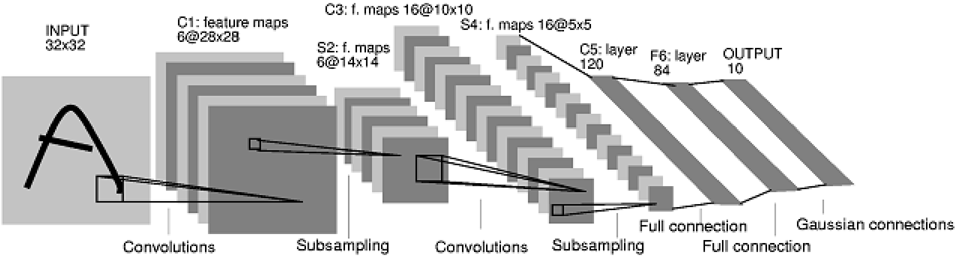
On the Benefits of Convolutional Models: a Kernel Perspective

Alberto Bietti

NYU

BIRS. May 26, 2022.

Convolutional Networks

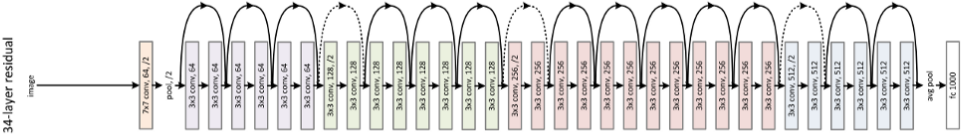


(LeCun et al., 1998)

Exploiting the structure of natural images

- Model local information at different scales, hierarchically
- Provide some invariance through pooling
- Useful **inductive biases** for learning efficiently on natural signals

Convolutional Networks



(He et al., 2016)

Exploiting the structure of natural images

- Model local information at different scales, hierarchically
- Provide some invariance through pooling
- Useful **inductive biases** for learning efficiently on natural signals

Setup

Nonparametric regression with kernels

- Data model: $y = f^*(x) + \epsilon$
- Kernel ridge regression with kernel K (with RKHS $\mathcal{H}$)

$$\hat{f}_n = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Setup

Nonparametric regression with kernels

- Data model: $y = f^*(x) + \epsilon$
- Kernel ridge regression with kernel K (with RKHS $\mathcal{H}$)

$$\hat{f}_n = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2$$

Questions

- What are good assumptions on f^* for image problems?
- How does the kernel/norm/architecture exploit this for sample efficiency?

Kernels for Convolutional Models

This talk (B. et al., 2021; B., 2022):

- Formal study of **convolutional kernels** and their RKHS
- **Benefits** of (deep) convolutional structure

Kernels for Convolutional Models

This talk (B. et al., 2021; B., 2022):

- Formal study of **convolutional kernels** and their RKHS
- **Benefits** of (deep) convolutional structure

Invariance

Kernels for Convolutional Models

This talk (B. et al., 2021; B., 2022):

- Formal study of **convolutional kernels** and their RKHS
- **Benefits** of (deep) convolutional structure

Invariance

Locality
Long-range interactions

Why Kernels?

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems
  - *e.g.*, smooth functions (Caponnetto and De Vito, 2007), interaction splines (Wahba, 1990)

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems
  - *e.g.*, smooth functions (Caponnetto and De Vito, 2007), interaction splines (Wahba, 1990)

We rarely have *all three*, *e.g.*:

- Approximation benefits of depth: *no algorithms*
  - (Eldan and Shamir, 2016; Mhaskar and Poggio, 2016; Telgarsky, 2016; Cohen and Shashua, 2017; Schmidt-Hieber, 2020)
- Optimization landscape/algorithmic regularization: *no universal approximation*
  - (Soltanolkotabi et al., 2018; Gunasekar et al., 2018; Jagadeesan et al., 2022; Razin et al., 2022)

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems
  - *e.g.*, smooth functions (Caponnetto and De Vito, 2007), interaction splines (Wahba, 1990)

We rarely have *all three*, *e.g.*:

- Approximation benefits of depth: *no algorithms*
  - (Eldan and Shamir, 2016; Mhaskar and Poggio, 2016; Telgarsky, 2016; Cohen and Shashua, 2017; Schmidt-Hieber, 2020)
- Optimization landscape/algorithmic regularization: *no universal approximation*
  - (Soltanolkotabi et al., 2018; Gunasekar et al., 2018; Jagadeesan et al., 2022; Razin et al., 2022)

A starting point to understand CNNs

- Understand the **features** $\Phi(x)$ provided by architectures ($\approx$ least squares before Lasso)

Why Kernels?

Clean and well-developed theory

- Tractable **optimization** algorithms (convex)
- Universal **approximation** guarantees
- Optimal **statistical** rates for many problems
  - *e.g.*, smooth functions (Caponnetto and De Vito, 2007), interaction splines (Wahba, 1990)

We rarely have *all three*, *e.g.*:

- Approximation benefits of depth: *no algorithms*
  - (Eldan and Shamir, 2016; Mhaskar and Poggio, 2016; Telgarsky, 2016; Cohen and Shashua, 2017; Schmidt-Hieber, 2020)
- Optimization landscape/algorithmic regularization: *no universal approximation*
  - (Soltanolkotabi et al., 2018; Gunasekar et al., 2018; Jagadeesan et al., 2022; Razin et al., 2022)

A starting point to understand CNNs

- Understand the **features** $\Phi(x)$ provided by architectures ($\approx$ least squares before Lasso)
- Good performance on Cifar10 (Mairal, 2016; Li et al., 2019; Shankar et al., 2020; B., 2022)

Outline

1. Invariance and Stability (B., Venturi, and Bruna, 2021)

2. Locality and Depth (B., 2022)

Outline

1. Invariance and Stability (B., Venturi, and Bruna, 2021)

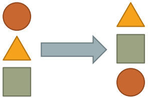
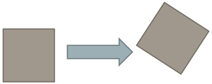
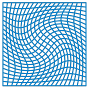
2. Locality and Depth (B., 2022)

Invariance and Geometric Stability

Invariance and Geometric Stability

Q: Does invariance improve statistical efficiency?

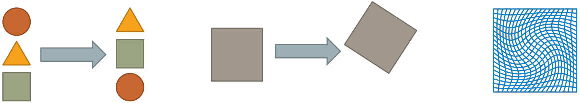
Invariance and Geometric Stability: Definitions

Functions $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

- *e.g.*, translations, rotations, permutations, deformations

Invariance and Geometric Stability: Definitions

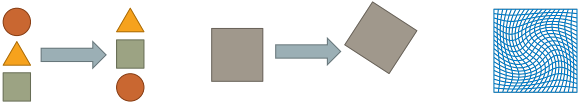


Functions $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

- *e.g.*, translations, rotations, permutations, deformations
- We consider: **permutations** $\sigma \in G$

$$(\sigma \cdot x)_i = x_{\sigma^{-1}(i)}$$

Invariance and Geometric Stability: Definitions



Functions $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

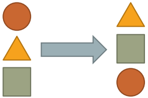
- *e.g.*, translations, rotations, permutations, deformations
- We consider: **permutations** $\sigma \in G$

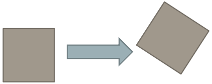
$$(\sigma \cdot x)_i = x_{\sigma^{-1}(i)}$$

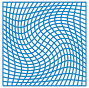
Group invariance: If G is a group (*e.g.*, cyclic shifts, all permutations), we want

$$f(\sigma \cdot x) = f(x), \quad \sigma \in G$$

Invariance and Geometric Stability: Definitions







Functions $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that are "smooth" along known transformations of input x

- *e.g.*, translations, rotations, permutations, deformations
- We consider: **permutations** $\sigma \in G$

$$(\sigma \cdot x)_i = x_{\sigma^{-1}(i)}$$

Group invariance: If G is a group (*e.g.*, cyclic shifts, all permutations), we want

$$f(\sigma \cdot x) = f(x), \quad \sigma \in G$$

Geometric stability: For other sets G (*e.g.*, local shifts, deformations), we want

$$f(\sigma \cdot x) \approx f(x), \quad \sigma \in G$$

Interlude: Kernels for Wide Shallow Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(\langle w_i, x \rangle)$$

Interlude: Kernels for Wide Shallow Networks

$$\begin{aligned} f(x) &= \frac{1}{\sqrt{m}} \sum_{i=1}^m v_i \rho(\langle w_i, x \rangle) \\ &= \langle v, \varphi(x) \rangle, \qquad \text{with } \varphi(x) = \frac{1}{\sqrt{m}} \rho(Wx) \in \mathbb{R}^m \end{aligned}$$

Interlude: Kernels for Wide Shallow Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(\langle w_i, x \rangle)$$

$$= \langle v, \varphi(x) \rangle, \qquad \text{with } \varphi(x) = \frac{1}{\sqrt{m}} \rho(Wx) \in \mathbb{R}^m$$

- **Random Features** (RF, Neal, 1996; Rahimi and Recht, 2007): $w_i \sim \mathcal{N}(0, I)$, learn v

$$K_{RF}(x, x') = \lim_{m \to \infty} \langle \varphi(x), \varphi(x') \rangle$$

$$= \mathbb{E}_w[\rho(\langle w, x \rangle)\rho(\langle w, x' \rangle)] = \kappa_\rho(\langle x, x' \rangle) \quad \text{when } x, x' \in \mathbb{S}^{d-1}$$

Interlude: Kernels for Wide Shallow Networks

$$f(x) = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} v_i \rho(\langle w_i, x \rangle)$$
$$= \langle v, \varphi(x) \rangle, \qquad \text{with } \varphi(x) = \frac{1}{\sqrt{m}} \rho(Wx) \in \mathbb{R}^m$$

- **Random Features** (RF, Neal, 1996; Rahimi and Recht, 2007): $w_i \sim \mathcal{N}(0, I)$, learn v

$$K_{RF}(x, x') = \lim_{m \to \infty} \langle \varphi(x), \varphi(x') \rangle$$
$$= \mathbb{E}_w[\rho(\langle w, x \rangle)\rho(\langle w, x' \rangle)] = \kappa_\rho(\langle x, x' \rangle) \text{ when } x, x' \in \mathbb{S}^{d-1}$$

- Related to **Neural Tangent Kernel** (NTK, Jacot et al., 2018): train both w_i and v_i near random initialization

Group-Invariant Models through Pooling

$$\varphi(x) = \frac{1}{\sqrt{m}} \rho(Wx)$$

Convolutional network with pooling (group averaging)

$$f_G(x) = \langle v, \underbrace{\frac{1}{|G|} \sum_{\sigma \in G} \varphi(\sigma \cdot x)}_{\Phi(x)} \rangle$$

Group-Invariant Models through Pooling

$$\varphi(x) = \frac{1}{\sqrt{m}}\rho(Wx)$$

Convolutional network with pooling (group averaging)

$$f_G(x) = \langle v, \underbrace{\frac{1}{|G|}\sum_{\sigma \in G} \varphi(\sigma \cdot x)}_{\Phi(x)} \rangle$$

Invariant kernel (Haasdonk and Burkhardt, 2007; Mroueh et al., 2015)

$$K_G(x, x') = \frac{1}{|G|}\sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle), \quad \text{when } x, x' \in \mathbb{S}^{d-1}$$

- When $\kappa = \kappa_\rho$, this corresponds to Random Features kernel for f_G

Statistical Benefits of Group Invariance

- Regression: $R(f) := \mathbb{E}(y - f(x))^2$, x uniform on the sphere $\mathbb{S}^{d-1}$, and $f^*(x) = \mathbb{E}[y|x]$.

Statistical Benefits of Group Invariance

- Regression: $R(f) := \mathbb{E}(y - f(x))^2$, x uniform on the sphere $\mathbb{S}^{d-1}$, and $f^*(x) = \mathbb{E}[y|x]$.
- Kernel ridge regression (KRR) using:

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

Statistical Benefits of Group Invariance

- Regression: $R(f) := \mathbb{E}(y - f(x))^2$, x uniform on the sphere $\mathbb{S}^{d-1}$, and $f^*(x) = \mathbb{E}[y|x]$.
- Kernel ridge regression (KRR) using:

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

Theorem (Benefits of invariance (B., Venturi, and Bruna, 2021))

*Assume f^* is G-**invariant** and s-**smooth**. KRR with kernel K_G vs K achieves*

$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \leq C_d \left(\frac{1 + o(1)}{|G| n} \right)^{\frac{2s}{2s+d-1}} \qquad \text{vs.} \qquad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \leq C_d \left(\frac{1}{n} \right)^{\frac{2s}{2s+d-1}}$$

Statistical Benefits of Group Invariance

- Regression: $R(f) := \mathbb{E}(y - f(x))^2$, x uniform on the sphere $\mathbb{S}^{d-1}$, and $f^*(x) = \mathbb{E}[y|x]$.
- Kernel ridge regression (KRR) using:

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

Theorem (Benefits of invariance (B., Venturi, and Bruna, 2021))

*Assume f^* is G-**invariant** and s-**smooth**. KRR with kernel K_G vs K achieves*

$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \leq C_d \left(\frac{1 + o(1)}{|G|\, n} \right)^{\frac{2s}{2s+d-1}} \qquad vs. \qquad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \leq C_d \left(\frac{1}{n} \right)^{\frac{2s}{2s+d-1}}$$

$\Longrightarrow$ **asymptotic gains by a factor $|G|$ in sample complexity.**

Statistical Benefits of Group Invariance

- Regression: $R(f) := \mathbb{E}(y - f(x))^2$, x uniform on the sphere $\mathbb{S}^{d-1}$, and $f^*(x) = \mathbb{E}[y|x]$.
- Kernel ridge regression (KRR) using:

$$K_G(x, x') = \frac{1}{|G|} \sum_{\sigma \in G} \kappa(\langle \sigma \cdot x, x' \rangle) \qquad \text{vs.} \qquad K(x, x') = \kappa(\langle x, x' \rangle)$$

Theorem (Benefits of invariance (B., Venturi, and Bruna, 2021))

*Assume f^* is G-**invariant** and s-**smooth**. KRR with kernel K_G vs K achieves*

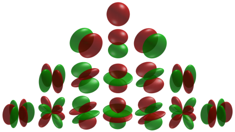
$$\mathbb{E}\, R(\hat{f}_{K_G,n}) - R(f^*) \leq C_d \left(\frac{1 + o(1)}{|G| n} \right)^{\frac{2s}{2s+d-1}} \qquad \text{vs.} \qquad \mathbb{E}\, R(\hat{f}_{K,n}) - R(f^*) \leq C_d \left(\frac{1}{n} \right)^{\frac{2s}{2s+d-1}}$$

$\Longrightarrow$ **asymptotic gains by a factor $|G|$ in sample complexity.**

- $|G|$ can be exponential in d for some groups (*e.g.*, the full permutation group)
- Rate and dimension-dependence in constant C_d are asymptotically minimax optimal

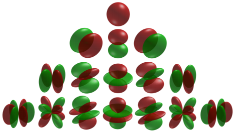
Key Technical Ingredient: Counting Invariant Harmonics

- Expand in $L^2(\mathbb{S}^{d-1})$ basis of **spherical harmonics** $Y_{k,j}$
- $N(d,k)$ harmonics of degree k, form a basis of $V_{d,k}$
- Pooling projects $V_{d,k}$ to $\overline{N}(d,k)$ **invariant harmonics**



Key Technical Ingredient: Counting Invariant Harmonics

- Expand in $L^2(\mathbb{S}^{d-1})$ basis of **spherical harmonics** $Y_{k,j}$
- $N(d,k)$ harmonics of degree k, form a basis of $V_{d,k}$
- Pooling projects $V_{d,k}$ to $\overline{N}(d,k)$ **invariant harmonics**



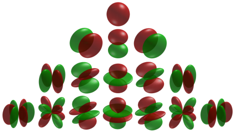
Theorem ((B., Venturi, and Bruna, 2021))

As $k \to \infty$, we have

$$\gamma_d(k) := \frac{\overline{N}(d,k)}{N(d,k)} = \frac{1}{|G|} + o(1).$$

Key Technical Ingredient: Counting Invariant Harmonics

- Expand in $L^2(\mathbb{S}^{d-1})$ basis of **spherical harmonics** $Y_{k,j}$
- $N(d,k)$ harmonics of degree k, form a basis of $V_{d,k}$
- Pooling projects $V_{d,k}$ to $\overline{N}(d,k)$ **invariant harmonics**



Theorem ((B., Venturi, and Bruna, 2021))

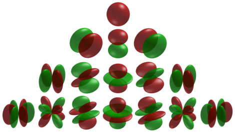
As $k \to \infty$, we have

$$\gamma_d(k) := \frac{\overline{N}(d,k)}{N(d,k)} = \frac{1}{|G|} + o(1).$$

- Relies on singularity analysis of density of $\langle \sigma \cdot x, x \rangle$ (Saldanha and Tomei, 1996)
- Decay rate can be quantified using cycle statistics of $\sigma \in G$

Key Technical Ingredient: Counting Invariant Harmonics

- Expand in $L^2(\mathbb{S}^{d-1})$ basis of **spherical harmonics** $Y_{k,j}$
- $N(d,k)$ harmonics of degree k, form a basis of $V_{d,k}$
- Pooling projects $V_{d,k}$ to $\overline{N}(d,k)$ **invariant harmonics**



Theorem ((B., Venturi, and Bruna, 2021))

As $k \to \infty$, we have

$$\gamma_d(k) := \frac{\overline{N}(d,k)}{N(d,k)} = \frac{1}{|G|} + o(1).$$

- Relies on singularity analysis of density of $\langle \sigma \cdot x, x \rangle$ (Saldanha and Tomei, 1996)
- Decay rate can be quantified using cycle statistics of $\sigma \in G$
- Uses a characterization of $\gamma_d(k)$ due to Mei et al. (2021), who study a different regime:
  - They study $d \to \infty$ with fixed k ($\gamma_d(k) = \Theta_d(d^{-\alpha})$), gains at most polynomial in d
  - We study $k \to \infty$ with fixed d, gain $|G|$ can be exponential in d.

Extension to Stability and Discussion

Extension to geometric stability: G is not a group (*e.g.*, local shifts/deformations)

- Pooling operation is no longer a projection, but leads to natural assumption
- Similar bounds with effective sample size $n|G|$
- $|G|$ is exponential in d for a simple toy model of deformations!

Extension to Stability and Discussion

Extension to geometric stability: G is not a group (*e.g.*, local shifts/deformations)

- Pooling operation is no longer a projection, but leads to natural assumption
- Similar bounds with effective sample size $n|G|$
- $|G|$ is exponential in d for a simple toy model of deformations!

Curse of dimensionality

- If the target f^* is non-smooth, *e.g.*, only Lipschitz, the rate is cursed! (and unimprovable)

$$R(\hat{f}_n) - f(f^*) \lesssim n^{-\frac{2}{2+d-1}}$$

Extension to Stability and Discussion

Extension to geometric stability: G is not a group (*e.g.*, local shifts/deformations)

- Pooling operation is no longer a projection, but leads to natural assumption
- Similar bounds with effective sample size $n|G|$
- $|G|$ is exponential in d for a simple toy model of deformations!

Curse of dimensionality

- If the target f^* is non-smooth, *e.g.*, only Lipschitz, the rate is cursed! (and unimprovable)

$$R(\hat{f}_n) - f(f^*) \lesssim n^{-\frac{2}{2+d-1}}$$

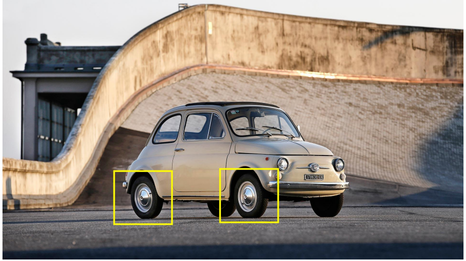
Q: How can we break this curse?

Outline

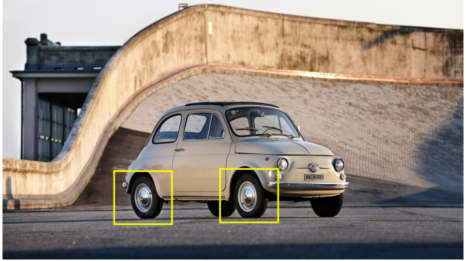
1. Invariance and Stability (B., Venturi, and Bruna, 2021)

2. Locality and Depth (B., 2022)

Locality

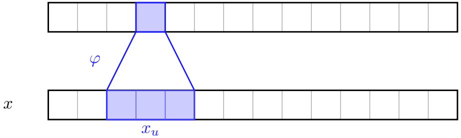


Locality



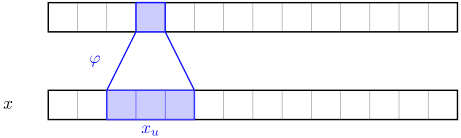
Q: Can locality improve statistical efficiency?

One-Layer Convolutional Kernels on Patches



- 1D signal: $x[u]$, $u \in \Omega$
- **Patches**: $x_u = (x[u], \dots, x[u + p - 1]) \in \mathbb{R}^{p}$, features $\varphi(x_u) = \frac{1}{\sqrt{m}} \rho(W x_u)$, $m \to \infty$

One-Layer Convolutional Kernels on Patches



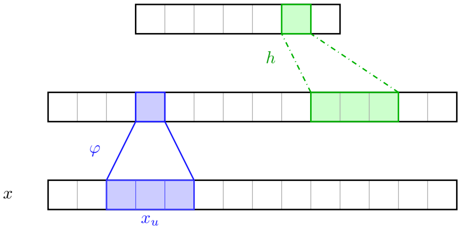
- 1D signal: $x[u]$, $u \in \Omega$
- **Patches**: $x_u = (x[u], \dots, x[u+p-1]) \in \mathbb{R}^p$, features $\varphi(x_u) = \frac{1}{\sqrt{m}}\rho(Wx_u)$, $m \to \infty$
- **Convolutional network**:

$$f(x) = \sum_{u \in \Omega} \langle v_u, \varphi(x_u) \rangle =: \langle v, \Phi(x) \rangle$$

- **Convolutional kernel** (with $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$ the RF **patch kernel**)

$$K(x, x') = \sum_{u \in \Omega} k(x_u, x'_u)$$

One-Layer Convolutional Kernels on Patches



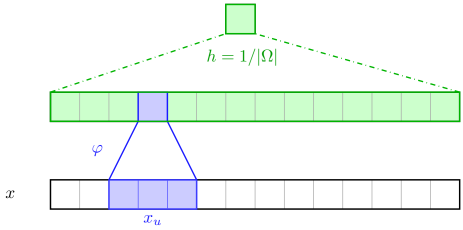
- 1D signal: $x[u]$, $u \in \Omega$
- **Patches**: $x_u = (x[u], \dots, x[u+p-1]) \in \mathbb{R}^p$, features $\varphi(x_u) = \frac{1}{\sqrt{m}} \rho(W x_u)$, $m \to \infty$
- **Convolutional network**: with pooling filter h

$$f_h(x) = \sum_{u \in \Omega} \langle v_u, \sum_v h[u-v] \varphi(x_v) \rangle$$

- **Convolutional kernel** (with $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$ the RF **patch kernel**)

$$K_h(x, x') = \sum_{u \in \Omega} \sum_{v, v'} h[u-v] h[u-v'] k(x_v, x'_{v'})$$

One-Layer Convolutional Kernels on Patches



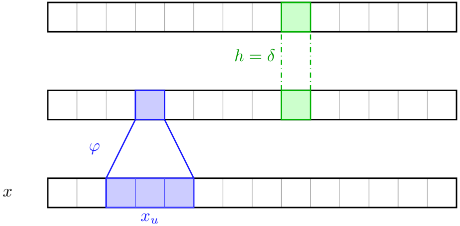
- 1D signal: $x[u]$, $u \in \Omega$
- **Patches**: $x_u = (x[u], \dots, x[u+p-1]) \in \mathbb{R}^p$, features $\varphi(x_u) = \frac{1}{\sqrt{m}} \rho(W x_u)$, $m \to \infty$
- **Convolutional network**: with global pooling ($h = 1/|\Omega|$)

$$f_h(x) = \sum_{u \in \Omega} \langle v_u, |\Omega|^{-1} \sum_v \varphi(x_v) \rangle$$

- **Convolutional kernel** (with $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$ the RF **patch kernel**)

$$K_h(x, x') = |\Omega|^{-1} \sum_{v, v'} k(x_v, x'_{v'})$$

One-Layer Convolutional Kernels on Patches



- 1D signal: $x[u]$, $u \in \Omega$
- **Patches**: $x_u = (x[u], \dots, x[u+p-1]) \in \mathbb{R}^p$, features $\varphi(x_u) = \frac{1}{\sqrt{m}} \rho(W x_u)$, $m \to \infty$
- **Convolutional network**: with no pooling (Dirac $h = \delta$)

$$f_h(x) = \sum_{u \in \Omega} \langle v_u, \varphi(x_u) \rangle$$

- **Convolutional kernel** (with $k(z, z') = \langle \varphi(z), \varphi(z') \rangle$ the RF **patch kernel**)

$$K_h(x, x') = \sum_{u \in \Omega} k(x_u, x'_u)$$

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

$$\text{(global pool)}\ K_g(x, x') = \sum_{v, v'} k(x_v, x'_{v'}) \qquad \text{vs} \qquad \text{(no pool)}\ K_\delta(x, x') = \sum_u k(x_u, x'_u)$$

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

$$\text{(global pool) } K_g(x, x') = \sum_{v, v'} k(x_v, x'_{v'}) \quad \text{vs} \quad \text{(no pool) } K_\delta(x, x') = \sum_u k(x_u, x'_u)$$

Theorem (Statistical rates with one-layer (B., 2022))

*Assume g^* is s-**smooth**, non-overlapping patches on $\mathbb{S}^{p-1}$. KRR with K_h yields*

$$\mathbb{E}\, R(\hat{f}_{g,n}) - R(f^*) \leq C_p \left(\frac{1}{n}\right)^{\frac{2s}{2s+p-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \leq C_p \left(\frac{|\Omega|}{n}\right)^{\frac{2s}{2s+p-1}}$$

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

$$\text{(global pool) } K_g(x, x') = \sum_{v, v'} k(x_v, x'_{v'}) \quad \text{vs} \quad \text{(no pool) } K_\delta(x, x') = \sum_u k(x_u, x'_u)$$

Theorem (Statistical rates with one-layer (B., 2022))

*Assume g^* is s-**smooth**, non-overlapping patches on $\mathbb{S}^{p-1}$. KRR with K_h yields*

$$\mathbb{E}\, R(\hat{f}_{g,n}) - R(f^*) \leq C_p \left(\frac{1}{n}\right)^{\frac{2s}{2s+p-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \leq C_p \left(\frac{|\Omega|}{n}\right)^{\frac{2s}{2s+p-1}}$$

- Patch dimension $p \ll d = p|\Omega|$ in the rate (**breaks the curse**!)

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

$$\text{(global pool) } K_g(x, x') = \sum_{v, v'} k(x_v, x'_{v'}) \quad \text{vs} \quad \text{(no pool) } K_\delta(x, x') = \sum_u k(x_u, x'_u)$$

Theorem (Statistical rates with one-layer (B., 2022))

*Assume g^* is s-**smooth**, non-overlapping patches on $\mathbb{S}^{p-1}$. KRR with K_h yields*

$$\mathbb{E}\, R(\hat{f}_{g,n}) - R(f^*) \leq C_p \left(\frac{1}{n}\right)^{\frac{2s}{2s+p-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \leq C_p \left(\frac{|\Omega|}{n}\right)^{\frac{2s}{2s+p-1}}$$

- Patch dimension $p \ll d = p|\Omega|$ in the rate (**breaks the curse**!)
- With localized pooling h, we can also learn $f^*(x) = \sum_{u \in \Omega} g^*_u(x_u)$ with different g^*_u
  - The bound above interpolates between 1 and $|\Omega|$ via $\|h\|_2^2$

Benefits of Locality and Pooling

- Assume **additive**, **invariant** target $f^*(x) = \sum_{u \in \Omega} g^*(x_u)$
- Consider the kernels:

$$\text{(global pool)}\ K_g(x, x') = \sum_{v, v'} k(x_v, x'_{v'}) \qquad \text{vs} \qquad \text{(no pool)}\ K_\delta(x, x') = \sum_u k(x_u, x'_u)$$

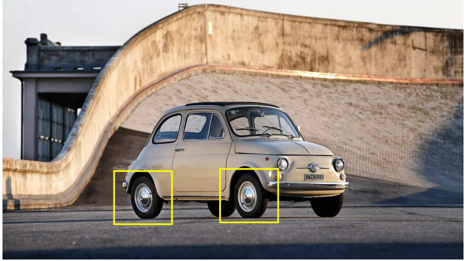
Theorem (Statistical rates with one-layer (B., 2022))

*Assume g^* is s-**smooth**, non-overlapping patches on $\mathbb{S}^{p-1}$. KRR with K_h yields*

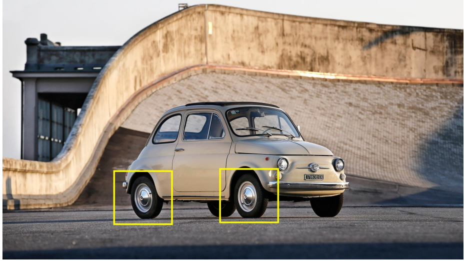
$$\mathbb{E}\, R(\hat{f}_{g,n}) - R(f^*) \leq C_p \left(\frac{1}{n}\right)^{\frac{2s}{2s+p-1}} \quad vs \quad \mathbb{E}\, R(\hat{f}_{\delta,n}) - R(f^*) \leq C_p \left(\frac{|\Omega|}{n}\right)^{\frac{2s}{2s+p-1}}$$

- Patch dimension $p \ll d = p|\Omega|$ in the rate (**breaks the curse**!)
- With localized pooling h, we can also learn $f^*(x) = \sum_{u \in \Omega} g_u^*(x_u)$ with different g_u^*
  - The bound above interpolates between 1 and $|\Omega|$ via $\|h\|_2^2$
- For overlapping patches, see (Favero et al., 2021; Misiakiewicz and Mei, 2021)

Long-Range Interactions

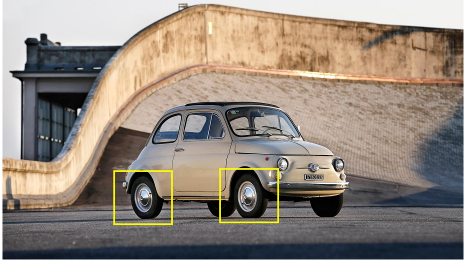


Long-Range Interactions



Q: How to capture interactions between multiple patches?

Long-Range Interactions

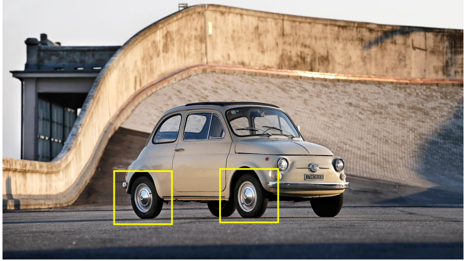


Q: How to capture interactions between multiple patches?

$\rightarrow$ **"add more layers"! Hierarchical kernels** (Cho and Saul, 2009):

$$K(x, x') = \langle \varphi_2(\varphi_1(x)), \varphi_2(\varphi_1(x')) \rangle$$

Long-Range Interactions

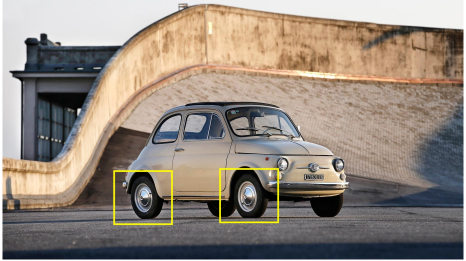


Q: How to capture interactions between multiple patches?

$\rightarrow$ **"add more layers"! Hierarchical kernels** (Cho and Saul, 2009):

$$K(x, x') = \kappa_2(\langle \varphi_1(x), \varphi_1(x') \rangle)$$

Long-Range Interactions

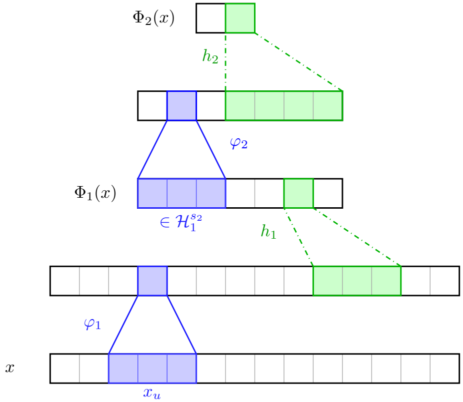


Q: How to capture interactions between multiple patches?

$\rightarrow$ **"add more layers"! Hierarchical kernels** (Cho and Saul, 2009):

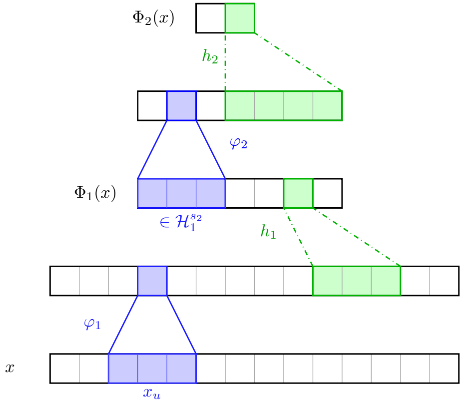
$$K(x, x') = \kappa_2(\kappa_1(\langle x, x' \rangle))$$

RKHS of Two-Layer Convolutional Kernels (B., 2022)



- φ_2 / κ_2 captures **interactions** between patches

RKHS of Two-Layer Convolutional Kernels (B., 2022)

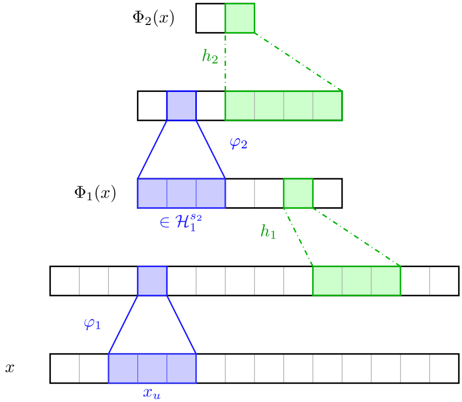


- φ_2/κ_2 captures **interactions** between patches
- Take $\kappa_2(u) = u^2$. RKHS contains

$$f(x) = \sum_{|u-v|\leq r} g_{u,v}(x_u, x_v)$$

- Receptive field r depends on h_1 and s_2
- $g_{u,v} \in \mathcal{H}_1 \otimes \mathcal{H}_1$

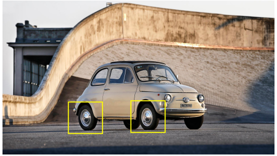
RKHS of Two-Layer Convolutional Kernels (B., 2022)



- φ_2 / κ_2 captures **interactions** between patches
- Take $\kappa_2(u) = u^2$. RKHS contains

$$f(x) = \sum_{|u-v| \leq r} g_{u,v}(x_u, x_v)$$

- Receptive field r depends on h_1 and s_2
- $g_{u,v} \in \mathcal{H}_1 \otimes \mathcal{H}_1$



- Effect of RKHS norm:
  - Pooling h_1: invariance to **relative** position
  - Pooling h_2: invariance to **global** position

Is it a Good Model for Cifar10? (B., 2022)

Compute $50\,000 \times 50\,000$ kernel matrix (costly!) and run Kernel Ridge Regression (ok!)

Is it a Good Model for Cifar10? (B., 2022)

Compute $50\,000 \times 50\,000$ kernel matrix (costly!) and run Kernel Ridge Regression (ok!)

2-layers, patch sizes (3, 5), Gaussian pooling factors (2,5).

κ_1	κ_2	Test acc.
Exp	Exp	88.3%
Exp	Poly4	88.3%
Exp	Poly3	88.2%
Exp	Poly2	87.4%
Exp	Linear	80.9%

Is it a Good Model for Cifar10? (B., 2022)

Compute $50\,000 \times 50\,000$ kernel matrix (costly!) and run Kernel Ridge Regression (ok!)

2-layers, patch sizes (3, 5), Gaussian pooling factors (2,5).

κ_1	κ_2	Test acc.
Exp	Exp	88.3%
Exp	Poly4	88.3%
Exp	Poly3	88.2%
Exp	Poly2	87.4%
Exp	Linear	80.9%

- **Polynomial kernels at second layer suffice!**
- **State-of-the-art for kernels on Cifar10** (at a large computational cost...)
    - Shankar et al. (2020): 88.2% with 10 layers (90% with data augmentation)

Statistical Benefits with Two Layers (B., 2022)

- Consider invariant $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Assume $\mathbb{E}_x[k(x_u, x_{u'})k(x_v, x_{v'})] \leq \epsilon$ if $u \neq u'$ or $v \neq v'$
- Compare different pooling layers ($h_1, h_2 \in \{\text{global}, \delta\}$) and patch sizes ($s_2$):

Statistical Benefits with Two Layers (B., 2022)

- Consider invariant $f^*(x) = \sum_{u,v\in\Omega} g^*(x_u, x_v)$
- Assume $\mathbb{E}_x[k(x_u, x_{u'})k(x_v, x_{v'})] \leq \epsilon$ if $u \neq u'$ or $v \neq v'$
- Compare different pooling layers ($h_1, h_2 \in \{\text{global}, \delta\}$) and patch sizes ($s_2$):

Excess risk bounds when $g^* \in \mathcal{H}_k \otimes \mathcal{H}_k$ (slow rates)

h_1	h_2	s_2	$R(\hat{f}_n) - R(f^*)$ (for $\epsilon \to 0$)
δ	δ	$\|\Omega\|$	$\|g^*\| \|\Omega\|^{2.5}/\sqrt{n}$
δ	global	$\|\Omega\|$	$\|g^*\| \|\Omega\|^{2}/\sqrt{n}$
global	global	$\|\Omega\|$	$\|g^*\| \|\Omega\|/\sqrt{n}$
global	global or δ	1	$\|g^*\|/\sqrt{n}$

Statistical Benefits with Two Layers (B., 2022)

- Consider invariant $f^*(x) = \sum_{u,v \in \Omega} g^*(x_u, x_v)$
- Assume $\mathbb{E}_x[k(x_u, x_{u'})k(x_v, x_{v'})] \leq \epsilon$ if $u \neq u'$ or $v \neq v'$
- Compare different pooling layers ($h_1, h_2 \in \{\text{global}, \delta\}$) and patch sizes ($s_2$):

Excess risk bounds when $g^* \in \mathcal{H}_k \otimes \mathcal{H}_k$ (slow rates)

h_1	h_2	s_2	$R(\hat{f}_n) - R(f^*)$ (for $\epsilon \to 0$)
δ	δ	$\lvert\Omega\rvert$	$\lVert g^*\rVert \lvert\Omega\rvert^{2.5}/\sqrt{n}$
δ	global	$\lvert\Omega\rvert$	$\lVert g^*\rVert \lvert\Omega\rvert^{2}/\sqrt{n}$
global	global	$\lvert\Omega\rvert$	$\lVert g^*\rVert \lvert\Omega\rvert/\sqrt{n}$
global	global or δ	1	$\lVert g^*\rVert/\sqrt{n}$

Polynomial gains in $|\Omega|$ when using the right architecture![1]

[1] Best $\approx$ deep sets (Zaheer et al., 2017)

Concluding Remarks

Benefits of deep convolutional models

- Pooling improves generalization under invariance and stability
- Locality + depth + pooling capture structured interaction models with invariances

Concluding Remarks

Benefits of deep convolutional models

- Pooling improves generalization under invariance and stability
- Locality + depth + pooling capture structured interaction models with invariances

Kernels can help us understand structured architectures

Concluding Remarks

Benefits of deep convolutional models

- Pooling improves generalization under invariance and stability
- Locality + depth + pooling capture structured interaction models with invariances

Kernels can help us understand structured architectures

What's missing?

- Sparsity/adaptivity
  - First layer: adaptive convolutional filters (Gabors)
  - Following layers: structured interactions/symmetries
- Beyond CNNs
  - GNNs, Transformers

References I

A. B. Approximation and learning with deep convolutional models: a kernel perspective. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

A. B., L. Venturi, and J. Bruna. On the sample complexity of learning with geometric stability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

N. Cohen and A. Shashua. Inductive bias of deep convolutional networks through pooling geometry. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory (COLT)*, 2016.

A. Favero, F. Cagnetta, and M. Wyart. Locality defeats the curse of dimensionality in convolutional teacher-student scenarios. *arXiv preprint arXiv:2106.08619*, 2021.

S. Gunasekar, J. D. Lee, D. Soudry, and N. Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

References II

B. Haasdonk and H. Burkhardt. Invariant kernel functions for pattern analysis and machine learning. *Machine learning*, 68(1):35–61, 2007.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

M. Jagadeesan, I. Razenshteyn, and S. Gunasekar. Inductive bias of multi-channel linear convolutional networks with bounded weight norm. In *Conference on Learning Theory (COLT)*, 2022.

Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Z. Li, R. Wang, D. Yu, S. S. Du, W. Hu, R. Salakhutdinov, and S. Arora. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

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

S. Mei, T. Misiakiewicz, and A. Montanari. Learning with invariances in random features and kernel models. In *Conference on Learning Theory (COLT)*, 2021.

References III

H. N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

T. Misiakiewicz and S. Mei. Learning with convolution and pooling operations in kernel methods. *arXiv preprint arXiv:2111.08308*, 2021.

Y. Mroueh, S. Voinea, and T. A. Poggio. Learning with group invariant features: A kernel perspective. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

R. M. Neal. *Bayesian learning for neural networks*. Springer, 1996.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

N. Razin, A. Maman, and N. Cohen. Implicit regularization in hierarchical tensor factorization and deep convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

N. C. Saldanha and C. Tomei. The accumulated distribution of quadratic forms on the sphere. *Linear algebra and its applications*, 245:335–351, 1996.

J. Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48(4):1875–1897, 2020.

References IV

V. Shankar, A. Fang, W. Guo, S. Fridovich-Keil, L. Schmidt, J. Ragan-Kelley, and B. Recht. Neural kernels without tangents. *arXiv preprint arXiv:2003.02237*, 2020.

M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2): 742–769, 2018.

M. Telgarsky. Benefits of depth in neural networks. In *Conference on learning theory*, pages 1517–1539. PMLR, 2016.

G. Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola. Deep sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.