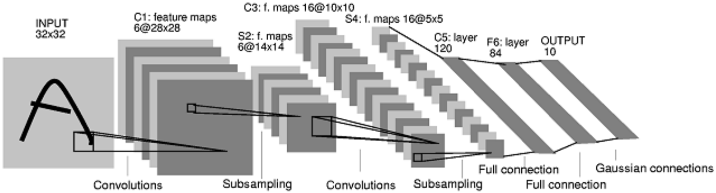
Deep Convolutional Representations: Invariance, Stability, Signal Preservation, Model Complexity

Alberto Bietti Julien Mairal

Inria, Grenoble

NYU, October 16, 2017

Motivation: success of deep CNNs



Convolutional Neural Networks:

- Work very well for natural signals (images, audio, graphs...)
- Key ingredient for state-of-the-art in image classification, object detection, speech recognition
- Exploit properties of natural signals:
  - multi-scale, compositional structure
  - local stationarity
  - some invariance

This work

Why do CNNs work so well?

- Formal study of desirable properties
- Understand the impact of the network architecture

This work

Approach:

- Introduce a generic deep convolutional representation based on *kernels*
  - $\approx$ CNN with large number of feature maps/filters
  - Only depends on **architecture**, not data
  - Leads to successful, tractable approximation (CKNs, Mairal, 2016)

This work

Approach:

- Introduce a generic deep convolutional representation based on *kernels*
  - $\approx$ CNN with large number of feature maps/filters
  - Only depends on **architecture**, not data
  - Leads to successful, tractable approximation (CKNs, Mairal, 2016)
- Formal study of its **properties** (stability, invariance, signal preservation)

This work

Approach:

- Introduce a generic deep convolutional representation based on *kernels*
  - $\approx$ CNN with large number of feature maps/filters
  - Only depends on **architecture**, not data
  - Leads to successful, tractable approximation (CKNs, Mairal, 2016)
- Formal study of its **properties** (stability, invariance, signal preservation)
- How do results apply to learned CNNs?
  - Induced space of functions **contains CNNs**
  - Study model complexity ("norm") of a given CNN
  - $\implies$ stability, invariance, generalization

A kernel perspective...

What??

- Map data x to high-dimensional space, $\Phi(x) \in \mathcal{H}$ ($\mathcal{H}$: "RKHS")
- Non-linear function $f \in \mathcal{H}$ becomes linear: $f(x) = \langle f, \Phi(x) \rangle$
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$

A kernel perspective...

What??

- Map data x to high-dimensional space, $\Phi(x) \in \mathcal{H}$ ($\mathcal{H}$: "RKHS")
- Non-linear function $f \in \mathcal{H}$ becomes linear: $f(x) = \langle f, \Phi(x) \rangle$
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$

Why?

- Separate learning and data representation: $f(x) = \langle f, \Phi(x) \rangle$
  - $\Phi(x)$: CNN architecture (stability, invariance, signal preservation)
  - f: CNN parameters, learning, generalization through RKHS norm $\|f\|$

A kernel perspective...

What??

- Map data x to high-dimensional space, $\Phi(x) \in \mathcal{H}$ ($\mathcal{H}$: "RKHS")
- Non-linear function $f \in \mathcal{H}$ becomes linear: $f(x) = \langle f, \Phi(x) \rangle$
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$

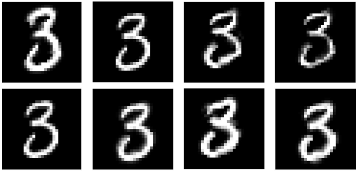
Why?

- Separate learning and data representation: $f(x) = \langle f, \Phi(x) \rangle$
  - $\Phi(x)$: CNN architecture (stability, invariance, signal preservation)
  - f: CNN parameters, learning, generalization through RKHS norm $\|f\|$
- Properties of representation extend to predictions:

$$|f(x) - f(x')| \leq \|f\| \cdot \|\Phi(x) - \Phi(x')\|$$

Outline

Property 1: Stability to deformations



- Go beyond simple translation invariance
- Small local deformations don't change content of images ("label")
- Formally studied for wavelet-based scattering transform (Mallat, 2012; Bruna and Mallat, 2013)
- Can we do the same for deep CNNs?

Property 2: Group invariance

- Convolutions + pooling $\rightarrow$ translation invariance
- Encode more general **transformation groups** in the architecture? (e.g. rotations, roto-translations, rigid motion)
- How does this relate to stability?
- (Cohen and Welling, 2016; Mallat, 2012; Sifre and Mallat, 2013)

Property 3: Signal preservation

- How do deep convolutional representations preserve signal information?
- Can x be recovered from $\Phi(x)$?
- At odds with invariance and stability
- Tentative study through kernel methods

Property 4: Model Complexity and Generalization

- How do we measure model complexity of a generic, learned CNN?
- Can we get meaningful bounds on generalization for a CNN?
- Tentative study through kernel methods:
  - Some CNNs are contained in our RKHS
  - RKHS norm of a generic CNN
  - Impact of activation function
  - Same norm also controls stability (“stable functions generalize better”)

Outline

A generic deep convolutional representation

- $x_0 : \Omega \to \mathcal{H}_0$: initial (**continuous**) signal
  - $u \in \Omega = \mathbb{R}^d$: location ($d = 2$ for images)
  - $x_0(u) \in \mathcal{H}_0$: value ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images)

A generic deep convolutional representation

- $x_0 : \Omega \to \mathcal{H}_0$: initial (**continuous**) signal
  - $u \in \Omega = \mathbb{R}^d$: location ($d = 2$ for images)
  - $x_0(u) \in \mathcal{H}_0$: value ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images)
- $x_k : \Omega \to \mathcal{H}_k$: *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

A generic deep convolutional representation

- $x_0 : \Omega \to \mathcal{H}_0$: initial (**continuous**) signal
  - $u \in \Omega = \mathbb{R}^d$: location ($d = 2$ for images)
  - $x_0(u) \in \mathcal{H}_0$: value ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images)
- $x_k : \Omega \to \mathcal{H}_k$: *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
  - M_k: **non-linear mapping** operator, maps each patch to a new point with a **pointwise** non-linear function $\varphi_k(\cdot)$

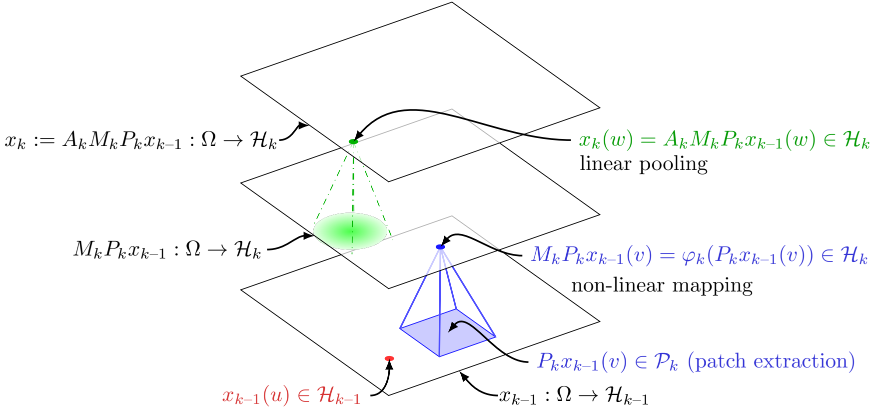
A generic deep convolutional representation

- $x_0 : \Omega \to \mathcal{H}_0$: initial (**continuous**) signal
  - $u \in \Omega = \mathbb{R}^d$: location ($d = 2$ for images)
  - $x_0(u) \in \mathcal{H}_0$: value ($\mathcal{H}_0 = \mathbb{R}^3$ for RGB images)
- $x_k : \Omega \to \mathcal{H}_k$: *feature map* at layer k

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

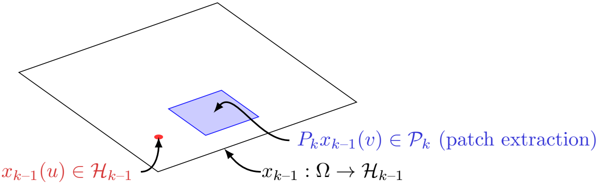
  - P_k: **patch extraction** operator, extract small patch of feature map x_{k-1} around each point u
  - M_k: **non-linear mapping** operator, maps each patch to a new point with a **pointwise** non-linear function $\varphi_k(\cdot)$
  - A_k: (linear, Gaussian) **pooling** operator at scale σ_k

A generic deep convolutional representation



Patch extraction operator P_k

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



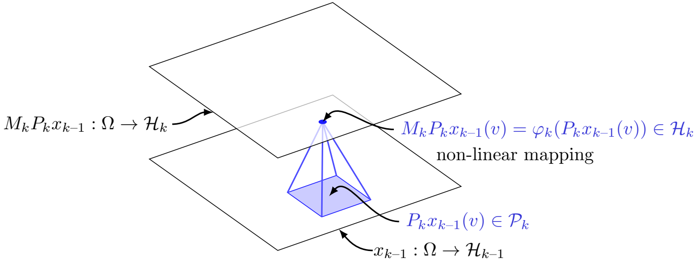
Patch extraction operator P_k

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

- S_k: patch shape, e.g. box
- $\mathcal{P}_k = \mathcal{H}_{k-1}^{S_k}$
- P_k is **linear**, and **preserves the norm**: $\|P_k x_{k-1}\| = \|x_{k-1}\|$

Non-linear 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 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 (kernel map)
- 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 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: 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 $x, x' \in L^2(\Omega, \mathcal{P}_k)$

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

- (can think instead: $\varphi_k(z) = \mathrm{ReLU}(W_k z)$, ρ_k-**Lipschitz** with $\rho_k = \|W_k\|$)

φ_k from kernels

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

$$K_k(z, z') = \|z\|\|z'\|\kappa_k\left(\frac{\langle z, z'\rangle}{\|z\|\|z'\|}\right) \quad \text{with} \quad \kappa_k(1) = 1.$$

- Commonly used for hierarchical kernels
- $\|\varphi_k(z)\| = K_k(z, z)^{1/2} = \|z\|$
- $\|\varphi_k(z) - \varphi_k(z')\| \leq \|z - z'\|$ if $\kappa_k'(1) \leq 1$
- $\implies$ non-expansive

φ_k from kernels

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

$$K_k(z, z') = \|z\|\|z'\|\kappa_k\left(\frac{\langle z, z'\rangle}{\|z\|\|z'\|}\right) \quad \text{with} \quad \kappa_k(1) = 1.$$

- Commonly used for hierarchical kernels
- $\|\varphi_k(z)\| = K_k(z, z)^{1/2} = \|z\|$
- $\|\varphi_k(z) - \varphi_k(z')\| \leq \|z - z'\|$ if $\kappa'_k(1) \leq 1$
- $\implies$ non-expansive
- Examples:
  - $\kappa_{\text{exp}}(\langle z, z'\rangle) = e^{\langle z, z'\rangle - 1}$ (Gaussian kernel on the sphere)
  - $\kappa_{\text{inv-poly}}(\langle z, z'\rangle) = \frac{1}{2 - \langle z, z'\rangle}$

φ_k from kernels: CKNs approximation

φ_k from kernels: CKNs approximation

Convolutional Kernel Networks approximation (Mairal, 2016):

- Approximate $\varphi_k(z)$ by **projection** on $span(\varphi_k(z_1), \dots, \varphi_k(z_p))$ (Nystrom)
- Leads to **tractable**, p-dimensional representation $\psi_k(z)$

φ_k from kernels: CKNs approximation

Convolutional Kernel Networks approximation (Mairal, 2016):

- Approximate $\varphi_k(z)$ by **projection** on $span(\varphi_k(z_1), \ldots, \varphi_k(z_p))$ (Nystrom)
- Leads to **tractable**, p-dimensional representation $\psi_k(z)$
- Norm is preserved, and projection is non-expansive:

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

φ_k from kernels: CKNs approximation

Convolutional Kernel Networks approximation (Mairal, 2016):

- Approximate $\varphi_k(z)$ by **projection** on $span(\varphi_k(z_1), \ldots, \varphi_k(z_p))$ (Nystrom)
- Leads to **tractable**, p-dimensional representation $\psi_k(z)$
- Norm is preserved, and projection is non-expansive:

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

- Non-expansive $\implies$ **robust to additive perturbations**! (e.g., adversarial examples, Cisse et al., 2017)

φ_k from kernels: CKNs approximation

Convolutional Kernel Networks approximation (Mairal, 2016):

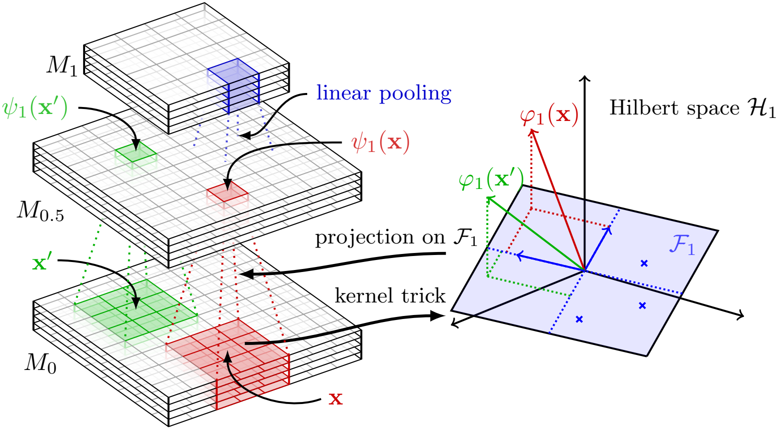
- Approximate $\varphi_k(z)$ by **projection** on $span(\varphi_k(z_1), \dots, \varphi_k(z_p))$ (Nystrom)
- Leads to **tractable**, p-dimensional representation $\psi_k(z)$
- Norm is preserved, and projection is non-expansive:

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

- Non-expansive $\implies$ **robust to additive perturbations**! (e.g., adversarial examples, Cisse et al., 2017)
- Anchor points $z_1, \dots, z_p$ ($\approx$ filters) can be **learned from data** (K-means or backprop)

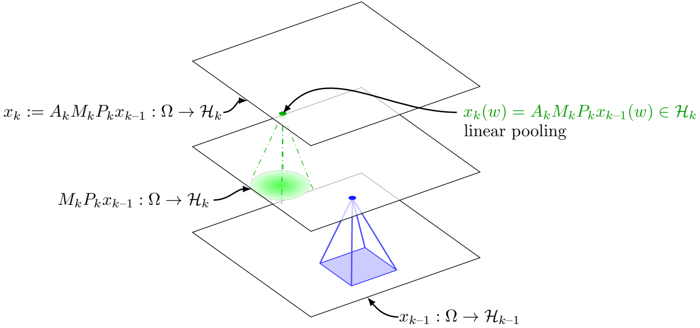
φ_k from kernels: CKNs approximation

Convolutional Kernel Networks approximation (Mairal, 2016):



Pooling operator A_k

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

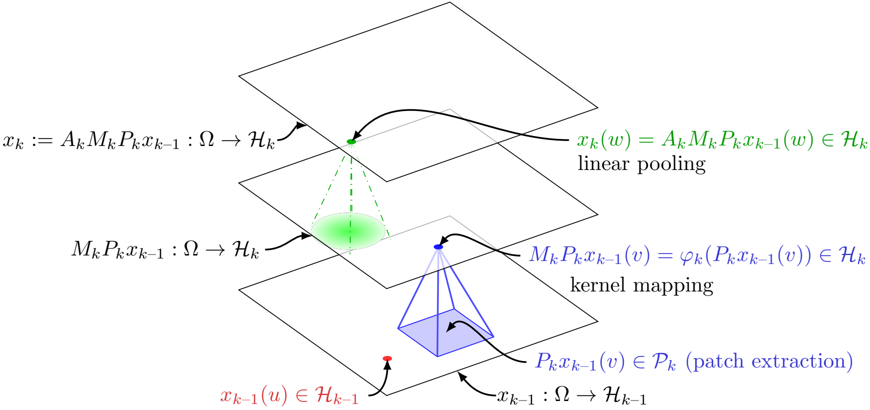


Pooling operator A_k

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

- 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**: $\|A_k\| \leq 1$

Recap: P_k, M_k, A_k



Multilayer construction

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

- S_k, σ_k grow exponentially in practice (i.e. fixed with subsampling)

Multilayer construction

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

- S_k, σ_k grow exponentially in practice (i.e. fixed with subsampling)
- 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 (**anti-aliasing**)

Multilayer construction

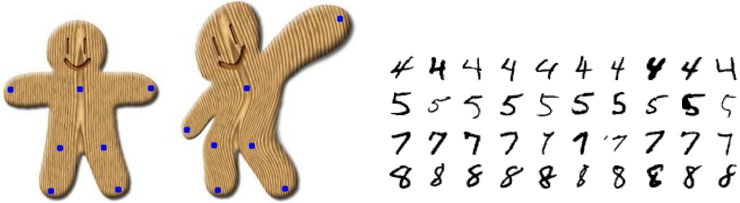
$$x_n := A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 x_0 \ \in \ L^2(\Omega, \mathcal{H}_n)$$

- S_k, σ_k grow exponentially in practice (i.e. fixed with subsampling)
- 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 (**anti-aliasing**)
- **Prediction layer**: e.g. linear
  - $f(x_0) = \langle w, x_n \rangle$
  - "linear kernel" $\mathcal{K}(x_0, x_0') = \langle x_n, x_n' \rangle = \int_\Omega \langle x_n(u), x_n'(u) \rangle du$

Outline

Stability to deformations: definitions

- $\tau : \Omega \to \Omega$: C^1-diffeomorphism
- $L_\tau x(u) = x(u - \tau(u))$: action operator
- Much richer group of transformations than translations



Stability to deformations: definitions

- 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

Warmup: translation invariance

- Representation:

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

- Translation: $L_c x(u) = x(u - c)$

Warmup: translation invariance

- Representation:

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

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

$$\begin{aligned}\|\Phi(L_c x) - \Phi(x)\| &= \|L_c \Phi(x) - \Phi(x)\| \\ &\leq \|L_c A_n - A_n\| \cdot \|x\|\end{aligned}$$

Warmup: translation invariance

- Representation:

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

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

$$\begin{aligned} \|\Phi(L_c x) - \Phi(x)\| &= \|L_c \Phi(x) - \Phi(x)\| \\ &\leq \|L_c A_n - A_n\| \cdot \|x\| \end{aligned}$$

- Mallat (2012): $\|L_\tau A_n - A_n\| \leq \frac{C_2}{\sigma_n}\|\tau\|_\infty$

Warmup: translation invariance

- Representation:

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

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

$$\begin{aligned} \|\Phi(L_c x) - \Phi(x)\| &= \|L_c \Phi(x) - \Phi(x)\| \\ &\leq \|L_c A_n - A_n\| \cdot \|x\| \end{aligned}$$

- Mallat (2012): $\|L_c A_n - A_n\| \leq \frac{C_2}{\sigma_n} c$

Stability to deformations

- Representation:

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

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

Stability to deformations

- Representation:

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

- Patch extraction P_k and pooling A_k **do not commute** with L_τ!
- Mallat (2012): $\|A_k L_\tau - L_\tau A_k\| \leq C_1 \|\nabla \tau\|_\infty$

Stability to deformations

- Representation:

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

- Patch extraction P_k and pooling A_k **do not commute** with L_τ!
- Mallat (2012): $\|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$

Stability to deformations

- Representation:

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

- Patch extraction P_k and pooling A_k **do not commute** with L_τ!
- Mallat (2012): $\|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$
- But: $[P_k, L_\tau]$ is **unstable** at high frequencies!

Stability to deformations

- Representation:

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

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

$$\|[P_k A_{k-1}, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty \qquad \sup_{u \in S_k} |u| \leq \kappa \sigma_{k-1}$$

Stability to deformations

- Representation:

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

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

$$\|[P_k A_{k-1}, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty \qquad \sup_{u \in S_k} |u| \leq \kappa \sigma_{k-1}$$

- C_1 grows as κ^{d+1} $\implies$ more stable with **small patches** (e.g., 3x3, VGG et al.)

Stability to deformations: final result

- Representation:

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

- **Result**: if $\|\nabla\tau\|_\infty \leq 1/2$,

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| \leq \left(C_1 \, (1+n) \, \|\nabla\tau\|_\infty + \frac{C_2}{\sigma_n} \|\tau\|_\infty \right) \|x\|$$

Stability to deformations: final result

- Representation:

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

- **Result**: if $\|\nabla \tau\|_\infty \leq 1/2$,

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| \leq \prod_k \rho_k \left(C_1 (1+n) \|\nabla \tau\|_\infty + \frac{C_2}{\sigma_n} \|\tau\|_\infty \right) \|x\|$$

- (for generic CNNs, multiply by $\prod_k \rho_k = \prod_k \|W_k\|$)

Controlling stability

How is stability controlled?

- full kernels: $\|f\|_{\mathcal{H}_\mathcal{K}}$ (regularizer)
- CKN: $\|W\|_2$, ℓ_2 norm of last layer (regularizer)
- CNN: $\|W\|_2 \cdot \prod_k \rho_k$ (luck...? SGD magic? Parseval nets?)

Beyond the translation group

- Global invariance to other groups? (rotations, reflections, roto-translations, ...)
- Group action $L_g x(u) = x(g^{-1}u)$
- **Equivariance** in inner layers + **(global) pooling** in last layer
- Similar construction to (Cohen and Welling, 2016)

G-equivariant layer construction

- Feature maps $x(u)$ defined on $u \in G$ (G: locally compact group)
- **Patch extraction**:

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

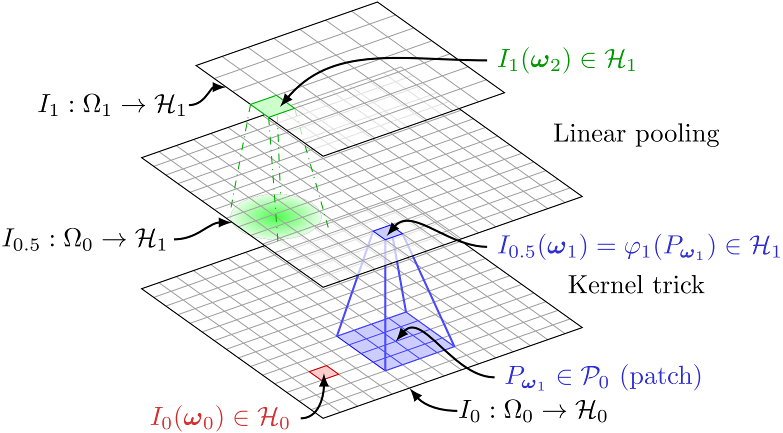
- **Non-linear mapping**: equivariant because pointwise!
- **Pooling** (μ: left-invariant Haar measure):

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

Group invariance and stability

- Stability analysis should work on "compact Lie groups" (similar to Mallat, 2012), e.g., rotations only
- For more complex groups (e.g., roto-translations):
  - Stability only w.r.t. subgroup (translations) is enough?
  - Inner layers: only pool on translation group
  - Last layer: global pooling on rotations
  - Cohen and Welling (2016): rotation pooling in inner layers hurts performance on Rotated MNIST

Discretization and signal preservation



Discretization and signal preservation

- $\bar{x}_k$: subsampling factor s_k after pooling with scale $\sigma_k \approx s_k$:

$$\bar{x}_k[n] = A_k M_k P_k \bar{x}_{k-1}[n s_k]$$

Discretization and signal preservation

- $\bar{x}_k$: 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 **subsampling** $s_k \leq$ **patch size**

Discretization and signal preservation

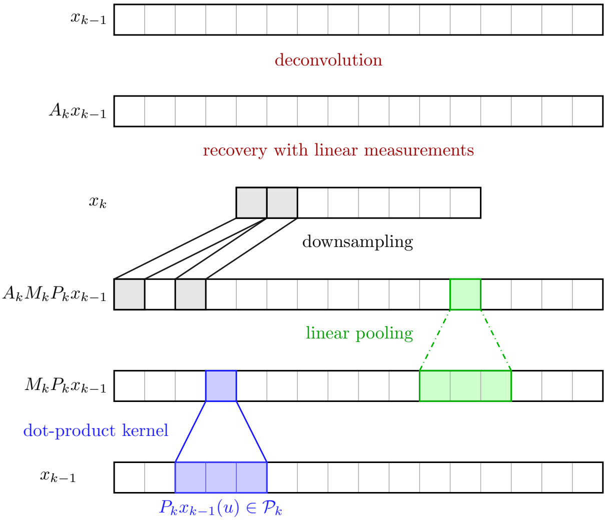
- $\bar{x}_k$: subsampling factor s_k after pooling with scale $\sigma_k \approx s_k$:

$$\bar{x}_k[n] = A_k M_k P_k \bar{x}_{k-1}[n s_k]$$

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

$$\langle f_w, M_k P_k x(u) \rangle = f_w(P_k x(u)) = \langle w, P_k x(u) \rangle$$

Signal recovery: example in 1D



Outline

1. Studied Properties
2. Construction of the Convolutional Representation
3. Invariance, Stability, Signal Preservation
4. Model Complexity and Generalization

From kernel representation to CNNs?

- Functions in the RKHS $\mathcal{H}_k$ of **patch kernels** K_k?
- CNNs in the RKHS $\mathcal{H}_\mathcal{K}$ of the **full kernel** $\mathcal{K}(x, x') = \langle \Phi(x), \Phi(x') \rangle$?
- RKHS norm $\|f\|_{\mathcal{H}_\mathcal{K}}$ for a typical CNN:
  - Stability
  - Generalization

RKHS of patch kernels K_k

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

- 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

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

- RKHS contains **homogeneous functions**:

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

- **Smooth activations**: $\sigma(u) = \sum_{j=0}^{\infty} a_j u^j$
- Norm: $\|f\|^2_{\mathcal{H}_k} \leq C^2_\sigma(\|g\|^2)$

Homogeneous version of (Zhang et al., 2016, 2017)

RKHS of patch kernels K_k

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

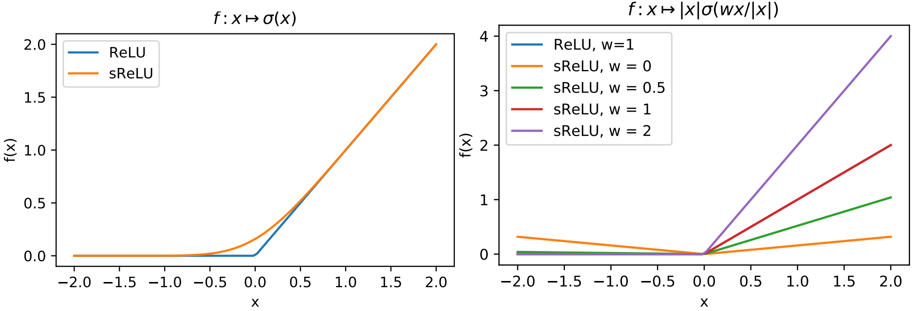
- RKHS contains **homogeneous functions**:

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

- **Smooth activations**: $\sigma(u) = \sum_{j=0}^{\infty} a_j u^j$
- Norm: $\|f\|^2_{\mathcal{H}_k} \leq C^2_\sigma(\|g\|^2)$
- Examples:
  - $\sigma(u) = u$ (linear): $C^2_\sigma(\lambda^2) = O(\lambda^2)$
  - $\sigma(u) = u^p$ (polynomial): $C^2_\sigma(\lambda^2) = O(\lambda^{2p})$
  - $\sigma \approx$ sin, sigmoid, smooth ReLU: $C^2_\sigma(\lambda^2) = O(e^{c\lambda^2})$

Homogeneous version of (Zhang et al., 2016, 2017)

RKHS of patch kernels K_k



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

- Consider a CNN with filters $w_k^{ij}(u), u \in S_k$
- "Homogeneous" activations σ
- The CNN can be constructed hierarchically in $\mathcal{H}_\mathcal{K}$ (define one function $f_k^i \in \mathcal{H}_k$ for each feature map)
- Norm:

$$\|f_\sigma\|^2 \leq \|w_{n+1}\|_2^2 C_\sigma^2(\|w_n\|_2^2 C_\sigma^2(\|w_{n-1}\|_2^2 C_\sigma^2(\dots)))$$

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

- Consider a CNN with filters $w_k^{ij}(u), u \in S_k$
- "Homogeneous" activations σ
- The CNN can be constructed hierarchically in $\mathcal{H}_\mathcal{K}$ (define one function $f_k^i \in \mathcal{H}_k$ for each feature map)
- 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

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

$$\mathcal{F}_B = \{f \in \mathcal{H}_{\mathcal{K}}, \|f\| \leq B\} \implies \mathrm{Rad}_n(\mathcal{F}_B) \leq O\left(\frac{BR}{\sqrt{n}}\right)$$

- Leads to margin bound $O(\|\hat{f}_n\| R/\sqrt{n})$ for a learned CNN $\hat{f}_n$ (margin $= 1/\|\hat{f}_n\|$)

Link with generalization

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

$$\mathcal{F}_B = \{f \in \mathcal{H}_\mathcal{K}, \|f\| \leq B\} \implies \mathrm{Rad}_n(\mathcal{F}_B) \leq O\left(\frac{BR}{\sqrt{n}}\right)$$

- Leads to margin bound $O(\|\hat{f}_n\|R/\sqrt{n})$ for a learned CNN $\hat{f}_n$ (margin $= 1/\|\hat{f}_n\|$)
- For linear activations ($\|f\| \leq \|w_{n+1}\| \cdots \|w_1\|$), similar to Rademacher complexity lower bound of Bartlett et al. (2017)
- Their bound has additional factors:

$$R_\mathcal{A} := \left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma\right) \left(\sum_{i=1}^{L} \frac{\|A_i - M_i\|_1^{2/3}}{\|A_i\|_\sigma^{2/3}}\right)^{3/2}$$

Deep convolutional representations: conclusions

Study of generic properties

- 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

- 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

- RKHS norm as a measure of model complexity
- Useful generalization bounds for CNNs
- Same quantity controls stability and generalization:
  - "higher capacity" (small margin) is needed to discriminate small deformations
  - Learning is "easier" on deformation manifold? ("manifold assumption")
  - Open: how do SGD and friends control capacity in generic CNNs?

References I

P. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1706.08498*, 2017.

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

M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2017.

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

J. Mairal. End-to-End Kernel Learning with Supervised Convolutional Kernel Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

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

S. Saitoh. *Integral transforms, reproducing kernels and their applications*, volume 369. CRC Press, 1997.

References II

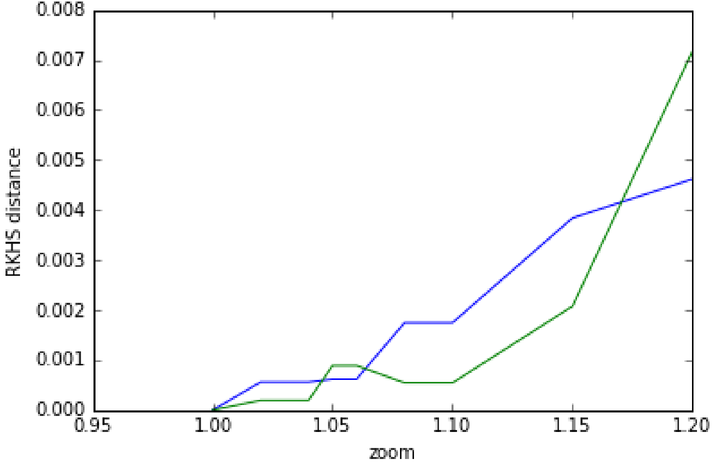
L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2013.

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

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

Simple stability experiment: scaling

$\tau(u) = \epsilon u$ ($1 + \epsilon \equiv$ zoom), full kernel, 2 layers, single CIFAR image



Stability to deformations: proof idea

- Generic bound with **commutators** $[A, B] = AB - BA$:

$$\|\Phi_n(L_\tau x) - \Phi_n(x)\| \\ \leq \left(\sum_{k=1}^{n} \|[P_k A_{k-1}, L_\tau]\| + \|[A_n, L_\tau]\| + \|L_\tau A_n - A_n\| \right) \|x\|.$$

- Use small patch assumption to bound:

$$\|[P_k A_{k-1}, L_\tau]\| \leq \sup_{c \in S_k} \|[L_c A_{k-1}, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$$

- From (Mallat, 2012):

$$\|L_\tau A_\sigma - A_\sigma\| \leq \frac{C_2}{\sigma} \|\tau\|_\infty.$$

Stability to deformations: takeaways

- Small patches adapted to resolution are important for stability
- Translation invariance comes from
  - Last pooling layer
  - Exact *equivariance* in inner layers ("commute with translations")
- Intermediate pooling is for antialiasing/stable downsampling (strided convolutions enough in practice?)
- Why not just skip intermediate layers..? Loss of signal information! (See discretization below...)
- How is stability controlled?
  - full kernels: $\|f\|_{\mathcal{H}}$ (regularizer)
  - CKN: $\|W\|_2$, ℓ_2 norm of last layer (regularizer)
  - CNN: $\|W\|_2 \cdot \prod_k \rho_k$ (luck...? SGD magic? Parseval nets?)

Signal recovery with kernels

Idea:

- "Invert" kernel mapping with **linear functions** to reconstruct patches (non-overlapping)
- Recover full higher resolution (pooled) signal before downsampling
- Deconvolve to recover signal before pooling

Signal recovery with kernels

Idea:

- "Invert" kernel mapping with **linear functions** to reconstruct patches (non-overlapping)
- Recover full higher resolution (pooled) signal before downsampling
- Deconvolve to recover signal before pooling

Linear functions?

- $f_w \in \mathcal{H}_k$ s.t. $f_w(z) = \langle f_w, \varphi_k(z) \rangle_{\mathcal{H}_k} = \langle w, z \rangle_{\mathcal{P}_k}$ for a patch z
- Consider w in a basis of $\mathcal{H}_{k-1}$ for each patch location to recover signal
- Contained in RKHS of most dot-product kernels considered!

Signal recovery: takeaways

- Kernels allow recovery of the signal (up to pooling deconvolutions), when **subsampling $\leq$ patch size**
- $\Phi(x)$ contains all signal information, $f(x) = \langle f, \Phi(x) \rangle$ may focus on what's relevant to the task
- Harder to obtain for CNNs or kernel approximations, but can do well when data-dependent?
- High frequencies are hard to recover if we want translation invariance (vs. full "horizontal" multi-resolution approach like scattering): $A_n \dots A_0 x \approx A_n x$

RKHS of patch kernels K_k

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

- Expansion $\kappa_k(u) = \sum_{j=0}^{\infty} b_j u^j$
- If
  - $\sigma(u) := \sum_{j=0}^{\infty} a_j u^j$ (activation)
  - $C_\sigma^2(\|w\|^2) := \sum_{j=0}^{\infty}(a_j^2/b_j)\|w\|^{2j} < +\infty$
- Then

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

  is in $\mathcal{H}_k$ with $\|f\|_{\mathcal{H}_k}^2 \leq C_\sigma^2(\|w\|^2)$.
- Homogeneous version of (Zhang et al., 2016, 2017)

RKHS of patch kernels K_k

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

- Expansion $\kappa_k(u) = \sum_{j=0}^{\infty} b_j u^j$
- If
  - $\sigma(u) := \sum_{j=0}^{\infty} a_j u^j$ (activation)
  - $C_\sigma^2(\|w\|^2) := \sum_{j=0}^{\infty}(a_j^2/b_j)\|w\|^{2j} < +\infty$
- Then

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

  is in $\mathcal{H}_k$ with $\|f\|_{\mathcal{H}_k}^2 \leq C_\sigma^2(\|w\|^2)$.
- Homogeneous version of (Zhang et al., 2016, 2017)
- Linear functions contained when $b_1 > 0$

RKHS of full kernel $\mathcal{K}$

Theorem (e.g., Saitoh, 1997)

- If $\Phi : \mathcal{X} \to H$ (e.g., $\mathcal{X} = L^2(\Omega, \mathcal{H}^0), H = L^2(\Omega, \mathcal{H}_n)$)
- The RKHS of $\mathcal{K}(x, x') = \langle \Phi(x), \Phi(x') \rangle_H$ is

$$\mathcal{H}_\mathcal{K} := \{f_w \; ; \; w \in H\} \quad \text{s.t.} \quad f_w : z \mapsto \langle w, \Phi(z) \rangle_H,$$

$$\|f_w\|^2_{\mathcal{H}_\mathcal{K}} := \inf_{w' \in H} \{\|w'\|^2_H \quad \text{s.t.} \quad f_w = f_{w'}\} \leq \|w\|^2_H$$

Goal: construct a $w \in L^2(\Omega, \mathcal{H}_n)$ hierarchically to obtain a CNN

Constructing a CNN in the RKHS

CNN:

- Filters $w_k^{ij} \in L^2(S_k, \mathbb{R})$
- Feature maps $z_k^i = A_k \tilde{z}_k^i \in L^2(\Omega, \mathbb{R})$ ($z_0 = x_0$):

$$\tilde{z}_k^i(u) = \sigma\Big(\langle w_k^i, P_k z_{k-1}(u)\rangle\Big)$$

Constructing a CNN in the RKHS

CNN:

- Filters $w_k^{ij} \in L^2(S_k, \mathbb{R})$
- Feature maps $z_k^i = A_k \tilde{z}_k^i \in L^2(\Omega, \mathbb{R})$ ($z_0 = x_0$):

$$\tilde{z}_k^i(u) = \sigma\Big(\langle w_k^i, P_k z_{k-1}(u)\rangle\Big)$$

RKHS construction:

- f_k^i in $\mathcal{H}_k$ and g_k^i in $\mathcal{P}_k$

$$g_k^i(v) = \sum_{j=1}^{p_{k-1}} w_k^{ij}(v) f_{k-1}^j \quad \text{where} \quad w_k^i(v) = (w_k^{ij}(v))_{j=1,\dots,p_{k-1}}$$

$$f_k^i(z) = \|z\| \sigma(\langle g_k^i, z\rangle / \|z\|) \quad \text{for } z \in \mathcal{P}_k.$$

Constructing a CNN in the RKHS

CNN:

- Filters $w_k^{ij} \in L^2(S_k, \mathbb{R})$
- Feature maps $z_k^i = A_k \tilde{z}_k^i \in L^2(\Omega, \mathbb{R})$ ($z_0 = x_0$):

$$\tilde{z}_k^i(u) = n_k(u)\sigma\Big(\langle w_k^i, P_k z_{k-1}(u)\rangle / n_k(u)\Big)$$

RKHS construction:

- f_k^i in $\mathcal{H}_k$ and g_k^i in $\mathcal{P}_k$

$$g_k^i(v) = \sum_{j=1}^{p_{k-1}} w_k^{ij}(v) f_{k-1}^j \quad \text{where} \quad w_k^i(v) = (w_k^{ij}(v))_{j=1,\ldots,p_{k-1}}$$

$$f_k^i(z) = \|z\|\sigma(\langle g_k^i, z\rangle / \|z\|) \quad \text{for } z \in \mathcal{P}_k.$$

Constructing a CNN in the RKHS

CNN:

- Linear prediction layer: $w_{n+1}^j \in L^2(\Omega, \mathbb{R})$
- $f_\sigma(x_0) = \langle w_{n+1}, z_n \rangle$

Constructing a CNN in the RKHS

CNN:

- Linear prediction layer: $w_{n+1}^j \in L^2(\Omega, \mathbb{R})$
- $f_\sigma(x_0) = \langle w_{n+1}, z_n \rangle$

RKHS construction:

- $g_\sigma \in L^2(\Omega, \mathcal{H}_n)$

$$g_\sigma(u) = \sum_{j=1}^{p_n} w_{n+1}^j(u) f_n^j \quad \text{for all } u \in \Omega,$$

Constructing a CNN in the RKHS

CNN:

- Linear prediction layer: $w^j_{n+1} \in L^2(\Omega, \mathbb{R})$
- $f_\sigma(x_0) = \langle w_{n+1}, z_n \rangle$

RKHS construction:

- $g_\sigma \in L^2(\Omega, \mathcal{H}_n)$

$$g_\sigma(u) = \sum_{j=1}^{p_n} w^j_{n+1}(u) f^j_n \quad \text{for all } u \in \Omega,$$

We have: $\langle g_\sigma, \Phi(x_0) \rangle = f_\sigma(x_0) \implies f_\sigma \in \mathcal{H}_\mathcal{K}$

Norm of the CNN

Simple recursive bound

$$\|f_\sigma\|^2 \leq p_n \sum_{i=1}^{p_n} \|w_{n+1}^i\|_2^2 B_{n,i},$$

with

$$B_{1,i} = C_\sigma^2(\|w_1^i\|_2^2)$$
$$B_{k,i} = C_\sigma^2 \left(p_{k-1} \sum_{j=1}^{p_{k-1}} \|w_k^{ij}\|_2^2 B_{k-1,j} \right).$$

Norm of the CNN

Spectral norm bound

$$\|f_\sigma\|^2 \leq \|w_{n+1}\|_2^2 C_\sigma^2(\|w_n\|_2^2 C_\sigma^2(\|w_{n-1}\|_2^2 C_\sigma^2(\ldots))),$$

where $\|w_k\|_2^2 = \int_{S_k} \|w_k(u)\|_2^2 du$ and $\|w_k(u)\|_2$ is the spectral norm of the matrix $(w_k^{ij}(u))_{ij}$.

- With 1x1 patches (fully-connected) and no activations (linear), $C_\sigma^2(\lambda) = \lambda$, we get **product of spectral norms**
  - Similar form to Rademacher complexity lower bound of (Bartlett et al., 2017)
  - In contrast, their bound has L^1 norm factors