Adversarial Attacks

Deep Learning and Neural Networks: Advanced Topics

Fabio Brau

March 21, 2022

SSSA, Emerging Digital Technologies, Pisa.

Introduction, Definitions and Taxonomy

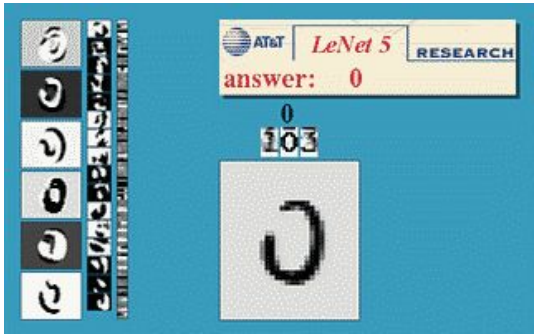
Adversarial Examples Generation

Defense Mechanisms

Conclusions

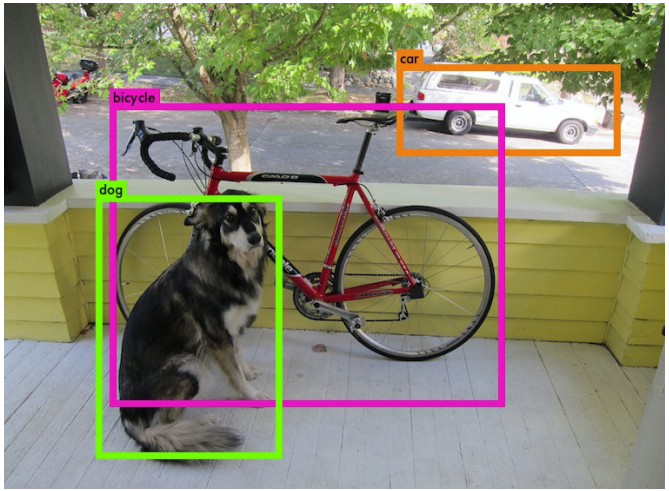
Neural Networks achieve Human Performances...

LeNet-5 (MNIST), 1998



AlexNet (ImagNet), 2013

Yolo, 2016



Neural Networks become so powerful that they reach human capabilities in the classification and object detection tasks

… but they make weird mistakes.

Evading Spam Detection (2010)

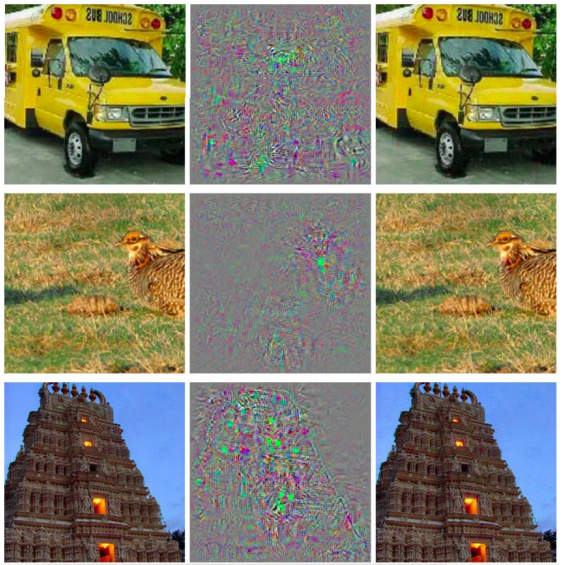
FIRECRACKER STOCK ALERT!!!
MONDAY, JULY 3, 2006
Waypoint Biomedical, Inc
Symbol: WYPH
Price: $ 17

Sometimes these little stocks can make gains in days or weeks accomplish, if at all! Know what we mean? You be the judge!

RECENT HEALINES:

Spam Detector based on OCR

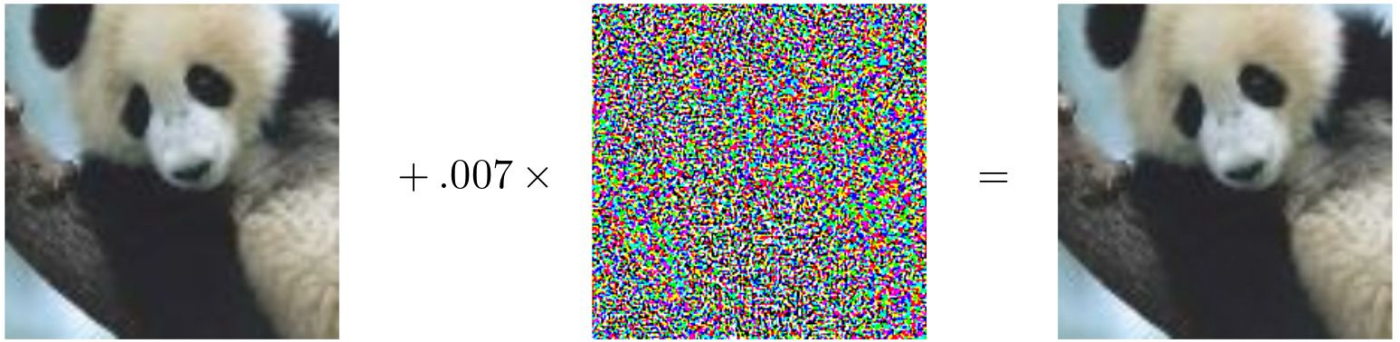
Attacks on Images (2014)



Perturbed Images are no long properly recognized by the network

Delvi et al. “Adversarial Classification” (2004)
Battista Biggio et al. ”Adversarial Pattern Classification” (2010)
Christian Szegedy et al. “Intriguing properties of neural networks” (2013)

Adversarial Examples



panda 57.7%

gibbon 99.3%

Adversarial examples are inputs **intentionally** designed to cause ML models to make **errors**.

They can be created so that can appear **nearly identical** to normal inputs.

Goodfellow et al. “Explaining and Harnessing Adversarial Examples” (2014)

Adversarial Examples

Speech-to-Text models

Can be fooled by adding an imperceptible noise to input audio

“without the dataset the article is useless”

Adversarial Noise (30dB)

“okay google browse to evil dot com”

Carlini et al. “Audio Adversarial Examples: Targeted Attacks on Speech-to-Text” (2018)

Adversarial Examples become popular

WRITING

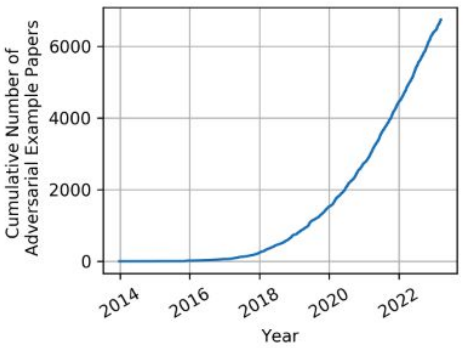
MAIN PAPERS TALKS CODE WRITING

A Complete List of All (arXiv) Adversarial Example Papers

by **Nicholas Carlini** 2019-06-15

It can be hard to stay up-to-date on the published papers in the field of adversarial examples, where we have seen massive growth in the number of papers written each year. I have been somewhat religiously keeping track of these papers for the last few years, and realized it may be helpful for others to release this list.

The only requirement I used for selecting papers for this list is that it is primarily a paper about adversarial examples, or extensively uses adversarial examples. Due to the sheer quantity of papers, I can't guarantee that I actually have found *all* of them.



link to the website

Why are adversarial Examples worth studying?

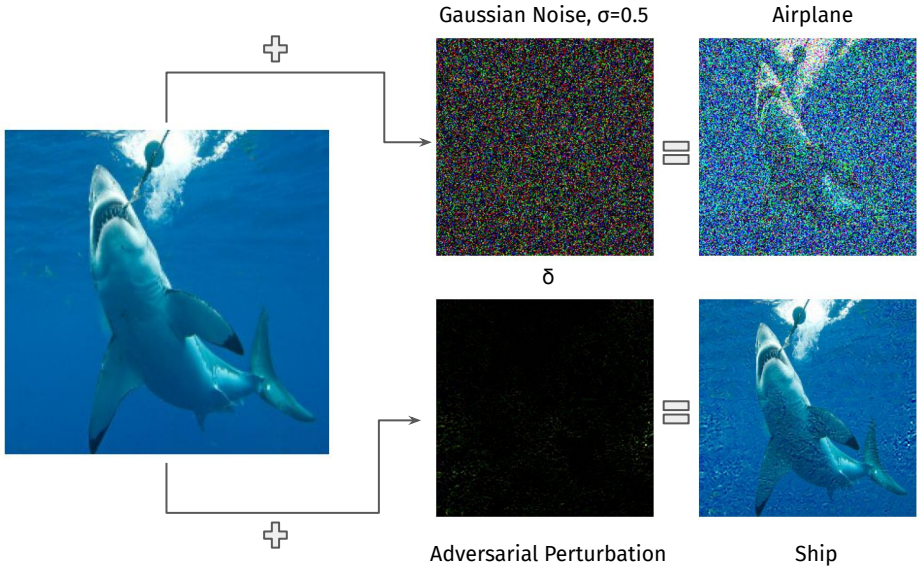
Security Threat (Known your Enemy)

Neural Networks are involved in a growing number of tasks. How can we fill safe?

(a) Input 1 (b) Input 2 (darker version of 1)

DeepExplore (2021). Erroneous behaviour of DAVE-2. A tiny change in the light condition causes the model turns right.

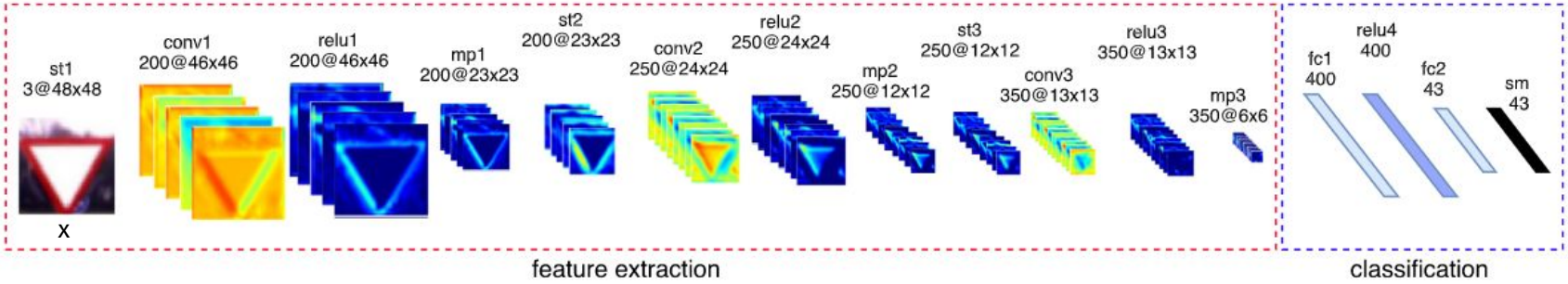
Foundational Perspective

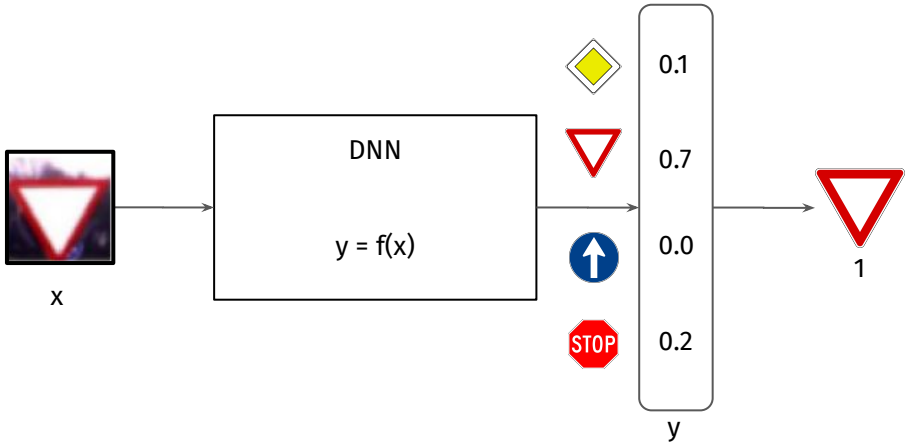


Small maliciously perturbations are sufficient to fool the model, why gaussian noise has not this property?

Definition and Taxonomy

The Classification Problem



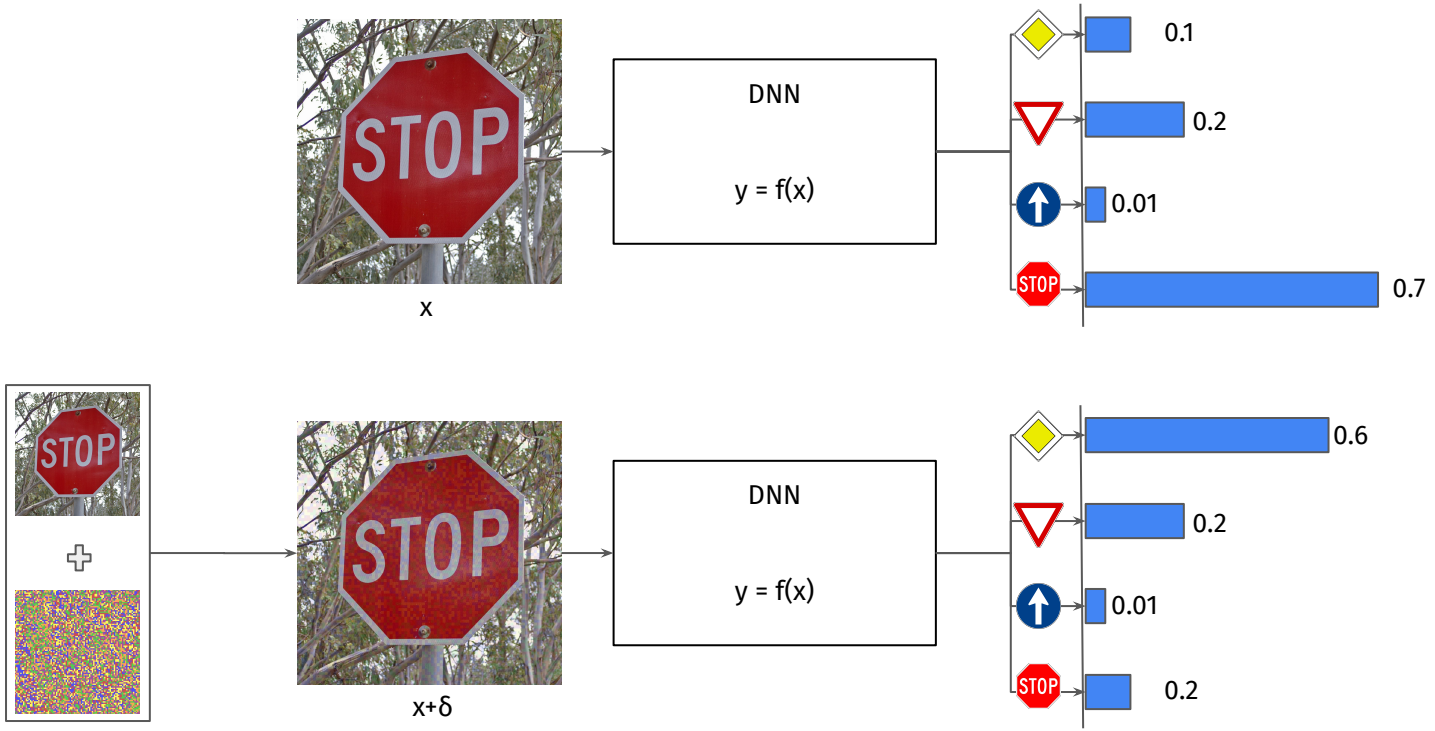


Encoding Table

0	
1	
2	
4	STOP

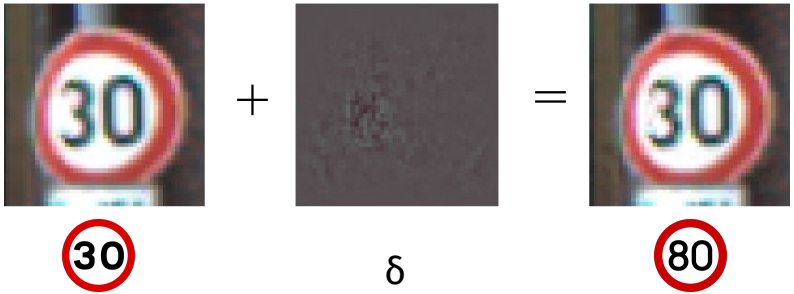
Arcos-García et al. "Deep neural network for traffic sign recognition systems", (2018)

Adversarial Examples



Adversarial Examples: Digital or Real World

Digital Adversarial Perturbation



A malicious perturbation δ undetectable from human eye

Focus of this Lecture

Physical Adversarial Attacks

PATCH

POSTER

Adversarial perturbations that can be perceived from cameras, microphones, etc...

Focus of Next Lectures

Definition of Adversarial Examples

Definition (Adversarial Examples)
The result of perturbations, with a *small* l^p norm, that fool a classification model.

Mandatory Right | Adversarial Perturbation | Speed Limit 20 Km/h

$$\mathcal{K}(x+\delta) \neq \mathcal{K}(x) \quad \text{and} \quad \|\delta\| \approx 0$$

Some l^p Norm

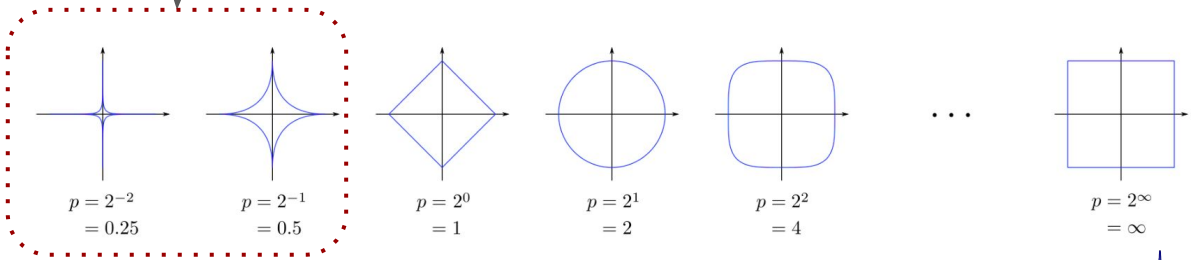
l^p - norms (Minkowski Distances)

$$\|\delta\|_0 = \#\{i \,:\, \delta_i \neq 0\}$$

$$\|\delta\|_p = \left(\sum_{i=1}^{n} |\delta_i|^p\right)^{\frac{1}{p}}$$

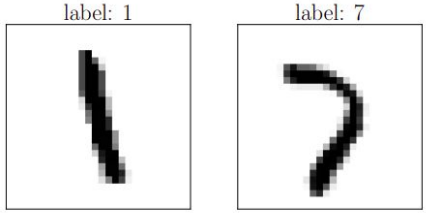
$$\|\delta\|_\infty = \max_i |\delta_i|$$

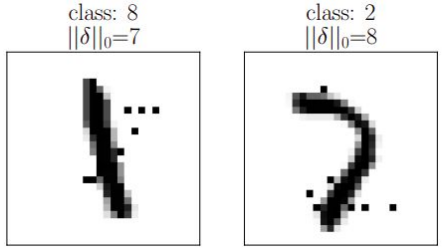
No Convex (pseudo-norm)



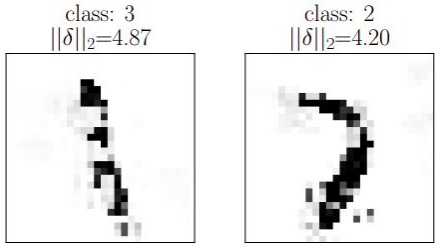
Different visualization of l^p bounded attacks

Definition in Minkowski norm covers many visually different cases





Few pixels, Unbounded values

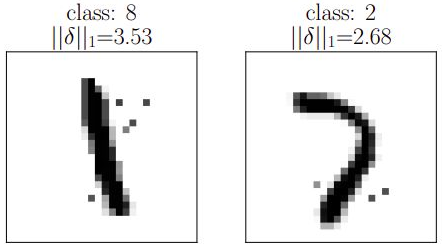


Few pixels, Bounded values

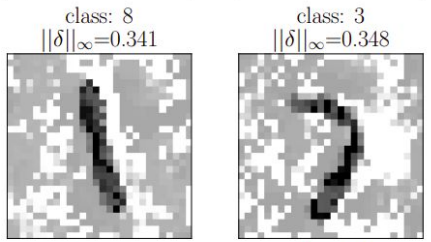
$$||\delta||_0 = \#\{i : \delta_i \neq 0\}$$

$$||\delta||_p = \left(\sum_{i=1}^{n} |\delta_i|^p\right)^{\frac{1}{p}}$$

$$||\delta||_\infty = \max_i |\delta_i|$$



Few pixels, Bounded values



Many pixels, Bounded values

Maura Pintor et al. “Fast Minimum-norm Adversarial Attacks through Adaptive Norm Constraints” (2021)

Targeted or Untargeted

Targeted
The attacker aims at making the model answering `<target>`

Un-Targeted
The attacker only aims at fooling the model, without setting the output

`bird` `truck` `frog`

Black or White Box

White Box

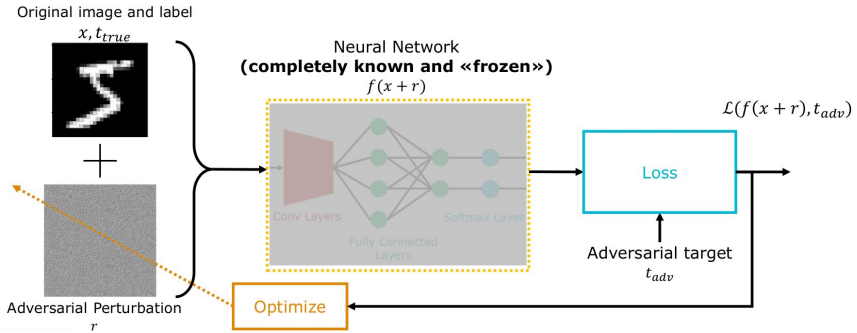
The attacker has full access to:

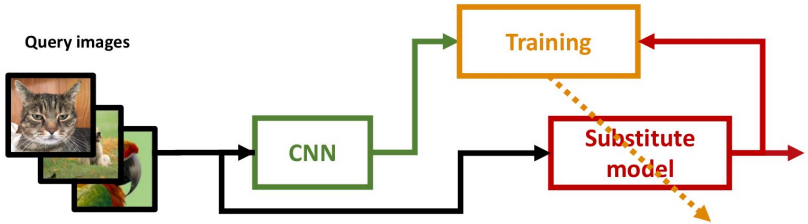
1. *Model architecture*
2. *Model weights*
3. *Data*

Black Box

The model acts has **oracle** and
the attacker has full access to:

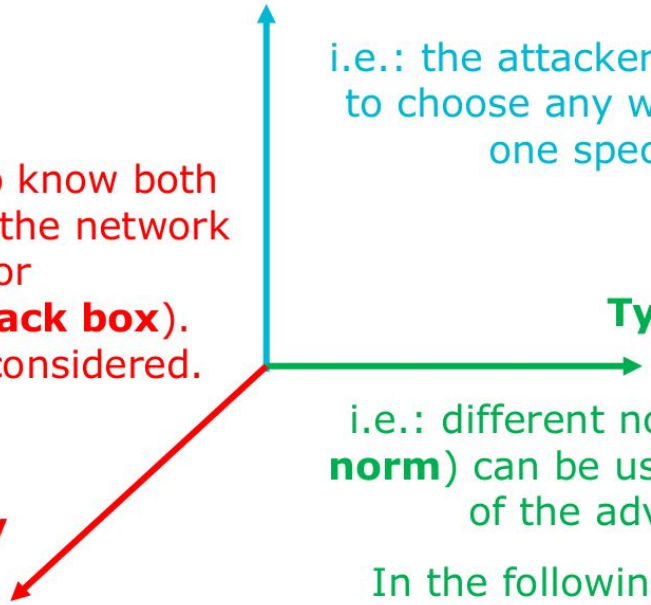
1. *Model Architecture* but not *data*
2. *Data* but not *Model Architecture*
3. Neither *Data* or *Model Architecture*





Federico Nesti. “An introduction to Adversarial Examples”, lecture of Deep Learning Course (2022)

Minimal Taxonomy of Adversarial Examples



Targeted/untargeted

i.e.: the attacker aims at forcing the classifier to choose any wrong class (**untargeted**) vs one specific class (**targeted**)

i.e.: the attacker is assumed to know both architecture and parameters of the network (**white box** setting) or anything about the model (**black box**). Intermediate settings can be considered.

Type of norm

i.e.: different norms (**2-norm, 1-norm, ∞-norm**) can be used to express the magnitude of the adversarial perturbation.

In the following slides no specific norm is considered, any of these will work (with different effects)

Knowledge of the adversary

Federico Nesti. “An introduction to Adversarial Examples”, lecture of Deep Learning Course (2022)

Generation of Adversarial Examples

Generation of Adversarial Examples

Methods based on the Loss Function

Un-Targeted
Increase the Loss Function in the correct class close to the input x

$$\max_{\delta \in \mathbb{R}^n} \quad \mathcal{L}(f(x+\delta), l_{true})$$

$$\text{s.t.} \quad \|\delta\|_p \leq \varepsilon$$

$$0 \leq x + \delta \leq 1$$

$$Q * (x + \delta) \in \{0, \cdots, Q\}$$

Targeted
Decrease the Loss Function in the target class close to the input x

$$\min_{\delta \in \mathbb{R}^n} \quad \mathcal{L}(f(x+\delta), l_{target})$$

$$\text{s.t.} \quad \|\delta\|_p \leq \varepsilon$$

$$0 \leq x + \delta \leq 1$$

$$Q * (x + \delta) \in \{0, \cdots, Q\}$$

Box Constraint

Integer Constraint, Q=255

Box and Integer Constraints ensure that the Adv.Ex is a representable image.

Generation of Adversarial Examples

Methods based on computing the
Minimal Adversarial Perturbation

Un-Targeted

Searching for the closest sample to x that is classified in a different class

$$\min_{\delta \in \mathbb{R}^n} \quad \|\delta\|_p$$

$$\text{s.t.} \quad \mathcal{K}(x+\delta) \neq l_{true}$$

$$0 \leq x+\delta \leq 1$$

$$Q * (x+\delta) \in \{0, \cdots, Q\}$$

Targeted

Searching for the closest sample to x that is classified in the target class

$$\min_{\delta \in \mathbb{R}^n} \quad \|\delta\|_p$$

$$\text{s.t.} \quad \mathcal{K}(x+\delta) = l_{target}$$

$$0 \leq x+\delta \leq 1$$

$$Q * (x+\delta) \in \{0, \cdots, Q\}$$

Box Constraint

Integer Constraint,
Q=255

Box and Integer Constraints ensure that the Adv.Ex is a representable image.

Fast Gradient Sign Method

Leveraging the Loss Function

The loss function is used to search for the **worst** direction that fools the model and **not** to train the model $f : \mathbb{R}^n \to \mathbb{R}^C$

$$\delta^* = \varepsilon \cdot \text{sign}\left(\nabla_x \mathcal{L}(f(x), l_{true})\right)$$

Desired Magnitude in l^∞ norm

The gradient provides the direction of the largest growth

Observation.
The adversarial perturbation has bounded uniform norm

$$x^* = x + \delta^*, \qquad \|\delta^*\|_\infty = \varepsilon$$

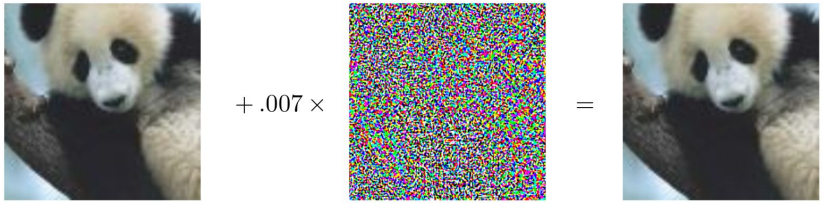
Goodfellow et al. “Explaining and Harnessing Adversarial Examples” (2014)

Fast Gradient Sign Method

FGSM

$$\delta^* = \varepsilon \cdot \text{sign}\left(\nabla_x \mathcal{L}(f(x), l_{true})\right)$$

$$x^* = x + \delta^*$$



The most famous Adv Ex. Attack on GoogLeNet

Pros

1. Easy to implement
2. Fast (useful in adv. training)
3. Bounded in uniform norm

Cons

1. Requires ε large (sometime visible)
2. Not always provide an Adv

Adversarial Training

Train the models by leveraging the following loss function

$$\tilde{\mathcal{L}}(f(x), l) = \alpha \cdot \mathcal{L}(f(x), l) \quad + \quad (1-\alpha) \cdot \mathcal{L}(f(x+\delta^*), l)$$

Adversarial Loss

Goodfellow et al. “Explaining and Harnessing Adversarial Examples” (2014)

Minimal Adversarial Perturbation with Penalty

Minimum Problem with Constraints

$$\min_{\delta \in \mathbb{R}^n} \quad \|\delta\|_p \qquad *$$

$$\text{s.t.} \quad \mathcal{K}(x+\delta) = l_{target}$$

$$0 \leq x + \delta \leq 1$$

Minimum Problem with Penalty

$$\min_{\delta \in \mathbb{R}^n} \quad c\|\delta\|_p + \mathcal{L}(f(x+\delta), l_{target})$$

$$\text{s.t.} \quad 0 \leq x + \delta \leq 1$$

Penalty Constant.
Unbounded solution for $c=0$,
$\delta(c) \sim 0$ for large values of c

Theorem.
Under convex assumption on $\mathcal{L}(f(x+\delta), l_{target})$ in the variable x we have that

1. For each c, the Box constrained L-BFGS has a solution $\delta(c)$
2. The penalty c can be optimized by line-search such that $\mathcal{K}(x+\delta(c)) = l_{target}$

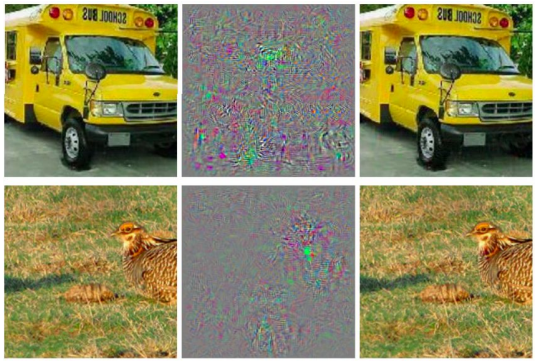
Christian Szegedy et al. “Intriguing properties of neural networks” (2013)

Minimal Adversarial Perturbation with Penalty

BL-BFGS

Box constrained **L**imited memory **B**royden-**F**letcher-**G**oldfarb-**S**hanno is a quasi-Newton method for minimum problem with box constraints. [a]

$$\min_{\delta \in \mathbb{R}^n} \quad c\,\|\delta\|_p + \mathcal{L}(f(x+\delta), l_{target})$$
$$\text{s.t.} \quad 0 \leq x + \delta \leq 1$$



First Adversarial Examples presented [b] to the community. Attack of AlexNet

Pros.

1. Provide particularly low perturbations (approximate the Minimal Adversarial Perturbation)
2. Reliable. Almost always converge.

Cons.

1. Slow. Requires ~20k forwards of the model for each line search of c

[a] Dimitri P. Bertsekas. “Constrained Optimization and Lagrange Multiplier Methods” (1996)
[b] Christian Szegedy et al. “Intriguing properties of neural networks” (2013)

Including the Box Constraint in the Objective (CW)

Minimum Problem with Constraints

$$\min_{\delta \in \mathbb{R}^n} \quad \|\delta\|_p \qquad * \qquad *$$

$$\text{s.t.} \quad \mathcal{K}(x+\delta) = l_{target}$$

$$0 \leq x + \delta \leq 1$$

Change of Variable (Simplification 1)

Optimizing w instead of directly optimize over the adversarial perturbation δ

$$\delta = \frac{1}{2}\left(\tanh(w) + 1\right) - x$$

Remark!

The perturbation δ satisfies the Box.Con.

Improve Loss Function (Simplification 2)

$$\mathcal{L}(y, l) = \left[y_l - \max_{j \neq l} y_j \right]^+$$

Positive part of the "margin" computed on the logits of f

Carlini et al. "Towards Evaluating the Robustness of Neural Networks" (2016)

Including the Box Constraint in the Objective (CW)

Minimum Problem with Constraints

$$\min_{\delta \in \mathbb{R}^n} \quad \|\delta\|_p \qquad * \qquad *$$

$$\text{s.t.} \quad \mathcal{K}(x+\delta) = l_{target}$$

$$0 \leq x + \delta \leq 1$$

Minimum Problem with Penalty

$$\min_{\delta \in \mathbb{R}^n} \quad c\,\|\delta\|_p + \mathcal{L}(f(x+\delta), l_{target})$$

$$\text{s.t.} \quad 0 \leq x + \delta \leq 1$$

CW Minimum Problem (Unconstrained)

$$\min_{w \in \mathbb{R}^n} \quad \| \underbrace{\frac{1}{2}(\tanh(w)+1) - x}_{\delta} \|_p + c \cdot \mathcal{L}(f(\underbrace{\frac{1}{2}(\tanh(w)+1)}_{x+\delta}), l_{target})$$

Remark

The penalty c can be optimized by line-search such that $\mathcal{K}(x+\delta(c)) = l_{target}$

Carlini et al. “Towards Evaluating the Robustness of Neural Networks” (2016)

Including the Box Constraint in the Objective (CW)

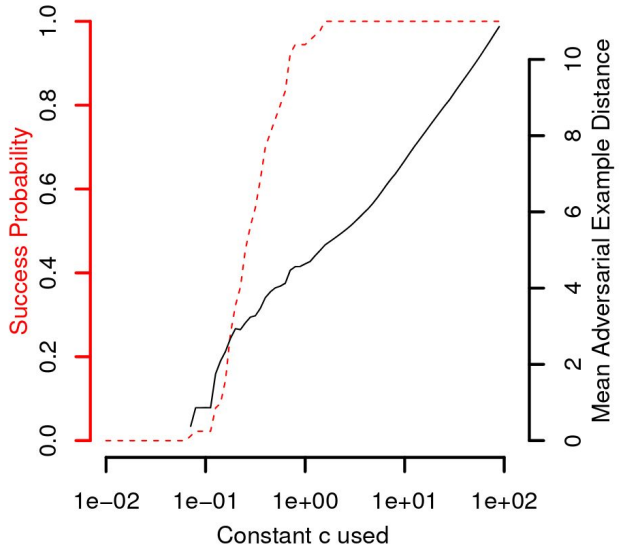
CW Minimum Problem (Unconstrained)

$$\min_{w \in \mathbb{R}^n} \; \| \underbrace{\frac{1}{2}(\tanh(w)+1) - x}_{\delta} \|_p + c \cdot \mathcal{L}(f(\underbrace{\frac{1}{2}(\tanh(w)+1)}_{x+\delta}), l_{target})$$

Pros.
The first work that **directly** aims at solving MAP

MAP provides **Certifiable Guarantees** (next lecture)

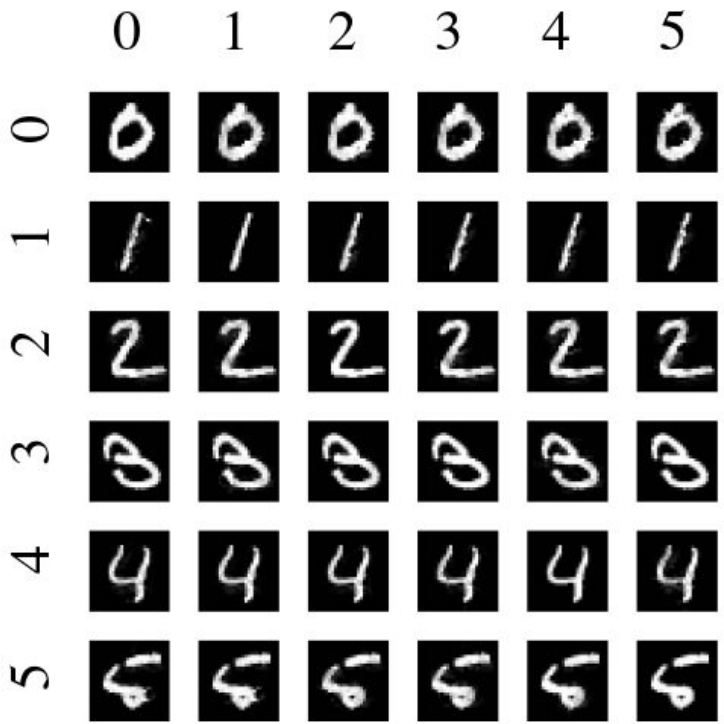
Cons.
Forward required for each c : ~20k (Hence slow)
Binary steps required : ~ 12



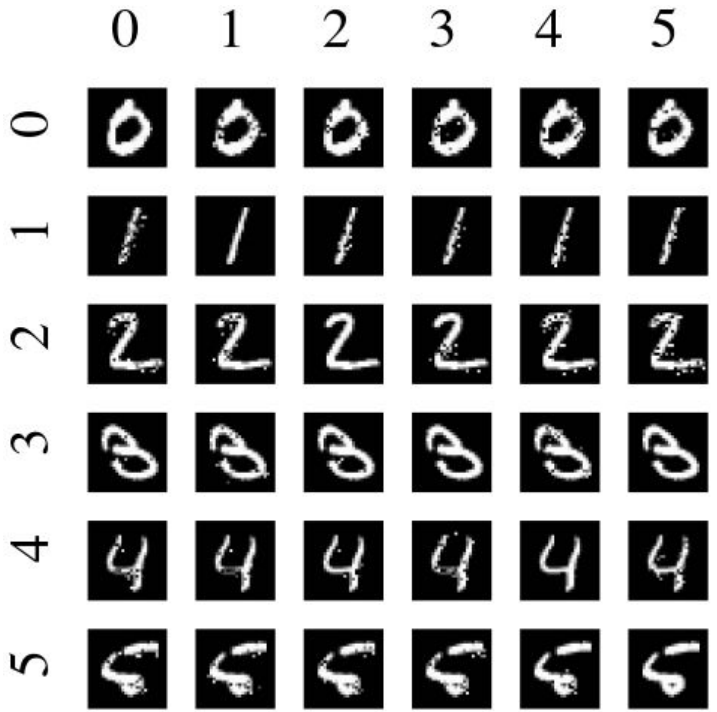
- A large penalty implies high success rate but farther adversarial examples
- A small penalty implies accurate solutions but small convergence rate.

Carlini et al. “Towards Evaluating the Robustness of Neural Networks” (2016)

Examples of attack with CW



l^2 Minimal Adversarial Perturbation



l^0 Minimal Adversarial Perturbation

Carlini et al. “Towards Evaluating the Robustness of Neural Networks” (2016)

DeepFool: A Fast and Accurate Method

The Binary Linear Case

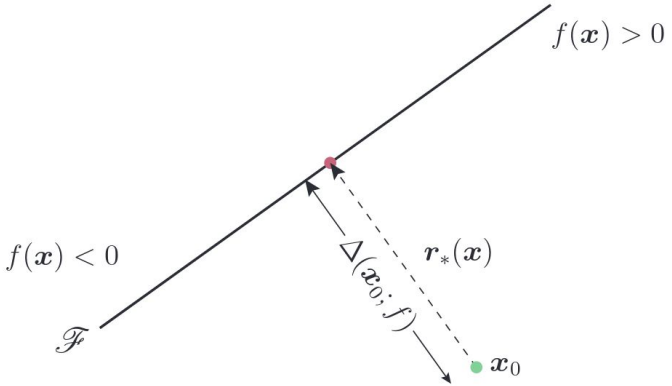
The following is an explicit solution.

If $f(x) = w^T x + b$

then

$$\delta = -\frac{f(x)}{\|w\|^2} w$$

Minimal Adversarial Perturbation
that fools the binary linear classifier

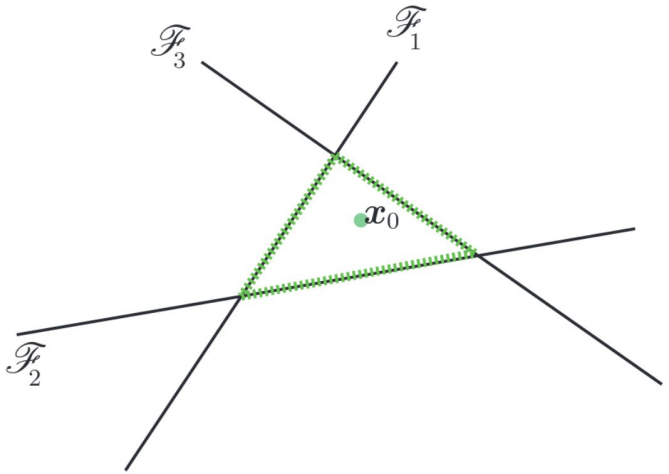


Seyed-Mohsen Moosavi-Dezfooli et al. “DeepFool: a simple and accurate method to fool deep neural networks” (2016)

DeepFool: A Fast and Accurate Method

Extension to the Multiclass Case

$$f(x) = Wx + b, \quad W = \begin{bmatrix} w^{(1)T} \\ \vdots \\ w^{(C)T} \end{bmatrix} \in \mathbb{R}^{C \times n}$$



Remark.
Minimal Adversarial Perturbations targeted for each class i

$$\delta^{(i)} = -\frac{f_i(x)}{\|w^{(i)}\|^2} w^{(i)}$$

The smallest corresponds to the **untargeted** attack.

Seyed-Mohsen Moosavi-Dezfooli et al. “DeepFool: a simple and accurate method to fool deep neural networks” (2016)

DeepFool: A Fast and Accurate Method

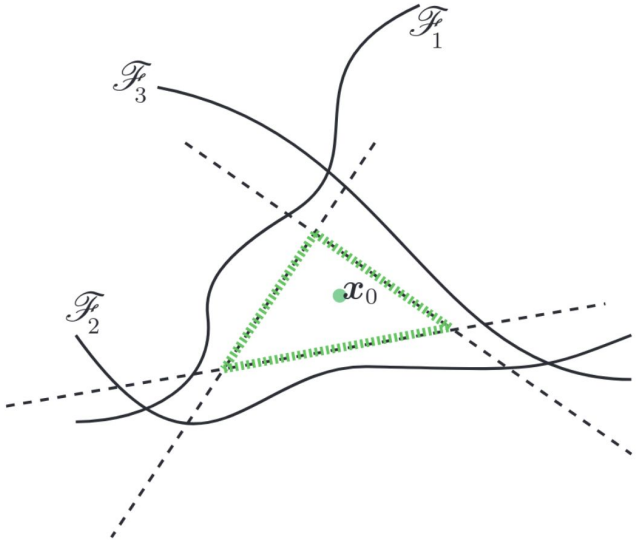
Not-Linear Multiclass Case

Algorithm 2 DeepFool: multi-class case

input: Image $\boldsymbol{x}$, classifier f.
output: Perturbation $\hat{\boldsymbol{r}}$.

Initialize $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$, $i \leftarrow 0$.
while $\hat{k}(\boldsymbol{x}_i) = \hat{k}(\boldsymbol{x}_0)$ **do**
  for $k \neq \hat{k}(\boldsymbol{x}_0)$ **do**
    $\boldsymbol{w}'_k \leftarrow \nabla f_k(\boldsymbol{x}_i) - \nabla f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)$
    $f'_k \leftarrow f_k(\boldsymbol{x}_i) - f_{\hat{k}(\boldsymbol{x}_0)}(\boldsymbol{x}_i)$
  end for
  $\hat{l} \leftarrow \arg\min_{k \neq \hat{k}(\boldsymbol{x}_0)} \frac{|f'_k|}{\|\boldsymbol{w}'_k\|_2}$
  $\boldsymbol{r}_i \leftarrow \frac{|f'_{\hat{l}}|}{\|\boldsymbol{w}'_{\hat{l}}\|_2^2} \boldsymbol{w}'_{\hat{l}}$
  $\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i + \boldsymbol{r}_i$
  $i \leftarrow i + 1$
end while
return $\hat{\boldsymbol{r}} = \sum_i \boldsymbol{r}_i$

Idea: Iterate the procedure



Seyed-Mohsen Moosavi-Dezfooli et al. “DeepFool: a simple and accurate method to fool deep neural networks” (2016)

DeepFool: A Fast and Accurate Method

Empirical Robustness Estimation

$$\rho_{adv}(\mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \frac{\|\delta(x)\|_2}{\|x\|_2}$$

Testing Dataset

Minimal Adversarial Perturbation on x

Classifier	Test error	$\hat{\rho}_{\text{adv}}$ [DeepFool]	time
LeNet (MNIST)	1%	2.0×10^{-1}	110 ms
FC500-150-10 (MNIST)	1.7%	1.1×10^{-1}	50 ms
NIN (CIFAR-10)	11.5%	2.3×10^{-2}	1100 ms
LeNet (CIFAR-10)	22.6%	3.0×10^{-2}	220 ms
CaffeNet (ILSVRC2012)	42.6%	2.7×10^{-3}	510 ms*
GoogLeNet (ILSVRC2012)	31.3%	1.9×10^{-3}	800 ms*

Pros.

1. Fast, requires ~3C forwards
2. Provide Empirical MAP for the whole dataset (CW can not)

Cons.

1. Still no theoretical guarantees
2. Suboptimal solutions are found

Seyed-Mohsen Moosavi-Dezfooli et al. “DeepFool: a simple and accurate method to fool deep neural networks” (2016)

Projected Gradient Descent (PGD)

Maximum Problem (Un-Targeted)

$$\begin{aligned} \max_{\delta \in \mathbb{R}^n} \quad & \mathcal{L}(f(x+\delta), l_{true}) \\ \text{s.t.} \quad & \|\delta\|_p \leq \varepsilon \\ & 0 \leq x+\delta \leq 1 \\ & Q*(x+\delta) \in \{0, \cdots, Q\} \end{aligned}$$

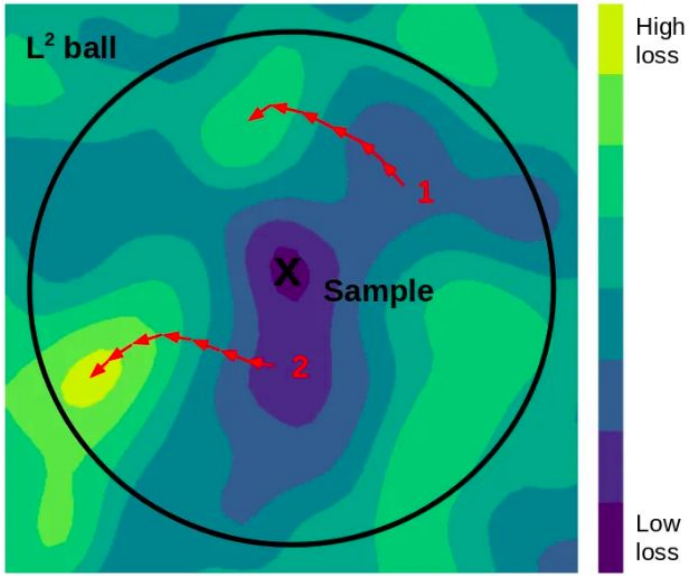
Iterating FGSM

Learning rate of the Gradient **A**scent

$$\tilde{x}^{(t)} = x^{(t-1)} + \alpha \cdot \text{sgn}\left(\nabla_x \mathcal{L}(f(x^{(t-1)}), l_{true})\right)$$

$$x^{(t)} = \Pi_{B_p(x,\varepsilon)}\left(\tilde{x}^{(t)}\right)$$

Projection into the p-Ball of radius ε



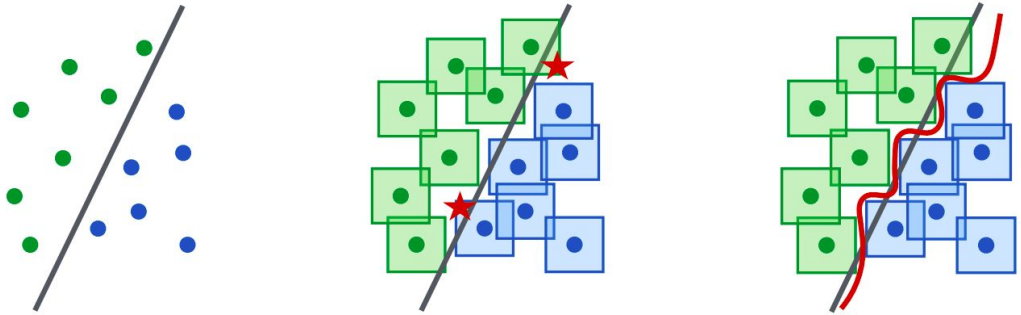
1. Take a random starting point
2. Iterate the SG(A) for few steps
3. Project into the l^p Ball
4. Repeat 1,2,3

Aleksander Madry et al. “Towards Deep Learning Models Resistant to Adversarial” (2018)

Adversarial Training by PGD

Minimization of the Empirical Worst Case

$$\min_{\theta} \rho(\theta), \quad \rho(\theta) = \mathbb{E}_{(x,l)\in\mathcal{X}} \left[\max_{\|\delta\|_p < \varepsilon} \mathcal{L}\left(f_\theta(x+\delta), l; \theta\right) \right]$$



Aleksander Madry et al. “Towards Deep Learning Models Resistant to Adversarial” (2018)

A Black Box Attack

Model acts as an Oracle [a]

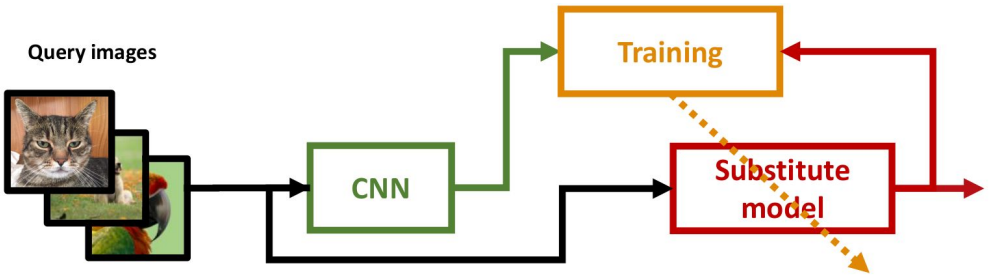
The model $\mathcal{K}$ is given without:

1. Training dataset
2. Model Architecture and Weights

Procedure

1. Initial Collection
2. Architecture Selection
3. Substitute Training
4. Adversarial sample Crafting

Attacking the Substitute Model [b]



A new model is trained, and a white-box attack is performed on the new one; the procedure leverages the **TRANSFERABILITY PROPERTY** of Adversarial Examples

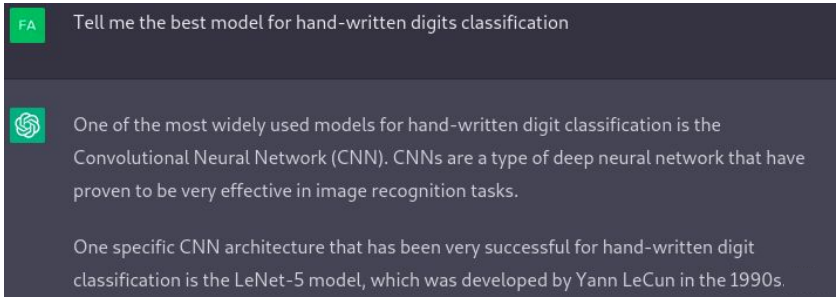
[a] Nicolas Papernot et al. “Practical Black-Box Attacks against Machine Learning” (2017)
[b] Federico Nesti. “An introduction to Adversarial Examples”, lecture of Deep Learning Course (2022)

A Black Box Attack: Substitute DNN and Training

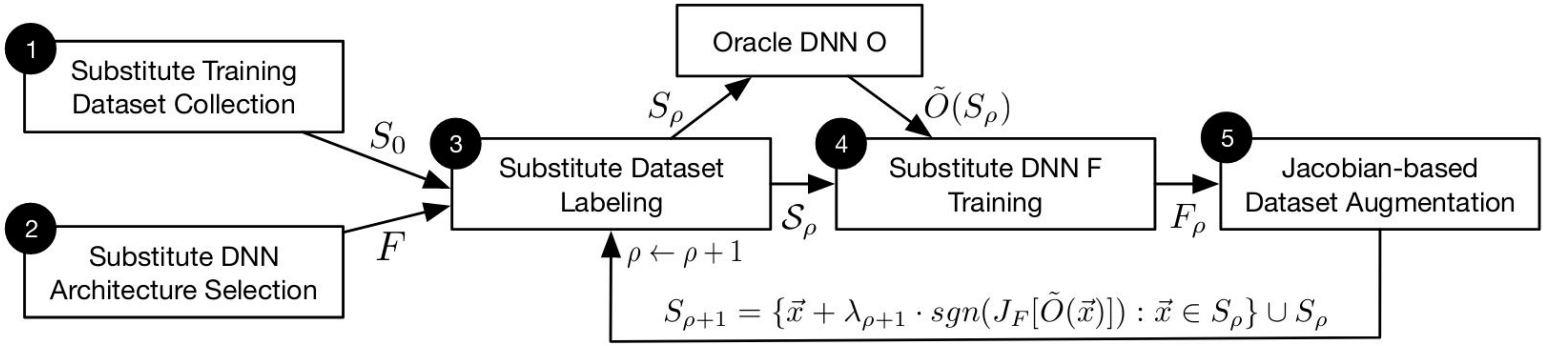
Initial Collection

Hand-crafted images of 10 digits. At least one image per class has to be chosen in order to start the procedure.

Architecture Selection



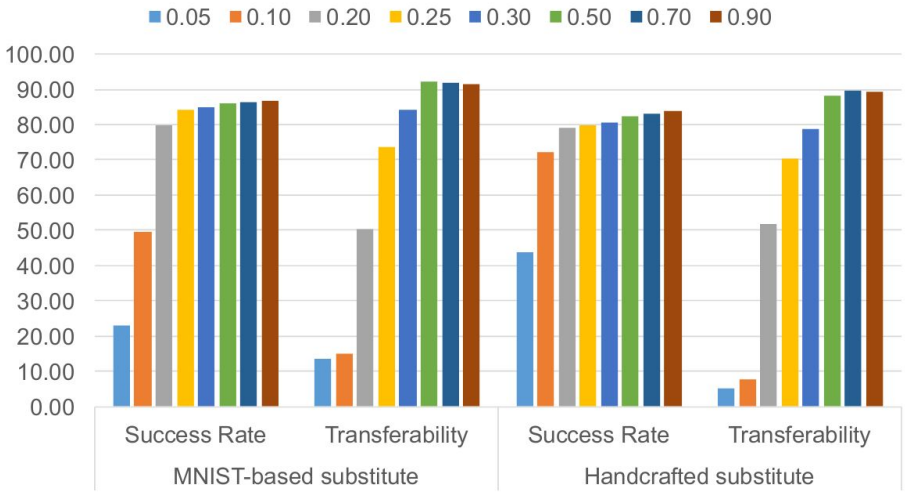
Training Schema



Nicolas Papernot et al. “Practical Black-Box Attacks against Machine Learning” (2017)

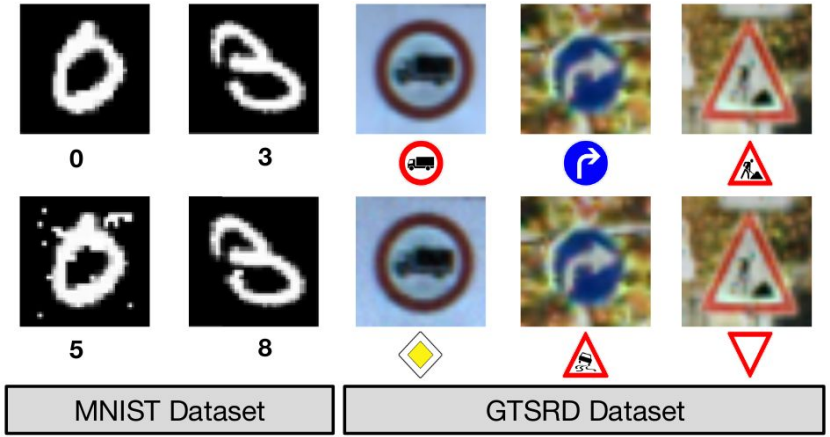
A Black Box Attack: Evaluation

Success rate and Transferability



For both models $\varepsilon=0.2$

Graphical Evaluation



Nicolas Papernot et al. “Practical Black-Box Attacks against Machine Learning” (2017)

Defense Strategies

Attacker-Defensor Paradigm

1. Adversarial training: Augmenting the training data with adversarial examples.

2. Defensive distillation: Training a second DNN (**distilled model)** to mimic the output of the first DNN on the training set. The distilled model is, empirically, more robust.

3. Gradient masking: Modifying the DNN architecture to mask the gradients in certain layers (avoid white-box adversarial attacks).

4. Input Preprocessing: Modifying the input data. E.g., adding noise or blurring the image, to make it more difficult for an attacker to create effective adversarial examples.

5. Randomized defenses: Adding randomness to the DNN at different stages, such as the weights, the architecture, or the input. (**next lecture**)

6. Lipschitz Regularization: Crafting models with a bounded Lipschitz constant to certify the robustness against adversarial examples. (**next lecture**)

Adversarial Training

Adding Adversarial Examples at training stage to improve the robustness.

FGSM

Goodfellow et al. proposed a new loss function

$$\tilde{\mathcal{L}}(f(x), l) = \alpha \cdot \mathcal{L}(f(x), l) \quad + \quad (1-\alpha) \cdot \mathcal{L}(f(x+\delta^*), l)$$

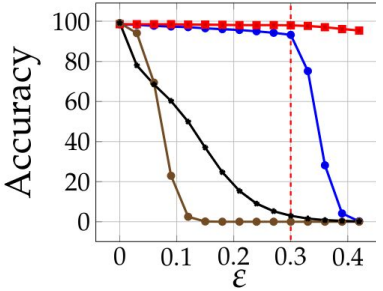
Adversarial Loss

Projected Gradient Descent

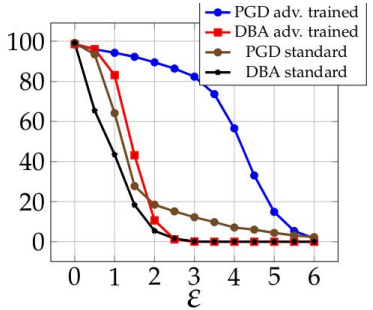
Madry et al. Improved the strategies with a finer estimation

$$\min_{\theta} \rho(\theta), \quad \rho(\theta) = \mathbb{E}_{(x,l)\in\mathcal{X}} \left[\max_{\|\delta\|_p < \varepsilon} \mathcal{L}\left(f_\theta(x+\delta), l; \theta\right)) \right]$$

Remark! The Robustness evaluation is empirical (no guaranty)



(a) MNIST, ℓ_∞-norm



(b) MNIST, ℓ_2-norm

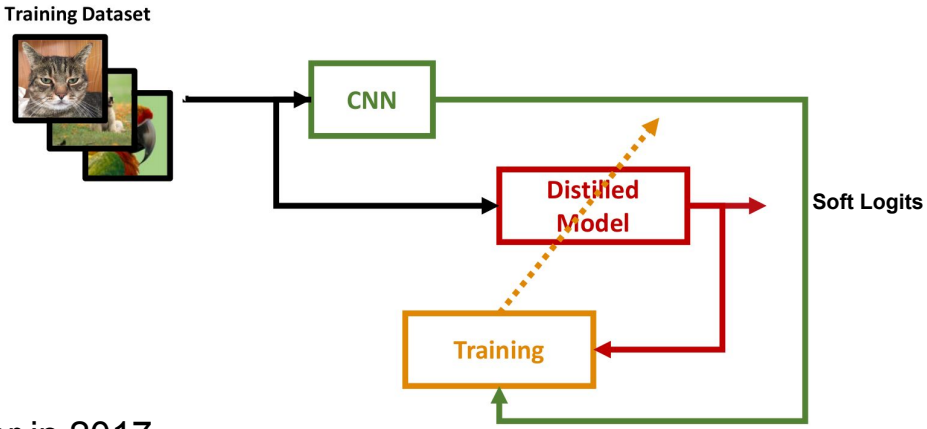
Goodfellow et al. “Explaining and Harnessing Adversarial Examples” (2014)
Aleksander Madry et al. “Towards Deep Learning Models Resistant to Adversarial” (2018)

Defensive Distillation

Training a Smaller Model by leveraging Cross-Entropy with high temperature

Soft Training

The **distilled model** is trained not with hard labels (one-hot encoded vectors), but with soft labels (probability values of the prediction of the original network)



Remark. The strategy is broken by Carlini-Wagner in 2017.

[a] Nicolas Papernot et al. “Distillation as a defense to adversarial perturbations against deep neural networks” (2016)
[b] Federico Nesti. “An introduction to Adversarial Examples”, lecture of Deep Learning Course (2022)

Defensive Distillation

Cross-Entropy with Temperature

$$\mathcal{L}(y, l; T) = -\log\left(\frac{e^{\frac{y_l}{T}}}{\sum_{i=1}^{C} e^{\frac{y_i}{T}}}\right)$$

Temperature

High temperature increases the **uncertainty**, and hence improves the (empirical) **robustness** and the **generalization** of the distilled model

$T = 1$ (regular softmax)

$$y_{soft}(x) = \begin{bmatrix} 0{,}007 \\ 0{,}011 \\ 0{,}062 \\ 0{,}920 \end{bmatrix}$$

$T = 2$

$$y_{soft}(x) = \begin{bmatrix} 0.059 \\ 0.076 \\ 0.178 \\ 0.687 \end{bmatrix}$$

$T \to \infty$

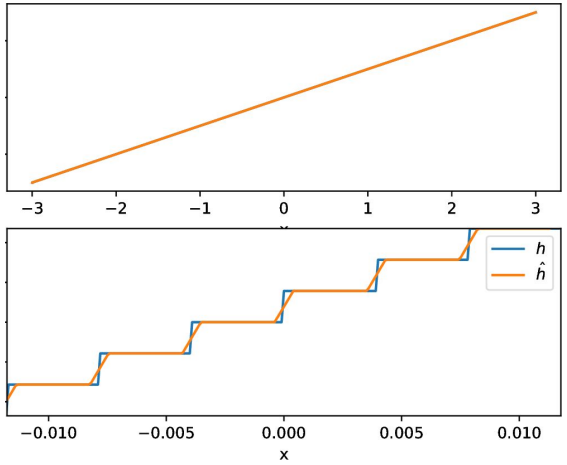
$$y_{soft}(x) = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}$$

Federico Nesti. “An introduction to Adversarial Examples”, lecture of Deep Learning Course (2022)

Gradient Masking

Avoid white-box attacks and evaluation based on the gradient by a.e flat activation function

Remark. The strategy is broken by Papernot with black box attacks



Gradient of a model with such a activation function is 0 almost-everywhere

Fun fact

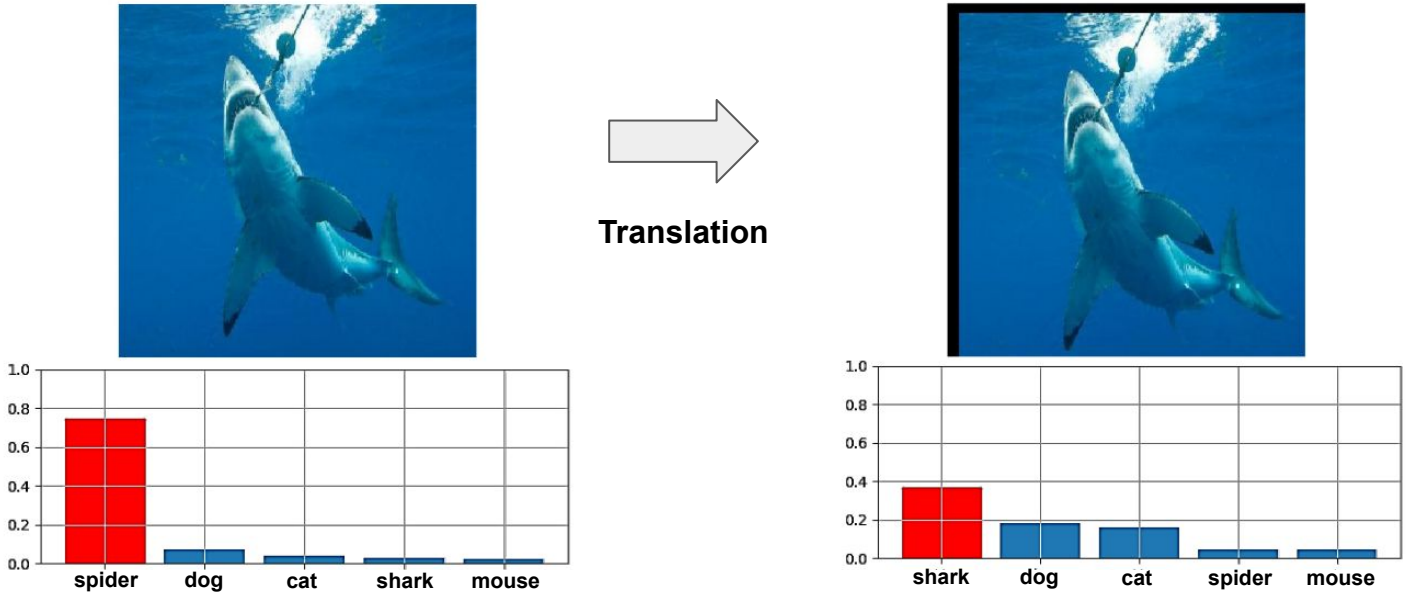
Discussion with Ian Goodfellow and Our Clarifications

We received some inquires on Ian Goodfellow's comment "*Gradient Masking Causes CLEVER to Overestimate Adversarial Perturbation Size*" on our paper. We thank Ian for the discussion but the comments are inappropriate and not applicable to our paper. CLEVER is intended to be a tool for network designer and to evaluate network robustness in the "white-box" setting. Especially, the argument that on digital computers all functions are not Lipschitz continuous and behave like a staircase function (where the gradient is zero almost everywhere) is incorrect. Under the white-box setting, gradients can be computed via automatic differentiation, which is well supported by mature packages like TensorFlow. See our reply and discussions with Ian Goodfellow on gradient masking and implementation on digital computers.

CLEVER is a verification strategy that leverages the gradient to estimate the robustness of a sample

CLEVER website: https://github.com/IBM/CLEVER-Robustness-Score/blob/master/README.md
Ian Goodfellow. "Gradient Masking Causes CLEVER to Overestimate Adversarial Perturbation Size" (2018)

Input Transformations



Other Transformations
Rotations, Mirror, Blur, Gaussian Noise

Federico Nesti. “An introduction to Adversarial Examples”, lecture of Deep Learning Course (2022)

Conclusion

Adversarial Examples for Biological Vision Systems

Polyphemus Moth

Defense mechanisms that involves the **large eyespots** on its hindwings.

Chihuahua or Muffin

Shankar, Goodfellow and Papernot. “Adversarial Examples that Fool both Human and Computer Vision” (2018)

Conclusion

Introduction, Definitions and Taxonomy

Adversarial Examples Generation

Defense Mechanisms

Thanks for the attention

Fabio Brau

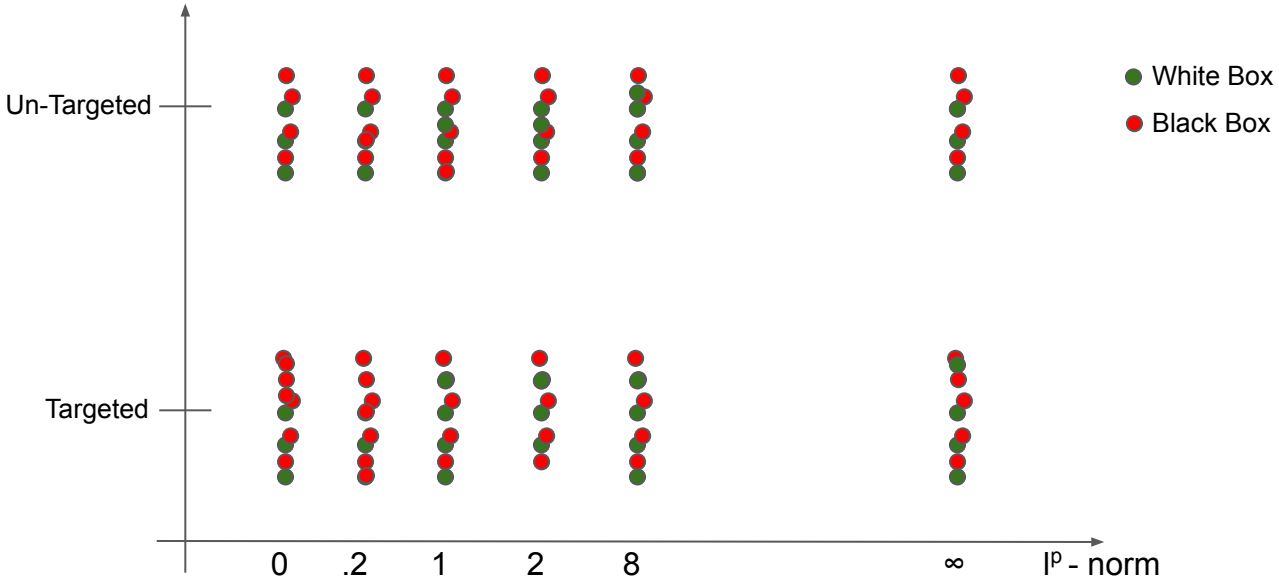
Scuola Superiore Sant'Anna, Pisa

fabio.brau@santannapisa.it

https://www.linkedin.com/in/fabio-brau/

http://retis.santannapisa.it/~f.brau

Taxonomy Map



Generation of Adversarial Examples (Un-Targeted)

Adversarial Generation follows two main strategies

Loss Dependent

Increase the Loss Function on the correct class of the input x

$$\max_{\delta \in \mathbb{R}^n} \quad \mathcal{L}(f(x+\delta), l_{true})$$

$$\text{s.t.} \quad \|\delta\|_p \leq \varepsilon$$

Minimum Distance

Searching for the Closest Sample to x that is classified in a different class

$$\min_{\delta \in \mathbb{R}^n} \quad \|\delta\|_p$$

$$\text{s.t.} \quad \mathcal{K}(x+\delta) \neq l_{true}$$

$0 \leq x + \delta \leq 1$ ← Box Constraint → $0 \leq x + \delta \leq 1$

$Q * (x + \delta) \in \{0, \cdots, Q\}$ $\quad$ $Q * (x + \delta) \in \{0, \cdots, Q\}$

Quantized Constraint

Generation of Adversarial Examples (Targeted)

Adversarial Generation follows two main strategies

Loss Dependent

Decrease the Loss Function on the target class

$$\max_{\delta \in \mathbb{R}^n} \quad \mathcal{L}(f(x+\delta), l_{true})$$

$$\text{s.t.} \quad \|\delta\|_p \leq \varepsilon$$

$$0 \leq x + \delta \leq 1$$

$$Q * (x + \delta) \in \{0, \cdots, Q\}$$

Minimum Distance

Searching for the Closest Sample to x that is classified in the target class

$$\min_{\delta \in \mathbb{R}^n} \quad \|\delta\|_p$$

$$\text{s.t.} \quad \mathcal{K}(x+\delta) \neq l_{true}$$

$$0 \leq x + \delta \leq 1$$

$$Q * (x + \delta) \in \{0, \cdots, Q\}$$

Box Constraint

Quantized Constraint