Invariance and Stability to Deformations of Deep Convolutional Representations

Julien Mairal

Inria Grenoble

International Workshop on Compact and Efficient Feature Representation and Learning in Computer Vision, ECCV, 2018

This is mostly the work of Alberto Bietti

- A. Bietti and J. Mairal. **Group Invariance, Stability to Deformations, and Complexity of Deep Convolutional Representations.** arXiv:1706.03078. 2018.
- A. Bietti and J. Mairal. Invariance and Stability of Deep Convolutional Representations. NIPS. 2017.

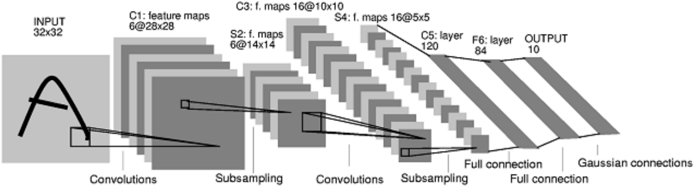
Objectives

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!).
- predictive models f in $\mathcal{H}$ are **linear forms** in $\mathcal{H}$: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$.
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}$.

[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!).
- predictive models f in $\mathcal{H}$ are **linear forms** in $\mathcal{H}$: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$.
- 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].

$$f(x) = \sigma_k(W_k \sigma_{k–1}(W_{k–1} \ldots \sigma_2(W_2 \sigma_1(W_1 x)) \ldots)) = \langle f, \Phi(x) \rangle_{\mathcal{H}}.$$

- 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!).
- predictive models f in $\mathcal{H}$ are **linear forms** in $\mathcal{H}$: $f(x) = \langle f, \Phi(x) \rangle_{\mathcal{H}}$.
- 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?

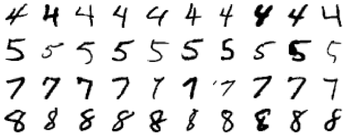
$$|f(x) - f(x')| \leq \|f\|_{\mathcal{H}} \|\Phi(x) - \Phi(x')\|_{\mathcal{H}}.$$

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

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.

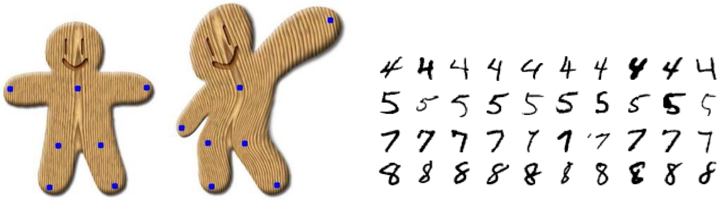


[Mallat, 2012, Bruna and Mallat, 2013, Sifre and Mallat, 2013]...

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.



Relation with deep convolutional representations

Stability to deformations studied for wavelet-based scattering transform.

[Mallat, 2012, Bruna and Mallat, 2013, Sifre and Mallat, 2013]...

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.

[Mallat, 2012, Bruna and Mallat, 2013, Sifre and Mallat, 2013]...

Summary of our results

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

- Contains CNNs with smooth homogeneous activations functions.

Summary of our results

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

- Contains CNNs with smooth homogeneous activations functions.

Signal representation

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

Summary of our results

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

- Contains CNNs with smooth homogeneous activations functions.

Signal representation

- **Signal preservation** of the multi-layer kernel mapping Φ.
- Conditions of **non-trivial stability** for Φ.
- 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 $L^2(\Omega, \mathcal{H}_0)$

$x_0 : \Omega \rightarrow \mathcal{H}_0$: **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 $(d = 2$ for images).
- $x_0(u) \in \mathcal{H}_0$: input value at location u $(\mathcal{H}_0 = \mathbb{R}^3$ for RGB images).

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 ($d = 2$ for images).
- $x_0(u) \in \mathcal{H}_0$: input value at location u ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images).

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 $L^2(\Omega, \mathcal{H}_0)$

$x_0 : \Omega \rightarrow \mathcal{H}_0$: **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).

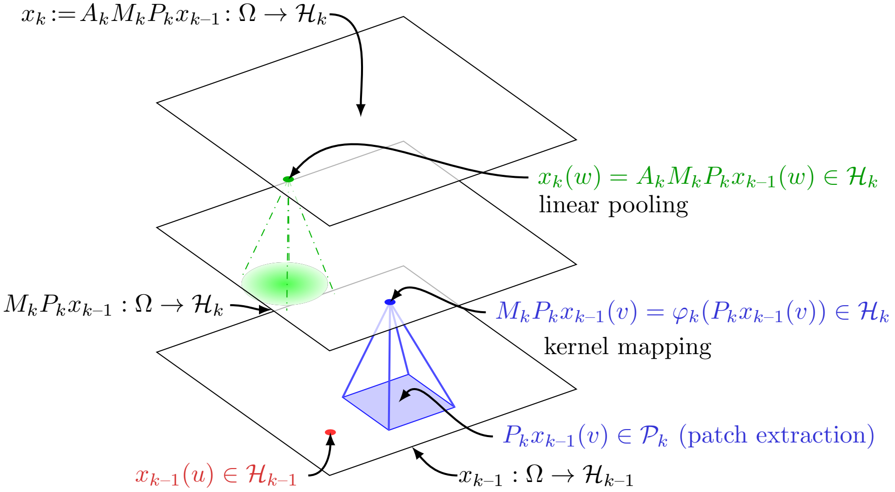
Building map x_k in $L^2(\Omega, \mathcal{H}_k)$ from x_{k-1} in $L^2(\Omega, \mathcal{H}_{k-1})$

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

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

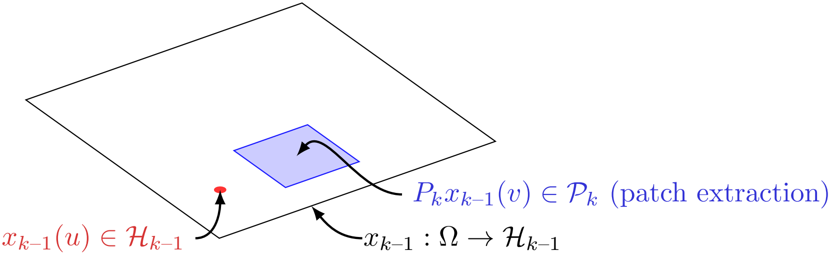
- P_k: **patch extraction** operator, extract small patch of feature map x_{k-1} around each point u ($P_k x_{k-1}(u)$ is a patch centered at u).
- M_k: **non-linear mapping** operator, maps each patch to a new Hilbert space $\mathcal{H}_k$ with a **pointwise** non-linear function $\varphi_k(\cdot)$.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

A generic deep convolutional representation



Patch extraction operator P_k

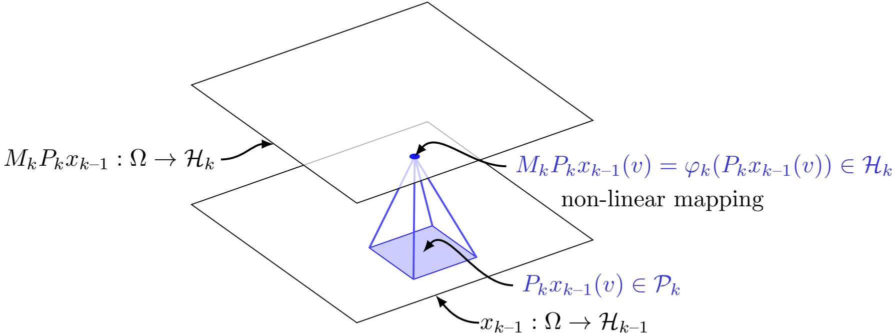
$$P_k x_{k-1}(u) := (v \in S_k \mapsto x_{k-1}(u+v)) \in \mathcal{P}_k = \mathcal{H}_{k-1}^{S_k}.$$



- 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**

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

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

$$\|M_k x\| \leq \|x\| \quad \text{and} \quad \|M_k x - M_k x'\| \leq \|x - x'\|.$$

φ_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**).

φ_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**).

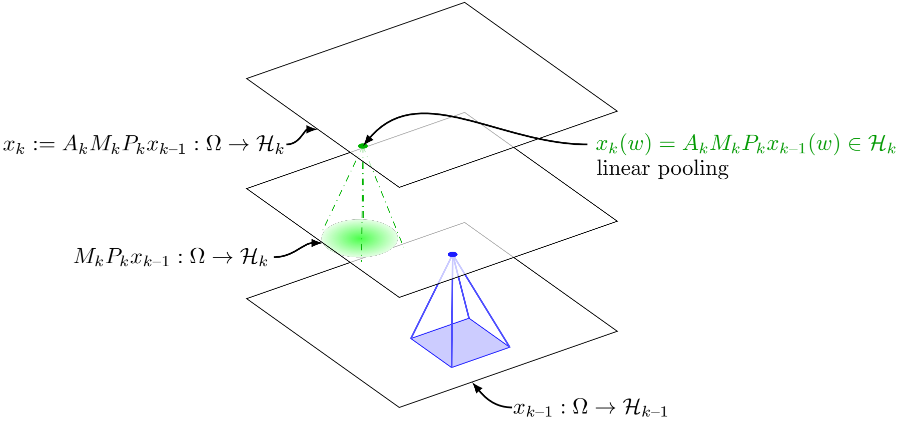
Examples

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

[Schoenberg, 1942, Scholkopf, 1997, Smola et al., 2001, Cho and Saul, 2010, Zhang et al., 2016, 2017, Daniely et al., 2016, Bach, 2017, Mairal, 2016]...

Pooling operator A_k

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

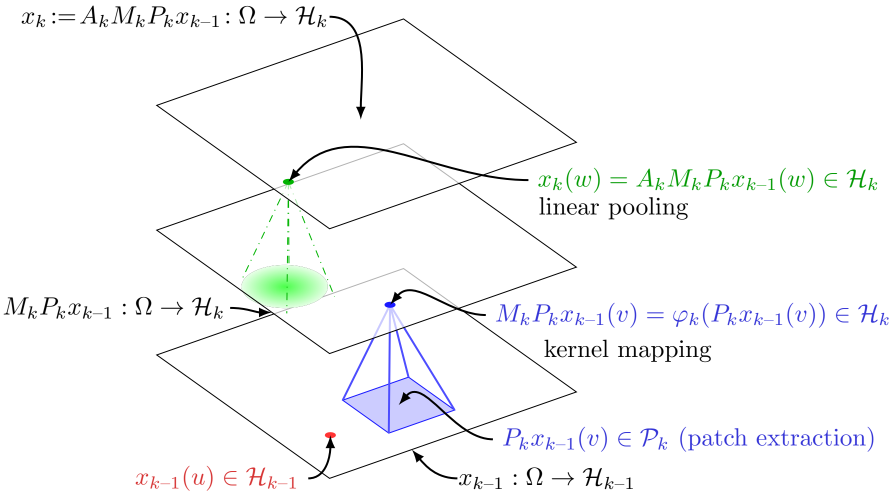


Pooling operator A_k

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

- h_{σ_k}: pooling filter at scale σ_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



Multilayer construction

Assumption on x_0

- x_0 is typically a **discrete** signal aquired with physical device.
- Natural assumption: $x_0 = A_0 x$, with x the original continuous signal, A_0 local integrator with scale σ_0 (**anti-aliasing**).

Multilayer construction

Assumption on x_0

- x_0 is typically a **discrete** signal aquired with physical device.
- Natural assumption: $x_0 = A_0 x$, with x the original continuous signal, A_0 local integrator with scale σ_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, σ_k grow exponentially in practice (i.e., fixed with subsampling).

Multilayer construction

Assumption on x_0

- x_0 is typically a **discrete** signal aquired with physical device.
- Natural assumption: $x_0 = A_0 x$, with x the original continuous signal, A_0 local integrator with scale σ_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, σ_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" $\mathcal{K}(x, x') = \langle \Phi_n(x), \Phi_n(x') \rangle = \int_\Omega \langle x_n(u), x'_n(u) \rangle du$.

Discretization and signal preservation: example in 1D

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

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

Discretization and signal preservation: example in 1D

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

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

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

Discretization and signal preservation: example in 1D

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

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

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

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

and

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

Discretization and signal preservation: example in 1D

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

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

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

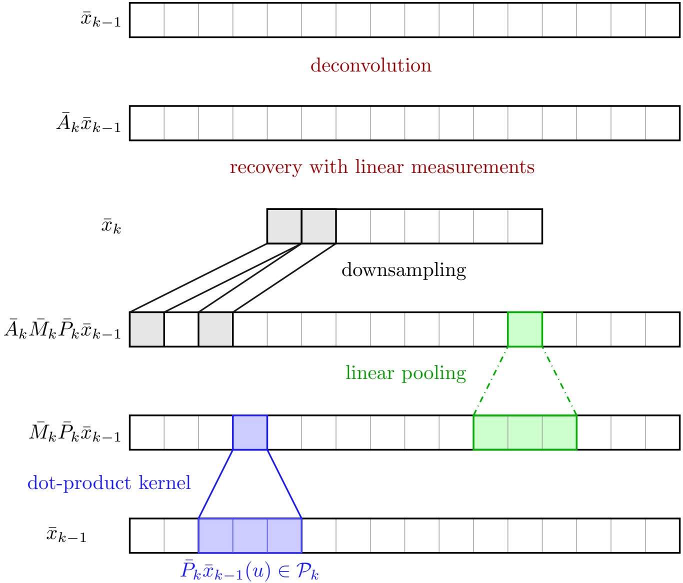
$$\langle f_w, \bar{M}_k \bar{P}_k \bar{x}_{k-1}(u) \rangle = f_w(\bar{P}_k \bar{x}_{k-1}(u)) = \langle w, \bar{P}_k \bar{x}_{k-1}(u) \rangle,$$

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



Outline

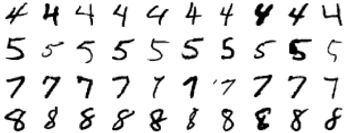
1 Construction of the multi-layer convolutional representation

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.



[Mallat, 2012, Bruna and Mallat, 2013, Sifre and Mallat, 2013]...

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.

[Mallat, 2012, Bruna and Mallat, 2013, Sifre and Mallat, 2013]...

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: $\Box L_c = L_c \Box$.

$$\begin{aligned} \|\Phi_n(L_c x) - \Phi_n(x)\| &= \|L_c \Phi_n(x) - \Phi_n(x)\| \\ &\leq \|L_c A_n - A_n\| \cdot \|M_n P_n \Phi_{n-1}(x)\| \\ &\leq \|L_c A_n - A_n\| \|x\|. \end{aligned}$$

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: $\square L_c = L_c \square$.

$$\begin{aligned}
\|\Phi_n(L_c x) - \Phi_n(x)\| &= \|L_c \Phi_n(x) - \Phi_n(x)\| \\
&\leq \|L_c A_n - A_n\| \cdot \|M_n P_n \Phi_{n-1}(x)\| \\
&\leq \|L_c A_n - A_n\| \|x\|.
\end{aligned}$$

- 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: $\square L_c = L_c \square$.

$$\begin{aligned} \|\Phi_n(L_c x) - \Phi_n(x)\| &= \|L_c \Phi_n(x) - \Phi_n(x)\| \\ &\leq \|L_c A_n - A_n\| \cdot \|M_n P_n \Phi_{n-1}(x)\| \\ &\leq \|L_c A_n - A_n\| \|x\|. \end{aligned}$$

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

Stability to deformations

Representation

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

How to achieve stability to deformations?

- Patch extraction P_k and pooling A_k **do not commute** with L_τ!

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_τ!
- $\|A_k L_\tau - L_\tau A_k\| \leq C_1 \|\nabla \tau\|_\infty$ [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_τ!
- $\|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$ [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_τ!
- $\|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$ [from Mallat, 2012].
- But: $[P_k, L_\tau]$ is **unstable** at high frequencies!

Stability to deformations

Representation

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

How to achieve stability to deformations?

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

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

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_τ!
- $\|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$ [from Mallat, 2012].
- But: $[P_k, L_\tau]$ is **unstable** at high frequencies!
- Adapt to **current layer resolution**, patch size controlled by σ_{k-1}:

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

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

Stability to deformations: final result

Theorem

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

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

- translation invariance: large σ_n.
- stability: small patch sizes.
- signal preservation: subsampling factor $\approx$ patch size.
- $\Longrightarrow$ **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 σ_n.
- stability: small patch sizes.
- signal preservation: subsampling factor $\approx$ patch size.
- $\Longrightarrow$ **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 σ_n.
- stability: small patch sizes.
- signal preservation: subsampling factor $\approx$ patch size.
- $\Longrightarrow$ **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**:

$$Px(u) = (x(uv))_{v \in S}.$$

- **Non-linear mapping**: equivariant because pointwise!
- **Pooling** (μ: left-invariant Haar measure):

$$Ax(u) = \int_G x(uv)h(v)d\mu(v) = \int_G x(v)h(u^{-1}v)d\mu(v).$$

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

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

What does the RKHS contain?

Homogeneous version of [Zhang et al., 2016, 2017]

RKHS of patch kernels K_k

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

What does the RKHS contain?

- RKHS contains **homogeneous functions**:

$$f : z \mapsto \|z\| \sigma(\langle g, z \rangle / \|z\|).$$

Homogeneous version of [Zhang et al., 2016, 2017]

RKHS of patch kernels K_k

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

What does the RKHS contain?

- RKHS contains **homogeneous functions**:

$$f : z \mapsto \|z\|\sigma(\langle g, z\rangle/\|z\|).$$

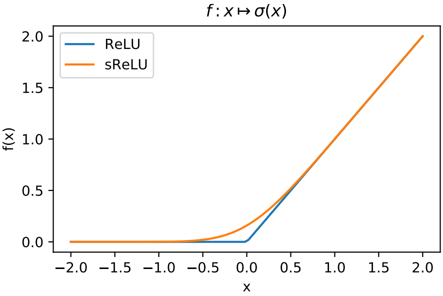
- **Smooth activations**: $\sigma(u) = \sum_{j=0}^{\infty} a_j u^j$ with $a_j \geq 0$.
- Norm: $\|f\|_{\mathcal{H}_k}^2 \leq C_\sigma^2(\|g\|^2) = \sum_{j=0}^{\infty} \frac{a_j^2}{b_j}\|g\|^2 < \infty$.

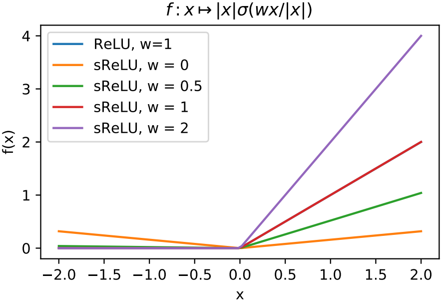
Homogeneous version of [Zhang et al., 2016, 2017]

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, sigmoid, smooth ReLU: $C_\sigma^2(\lambda^2) = O(e^{c\lambda^2})$.





Constructing a CNN in the RKHS $\mathcal{H}_{\mathcal{K}}$

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 $\mathcal{H}_{\mathcal{K}}$

Some CNNs live in the RKHS: "linearization" principle

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

- 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 σ.
- The CNN can be constructed hierarchically in $\mathcal{H}_{\mathcal{K}}$.
- Norm (linear layers):

$$\|f_\sigma\|^2 \leq \|W_{n+1}\|_2^2 \cdot \|W_n\|_2^2 \cdot \|W_{n-1}\|_2^2 \ldots \|W_1\|_2^2.$$

- Linear layers: product of spectral norms.

Link with generalization

Direct application of classical generalization bounds

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

$$\mathcal{F}_B = \{f \in \mathcal{H}_{\mathcal{K}}, \|f\| \leq B\} \implies \mathsf{Rad}_N(\mathcal{F}_B) \leq O\left(\frac{BR}{\sqrt{N}}\right).$$

Link with generalization

Direct application of classical generalization bounds

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

$$\mathcal{F}_B = \{f \in \mathcal{H}_\mathcal{K}, \|f\| \leq B\} \implies \mathsf{Rad}_N(\mathcal{F}_B) \leq O\left(\frac{BR}{\sqrt{N}}\right).$$

- 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?

References I

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research (JMLR)*, 18:1–38, 2017.

Peter Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1706.08498*, 2017.

Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE Transactions on pattern analysis and machine intelligence (PAMI)*, 35 (8):1872–1886, 2013.

Y. Cho and L. K. Saul. Large-margin classification in infinite neural networks. *Neural Computation*, 22(10):2678–2697, 2010.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning (ICML)*, 2016.

References II

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pages 2253–2261, 2016.

J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Anant Raj, Abhishek Kumar, Youssef Mroueh, P Thomas Fletcher, and Bernhard Scholkopf. Local group invariant representations via orbit embeddings. *preprint arXiv:1612.01988*, 2016.

I. Schoenberg. Positive definite functions on spheres. *Duke Math. J.*, 1942.

References III

B. Scholkopf. *Support Vector Learning*. PhD thesis, Technischen Universität Berlin, 1997.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

John Shawe-Taylor and Nello Cristianini. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2004.

Laurent Sifre and Stéphane Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2013.

Alex J Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2000.

References IV

Alex J Smola, Zoltan L Ovari, and Robert C Williamson. Regularization with dot-product kernels. In *Advances in neural information processing systems*, pages 308–314, 2001.

Thomas Wiatowski and Helmut Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 2017.

C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

Kai Zhang, Ivor W Tsang, and James T Kwok. Improved nyström low-rank approximation and error analysis. In *International Conference on Machine Learning (ICML)*, 2008.

Y. Zhang, P. Liang, and M. J. Wainwright. Convexified convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Yuchen Zhang, Jason D Lee, and Michael I Jordan. ℓ_1-regularized neural networks are improperly learnable in polynomial time. In *International Conference on Machine Learning (ICML)*, 2016.

φ_k from kernel approximations: CKNs [Mairal, 2016]

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

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

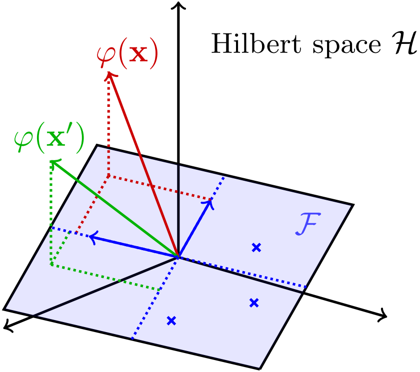


Figure: Nyström approximation.

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

φ_k from kernel approximations: CKNs [Mairal, 2016]

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

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

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

$$\begin{aligned} \|\psi_k(z) - \psi_k(z')\| &= \|\Pi_k \varphi_k(z) - \Pi_k \varphi_k(z')\| \\ &\leq \|\varphi_k(z) - \varphi_k(z')\| \leq \|z - z'\|. \end{aligned}$$

- Anchor points $z_1, \ldots, z_p$ ($\approx$ 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

$$\sup_{x,x' \in L^2(\Omega,\mathcal{H}_0)} \frac{\|\Phi(x) - \Phi(x')\|}{\|x - x'\|} = 1.$$