

# Adversarial examples and robustness certificates

Stéphane Canu, Ismaïla Seck & Gaëlle Loosli

<https://chaire-raimo.github.io/>



The banner features the word "SOPHIA" in large, bold, black letters, with a stylized blue "A" that has a neural network-like pattern inside its right side. Below "SOPHIA" is a black rectangular box containing the text "SPRINGBOARD FOR ARTIFICIAL INTELLIGENCE" and "EDITION 2020". To the right of the main title, there are four logos: "COMMUNAUTÉ D'AGGLOMERATION SOPHIA ANTIPOLIS" (with a grid of colored icons), "UNIVERSITÉ CÔTE D'AZUR" (with a blue circular logo), and "06 DÉPARTEMENT DES ALPES-MARITIMES" (with a green and blue "06" logo). At the bottom right, it says "18 · 19 · 20 NOV 20" and "SOPHIA-SUMMIT.COM". The background of the banner is a dark blue gradient with a subtle hexagonal grid pattern.

November 19, 2020

# Road map

1 Attacking deep learning

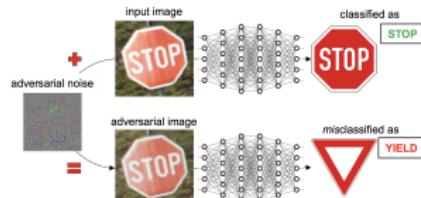
2 Adversarial attacks

3 Typology of Attacks

4 Robustness certificates and MIP

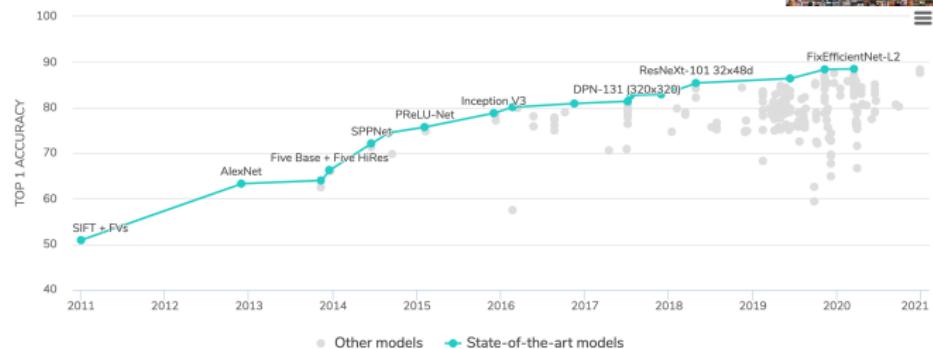
- Formalizing the search for adversarial examples
- Robust training

5 Conclusions

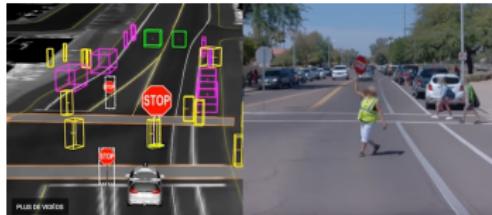


# The amazing achievements of deep learning

Image Classification on ImageNet



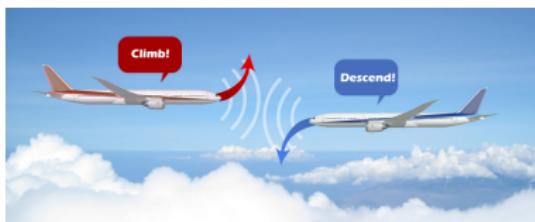
# Machine (deep) Learning in Safety-Critical Tasks



Autonomous Driving Vehicles



Facial Recognition Payment System



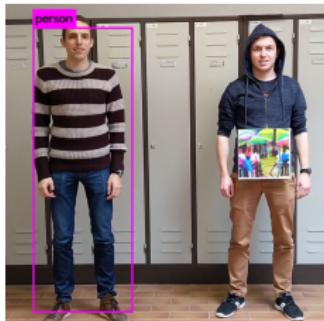
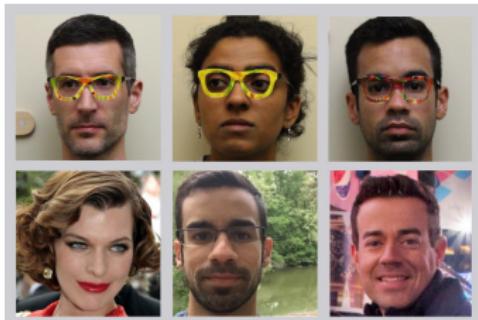
Airborne Collision-Avoidance System

Is ML Reliable and Safe for real-world applications?

# Example of recognition system under attacks



<https://www.kaggle.com/c/adversarial-attacks-against-spam-detectors/overview/description>  
Imani & Vassilakis, A Survey of Attacks Against Twitter Spam Detectors in an Adversarial Environment, 2019

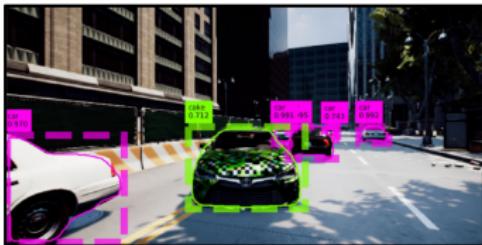


Sharif et al., ACM CCS, 2016  
Thys, Van Ranst & Toon Goedemé, Proceedings of the IEEE, 2019

# Attacks against autonomous vehicles



Eykholz et al., Robust Physical-World Attacks on Deep Learning Visual Classification, CVPR 2018



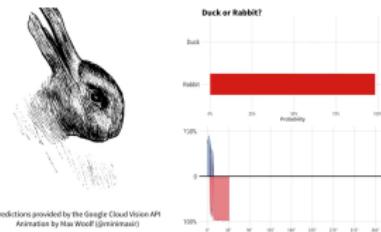
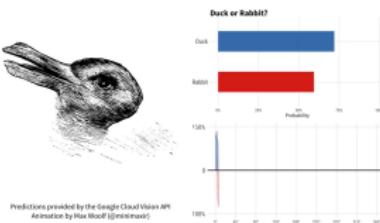
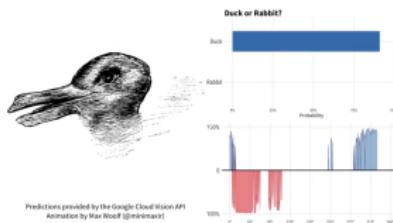
Zhang et al., CAMOU: Learning Physical Vehicle Camouflages to Adversarially Attack Detectors in the Wild, ICLR 2019



<https://www.mcafee.com/blogs/other-blogs/mcafee-labs/model-hacking-adas-to-pave-safer-roads-for-autonomous-vehicles/>  
Nassi et al., Phantom of the ADAS: Securing Advanced Driver-AssistanceSystems from Split-Second Phantom Attacks, 2020  
Qayyum, et al., Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial ML, IEEE Communications, 2019

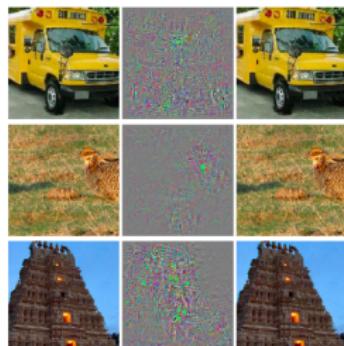
# Attack or illusion: Duck or a Rabbit?

## From Google Cloud Vision

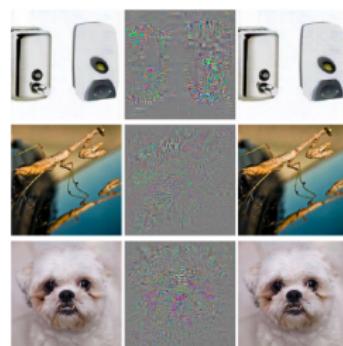


<https://github.com/minimaxir/optillusion-animation>

# Intriguing properties of neural networks, Szegedy ICLR 2014



(a)



(b)

Figure 5: Adversarial examples generated for AlexNet [9].(Left) is a correctly predicted sample, (center) difference between correct image, and image predicted incorrectly magnified by 10x (values shifted by 128 and clamped), (right) adversarial example. All images in the right column are predicted to be an “ostrich, *Struthio camelus*”. Average distortion based on 64 examples is 0.006508. Please refer to <http://goo.gl/huaGPb> for full resolution images. The examples are strictly randomly chosen. There is not any postselection involved.



Adversarial examples

# Road map

1 Attacking deep learning

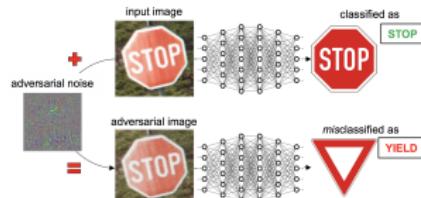
2 Adversarial attacks

3 Typology of Attacks

4 Robustness certificates and MIP

- Formalizing the search for adversarial examples
- Robust training

5 Conclusions



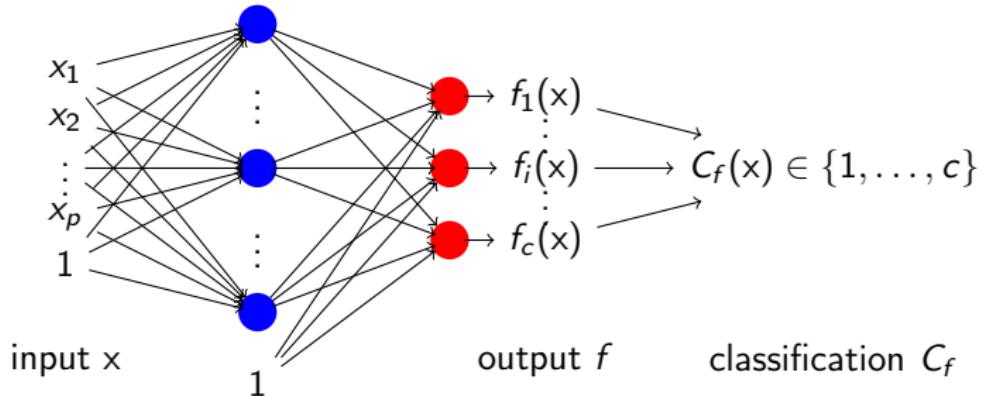
## Classification model

A classification model (e.g. Neural Network) with  $c$  output nodes

$$f : \begin{array}{c} \mathcal{X} \subseteq \mathbb{R}^p \\ x \end{array} \longrightarrow \begin{array}{c} \mathbb{R}^c \\ f(x) \end{array}$$

The associated classification (or decision function)

$$C_f(x) = \operatorname{argmax}_{k=1,\dots,c} f_k(x)$$

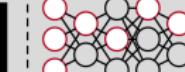
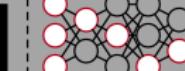


# Adversarial examples

## Definition (Generic adversarial)

$a_{f,x}$  is an adversarial example of  $f$  at  $x$  if  $a_{f,x}$  is a valid input close to  $x$  and

$$C_f(x) \neq C_f(a_{f,x}) \quad \text{that is} \quad c^* = \operatorname{argmax}_{k=1,\dots,c} f_k(x) \neq \operatorname{argmax}_{k=1,\dots,c} f_k(a_{f,x})$$

	Input	Model Activations	Output
$x$ Legitimate			1
$a_{f,x}$ Adversarial			4

from Papernot et al., 2016

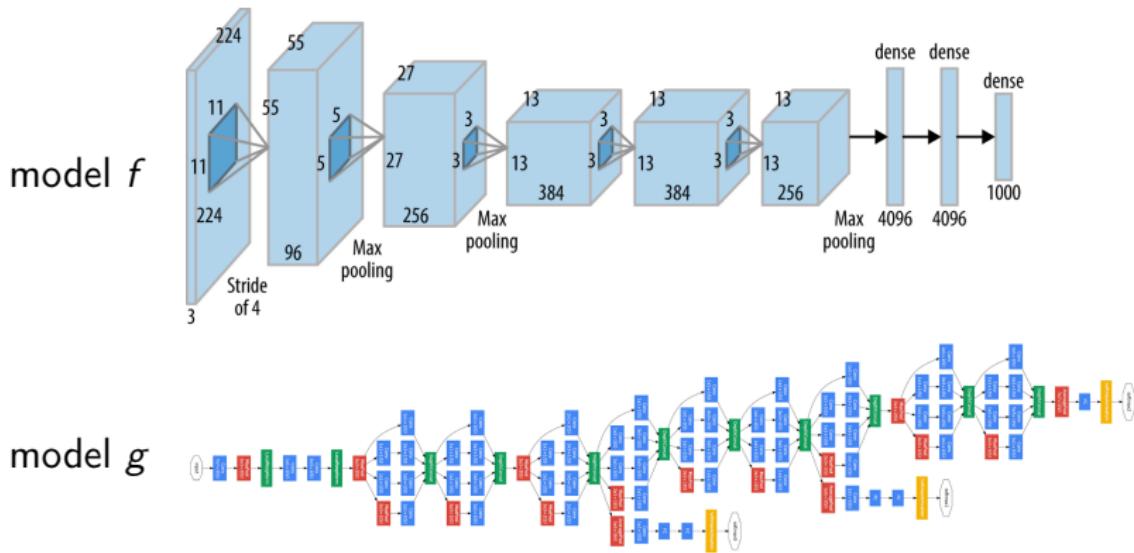
## Definition (Specific (or targeted) adversarial)

$a_{f,x,t_a}$  is a specific adversarial example of  $f$  at  $x$  for the adversarial targeted class  $t_a$  if  $a$  is a valid input close to  $x$  and

$$f_{c^*}(a_{f,x,t_a}) + \alpha \leq f_{t_a}(a_{f,x,t_a}) \quad \text{with} \quad c^* = \operatorname{argmax}_{k=1,\dots,c} f_k(x)$$

for a given scalar  $0 \leq \alpha$  called the confidence level.

# Adversarial transfer



## Definition (Adversarial transfer)

An adversarial example  $a_{f,x}$  of classification model  $f$  at input  $x$  adversarially transfers on model  $g$  if

$$C_g(x) \neq C_g(a_{f,x})$$

### 3 components to define adversarial examples

- a valid example  $a \in \mathcal{X}$  (feasible solution)
- adversarial close to  $x$ :  $D(x, a)$  a dissimilarity measure (a distance)
- $\underset{k=1, \dots, c}{\operatorname{argmax}} f_k(x) \neq \underset{k=1, \dots, c}{\operatorname{argmax}} f_k(a)$ : adversarial loss  $L$ ,

$$\begin{aligned} L : \quad \mathbb{R}^c \times \mathbb{R}^c &\longrightarrow \mathbb{R} \\ s, t &\longmapsto L(s, t) \end{aligned}$$

- ▶ training class:  $c^* = \underset{k=1, \dots, c}{\operatorname{argmax}} f_k(x)$  with training pair

$$(x, c^*) \Rightarrow \max L(s, c^*)$$

- ▶ targeted class:  $t_a \neq \underset{k=1, \dots, c}{\operatorname{argmax}} f_k(x) \Rightarrow \min L(s, t_a)$

May be different from the training loss  $L(f(a), c^*) \neq J(f(a), c^*)$

## Adversarial noise

Definition (adversarial noise (or perturbation or distortion))

A vector  $\Delta_{f,x}$  is an adversarial noise of  $f$  at  $x$  if

$$a_{f,x} = x + \Delta_{f,x}$$

is an adversarial example for  $f$  at  $x$

Given  $a_{f,x}$  the associated adversarial noise is  $\Delta_{f,x} = a_{f,x} - x$

Definition (**Universal** adversarial perturbation)

A perturbation  $\Delta_f$  is a universal of  $f$  if, **for any**  $x \in \mathcal{X}$ ,  $a_f = x + \Delta_f$  is a generic adversarial example for  $f$  at  $x$ , that is

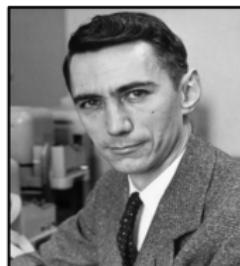
$$\forall x \in \mathcal{X}, \quad C_f(x) \neq C_f(x + \Delta_f)$$

note that  $x + \Delta_f$  must be a valid example.

This can be relaxed by asking to be verify with high probability when randomly picking  $x$

# Adversarial Noise vs. Stochastic Noise

This distinction is not new (*cf* Adversarial error in the Coding Theory)



**Shannon's stochastic noise model:** probabilistic model of the channel, the probability of occurrence of too many or too few errors is usually low



**Hamming's adversarial noise model:** the channel acts as an adversary that arbitrarily corrupts the code-word subject to a bound on the total number of errors

Noise is corrupting pattern, crafted to maximize the classification error  
It is an attack

# Road map

- 1 Attacking deep learning
- 2 Adversarial attacks
- 3 Typology of Attacks
- 4 Robustness certificates and MIP
  - Formalizing the search for adversarial examples
  - Robust training
- 5 Conclusions



## Threat Models

- Poisoning vs. Adversarial (evasion)

- Adversarial Goals:

$$a_{f,x} = x + \Delta_{f,x}$$

- ① Confidence reduction
  - ② Specific (targeted) misclassification attack: given class  $k$  and  $x$   $a_{f,x}$
  - ③ Generic (untargeted) misclassification: any class for a given  $x$   $a_{f,x}$
  - ④ Universal attack (generic misclassification) for any class any  $x$   $\Delta_f$
- 
- White-box (Perfect-Knowledge) vs. black-box (Zero-Knowledge)  
It can also be adaptive (or not)
  - Different ways: random search, gradient-based, transfer-based...

How can we produce (strong) adversarial examples?

## Generating adversarial examples in one step

Evasion Attacks against ML at Test Time Biggio, et al., ECML 2013

$$\left\{ \begin{array}{ll} \min_{a \in \mathcal{X}} & f_{c^*}(a) \\ \text{subject to} & \|x - a\| \leq \delta \end{array} \right. \quad (1)$$

One step projected gradient descent ( $\rho$  large enough)

$$a_{f,x} = \text{Proj}_{\mathcal{A}_x}(x - \rho \nabla_x f_i(x)) \quad \text{with} \quad \mathcal{A}_x = \{a \in \mathcal{X} \mid \|x - a\| \leq \delta\}$$

## Fast Gradient Sign Method (FGSM), (I. Goodfellow et al, ICLR 2015)

The problem, given  $(x, t)$

$$\left\{ \begin{array}{ll} \max_{a \in \mathcal{X}} & J(f(a), t) \\ \text{subject to} & \|x - a\| \leq \delta \end{array} \right. \quad \text{training loss}$$

Fast Gradient Sign Method (FGSM) ( $\rho = \frac{1}{4}, .1$  or  $.007$ )

$$a = x + \rho \text{sign}(\nabla_x J(f(x), t))$$

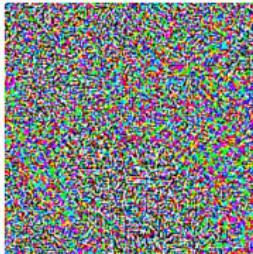
# Fast Gradient Sign Method (FGSM)

$$a = x + \rho \operatorname{sign}(\nabla_x J(f(x), t))$$



$x$   
“panda”  
57.7% confidence

+ .007 ×



$\operatorname{sign}(\nabla_x J(\theta, x, y))$   
“nematode”  
8.2% confidence

=



$x +$   
 $\epsilon \operatorname{sign}(\nabla_x J(\theta, x, y))$   
“gibbon”  
99.3 % confidence

## Specific Optimization formulation

Specific adversarial for some ( $t_a \neq c^*$ ),  $t_a$  being the adversarial target

The problem:

$$\left\{ \begin{array}{l} \min_{a \in \mathcal{X}} J(f(a), t_a) \\ \text{subject to } \|x - a\| \leq \delta \end{array} \right. \quad \left\{ \begin{array}{l} \min_{a \in \mathcal{X}} \|x - a\| \\ \text{subject to } f_{c^*}(a) + \alpha \leq f_{t_a}(a) \end{array} \right.$$

$J(f(a), t_a)$  training loss

The proposed solution: lagrangian form (not convex/not equivalent)

$$\min_{a \in \mathcal{X}} L(f(a), t_a) + \lambda \|x - a\|$$

Solved using a box-constrained L-BFGS (with  $\mathcal{X} = [0, 1]^p$ )

## Multi-step (iterative) approach

### Iterative FGSM, (PGD) (Kurakin et al, ICLR 2017)

The problem, given  $(x, c^*)$        $\begin{cases} \max_{a \in \mathcal{X}} J(f(a), c^*) & \text{training loss} \\ \text{subject to} & \|x - a\| \leq \delta \end{cases}$

The i-FGSM (PGD) proposed solution: build a sequence with (small)  $\rho_i$

$$\begin{cases} a_0 = x \\ a_{i+1} = \text{Proj}_{\mathcal{A}_x} \left( a_i + \rho_i \text{ sign} \left( \nabla_x J(f(a_i), c^*) \right) \right) \end{cases}$$

- $\rho$  chosen to change the value of each pixel only by 1 on each step
- due to the non concavity, it only converges towards local maxima
- i-FGSM is equivalent to (the  $\ell_\infty$  version of) Projected Gradient Descent (PGD), Madry et al., ICLR 2018 (sign?)
- Specific version: with  $c^*$  the target class

$$a_{i+1} = \text{Proj}_{\mathcal{A}_x} \left( a_i - \rho_i \text{ sign} \left( \nabla_x J(f(a_i), c^*) \right) \right)$$

## Optimization attack: Carlini & Wagner (CW), 2017

Specific attack: Given  $x$  and  $t_a \neq c^*$   $\left\{ \begin{array}{ll} \min_{a \in \mathcal{X}} & D(x, a) \\ \text{subject to} & C_f(a) = t_a \end{array} \right.$

Define an objective function  $L$  such that  $C_f(a) = t_a$  iff  $L(f(a), t_a) \leq 0$

$$\left\{ \begin{array}{ll} \min_{a \in \mathcal{X}} & D(x, a) \\ \text{subject to} & L(f(a), t_a) \leq 0 \end{array} \right. \quad \min_{a \in \mathcal{X}} D(x, a) + \lambda L(f(a), t_a)$$

- stochastic gradient descent solver (SGD is slow, use GPU)
- compare 3  $D(x, a) = \|x - a\|_p^p$ ,  $\ell_2$ ,  $\ell_0$  and  $\ell_\infty$  attacks
- compare 7 objective function  $L$

and the winner is the Carlini & Wagner  $\ell_2$  attack

Euclidean distance  $\ell_2$  and hinge loss (with confidence  $\alpha$ )

$$L(f(a), t_a) = \max[\alpha - (f_{t_a}(a) - \max_{k \neq t_a} f_k(a)), 0]$$

## Carlini & Wagner hinge loss details and variants

$$\left\{ \begin{array}{l} \min_{a \in \mathcal{X}} \|x - a\|_2^2 \\ \text{subject to } C_f(a) = t_a \end{array} \right. \quad \left\{ \begin{array}{l} \min_{a \in \mathcal{X}} \|x - a\|_2^2 \\ \text{subject to } f_{t_a}(a) \geq \max_{k \neq t_a} f_k(a) + \alpha \end{array} \right.$$

Multiclass hinge loss similar to Crammer and Singer (for SVM experts)

$$\min_{a \in \mathcal{X}} \frac{1}{2} \|x - a\|_2^2 + \lambda \max \left[ \alpha - (f_{t_a}(a) - \max_{k \neq t_a} f_k(a)), 0 \right]$$

Generic variant

$$\min_{a \in \mathcal{X}} \frac{1}{2} \|x - a\|_2^2 + \lambda \max \left[ \alpha - (\max_{k \neq c^*} f_k(a) - f_{c^*}(a)), 0 \right]$$

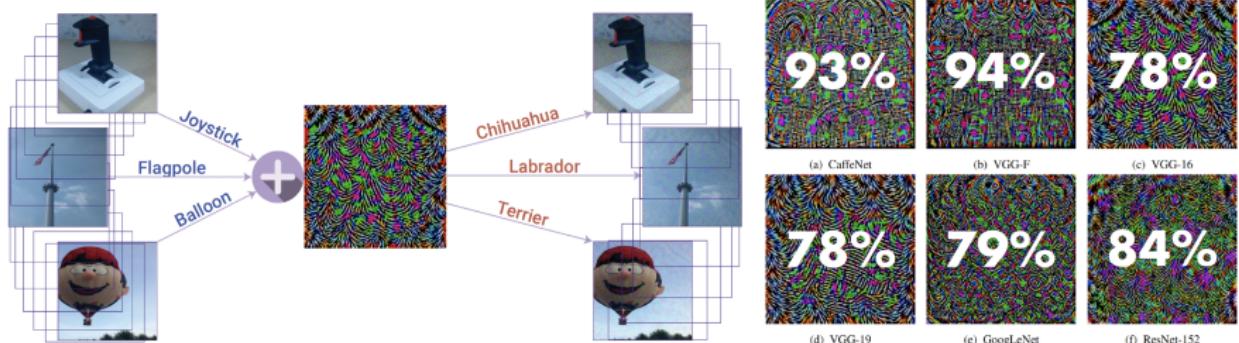
# Universal Adversarial Perturbations

Given  $f$ , find  $\Delta$  small s.t. for "most"  $(x, c^*) \max_{k \neq c^*} f_k(x + \Delta) > f_{c^*}(x + \Delta)$

The problem:

$$\left\{ \begin{array}{ll} \min_{\Delta} & L(f(a), t) = \text{IP} \left( \max_{k \neq c^*} f_k(x + \Delta) > f_{c^*}(x + \Delta) \right) \\ \text{subject to} & \|\Delta\|_p \leq \delta \\ & x + \Delta \in \mathcal{X} \end{array} \right.$$

The proposed solution: Lagrangian formulation + SGD on minibach



# Comparizon of some attacking methods:

**TABLE 1.** Summary of the attributes of diverse attacking methods: The ‘perturbation norm’ indicates the restricted  $\ell_p$ -norm of the perturbations to make them imperceptible. The strength (higher for more asterisks) is based on the impression from the reviewed literature.

Method	Black/White box	Targeted/Non-targeted	Image-specific/Universal	Perturbation norm	Learning	Strength
L-BFGS [22]	White box	Targeted	Image specific	$\ell_\infty$	One shot	***
FGSM [23]	White box	Targeted	Image specific	$\ell_\infty$	One shot	***
BIM & ILCM [35]	White box	Non targeted	Image specific	$\ell_\infty$	Iterative	****
JSMA [60]	White box	Targeted	Image specific	$\ell_0$	Iterative	***
One-pixel [68]	Black box	Non Targeted	Image specific	$\ell_0$	Iterative	**
C&W attacks [36]	White box	Targeted	Image specific	$\ell_0, \ell_2, \ell_\infty$	Iterative	*****
DeepFool [72]	White box	Non targeted	Image specific	$\ell_2, \ell_\infty$	Iterative	****
Universal perturbations [16]	White box	Non targeted	Universal	$\ell_2, \ell_\infty$	Iterative	*****
UPSET [146]	Black box	Targeted	Universal	$\ell_\infty$	Iterative	****
ANGRI [146]	Black box	Targeted	Image specific	$\ell_\infty$	Iterative	****
Houdini [131]	Black box	Targeted	Image specific	$\ell_2, \ell_\infty$	Iterative	****
ATNs [42]	White box	Targeted	Image specific	$\ell_\infty$	Iterative	****

Akhtar & Mian, Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey, 2018

Most popular attack algorithms (strong first order attacks):

- $\ell_\infty$ : PGD (Madry et al)
- $\ell_2$ : CW (Carlini & Wagner)
- $\ell_0$ :

Popular software: Cleverhans and Adversarial Robustness Toolbox (ART)



Python library for  
Adversarial attacks



# Road map

1 Attacking deep learning

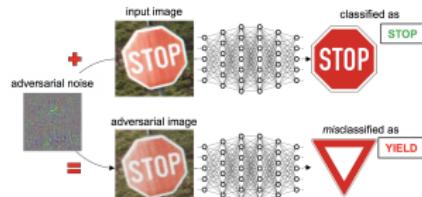
2 Adversarial attacks

3 Typology of Attacks

4 Robustness certificates and MIP

- Formalizing the search for adversarial examples
- Robust training

5 Conclusions



## 3 formal ways to search for adversarial examples

- ① Minimizing the Adversarial Distortion (Bunel et al., NeurIPS 2018)

$$\left\{ \begin{array}{l} \min_{a \in \mathcal{X}} D(x, a) = \|x - a\| \\ \text{subject to } L(f(x), f(a)) \geq \alpha = \max_{k \neq c^*} f_k(a) > f_{c^*}(a) + \alpha \end{array} \right. \quad (2)$$

- ② Maximizing the adversarial loss (Wong & Kolter, ICML 2018)

$$\left\{ \begin{array}{l} \max_{a \in \mathcal{X}} L(f(x), f(a)) = f_{t_a}(a) - f_{c^*}(a) \\ \text{subject to } D(x, a) \leq \delta = \|x - a\| \leq \delta \end{array} \right. \quad (3)$$

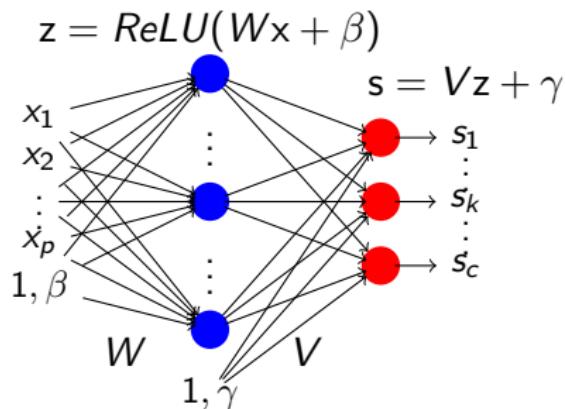
- ③ Robustness as a verification problem (Katz et al, CAV, 2017)

A classifier  $f$  is robust to perturbations on  $x$  if and only if:

$$\forall a \in \mathcal{A}_x, (s = f(a)) \implies \mathcal{P}(s)$$
$$\mathcal{A}_x = \{a \in \mathcal{X} \mid D(x, a) \leq \delta\} \quad \mathcal{P}(s) = \max_{k \neq c^*} s_k < s_{c^*} \quad (4)$$

Positive answer (SAT) includes a counter example (adversarial)

## The particular case of a one hidden layer MLP



The Neural Network function  $f$  with  $c$  output nodes

$$\begin{aligned} z &= \text{ReLU}(Wx + \beta) \\ f(x) &= Vz + \gamma \end{aligned}$$

$$\begin{aligned} h &= Wx + \beta, \\ z &= \max(h, 0) \\ f(x) &= s = Vz + \gamma \end{aligned}$$

The associated classification (or decision function)

$$C_f(x) = \underset{k=1, \dots, c}{\operatorname{argmax}} s_k$$

## Formal verification as an optimization problem

### ① Minimizing the Adversarial Distortion

$$\left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\| \\ \text{subject to} & \max_{k \neq c^*} f_k(a) > f_{c^*}(a) \end{array} \right\} \quad \left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\|^2 \\ \text{subject to} & h = Wa + \beta \\ & z = \max(h, 0) \\ & s = Vz + \gamma \\ & \max_{k \neq c^*} s_k > s_{c^*} \end{array} \right.$$

### ② Maximizing the adversarial loss

$$\left\{ \begin{array}{ll} \max_{a \in [0,1]^p} & s_{t_a} - s_{c^*} = e_{t_a, c^*}^\top (Vz + \gamma) \\ \text{subject to} & h = Wa + \beta, \\ & z = \max(h, 0), \\ & s = Vz + \gamma \\ & \|x - a\| \leq \delta \end{array} \right.$$

### ③ Use satisfiability modulo theories (SAT/SMT) constraints

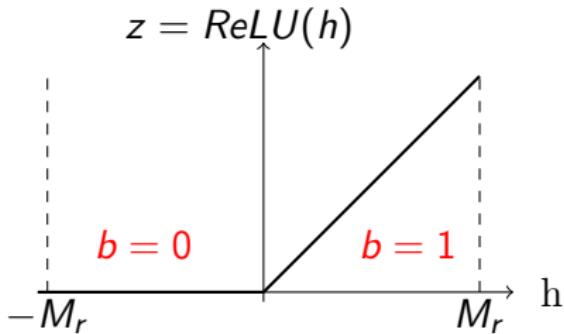
# The ReLUplex (Lomuscio & Maganti, Katz et al., 2017)

the ReLU can be formulated as a set of linear constraints

Given  $M_r \geq \|h\|_\infty$  and binary variables  $b \in \{0, 1\}^e$

$$z = \max(h, 0) \quad \Leftrightarrow \quad \begin{aligned} z_i &\geq 0, & i &= 1, \dots, e \\ z_i &\leq M_r b_i, & i &= 1, \dots, e \\ z_i &\leq h_i + M_r(1 - b_i), & i &= 1, \dots, e \\ z_i &\geq h_i, & i &= 1, \dots, e \end{aligned}$$

$$\begin{aligned} b_i = 0 &\Leftrightarrow z_i = 0 \\ b_i = 1 &\Leftrightarrow z_i = h_i \geq 0 \end{aligned}$$



## Exact search for adversarial examples as a MIP

Thanks to the ReLUplex,

$$\left\{ \begin{array}{ll} \min_{a \in [0,1]^p} & \|x - a\|_p^p \\ \text{subject to} & h = Wa + \beta \\ & z = \max(h, 0) \\ & s = Vz + \gamma \\ & \max_{i \neq i^*} s_i > s_{i^*} \end{array} \right\} \left\{ \begin{array}{ll} \min_{\substack{a \in [0,1]^p, \\ b \in \{0,1\}^e}} & \|x - a\|_p^p \\ \text{subject to} & h = Wa + \beta \\ & z_i \geq 0 & i = 1, \dots, e \\ & z_i \leq M_r b_i & i = 1, \dots, e \\ & z_i \leq h_i + M_r(1 - b_i) & i = 1, \dots, e \\ & z_i \geq h_i & i = 1, \dots, e \\ & s = Vz + \gamma \\ & \max_{i \neq i^*} s_i > s_{i^*} \end{array} \right\}$$

$\|x - a\|_\infty, \|x - a\|_1$  MILP

$\|x - a\|_2^2$  MIQP

$\|x - a\|_0$  MILP with more binary variables

→ max, convolution, pooling can also be linearized

# Mixed integer linear program (MILP)

- linear cost
- linear constraints
- **integer** and continuous variables

Definition (mixed integer linear program – MILP (canonical form))

$$\left\{ \begin{array}{ll} \min_{a \in \mathbb{R}^p, b \in \mathbb{N}^q} & J(a, b) = w^t a + d^t b \quad \leftarrow \text{linear} \\ \text{s.t.} & Aw + Bz \leq c \quad \leftarrow \text{linear} \\ & w \geq 0, \end{array} \right.$$

for some given  $w \in \mathbb{R}^p$ ,  $c \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{m \times p}$ ,  $B \in \mathbb{R}^{m \times q}$  and  $d \in \mathbb{R}^q$ .

- A **mixed binary linear program** is a MILP with  $b \in \{0, 1\}^q$  binary.
- When its domain is not empty and bounded, a MILP admit a unique global minimum.

## Mixed integer quadratic program (MIQP)

- quadratic cost
- linear constraints
- integer and continuous variables

### Definition (mixed integer quadratic program – MIQP)

$$\left\{ \begin{array}{ll} \min_{x=(a,b) \in \mathbb{R}^p \times \mathbb{N}^q} & f(x) = \frac{1}{2} x^t Q x + c^t x \quad \leftarrow \text{quadratic} \\ \text{s.t.} & Ax \leq b \quad \leftarrow \text{linear} \\ & x \geq 0, \end{array} \right.$$

for some given symmetric matrix  $Q \in \mathbb{R}^{(p+q) \times (p+q)}$

### Mixed integer quadratically constrained quadratic program (MIQCP).

- quadratic cost, quadratic constraints, integer and continuous variables

### Problems hierarchy

$$\text{MILP} \text{ (= MIQP with } Q = 0) \subset \text{MIQP} \subset \text{MIQCP}$$

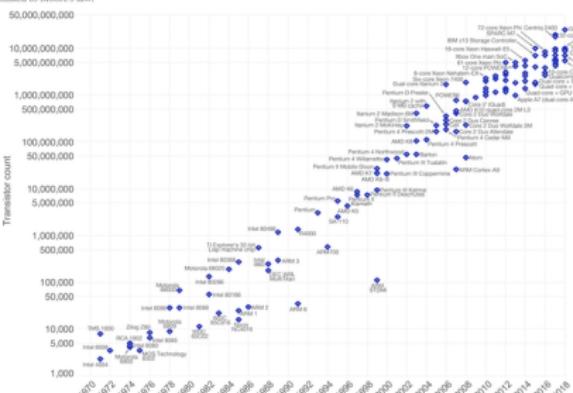
# Progresses in MILP

in 1989

MILP is a powerful modeling tool, “They are, however, theoretically complicated and computationally **cumbersome**”

Moore's Law – The number of transistors on integrated circuit chips (1971–2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.  
Measurement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia ([https://en.wikipedia.org/w/index.php?title=Moore%27s\\_law&oldid=90000000](https://en.wikipedia.org/w/index.php?title=Moore%27s_law&oldid=90000000))  
The data visualization is available at OurWorldInData.org. There you find more visualizations and research on this topic.

Our World  
in Data

from 1998 to 2018

	improvement factor
machine solver	$\times 2^{10} = 1000 - 1600$
formulation	$\times 1000 - 3600$
global	???
	$\times 1 - 5 \cdot 10^6$

a year to solve 10 – 20 years ago → now 30 seconds

“mixed integer linear techniques are nowadays **mature**, that is fast, robust, and are able to solve problems with up to millions of variables”

# Mixed integer software (available with python)

## Software package

---

### Open source

GLPK glpk for mixed integer linear programming

LP\_Solve

ECOS\_BB

### Commercial

CVXpy

CPLEX

GUROBI

Mosek

NAS

(with academic license)

cvx for mixed integer linear programming

cplexmilp for mixed integer linear programming

cplexmiqp for mixed integer quadratic programming

cplexmiqcp for mixed integer quadratically constrained pg

gurobi for MILP, MIQP and MIQCQP

mosekopt for MILP, MIQP and MIQCQP

NAS for MILP, MIQP and MIQCQP

---

## Mixed Integer Linear Programming Benchmark (MIPLIB2017)

recommend CVXpy, CPLEX, GUROBI and NAS

## MIP, lower bound & upper bound

$$\left\{ \begin{array}{ll} \min_{\substack{\mathbf{a} \in [0,1]^p, \\ \mathbf{b} \in \{0,1\}^e}} & \|\mathbf{x} - \mathbf{a}\|_p^p \\ \text{subject to} & \mathbf{h} = W\mathbf{a} + \beta \\ & \mathbf{z}_i \geq 0, \mathbf{z}_i \geq M_r \\ & \mathbf{z}_i \leq M_r b_i, \mathbf{z}_i \leq \mathbf{h}_i + M_r(1 - b_i) \\ & \mathbf{e}^\top (\mathbf{V}\mathbf{z} + \gamma) \geq \alpha \end{array} \right.$$

Lower bound: continuous relaxation

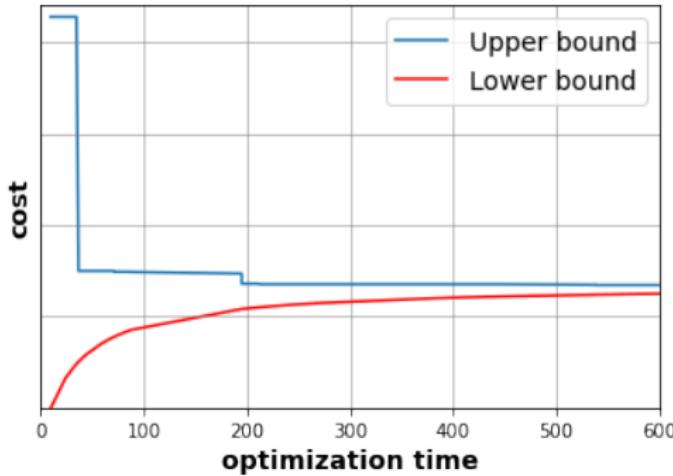
$$\left\{ \begin{array}{ll} \min_{\substack{\mathbf{a} \in [0,1]^p, \\ \mathbf{b} \in [0,1]^e}} & \|\mathbf{x} - \mathbf{a}\|_p^p \\ \text{subject to} & \mathbf{h} = W\mathbf{a} + \beta \\ & \mathbf{z}_i \geq 0, M_r \\ & \mathbf{z}_i \leq M_r b_i, \mathbf{h}_i + M_r(1 - b_i) \\ & \mathbf{e}^\top (\mathbf{V}\mathbf{z} + \gamma) \geq \alpha \end{array} \right.$$

Upper bound: fix  $b$  (feasible)

$$\left\{ \begin{array}{ll} \min_{\substack{\mathbf{a} \in [0,1]^p, \\ \mathbf{b} \in \{0,1\}^e}} & \|\mathbf{x} - \mathbf{a}\|_p^p \\ \text{subject to} & \mathbf{h} = W\mathbf{a} + \beta \\ & \mathbf{z}_i \geq 0, M_r \\ & \mathbf{z}_i \leq M_r b_i, \mathbf{h}_i + M_r(1 - b_i) \\ & \mathbf{e}^\top (\mathbf{V}\mathbf{z} + \gamma) \geq \alpha \end{array} \right.$$

## MIP, Upper bound & Lower bound

$$\|x - a_{lb}\|_p^p \leq \|x - a_{x,f}^*\|_p^p \leq \|x - a_{ub}\|_p^p$$



- Optimality:
  - ▶ it may be "easy" to find the optimal solution...
  - ▶ ...and very hard to prove it
  - ▶
- Computational efficiency: how to manage your time budget?
  - ▶ initialization
  - ▶ acceleration through stronger relaxation

## MIP acceleration using asymmetric bounds

$$\begin{array}{c|c|c} & \begin{matrix} 1.1 \\ 2.8 \\ -0.2 \\ 0.9 \\ -2.2 \end{matrix} & = & \begin{matrix} 1.1 \\ 2.8 \\ 0 \\ 0.9 \\ 0 \end{matrix} & - & \begin{matrix} 0 \\ 0 \\ 0.2 \\ 0 \\ 2.2 \end{matrix} \\ w & \hline & w_+ & - & w_- & \end{array}$$

$$\ell \leq a \leq u \text{ & } h = w^\top a + \beta \Rightarrow \underbrace{w_+^\top \ell - w_-^\top u + \beta}_{\ell'} \leq h \leq \underbrace{w_+^\top u - w_-^\top \ell + \beta}_{u'}$$

Pre computing binary variables: if  $0 \leq \ell'_i$  then  $b_i = 1$   
if  $u'_i \leq 0$  then  $b_i = 0$

Non symmetric bound (ReLU)

$$\begin{aligned} z_i &\geq 0, & i &= 1, \dots, e \\ z_i &\leq u' b_i, & i &= 1, \dots, e \\ z_i &\leq h_i - \ell'(1 - b_i), & i &= 1, \dots, e \\ z_i &\geq h_i, & i &= 1, \dots, e \end{aligned}$$

# MIPVerify (Julia package + Gurobi)

## Finding an Adversarial Example

We now try to find the closest  $L_{\infty}$  norm adversarial example to the first image, setting the target category as index 10 (corresponding to a true label of 9). Note that we restrict the search space to a distance of 0.05 around the original image via the specified `pp`.

```
In [12]: target_label_index = 10
d = MIPVerify.find_adversarial_example(
    n1,
    sample_image,
    target_label_index,
    Gurobi.Optimizer,
    Dict(),
    norm_order = Inf,
    pp=MIPVerify.LINfNormBoundedPerturbationFamily(0.05)
)

Academic license - for non-commercial use only
[notice | MIPVerify]: Attempting to find adversarial example. Neural net predicted label is 8, target labels are [
[notice | MIPVerify]: Determining upper and lower bounds for the input to each non-linear unit.

    Calculating upper bounds: 100% |██████████| Time: 0:00:00

Academic license - for non-commercial use only

    Calculating lower bounds: 100% |██████████| Time: 0:00:00
    Imposing relu constraint: 100% |██████████| Time: 0:00:00
    Calculating upper bounds: 10% |██████████| ETA: 0:02:41

Academic license - for non-commercial use only

    Calculating upper bounds: 100% |██████████| Time: 0:00:26
    Calculating lower bounds: 100% |██████████| Time: 0:00:08
    Imposing relu constraint: 100% |██████████| Time: 0:00:00

Academic license - for non-commercial use only
```

## Robust training

- Adversarial robustness error:  $\text{IP}_{(X, T)}(\exists \mathbf{a}_{f, X} \in \mathcal{A}_x \mid C(\mathbf{a}_{f, X}) \neq T)$   
with  $\mathcal{A}_x = \{\mathbf{a} \in \mathcal{X} \mid D(x, \mathbf{a}) \leq \delta\}$
- Distance to error set:  $\text{IE}_{(X, T)} \min_{\mathbf{a}_{f, X} \in \mathcal{B}_x} D(X, \mathbf{a}_{f, X})$   
with  $\mathcal{B}_x = \{\mathbf{a} \in \mathcal{X} \mid C(\mathbf{a}_{f, X}) \neq T\}$
- How can we train deep neural networks robust to adversarial inputs?
$$\min_f \text{IE}_{(X, T)} \left[ \max_{\Delta \in \mathcal{A}_x} L(f(X + \Delta), T) \right]$$
  - ▶ Long history in robust optimization, going back to Wald
  - ▶ Towards deep learning models resistant to adversarial A., Madry, 2019
- Adversarial robustness is impossible in general, Dohmatob, ICML 2019

# Road map

1 Attacking deep learning

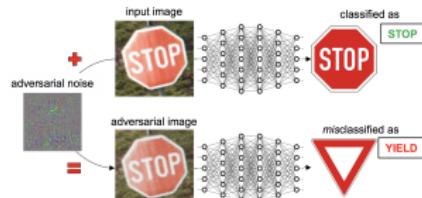
2 Adversarial attacks

3 Typology of Attacks

4 Robustness certificates and MIP

- Formalizing the search for adversarial examples
- Robust training

5 Conclusions



## Conclusion

- Deep networks can be (and will be) attacked
- The problem can be formalized as a MIP (NP hard)
  - ▶ looking for a formal solution
- Improve the model (Wasserstein distance, Wong et al ICML 2019)
  - ▶ improve the solver
- Think about proofs
  - ▶ Robustness certificate
  - ▶ Are adversarial examples inevitable? A. Shafahi et al, ICLR 2019.
  - ▶ Limits on robustness to adversarial examples, E. Dohmatob, ICML 2019
- Think about defenses: change training

# Some links

- Cleverhans  
<http://www.cleverhans.io/>
- Adversarial Robustness Toolbox (ART)  
<https://adversarial-robustness-toolbox.readthedocs.io/en/stable/>
- Robust ML  
<https://www.robust-ml.org/defenses>
- A (Complete) List of All (arXiv) Adversarial Example Papers by N. Carlini  
<nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html>
- ForMaL: DigiCosme Spring School on Formal Methods and Machine Learning 4th-7th June 2019, ENS Paris-Saclay, Cachan, France  
<https://formal-paris-saclay.fr/>
- NeurIPS 2018 tutorial, “Adversarial Robustness: Theory and Practice”, by Zico Kolter and Aleksander Madry  
<https://adversarial-ml-tutorial.org/>
- Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey Silva & Najafirad, submitted to IEEE Transactions on Knowledge and Data Engineering, 2020  
<https://arxiv.org/abs/2007.00753>

