Assignment 3

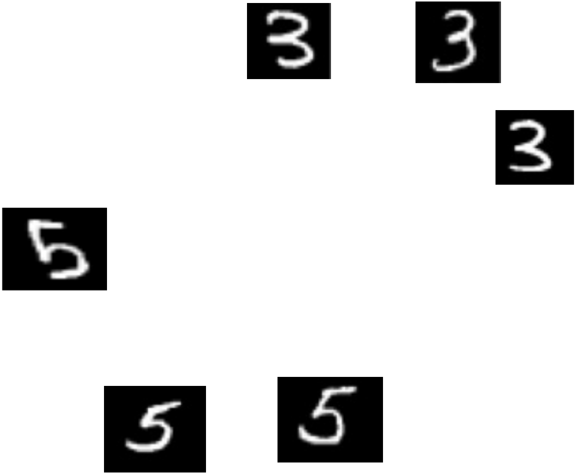
Training robust neural networks

Benjamin Negrevergne, Laurent Meunier, Alexandre Vérine

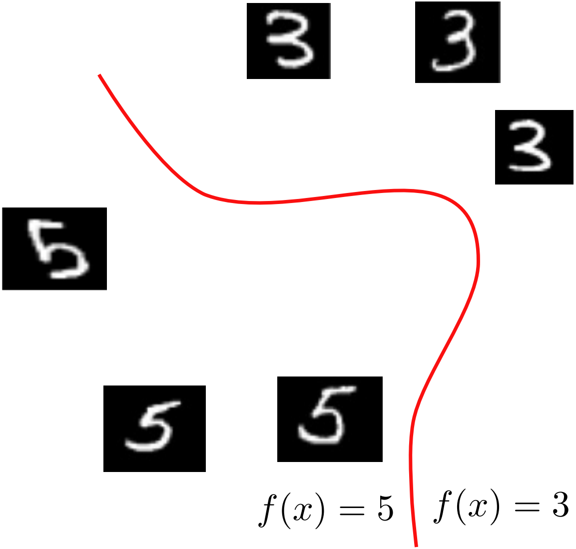
PSL University – Paris Dauphine – Équipes *MILES*

Outline

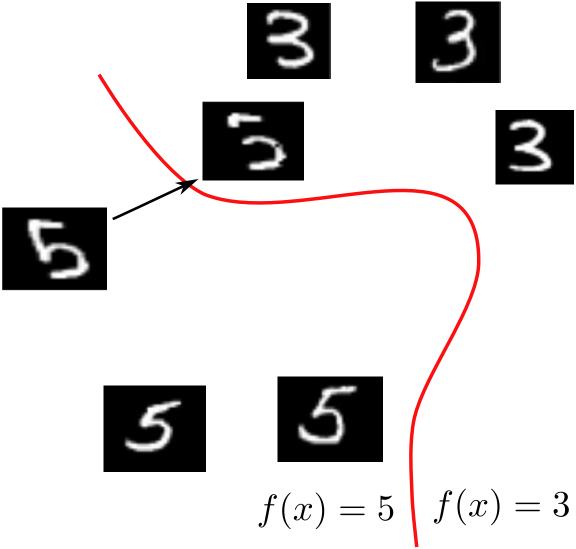
Adversarial examples explained



Adversarial examples explained

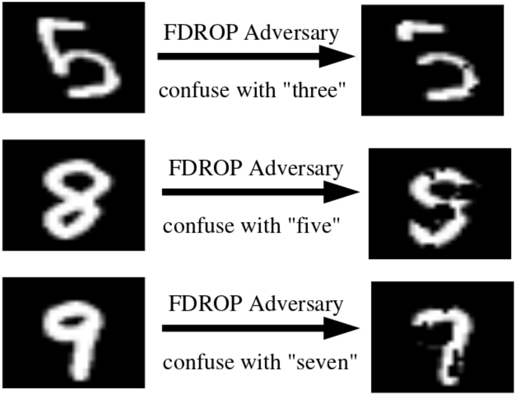


Adversarial examples explained



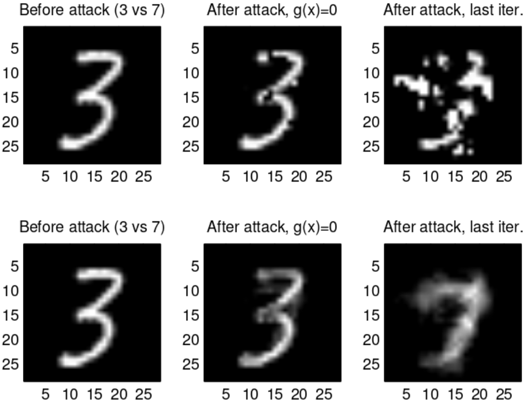
Early work on adversarial attacks

Globerson et al. (ICML, 2006)

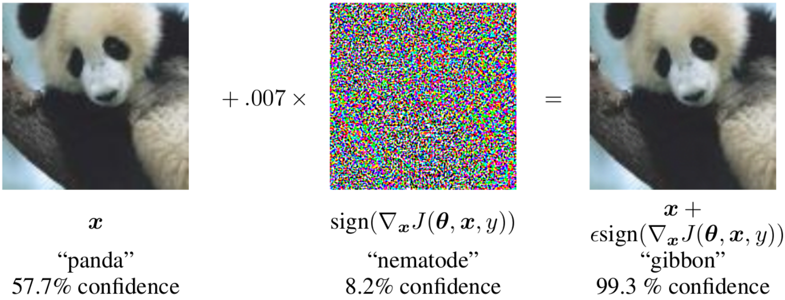


Early work on adversarial attacks

Biggio et al. (ECML, 2013)



FGSM (2015)



Goodfellow et al. (ICLR, 2015)

The modification is imperceptible!

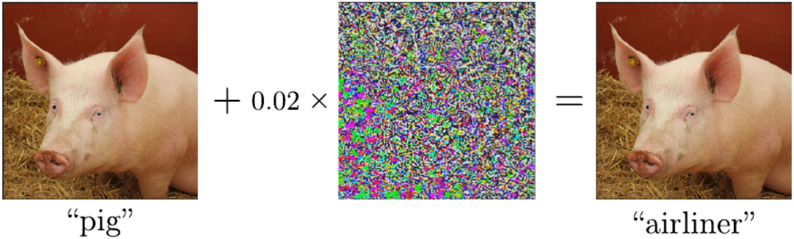
Modern attacks

Natural	ℓ_1 **– EAD 60**	ℓ_2 **– C&W 60**	ℓ_∞ **– PGD 20**
0.958	0.035	0.034	0.384

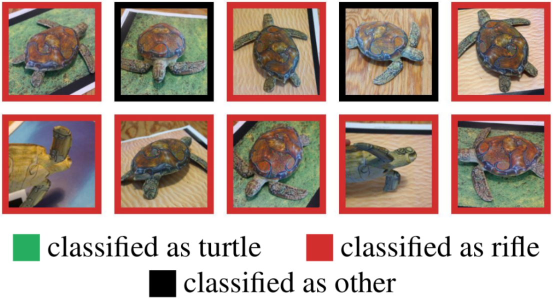
$\sim$ 3% accuracy under attack

- Almost every input image can be attacked!

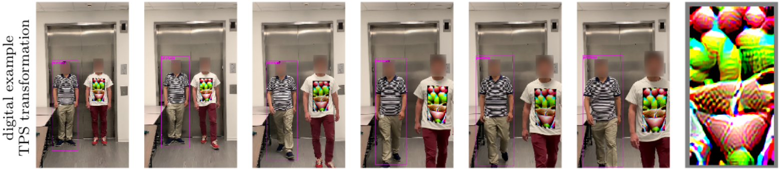
Pig vs. Airliner



Real life adversarial examples



Synthesizing Robust Adversarial Examples, Athalye et al. 2017



Evading Real-Time Person Detectors by Adversarial T-shirt, Xu et al. 2019

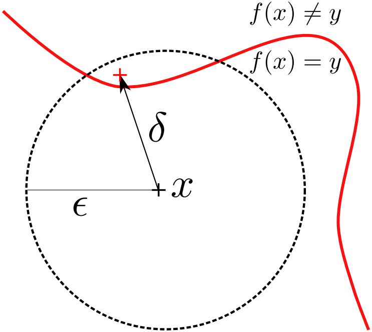
Goal of this assignment

- Understand the weaknesses of machine learning models
  - Learn attack mechanisms
  - Learn defence mechanisms

- Learn how to reason about the decision boundary

Generating adversarial examples

Let $f : \mathbb{R}^n \to Y$ a classifier
Given an example $x \in \mathbb{R}^n$ and its true label $y \in Y$
find a $\delta \in \mathbb{R}^n$ such that:

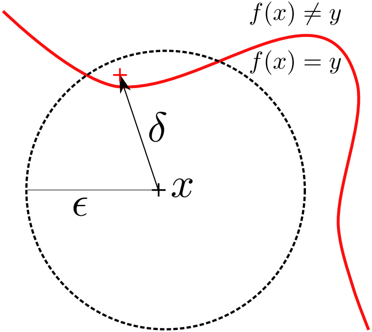


Untargeted attacks
$\|\delta\| \leq \epsilon$
$f(x+\delta) \neq y$

Targeted attacks
$\|\delta\| \leq \epsilon$
$f(x+\delta) = t, t \neq y$

Generating adversarial examples

Let $f : \mathbb{R}^n \to Y$ a classifier
Given an example $x \in \mathbb{R}^n$ and its true label $y \in Y$
find a $\delta \in \mathbb{R}^n$ such that:



Untargeted attacks
$\|\delta\| \leq \epsilon$
$f(x + \delta) \neq y$

Targeted attacks
$\|\delta\| \leq \epsilon$
$f(x + \delta) = t, t \neq y$

Most damaging perturbation:

$$\delta^* = \underset{\|\delta\| \leq \epsilon}{\arg\max} \ \ell_f(x + \delta, y)$$

Measuring the magnitude of perturbations

- **Using ℓ_2 norm**

$$\|\delta\|_2 \leq \epsilon \quad = \quad \sqrt{\sum_i \delta_i^2} \leq \epsilon$$

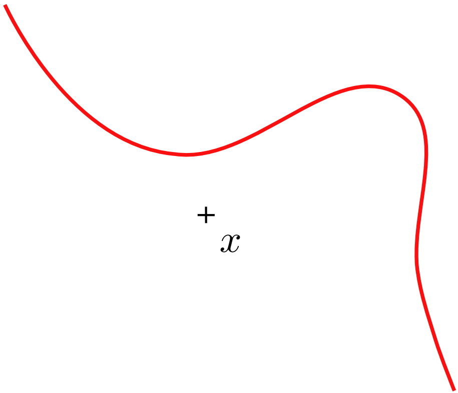
- Natural norm used in most loss functions.

- **Using ℓ_∞ norm**

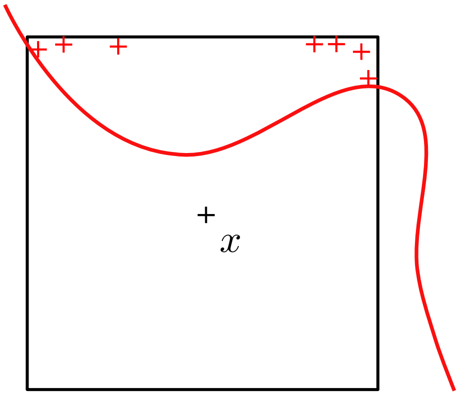
$$\|\delta\|_\infty \leq \epsilon \quad = \quad \max_i \delta_i \leq \epsilon$$

- Fits the human perception better when dealing with images.

ℓ_∞ Adversarial training

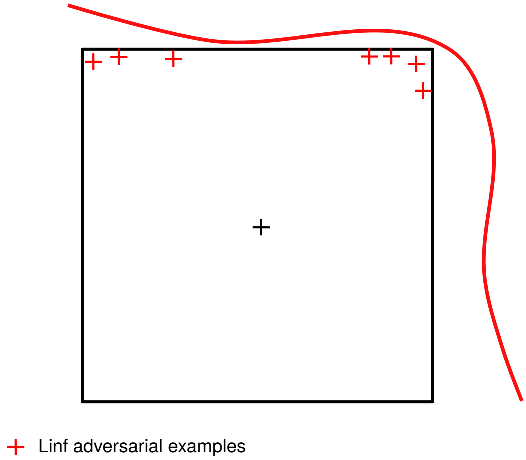


ℓ_∞ Adversarial training



$+$ Linf adversarial examples

ℓ_∞ Adversarial training



Linf adversarial examples

$$\forall \delta \text{ s.t. } \delta < \|\epsilon\|_\infty \quad f(x+\delta) = f(x)$$

Accuracy under attacks

Model	Natural examples	ℓ_∞ Attack
normal training	95%	0.8%
ℓ_∞ adv. training	high	40%

Outline

1. Principle of adversarial attacks
2. Attacks and defenses
   - FGSM attack
   - PGD attack
   - Carlini & Wagner attack (C&W)
3. Black box attacks
4. Approaches to defend against adversarial attacks
   - Adversarial training
   - Randomized networks
5. Projects

FGSM attack

Target function for ϵ-bounded attack:

$$\max_{||\delta|| \leq \epsilon} \ell_f(x + \delta, y)$$

FGSM attack

Target function for ϵ-bounded attack:

$$\max_{||\delta|| \leq \epsilon} \ell_f(x + \delta, y)$$

If ϵ is small, the optimization problem can be approximated using one gradient step:

$$\max_{||\delta|| \leq \epsilon} \delta^T \nabla_x \ell_f(x, y)$$

FGSM attack

Target function for ϵ-bounded attack:

$$\max_{||\delta|| \leq \epsilon} \ell_f(x + \delta, y)$$

If ϵ is small, the optimization problem can be approximated using one gradient step:

$$\max_{||\delta|| \leq \epsilon} \delta^T \nabla_x \ell_f(x, y)$$

If $||.|| = ||.||_\infty$, then:

$$\delta^* = \epsilon sign(\nabla_x \ell_f(x, y)$$

is a solution to the problem.
(FGSM attack (Goodfellow, 2015))

PGD attack

PGD attack (Madry, 2017) is an iterative version of FGSM:

$$x_0 = x$$

$$x_{t+1} = \Pi_{B(x_0,\epsilon)}(x_t + \delta sign(\nabla_x \ell_f(x, y)))$$

With

- Π: projection operator
- $B(x_0, \epsilon)$: hyperball centered in x_0 with radius ϵ

PGD attack

PGD attack (Madry, 2017) is an iterative version of FGSM:

$$x_0 = x$$

$$x_{t+1} = \Pi_{B(x_0,\epsilon)}(x_t + \delta sign(\nabla_x \ell_f(x, y)))$$

With

- Π: projection operator
- $B(x_0, \epsilon)$: hyperball centered in x_0 with radius ϵ

▶ Simple and very efficient bounded attack. Can be adapted to ℓ_1 and ℓ_2 constraints.

Carlini and Wagner attack

Norm bounded attack:

$$\min_{\ell_f(x+\delta,y)\geq\kappa} \|\delta\|$$

Carlini & Wagner solves the Lagrangian relaxation:

$$\min_{\delta} \|\delta\|_2 + \lambda \times g(x+\delta)$$

Where $g(x+\delta) < 0$ iff $\ell_f(x+\delta, y) \geq \kappa$

Carlini and Wagner attack

Norm bounded attack:

$$\min_{\ell_f(x+\delta,y)\geq\kappa} \|\delta\|$$

Carlini & Wagner solves the Lagrangian relaxation:

$$\min_{\delta} \|\delta\|_2 + \lambda \times g(x+\delta)$$

Where $g(x+\delta) < 0$ iff $\ell_f(x+\delta, y) \geq \kappa$

E.g.

$$g(x) = \max\left(f_c(x) - \max_{i\neq c}(f_i(x)), -\kappa\right)$$

- $f_i(x)$: i^{th} component of vector $f(x)$
- c: index of the actual class y of x

Outline

Black box methods

No access to the weights of the networks: access to the logits or only the labels. The goal is in most cases to estimate the gradient.

- Finite difference (Chen, 2017): Not very efficient, because it requires a huge number of queries.
- NES (Ilyas, 2018): Uses random directions instead of coordinate directions: simple and efficient
- Other methods bases on combinatorial optimization (Moon, 2019) and evolutionary strategies (Meunier, 2019).

Outline

1. Principle of adversarial attacks
2. Attacks and defenses
   - FGSM attack
   - PGD attack
   - Carlini & Wagner attack (C&W)
3. Black box attacks
4. Approaches to defend against adversarial attacks
   - Adversarial training
   - Randomized networks
5. Projects

Adversarial training

Train the network with this objective (Goodfellow, 2015):

$$\min_{\theta} \mathbb{E}_{(x,y)} \left(\max_{||\delta|| \leq \epsilon} L_{\theta}(x + \delta, y) \right)$$

In general, the inner maximization problem is solved with PGD or FGSM attack. This is so far the most efficient way to defend against adversarial attacks. There are no theoretical guarantees neither so far.

Randomized networks

(Lecuyer, 2018; Cohen, 2019; Pinot et al., 2019)
Inject noise at inference time to make the network robust. In practice, if x is the input and f a classifier returning the logits.
It is possible to show predicting

$$\mathbb{E}_\eta \left(f(x + \eta) \right)$$

brings more robustness than vanilla neural networks.

Outline

1. Principle of adversarial attacks
2. Attacks and defenses
   - FGSM attack
   - PGD attack
   - Carlini & Wagner attack (C&W)
3. Black box attacks
4. Approaches to defend against adversarial attacks
   - Adversarial training
   - Randomized networks
5. Projects

2-stage project

- Stage-1: (2 weeks)
  - Train a basic classifier
    - Dataset: CIFAR-10
    - Basic Architecture: (Conv+MaxPool+Conv+FC+FC+FC)
  - Implement attack mechanisms
    - FGSM
    - PGD
  - Implement Adversarial Training
- Stage-2: innovate
  - consider new defense mechanisms (e.g. randomized networks, lipschitz regularization, models robust against multiple defense mechanisms, etc. see refs)
  - consider new attack mechanisms
  - test and experiment

References

- Goodfellow,2015 (FGSM +Adverarial Training)
- Madry 2017 (PGD+Adversarial Training)
- Carlini & Wagner, 2017: Towards Evaluating the Robustnessof Neural Networks
- Athalye et al.: Obfuscated Gradients Give a False Sense of Security:Circumventing Defenses to Adversarial Examples
- Ilyas, 2018 (NES attack): Black-box Adversarial Attacks with Limited Queries and Information
- Randomized networks: Cohen, 2019: Certified Adversarial Robustness via Randomized Smoothing, Pinot,2019: Theoretical evidence for adversarial robustness through randomization
- Araujo et al.: Advocating for Multiple Defense Strategies against Adversarial Examples

Testing platform

https://www.lamsade.dauphine.fr/~testplatform/prds/