tau : \Omega \rightarrow \Omega \): \( C^1 \)-diffeomorphism with \( \Omega = \mathbb{R}^d \).
- \( L_\tau x(u) = x(u - \tau(u)) \): action operator.
- Much richer group of transformations than translations.

Definition of stability

- Representation \( \Phi(\cdot) \) is stable [Mallat, 2012] if:
  \[
  \| \Phi(L_\tau x) - \Phi(x) \| \leq (C_1 \| \nabla \tau \|_\infty + C_2 \| \tau \|_\infty) \| x \|.
  \]

- \( \| \nabla \tau \|_\infty = \sup_u \| \nabla \tau(u) \| \) controls deformation.
- \( \| \tau \|_\infty = \sup_u |\tau(u)| \) controls translation.
- \( C_2 \rightarrow 0 \): translation invariance.

Warmup: translation invariance

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve translation invariance?

- Translation: \( L_c x(u) = x(u - c). \)
Warmup: translation invariance

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve translation invariance?

- Translation: \( L_c x(u) = x(u - c) \).
- Equivariance - all operators commute with \( L_c \): \( \Box L_c = L_c \Box \).

\[
\| \Phi_n(L_c x) - \Phi_n(x) \| = \| L_c \Phi_n(x) - \Phi_n(x) \| \\
\leq \| L_c A_n - A_n \| \cdot \| M_n P_n \Phi_{n-1}(x) \| \\
\leq \| L_c A_n - A_n \| \| x \|. 
\]
Warmup: translation invariance

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve translation invariance?

- **Translation**: \( L_c x(u) = x(u - c) \).
- **Equivariance** - all operators commute with \( L_c \): \( \Box L_c = L_c \Box \).

\[
\| \Phi_n(L_c x) - \Phi_n(x) \| \leq \| L_c \Phi_n(x) - \Phi_n(x) \| \\
\leq \| L_c A_n - A_n \| \cdot \| M_n P_n \Phi_{n-1}(x) \| \\
\leq \| L_c A_n - A_n \| \| x \|.
\]

- **Mallat [2012]**: \( \| L_\tau A_n - A_n \| \leq \frac{C_2}{\sigma_n} \| \tau \|_\infty \) (operator norm).
Warmup: translation invariance

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve translation invariance?

- **Translation**: \( L_c x(u) = x(u - c). \)
- **Equivariance** - all operators commute with \( L_c \): \( \Box L_c = L_c \Box \).

\[
\| \Phi_n(L_c x) - \Phi_n(x) \| = \| L_c \Phi_n(x) - \Phi_n(x) \| \\
\leq \| L_c A_n - A_n \| \cdot \| M_n P_n \Phi_{n-1}(x) \| \\
\leq \| L_c A_n - A_n \| \| x \|.
\]

- Mallat [2012]: \( \| L_c A_n - A_n \| \leq \frac{C_2}{\sigma_n} c \) (operator norm).
- **Scale** \( \sigma_n \) of the last layer controls translation invariance.
Stability to deformations

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve stability to deformations?

- Patch extraction \( P_k \) and pooling \( A_k \) do not commute with \( L_\tau \)!
Stability to deformations

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve stability to deformations?

- Patch extraction \( P_k \) and pooling \( A_k \) do not commute with \( L_\tau \)!
- \[ \| A_k L_\tau - L_\tau A_k \| \leq C_1 \| \nabla \tau \|_\infty \] [from Mallat, 2012].
Stability to deformations

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve stability to deformations?

- Patch extraction \( P_k \) and pooling \( A_k \) do not commute with \( L_\tau \)!
- \[ ||[A_k, L_\tau]|| \leq C_1 ||\nabla \tau||_\infty \] [from Mallat, 2012].
Stability to deformations

Representation

$$\Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x.$$ 

How to achieve stability to deformations?

- Patch extraction $P_k$ and pooling $A_k$ do not commute with $L_\tau$!
- $||[A_k, L_\tau]|| \leq C_1 ||\nabla \tau||_\infty$ [from Mallat, 2012].
- But: $[P_k, L_\tau]$ is unstable at high frequencies!
Stability to deformations

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve stability to deformations?

- Patch extraction \( P_k \) and pooling \( A_k \) do not commute with \( L_\tau \)!
- \( \|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty \) [from Mallat, 2012].
- But: \([P_k, L_\tau]\) is unstable at high frequencies!
- Adapt to current layer resolution, patch size controlled by \( \sigma_{k-1} \):

\[ \|[P_k A_{k-1}, L_\tau]\| \leq C_{1,\kappa} \|\nabla \tau\|_\infty \sup_{u \in S_k} |u| \leq \kappa \sigma_{k-1} \]
Stability to deformations

Representation

\[ \Phi_n(x) \triangleq A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 A_0 x. \]

How to achieve stability to deformations?

- Patch extraction \( P_k \) and pooling \( A_k \) **do not commute** with \( L_\tau \)!
- \( \| [A_k, L_\tau] \| \leq C_1 \| \nabla \tau \|_\infty \) [from Mallat, 2012].
- But: \( [P_k, L_\tau] \) is **unstable** at high frequencies!
- Adapt to **current layer resolution**, patch size controlled by \( \sigma_{k-1} \):
  \[ \| [P_k A_{k-1}, L_\tau] \| \leq C_{1, \kappa} \| \nabla \tau \|_\infty \sup_{u \in S_k} |u| \leq \kappa \sigma_{k-1} \]

- \( C_{1, \kappa} \) grows as \( \kappa^{d+1} \) \( \implies \) more stable with **small patches** (e.g., 3x3, VGG et al.).
Stability to deformations: final result

Theorem

If $\|\nabla \tau\|_\infty \leq 1/2$,

$$
\|\Phi_n(L_\tau x) - \Phi_n(x)\| \leq \left(C_{1,\kappa} (n + 1) \|\nabla \tau\|_\infty + \frac{C_2}{\sigma_n} \|\tau\|_\infty\right) \|x\|
$$

- translation invariance: large $\sigma_n$.
- stability: small patch sizes.
- signal preservation: subsampling factor $\approx$ patch size.
- $\implies$ needs several layers.

related work on stability [Wiatowski and Bölcskei, 2017]
Stability to deformations: final result

Theorem

If \( \| \nabla \tau \|_\infty \leq 1/2 \),

\[
\| \Phi_n (L_\tau x) - \Phi_n (x) \| \leq \left( C_{1, \kappa} (n + 1) \| \nabla \tau \|_\infty + \frac{C_2}{\sigma_n} \| \tau \|_\infty \right) \| x \|.
\]

- translation invariance: large \( \sigma_n \).
- stability: small patch sizes.
- signal preservation: subsampling factor \( \approx \) patch size.
- \( \implies \) needs several layers.
- requires additional discussion to make stability non-trivial.

related work on stability [Wiatowski and Bölcskei, 2017]
Stability to deformations: final result

Theorem

If \( \| \nabla \tau \|_\infty \leq 1/2 \),

\[
\| \Phi_n(L \tau x) - \Phi_n(x) \| \leq \prod_k \rho_k \left( C_{1,\kappa} (n + 1) \| \nabla \tau \|_\infty + \frac{C_2}{\sigma_n} \| \tau \|_\infty \right) \| x \|.
\]

- translation invariance: large \( \sigma_n \).
- stability: small patch sizes.
- signal preservation: subsampling factor \( \approx \) patch size.
- \( \implies \) needs several layers.
- requires additional discussion to make stability non-trivial.
- (also valid for generic CNNs with ReLUs: multiply by \( \prod_k \rho_k = \prod_k \| W_k \| \), but no signal preservation).

related work on stability [Wiatowski and Bölcskei, 2017]
Beyond the translation group

Can we achieve invariance to other groups?

- Group action: \( L_g x(u) = x(g^{-1}u) \) (e.g., rotations, reflections).
- Feature maps \( x(u) \) defined on \( u \in G \) (\( G \): locally compact group).
Beyond the translation group

Can we achieve invariance to other groups?

- **Group action**: \( L_g x(u) = x(g^{-1}u) \) (e.g., rotations, reflections).
- **Feature maps** \( x(u) \) defined on \( u \in G \) (\( G \): locally compact group).

Recipe: Equivariant inner layers + global pooling in last layer

- **Patch extraction**:
  \[
P x(u) = (x(uv))_{v \in S}.
  \]

- **Non-linear mapping**: equivariant because pointwise!
- **Pooling** (\( \mu \): left-invariant Haar measure):
  \[
  A x(u) = \int_G x(uv) h(v) d\mu(v) = \int_G x(v) h(u^{-1}v) d\mu(v).
  \]

related work [Sifre and Mallat, 2013, Cohen and Welling, 2016, Raj et al., 2016]...
Group invariance and stability

Previous construction is similar to Cohen and Welling [2016] for CNNs.

A case of interest: the roto-translation group

- $G = \mathbb{R}^2 \rtimes SO(2)$ (mix of translations and rotations).
- **Stability** with respect to the translation group.
- **Global invariance** to rotations (only global pooling at final layer).
  - Inner layers: only pool on translation group.
  - Last layer: global pooling on rotations.
  - Cohen and Welling [2016]: pooling on rotations in inner layers hurts performance on Rotated MNIST.
Outline

1. Construction of the multi-layer convolutional representation

2. Invariance and stability

3. Learning aspects: model complexity
RKHS of patch kernels $K_k$

$$K_k(z, z') = \|z\|\|z'\|\kappa\left(\frac{\langle z, z' \rangle}{\|z\|\|z'\|}\right), \quad \kappa(u) = \sum_{j=0}^{\infty} b_j u^j.$$ 

What does the RKHS contain?

Homogeneous version of [Zhang et al., 2016, 2017]
RKHS of patch kernels $K_k$

$$K_k(z, z') = \|z\|\|z'\|\kappa\left(\frac{\langle z, z' \rangle}{\|z\|\|z'\|}\right), \quad \kappa(u) = \sum_{j=0}^{\infty} b_j u^j.$$ 

What does the RKHS contain?

- RKHS contains **homogeneous functions**:

  $$f : z \mapsto \|z\|\sigma\left(\frac{\langle g, z \rangle}{\|z\|}\right).$$

Homogeneous version of [Zhang et al., 2016, 2017]
RKHS of patch kernels $K_k$:

$$K_k(z, z') = \|z\|\|z'\|\kappa\left(\frac{\langle z, z' \rangle}{\|z\|\|z'\|}\right), \quad \kappa(u) = \sum_{j=0}^{\infty} b_j u^j.$$

What does the RKHS contain?

- RKHS contains **homogeneous functions**:
  $$f : z \mapsto \|z\| \sigma(\langle g, z \rangle / \|z\|).$$

- **Smooth activations**: $\sigma(u) = \sum_{j=0}^{\infty} a_j u^j$ with $a_j \geq 0$.

- **Norm**: $\|f\|_{\mathcal{H}_k}^2 \leq C_{\sigma}^2(\|g\|^2) = \sum_{j=0}^{\infty} \frac{a_j^2}{b_j} \|g\|^2 < \infty.$

Homogeneous version of [Zhang et al., 2016, 2017]
RKHS of patch kernels $K_k$

Examples:

- $\sigma(u) = u$ (linear): $C_\sigma^2(\lambda^2) = O(\lambda^2)$.
- $\sigma(u) = u^p$ (polynomial): $C_\sigma^2(\lambda^2) = O(\lambda^{2p})$.
- $\sigma \approx \sin, \text{sigmoid, smooth ReLU}$: $C_\sigma^2(\lambda^2) = O(e^{c\lambda^2})$. 

![Graphs of functions $f: x \mapsto \sigma(x)$ and $f: x \mapsto |x|\sigma(wx/|x|)$ with various activation functions and parameters.](image-url)
Constructing a CNN in the RKHS $\mathcal{H}_K$

Some CNNs live in the RKHS: “linearization” principle

$$f(x) = \sigma_k(W_k \sigma_{k-1}(W_{k-1} \ldots \sigma_2(W_2 \sigma_1(W_1 x)) \ldots)) = \langle f, \Phi(x) \rangle_{\mathcal{H}}.$$
Constructing a CNN in the RKHS $\mathcal{H}_K$

Some CNNs live in the RKHS: “linearization” principle

$$f(x) = \sigma_k(W_k\sigma_{k-1}(W_{k-1} \cdots \sigma_2(W_2\sigma_1(W_1x)) \cdots)) = \langle f, \Phi(x) \rangle_{\mathcal{H}}.$$ 

- Consider a CNN with filters $W_k^{ij}(u), u \in S_k$.
  - $k$: layer;
  - $i$: index of filter;
  - $j$: index of input channel.

- “Smooth homogeneous” activations $\sigma$.

- The CNN can be constructed hierarchically in $\mathcal{H}_K$.

- Norm (linear layers):
  $$\|f\sigma\|^2 \leq \|W_{n+1}\|^2_2 \cdot \|W_n\|^2_2 \cdot \|W_{n-1}\|^2_2 \cdots \|W_1\|^2_2.$$ 

- Linear layers: product of spectral norms.
Link with generalization

Direct application of classical generalization bounds

- Simple bound on Rademacher complexity for linear/kernel methods:

\[ \mathcal{F}_B = \{ f \in \mathcal{H}_K, \|f\| \leq B \} \implies \text{Rad}_N(\mathcal{F}_B) \leq O \left( \frac{BR}{\sqrt{N}} \right). \]
Direct application of classical generalization bounds

- Simple bound on Rademacher complexity for linear/kernel methods:

\[ \mathcal{F}_B = \{ f \in \mathcal{H}_K, \|f\| \leq B \} \implies \text{Rad}_N(\mathcal{F}_B) \leq O\left(\frac{BR}{\sqrt{N}}\right). \]

- Leads to margin bound \( O(\|\hat{f}_N\| R/\gamma \sqrt{N}) \) for a learned CNN \( \hat{f}_N \) with margin (confidence) \( \gamma > 0 \).

- Related to recent generalization bounds for neural networks based on product of spectral norms [e.g., Bartlett et al., 2017, Neyshabur et al., 2018].

[see, e.g., Boucheron et al., 2005, Shalev-Shwartz and Ben-David, 2014]...
Deep convolutional representations: conclusions

Study of generic properties of signal representation

- **Deformation stability** with small patches, adapted to resolution.
- **Signal preservation** when subsampling $\leq$ patch size.
- **Group invariance** by changing patch extraction and pooling.
Deep convolutional representations: conclusions

Study of generic properties of signal representation

- **Deformation stability** with small patches, adapted to resolution.
- **Signal preservation** when subsampling \( \leq \) patch size.
- **Group invariance** by changing patch extraction and pooling.

Applies to learned models

- Same quantity \( \|f\| \) controls stability and generalization.
- “higher capacity” is needed to discriminate small deformations.
Deep convolutional representations: conclusions

Study of generic properties of signal representation

- **Deformation stability** with small patches, adapted to resolution.
- **Signal preservation** when subsampling $\leq$ patch size.
- **Group invariance** by changing patch extraction and pooling.

Applies to learned models

- Same quantity $\|f\|$ controls stability and generalization.
- “higher capacity” is needed to discriminate small deformations.

Questions:

- Better regularization?
- How does SGD control capacity in CNNs?
- What about networks with no pooling layers? ResNet?
Allez les bleus !


References II


References III


Approximate $\varphi_k(z)$ by projection (Nyström approximation) on

$$\mathcal{F} = \text{Span}(\varphi_k(z_1), \ldots, \varphi_k(z_p))$$

Figure: Nyström approximation.

[Williams and Seeger, 2001, Smola and Schölkopf, 2000, Zhang et al., 2008]...
Approximate \( \varphi_k(z) \) by **projection** (Nyström approximation) on

\[
\mathcal{F} = \text{Span}(\varphi_k(z_1), \ldots, \varphi_k(z_p)).
\]

- Leads to **tractable**, \( p \)-dimensional representation \( \psi_k(z) \).
- Norm is preserved, and projection is **non-expansive**:

\[
\|\psi_k(z) - \psi_k(z')\| = \|\Pi_k \varphi_k(z) - \Pi_k \varphi_k(z')\| \\
\leq \|\varphi_k(z) - \varphi_k(z')\| \leq \|z - z'\|.
\]

- Anchor points \( z_1, \ldots, z_p \) (≈ filters) can be **learned from data** (K-means or backprop).

[Williams and Seeger, 2001, Smola and Schölkopf, 2000, Zhang et al., 2008]...
Discussion

- norm of $\|\Phi(x)\|$ is of the same order (or close enough) to $\|x\|$.
- the kernel representation is non-expansive but not contractive

$$\sup_{x,x' \in L^2(\Omega, H_0)} \frac{\|\Phi(x) - \Phi(x')\|}{\|x - x'\|} = 1.$$