Inductive Bias of Multi-Channel Linear Convolutional Networks with Bounded Weight Norm

Meena Jagadeesan (UC Berkeley)

Joint work w/ Ilya Razenshteyn (CipherMode Labs) & Suriya Gunasekar (MSR)

COLT 2022

The Weight Norm of a Network

“The size [magnitude] of the weights is more important than the size [number of parameters] of the network.” (Bartlett, ‘97)

The Weight Norm of a Network

"The size [magnitude] of the weights is more important than the size [number of parameters] of the network." (Bartlett, '97)

The ℓ_2 norm of the weights is "controlled" during training:

- Implicit regularization of gradient descent (e.g. Nacson et al. '19, Lyu and Li '20, etc.)
- Explicit regularization (e.g. weight decay, etc.)

The Weight Norm of a Network

"The size [magnitude] of the weights is more important than the size [number of parameters] of the network." (Bartlett, '97)

The ℓ_2 norm of the weights is "controlled" during training:

- Implicit regularization of gradient descent (e.g. Nacson et al. '19, Lyu and Li '20, etc.)
- Explicit regularization (e.g. weight decay, etc.)

How does controlling the ℓ_2 norm affect which functions are learned?

From Parameter Space to Function Space

The ℓ_2 norm of the network weights is a *parameter space view*.

From Parameter Space to Function Space

The ℓ_2 norm of the network weights is a *parameter space view*.

But how should this be interpreted in function space?

From Parameter Space to Function Space

The ℓ_2 norm of the network weights is a *parameter space view*.

But how should this be interpreted in function space?

Induced regularizer = the minimum norm of weights needed to realize a function using a given network architecture or model class

From Parameter Space to Function Space

The ℓ_2 norm of the network weights is a *parameter space view.*

But how should this be interpreted in function space?

Induced regularizer = the minimum norm of weights needed to realize a function using a given network architecture or model class

$$\mathcal{R}_\Phi(f) := \inf_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_2^2 \quad \text{s.t.,} \quad \forall \mathrm{x}, f(\mathrm{x}) = \Phi(\boldsymbol{\theta}; \mathrm{x})$$

From Parameter Space to Function Space

The ℓ_2 norm of the network weights is a *parameter space view.*

But how should this be interpreted in function space?

Induced regularizer = the minimum norm of weights needed to realize a function using a given network architecture or model class

$$\mathcal{R}_\Phi(f) := \inf_{\theta} \|\theta\|_2^2 \quad \text{s.t.,} \quad \forall x, f(x) = \Phi(\theta; x)$$

Model class

Function on input space

Induced Regularizer is Highly Architecture Dependent

$$\mathcal{R}_\Phi(f) := \inf_{\theta} \|\theta\|_2^2 \quad \text{s.t.}, \quad \forall x, f(x) = \Phi(\theta; x)$$

Fully connected networks: ℓ_2 norm of the linear predictor (Gunasekar et al., 2018)

Diagonal networks with L layers: $\ell_{2/L}$ norm of the linear predictor (Gunasekar et al., 2018)

Linear convolutional networks with full-dimensional kernels: ℓ_1 norm of the predictor in Fourier space (Gunasekar et al., 2018, Pilanci and Ergen '20, Yun et al. '21)

Infinite-width two-layer ReLU networks: ℓ_1 norm of the partial derivatives of the Radon transform of the function (Savarese et al. '19, Pilanci and Ergen '20)

Induced Regularizer is Highly Architecture Dependent

$$\mathcal{R}_\Phi(f) := \inf_{\theta} \|\theta\|_2^2 \quad \text{s.t.}, \quad \forall x, f(x) = \Phi(\theta; x)$$

Fully connected networks: ℓ_2 norm of the linear predictor (Gunasekar et al., 2018)

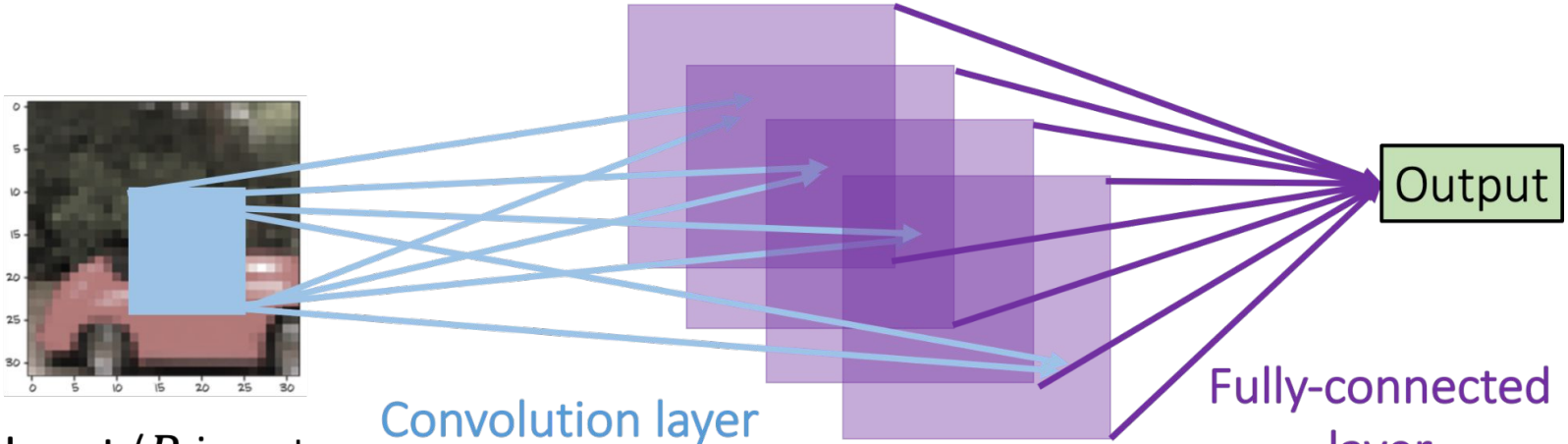
Diagonal networks with L layers: $\ell_{2/L}$ norm of the linear predictor (Gunasekar et al., 2018)

Linear convolutional networks with full-dimensional kernels: ℓ_1 norm of the predictor in Fourier space (Gunasekar et al., 2018, Pilanci and Ergen '20, Yun et al. '21)

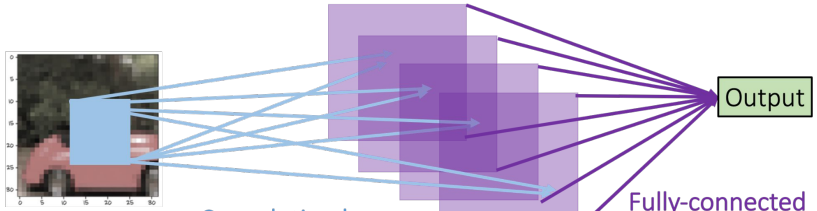
Infinite-width two-layer ReLU networks: ℓ_1 norm of the partial derivatives of the Radon transform of the function (Savarese et al. '19, Pilanci and Ergen '20)

Our focus: multi-channel linear convolutional networks with arbitrary kernel size

2-Layer Linear Convolutional Neural Networks



Induced Regularizer for 2-Layer Linear CNNs

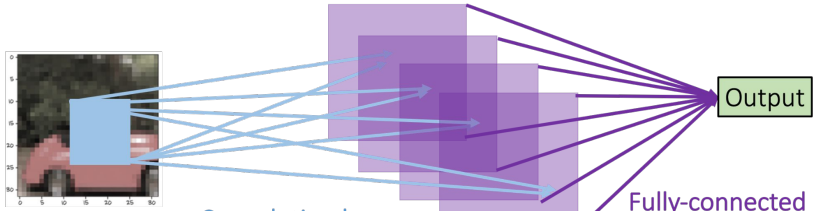


This talk: we focus on single-channel inputs (see the paper for treatment of general R).

Observations:

- All linear functions can be represented regardless of C and K.
- Adding more channels (*weakly) decreases* $\mathcal{R}_\Phi(f)$
- Increasing the kernel size (*weakly) decreases* $\mathcal{R}_\Phi(f)$

Induced Regularizer for 2-Layer Linear CNNs



This talk: we focus on single-channel inputs (see the paper for treatment of general R).

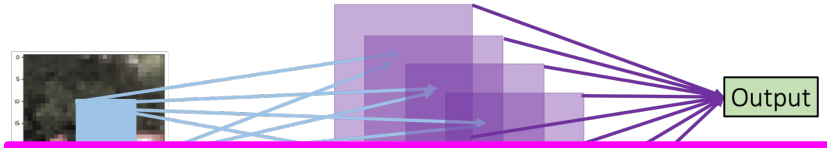
Observations:

- All linear functions can be represented regardless of C and K.
- Adding more channels (*weakly) decreases* $\mathcal{R}_\Phi(f)$
- Increasing the kernel size (*weakly) decreases* $\mathcal{R}_\Phi(f)$

Induced regularizer in special cases:

- K = 1, any C: ℓ_2 norm of the linear predictor
- K = D, any C: ℓ_1 norm of the linear predictor in Fourier space.
- Appears to be *no clean closed-form expression* even for K = 2.

Induced Regularizer for 2-Layer Linear CNNs



This talk: we focus on single-channel inputs (see the paper for treatment of general P)

Main Result: $\mathcal{R}_\Phi(f)$ is **independent** of C for any kernel size K.

Observations:

- All linear functions can be represented regardless of C and K.
- Adding more channels *(weakly) decreases* $\mathcal{R}_\Phi(f)$
- Increasing the kernel size *(weakly) decreases* $\mathcal{R}_\Phi(f)$

Induced regularizer in special cases:

- K = 1, any C: ℓ_2 norm of the linear predictor
- K = D, any C: ℓ_1 norm of the linear predictor in Fourier space.
- Appears to be *no clean closed-form expression* even for K = 2.

Main Theorem

Theorem: For networks with single-channel inputs, the induced regularizer $\mathcal{R}_\Phi(f)$ is **independent of the number of output channels C for any kernel size K.**

Main Theorem

Theorem: For networks with single-channel inputs, the induced regularizer $\mathcal{R}_\Phi(f)$ is **independent of the number of output channels C for any kernel size K.**

Proof sketch:

- Need to analyze $\mathcal{R}_\Phi(f)$ implicitly since it does not appear to have a simple closed-form .

Main Theorem

Theorem: For networks with single-channel inputs, the induced regularizer $\mathcal{R}_\Phi(f)$ is **independent of the number of output channels C for any kernel size K.**

Proof sketch:

- Need to analyze $\mathcal{R}_\Phi(f)$ implicitly since it does not appear to have a simple closed-form .
- **Key technical tool:** We express $\mathcal{R}_\Phi(f)$ as a **semidefinite program with a rank <= C constraint.**

Main Theorem

Theorem: For networks with single-channel inputs, the induced regularizer $\mathcal{R}_\Phi(f)$ is **independent of the number of output channels C for any kernel size K.**

Proof sketch:

- Need to analyze $\mathcal{R}_\Phi(f)$ implicitly since it does not appear to have a simple closed-form .
- **Key technical tool:** We express $\mathcal{R}_\Phi(f)$ as a **semidefinite program with a rank <= C constraint.**
- Tightness of SDP relaxation ⇔ independence of the induced regularizer to C.

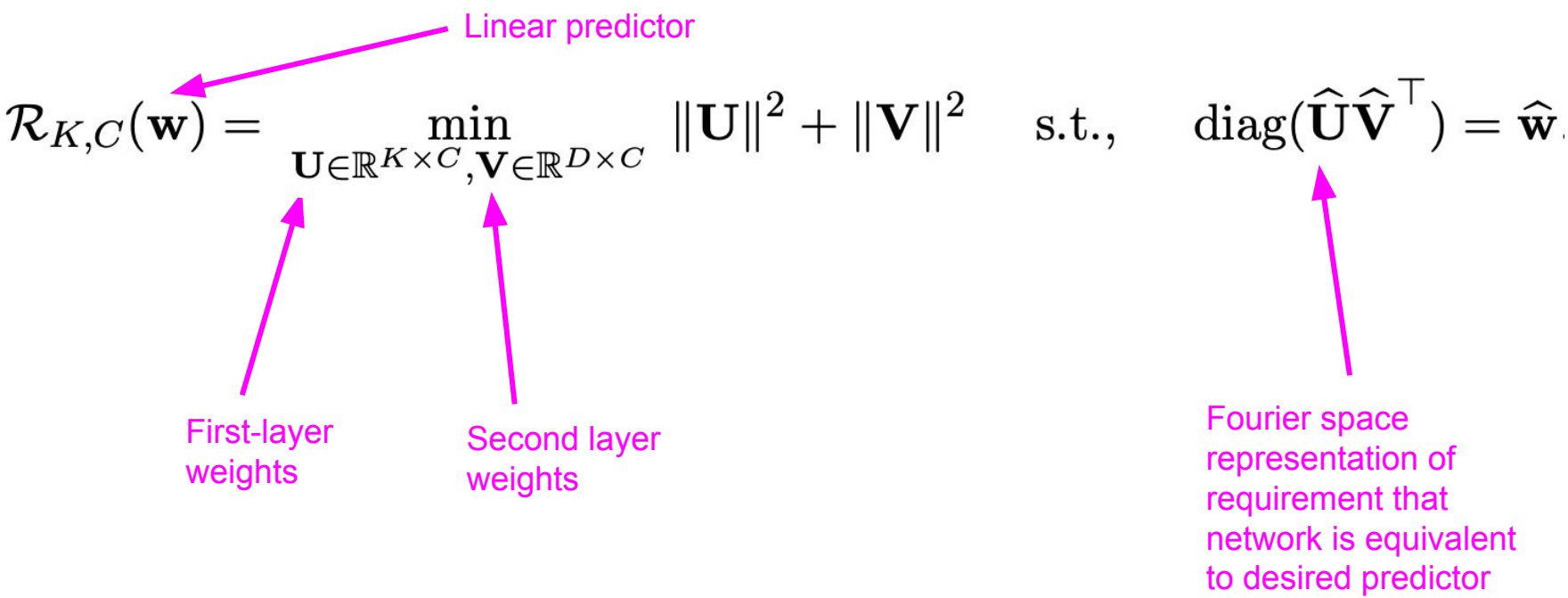
Main Theorem

Theorem: For networks with single-channel inputs, the induced regularizer $\mathcal{R}_\Phi(f)$ is **independent of the number of output channels C for any kernel size K.**

Proof sketch:

- Need to analyze $\mathcal{R}_\Phi(f)$ implicitly since it does not appear to have a simple closed-form .
- **Key technical tool:** We express $\mathcal{R}_\Phi(f)$ as a **semidefinite program with a rank <= C constraint.**
- Tightness of SDP relaxation ⇔ independence of the induced regularizer to C.
- Main part of proof: showing tightness of the SDP relaxation

Induced Regularizer Definition for CNNs



Induced Regularizer as a Semidefinite Program

$$\mathcal{R}_{K,C}(\mathbf{w}) = \min_{\mathbf{U}\in\mathbb{R}^{K\times C},\mathbf{V}\in\mathbb{R}^{D\times C}} \|\mathbf{U}\|^2 + \|\mathbf{V}\|^2 \quad \text{s.t.,} \quad \operatorname{diag}(\widehat{\mathbf{U}}\widehat{\mathbf{V}}^\top) = \widehat{\mathbf{w}}$$

Induced Regularizer as a Semidefinite Program

$$\mathcal{R}_{K,C}(\mathbf{w}) = \min_{\mathbf{U}\in\mathbb{R}^{K\times C},\mathbf{V}\in\mathbb{R}^{D\times C}} \|\mathbf{U}\|^2 + \|\mathbf{V}\|^2 \quad \text{s.t.,} \quad \operatorname{diag}(\widehat{\mathbf{U}}\widehat{\mathbf{V}}^\top) = \widehat{\mathbf{w}}$$

Requirement that network corresponds to desired predictor

$$\mathcal{R}_{K,C}(\mathbf{w}) = \min_{\mathbf{Z}\succcurlyeq 0} \langle \mathbf{Z}, \mathbf{I}\rangle \quad \text{s.t.,} \quad \begin{aligned} &\forall_{d\in[D]}, \langle \mathbf{Z}, \mathbf{A}_d^{\text{real}}\rangle = 2\operatorname{Re}(\widehat{\mathbf{w}}[d]) \\ &\forall_{d\in[D]}, \langle \mathbf{Z}, \mathbf{A}_d^{\text{img}}\rangle = 2\operatorname{Im}(\widehat{\mathbf{w}}[d]) \\ &\operatorname{rank}(\mathbf{Z}) \le C. \end{aligned}$$

(D+K) x (D + K) matrix intended to correspond to:

$$\begin{bmatrix} \mathbf{U}\mathbf{U}^\top & \mathbf{U}\mathbf{V}^\top \\ \mathbf{V}\mathbf{U}^\top & \mathbf{V}\mathbf{V}^\top \end{bmatrix}$$

Number of output channels induces a **rank constraint**

Induced Regularizer as a Semidefinite Program

$$\mathcal{R}_{K,C}(\mathbf{w}) = \min_{\mathbf{U}\in\mathbb{R}^{K\times C},\mathbf{V}\in\mathbb{R}^{D\times C}} \|\mathbf{U}\|^2 + \|\mathbf{V}\|^2 \quad \text{s.t.,} \quad \operatorname{diag}(\widehat{\mathbf{U}}\widehat{\mathbf{V}}^\top) = \widehat{\mathbf{w}}$$

Requirement that network corresponds to desired predictor

$$\mathcal{R}_{K,C}(\mathbf{w}) = \min_{\mathbf{Z}\succcurlyeq 0} \langle \mathbf{Z}, \mathbf{I}\rangle \quad \text{s.t.,} \quad \forall_{d\in[D]}, \langle \mathbf{Z}, \mathbf{A}_d^{\text{real}}\rangle = 2\operatorname{Re}(\widehat{\mathbf{w}}[d])$$

$$\forall_{d\in[D]}, \langle \mathbf{Z}, \mathbf{A}_d^{\text{img}}\rangle = 2\operatorname{Im}(\widehat{\mathbf{w}}[d])$$

$$\operatorname{rank}(\mathbf{Z}) \leq C.$$

(D+K) x (D + K) matrix intended to correspond to:

$$\begin{bmatrix} \mathbf{U}\mathbf{U}^\top & \mathbf{U}\mathbf{V}^\top \\ \mathbf{V}\mathbf{U}^\top & \mathbf{V}\mathbf{V}^\top \end{bmatrix}$$

Number of output channels induces a **rank constraint**

Proof of SDP tightness

Suffices to show that there exists a rank 1 solution

Boils down to a natural fact about convolutions

Lemma *For any $1 \leq K \leq D$, and for any vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^K$, there exists a vector $\mathbf{c} \in \mathbb{R}^K$ such that $\mathbf{a} \star \mathbf{a} + \mathbf{b} \star \mathbf{b} = \mathbf{c} \star \mathbf{c}$, where convolutions are w.r.t. dimension D.*

To show this fact, we leverage the **polynomial representation of convolutions** to show that it suffices to construct a polynomial p_c satisfying the following identity:

$$x^{K-1} p_{\mathbf{c}}(x) p_{\mathbf{c}}(1/x) = x^{K-1} p_{\mathbf{a}}(x) p_{\mathbf{a}}(1/x) + x^{K-1} p_{\mathbf{b}}(x) p_{\mathbf{b}}(1/x)$$

Extensions

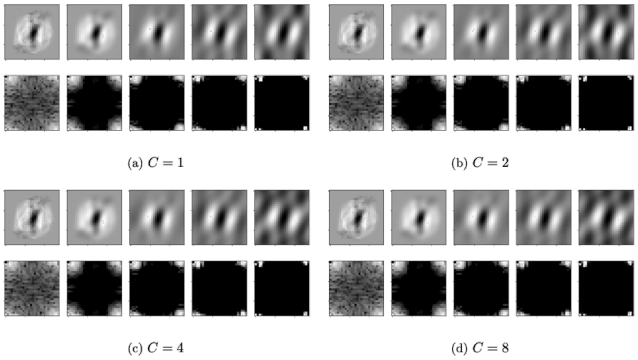
We extend our results to networks with multi-channel inputs.

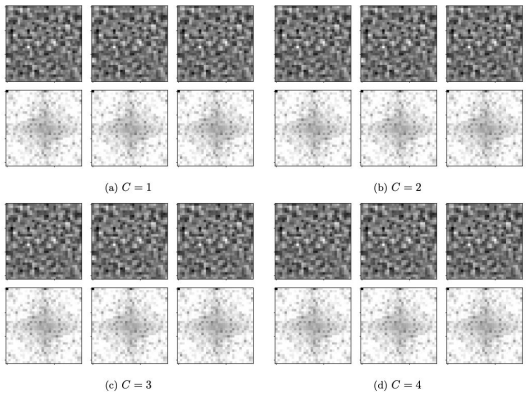
- No longer get independence for any C ≥ 1
- We show $\mathcal{R}_\Phi(f)$ is independent of C as long as C ≥ R * K.

Conjecture: $\mathcal{R}_\Phi(f)$ is independent of C as long as C ≥ R.

We empirically connect our results to the implicit regularization of gradient descent.

- *Asymptotic behavior on 2-layer linear CNNs appears to be invariant for C ≥ R on MNIST and CIFAR-10*





Conclusion and Future Work

We studied the induced bias of the ℓ_2 norm of the weights for two-layer linear CNNs.

We showed that for single-channel inputs, the induced regularizer is **independent of the number of output channels** regardless of the kernel size.

We partially generalized this to multi-channel inputs and empirically connected it to the implicit regularization of gradient descent.

Future work:

- Prove conjecture for multi-channel inputs
- Study the role of other architectural features (e.g. pooling, depth, nonlinearities, etc.)