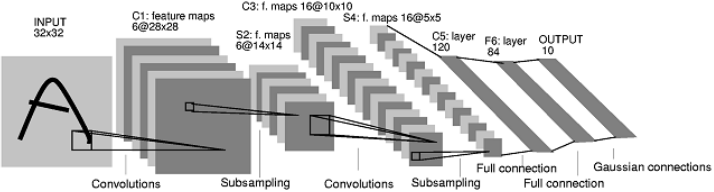
Invariance and Stability to Deformations of Deep Convolutional Representations

Alberto Bietti

Inria, Grenoble

TTI Chicago. June 3, 2019.

Success of deep convolutional networks



Convolutional Neural Networks (CNNs):

- Capture **multi-scale** and **compositional** structure in natural signals
- Provide some **invariance**
- Model **local stationarity**
- **State-of-the-art** in many applications

Understanding deep convolutional representations

- Are they **stable to deformations**?
- How can we achieve **invariance to transformation groups**?
- Do they **preserve signal information**?
- What are good measures of **model complexity**?

A kernel perspective

Kernels?

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

A kernel perspective

Kernels?

- Map data x to high-dimensional space, $\Phi(x) \in \mathcal{H}$ ($\mathcal{H}$: "RKHS")
- Non-linear $f \in \mathcal{H}$ takes linear form: $f(x) = \langle f, \Phi(x) \rangle$
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$
- Here, we construct an RKHS following Mairal (2016) and CNNs:

$$f(x) = W_{n+1}\sigma(W_n\sigma(W_{n-1}...\sigma(W_2\sigma(W_1x))...)) = \langle f, \Phi(x) \rangle$$

A kernel perspective

Kernels?

- Map data x to high-dimensional space, $\Phi(x) \in \mathcal{H}$ ($\mathcal{H}$: "RKHS")
- Non-linear $f \in \mathcal{H}$ takes linear form: $f(x) = \langle f, \Phi(x) \rangle$
- Learning with a positive definite kernel $K(x, x') = \langle \Phi(x), \Phi(x') \rangle$
- Here, we construct an RKHS following Mairal (2016) and CNNs:

$$f(x) = W_{n+1}\sigma(W_n\sigma(W_{n-1}...\sigma(W_2\sigma(W_1x))...)) = \langle f, \Phi(x) \rangle$$

- (also related to *neural tangent kernels* for CNNs (Bietti and Mairal, 2019b))

A kernel perspective

Why? Separate learning from representation: $f(x) = \langle f, \Phi(x) \rangle$

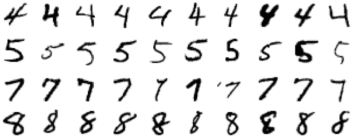
- $\Phi(x)$: CNN **architecture** (stability, invariance, signal preservation)
- f: CNN **model**, learning, generalization through RKHS norm $\|f\|$

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

- $\|f\|$ **controls both stability and model complexity**!
  $\rightarrow$ discriminating small perturbations requires large $\|f\|$
  $\rightarrow$ learning stable functions may be "easier"

A signal processing perspective

- Consider images defined on a **continuous** domain $\Omega = \mathbb{R}^2$.
- $\tau : \Omega \to \Omega$: C^1-diffeomorphism.
- $L_\tau x(u) = x(u - \tau(u))$: action operator.
- Much richer group of transformations than translations.



A signal processing perspective

- Consider images defined on a **continuous** domain $\Omega = \mathbb{R}^2$.
- $\tau : \Omega \to \Omega$: 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.

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 $\varphi_k(\cdot)$

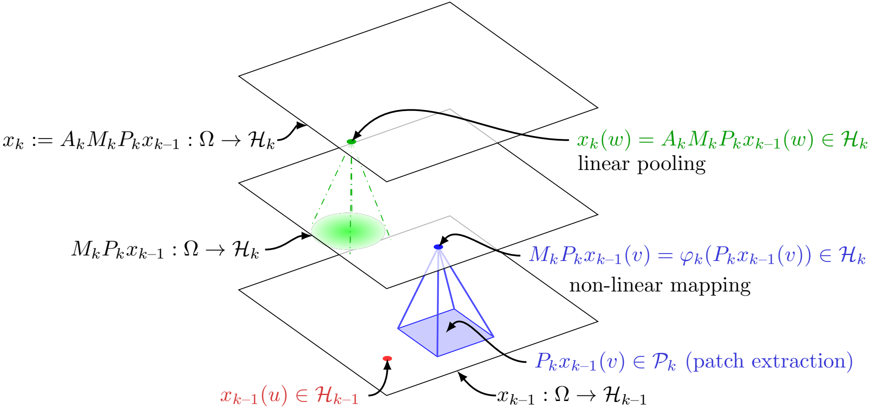
A generic deep convolutional representation

- $x_0 : \Omega \to \mathcal{H}_0$: 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

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

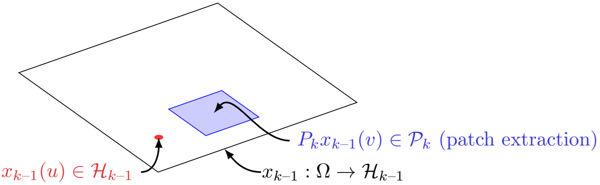
- P_k: **patch extraction** operator, extract small patch of feature map x_{k-1} around each point u
- M_k: **non-linear mapping** operator, maps each patch to a new point with a **pointwise** non-linear function $\varphi_k(\cdot)$
- A_k: (linear, Gaussian) **pooling** operator at scale σ_k

A generic deep convolutional representation



Patch extraction operator P_k

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



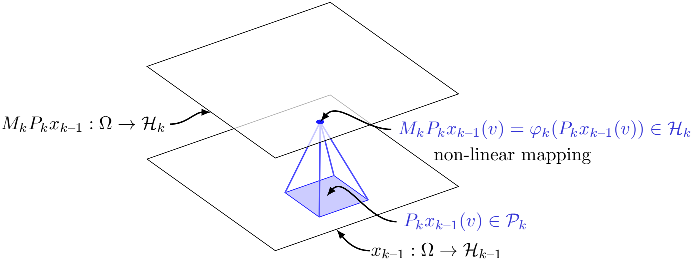
Patch extraction operator P_k

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

- S_k: patch shape, e.g. box
- P_k is **linear**, and **preserves the L^2 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 $x, x' \in L^2(\Omega, \mathcal{P}_k)$

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

- (at the cost of paying $\prod_k \rho_k$ later)

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 $b_j \geq 0$, $\kappa_k(1) = 1$
- 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**:

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

- $\kappa_k(u) = \sum_{j=0}^{\infty} b_j u^j$ with $b_j \geq 0$, $\kappa_k(1) = 1$
- 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)
  - $\kappa_{\text{inv-poly}}(\langle z, z'\rangle) = \frac{1}{2 - \langle z, z'\rangle}$
  - arc-cosine kernel of degree 1 (random features with ReLU activation)

φ_k from kernels: CKNs approximation

Convolutional Kernel Networks approximation (Mairal, 2016):

φ_k from kernels: CKNs approximation

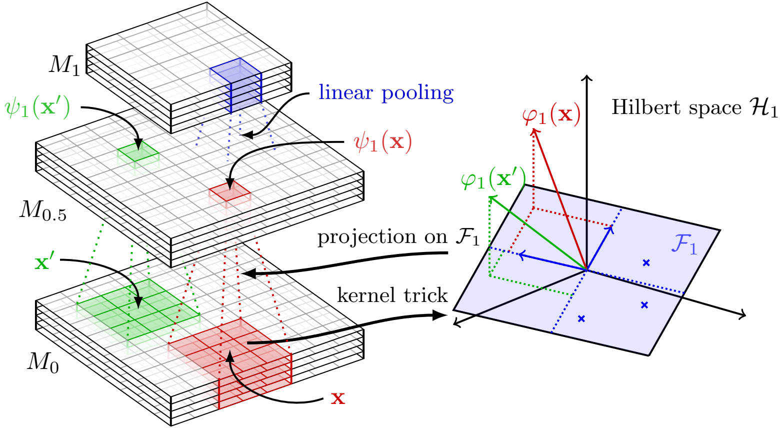
Convolutional Kernel Networks approximation (Mairal, 2016):

- 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:

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

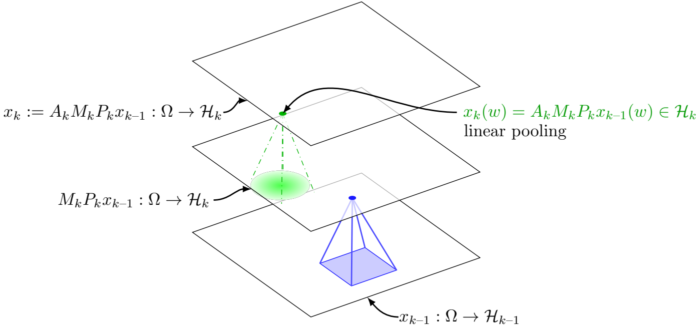
- Anchor points $z_1, \dots, z_p$ ($\approx$ filters) can be **learned from data** (K-means or backprop)

φ_k from kernels: CKNs approximation



Pooling operator A_k

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

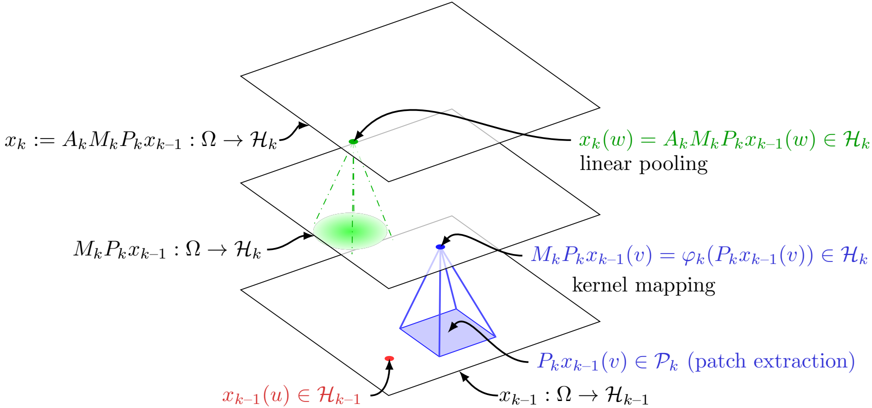


Pooling operator A_k

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

- 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

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

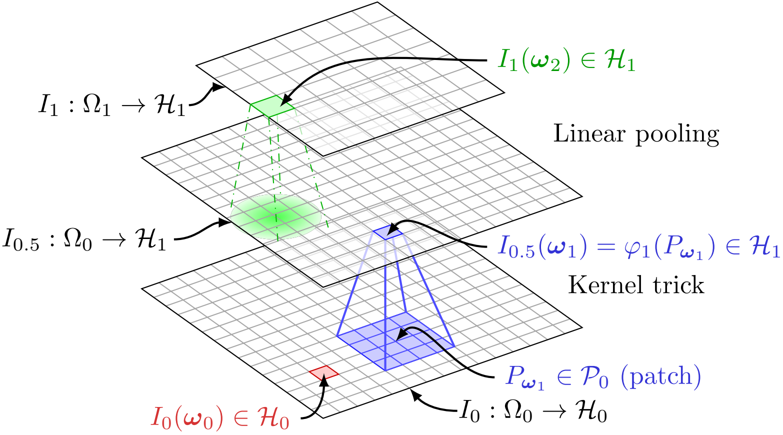
$$\Phi_n(x) = A_n M_n P_n A_{n-1} M_{n-1} P_{n-1} \cdots A_1 M_1 P_1 x_0 \in L^2(\Omega, \mathcal{H}_n).$$

- 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



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 $\sigma_k \approx s_k$:

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

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

Discretization and signal preservation

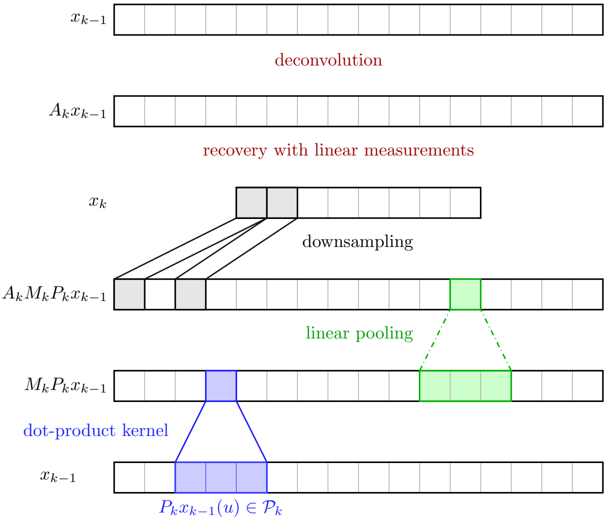
- $\bar{x}_k$: subsampling factor s_k after pooling with scale $\sigma_k \approx s_k$:

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

- **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)

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

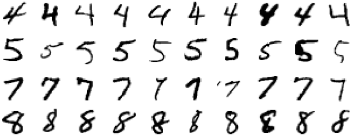
Signal recovery: example in 1D



Outline

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



- Studied for wavelet-based scattering transform (Mallat, 2012; Bruna and Mallat, 2013)

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: $\square L_c = L_c \square$

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

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: $\square L_c = L_c \square$

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

- 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: $\square L_c = L_c \square$

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

- 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!
- $\|A_k L_\tau - L_\tau A_k\| \leq C_1 \|\nabla \tau\|_\infty$ (from Mallat, 2012)

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!
- $\|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$ (from Mallat, 2012)

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

Stability to deformations

- Representation:

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

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

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

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!
- $\|[A_k, L_\tau]\| \leq C_1 \|\nabla \tau\|_\infty$ (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}:

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

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

Stability to deformations: final result

Theorem

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

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

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

Stability to deformations: final result

Theorem

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

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

- translation invariance: large σ_n
- stability: small patch sizes
- signal preservation: subsampling factor $\approx$ patch size
- $\Longrightarrow$ **needs several layers**
- (also valid for generic CNNs with ReLUs: multiply by $\prod_k \rho_k = \prod_k \|W_k\|$, but no direct signal preservation).

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); Kondor and Trivedi (2018)

G-equivariant layer construction

- Feature maps $x(u)$ defined on $u \in G$ (G: locally compact group)
  - Input needs special definition when $G \neq \Omega$
- **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

Roto-translation group $G = \mathbb{R}^2 \rtimes SO(2)$ (translations + rotations)

- **Stability** w.r.t. 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

RKHS of patch kernels K_k

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

- 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

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

- RKHS contains **homogeneous functions**:

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

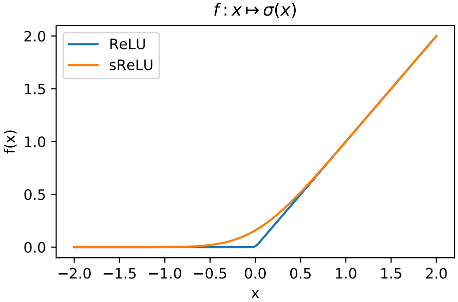
- **Smooth activations**: $\sigma(u) = \sum_{j=0}^{\infty} a_j u^j$
- Norm: $\|f\|_{\mathcal{H}_k}^2 \leq C_\sigma^2(\|g\|^2) = \sum_{j=0}^{\infty} \frac{a_j^2}{b_j} \|g\|^2 < \infty$

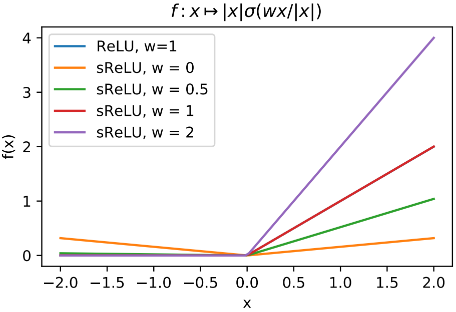
Homogeneous version of (Zhang et al., 2016, 2017)

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

- Consider a CNN with filters $W_k^{ij}(u), u \in S_k$
- "Smooth homogeneous" activations σ
- The CNN can be constructed hierarchically in $\mathcal{H}_{\mathcal{K}}$
- Norm upper bound:

$$\|f_\sigma\|_{\mathcal{H}}^2 \leq \|W_{n+1}\|_2^2 \; C_\sigma^2(\|W_n\|_2^2 \; C_\sigma^2(\|W_{n-1}\|_2^2 \; C_\sigma^2(\dots)))$$

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

- Consider a CNN with filters $W_k^{ij}(u), u \in S_k$
- "Smooth homogeneous" activations σ
- The CNN can be constructed hierarchically in $\mathcal{H}_\mathcal{K}$
- Norm upper bound (linear layers):

$$\|f_\sigma\|_\mathcal{H}^2 \leq \|W_{n+1}\|_2^2 \cdot \|W_n\|_2^2 \cdot \|W_{n-1}\|_2^2 \dots \|W_1\|_2^2$$

- Linear layers: product of spectral norms

Link with generalization

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

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

Link with generalization

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

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

- Leads to margin bound $O(\|\hat{f}_N\|_{\mathcal{H}} R/\gamma\sqrt{N})$ for a learned CNN $\hat{f}_N$ with margin (confidence) $\gamma > 0$
- Related to generalization bounds for neural networks based on **product of spectral norms** (*e.g.*, Bartlett et al., 2017; Neyshabur et al., 2018)

Outline

1. Construction of the Convolutional Representation
2. Invariance and Stability
3. Learning Aspects: Model Complexity of CNNs
4. Regularizing with the RKHS norm

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

- Controlling **upper bounds**: spectral norm penalties/constraints

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

- Controlling **upper bounds**: spectral norm penalties/constraints
- Controlling **lower bounds** using $\|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq 1} \langle f, u \rangle$

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

- Controlling **upper bounds**: spectral norm penalties/constraints
- Controlling **lower bounds** using $\|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq 1} \langle f, u \rangle$
- $\implies$ consider tractable subsets of the unit ball

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

- Controlling **upper bounds**: spectral norm penalties/constraints
- Controlling **lower bounds** using $\|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq 1} \langle f, u \rangle$
- $\implies$ consider tractable subsets of the unit ball

$$\|f\|_{\mathcal{H}} \geq \sup_{x, \|\delta\| \leq 1} f(x + \delta) - f(x) \quad \text{(adversarial perturbations)}$$

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

- Controlling **upper bounds**: spectral norm penalties/constraints
- Controlling **lower bounds** using $\|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq 1} \langle f, u \rangle$
- $\implies$ consider tractable subsets of the unit ball

$$\|f\|_{\mathcal{H}} \geq \sup_{x, \|\delta\| \leq 1} f(x + \delta) - f(x) \quad \text{(adversarial perturbations)}$$

$$\|f\|_{\mathcal{H}} \geq \sup_{x, \|\tau\| \leq C} f(L_\tau x) - f(x) \quad \text{(adversarial deformations)}$$

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

- Controlling **upper bounds**: spectral norm penalties/constraints
- Controlling **lower bounds** using $\|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq 1} \langle f, u \rangle$
- $\implies$ consider tractable subsets of the unit ball

$$\|f\|_{\mathcal{H}} \geq \sup_{x, \|\delta\| \leq 1} f(x + \delta) - f(x) \quad \text{(adversarial perturbations)}$$

$$\|f\|_{\mathcal{H}} \geq \sup_{x, \|\tau\| \leq C} f(L_\tau x) - f(x) \quad \text{(adversarial deformations)}$$

$$\|f\|_{\mathcal{H}} \geq \sup_{x} \|\nabla f(x)\|_2 \quad \text{(gradient penalty)}$$

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

- Controlling **upper bounds**: spectral norm penalties/constraints
- Controlling **lower bounds** using $\|f\|_{\mathcal{H}} = \sup_{\|u\|_{\mathcal{H}} \leq 1} \langle f, u \rangle$
- $\implies$ consider tractable subsets of the unit ball

$$\|f\|_{\mathcal{H}} \geq \sup_{x, \|\delta\| \leq 1} f(x + \delta) - f(x) \quad \text{(adversarial perturbations)}$$

$$\|f\|_{\mathcal{H}} \geq \sup_{x, \|\tau\| \leq C} f(L_\tau x) - f(x) \quad \text{(adversarial deformations)}$$

$$\|f\|_{\mathcal{H}} \geq \sup_{x} \|\nabla f(x)\|_2 \quad \text{(gradient penalty)}$$

- Best performance by combining upper + lower bound approaches

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

Table 2. Regularization on 300 or 1 000 examples from MNIST, using deformations from Infinite MNIST. $(*)$ indicates that random deformations were included as training examples, while $\|f\|_\tau^2$ and $\|D_\tau f\|^2$ use them as part of the regularization penalty.

Method	300 VGG	1k VGG
Weight decay	89.32	94.08
SN projection	90.69	95.01
grad-ℓ_2	93.63	96.67
$\|f\|_\delta^2$ penalty	94.17	96.99
$\|\nabla f\|^2$ penalty	94.08	96.82
Weight decay $(*)$	92.41	95.64
grad-ℓ_2 $(*)$	95.05	97.48
$\|D_\tau f\|^2$ penalty	94.18	96.98
$\|f\|_\tau^2$ penalty	94.42	97.13
$\|f\|_\tau^2 + \|\nabla f\|^2$	94.75	97.40
$\|f\|_\tau^2 + \|f\|_\delta^2$	95.23	**97.66**
$\|f\|_\tau^2 + \|f\|_\delta^2$ $(*)$	**95.53**	**97.56**
$\|f\|_\tau^2 + \|f\|_\delta^2$ + SN proj	95.20	**97.60**
$\|f\|_\tau^2 + \|f\|_\delta^2$ + SN proj $(*)$	**95.40**	**97.77**

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularizing with the RKHS norm in practice

Can we obtain better models with little data?

Table 3. Regularization on protein homology detection tasks, with or without data augmentation (DA). Fixed hyperparameters are selected using the first half of the datasets, and we report the average auROC50 score on the second half.

Method	No DA	DA
No weight decay	0.446	0.500
Weight decay	0.501	0.546
SN proj	0.591	**0.632**
PGD-ℓ_2	0.575	0.595
grad-ℓ_2	0.540	0.552
$\|f\|_\delta^2$	**0.600**	0.608
$\|\nabla f\|^2$	0.585	0.611
PGD-ℓ_2 + SN proj	**0.596**	**0.627**
grad-ℓ_2 + SN proj	0.592	**0.624**
$\|f\|_\delta^2$ + SN proj	**0.630**	**0.644**
$\|\nabla f\|^2$ + SN proj	**0.603**	**0.625**

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularization for robustness

- Robust optimization yields another lower bound (hinge/logistic loss)

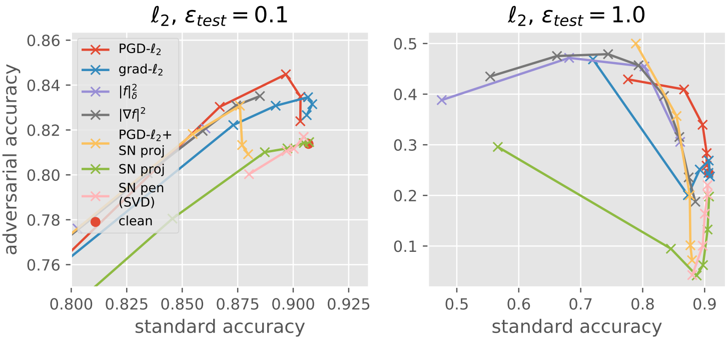
$$\frac{1}{N}\sum_{i=1}^{N}\sup_{\|\delta\|_2\leq\epsilon}\ell(y_i, f(x_i+\delta)) \leq \frac{1}{N}\sum_{i=1}^{N}\ell(y_i, f(x_i)) + \epsilon\|f\|_{\mathcal{H}}$$

- Controlling $\|f\|_{\mathcal{H}}$ allows a more **global** form of robustness
- Leads to margin bounds for *adversarial generalization* with ℓ_2 perturbations
  - Using $\|f\|_{\mathcal{H}} \geq \|f\|_{\mathrm{Lip}}$ near the margin (Bietti et al., 2019)
- But, may cause a loss in accuracy in practice

(Bietti, Mialon, Chen, and Mairal, 2019)

Regularization for robustness

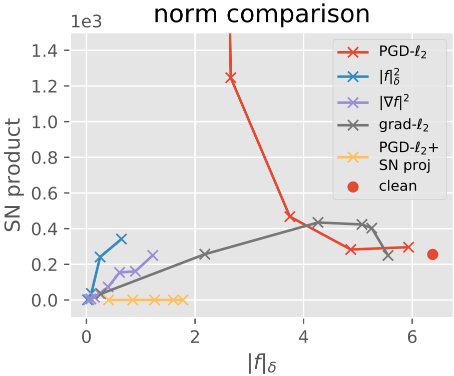
Robust vs standard accuracy trade-offs



(Bietti, Mialon, Chen, and Mairal, 2019)

Regularization for robustness

Upper vs lower bounds



(Bietti, Mialon, Chen, and Mairal, 2019)

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

- Same quantity $\|f\|$ controls stability and generalization:
  - "higher capacity" is needed to discriminate small deformations
  - Learning may be "easier" with stable functions
- Better regularization of generic CNNs using RKHS norm

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

- Same quantity $\|f\|$ controls stability and generalization:
    - "higher capacity" is needed to discriminate small deformations
    - Learning may be "easier" with stable functions
- Better regularization of generic CNNs using RKHS norm

Links with optimization (Bietti and Mairal, 2019b)

- Similar kernel (NTK) arises from optimization in a certain regime
- Weaker stability guarantees, but better approximation properties

References I

P. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

A. Bietti and J. Mairal. Invariance and stability of deep convolutional representations. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

A. Bietti and J. Mairal. Group invariance, stability to deformations, and complexity of deep convolutional representations. *Journal of Machine Learning Research (JMLR)*, 20(25):1–49, 2019a.

A. Bietti and J. Mairal. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019b.

A. Bietti, G. Mialon, D. Chen, and J. Mairal. A kernel perspective for regularizing deep neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019.

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

References II

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

R. Kondor and S. Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

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.

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

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.