Optimization, Robustness and Attention in Deep Learning: Insights from Random and NTK Feature Models

Marco Mondelli

Institute of Science and Technology Austria (ISTA)

ISL Colloquium, April 11, 2024

Institute of Science and Technology Austria

Supervised learning

Data: $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\} \sim_{\text{i.i.d.}} \mathbb{P}(\mathbb{R}^d \times \mathbb{R})$

Goal: Given $(\boldsymbol{x}, y) \sim \mathbb{P}$, find $f : \mathbb{R}^d \to \mathbb{R}$ to predict y from $\boldsymbol{x}$

Supervised learning

Data: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \sim_{\text{i.i.d.}} \mathbb{P}(\mathbb{R}^d \times \mathbb{R})$

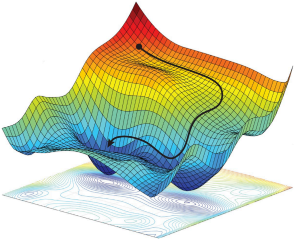
Goal: Minimize empirical risk $L_f(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\big(y_i - f(\boldsymbol{x}_i; \boldsymbol{\theta})\big)^2$

Supervised learning

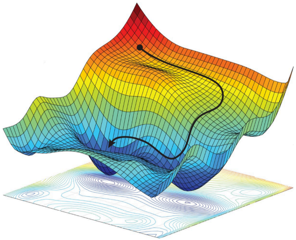
Data: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \sim_{\text{i.i.d.}} \mathbb{P}(\mathbb{R}^d \times \mathbb{R})$

Goal: Minimize empirical risk $L_f(\boldsymbol{\theta}) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \big(y_i - f(\boldsymbol{x}_i; \boldsymbol{\theta})\big)^2$

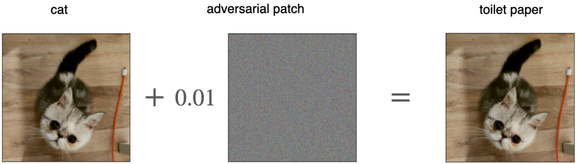
Gradient flow: $\dot{\boldsymbol{\theta}}(t) = -\nabla_{\boldsymbol{\theta}} L_f(\boldsymbol{\theta}(t))$



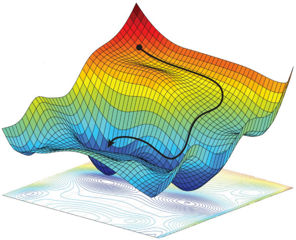
optimization



optimization

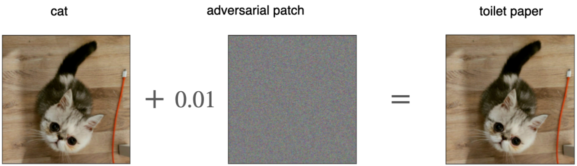


robustness

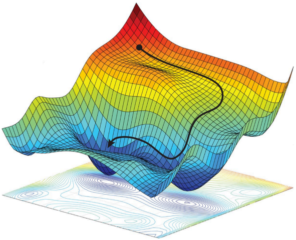


optimization

attention

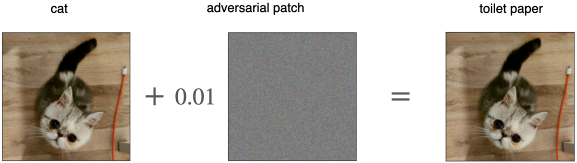


robustness



optimization

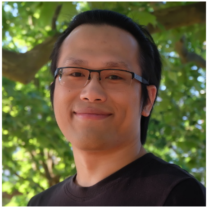
attention



robustness

Simone Bombari (ISTA)

Mohammad Amani (ISTA $\rightarrow$ EPFL)



Quynh Nguyen (MPI)

Guido Montufar (MPI & UCLA)

Insights from the Neural Tangent Kernel (NTK)

$$\dot{\boldsymbol{\theta}}(t) = -\nabla_{\boldsymbol{\theta}} L_f(\boldsymbol{\theta}(t)), \quad \boldsymbol{\theta} \in \mathbb{R}^p, \quad p \gg n$$

Idea: $\boldsymbol{\theta}(0)$ not too far from an interpolator, so throughout the gradient flow trajectory we have

$$f(\boldsymbol{x}; \boldsymbol{\theta}(t)) \approx f(\boldsymbol{x}; \boldsymbol{\theta}(0)) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}(0)), \boldsymbol{\theta}(t) - \boldsymbol{\theta}(0) \rangle$$

Insights from the Neural Tangent Kernel (NTK)

$$\dot{\boldsymbol{\theta}}(t) = -\nabla_{\boldsymbol{\theta}} L_f(\boldsymbol{\theta}(t)), \quad \boldsymbol{\theta} \in \mathbb{R}^p, \quad p \gg n$$

Idea: $\boldsymbol{\theta}(0)$ not too far from an interpolator, so throughout the gradient flow trajectory we have

$$f(\boldsymbol{x}; \boldsymbol{\theta}(t)) \approx f(\boldsymbol{x}; \boldsymbol{\theta}(0)) + \langle \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}; \boldsymbol{\theta}(0)), \boldsymbol{\theta}(t) - \boldsymbol{\theta}(0) \rangle$$

$$L_f(\boldsymbol{\theta}(t)) \leq L_f(\boldsymbol{\theta}(0)) e^{-\lambda_{\min}(\boldsymbol{K}) t/2}$$

- NTK: $\boldsymbol{K} = \boldsymbol{J}_f(\boldsymbol{\theta}_0) \boldsymbol{J}_f(\boldsymbol{\theta}_0)^\top \in \mathbb{R}^{n \times n}$
- Jacobian of f at initialization: $\boldsymbol{J}_f(\boldsymbol{\theta}_0) \in \mathbb{R}^{n \times p}$

[Jacot et al., 2018; Chizat et al., 2019; Du et al., 2019; Oymak et al., 2019; Bartlett et al., 2021; ...]

Convergence for (very) wide networks

	Deep?	Activation	Layer Width	# Wide Layers
[Oymak et al., '20]	No	Smooth	$\Omega(n^2\lambda_0^{-2})$	x
[Montanari & Zhong, '22]	No	General	$\tilde{\Omega}(n/d)$	x
[Allen-Zhu et al., '19]	Yes	General	$\Omega(n^{24}L^{12}\phi^{-4})$	All
[Zou et al., '19]	Yes	ReLU	$\Omega(n^8L^{12}\phi^{-4})$	All
[Du et al., '19]	Yes	Smooth	$\Omega(\frac{n^4 2^{\mathcal{O}(L)}}{\lambda_{\min}^4(\bar{K}^{(L)})})$	All

Convergence for (not so) wide networks

	Deep?	Activation	Layer Width	# Wide Layers
[Oymak et al., '20]	No	Smooth	$\Omega(n^2\lambda_0^{-2})$	x
[Montanari & Zhong, '22]	No	General	$\tilde{\Omega}(n/d)$	x
[Allen-Zhu et al., '18]	Yes	General	$\Omega(n^{24}L^{12}\phi^{-4})$	All
[Zou et al., '19]	Yes	ReLU	$\Omega(n^8L^{12}\phi^{-4})$	All
[Du et al., '19]	Yes	Smooth	$\Omega(\frac{n^4 2^{\mathcal{O}(L)}}{\lambda^4_{\min}(\bar{K}^{(L)})})$	All
[Nguyen and M., '20]	**Yes**	**Smooth**	$\boldsymbol{n}$	**One**

Need only **one wide layer + pyramidal topology**

Q. Nguyen and M. Mondelli, "Global Convergence of Deep Networks with One Wide Layer Followed by Pyramidal Topology", *NeurIPS*, 2020.

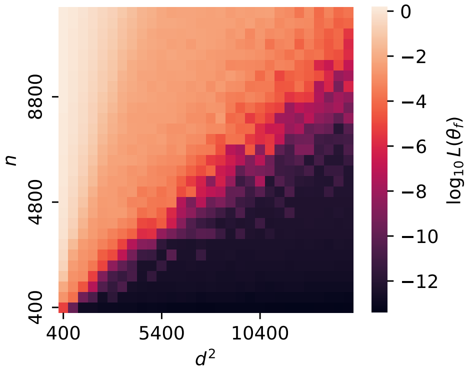
Convergence for (not so) wide networks

	Deep?	Activation	Layer Width	# Wide Layers
[Oymak et al., '20]	No	Smooth	$\Omega(n^2\lambda_0^{-2})$	x
[Montanari & Zhong, '22]	No	General	$\tilde{\Omega}(n/d)$	x
[Allen-Zhu et al., '18]	Yes	General	$\Omega(n^{24}L^{12}\phi^{-4})$	All
[Zou et al., '19]	Yes	ReLU	$\Omega(n^8L^{12}\phi^{-4})$	All
[Du et al., '19]	Yes	Smooth	$\Omega(\frac{n^4 2^{\mathcal{O}(L)}}{\lambda_{\min}^4(\bar{K}^{(L)})})$	All
[Nguyen and M., '20]	**Yes**	**Smooth**	$\boldsymbol{n}$	**One**
[Bombari, Amani and M., '22]	**Yes**	**Smooth**	$\tilde{\Omega}(\sqrt{n})$	**All**

Need only **minimum over-parameterization**

S. Bombari, M. H. Amani, and M. Mondelli, "Memorization and Optimization in Deep Neural Networks with Minimum Over-parameterization", *NeurIPS*, 2022.

Optimization with minimum over-parameterization



- $\Omega(\sqrt{n})$ neurons **necessary** to interpolate (parameter counting or VC dimension bound [Bartlett et al., 2019])
- Scaling **close to practice** (back-of-the-envelope estimates on CIFAR-10, ImageNet)

Bounding $\lambda_{\min}(\boldsymbol{K})$

$$\boldsymbol{K} = \sum_{k=0}^{L-1} \boldsymbol{F}_k \boldsymbol{F}_k^\top \circ \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^\top$$

- $\boldsymbol{F}_k = [\boldsymbol{f}_k(\boldsymbol{x}_1), \ldots, \boldsymbol{f}_k(\boldsymbol{x}_n)]^\top$, with $\boldsymbol{f}_k(\boldsymbol{x}_i) =$ feature vector at layer k with input $\boldsymbol{x}_i$
- $\boldsymbol{B}_{k+1} = [\boldsymbol{b}_{k+1}(\boldsymbol{x}_1), \ldots, \boldsymbol{b}_{k+1}(\boldsymbol{x}_n)]^\top$, with $\boldsymbol{b}_{k+1}(\boldsymbol{x}_i) =$ back-propagation vector at layer $k+1$ with input $\boldsymbol{x}_i$

Bounding $\lambda_{\min}(\boldsymbol{K})$

$$\boldsymbol{K} = \sum_{k=0}^{L-1} \boldsymbol{F}_k \boldsymbol{F}_k^\top \circ \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^\top$$

- $\boldsymbol{F}_k = [\boldsymbol{f}_k(\boldsymbol{x}_1), \ldots, \boldsymbol{f}_k(\boldsymbol{x}_n)]^\top$, with $\boldsymbol{f}_k(\boldsymbol{x}_i) =$ feature vector at layer k with input $\boldsymbol{x}_i$
- $\boldsymbol{B}_{k+1} = [\boldsymbol{b}_{k+1}(\boldsymbol{x}_1), \ldots, \boldsymbol{b}_{k+1}(\boldsymbol{x}_n)]^\top$, with $\boldsymbol{b}_{k+1}(\boldsymbol{x}_i) =$ back-propagation vector at layer $k+1$ with input $\boldsymbol{x}_i$

First attempt: use matrix concentration on $\boldsymbol{F}_k \boldsymbol{F}_k^\top$

Q. Nguyen, M. Mondelli and G. Montufar, "Tight Bounds on the Smallest Eigenvalue of the Neural Tangent Kernel for Deep ReLU Networks", *ICML*, 2021.

Bounding $\lambda_{\min}(\boldsymbol{K})$ with one wide layer

$$\boldsymbol{K} = \sum_{k=0}^{L-1} \boldsymbol{F}_k \boldsymbol{F}_k^\top \circ \boldsymbol{B}_{k+1} \boldsymbol{B}_{k+1}^\top$$

- $\boldsymbol{F}_k = [\boldsymbol{f}_k(\boldsymbol{x}_1), \ldots, \boldsymbol{f}_k(\boldsymbol{x}_n)]^\top$, with $\boldsymbol{f}_k(\boldsymbol{x}_i) =$ feature vector at layer k with input $\boldsymbol{x}_i$
- $\boldsymbol{B}_{k+1} = [\boldsymbol{b}_{k+1}(\boldsymbol{x}_1), \ldots, \boldsymbol{b}_{k+1}(\boldsymbol{x}_n)]^\top$, with $\boldsymbol{b}_{k+1}(\boldsymbol{x}_i) =$ back-propagation vector at layer $k+1$ with input $\boldsymbol{x}_i$

First attempt: use matrix concentration on $\boldsymbol{F}_k \boldsymbol{F}_k^\top$

Need **one wide layer** with $\Omega(n)$ neurons!

Q. Nguyen, M. Mondelli and G. Montufar, "Tight Bounds on the Smallest Eigenvalue of the Neural Tangent Kernel for Deep ReLU Networks", *ICML*, 2021.

Bounding $\lambda_{\min}(\boldsymbol{K})$ with minimum over-parameterization

$$\boldsymbol{K} \succeq \boldsymbol{F}_{L-2}\boldsymbol{F}_{L-2}^\top \circ \boldsymbol{B}_{L-1}\boldsymbol{B}_{L-1}^\top := \boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top$$

- $(\boldsymbol{J}_{L-2})_{i,:} = \boldsymbol{f}_{L-2}(\boldsymbol{x}_i) \otimes \boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$
- $\boldsymbol{f}_{L-2}(\boldsymbol{x}_i) =$ feature vector at layer $L-2$ with input $\boldsymbol{x}_i$
- $\boldsymbol{b}_{L-1}(\boldsymbol{x}_i) =$ back-propagation vector at layer $L-1$ with input $\boldsymbol{x}_i$

Second attempt: directly center $\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top$

Bounding $\lambda_{\min}(\boldsymbol{K})$ with minimum over-parameterization

$$\boldsymbol{K} \succeq \boldsymbol{F}_{L-2}\boldsymbol{F}_{L-2}^\top \circ \boldsymbol{B}_{L-1}\boldsymbol{B}_{L-1}^\top := \boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top$$

- $(\boldsymbol{J}_{L-2})_{i,:} = \boldsymbol{f}_{L-2}(\boldsymbol{x}_i) \otimes \boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$
- $\boldsymbol{f}_{L-2}(\boldsymbol{x}_i)$ = feature vector at layer $L-2$ with input $\boldsymbol{x}_i$
- $\boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$ = back-propagation vector at layer $L-1$ with input $\boldsymbol{x}_i$

Second attempt: directly center $\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top$

$$\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top \gtrsim \boldsymbol{J}_{FB}\boldsymbol{J}_{FB}^\top$$

- $(\boldsymbol{J}_{FB})_{i,:} = \tilde{\boldsymbol{f}}_{L-2}(\boldsymbol{x}_i) \otimes \tilde{\boldsymbol{b}}_{L-1}(\boldsymbol{x}_i)$
- $\tilde{\boldsymbol{f}}_{L-2}(\boldsymbol{x}_i) = \boldsymbol{f}_{L-2}(\boldsymbol{x}_i) - \mathbb{E}_{\boldsymbol{x}_i}\boldsymbol{f}_{L-2}(\boldsymbol{x}_i)$
- $\tilde{\boldsymbol{b}}_{L-1}(\boldsymbol{x}_i) = \boldsymbol{b}_{L-1}(\boldsymbol{x}_i) - \mathbb{E}_{\boldsymbol{x}_i}\boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$

Features and back-propagations centered **together**

Bounding $\lambda_{\min}(\boldsymbol{K})$ with minimum over-parameterization

$$\boldsymbol{K} \succeq \boldsymbol{F}_{L-2}\boldsymbol{F}_{L-2}^{\top} \circ \boldsymbol{B}_{L-1}\boldsymbol{B}_{L-1}^{\top} := \boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^{\top}$$

- $(\boldsymbol{J}_{L-2})_{i,:} = \boldsymbol{f}_{L-2}(\boldsymbol{x}_i) \otimes \boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$
- $\boldsymbol{f}_{L-2}(\boldsymbol{x}_i)$ = feature vector at layer $L-2$ with input $\boldsymbol{x}_i$
- $\boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$ = back-propagation vector at layer $L-1$ with input $\boldsymbol{x}_i$

Second attempt: directly center $\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^{\top}$

$$\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^{\top} \gtrsim \boldsymbol{J}_{FB}\boldsymbol{J}_{FB}^{\top} \gtrsim \tilde{\boldsymbol{J}}_{FB}\tilde{\boldsymbol{J}}_{FB}^{\top}$$

- $(\boldsymbol{J}_{FB})_{i,:} = \tilde{\boldsymbol{f}}_{L-2}(\boldsymbol{x}_i) \otimes \tilde{\boldsymbol{b}}_{L-1}(\boldsymbol{x}_i)$
- $\tilde{\boldsymbol{J}}_{FB} = \boldsymbol{J}_{FB} - \mathbb{E}_{\boldsymbol{X}}\boldsymbol{J}_{FB}$

Center **again** the whole matrix

Bounding $\lambda_{\min}(\boldsymbol{K})$ with minimum over-parameterization

$$\boldsymbol{K} \succeq \boldsymbol{F}_{L-2}\boldsymbol{F}_{L-2}^{\top} \circ \boldsymbol{B}_{L-1}\boldsymbol{B}_{L-1}^{\top} := \boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^{\top}$$

- $(\boldsymbol{J}_{L-2})_{i,:} = \boldsymbol{f}_{L-2}(\boldsymbol{x}_i) \otimes \boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$
- $\boldsymbol{f}_{L-2}(\boldsymbol{x}_i)$ = feature vector at layer $L-2$ with input $\boldsymbol{x}_i$
- $\boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$ = back-propagation vector at layer $L-1$ with input $\boldsymbol{x}_i$

Second attempt: directly center $\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^{\top}$

$$\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^{\top} \gtrsim \boldsymbol{J}_{FB}\boldsymbol{J}_{FB}^{\top} \gtrsim \tilde{\boldsymbol{J}}_{FB}\tilde{\boldsymbol{J}}_{FB}^{\top} \approx \boldsymbol{I}_n$$

- $(\boldsymbol{J}_{FB})_{i,:} = \tilde{\boldsymbol{f}}_{L-2}(\boldsymbol{x}_i) \otimes \tilde{\boldsymbol{b}}_{L-1}(\boldsymbol{x}_i)$
- $\tilde{\boldsymbol{J}}_{FB} = \boldsymbol{J}_{FB} - \mathbb{E}_{\boldsymbol{X}}\boldsymbol{J}_{FB}$

Concentration for i.i.d. rows with well-controlled $\|\cdot\|_{\psi_1}$ [Adamczak et al., 2011]

Bounding $\lambda_{\min}(\boldsymbol{K})$ with minimum over-parameterization

$$\boldsymbol{K} \succeq \boldsymbol{F}_{L-2}\boldsymbol{F}_{L-2}^\top \circ \boldsymbol{B}_{L-1}\boldsymbol{B}_{L-1}^\top := \boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top$$

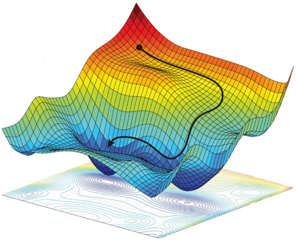
- $(\boldsymbol{J}_{L-2})_{i,:} = \boldsymbol{f}_{L-2}(\boldsymbol{x}_i) \otimes \boldsymbol{b}_{L-1}(\boldsymbol{x}_i)$
- $\boldsymbol{f}_{L-2}(\boldsymbol{x}_i) =$ feature vector at layer $L-2$ with input $\boldsymbol{x}_i$
- $\boldsymbol{b}_{L-1}(\boldsymbol{x}_i) =$ back-propagation vector at layer $L-1$ with input $\boldsymbol{x}_i$

Second attempt: directly center $\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top$

$$\boldsymbol{J}_{L-2}\boldsymbol{J}_{L-2}^\top \gtrsim \boldsymbol{J}_{FB}\boldsymbol{J}_{FB}^\top \gtrsim \tilde{\boldsymbol{J}}_{FB}\tilde{\boldsymbol{J}}_{FB}^\top \approx \boldsymbol{I}_n$$

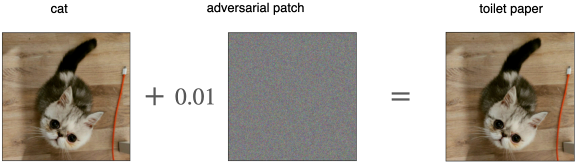
Concentration for i.i.d. rows with well-controlled $\|\cdot\|_{\psi_1}$ [Adamczak et al., 2011]

S. Bombari, M. H. Amani, and M. Mondelli, "Memorization and Optimization in Deep Neural Networks with Minimum Over-parameterization", *NeurIPS*, 2022.



optimization

attention



robustness

Simone Bombari (ISTA)

Shayan Kiyani (ISTA $\rightarrow$ UPenn)

S. Bombari, S. Kiyani, and M. Mondelli, "Beyond the Universal Law of Robustness: Sharper Laws for Random Features and Neural Tangent Kernels", *ICML*, 2023 (oral).

Robust interpolation needs more parameters

$p > n$ enough for interpolation...

Robust interpolation needs more parameters

$p > n$ enough for interpolation...
but $p > nd$ **necessary** for **robust** interpolation

A Universal Law of Robustness via Isoperimetry

Sébastien Bubeck
Microsoft Research
sebubeck@microsoft.com

Mark Sellke
Stanford University
msellke@stanford.edu

$$\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2 \le \sigma^2 - \epsilon \;\Rightarrow\; \mathrm{Lip}(f) \ge \widetilde{\Omega}\left(\epsilon\sqrt{\frac{nd}{p}}\right).$$

[Bubeck & Sellke, 2021]

Robust interpolation needs more parameters

$p > n$ enough for interpolation...
but $p > nd$ **necessary** for **robust** interpolation

A Universal Law of Robustness via Isoperimetry

Sébastien Bubeck
Microsoft Research
sebubeck@microsoft.com

Mark Sellke
Stanford University
msellke@stanford.edu

$$\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2 \le \sigma^2-\epsilon \;\Rightarrow\; \mathrm{Lip}(f) \ge \widetilde{\Omega}\left(\epsilon\sqrt{\frac{nd}{p}}\right).$$

[Bubeck & Sellke, 2021]

[Bubeck et al., 2021] conjecture it is **sufficient** for two-layer networks

Supervised learning

Data: $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \sim_{\text{i.i.d.}} \mathbb{P}(\mathbb{R}^d \times \mathbb{R})$

Goal: Minimize empirical risk $L_f(\boldsymbol{\theta}) = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \big(y_i - f(\boldsymbol{x}_i; \boldsymbol{\theta})\big)^2$

Gradient flow: $\dot{\boldsymbol{\theta}}(t) = -\nabla_{\boldsymbol{\theta}} L_f(\boldsymbol{\theta}(t))$

Generalized linear regression

Data: $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\} \sim_{\text{i.i.d.}} \mathbb{P}(\mathbb{R}^d \times \mathbb{R})$

Goal: Minimize empirical risk $L_f(\boldsymbol{\theta}) = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \left(y_i - \varphi(\boldsymbol{x}_i)^\top \boldsymbol{\theta}\right)^2$

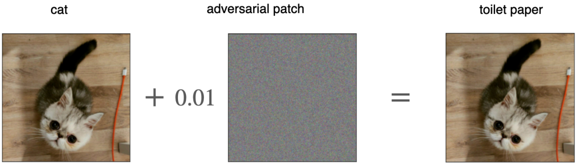
- $\varphi : \mathbb{R}^d \to \mathbb{R}^p$ feature map

Gradient flow solution: $\boldsymbol{\theta}^* = \Phi^\top (\Phi \Phi^\top)^{-1} \boldsymbol{y}$

- $\Phi = [\varphi(\boldsymbol{x}_1), \dots, \varphi(\boldsymbol{x}_n)]^\top \in \mathbb{R}^{n \times p}$ feature matrix
- $\boldsymbol{y} = [y_1, \dots, y_n]^\top \in \mathbb{R}^n$ label vector

Sensitivity to adversarial attacks

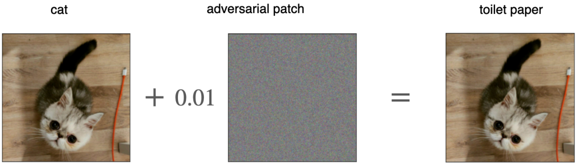
$$\boldsymbol{x} \quad + \quad \Delta_{\mathrm{adv}} \quad = \quad \boldsymbol{x}_{\mathrm{adv}}$$



- $\|\Delta_{\mathrm{adv}}\|_2 \leq \delta \|\boldsymbol{x}\|_2 \qquad (\delta = 0.01)$

Sensitivity to adversarial attacks

$$\boldsymbol{x} \quad + \quad \Delta_{\mathrm{adv}} \quad = \quad \boldsymbol{x}_{\mathrm{adv}}$$

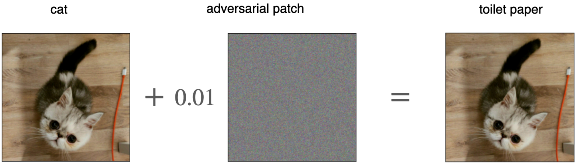


- $\|\Delta_{\mathrm{adv}}\|_2 \leq \delta \|\boldsymbol{x}\|_2 \qquad (\delta = 0.01)$

$$|f(\boldsymbol{x}_{\mathrm{adv}}; \boldsymbol{\theta}) - f(\boldsymbol{x}; \boldsymbol{\theta})| \approx |\nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{\theta})^{\top} \Delta_{\mathrm{adv}}|$$

Sensitivity to adversarial attacks

$$\boldsymbol{x} \qquad + \qquad \Delta_{\mathrm{adv}} \qquad = \qquad \boldsymbol{x}_{\mathrm{adv}}$$



- $\|\Delta_{\mathrm{adv}}\|_2 \leq \delta \|\boldsymbol{x}\|_2 \qquad (\delta = 0.01)$

$$|f(\boldsymbol{x}_{\mathrm{adv}};\boldsymbol{\theta}) - f(\boldsymbol{x};\boldsymbol{\theta})| \approx |\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta})^\top \Delta_{\mathrm{adv}}| \leq \delta \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta})\|_2$$

- $f(\boldsymbol{x};\boldsymbol{\theta}) = \varphi(\boldsymbol{x})^\top \boldsymbol{\theta}$

Sensitivity to adversarial attacks

$$|f(\boldsymbol{x}_{\mathrm{adv}};\boldsymbol{\theta}) - f(\boldsymbol{x};\boldsymbol{\theta})| \approx |\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta})^{\top}\Delta_{\mathrm{adv}}| \leq \delta\|\boldsymbol{x}\|_2\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta})\|_2$$

- $f(\boldsymbol{x};\boldsymbol{\theta}) = \varphi(\boldsymbol{x})^{\top}\boldsymbol{\theta}$

Sensitivity: $\mathcal{S}_{\varphi}(\boldsymbol{x}) = \|\boldsymbol{x}\|_2\|\nabla_{\boldsymbol{x}}\varphi(\boldsymbol{x})^{\top}\boldsymbol{\theta}^*\|_2$

- $\mathcal{S}_{\varphi}(\boldsymbol{x}) = O(1) \implies$ model (at interpolation) is **robust**
- $\mathcal{S}_{\varphi}(\boldsymbol{x}) \gg 1 \implies$ model (at interpolation) is **not robust**

Sensitivity to adversarial attacks

$$|f(\boldsymbol{x}_{\mathrm{adv}};\boldsymbol{\theta}) - f(\boldsymbol{x};\boldsymbol{\theta})| \approx |\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta})^{\top}\Delta_{\mathrm{adv}}| \le \delta\|\boldsymbol{x}\|_2\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta})\|_2$$

- $f(\boldsymbol{x};\boldsymbol{\theta}) = \varphi(\boldsymbol{x})^{\top}\boldsymbol{\theta}$

Sensitivity: $\mathcal{S}_{\varphi}(\boldsymbol{x}) = \|\boldsymbol{x}\|_2\|\nabla_{\boldsymbol{x}}\varphi(\boldsymbol{x})^{\top}\boldsymbol{\theta}^*\|_2$

- $\mathcal{S}_{\varphi}(\boldsymbol{x}) = O(1) \Longrightarrow$ model (at interpolation) is **robust**
- $\mathcal{S}_{\varphi}(\boldsymbol{x}) \gg 1 \Longrightarrow$ model (at interpolation) is **not robust**

Having $\|\boldsymbol{x}\|_2$ on the RHS makes the sensitivity **scale-invariant**

Related work

- Adversarial training (instead of ERM) in linear models [Donhauser et al., 2021; Javanmard et al., 2020, 2022; Taheri et al., 2020]
- [Bubeck & Sellke, 2021; Bubeck et al., 2021] consider Lipschitz constant
- [Dohmatob & Bietti, 2022; Dohmatob, 2022] consider $\mathbb{E}_{\boldsymbol{x}} \mathcal{S}_{\varphi}^{2}(\boldsymbol{x})$:
  - The former in the infinite-data regime ($n \to \infty$)
  - The latter in the infinite-width regime ($p \to \infty$) or proportional regime ($n = \Theta(p) = \Theta(d)$)
- [Zhu et al., 2022] consider average robustness $\mathbb{E}_{\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{\theta}} \nabla_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta})^{\mathsf{T}}(\boldsymbol{x}-\hat{\boldsymbol{x}})$

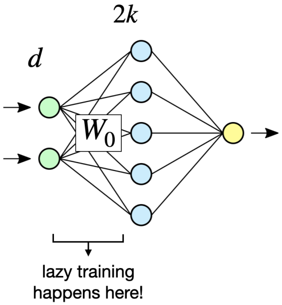
NTK features

$$f_{\rm NN}(\boldsymbol{x};\boldsymbol{W}) = \sum_{i=1}^{k} \phi(\boldsymbol{w}_i^\top \boldsymbol{x}) - \sum_{i=k+1}^{2k} \phi(\boldsymbol{w}_i^\top \boldsymbol{x})$$

NTK features

$$f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{W}) = \sum_{i=1}^{k} \phi(\boldsymbol{w}_i^\top \boldsymbol{x}) - \sum_{i=k+1}^{2k} \phi(\boldsymbol{w}_i^\top \boldsymbol{x})$$

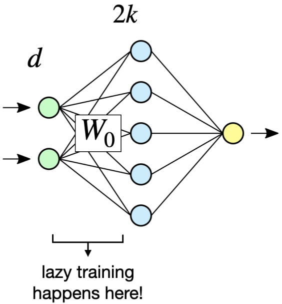
$$f_{\mathrm{NTK}}(\boldsymbol{x}; \boldsymbol{\theta}) = \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \boldsymbol{\theta}, \qquad \varphi_{\mathrm{NTK}}(\boldsymbol{x}) = \nabla_{\boldsymbol{W}} f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{W})\big|_{\boldsymbol{W}=\boldsymbol{W}_0}$$



NTK features

$$f_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{W})=\sum_{i=1}^{k}\phi(\boldsymbol{w}_i^\top\boldsymbol{x})-\sum_{i=k+1}^{2k}\phi(\boldsymbol{w}_i^\top\boldsymbol{x})$$

$$f_{\mathrm{NTK}}(\boldsymbol{x};\boldsymbol{\theta})=\varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top\boldsymbol{\theta},\qquad \varphi_{\mathrm{NTK}}(\boldsymbol{x})=\boldsymbol{x}\otimes\phi'(\boldsymbol{W}_0\boldsymbol{x})$$



NTK features are robust

Theorem [Bombari, Kiyani, and M., 2023]

Let $\varphi_{\mathrm{NTK}}(\boldsymbol{x}) = \boldsymbol{x} \otimes \phi'(\boldsymbol{W}_0 \boldsymbol{x})$. Assume $p \gg n$, $k = O(d)$, $n = O(k)$, ϕ even and smooth. Then, with high probability,

$$\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) = \tilde{O}\left(\sqrt{\frac{nd}{p}}\right).$$

NTK features are robust

Theorem [Bombari, Kiyani, and M., 2023]

Let $\varphi_{\mathrm{NTK}}(\boldsymbol{x}) = \boldsymbol{x} \otimes \phi'(\boldsymbol{W}_0 \boldsymbol{x})$. Assume $p \gg n$, $k = O(d)$, $n = O(k)$, ϕ even and smooth. Then, with high probability,

$$\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) = \tilde{O}\left(\sqrt{\frac{nd}{p}}\right).$$

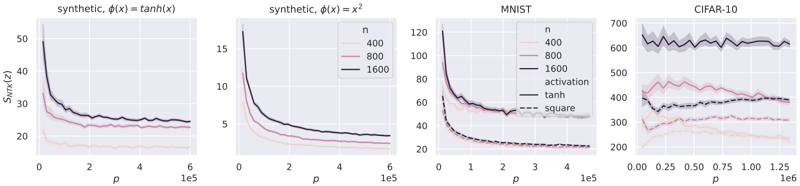
Saturates lower bound!

NTK features are robust

Theorem [Bombari, Kiyani, and M., 2023]

Let $\varphi_{\mathrm{NTK}}(\boldsymbol{x}) = \boldsymbol{x} \otimes \phi'(\boldsymbol{W}_0 \boldsymbol{x})$. Assume $p \gg n$, $k = O(d)$, $n = O(k)$, ϕ even and smooth. Then, with high probability,

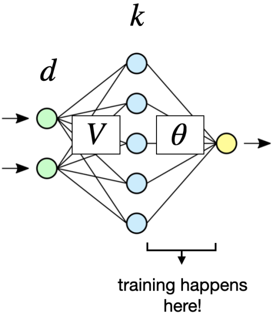
$$\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) = \tilde{O}\left(\sqrt{\frac{nd}{p}}\right).$$



- Even activations more robust than odd ones.

Random features

$$f_{\mathrm{RF}}(\boldsymbol{x};\boldsymbol{\theta}) = \varphi_{\mathrm{RF}}(\boldsymbol{x})^\top \boldsymbol{\theta}, \qquad \varphi_{\mathrm{RF}}(\boldsymbol{x}) = \phi(\boldsymbol{V}\boldsymbol{x})$$



Random features are not robust

Theorem [Bombari, Kiyani, and M., 2023]

Let $\varphi_{\mathrm{RF}}(\boldsymbol{x}) = \phi(\boldsymbol{V}\boldsymbol{x})$. Assume $p \gg n$, $p \gg d$, $d \gg n^{2/3}$, ϕ smooth and $\mathbb{E}_{\rho \sim \mathcal{N}(0,1)} \phi'(\rho) \neq 0$. Then, with high probability,

$$\mathcal{S}_{\mathrm{RF}}(\boldsymbol{x}) = \Omega\left(n^{1/6}\right) \gg 1.$$

Random features are not robust

Theorem [Bombari, Kiyani, and M., 2023]

Let $\varphi_{\mathrm{RF}}(\boldsymbol{x}) = \phi(\boldsymbol{V}\boldsymbol{x})$. Assume $p \gg n$, $p \gg d$, $d \gg n^{2/3}$, ϕ smooth and $\mathbb{E}_{\rho \sim \mathcal{N}(0,1)} \phi'(\rho) \neq 0$. Then, with high probability,

$$\mathcal{S}_{\mathrm{RF}}(\boldsymbol{x}) = \Omega\left(n^{1/6}\right) \gg 1.$$

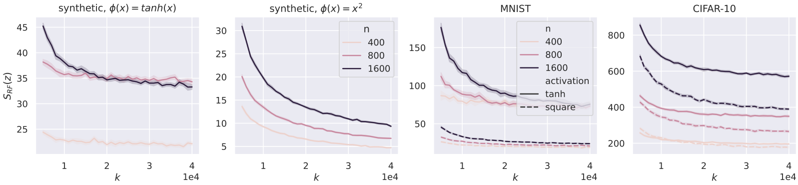
Never robust, regardless of over-parameterization!

Random features are not robust

Theorem [Bombari, Kiyani, and M., 2023]

Let $\varphi_{\mathrm{RF}}(\boldsymbol{x}) = \phi(\boldsymbol{V}\boldsymbol{x})$. Assume $p \gg n$, $p \gg d$, $d \gg n^{2/3}$, ϕ smooth and $\mathbb{E}_{\rho\sim\mathcal{N}(0,1)}\phi'(\rho) \neq 0$. Then, with high probability,

$$\mathcal{S}_{\mathrm{RF}}(\boldsymbol{x}) = \Omega\left(n^{1/6}\right) \gg 1.$$



- Having $\mathbb{E}_{\rho\sim\mathcal{N}(0,1)}\phi'(\rho) = 0$ improves robustness.

Proof ideas for NTK

$$\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) = \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \boldsymbol{\theta}^*_{\mathrm{NTK}}\|_2$$

Proof ideas for NTK

$$
\begin{aligned}
\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) &= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{\theta}^*_{\mathrm{NTK}}\|_2 \\
&= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^{\mathsf{T}} \Phi_{\mathrm{NTK}}^{\mathsf{T}} (\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^{\mathsf{T}})^{-1} \boldsymbol{y}\|_2
\end{aligned}
$$

- $\Phi_{\mathrm{NTK}} = [\varphi_{\mathrm{NTK}}(\boldsymbol{x}_1), \ldots, \varphi_{\mathrm{NTK}}(\boldsymbol{x}_n)]^{\mathsf{T}}$

Proof ideas for NTK

$$\begin{aligned}
\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) &= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \boldsymbol{\theta}^*_{\mathrm{NTK}}\|_2 \\
&= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \Phi_{\mathrm{NTK}}^\top (\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top)^{-1} \boldsymbol{y}\|_2 \\
&\le \|\boldsymbol{x}\|_2 \|\mathcal{I}_{\mathrm{NTK}}\|_{\mathrm{op}}\, \lambda_{\min}^{-1}\left(\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top\right) \|\boldsymbol{y}\|_2
\end{aligned}$$

- $\Phi_{\mathrm{NTK}} = [\varphi_{\mathrm{NTK}}(\boldsymbol{x}_1), \ldots, \varphi_{\mathrm{NTK}}(\boldsymbol{x}_n)]^\top$
- $\mathcal{I}_{\mathrm{NTK}} = \nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \Phi_{\mathrm{NTK}}^\top$ **interaction matrix**

Proof ideas for NTK

$$
\begin{aligned}
\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) &= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \boldsymbol{\theta}^*_{\mathrm{NTK}}\|_2 \\
&= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \Phi_{\mathrm{NTK}}^\top (\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top)^{-1} \boldsymbol{y}\|_2 \\
&\leq \|\boldsymbol{x}\|_2 \|\mathcal{I}_{\mathrm{NTK}}\|_{\mathrm{op}} \, \lambda_{\min}^{-1}\left(\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top\right) \|\boldsymbol{y}\|_2
\end{aligned}
$$

- $\Phi_{\mathrm{NTK}} = [\varphi_{\mathrm{NTK}}(\boldsymbol{x}_1), \ldots, \varphi_{\mathrm{NTK}}(\boldsymbol{x}_n)]^\top$
- $\mathcal{I}_{\mathrm{NTK}} = \nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \Phi_{\mathrm{NTK}}^\top$ **interaction matrix**

$\|\mathcal{I}_{\mathrm{NTK}}\|_{\mathrm{op}}$ computed explicitly for even ϕ

$\lambda_{\min}\left(\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top\right)$ is the smallest NTK eigenvalue

Proof ideas for NTK

$$\begin{aligned}\mathcal{S}_{\mathrm{NTK}}(\boldsymbol{x}) &= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \boldsymbol{\theta}^*_{\mathrm{NTK}}\|_2 \\ &= \|\boldsymbol{x}\|_2 \|\nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \Phi_{\mathrm{NTK}}^\top (\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top)^{-1} \boldsymbol{y}\|_2 \\ &\le \|\boldsymbol{x}\|_2 \|\mathcal{I}_{\mathrm{NTK}}\|_{\mathrm{op}}\, \lambda_{\min}^{-1}\left(\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top\right) \|\boldsymbol{y}\|_2 \\ &= \tilde{O}\left(\sqrt{\frac{nd}{p}}\right)\end{aligned}$$

- $\Phi_{\mathrm{NTK}} = [\varphi_{\mathrm{NTK}}(\boldsymbol{x}_1), \ldots, \varphi_{\mathrm{NTK}}(\boldsymbol{x}_n)]^\top$
- $\mathcal{I}_{\mathrm{NTK}} = \nabla_{\boldsymbol{x}} \varphi_{\mathrm{NTK}}(\boldsymbol{x})^\top \Phi_{\mathrm{NTK}}^\top$ **interaction matrix**

$\|\mathcal{I}_{\mathrm{NTK}}\|_{\mathrm{op}}$ computed explicitly for even ϕ

$\lambda_{\min}\left(\Phi_{\mathrm{NTK}} \Phi_{\mathrm{NTK}}^\top\right)$ is the smallest NTK eigenvalue

Proof ideas (RF)

A bit more involved...

$$\mathcal{S}_{\mathrm{RF}}(\boldsymbol{x}) = \Omega\left(\|\boldsymbol{x}\|_2 \|\tilde{\mathcal{I}}_{\mathrm{RF}}\|_F \, \lambda_{\max}^{-1}\left(\tilde{\Phi}_{\mathrm{RF}}\tilde{\Phi}_{\mathrm{RF}}^{\top}\right)\right)$$

- Remove low-rank components by centering: $\tilde{\Phi}_{\mathrm{RF}} = \Phi_{\mathrm{RF}} - \mathbb{E}_{\boldsymbol{X}}\left[\Phi_{\mathrm{RF}}\right]$
- Interaction matrix captures the effect of the activation

$$\|\tilde{\mathcal{I}}_{\mathrm{RF}}\|_F = \frac{k\sqrt{n}}{\sqrt{d}}\left(\mathbb{E}^2_{\rho\sim\mathcal{N}(0,1)}\phi'(\rho) + o(1)\right)$$

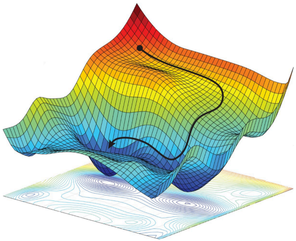
Proof ideas (RF)

A bit more involved...

$$\mathcal{S}_{\mathrm{RF}}(\boldsymbol{x}) = \Omega\left(\|\boldsymbol{x}\|_2 \|\tilde{\mathcal{I}}_{\mathrm{RF}}\|_F \, \lambda_{\max}^{-1}\left(\tilde{\Phi}_{\mathrm{RF}} \tilde{\Phi}_{\mathrm{RF}}^\top\right)\right) = \Omega(n^{1/6})$$

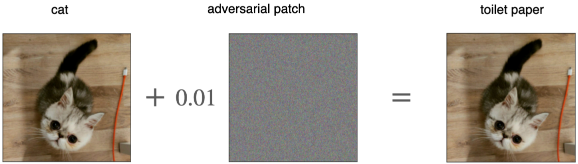
- Remove low-rank components by centering: $\tilde{\Phi}_{\mathrm{RF}} = \Phi_{\mathrm{RF}} - \mathbb{E}_{\boldsymbol{X}}\left[\Phi_{\mathrm{RF}}\right]$
- Interaction matrix captures the effect of the activation

$$\|\tilde{\mathcal{I}}_{\mathrm{RF}}\|_F = \frac{k\sqrt{n}}{\sqrt{d}}\left(\mathbb{E}^2_{\rho\sim\mathcal{N}(0,1)}\phi'(\rho) + o(1)\right)$$



optimization

attention



robustness

Simone Bombari (ISTA)

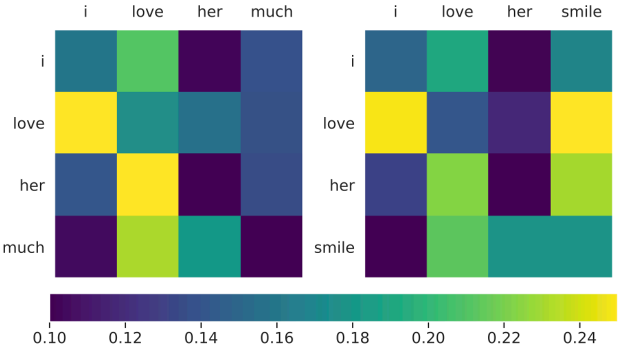
S. Bombari and M. Mondelli, “Towards Understanding the Word Sensitivity of Attention Layers: A Study via Random Features”, *arXiv preprint*, 2024.

Changing a word changes the meaning

Prompt	Output
Reply with "Yes" if the review I will provide you is positive, and "No" otherwise. Review: Sorry, gave it a 1, which is the rating I give to movies on which I walk out or fall asleep.	No
Reply with "Yes" if the review I will provide you is negative, and "No" otherwise. Review: Sorry, gave it a 1, which is the rating I give to movies on which I walk out or fall asleep.	Yes

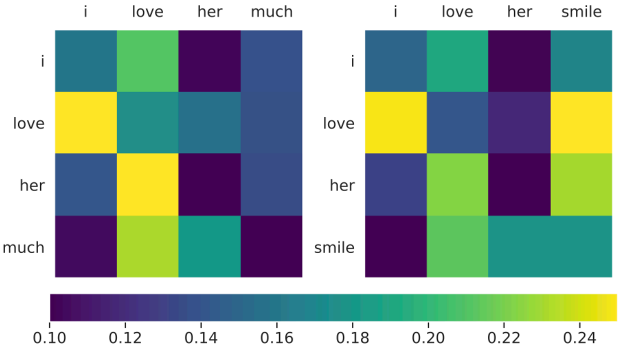
- Different output of the Llama2-7b-chat model

Changing a word changes the scores



- Different attention score pattern of BERT-Base model

Changing a word changes the scores



- Different attention score pattern of BERT-Base model

Transformers capture the effect of changing a single word in a sentence.

Insights from random features

Data: $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times d}$

- Tokens $\{\boldsymbol{x}_i\}_{i=1}^N$
- $N =$ context length, $d =$ embedding dimension

Insights from random features

Data: $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times d}$

- Tokens $\{\boldsymbol{x}_i\}_{i=1}^N$
- N = context length, d = embedding dimension

Random features: $\varphi_{\mathrm{RF}} : \mathbb{R}^{N \times d} \to \mathbb{R}^k$

$$\varphi_{\mathrm{RF}}(\boldsymbol{X}) = \phi(\boldsymbol{V}\,\mathrm{flat}(\boldsymbol{X}))$$

Insights from random features

Data: $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times d}$

- Tokens $\{\boldsymbol{x}_i\}_{i=1}^N$
- N = context length, d = embedding dimension

Random features: $\varphi_{\mathrm{RF}} : \mathbb{R}^{N \times d} \to \mathbb{R}^k$

$$\varphi_{\mathrm{RF}}(\boldsymbol{X}) = \phi(\boldsymbol{V}\,\mathrm{flat}(\boldsymbol{X}))$$

Random attention features: $\varphi_{\mathrm{QKV}} : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d'}$

$$\varphi_{\mathrm{QKV}}(\boldsymbol{X}) = \mathrm{softmax}\left(\frac{\boldsymbol{X}\boldsymbol{W}_Q^\top \boldsymbol{W}_K \boldsymbol{X}^\top}{\sqrt{d'}}\right)\boldsymbol{X}\boldsymbol{W}_V^\top$$

- $\mathrm{softmax}(\boldsymbol{s})_i = e^{s_i} / \sum_j e^{s_j}$
- $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V \in \mathbb{R}^{d' \times d}$ queries, keys and values matrices

Insights from random features

Data: $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\mathsf{T} \in \mathbb{R}^{N \times d}$

- Tokens $\{\boldsymbol{x}_i\}_{i=1}^N$
- $N =$ context length, $d =$ embedding dimension

Random features: $\varphi_{\mathrm{RF}} : \mathbb{R}^{N \times d} \to \mathbb{R}^k$

$$\varphi_{\mathrm{RF}}(\boldsymbol{X}) = \phi(\boldsymbol{V}\,\mathrm{flat}(\boldsymbol{X}))$$

Random attention features: $\varphi_{\mathrm{RAF}} : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d}$

$$\varphi_{\mathrm{RAF}}(\boldsymbol{X}) = \mathrm{softmax}\left(\frac{\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^\mathsf{T}}{\sqrt{d}}\right)\boldsymbol{X}$$

- $\mathrm{softmax}(\boldsymbol{s})_i = e^{s_i} / \sum_j e^{s_j}$

Sample complexity comparison between RF and RAF in (Fu et al., 2023)

Sensitivity to changing a word

Word sensitivity: $\mathcal{WS}_\varphi(\boldsymbol{X}) = \sup_{j\in[N],\, \|\boldsymbol{\Delta}\|_2 \le \sqrt{d}} \dfrac{\left\|\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta})) - \varphi(\boldsymbol{X})\right\|_2}{\|\varphi(\boldsymbol{X})\|_2}$

- $\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta})) = \boldsymbol{X} + \boldsymbol{e}_j\boldsymbol{\Delta}^\top$ (only j-th token changed)
- $\|\boldsymbol{\Delta}\|_2 \le \sqrt{d}$ (perturbation size bounded by token size)

Low WS of random features

Theorem [Bombari and M., 2024]

Assume ϕ Lipschitz and $k = \Omega(Nd)$. Then, with high probability over $\boldsymbol{V}$,

$$\mathcal{WS}_{\mathrm{RF}}(\boldsymbol{X}) = O\left(\frac{1}{\sqrt{N}}\right)$$

Word sensitivity vanishes as context length N grows

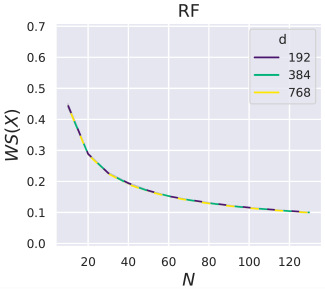
Low WS of random features

Theorem [Bombari and M., 2024]

Assume ϕ Lipschitz and $k = \Omega(Nd)$. Then, with high probability over $\boldsymbol{V}$,

$$\mathcal{WS}_{\text{RF}}(\boldsymbol{X}) = O\left(\frac{1}{\sqrt{N}}\right)$$

Word sensitivity vanishes as context length N grows



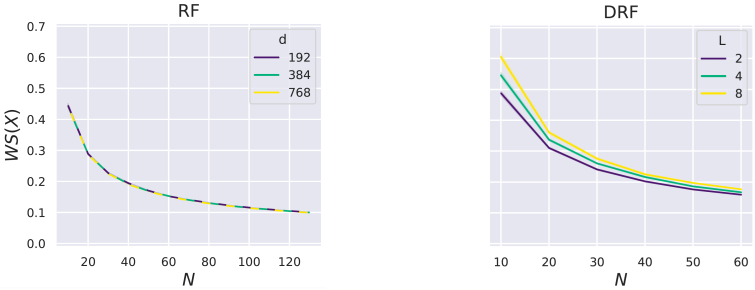
Low WS of random features

Theorem [Bombari and M., 2024]

Assume ϕ Lipschitz and $k = \Theta(Nd)$. Then, with high probability over $\boldsymbol{V}_1, \dots, \boldsymbol{V}_L$,

$$\mathcal{WS}_{\text{DRF}}(\boldsymbol{X}) = O\left(\frac{e^{CL}}{\sqrt{N}}\right)$$

Word sensitivity vanishes as context length N grows



High WS of random attention features

Theorem [Bombari and M., 2024]

Assume $d = \tilde{\Omega}(N)$. Then, with high probability over $\boldsymbol{W}$,

$$\mathcal{WS}_{\mathsf{RAF}}(\boldsymbol{X}) = \Omega(1)$$

High word sensitivity regardless of the context length N

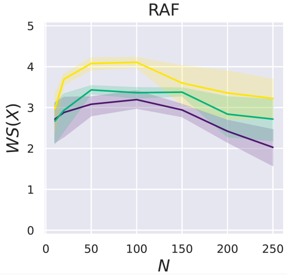
High WS of random attention features

Theorem [Bombari and M., 2024]

Assume $d = \tilde{\Omega}(N)$. Then, with high probability over $\boldsymbol{W}$,

$$\mathcal{WS}_{\mathrm{RAF}}(\boldsymbol{X}) = \Omega(1)$$

High word sensitivity regardless of the context length N



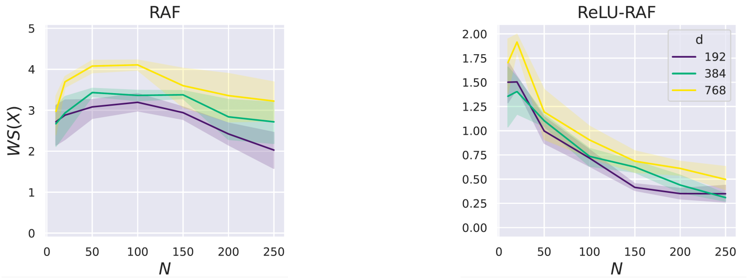
High WS of random attention features

Theorem [Bombari and M., 2024]

Assume $d = \tilde{\Omega}(N)$. Then, with high probability over $\boldsymbol{W}$,

$$\mathcal{WS}_{\mathsf{RAF}}(\boldsymbol{X}) = \Omega(1)$$

High word sensitivity regardless of the context length N



Word sensitivity decreases when replacing softmax with ReLU

Proof ideas for RAF

1. Find a direction $\boldsymbol{\delta}^*$ aligned with many words $\boldsymbol{x}_i$'s.

- Constant fraction of the entries of $\boldsymbol{X}\delta^*$ is $\Omega(d/\sqrt{N})$ in modulus

Proof ideas for RAF

1. Find a direction $\boldsymbol{\delta}^*$ aligned with many words $\boldsymbol{x}_i$'s.

- Constant fraction of the entries of $\boldsymbol{X}\delta^*$ is $\Omega(d/\sqrt{N})$ in modulus

2. Exhibit two different directions $\boldsymbol{\Delta}_1^*$ and $\boldsymbol{\Delta}_2^*$ both aligned with many words in the feature space $\{\boldsymbol{W}^\top \boldsymbol{x}_i\}_{i=1}^N$.

- $\|\boldsymbol{\Delta}_1^* - \boldsymbol{\Delta}_2^*\|_2 = \Omega(\sqrt{d})$
- Constant fraction of the entries of $\boldsymbol{X}\boldsymbol{W}\boldsymbol{\Delta}_k^*/\sqrt{d}$ is $\Omega\left(\log^2 d\right)$

Proof ideas for RAF

1. Find a direction $\boldsymbol{\delta}^*$ aligned with many words $\boldsymbol{x}_i$'s.

- Constant fraction of the entries of $\boldsymbol{X}\delta^*$ is $\Omega(d/\sqrt{N})$ in modulus

2. Exhibit two different directions $\boldsymbol{\Delta}_1^*$ and $\boldsymbol{\Delta}_2^*$ both aligned with many words in the feature space $\{\boldsymbol{W}^\top \boldsymbol{x}_i\}_{i=1}^N$.

- $\|\boldsymbol{\Delta}_1^* - \boldsymbol{\Delta}_2^*\|_2 = \Omega(\sqrt{d})$
- Constant fraction of the entries of $\boldsymbol{X}\boldsymbol{W}\boldsymbol{\Delta}_k^*/\sqrt{d}$ is $\Omega\left(\log^2 d\right)$

3. Attention concentrates towards the perturbed word.

- Constant fraction of rows of $\mathrm{softmax}(\boldsymbol{X}^j(\boldsymbol{\Delta}_k^*)\boldsymbol{W}(\boldsymbol{X}^j(\boldsymbol{\Delta}_k^*))^\top/\sqrt{d}) \approx \boldsymbol{e}_j$

Key role of softmax

Proof ideas for RAF

1. Find a direction $\boldsymbol{\delta}^*$ aligned with many words $\boldsymbol{x}_i$'s.

- Constant fraction of the entries of $\boldsymbol{X}\boldsymbol{\delta}^*$ is $\Omega(d/\sqrt{N})$ in modulus

2. Exhibit two different directions $\boldsymbol{\Delta}_1^*$ and $\boldsymbol{\Delta}_2^*$ both aligned with many words in the feature space $\{\boldsymbol{W}^\top \boldsymbol{x}_i\}_{i=1}^N$.

- $\|\boldsymbol{\Delta}_1^* - \boldsymbol{\Delta}_2^*\|_2 = \Omega(\sqrt{d})$
- Constant fraction of the entries of $\boldsymbol{X}\boldsymbol{W}\boldsymbol{\Delta}_k^*/\sqrt{d}$ is $\Omega\left(\log^2 d\right)$

3. Attention concentrates towards the perturbed word.

- Constant fraction of rows of $\text{softmax}(\boldsymbol{X}^j(\boldsymbol{\Delta}_k^*)\boldsymbol{W}(\boldsymbol{X}^j(\boldsymbol{\Delta}_k^*))^\top/\sqrt{d}) \approx \boldsymbol{e}_j$

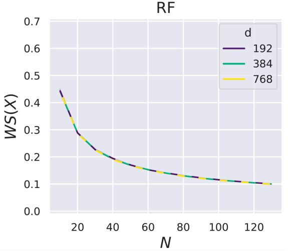
Key role of softmax

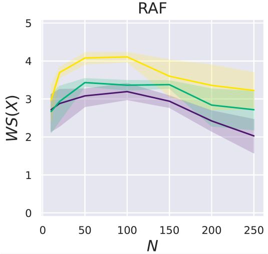
4. Conclude with at least one perturbation between $\boldsymbol{\Delta}_1^*$ and $\boldsymbol{\Delta}_2^*$.

- $\left\|\varphi_{\text{RAF}}(\boldsymbol{X}) - \varphi_{\text{RAF}}(\boldsymbol{X}^j(\boldsymbol{\Delta}_k^*))\right\|_F = o(\sqrt{dN})$ for $k = 1, 2 \Rightarrow$ $\|\boldsymbol{\Delta}_1^* - \boldsymbol{\Delta}_2^*\|_2 = o(\sqrt{d})$, which is a contradiction

Generalization on context modification

1. Random features have low word sensitivity:
   $\mathcal{WS}_{\mathrm{RF}}(\boldsymbol{X}) = O(1/\sqrt{N})$.
2. Random attention features have high word sensitivity:
   $\mathcal{WS}_{\mathrm{RAF}}(\boldsymbol{X}) = \Omega(1)$.





Idea: random features cannot learn to distinguish $\boldsymbol{X}$ and $\boldsymbol{X}^j(\boldsymbol{\Delta})$, while random attention features can!

Generalized linear regression

Data: $(\boldsymbol{X}_1, y_1), \dots, (\boldsymbol{X}_n, y_n) \in \mathbb{R}^{N \times d} \times \{-1, 1\}$

Goal: Minimize empirical risk $L(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \varphi(\boldsymbol{X}_i)^{\mathsf{T}} \boldsymbol{\theta} \right)^2$

- $\varphi : \mathbb{R}^{N \times d} \to \mathbb{R}^p$ feature map

Gradient descent solution: $\boldsymbol{\theta}^* = \boldsymbol{\theta}_0 + \Phi^{\mathsf{T}} (\Phi \Phi^{\mathsf{T}})^{-1} (\boldsymbol{y} - \Phi \boldsymbol{\theta}_0)$

- $\boldsymbol{\theta}_0$ initialization
- $\Phi = [\varphi(\boldsymbol{X}_1), \dots, \varphi(\boldsymbol{X}_n)]^{\mathsf{T}}$ feature matrix
- $\boldsymbol{y} = [y_1, \dots, y_n]^{\mathsf{T}}$ label vector

More training and generalization

Does further training on $(\boldsymbol{X}, \tilde{y})$ allow to generalize on $(\boldsymbol{X}^j(\boldsymbol{\Delta}), -\tilde{y})$?

	Prompt		Output
$\boldsymbol{X}$	Reply with "Yes" if the review I will provide you is positive, and "No" otherwise. Review: Sorry, gave it a 1, which is the rating I give to movies on which I walk out or fall asleep.	$\tilde{y}$	No
$\boldsymbol{X}^j(\boldsymbol{\Delta})$	Reply with "Yes" if the review I will provide you is negative, and "No" otherwise. Review: Sorry, gave it a 1, which is the rating I give to movies on which I walk out or fall asleep.	$-\tilde{y}$	Yes

More training and generalization

Does further training on $(\boldsymbol{X}, \tilde{y})$ allow to generalize on $(\boldsymbol{X}^j(\boldsymbol{\Delta}), -\tilde{y})$?

Fine-tuning. Initialize with $\boldsymbol{\theta}^*$ and train only the extra sample:

$$\boldsymbol{\theta}_f^* = \boldsymbol{\theta}^* + \frac{\varphi(\boldsymbol{X})}{\|\varphi(\boldsymbol{X})\|_2^2}\left(\tilde{y} - \varphi(\boldsymbol{X})^\top \boldsymbol{\theta}^*\right).$$

More training and generalization

Does further training on $(\boldsymbol{X}, \tilde{y})$ allow to generalize on $(\boldsymbol{X}^j(\boldsymbol{\Delta}), -\tilde{y})$?

Fine-tuning. Initialize with $\boldsymbol{\theta}^*$ and train only the extra sample:

$$\boldsymbol{\theta}_f^* = \boldsymbol{\theta}^* + \frac{\varphi(\boldsymbol{X})}{\|\varphi(\boldsymbol{X})\|_2^2}\left(\tilde{y} - \varphi(\boldsymbol{X})^\top \boldsymbol{\theta}^*\right).$$

Re-training. Add $(\boldsymbol{X}, \tilde{y})$ to training set and train from scratch:

$$\boldsymbol{\theta}_r^* = \boldsymbol{\theta}_0 + \Phi_r^\mathsf{T}(\Phi_r \Phi_r^\mathsf{T})^{-1}(\boldsymbol{y}_r - \Phi_r \boldsymbol{\theta}_0).$$

- $\boldsymbol{\theta}_0$ initialization
- $\Phi_r = [\varphi(\boldsymbol{X}_1), \ldots, \varphi(\boldsymbol{X}_n), \varphi(\boldsymbol{X})]^\top$ feature matrix
- $\boldsymbol{y}_r = [y_1, \ldots, y_n, \tilde{y}]^\mathsf{T}$ label vector

Random features do not generalize...

Theorem [Bombari and M., 2024]

Let $\left|\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta}))^\top \boldsymbol{\theta}^* - \varphi(\boldsymbol{X})^\top \boldsymbol{\theta}^*\right| \leq \gamma$ for $\gamma \in [0, 2)$.

Random features do not generalize...

Theorem [Bombari and M., 2024]

Let $\left|\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta}))^\top \boldsymbol{\theta}^* - \varphi(\boldsymbol{X})^\top \boldsymbol{\theta}^*\right| \leq \gamma$ for $\gamma \in [0, 2)$. Then, under some technical assumptions, with high probability,

$$\mathrm{Err}_{\mathrm{RF}}(\boldsymbol{X}^j(\boldsymbol{\Delta}), \boldsymbol{\theta}^*_{f/r}) := \left(\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta}))^\top \boldsymbol{\theta}^*_{f/r} + \tilde{y}\right)^2 > (2-\gamma)^2 - O\left(\frac{1}{\sqrt{N}}\right)$$

Random features do not generalize. . .

Theorem [Bombari and M., 2024]

Let $\left|\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta}))^\top \boldsymbol{\theta}^* - \varphi(\boldsymbol{X})^\top \boldsymbol{\theta}^*\right| \leq \gamma$ for $\gamma \in [0, 2)$. Then, under some technical assumptions, with high probability,

$$\mathrm{Err}_{\mathrm{RF}}(\boldsymbol{X}^j(\boldsymbol{\Delta}), \boldsymbol{\theta}^*_{f/r}) := \left(\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta}))^\top \boldsymbol{\theta}^*_{f/r} + \tilde{y}\right)^2 > (2-\gamma)^2 - O\left(\frac{1}{\sqrt{N}}\right)$$

Unless the correct label is already known ($\gamma = 2$),
fine-tuning/re-training does not help much.

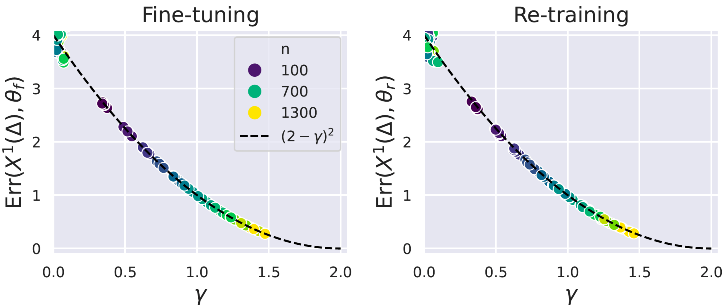
Idea: after perturbing the j-th token, the model cannot move more than its WS, which is $O(1/\sqrt{N})$.

Random features do not generalize...

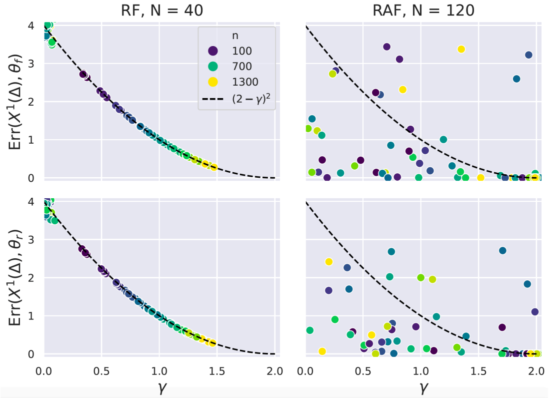
Theorem [Bombari and M., 2024]

Let $\left|\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta}))^\top \boldsymbol{\theta}^* - \varphi(\boldsymbol{X})^\top \boldsymbol{\theta}^*\right| \leq \gamma$ for $\gamma \in [0, 2)$. Then, under some technical assumptions, with high probability,

$$\mathrm{Err}_{\mathrm{RF}}(\boldsymbol{X}^j(\boldsymbol{\Delta}), \boldsymbol{\theta}^*_{f/r}) := \left(\varphi(\boldsymbol{X}^j(\boldsymbol{\Delta}))^\top \boldsymbol{\theta}^*_{f/r} + \tilde{y}\right)^2 > (2-\gamma)^2 - O\left(\frac{1}{\sqrt{N}}\right)$$

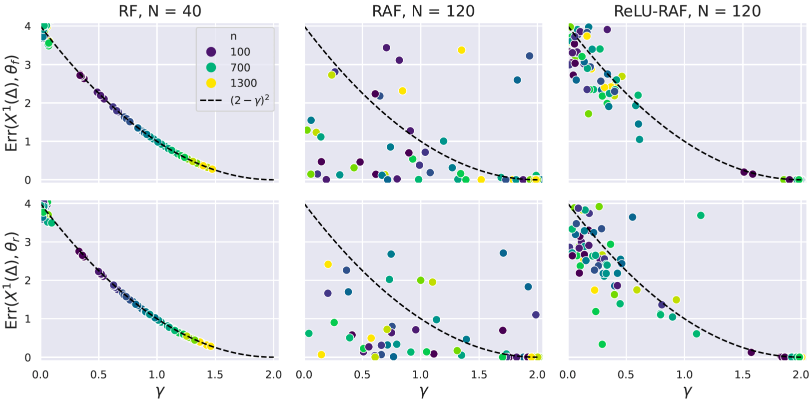


. . . but random attention features do generalize



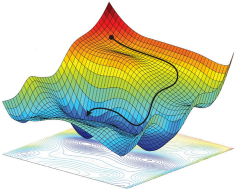
- Random attention features generalize even when $\boldsymbol{X}$ and $\boldsymbol{X}^j(\boldsymbol{\Delta})$ were indistiguishable before the extra training ($\gamma \approx 0$).

. . . but random attention features do generalize



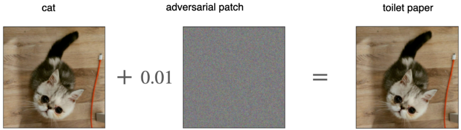
- Random attention features generalize even when $\boldsymbol{X}$ and $\boldsymbol{X}^j(\boldsymbol{\Delta})$ were indistiguishable before the extra training ($\gamma \approx 0$).
- Replacing softmax with ReLU increases error.

Take home



optimization

attention



robustness

Quantitative understanding via random and NTK features