Universität Stuttgart

Luiz F. O. Chamon

robust learning

IMPRS BootCamp
Sep. 15th, 2023

ellis

SimTech Cluster of Excellence

Who am I?

Luiz F. O. Chamon

2022–: ELLIS-SimTech research group leader
IMPRS-IS faculty

2021–2022: Simons Institute, UC Berkeley (Postdoc)

2020: University of Pennsylvania (PhD)

< 2015: University of São Paulo, Brazil (BSc. & MSc.)

- I speak 4.5 languages (German = 0)

Disclaimer

https://luizchamon.com/pdf/imprs_bootcamp.pdf

What is about to follow...

- ...is **very** opinionated. Change my mind!
- ...contains ~~affiliated links~~ **shameless** plugs to my own work
- ...contains images and illustrations that I do not own: **do not share publicly**
- ...is **not** exhaustive

Agenda

What is "robust learning"?

Why are models brittle?

How to learn robustly?

What else are adversaries good for?

Agenda

What is "robust learning"?

Why are models brittle?

How to learn robustly?

What else are adversaries good for?

What is learning?

What is learning?

- **Learning:** "model-free statistics" (data-driven)
- **Inference:** "model-driven statistics" (~~data-free~~)

[Rostamizadeh, Talwalkar, Mohri. Foundations of machine learning, 2012]; [Ben-David, Shalev-Shwartz. Understanding machine learning..., 2014]

What is learning?

- **Learning:** "model-free statistics" (data-driven)
- **Inference:** "model-driven statistics" (~~data-free~~)

Examples

- How good is an estimator (sample mean)? ⇒ confidence interval vs. cross-validation

[Rostamizadeh, Talwalkar, Mohri. Foundations of machine learning, 2012]; [Ben-David, Shalev-Shwartz. Understanding machine learning..., 2014]

What is learning?

- **Learning:** "model-free statistics" (data-driven)
- **Inference:** "model-driven statistics" (~~data-free~~)

Examples

- How good is an estimator (sample mean)? ⇒ confidence interval vs. cross-validation
- Statistical optimization: $P^\star = \min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\left[\ell\left(f_\theta(x), y\right)\right]$
  - ℓ is the bounded, Lipschitz continuous, possibly non-convex loss
  - f_θ is the (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]

What is learning?

- **Learning:** "model-free statistics" (data-driven)
- **Inference:** "model-driven statistics" (~~data-free~~)

Examples

- How good is an estimator (sample mean)? ⇒ confidence interval vs. cross-validation
- Statistical optimization: $\hat{P}^\star = \min_\theta \frac{1}{N}\sum_{n=1}^{N} \ell\left(f_\theta(x_n), y_n\right), \quad (x_n, y_n) \sim \mathfrak{D}$ (i.i.d.)
  - ℓ is the bounded, Lipschitz continuous, possibly non-convex loss
  - f_θ is the (possibly nonlinear) parametrization [e.g., logistic classifier, (G)(C)NN]

What is learning theory?

$$\min_\theta \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\left(f_\theta(x_n), y_n\right) \xrightarrow{\text{"LLN"}} \min_\theta \mathbb{E}\left[\text{Loss}\left(f_\theta(x), y\right)\right]$$

- Is this possible with a finite number of samples (learnability)? When?
  e.g., linear functions, smooth functions (finite RKHS norm, bandlimited), NNs...
- How many samples do we need to get a good approximation (sample complexity)?
  $\approx 1/\epsilon^2$

Robustness in learning

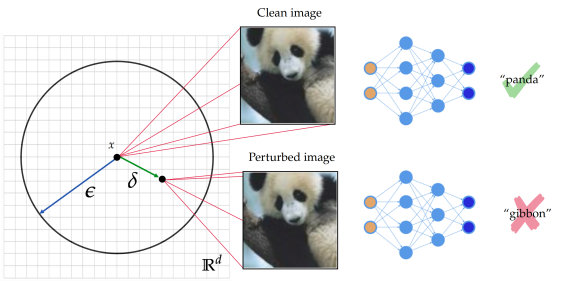
"Since such external effects are rarely expected to help achieve desired system behavior [...] they are commonly referred as *disturbances*.

One of the main objectives of feedback is to render a system insensitive to such disturbances."

[Dullerud, Paganini. A course in robust control theory, 2005.]

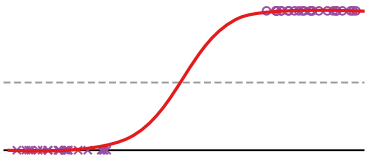
- Strong against input disturbances (≈ classical control: stability)
- Property of model/estimator/classifier

Robustness in learning



Smoothness vs. Robustness

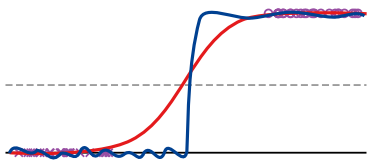
Smoothness ⇒ Robustness



[Hein, Andriushchenko, NeurIPS'17]; [Pauli et al., IEEE Control Systems Letters'22]; [Bungert, Trillos, Murray, IMA Information and Inference'23]

Smoothness vs. Robustness

Smoothness ⇍ Robustness



[Hein, Andriushchenko, NeurIPS'17]; [Pauli et al., IEEE Control Systems Letters'22]; [Bungert, Trillos, Murray, IMA Information and Inference'23]

Robustness in ~~statistics~~ inference

"*robustness signifies insensitivity to small deviations from the assumptions.*

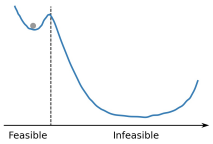
[...] the shape of the true underlying distribution deviates slightly from the assumed model (usually the Gaussian law)."

[Huber and Ronchetti. Robust Statistics, 2009.]

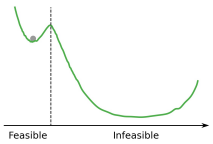
- Strong against model misspecification (≈ robust control)
  strong against "model misspecification" ≡ (*resistant*) [Hampel's theorem]
- Property of statistical problem

Resilience vs. Robustness

Robustness
(hard to break)



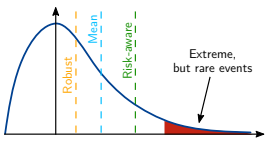
Resilience
(easy to fix)



Risk vs. Robustness

	Frequent	Rare
Relevant	Model	*Risk*
Irrelevant	Noise	Outlier

Robustness



- Value at risk (VaR, V@R), conditional VaR (CVaR, CV@R), expected shortfall…

[Shapiro et al. Lectures on Stochastic Programming, 2014]; [Kalogerias, **C.**, Pappas, Ribeiro. Better Safe Than Sorry…, IEEE ICASSP'20 (best paper)]

Summary

…in learning

- Strong against input disturbances
- Property of estimator
- "Test-time" distribution shift

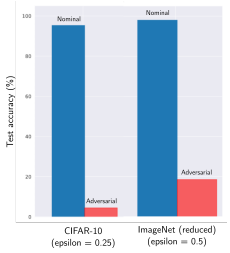
≠ …in inference (property of statistical problem)

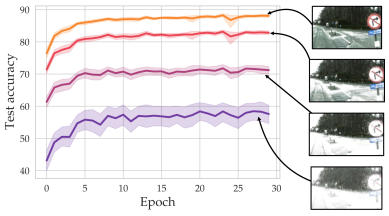
≠ smoothness

≠ resilience

≠ risk

Is it actually an issue?





[Ilyas et al. Adversarial examples are not bugs…, NeurIPS'19]; [Robey, Hassani, Pappas. Model-based robust deep learning…, arXiv'20]

Agenda

What is "robust learning"?

Why are models brittle?

How to learn robustly?

What else are adversaries good for?

Why are models brittle?

We don't really know
[long awkward pause]

- Overfitting
- Margin
- Features

Why are models brittle?

We don't really know
[long awkward pause]

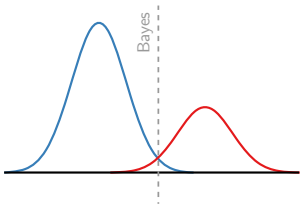
- Overfitting
- Margin
- Features

Why are models brittle?

We don't really know
[long awkward pause]

- ~~Overfitting~~ → optimal estimators (and even the "true" one) can be brittle
- Margin
- Features

Binary classification

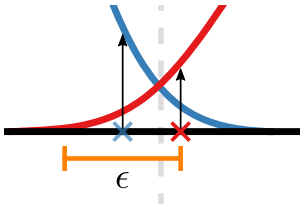


$$y = \begin{cases} +1, & \text{with prob. } \pi \\ -1, & \text{with prob. } 1-\pi \end{cases}$$

$$x \mid y \sim \text{Normal}(y\mu, \sigma^2 \boldsymbol{I})$$

$$\Delta = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\|_2 \leq \epsilon\}$$

Binary classification



$$y = \begin{cases} +1, & \text{with prob. } \pi \\ -1, & \text{with prob. } 1-\pi \end{cases}$$

$$x \mid y \sim \text{Normal}(y\mu, \sigma^2 \boldsymbol{I})$$

$$\Delta = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\|_2 \leq \epsilon\}$$

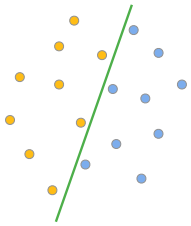
Why are models brittle?

We don't really know
[long awkward pause]

- ~~Overfitting~~ → optimal estimators (and even the "true" one) can be brittle
- Margin
- Features

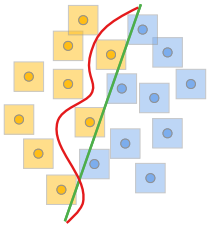
The margin argument

Data points close to the decision boundary are not robust



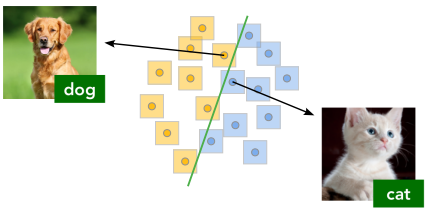
The margin argument

Data points close to the decision boundary are not robust



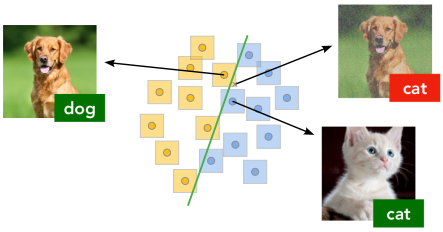
The margin argument

Data points close to the decision boundary are not robust



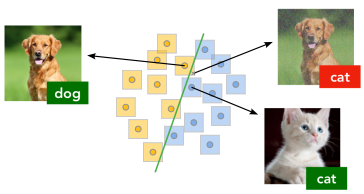
The margin argument

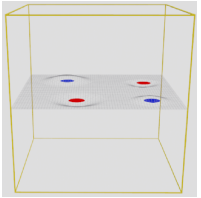
Data points close to the decision boundary are not robust



The margin argument

Data points close to the decision boundary are not robust





Why are models brittle?

We don't really know
[long awkward pause]

- ~~Overfitting~~ → optimal estimators (and even the "true" one) can be brittle
- Margin
- Features

The features argument

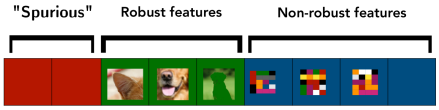
Some features are more robust than others

- Adversarial examples arise from "spurious" features

[Ilyas, Santurkar, Tsipras, Engstrom, Tran, Mądry. Adversarial examples are not bugs, they are features, NeurIPS'19]

The features argument

Some features are more robust than others



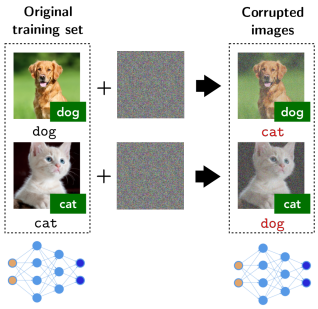
- Adversarial examples arise from "spurious" features
- Useful, non-robust features

[Ilyas, Santurkar, Tsipras, Engstrom, Tran, Mądry. Adversarial examples are not bugs, they are features, NeurIPS'19]

The features argument

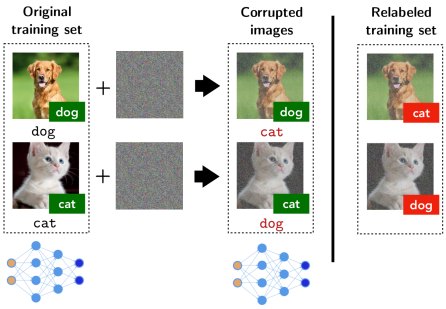
[Ilyas, Santurkar, Tsipras, Engstrom, Tran, Mądry. Adversarial examples are not bugs, they are features, NeurIPS'19]

The features argument



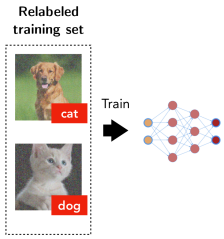
[Ilyas, Santurkar, Tsipras, Engstrom, Tran, Mądry. Adversarial examples are not bugs, they are features, NeurIPS'19]

The features argument



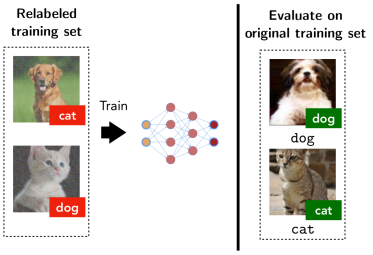
[Ilyas, Santurkar, Tsipras, Engstrom, Tran, Mądry. Adversarial examples are not bugs, they are features, NeurIPS'19]

The features argument



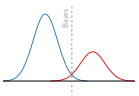
[Ilyas, Santurkar, Tsipras, Engstrom, Tran, Mądry. Adversarial examples are not bugs, they are features, NeurIPS'19]

The features argument



[Ilyas, Santurkar, Tsipras, Engstrom, Tran, Mądry. Adversarial examples are not bugs, they are features, NeurIPS'19]

Summary



~~Overfitting~~

Margin

Features

Why are models brittle?

Learning is doing exactly what we asked for...

...and robustness is NOT what we asked for!

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\text{Loss}(f_\theta(x), y)\right]$$

Agenda

What is "robust learning"?

Why are models brittle?

How to learn robustly?

What else are adversaries good for?

Robust learning(ish)

Post-processing

- Filtering

$$f_{\theta\uparrow} \longrightarrow f_{\theta\uparrow} \circ \varphi$$

[Das, Shanbhogue, Chen, Hohman, Li, Chen, Kounavis, Chau. SHIELD: Fast, Practical..., KDD'18]
[Shin, Dawn. JPEG-resistant adversarial images, arXiv'18]

Robust learning(ish)

Post-processing

- Filtering

$$f_{\theta\uparrow} \longrightarrow f_{\theta\uparrow} \circ \varphi$$

[Das, Shanbhogue, Chen, Hohman, Li, Chen, Kounavis, Chau. SHIELD: Fast, Practical..., KDD'18]
[Shin, Dawn. JPEG-resistant adversarial images, arXiv'18]

- Smoothed classifiers

$$f_{\theta\uparrow}(\cdot) \longrightarrow \mathbb{E}_{z\sim\mathrm{m}}\left[f_{\theta\uparrow}(\cdot + z)\right]$$

[Lecuyer, Atlidakis, Geambasu, Hsu, and Jana. Certified robustness to adversarial examples..., IEEE SSP'19]
[Cohen, Rosenfeld, Kolter. Certified adversarial robustness via randomized smoothing, ICML'19]
[Blum, Dick, Manoj, Zhang. Random smoothing might be unable to certify ℓ_∞ robustness..., JMLR'20]
[Anderson, Sojoudi. Certified Robustness via Locally Biased Randomized Smoothing, L4DC'22]

Robust learning(ish)

Post-processing

- Filtering

$$f_{\theta\uparrow} \longrightarrow f_{\theta\uparrow} \circ \varphi$$

[Das, Shanbhogue, Chen, Hohman, Li, Chen, Kounavis, Chau. SHIELD: Fast, Practical..., KDD'18]
[Shin, Dawn. JPEG-resistant adversarial images, arXiv'18]

- Smoothed classifiers

$$f_{\theta\uparrow}(\cdot) \longrightarrow \mathbb{E}_{z\sim\mathrm{m}}\left[f_{\theta\uparrow}(\cdot + z)\right] = (f_{\theta\uparrow} * \mathrm{m})(\cdot)$$

[Lecuyer, Atlidakis, Geambasu, Hsu, and Jana. Certified robustness to adversarial examples..., IEEE SSP'19]
[Cohen, Rosenfeld, Kolter. Certified adversarial robustness via randomized smoothing, ICML'19]
[Blum, Dick, Manoj, Zhang. Random smoothing might be unable to certify ℓ_∞ robustness..., JMLR'20]
[Anderson, Sojoudi. Certified Robustness via Locally Biased Randomized Smoothing, L4DC'22]

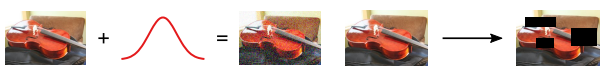
Robust learning

Data augmentation

- Random augmentation

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\left(f_\theta(x_n), y_n\right) \longrightarrow \min_{\theta} \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K} \text{Loss}\left(f_\theta(x_n + \delta_k), y_n\right), \quad \delta_k \sim \mathrm{m}$$

[Holmstrom, Koistinen. Using additive noise in back-propagation training, IEEE TNN'92]
[DeVries, Taylor. Improved regularization of convolutional neural networks with cutout, arXiv'17]
[Ford, Gilmer, Carlini, Cubuk. Adversarial examples are a natural consequence of test error in noise, ICML'19]



Robust learning

Data augmentation

- Random augmentation

$$\min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\left(f_\theta(x_n), y_n\right) \longrightarrow \min_{\theta} \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K} \text{Loss}\left(f_\theta(x_n + \delta_k), y_n\right), \quad \delta_k \sim \mathrm{m}$$

[Holmstrom, Koistinen. Using additive noise in back-propagation training, IEEE TNN'92]
[DeVries, Taylor. Improved regularization of convolutional neural networks with cutout, arXiv'17]
[Ford, Gilmer, Carlini, Cubuk. Adversarial examples are a natural consequence of test error in noise, ICML'19]

- "Pseudolabeling"

Pretrained network: $h^\dagger$
Unlabeled dataset: $\{z\}$

$$\longrightarrow \min_{\theta} \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\left(f_\theta(x_n), y_n\right) + \frac{1}{K}\sum_{k=1}^{K} \text{Loss}\left(f_\theta(z_n), h^\dagger(z_n)\right)$$

[Carmon, Raghunathan, Schmidt, Duchi, Liang. Unlabeled data improves adversarial robustness. NeurIPS'19]
[Gowal, Rebuffi, Wiles, Stimberg, Calian, Mann. Improving robustness using generated data. NeurIPS'21]

Robust learning

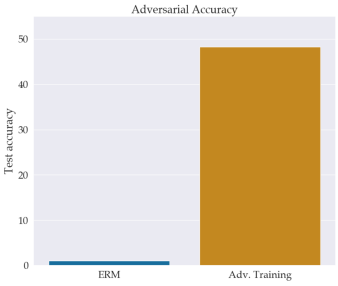
Robust optimization

- Adversarial training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}),y\big)\Big] \longrightarrow \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_\theta(\boldsymbol{x}+\delta),y\big)\Big]$$

[Goodfellow, Shlens, Szegedy. Explaining and harnessing adversarial examples, ICLR'15]
[Mądry, Makelov, Schmidt, Tsipras, Vladu. Towards deep learning models resistant to adversarial attacks, ICLR'18]
[HKSC'16, AJFR'17, PPAK'18, SBFZ'19, SCB'19, KCP'20, BTT'21, KTM'22, JFEPF'22, WDJTM'23, NFPIY'23…] (RobustBench)

Adversarial training



[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21]

Robust learning

Robust optimization

- Adversarial training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}),y\big)\Big] \longrightarrow \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_\theta(\boldsymbol{x}+\delta),y\big)\Big]$$

[Goodfellow, Shlens, Szegedy. Explaining and harnessing adversarial examples, ICLR'15]
[Mądry, Makelov, Schmidt, Tsipras, Vladu. Towards deep learning models resistant to adversarial attacks, ICLR'18]
[HKSC'16, AJFR'17, PPAK'18, SBFZ'19, SCB'19, KCP'20, BTT'21, KTM'22, JFEPF'22, WDJTM'23, NFPIY'23…] (RobustBench)

Robust learning

Robust optimization

- Adversarial training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}),y\big)\Big] \longrightarrow \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_\theta(G_\delta(\boldsymbol{x})),y\big)\Big]$$

[Goodfellow, Shlens, Szegedy. Explaining and harnessing adversarial examples, ICLR'15]
[Mądry, Makelov, Schmidt, Tsipras, Vladu. Towards deep learning models resistant to adversarial attacks, ICLR'18]
e.g., [Robey, Hassani, Pappas. Model-based robust deep learning: Generalizing to natural..., arXiv'20]

Robust learning

Robust optimization

- Adversarial training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}),y\big)\Big] \longrightarrow \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_\theta(G_\delta(\boldsymbol{x})),y\big)\Big]$$

[Goodfellow, Shlens, Szegedy. Explaining and harnessing adversarial examples, ICLR'15]
[Mądry, Makelov, Schmidt, Tsipras, Vladu. Towards deep learning models resistant to adversarial attacks, ICLR'18]
e.g., [Robey, Hassani, Pappas. Model-based robust deep learning: Generalizing to natural..., arXiv'20]

- Distributional robustness

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}),y\big)\Big] \longrightarrow \min_{\theta} \sup_{p\in\mathcal{P}} \mathbb{E}_{(x,y)\sim p}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}),y\big)\Big]$$

[Ben-Tal, Nemirovski, El Ghaoui. Robust Optimization, 2009]
[Sinha, Namkoong, Volpi, Duchi. Certifying some distributional robustness with principled adversarial..., ICLR'18]

Summary

- Post-processing: filtering ($f_{\theta\uparrow} \circ \varphi$), smoothed classifiers ($\mathbb{E}_{z\sim\mathrm{m}}[f_{\theta\uparrow}(\cdot + z)]$)
  - ✅ No need to retrain
  - ❌ Poor empirical performance, weak guarantees
- (Randomized) data augmentation: $\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\mathbb{E}_{\delta\sim\mathrm{m}}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}+\delta),y\big)\Big]\Big]$
  - ✅ Low complexity
  - ❌ Poor empirical performance, no guarantees
- Adversarial training: $\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_\theta(\boldsymbol{x}+\delta),y\big)\Big]$
  - ✅ Great(!) empirically performance
  - ❌ …

Break
(10 min)

Robust learning

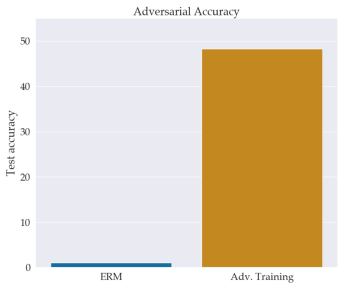
Robust optimization

- Adversarial training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\text{Loss}\big(f_\theta(\boldsymbol{x}),y\big)\Big] \longrightarrow \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{\delta\in\Delta} \text{Loss}\big(f_\theta(\boldsymbol{x}+\delta),y\big)\Big]$$

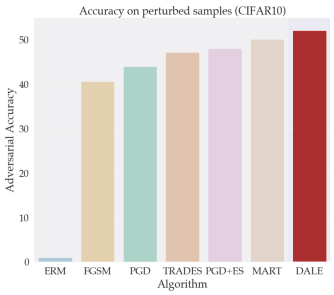
[Goodfellow, Shlens, Szegedy. Explaining and harnessing adversarial examples, ICLR'15]
[Mądry, Makelov, Schmidt, Tsipras, Vladu. Towards deep learning models resistant to adversarial attacks, ICLR'18]
[HKSC'16, AJFR'17, PPAK'18, SBFZ'19, SCB'19, KCP'20, BTT'21, KTM'22, JFEPF'22, WDJTM'23, NFPIY'23…] (RobustBench)

Adversarial training



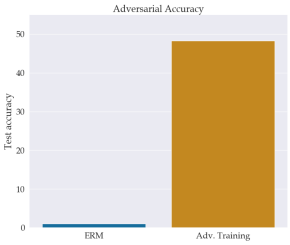
[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21]

Adversarial training



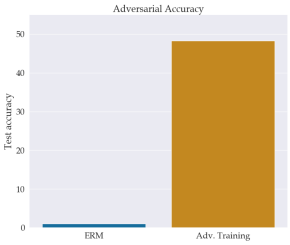
[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21]

Adversarial training



- Great(!) empirical performance
- Grounded on a *robust* literature
- Statistical complexity
- Performance trade-offs
- Computational complexity

Adversarial training



- Great(!) empirical performance
- Grounded on a *robust* literature
- Statistical complexity
- Performance trade-offs
- Computational complexity

Statistical complexity

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\Big)\right] \xrightarrow{\;?\;} \min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\Big)\right]$$

- Is robust learning harder than non-robust learning? Do we need more samples?

Statistical complexity

$$\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}), y_n\Big)\right] \xrightarrow{\;?\;} \min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)}\left[\max_{\boldsymbol{\delta}\in\Delta} \text{Loss}\Big(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), y\Big)\right]$$

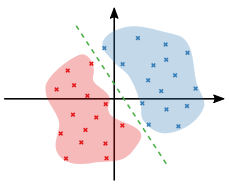
- Is robust learning harder than non-robust learning? Do we need more samples?
  A: YES *and* NO

[Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18]
[Yin, Ramchandran, Bartlett. Rademacher Complexity for Adversarially Robust Generalization, ICML'19]
[Montasser, Hanneke, Srebro. VC Classes are Adversarially Robustly Learnable, but Only Improperly, COLT'19]
[Awasthi, Frank, Mohri. Adversarial Learning Guarantees for Linear Hypotheses and Neural Networks, ICML'20]
[Montasser, Hanneke, Srebro. Adversarially robust learning: A generic minimax optimal learner & characterization, NeurIPS'22]

Halfspace classifiers with 0-1 loss

$$P^\star = \min_{\boldsymbol{\theta}} \Pr_{(\boldsymbol{x},y)}\Big[f_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq y\Big]$$

$$\hat{P}^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\Big[f_{\boldsymbol{\theta}}(\boldsymbol{x}_n) \neq y_n\Big]$$



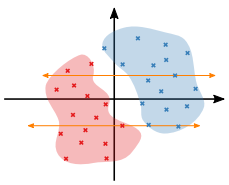
- 0-1 loss, i.e., probability of error
- Halfspace classifiers, i.e., $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbb{I}\left[\boldsymbol{\theta}^\top \boldsymbol{x} + \bar{\theta} \geq 0\right]$
- $\left|P^\star - \hat{P}^\star\right| \leq \epsilon$ w.h.p. $\Leftrightarrow N \approx \frac{d+1}{\epsilon^2}$

[Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18]

Halfspace classifiers with 0-1 loss

$$P_r^\star = \min_{\boldsymbol{\theta}} \Pr_{(\boldsymbol{x},y)}\Big[\exists \boldsymbol{\delta} \in \Delta \text{ s.t. } f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\Big]$$

$$\hat{P}_r^\star = \min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\Big[\exists \boldsymbol{\delta} \in \Delta \text{ s.t. } f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}) \neq y_n\Big]$$



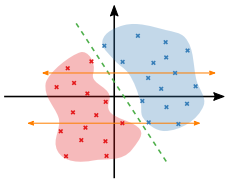
- Robust 0-1 loss
- $\Delta = \{\boldsymbol{\delta} \mid \boldsymbol{\delta} = c\boldsymbol{z}, c \in \mathbb{R}\}$ for fixed $\boldsymbol{z}$

[Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18]

Halfspace classifiers with 0-1 loss

$$P_r^\star = \min_\theta \Pr_{(x,y)}\left[\exists \delta \in \Delta \text{ s.t. } f_\theta(x+\delta) \neq y\right]$$

$$\hat{P}_r^\star = \min_\theta \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[\exists \delta \in \Delta \text{ s.t. } f_\theta(x_n+\delta) \neq y_n\right]$$



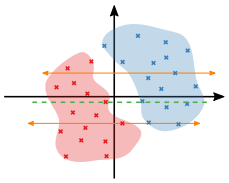
- Robust 0-1 loss
- $\Delta = \{\delta \mid \delta = cz, c \in \mathbb{R}\}$ for fixed z

[Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18]

Halfspace classifiers with 0-1 loss

$$P_r^\star = \min_\theta \Pr_{(x,y)}\left[\exists \delta \in \Delta \text{ s.t. } f_\theta(x+\delta) \neq y\right]$$

$$\hat{P}_r^\star = \min_\theta \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[\exists \delta \in \Delta \text{ s.t. } f_\theta(x_n+\delta) \neq y_n\right]$$



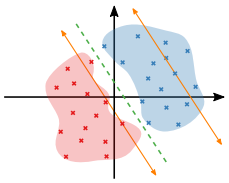
- Robust 0-1 loss
- $\Delta = \{\delta \mid \delta = cz, c \in \mathbb{R}\}$ for fixed z
- $\left|P_r^\star - \hat{P}_r^\star\right| \leq \epsilon$ w.h.p. $\Leftrightarrow N \approx \frac{d+1}{\epsilon^2}$ $(P_r^\star \gg P^\star)$

[Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18]

Halfspace classifiers with 0-1 loss

$$P_r^\star = \min_\theta \Pr_{(x,y)}\left[\exists \delta \in \Delta \text{ s.t. } f_\theta(x+\delta) \neq y\right]$$

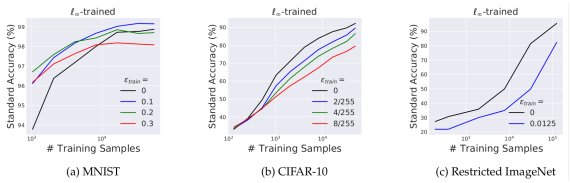
$$\hat{P}_r^\star = \min_\theta \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[\exists \delta \in \Delta \text{ s.t. } f_\theta(x_n+\delta) \neq y_n\right]$$



- Robust 0-1 loss
- $\Delta = \{\delta \mid \delta = cz, c \in \mathbb{R}\}$ for fixed z
- $\left|P_r^\star - \hat{P}_r^\star\right| \leq \epsilon$ w.h.p. $\Leftrightarrow N \approx \frac{d+1}{\epsilon^2}$ $(P_r^\star = P^\star \text{ !!!})$

[Cullina, Bhagoji, Mittal. PAC-learning in the presence of evasion adversaries, NeurIPS'18]

Nominal performance of robust models



(a) MNIST (b) CIFAR-10 (c) Restricted ImageNet

[Tsipras, Santurkar, Engstrom, Turner, Mądry. Robustness May Be at Odds with Accuracy, ICLR'19]

"Softer" robustness

- Softmax or *log-sum-exp*

$$\min_\theta \mathbb{E}_{(x,y)}\left[\frac{1}{\tau}\log\left(\mathbb{E}_{\delta\sim\mathrm{m}}\left[e^{\tau\cdot\mathrm{Loss}\left(f_\theta(x+\delta),y\right)}\right]\right)\right]$$

  - $\tau \to 0$: classical learning (with randomized data augmentation)
  - $\tau \to \infty$: adversarial robustness (ess sup)

[Li, Beirami, Sanjabi, Smith. Tilted empirical risk minimization, ICLR'21]

"Softer" robustness

- Softmax or *log-sum-exp*

$$\min_\theta \mathbb{E}_{(x,y)}\left[\frac{1}{\tau}\log\left(\mathbb{E}_{\delta\sim\mathrm{m}}\left[e^{\tau\cdot\mathrm{Loss}\left(f_\theta(x+\delta),y\right)}\right]\right)\right]$$

  - $\tau \to 0$: classical learning (with randomized data augmentation)
  - $\tau \to \infty$: adversarial robustness (ess sup)

[Li, Beirami, Sanjabi, Smith. Tilted empirical risk minimization, ICLR'21]

- L_p norms

$$\min_\theta \mathbb{E}_{(x,y)}\left[\mathbb{E}_{\delta\sim\mathrm{m}}\left[\left|\mathrm{Loss}\left(f_\theta(x+\delta),y\right)\right|^\tau\right]^{1/\tau}\right]$$

  - $\tau = 1$: classical learning (with randomized data augmentation)
  - $\tau \to \infty$: adversarial robustness (ess sup)

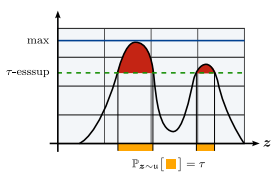
[Rice, Bair, Zhang, Kolter. Robustness between the worst and average case, NeurIPS'21]

"Softer" robustness

- Probabilistic robustness

$$\min_\theta \mathbb{E}_{(x,y)}\left[\underset{\delta\in\Delta}{\tau\text{-esssup}}\ \mathrm{Loss}\left(f_\theta(x+\delta),y\right)\right]$$

  - $\tau = 1/2$: classical learning (for symmetric m)
  - $\tau = 0$: adversarial robustness (ess sup)



[Robey, C., Pappas, Hassani. Probabilistically robust learning: Balancing average- and worst-case..., ICML'22 (spotlight)]

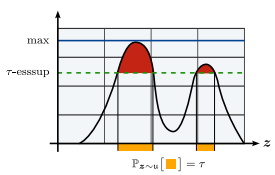
"Softer" robustness

- Probabilistic robustness

$$\min_\theta \mathbb{E}_{(x,y)}\left[\underset{\delta\in\Delta}{\tau\text{-esssup}}\ \mathrm{Loss}\left(f_\theta(x+\delta),y\right)\right]$$

  - $\tau = 1/2$: classical learning (for symmetric m)
  - $\tau = 0$: adversarial robustness (ess sup)

- Potentially better sample complexity
  [Robey, C., Pappas, Hassani, ICML'22 (spotlight)]
  [Raman, Tewari, Subedi, NeurIPS ML Safety Workshop'22]



[Robey, C., Pappas, Hassani. Probabilistically robust learning: Balancing average- and worst-case..., ICML'22 (spotlight)]

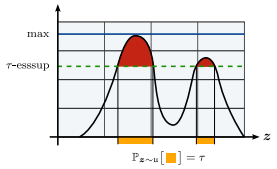
"Softer" robustness

- Probabilistic robustness

$$\min_\theta \mathbb{E}_{(x,y)}\left[\underset{\delta\in\Delta}{\tau\text{-esssup}}\ \text{Loss}\big(f_\theta(x+\delta),y\big)\right]$$

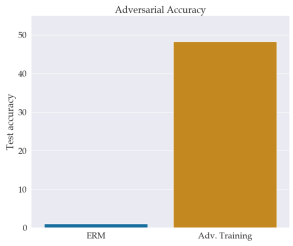
  - $\tau = 1/2$: classical learning (for symmetric m)
  - $\tau = 0$: adversarial robustness (ess sup)

- Potentially better sample complexity
  [Robey, C., Pappas, Hassani, ICML'22 (spotlight)]
  [Raman, Tewari, Subedi, NeurIPS ML Safety Workshop'22]
- Better performance trade-off
  [Robey, C., Pappas, Hassani, ICML'22 (spotlight)]



[Robey, C., Pappas, Hassani. Probabilistically robust learning: Balancing average- and worst-case..., ICML'22 (spotlight)]

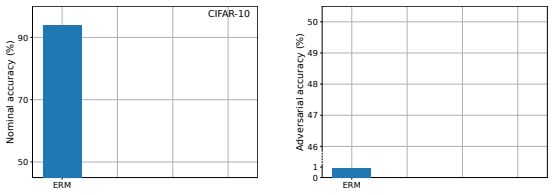
Adversarial training



- Great(!) empirical performance
- Grounded on a *robust* literature
- Statistical complexity
- Performance trade-offs
- Computational complexity

Nominal vs. adversarial performance

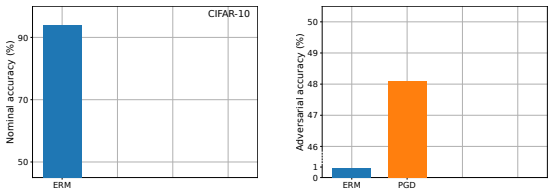
$$\min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x),y\big)\Big]$$



[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21]

Nominal vs. adversarial performance

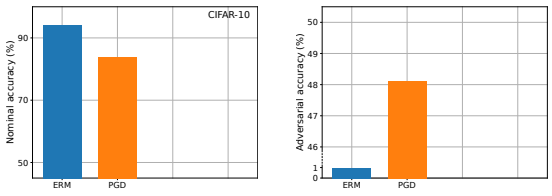
$$\min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x),y\big)\Big] \longrightarrow \min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\max_{\|\delta\|_\infty\le\epsilon} \text{Loss}\big(f_\theta(x+\delta),y\big)\Big]$$



[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21]

Nominal vs. adversarial performance

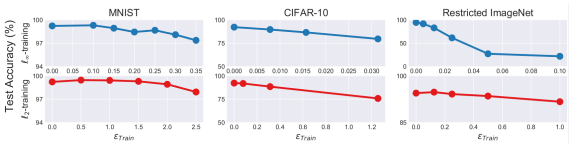
$$\min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x),y\big)\Big] \longrightarrow \min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\max_{\|\delta\|_\infty\le\epsilon} \text{Loss}\big(f_\theta(x+\delta),y\big)\Big]$$



[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21]

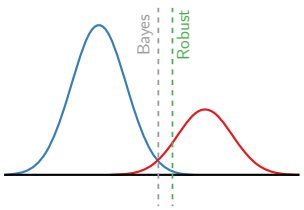
Nominal vs. adversarial performance

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\text{Loss}\big(f_\theta(x),y\big)\Big] \longrightarrow \min_\theta \mathbb{E}_{(x,y)\sim\mathfrak{D}}\Big[\max_{\|\delta\|_\infty\le\epsilon} \text{Loss}\big(f_\theta(x+\delta),y\big)\Big]$$



[Tsipras, Santurkar, Engstrom, Turner, Mądry. Robustness may be at odds with accuracy, ICLR'19]

Binary classification



$$y = \begin{cases} +1, & \text{with prob. } \pi \\ -1, & \text{with prob. } 1-\pi \end{cases}$$

$$x \mid y \sim \text{Normal}(y\mu, \sigma^2 \boldsymbol{I})$$

$$\Delta = \{\delta \mid \|\delta\|_2 \le \epsilon\}$$

[Dobriban, Hassani, Hong, Robey. Provable tradeoffs in adversarially robust classification, IEEE TIT'22]

Nominal vs. adversarial performance

Learning is doing exactly what we asked for!

Nominal vs. adversarial performance

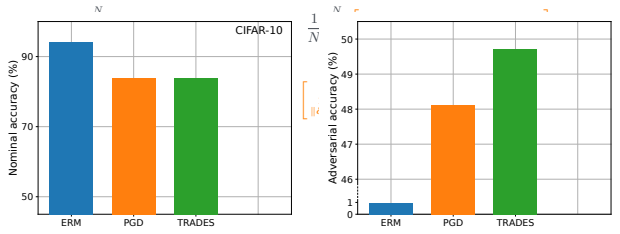
(Penalized) adversarial training

$$\min_\theta \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big) \longrightarrow \min_\theta \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

$$\min_\theta \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big) + \lambda \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

Nominal vs. adversarial performance

(Penalized) adversarial training



[Zhang, Yu, Jiao, Xing, El Ghaoui, Jordan. Theoretically principled trade-off between robustness and accuracy, ICML'19]

Nominal vs. adversarial performance

(Penalized) adversarial training

$$\min_\theta \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big) + \lambda \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right]$$

- No straightforward relation between λ and adversarial loss
- λ depends on the values of the losses (dataset, model, performance measure)
- Generalization

Nominal vs. adversarial performance

Constrained learning

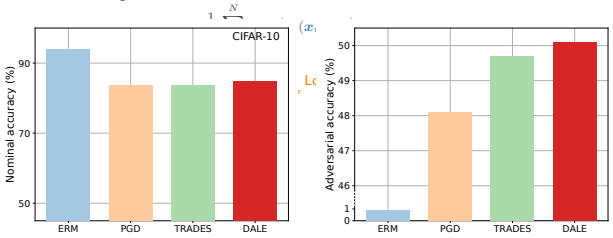
$$\min_\theta \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to } \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right] \leq c$$

[C. and Ribeiro, NeurIPS'20]; [Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]; [C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Nominal vs. adversarial performance

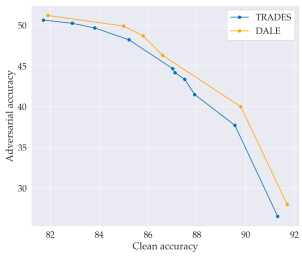
Constrained learning



[C. and Ribeiro, NeurIPS'20]; [Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]; [C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Nominal vs. adversarial performance

Constrained learning



[C. and Ribeiro, NeurIPS'20]; [Robey*, C.*, Pappas, Hassani, and Ribeiro, NeurIPS'21]; [C., Paternain, Calvo-Fullana, and Ribeiro, IEEE TIT'23]

Nominal vs. adversarial performance

Constrained learning

$$\min_\theta \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to } \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right] \leq c$$

- Does constrained learning generalize?
- How many samples do we need?
- How do we solve constrained learning problems?

Nominal vs. adversarial performance

Constrained learning

$$\min_\theta \frac{1}{N}\sum_{n=1}^{N} \text{Loss}\big(f_\theta(\boldsymbol{x}_n), y_n\big)$$

$$\text{subject to } \frac{1}{N}\sum_{n=1}^{N} \left[\max_{\|\boldsymbol{\delta}\|_\infty \leq \epsilon} \text{Loss}\big(f_\theta(\boldsymbol{x}_n + \boldsymbol{\delta}), y_n\big)\right] \leq c$$

- Does constrained learning generalize? ⇒ yes, same as unconstrained learning
- How many samples do we need? ⇒ essentially the same as unconstrained learning
- How do we solve constrained learning problems? ⇒ using duality (despite non-convexity)

[C., Paternain, Calvo-Fullana, and Ribeiro. The empirical duality gap of constrained..., IEEE ICASSP'20 (best student paper)]
[C. and Ribeiro. Probably approximately correct constrained learning, NeurIPS'20]
[C., Paternain, Calvo-Fullana, and Ribeiro. Constrained learning with non-convex losses, IEEE TIT'23]

Nominal vs. adversarial performance

Dual empirical constrained learning

$$\max_{\lambda\ge 0}\ \min_{\boldsymbol{\theta}}\ \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n),y_n\big)+\lambda\left[\max_{\|\boldsymbol{\delta}\|_\infty\le\epsilon}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}),y_n\big)-c\right]$$

Nominal vs. adversarial performance

Dual empirical constrained learning

$$\max_{\lambda\ge 0}\ \min_{\boldsymbol{\theta}}\ \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n),y_n\big)+\lambda\left[\max_{\|\boldsymbol{\delta}\|_\infty\le\epsilon}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}),y_n\big)-c\right]$$

Nominal vs. adversarial performance

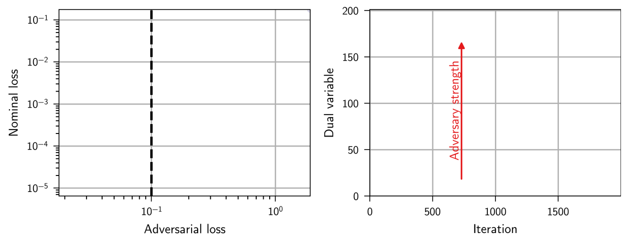
Dual empirical constrained learning

$$\max_{\lambda\ge 0}\ \min_{\boldsymbol{\theta}}\ \frac{1}{N}\sum_{n=1}^{N}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n),y_n\big)+\lambda\left[\max_{\|\boldsymbol{\delta}\|_\infty\le\epsilon}\text{Loss}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}_n+\boldsymbol{\delta}),y_n\big)-c\right]$$

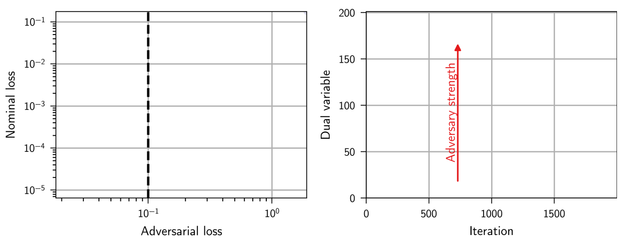
- c (static) is part of the problem, λ (dynamic) is part of the algorithm
- $\neq$ penalty/regularization (static λ)

[Byrd, Lipton. What is the effect of importance weighting in deep learning?, ICML'19]
[Sagawa*, Koh*, Hashimoto, Liang. Distributionally robust neural networks, ICLR'20]
[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21]
[C., Paternain, Calvo-Fullana, and Ribeiro. Constrained learning with non-convex losses, IEEE TIT'23]

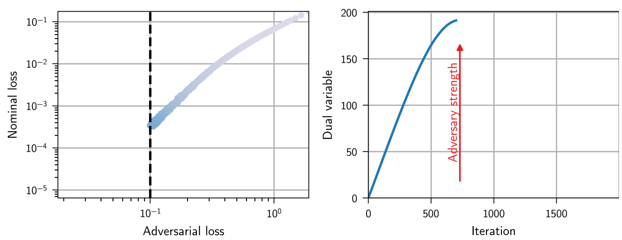
Robust image recognition



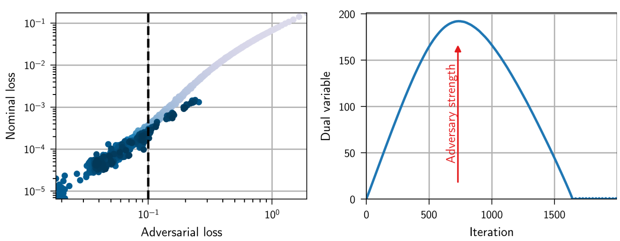
Robust image recognition



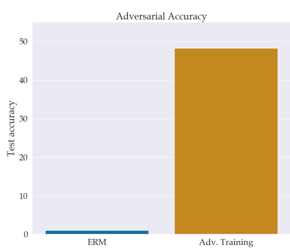
Robust image recognition



Robust image recognition



Adversarial training



- ✅ Great(!) empirical performance
- ✅ Grounded on a *robust* literature
- ❌ Statistical complexity
- ❌ Performance trade-offs
- ❌ Computational complexity

Computational complexity

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- Hard to evaluate/optimize

Computational complexity

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- Hard to evaluate/optimize

$$\text{“}\nabla_\theta\text{”} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right] \text{“}=\text{”} \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta^\star), y \Big)$$

$$\delta^\star \in \operatorname*{argmax}_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big)$$

  - Typically non-convex, e.g., $f_\theta(\cdot) = \sum_{m=1}^{M} \theta_m \kappa(\cdot, c_m)$
  - Often underparametrized ($\theta \in \mathbb{R}^p$, $x \in \mathbb{R}^n$, and $p \gg n$)

Computational complexity

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- Hard to evaluate/optimize

$$\partial_\theta \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right] \ni \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta^\star), y \Big)$$

$$\delta^\star \in \operatorname*{argmax}_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big)$$

  - Typically non-convex, e.g., $f_\theta(\cdot) = \sum_{m=1}^{M} \theta_m \kappa(\cdot, c_m)$
  - Often underparametrized ($\theta \in \mathbb{R}^p$, $x \in \mathbb{R}^n$, and $p \gg n$)

“PGD”

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- “PGD” [Mądry et al., ICLR'18] (alternating optimization, gradient ascent-descent)

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\quad \delta^{k+1} \leftarrow \operatorname*{proj}_{\Delta} \left[\delta^k + \eta \operatorname{sign}\Big(\nabla_\delta \text{Loss}\big(f_{\theta^k}(x+\delta^k), y \big) \Big) \right]$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y \Big)$

“PGD”

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- “PGD” [Mądry et al., ICLR'18] (alternating optimization, gradient ascent-descent)

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\quad \delta^{k+1} \leftarrow \operatorname*{proj}_{\Delta} \left[\delta^k + \eta \operatorname{sign}\Big(\nabla_\delta \text{Loss}\big(f_{\theta^k}(x+\delta^k), y \big) \Big) \right]$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y \Big)$

“PGD”

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- “PGD” [Mądry et al., ICLR'18] (alternating optimization, gradient ascent-descent)

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\quad \delta^{k+1} \leftarrow \operatorname*{proj}_{\Delta} \left[\delta^k + \eta \operatorname{sign}\Big(\nabla_\delta \text{Loss}\big(f_{\theta^k}(x+\delta^k), y \big) \Big) \right]$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y \Big)$

- Random initialization
- Restarts
- Pruning
- Adaptive step size

[Dhillon et al., ICLR'18]; [Carmon et al., NeurIPS'19]; [Wu et al., NeurIPS'20]; [Cheng et al., IJCAI'22]

“PGD”

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- “PGD” [Mądry et al., ICLR'18] (alternating optimization, gradient ascent-descent)

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\quad \delta^{k+1} \leftarrow \operatorname*{proj}_{\Delta} \left[\delta^k + \eta \operatorname{sign}\Big(\nabla_\delta \text{Loss}\big(f_{\theta^k}(x+\delta^k), y \big) \Big) \right]$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y \Big)$

“PGD”

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

- “PGD” [Mądry et al., ICLR'18] (alternating optimization, gradient ascent-descent)

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\quad \delta^{k+1} \leftarrow \operatorname*{proj}_{\Delta} \left[\delta^k + \eta \operatorname{sign}\Big(\nabla_\delta \text{Loss}\big(f_{\theta^k}(x+\delta^k), y \big) \Big) \right]$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y \Big)$ $\quad \approx$ data augmentation

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

$$\updownarrow \approx$$

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}} \left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)} \left[\text{Loss}\Big(f_\theta(x+\delta), y\Big) \right] \right]$$

- For any approximation error, $\exists\, \gamma(x,y)$ such that

$$\mu(\delta|x,y) \propto \Big[\text{Loss}\big(f_\theta(x+\delta), y\big) - \gamma(x,y)\Big]_+$$

Dual Adversarial LEarning

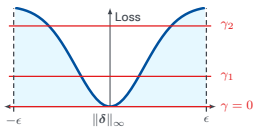
$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

$$\updownarrow \approx$$

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}} \left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)} \left[\text{Loss}\Big(f_\theta(x+\delta), y\Big) \right] \right]$$

- For any approximation error, $\exists\, \gamma(x,y)$ such that

$$\mu(\delta|x,y) \propto \Big[\text{Loss}\big(f_\theta(x+\delta), y\big) - \gamma(x,y)\Big]_+$$



Dual Adversarial LEarning

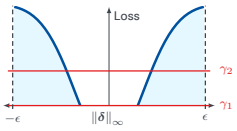
$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

$$\updownarrow \approx$$

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}} \left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)} \left[\text{Loss}\Big(f_\theta(x+\delta), y\Big) \right] \right]$$

- For any approximation error, $\exists\, \gamma(x,y)$ such that

$$\mu(\delta|x,y) \propto \Big[\text{Loss}\big(f_\theta(x+\delta), y\big) - \gamma(x,y)\Big]_+$$



Dual Adversarial LEarning

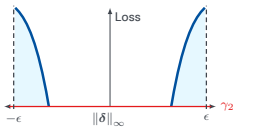
$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

$$\updownarrow \approx$$

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}} \left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)} \left[\text{Loss}\Big(f_\theta(x+\delta), y\Big) \right] \right]$$

- For any approximation error, $\exists\, \gamma(x,y)$ such that

$$\mu(\delta|x,y) \propto \Big[\text{Loss}\big(f_\theta(x+\delta), y\big) - \gamma(x,y)\Big]_+$$



Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

$$\updownarrow =$$

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}} \left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)} \left[\text{Loss}\Big(f_\theta(x+\delta), y\Big) \right] \right]$$

- For any approximation error, $\exists\, \gamma(x,y)$ such that

$$\mu(\delta|x,y) \propto \Big[\text{Loss}\big(f_\theta(x+\delta), y\big) - \gamma(x,y)\Big]_+$$

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

$$\updownarrow \approx$$

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathfrak{D}} \left[\mathbb{E}_{\delta\sim\mu(\delta|x,y)} \left[\text{Loss}\Big(f_\theta(x+\delta), y\Big) \right] \right]$$

- For any approximation error, $\exists\, \gamma(x,y)$ such that

$$\mu(\delta|x,y) \propto \text{Loss}\big(f_\theta(x+\delta), y\big)$$

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

1: $\delta_t \sim \text{Loss}\big(f_\theta(x+\delta), y\big)$
2: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y\Big)$

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y\Big) \right]$$

1: $\delta_t \sim \text{Loss}\big(f_\theta(x+\delta), y\big)$
2: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y\Big)$

- Use MCMC techniques (e.g., Langevin Monte Carlo)

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\quad \delta^{k+1} = \underset{\Delta}{\text{proj}} \left[\delta^k + \eta \, \text{sign} \left[\nabla_\delta \log \big(\text{Loss}\big(f_\theta(x+\delta^k), y \big) \big) \right] + \sqrt{2\eta T}\zeta \right], \quad \zeta \sim \text{Laplace}(0, I)$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y \Big)$

- Use MCMC techniques (e.g., Langevin Monte Carlo) [DALE → "PGD" as $T \to 0$, ...]

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

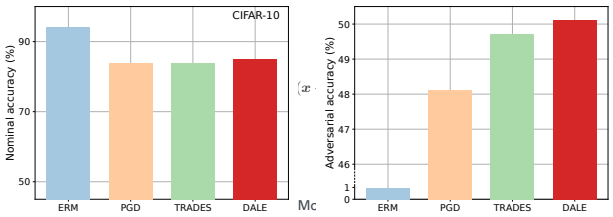
1: $\delta^1 \leftarrow \delta_{t-1}$
2: **for** $k = 1, \dots, K$
3: $\quad \delta^{k+1} = \underset{\Delta}{\text{proj}} \left[\delta^k + \eta \, \text{sign} \left[\nabla_\delta \log \big(\text{Loss}\big(f_\theta(x+\delta^k), y \big) \big) \right] + \sqrt{2\eta T}\zeta \right], \quad \zeta \sim \text{Laplace}(0, I)$
4: **end**
5: $\delta_t \leftarrow \delta^{K+1}$
6: $\theta_{t+1} = \theta_t - \eta \nabla_\theta \text{Loss}\Big(f_\theta(x+\delta_t), y \Big)$

- Use MCMC techniques (e.g., Langevin Monte Carlo) [DALE → "PGD" as $T \to 0$, ...]
- Sampling works under milder conditions than optimization

[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21] 58

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$

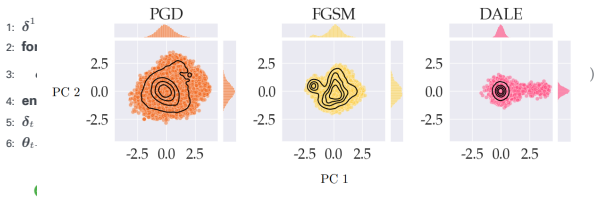


- Sampling works under milder conditions than optimization

[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21] 58

Dual Adversarial LEarning

$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta(x+\delta), y \Big) \right]$$



- Sampling works under milder conditions than optimization

[Robey*, C.*, Pappas, Ribeiro, Hassani. Adversarial robustness with semi-infinite constrained learning, NeurIPS'21] 58

Summary

- Adversarial training

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \Big[\text{Loss}\big(f_\theta(x), y \big) \Big] \longrightarrow \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \Delta} \text{Loss}\Big(f_\theta\big(x+\delta\big), y \Big) \right]$$

- Great(!) empirically performance
- Statistical complexity
  ⇒ change what robustness means (e.g., probabilistic robustness)
- Performance trade-offs
  ⇒ constrained learning
- Computational complexity
  ⇒ sampling and DALE

Agenda

What is "robust learning"?

Why are models brittle?

How to learn robustly?

What else are adversaries good for?

Friendly adversaries

- Non-robustness robust properties
  - Invariance [Hounie, C., Ribeiro. Automatic data augmentation via invariance-constrained learning, ICML'23]
  - Smoothness [Cervino, C., Haeffele, Vidal, Ribeiro. Learning Globally Smooth Functions on Manifolds, ICML'23]
  - ...

Friendly adversaries

- Non-robustness robust properties
  - Invariance [Hounie, C., Ribeiro. Automatic data augmentation via invariance-constrained learning, ICML'23]

$$\text{Loss}\big(f_\theta(gx), y \big) = \text{Loss}\big(f_\theta(x), y \big), \forall g \in \mathcal{G} \Rightarrow \max_{g \in \mathcal{G}} \text{Loss}\big(f_\theta(gx), y \big) - \text{Loss}\big(f_\theta(x), y \big) \leq \epsilon$$

$$\vdots$$

  - Smoothness [Cervino, C., Haeffele, Vidal, Ribeiro. Learning Globally Smooth Functions on Manifolds, ICML'23]
  - ...

61

Friendly adversaries

- Non-robustness robust properties
  - Invariance [Hounie, C., Ribeiro. Automatic data augmentation via invariance-constrained learning, ICML'23]

$$\text{Loss}\big(f_\theta(gx), y\big) = \text{Loss}\big(f_\theta(x), y\big), \forall g \in \mathcal{G} \Rightarrow \max_{g \in \mathcal{G}} \text{Loss}\big(f_\theta(gx), y\big) - \text{Loss}\big(f_\theta(x), y\big) \leq \epsilon$$

$$\vdots$$

  - Smoothness [Cervino, C., Haeffele, Vidal, Ribeiro. Learning Globally Smooth Functions on Manifolds, ICML'23]

$$\min_{\theta} \quad \max_{z \in \mathcal{M}} \left\| \nabla_{\mathcal{M}} f_\theta(z) \right\|^2 \quad \text{(Lipschitz constant)}$$
$$\text{subject to} \quad \mathbb{E}_{(x,y)\sim\mathfrak{D}} \Big[\text{Loss}\big(f_\theta(x), y\big) \Big] \leq c$$

  - ...

Friendly adversaries

- Density estimation (GANs) [Goodfellow et al. Generative adversarial nets, NeurIPS'14]

$$\min_{\theta} \max_{d} \mathbb{E}_{x\sim\mathfrak{D}} \Big[\log\big(d(x)\big) \Big] + E_{z\sim\mathfrak{m}} \Big[\log\big(1 - d(f_\theta(z))\big) \Big]$$
$$\big[d : x \leftarrow 1 \text{ and } f_\theta(z) \leftarrow 0 \big]$$

Friendly adversaries

- Density estimation (GANs) [Goodfellow et al. Generative adversarial nets, NeurIPS'14]

$$\min_{\theta} \max_{d} \mathbb{E}_{x\sim\mathfrak{D}} \Big[\log\big(d(x)\big) \Big] + E_{z\sim\mathfrak{m}} \Big[\log\big(1 - d(f_\theta(z))\big) \Big]$$
$$\big[d : x \leftarrow 1 \text{ and } f_\theta(z) \leftarrow 0 \big]$$

- Semi-supervised and unsupervised learning
  [Cervino, C., Haeffele, Vidal, Ribeiro. Learning Globally Smooth Functions on Manifolds, ICML'23]
  [Chen, Kornblith, Norouzi, Hinton. A simple framework for contrastive learning of visual representations, ICML'20] (SimCLR)
  [Bochkovskiy, Wang, Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv'20]
- ...
- ...

Concluding remarks

Concluding remarks

- robust learning = strong against input disturbances
  Important, yes... but essential?

Concluding remarks

- robust learning = strong against input disturbances
  Important, yes... but essential?
- Is *adversarial training* THE solution?
  Not when it matters...

Concluding remarks

- robust learning = strong against input disturbances
  Important, yes... but essential?
- Is *adversarial training* THE solution?
  Not when it matters...
- So are we done with *adversarial training*?
  Absolutely not!

Universität Stuttgart

Luiz F. O. Chamon

www.luizchamon.com
luiz.chamon@simtech.uni-stuttgart.de

robust learning