Revisitando o princípio da parcimônia na identificação de sistemas e aprendizado de máquina

Antônio Horta Ribeiro
Universidade de Uppsala, Suécia

Departamento de Engenharia Mecânica, PUC-Rio

24 de Maio de 2023

Presentation outline

I - Background

Deep networks for system identification: a Survey.
Gianluigi Pillonetto, Aleksandr Aravkin, Daniel Gedon, Lennart Ljung, **Antonio H. Ribeiro**, Thomas B. Schön.
Under review Automatica (2023)

The unreasonable effectiveness of overparameterized machine learning models
Antônio H. Ribeiro, Dave Zachariah, Per Mattsson.
Seminar PhD course, Uppsala University (Fall 2021)

II - Dynamical systems

Beyond Occam's Razor in System Identification: Double-Descent when Modeling Dynamics
Antônio H. Ribeiro, Johannes N. Hendriks, Adrian G. Wills, Thomas B. Schön.
IFAC Symposium on System Identification (SYSID), 2021.
Honorable mention: Young author award

III - Adversarial Examples

Regularization properties of adversarially-trained linear regression
Antônio H. Ribeiro, Dave Zachariah, Francis Bach, Thomas B. Schön.
Submited NeurIPS (2023)

Overparameterized Linear Regression under Adversarial Attack.
Antônio H. Ribeiro, Thomas B. Schön.
IEEE Transactions on Signal Processing (2023)

I - Background

Setup

- Train Dataset:

$$(x_i, y_i),\ \ i = 1, \cdots, \#train.$$

Setup

- Train Dataset:

$$(x_i, y_i),\ \ i = 1, \cdots, \#train.$$

- Model:

$$f_\beta : x \mapsto \widehat{y}$$

Setup

- Train Dataset:
$$(x_i, y_i), \;\; i = 1, \cdots, \#train.$$
- Model:
$$f_\beta : x \mapsto \widehat{y}$$
- Parameter estimation method:
$$\min_{\beta} \sum_{i=1}^{\#train} (y_i - f_\beta(x_i))^2$$

Setup

- Train Dataset:

$$(x_i, y_i), \; i = 1, \cdots, \#train.$$

- Model:

$$f_\beta : x \mapsto \widehat{y}$$

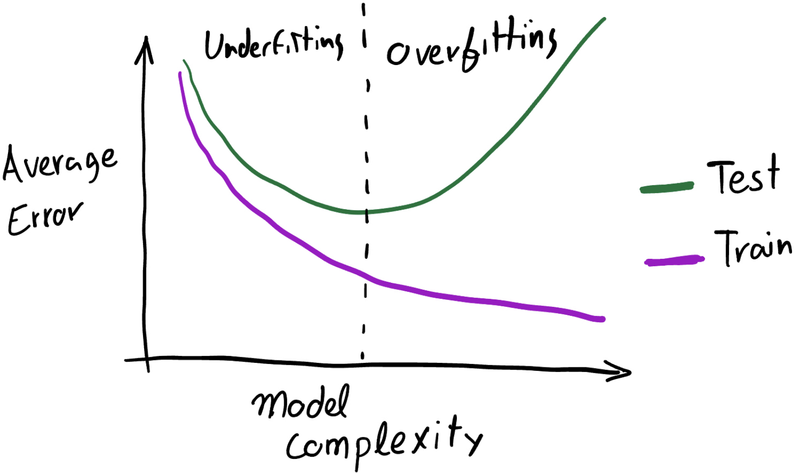
- Parameter estimation method:

$$\min_\beta \sum_{i=1}^{\#train} (y_i - f_\beta(x_i))^2$$

- Test dataset.

System identification vs machine learning

Bias-variance tradeoff



The principle of parsimony

- *"Everything should be made as simple as possible, but not simpler"* (**Albert Einsten**)

The principle of parsimony

- *"Everything should be made as simple as possible, but not simpler"* (**Albert Einsten**)
- *"Plurality should not be posited without necessity"* (**William of Ockham**)

The principle of parsimony

- *"Everything should be made as simple as possible, but not simpler"* (**Albert Einsten**)
- *"Plurality should not be posited without necessity"* (**William of Ockham**)
- *Of two competing theories, the simpler explanation of an entity is to be preferred* **(Occam's razor)**

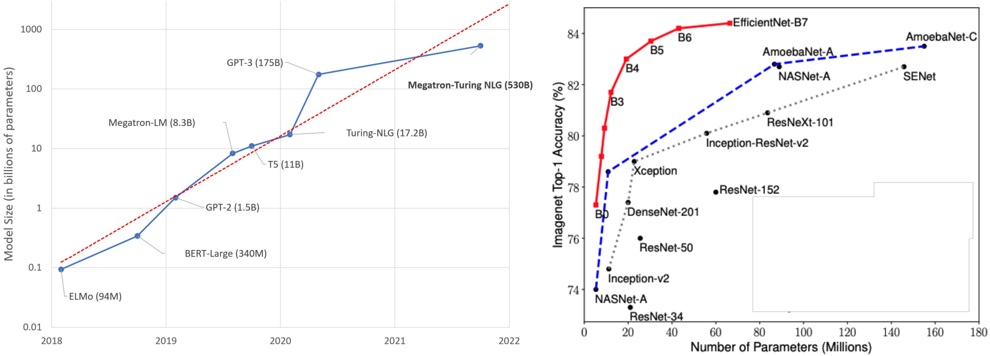
The principle of parsimony

- *"Everything should be made as simple as possible, but not simpler"* (**Albert Einsten**)
- *"Plurality should not be posited without necessity"* (**William of Ockham**)
- *Of two competing theories, the simpler explanation of an entity is to be preferred* **(Occam's razor)**
- *"It is superfluous to suppose that what can be accounted for by a few principles has been produced by many."*(**Summa Theologica, Thomas Aquinas**)

The principle of parsimony

- *"Everything should be made as simple as possible, but not simpler"* (**Albert Einsten**)
- *"Plurality should not be posited without necessity"* (**William of Ockham**)
- *Of two competing theories, the simpler explanation of an entity is to be preferred* **(Occam's razor)**
- *"It is superfluous to suppose that what can be accounted for by a few principles has been produced by many."*(**Summa Theologica, Thomas Aquinas**)
- *"To think is to forget a difference, to generalize, to abstract. In the overly replete world of Funes, there were nothing but details."* (**Funes, the Memorious, Jorge Luis Borges**)

Model size in neural networks



Language models | Image classification

Figure: Models number of parameters

J. Simon (2021) "Large Language Models: A New Moore's Law?". Online (acessed: 2021-11-09): huggingface.co/blog/large-language-models .
M. Tan and Q. V. Le (2019) "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," ICML.

Rethinking generalization

Deep neural networks can fit randomly labeled training data but still generalize well.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. ICLR, 2017

Rethinking generalization

Deep neural networks can fit randomly labeled training data but still generalize well.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. ICLR, 2017

Definition: Interpolator

We say the model f_β interpolates the training data if:

$$f_\beta(x_i) = y_i, \forall i = 1, \cdots, \#train$$

Linear-in-the-parameters models

- Model:
$$f_{\beta}(x) = \beta^{\top}\phi(x)$$
where ϕ map from input to feature space $\phi : \mathbb{R}^{\#inputs} \mapsto \mathbb{R}^{\#parameters}$.

Linear-in-the-parameters models

- Model:
$$f_\beta(x) = \beta^\top \phi(x)$$
where ϕ map from input to feature space $\phi : \mathbb{R}^{\#inputs} \mapsto \mathbb{R}^{\#parameters}$.
- Can we solve the system
$$\underbrace{\begin{bmatrix} \phi(x_1) \\ \phi(x_2) \\ \vdots \end{bmatrix}}_{X} \beta = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix}}_{y}$$

Solutions of a linear system

The system

$$X\beta = y$$

has:

- no solution if $\mathrm{rank}(X) < \#train$

Solutions of a linear system

The system

$$X\beta = y$$

has:

- no solution if $\mathrm{rank}(X) < \#\mathit{train}$
- one unique solution if $\mathrm{rank}(X) = \#\mathit{train}$

Solutions of a linear system

The system

$$X\beta = y$$

has:

- no solution if $\mathrm{rank}(X) < \#train$
- one unique solution if $\mathrm{rank}(X) = \#train$
- multiple solution if $\mathrm{rank}(X) > \#train$

Gradient descent on overparametrized

- Cost function:

$$V(\beta) = \|X\beta - y\|^2$$

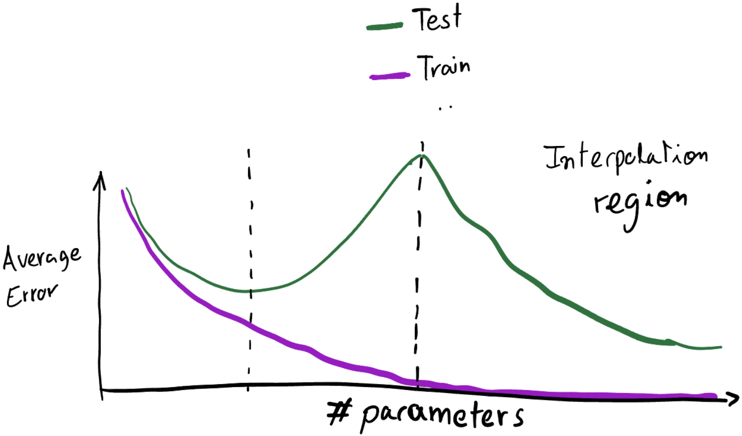
Gradient descent on overparametrized

- Cost function:
$$V(\beta) = \|X\beta - y\|^2$$
- Optimization:
$$\beta^{i+1} = \beta^{i} - \gamma \nabla V(\beta^{i})$$

Gradient descent on overparametrized

- Cost function:
$$V(\beta) = \|X\beta - y\|^2$$
- Optimization:
$$\beta^{i+1} = \beta^i - \gamma \nabla V(\beta^i)$$
- Gradient descent converges to the minimum-norm solution:
$$\min_{\theta} \|\beta\|_2 \quad \text{subject to} \quad X\beta = y.$$

Double-descent



M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," Proceedings of the National Academy of Sciences, vol. 116, no. 32, pp. 15849–15854, 2019, doi: 10.1073/pnas.1903070116.

II - Dynamical systems

Double-descent in system identification

Can we observe the phenomena in data from a dynamical system ?

Beyond Occam's Razor in System Identification: Double-Descent when Modeling Dynamics
Antônio H. Ribeiro, Johannes N. Hendriks, Adrian G. Wills, Thomas B. Schön.
IFAC Symposium on System Identification (SYSID), 2021.
Honorable mention: Young author award

Syntetic Dataset

$$y_t = \left(0.8 - 0.5e^{-y_{t-1}^2}\right) y_{t-1} - \left(0.3 + 0.9e^{-y_{t-1}^2}\right) y_{t-2}$$
$$+ u_{t-1} + 0.2u_{t-2} + 0.1u_{t-1}u_{t-2} + v_t,$$
$$v_t \sim \mathcal{N}(0, \sigma_v^2)$$

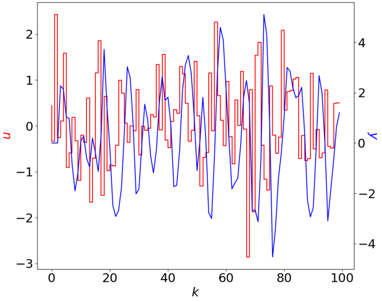


Figure: **System with process noise.**

Chen, S., Billings, S.A., and Grant, P.M. (1990). *Non-Linear System Identification Using Neural Networks.* International Journal of Control, 51(6), 1191–1214.

Model

- Input: (ARX)

$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$

Model

- Input: (ARX)

$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$

- Nonlinear feature map:

$$\phi(x_t) = \sigma(W x_t + b)$$

Model

- Input: (ARX)
$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$
- Nonlinear feature map:
$$\phi(x_t) = \sigma(W x_t + b)$$

$W \rightsquigarrow$ a matrix with dimension $\#parameters \times 4$:

Model

- Input: (ARX)
$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$
- Nonlinear feature map:
$$\phi(x_t) = \sigma(W x_t + b)$$

$W \rightsquigarrow$ a matrix with dimension $\#parameters \times 4$:

$b \rightsquigarrow$ is a vector with dimension $\#parameters$

Model

- Input: (ARX)

$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$

- Nonlinear feature map:

$$\phi(x_t) = \sigma(W x_t + b)$$

$W \rightsquigarrow$ a matrix with dimension $\#parameters \times 4$:

$b \rightsquigarrow$ is a vector with dimension $\#parameters$

$w_{i,j} \sim \mathcal{N}(0, \gamma^2), b_i \sim \mathcal{U}(0, 2\pi]$

Model

- Input: (ARX)
$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$
- Nonlinear feature map:
$$\phi(x_t) = \sigma(W x_t + b)$$

$W \rightsquigarrow$ a matrix with dimension $\#parameters \times 4$:

$b \rightsquigarrow$ is a vector with dimension $\#parameters$

$w_{i,j} \sim \mathcal{N}(0, \gamma^2)$, $b_i \sim \mathcal{U}(0, 2\pi]$

$\sigma \rightsquigarrow$ activation function.

Model

- Input: (ARX)

$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$

- Nonlinear feature map:

$$\phi(x_t) = \sigma(W x_t + b)$$

  $W \rightsquigarrow$ a matrix with dimension $\#parameters \times 4$:

  $b \rightsquigarrow$ is a vector with dimension $\#parameters$

  $w_{i,j} \sim \mathcal{N}(0, \gamma^2), b_i \sim \mathcal{U}(0, 2\pi]$

  $\sigma \rightsquigarrow$ activation function.

- Neural network with frozen first layer

Model

- Input: (ARX)

$$x_t = [u_{t-1}, u_{t-2}, y_{t-1}, y_{t-2}]^\top$$

- Nonlinear feature map:

$$\phi(x_t) = \sigma(W x_t + b)$$

  $W \rightsquigarrow$ a matrix with dimension $\#parameters \times 4$:

  $b \rightsquigarrow$ is a vector with dimension $\#parameters$

  $w_{i,j} \sim \mathcal{N}(0, \gamma^2)$, $b_i \sim \mathcal{U}(0, 2\pi]$

  $\sigma \rightsquigarrow$ activation function.

- Neural network with frozen first layer
- As $\#parameters \to \infty$ it approximates the Gaussian kernel map.

  Rahimi, A. and Recht, B. (2008). *Random Features for Large-Scale Kernel Machines.* Advances in Neural Information Processing Systems 20, 1177–1184

Results

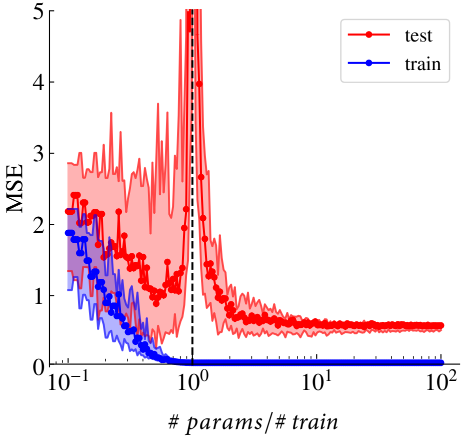


Figure: **Double-descent in system identification.** MSE = Mean square error.

Parameter Norm

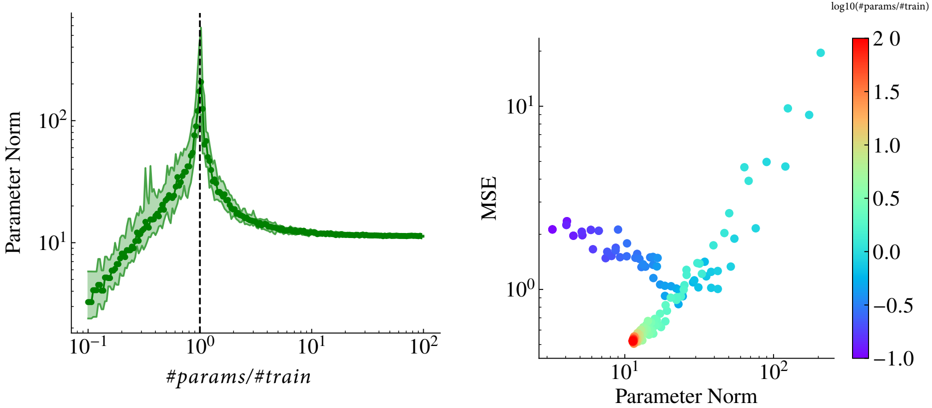


Figure: *Left:* Parameter norm double desent curve. *Right:* Test MSE *vs* parameter norm.

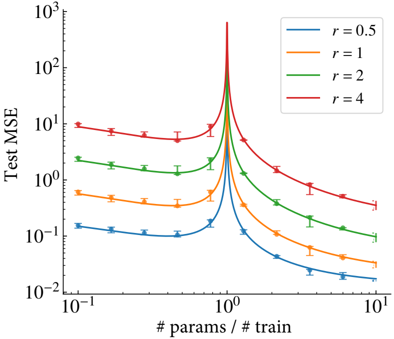
Double-descent and benign-overfitting in dynamical systems

- Asymptotic results are available for the i.i.d. case (but not for system identification)

  T. Hastie, A. Montanari, S. Rosset, and R. J. Tibshirani,"Surprises in High-Dimensional Ridgeless Least Squares Interpolation," Annals of Statisics. 50(2): 949-986 (2022).

- "Consistency results" available for the i.i.d. case (but not for system identification)

  P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, "Benign overfitting in linear regression," Proceedings of the National Academy of Sciences, vol. 117, no. 48, pp. 30063–30070, Apr. 2020.



III - Adversarial examples

Adversarial examples

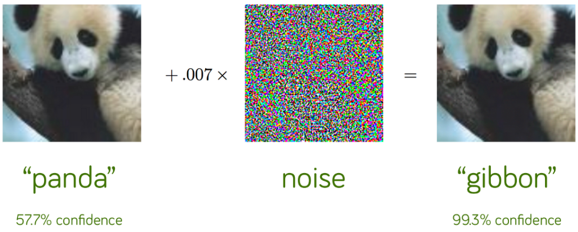


Figure: Adversarial examples in image classification.

Source: I. J. Goodfellow, J. Shlens, C. Szegedy , *“Explaining and Harnessing Adversarial Examples”*, ICLR 2015.

Adversarial robustness

"what is the role of high-dimensionality in model robustness?"

Regularization properties of adversarially-trained linear regression
Antônio H. Ribeiro, Dave Zachariah, Francis Bach, Thomas B. Schön.
Submited NeurIPS (2023)

Overparameterized Linear Regression under Adversarial Attack.
Antônio H. Ribeiro, Thomas B. Schön.
IEEE Transactions on Signal Processing (2023)

Framework: Linear regression

Simplest case where adversarial vulnerability has been observed.

I. J. Goodfellow, J. Shlens, C. Szegedy , *"Explaining and Harnessing Adversarial Examples"*, ICLR 2015
D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Ma, *"Robustness May Be At Odds with Accuracy,"* ICLR, p. 23, 2019.

- Training dataset:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \Rightarrow \widehat{\beta}$$

Framework: Linear regression

Simplest case where adversarial vulnerability has been observed.

I. J. Goodfellow, J. Shlens, C. Szegedy , *"Explaining and Harnessing Adversarial Examples"*, ICLR 2015
D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Ma, *"Robustness May Be At Odds with Accuracy,"* ICLR, p. 23, 2019.

- Training dataset:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \Rightarrow \widehat{\beta}$$

- Model prediction

$$\widehat{y} = \widehat{\beta}^\mathsf{T} x$$

Framework: Linear regression

Simplest case where adversarial vulnerability has been observed.

I. J. Goodfellow, J. Shlens, C. Szegedy , *"Explaining and Harnessing Adversarial Examples"*, ICLR 2015
D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Ma, *"Robustness May Be At Odds with Accuracy,"* ICLR, p. 23, 2019.

- Training dataset:
$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \Rightarrow \widehat{\beta}$$
- Model prediction
$$\widehat{y} = \widehat{\beta}^\mathsf{T} x$$
- $\text{Error}(\widehat{\beta}) = |y - x^\mathsf{T}\widehat{\beta}|$

Framework: Linear regression

Simplest case where adversarial vulnerability has been observed.

I. J. Goodfellow, J. Shlens, C. Szegedy , *"Explaining and Harnessing Adversarial Examples"*, ICLR 2015
D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Ma, *"Robustness May Be At Odds with Accuracy,"* ICLR, p. 23, 2019.

- Training dataset:
$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n) \Rightarrow \widehat{\beta}$$
- Model prediction
$$\widehat{y} = \widehat{\beta}^\mathsf{T} x$$
- $\text{Error}(\widehat{\beta}) = |y - x^\mathsf{T}\widehat{\beta}|$
- $\text{Adv-error}(\widehat{\beta}) = \max_{\|\Delta x\| \le \delta} \left|y - (x + \Delta x)^\mathsf{T}\widehat{\beta}\right|$

Adversarial error in linear regression

- $\text{Error}(\widehat{\beta}) = |y - x^\mathsf{T}\widehat{\beta}|$
- $\text{Adv-error}(\widehat{\beta}) = \max_{\|\Delta x\| \le \delta} \left| y - (x + \Delta x)^\mathsf{T}\widehat{\beta} \right|$
- *Dual formula for the adversarial error*

$$\left(\text{Adv-error}(\widehat{\beta})\right)^2 = \left(|\text{Error}(\widehat{\beta})| + \delta\|\widehat{\beta}\|_*\right)^2$$

- where $\|\cdot\|_*$ is the dual norm.

ℓ_p-adversarial attacks

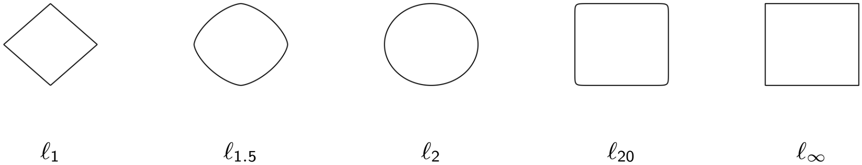
- ℓ_∞-adversarial attack: $\{\|\Delta x\|_\infty \leq \delta\} \Rightarrow$ dual norm: $\|\Delta x\|_1$

ℓ_p-adversarial attacks

- ℓ_∞-adversarial attack: $\{\|\Delta x\|_\infty \leq \delta\} \Rightarrow$ dual norm: $\|\Delta x\|_1$
- ℓ_2-adversarial attack: $\{\|\Delta x\|_2 \leq \delta\} \Rightarrow$ dual norm: $\|\Delta x\|_2$

ℓ_p-adversarial attacks

- ℓ_∞-adversarial attack: $\{\|\Delta x\|_\infty \leq \delta\} \Rightarrow$ dual norm: $\|\Delta x\|_1$
- ℓ_2-adversarial attack: $\{\|\Delta x\|_2 \leq \delta\} \Rightarrow$ dual norm: $\|\Delta x\|_2$
- ℓ_p-adversarial attack: $\{\|\Delta x\|_p \leq \delta\} \Rightarrow$ dual norm: $\|\Delta x\|_q$ for $1/p + 1/q = 1$



Analysing adversarial robustness

From:

$$\mathbb{E}\left[\left(\text{Adv-error}(\widehat{\beta})\right)^2\right] = \mathbb{E}\left[\left(|\text{Error}(\widehat{\beta})| + \delta\|\widehat{\beta}\|_*\right)^2\right]$$

Analysing adversarial robustness

From:

$$\mathbb{E}\left[\left(\text{Adv-error}(\widehat{\beta})\right)^2\right] = \mathbb{E}\left[\left(|\text{Error}(\widehat{\beta})| + \delta\|\widehat{\beta}\|_*\right)^2\right]$$

It follows that:

$$\mathbb{E}[\text{Error}(\widehat{\beta})^2] + \delta^2\|\widehat{\beta}\|_*^2 \leq \mathbb{E}[(\text{Adv. error}(\widehat{\beta}))^2] \leq 2\left(\mathbb{E}[\text{Error}(\widehat{\beta})^2] + \delta^2\|\widehat{\beta}\|_*^2\right).$$

Double-descent in the adversarial loss

$\|\widehat{\beta}\|_2$ also present a double descent behavior.

$$\mathbb{E}[(\ell_2\text{-adv. error}(\widehat{\beta}))^2] \propto \mathbb{E}[\text{Error}(\widehat{\beta})^2] + \delta^2 \|\widehat{\beta}\|_2^2.$$

Double-descent in the adversarial loss

$\|\widehat{\beta}\|_2$ also present a double descent behavior.

$$\mathbb{E}[(\ell_2\text{-adv. error}(\widehat{\beta}))^2] \propto \mathbb{E}[\text{Error}(\widehat{\beta})^2] + \delta^2\|\widehat{\beta}\|_2^2.$$

as illustrated in the example below:

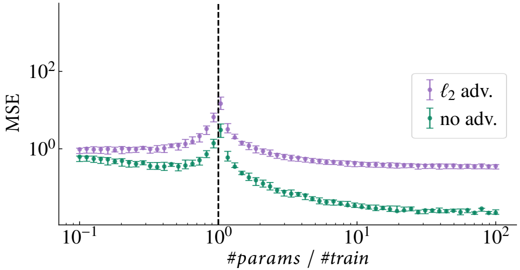


Figure: Adv. risk. minimum ℓ_2-norm interpolator

Adversarial training

- One of the most effective approaches for deep learning models to defend against adversarial attacks.
- Training models on samples that have been modified by an adversary
- Min-max problem, searching for the best solution to the worst-case attacks

Adversarial training in linear models

- Adversarial training,

$$\frac{1}{n}\sum_{i=1}^{n} \max_{\|\Delta x\| \leq \delta} (y_i - (x_i + \Delta x)^\mathsf{T} \beta)^2$$

Adversarial training in linear models

- Adversarial training,

$$\frac{1}{n}\sum_{i=1}^{n}\max_{\|\Delta x\|\le\delta}(y_i - (x_i + \Delta x)^{\mathsf{T}}\beta)^2$$

can be reformulated as

$$\frac{1}{n}\sum_{i=1}^{n}\left(|y_i - x_i^{\mathsf{T}}\beta| + \delta\|\beta\|_*\right)^2$$

Minimum-norm interpolator and adversarial training

Theorem

Adversarial training is minimized at the minimum norm interpolator

$$\min_{\beta} \|\beta\|_* \quad \text{subject to} \quad X\beta = y$$

iff $0 < \delta < \bar{\delta}$.

Regularization properties of adversarially-trained linear regression
Antônio H. Ribeiro, Dave Zachariah, Francis Bach, Thomas B. Schön.
Submited NeurIPS (2023)

New interpretation for minimum-norm interpolator

- Minimum norm interpolator is equivalent to a models adversarially trained with $\delta_{\text{train}} = \bar{\delta}$

New interpretation for minimum-norm interpolator

- Minimum norm interpolator is equivalent to a models adversarially trained with $\delta_{\text{train}} = \bar{\delta}$
- We can compute $\bar{\delta}$ from the last theorem

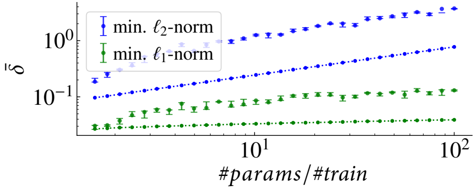


Figure: Threshold $\bar{\delta}$ *vs* number of features *m*.

New interpretation for minimum-norm interpolator

- Minimum norm interpolator is equivalent to a models adversarially trained with $\delta_{\text{train}} = \bar{\delta}$
- We can compute $\bar{\delta}$ from the last theorem

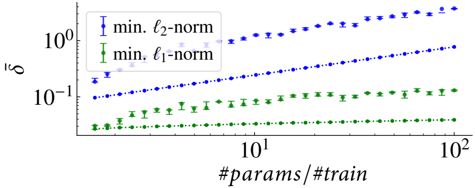


Figure: Threshold $\bar{\delta}$ *vs* number of features m.

- Upper bound on the test adversarial error of minimum-norm interpolators

$$\sqrt{\mathbb{E}[(\text{Adv. error}(\widehat{\beta}))^2]} - \sqrt{\mathbb{E}[\text{Error}(\widehat{\beta})^2]} \lesssim \frac{\delta_{\text{test}}}{\delta_{\text{train}}}$$

Minimum ℓ_2-norm interpolator under ℓ_∞ adversarial attacks

We had from before:

$$\mathbb{E}[(\ell_\infty\text{-adv. error}(\widehat{\beta}))^2] \propto \mathbb{E}[\text{Error}(\widehat{\beta})^2] + \delta^2 \|\widehat{\beta}\|_1^2.$$

Minimum ℓ_2-norm interpolator under ℓ_∞ adversarial attacks

We had from before:

$$\mathbb{E}[(\ell_\infty\text{-adv. error}(\widehat{\beta}))^2] \propto \mathbb{E}[\text{Error}(\widehat{\beta})^2] + \delta^2 \|\widehat{\beta}\|_1^2.$$

Minimum ℓ_2-norm interpolator and Gaussian features:

$$\|\widehat{\beta}\|_1 = \mathcal{O}(1) \qquad \delta \propto \mathbb{E}\|x\|_2 = \mathcal{O}(\sqrt{m}).$$

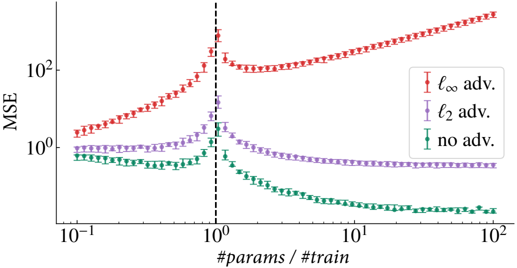


Figure: Adv. risk. minimum ℓ_2-norm interpolator

Overparameterized Linear Regression under Adversarial Attack.
Antônio H. Ribeiro, Thomas B. Schön.
IEEE Transactions on Signal Processing (2023)

Next directions

1. Tailored solver;

Next directions

1. Tailored solver;
2. Generalize to other losses;

Next directions

1. Tailored solver;
2. Generalize to other losses;
3. Generalization to nolinear models.

Summary

- Minimum-norm interpolators as a simple model to study generalization.
- Double-descent and benign-overfiting.
- Double descent can be observed in dynamic-systems.
- Dual formula for the adversarial error in linear models:

$$\left(\text{Adv-error}(\widehat{\beta})\right)^2 = \left(|\text{Error}(\widehat{\beta})| + \delta\|\widehat{\beta}\|_*\right)^2$$

- Minimum-norm interpolation is equivalent to adversarial training with $\bar{\delta}$

Thank you!

✉ antonio.horta.ribeiro@it.uu.se
🌐 antonior92.github.io