Certifiable Adversarial Robustness

Deep Learning and Neural Networks: Advanced Topics

Fabio Brau

March 22, 2022

SSSA, Emerging Digital Technologies, Pisa.

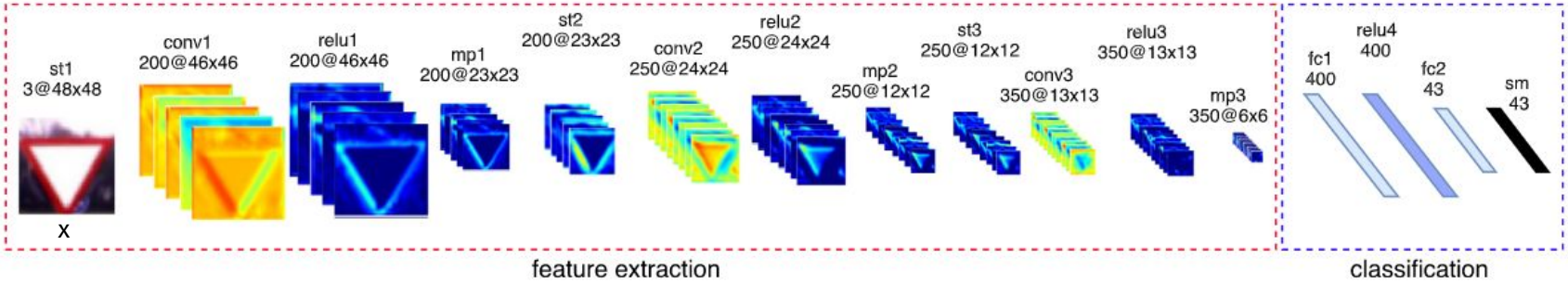
Introduction and Definition of Certifiable Robustness

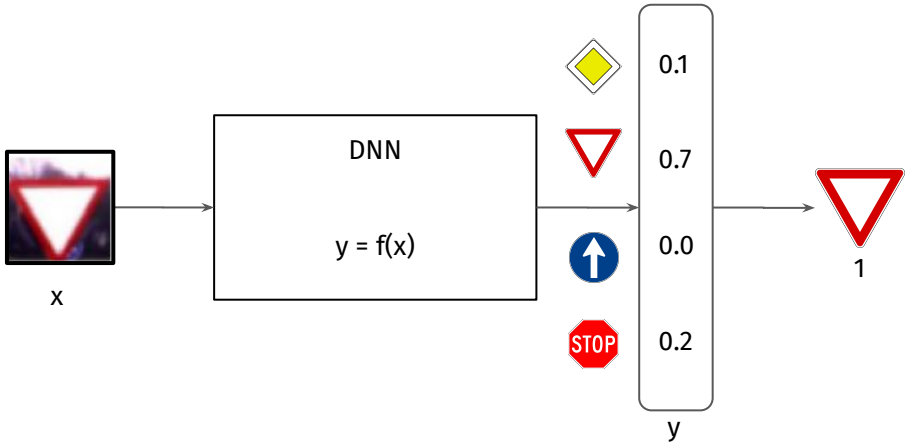
Verification Methods

Lipschitz Bounded Neural Networks

Randomized Smoothing

The Classification Problem





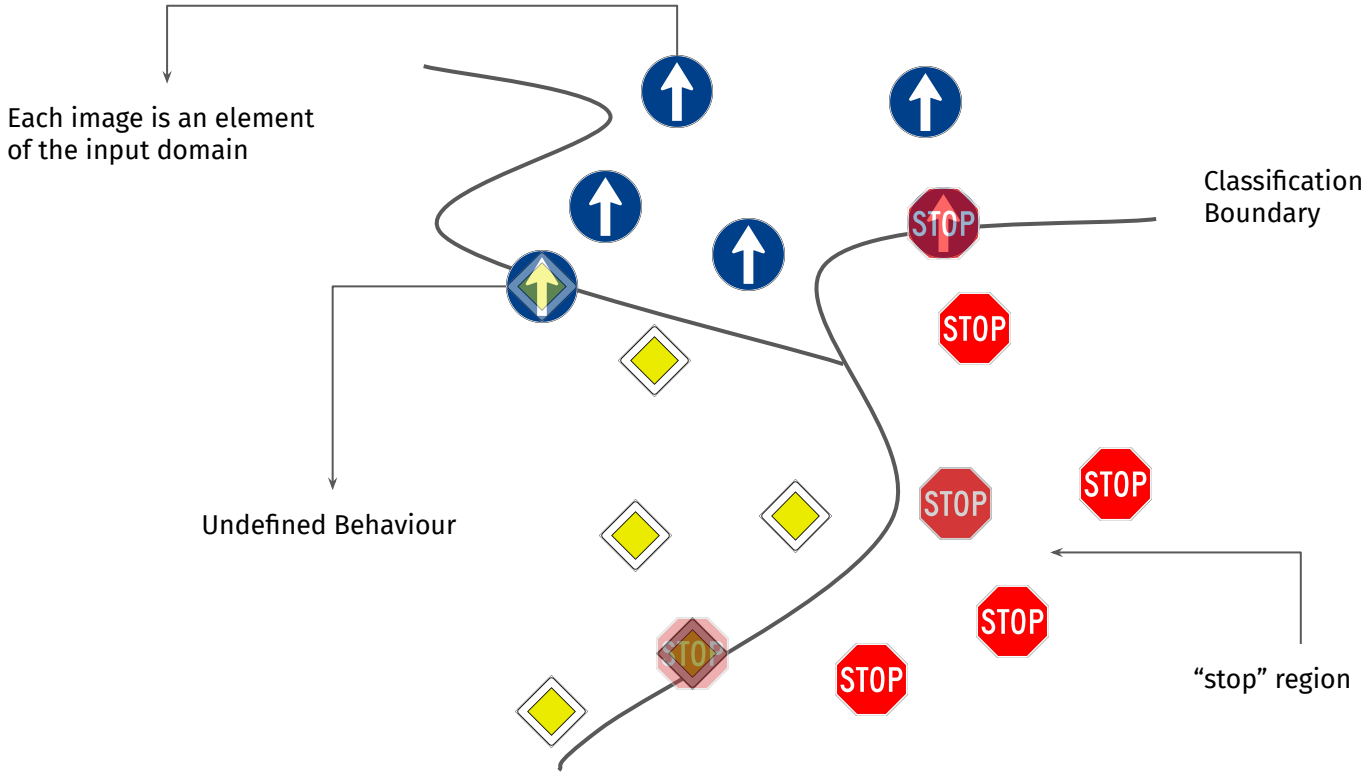
Encoding Table

0	
1	
2	
4	STOP

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

Classifier: A Geometrical Intuition

Classifiers divide the whole input domain in regions



Adversarial Examples: Lack of Robustness or Security Threat?

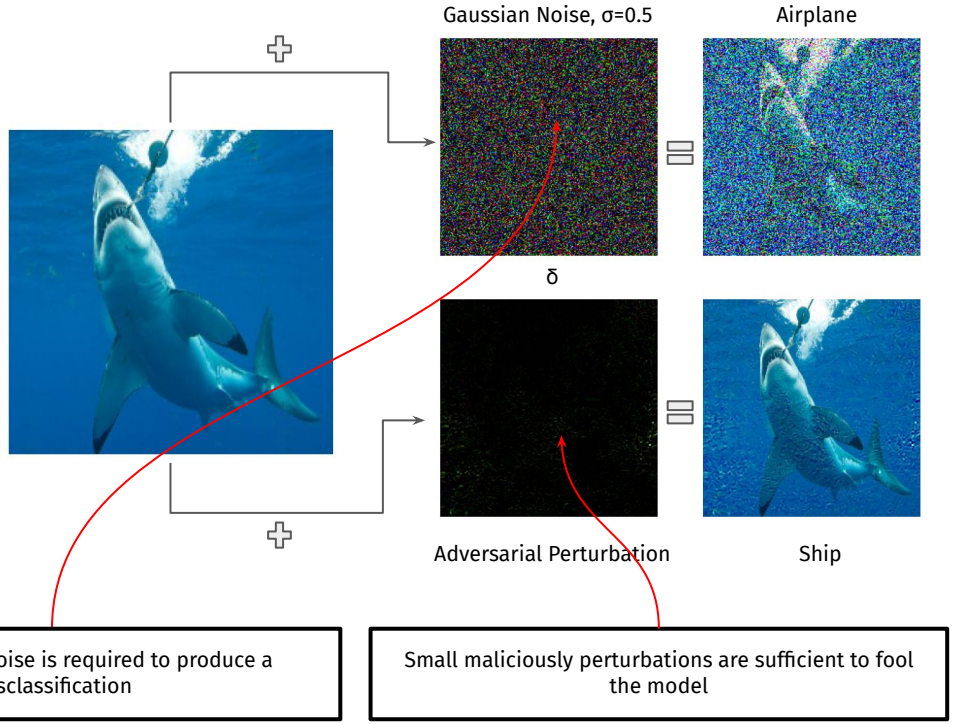
Adversarial Examples as a proof of lack of robustness

Decision Boundaries are close to training/testing samples.

*

MNIST	FMNIST	CIFAR10	GTSRB
0.0418	0.0091	0.0088	0.0250

*Average Magnitude of **Smallest Adversarial Perturbation** (l_2 / √d). Gaussian Random noise with σ=0.5 is 0.5 in average.*



* **Brau F.**, Rossolini G., **Biondi A., Buttazzo G.**, "***On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error***". TPAMI, 2022.

Adversarial Examples in Classification

Focus of the Lecture. Robustness of DDNs under **digital** 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

Minimal Adversarial Perturbation (Binary Case)

Binary classification from $f : \mathbb{R}^n \to \mathbb{R}$

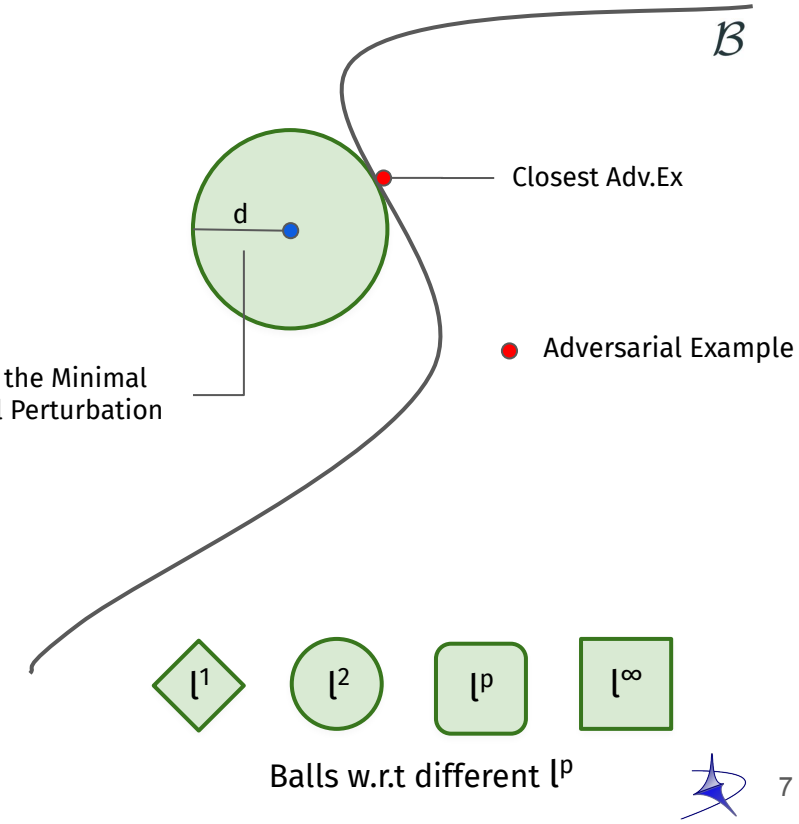
$$\mathcal{K}_f(x) = \begin{cases} -1, & \text{if } f(x) < 0 \\ 1, & \text{if } f(x) > 0 \end{cases}$$

Classification Boundary

$$\mathcal{B} = \{p \in \mathbb{R}^n : f(p) = 0\}$$

Minimal Adversarial Perturbation

$$d(x) = \min_{p \in \mathcal{B}} \|x - p\|$$



Minimal Adversarial Perturbation (General Case)

Classification from $f : \mathbb{R}^n \rightarrow \mathbb{R}^C$

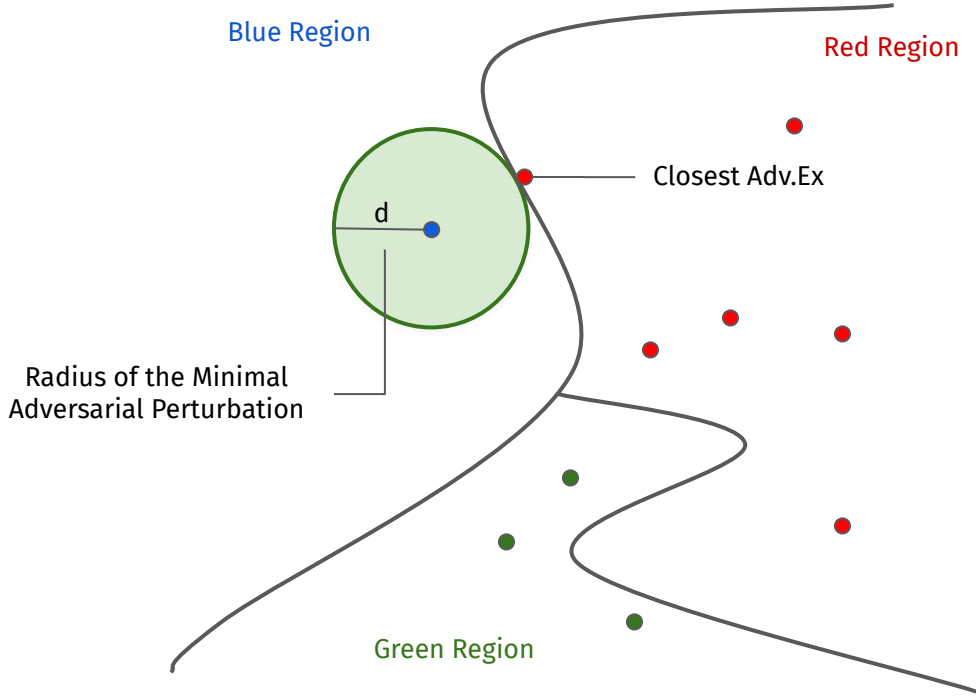
$$\mathcal{K}_f(x) = \underset{i}{\operatorname{argmax}} f_i(x)$$

Minimal Adversarial Perturbation

$$d(x, l) = \min_{\delta} \quad \|\delta\|$$

$$\text{s.t} \quad f_l(x+\delta) - \max_{j \neq l} f_j(x+\delta) \leq 0$$

where l *is the class of* x



Minimal Adversarial Perturbation

Observation. General case falls into the Binary case

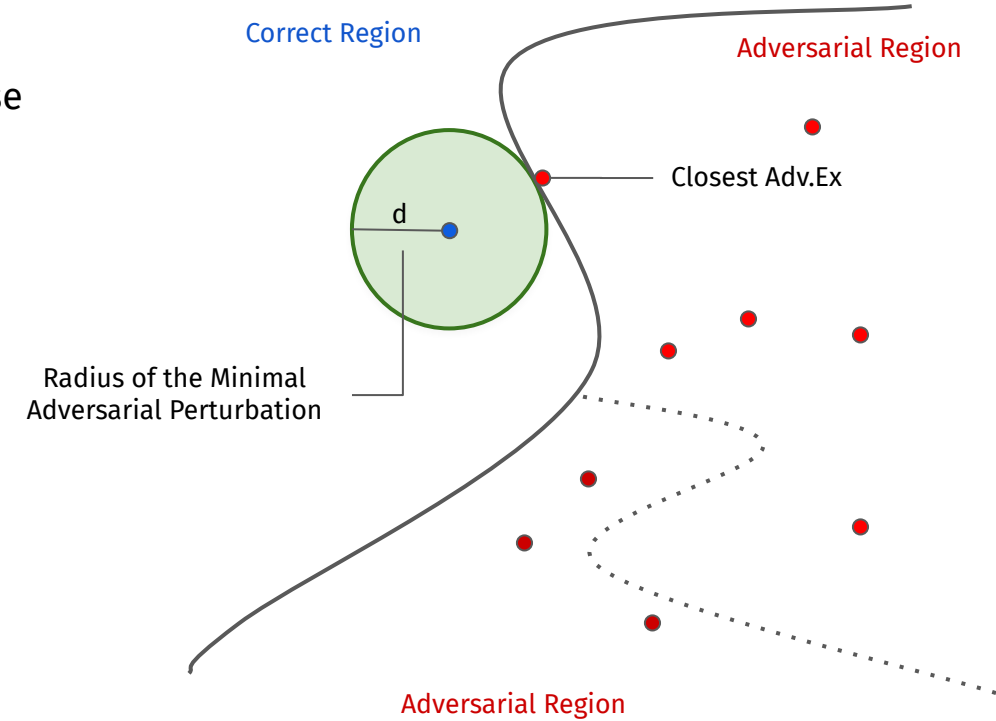
$$F_l(x) = f_l(x) - \max_{j \neq l} f_j(x)$$

provide a binary classification and

$$d(x, l) = d_l(x)$$

MAP of Multiclass

MAP of F_l



Certifiable ε-Robust Classification

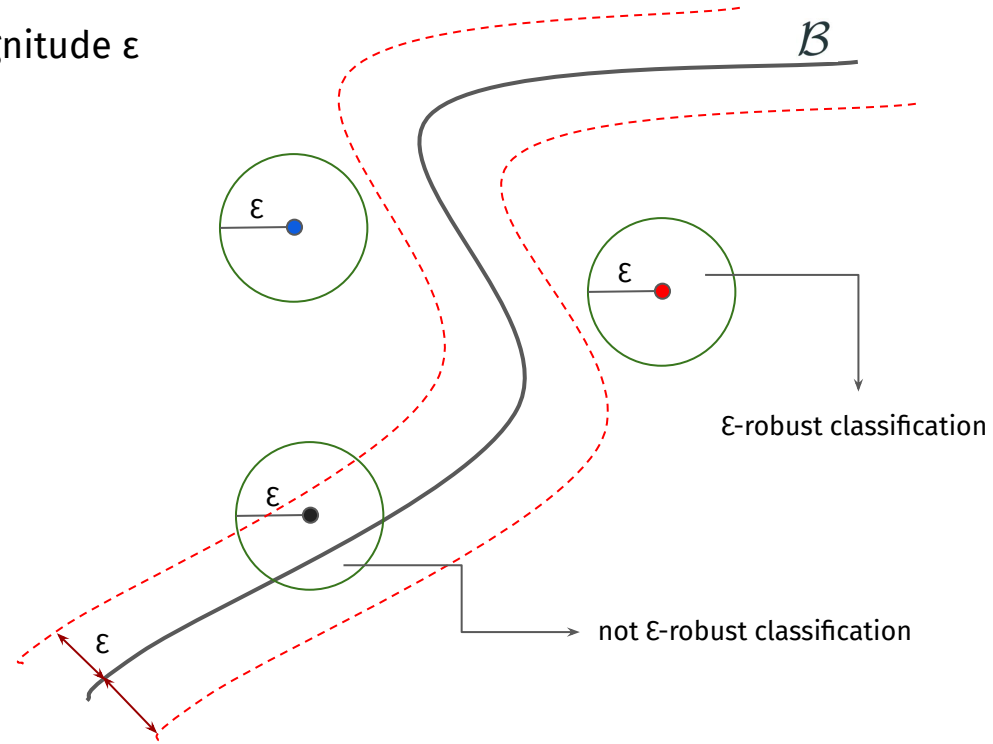
Definition. (Robustness in l^p norm)
Robust Classification under perturbation of magnitude ε

$\mathcal{K}(x)$ is a ε-robust classification if

$$\|\delta\| < \varepsilon \Rightarrow \mathcal{K}(x) = \mathcal{K}(x+\delta)$$

Remark.

1. "x is robust..." 🤔
2. "K is robust..." ❌
3. "The couple K, x is robust..." ✅



Robustness is a **local** property!!

Certifiable ε-Robust Classification by MAP computation

Minimal Adversarial Problem

$$d(x) = \min_{p \in \mathcal{B}} \|x - p\|$$

$\mathcal{K}(x)$ is a d(x)-robust classification

Method [a]	Solution	Guarantees	# Inferences
L-BFGS [b]	Accurate	✓	> 10k (slow)
CW [c]	Accurate	✓	≈ 10k (slow)
DeepFool [d]	Approximated	✗	≈ 20 (fast)
DDN	Approximated	✗	≈ 1k (fast)

Exact methods are computationally expensive

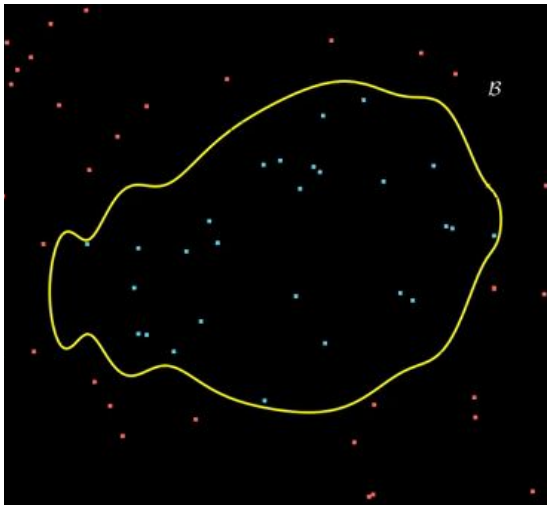
Approximated methods don't provide theoretical guarantees

a Szegedy, Christian, et al. "**Intriguing properties of neural networks.**" arXiv preprint arXiv:1312.6199 (2013).
b Carlini, Nicholas, and David Wagner. "**Towards evaluating the robustness of neural networks.**" 2017 ieee symposium on security and privacy (sp). IEEE, 2017.
c Moosavi-Dezfooli, et al. "**Deepfool: a simple and accurate method to fool deep neural networks.**" CVPR. 2016.
d Rony, Jérôme, et al. "**Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses.**" CVPR. 2019.

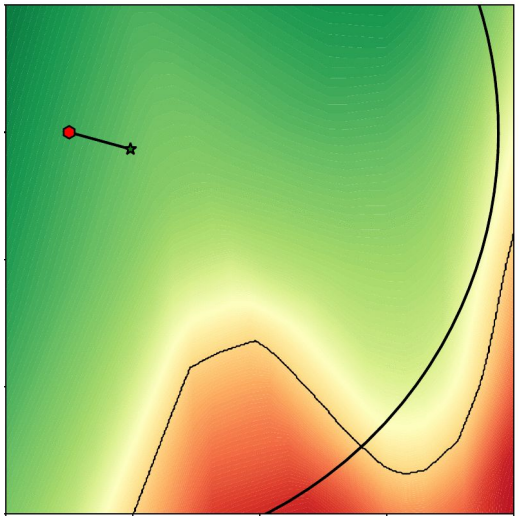
Certifiable ε-Robust Classification by MAP computation

MAP can be estimated by following the gradient direction

Fast Bisection [b]



FMN Strategy [a]



[a] Maura Pintor et al. *"Fast minimum-norm adversarial attacks through adaptive norm constraints"*

[b] Brau, Rossolini, Biondi, Buttazzo. *"On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error"*.

Verification Methods

Definition and Introduction

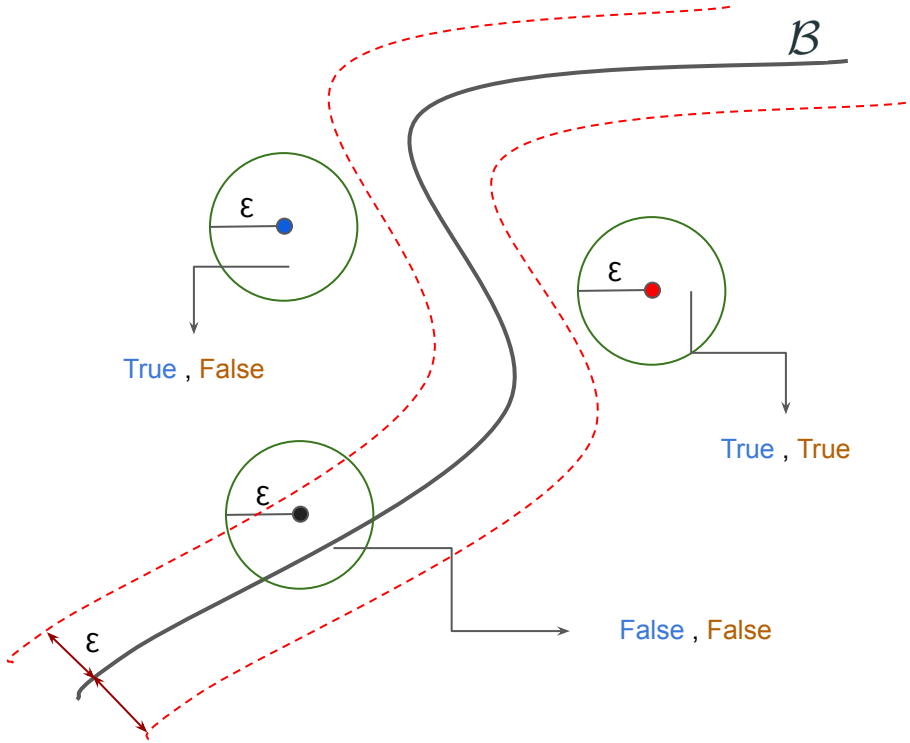
Definition. (Verification of a Classifier)
Given a classifier $\mathcal{K}$ and a sample x, check whether

$$\zeta(x): \quad "\forall y \in \mathcal{N}(x) \quad \mathcal{K}(x) = \mathcal{K}(y)"$$

where $\mathcal{N}$ is a neighborhood of x

Definition. (Complete and Incomplete Verifier)

$\zeta(x)$	True	False
Complete	True	False
Incomplete	True/False	False

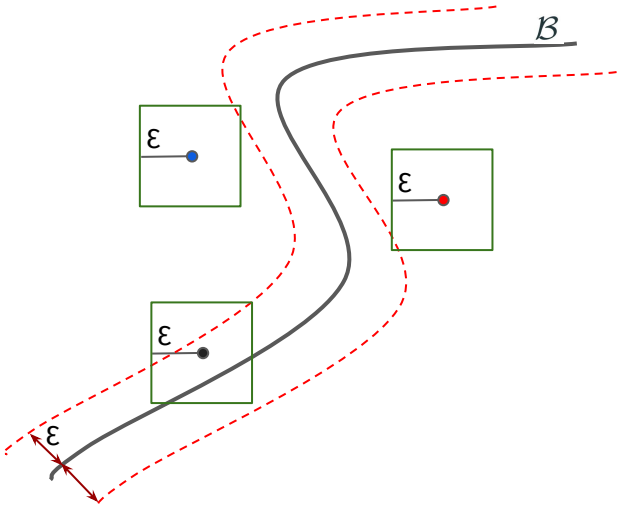


Is it the Complete problem easy?

Theorem.
Let us assume f a ReLU Deep Neural Network, and

$$\mathcal{N}(x) = \{y \in \mathbb{R}^n \ : \ \|y - x\|_\infty \leq \varepsilon\}$$

then completely check $\zeta(x)$ is **NP-HARD**



Minimum Problem Formulation.

$$P(x) = \min_{j \neq l} \min_{y \in \mathbb{R}^n} \quad f_l(y) - f_j(y)$$

$$\text{s.t.} \quad -\varepsilon \leq x_i - y_i \leq \varepsilon, \forall i$$

Linear Constraint

Remark! $\zeta(x)$: "$\forall y \in \mathcal{N}(x) \quad \mathcal{K}(x) = \mathcal{K}(y)$"

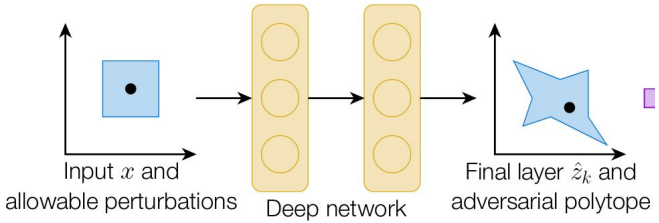
$$\zeta(x) \quad \Leftrightarrow \quad P(x) > 0$$

Guy Katz et al. "*Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks*"

Incomplete Verifiers

Deep Neural Network with ReLU

$$\hat{z}^{(i)} = W_i z^{(i)} + b_i \qquad i = 1, \cdots, L-1$$
$$z^{(i)} = \max\{0, \hat{z}^{(i)}\} \qquad i = 2, \cdots, L-1$$



Minimum Problem Formulation.

$$P(x) = \min_{j \neq l} \min_{y \in \mathbb{R}^n} \quad f_l(y) - f_j(y)$$
$$\text{s.t.} \quad -\varepsilon \leq x_i - y_i \leq \varepsilon, \forall i$$

Formulation with Inequality and Equality Constraints

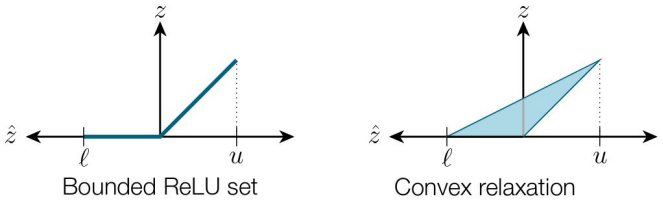
$$P(x) = \min_{j \neq l} \min_{y \in \mathbb{R}^n} \quad \hat{z}_l^{(L)} - \hat{z}_j^{(L)}$$
$$\text{subject to} \quad -\varepsilon \leq x - z^{(0)} \leq \varepsilon$$
$$\hat{z}^{(i)} = W_i z^{(i)} + b_i, \quad i = 1, \cdots, L-1$$
$$z^{(i)} = \max\{0, \hat{z}^{(i)}\}, \quad i = 2, \cdots, L-1$$

NON Linear Constraint

Eric Wong and Zico Kolter. "*Provable Defenses via the Convex Outer Adversarial Polytope*"

Relaxing Strategy

Convex Relaxation of ReLU



$z = \max\{0, \hat{z}\}$ relaxed to

$$z \geq 0$$
$$z \geq \hat{z}$$
$$-u\hat{z} + (u - l)z \leq -ul$$

Relaxed Minimum Problem

$$P(x) = \min_{j \neq l} \min_{y \in \mathbb{R}^n} \quad \hat{z}_l^{(L)} - \hat{z}_j^{(L)}$$

subject to $-\varepsilon \leq x - z^{(0)} \leq \varepsilon$

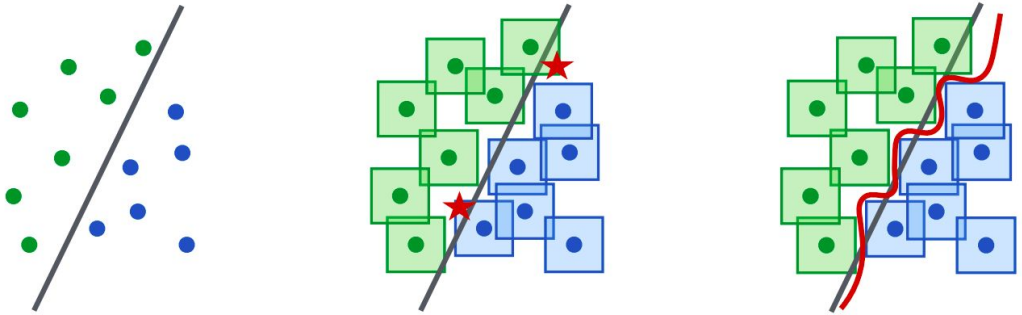
$$\hat{z}^{(i)} = W_i z^{(i)} + b_i, \quad i = 1, \cdots, L-1$$
$$z^{(i)} \geq 0, \quad i = 2, \cdots, L-1$$
$$z^{(i)} \geq \hat{z}^{(i)}, \quad ''$$
$$-u^{(i)}\hat{z}^{(i)} + (u^{(i)} - l^{(i)})z^{(i)} \leq -u^{(i)}l^{(i)}, \quad ''$$

Relaxed Linear Constraints

Eric Wong and Zico Kolter. "*Provable Defenses via the Convex Outer Adversarial Polytope*"

Robust Training

Remark. Robust training $\not\Rightarrow$ Certifiable Robust Classification (but can be helpful)



Robust Minimization Problem.

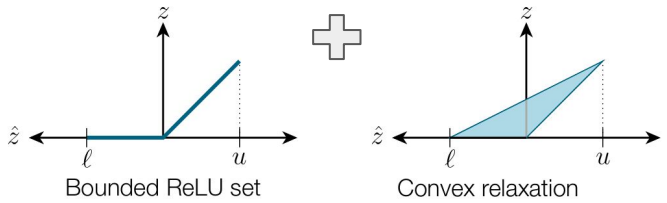
$$\theta^* \in \operatorname*{argmin}_{\theta} \frac{1}{N} \sum_{i=1}^{N} \max_{\|\delta\| \leq \varepsilon} L(f_\theta(x_i + \delta), y_i)$$

Robust Loss Function

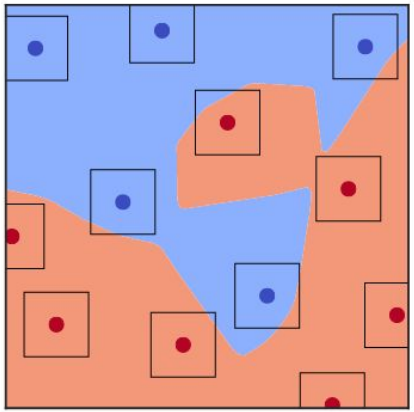
Madry et al. "*Towards Deep Learning Models Resistant to Adversarial Attacks*"

Convex Relaxed Robust Minimum Problem (no details)

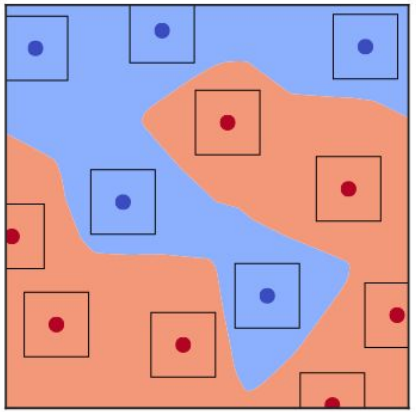
$$\theta^* \in \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \max_{\|\delta\| \leq \varepsilon} L(f_\theta(x_i + \delta), y_i)$$



The convex relaxation provides a *suboptimal* solution



Standard Training



Robust Training

Eric Wong and Zico Kolter. "*Provable Defenses via the Convex Outer Adversarial Polytope*"

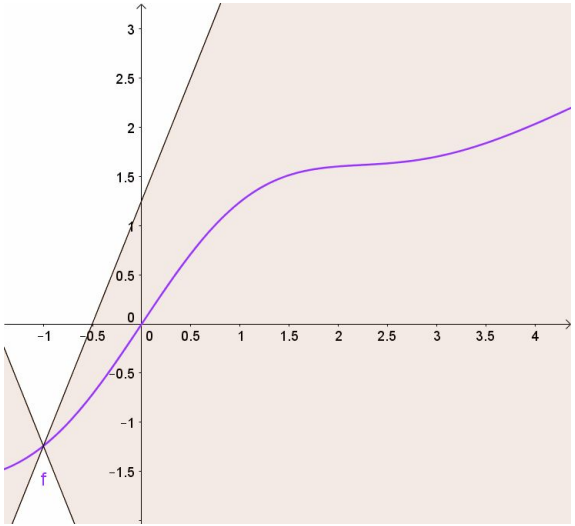
Verification by Estimating the Lipschitz Constant

Definition. (L-Lipschitz)
A function f is L-lipschitz w.r.t. p-norm if

$$\forall x, y \in \mathbb{R}^n, \quad \|f(x) - f(y)\|_p \leq L\|x - y\|_p$$

Definition. (Local L-Lipschitz)
A function f is local L-lipschitz w.r.t. p-norm if

$$\forall \delta, \|\delta\|_p \leq \varepsilon, \quad \|f(x) - f(x + \delta)\|_p \leq L\|\delta\|_p$$



The curve's slope is always lower than L.

Gif taken from https://en.wikipedia.org/wiki/Lipschitz_continuity#/media/File:Lipschitz_Visualisierung.gif

Verification by Estimating the (local) Lipschitz Constant

Theorem. (Lower bound of MAP)
Let us assume f be local L-lipschitz in a large radius R.

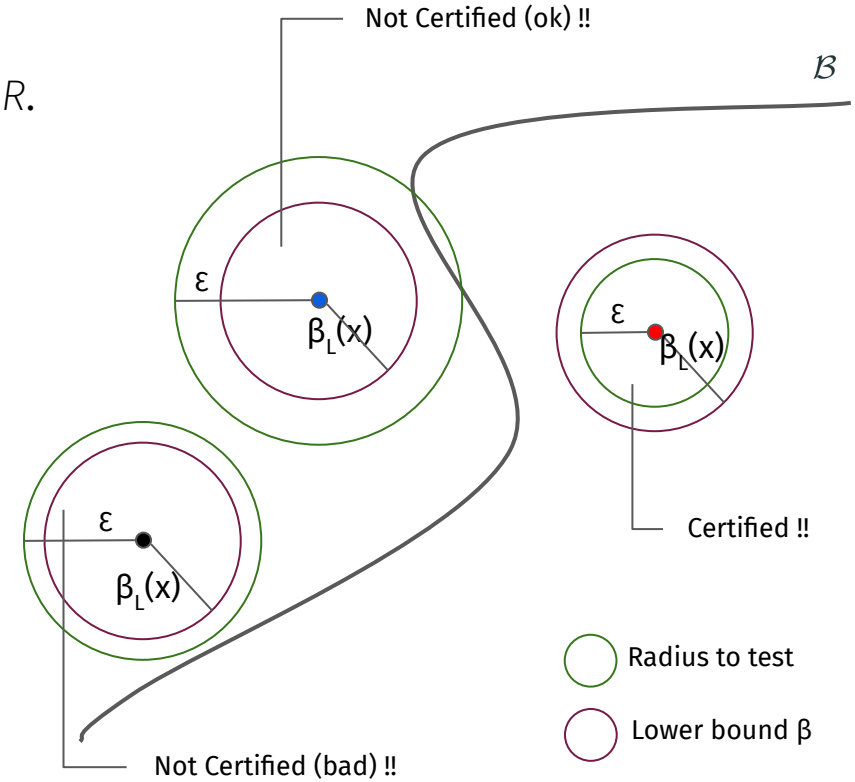
$$\beta_L(x) = \min_{j \neq l} \frac{f_l(x) - f_j(x)}{L\, 2^{\frac{p-1}{p}}}$$

is a bound of the *Minimal Adversarial Perturbation*

Remark! $\zeta(x)$: $\quad "\forall y \in \mathcal{N}(x) \quad \mathcal{K}(x) = \mathcal{K}(y)"$

$$"\varepsilon < \beta_L(x)" \Rightarrow \zeta(x)$$

Incomplete Verification



Qiyang Li et al. "*Preventing Gradient Attenuation in Lipschitz Constrained Convolutional Networks*"

Estimating L by the gradient of f

Cross Lipschitz Constant

Ball of radius R w.r.t p-norm

Theorem. (Cross Lipschitz Bound)
Let $f : \mathbb{R}^n \to \mathbb{R}^C$, x classified as l, and

$f_l - f_j$ is L_j-lipschitz in $B_p(x, R)$

then

$$\beta_L(x) = \min_{j \neq l} \frac{f_l(x) - f_j(x)}{L_j}$$

is still a lower bound of MAP.[a]

Theorem. (Gradient Approximation)
Approximation of the Cross-Lipschitz

$$L_j = \max_{y \in B_p(x,R)} \|\nabla f_l(y) - \nabla f_j(y)\|_q$$

where q is the dual number $\frac{1}{p} + \frac{1}{q} = 1$

[a] Hain and Andriushchenko. "*Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation*"

CLEVER

Cross **L**ipschitz **E**xtreme **V**alue for n**E**twork **R**obustness

Maximum Problem

$$L_j = \max_{y \in B_p(x,R)} \|\nabla f_l(y) - \nabla f_j(y)\|_q$$

Idea. Estimate the maximum problem with multiple samplings.

for $i \leftarrow 1$ **to** N_b **do**
  for $k \leftarrow 1$ **to** N_s **do**
    randomly select a point $\boldsymbol{x}^{(i,k)} \in B_p(\boldsymbol{x_0}, R)$
    compute $b_{ik} \leftarrow \|\nabla g(\boldsymbol{x}^{(i,k)})\|_q$ via back propagation
  end
  $S \leftarrow S \cup \{\max_k\{b_{ik}\}\}$
end

MAP estimation

	CW		I-FGSM		CLEVER	
	ℓ_2	ℓ_∞	ℓ_2	ℓ_∞	ℓ_2	ℓ_∞
MNIST-MLP	1.113	0.215	3.564	0.178	0.819	0.041
MNIST-CNN	1.500	0.455	4.439	0.288	0.721	0.057
MNIST-DD	1.548	0.409	5.617	0.283	0.865	0.063
MNIST-BReLU	1.337	0.433	3.851	0.285	0.833	0.065
CIFAR-MLP	0.253	0.018	0.885	0.016	0.219	0.005
CIFAR-CNN	0.195	0.023	0.721	0.018	0.072	0.002
CIFAR-DD	0.285	0.032	1.136	0.024	0.130	0.004
CIFAR-BReLU	0.159	0.019	0.519	0.013	0.045	0.001

Takes out.

1. Computationally expensive.
2. Not accurate

Tsui-Wei Weng et al. “Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach”

Summary: Verification Methods

Pros

1. Highly Reliable since they are based on the solution of well founded minimum problem
2. Can be involved in a training process to improve the (empirical) robustness of a model classification

Cons

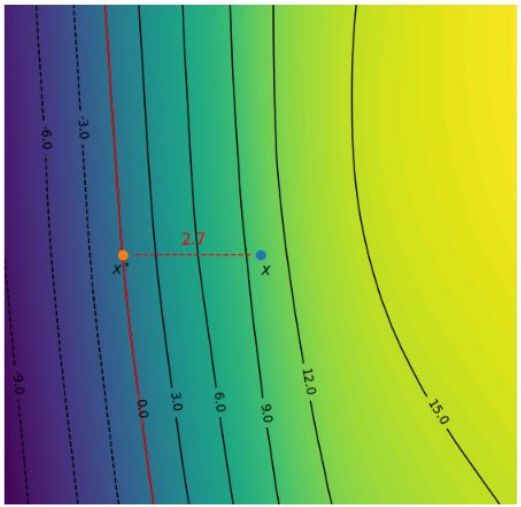
1. Do not scale to Larger Networks or are typically computational expensive
2. Can require a complete knowledge of the inner states of the model

Lipschitz Bounded Neural Networks

Lipschitz constant of Neural Networks

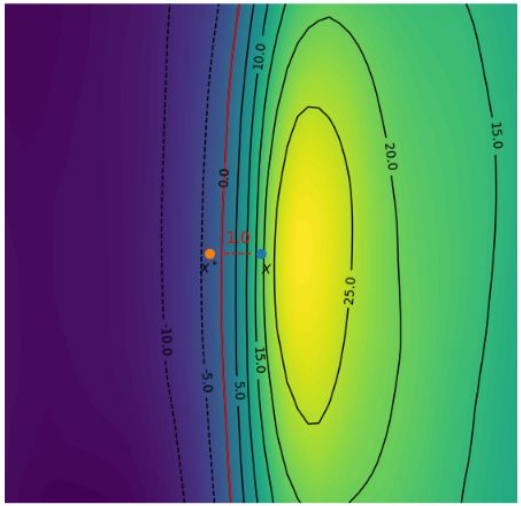
Vanilla Neural Networks feature, a **large** local Lipschitz constant

LeNet-5



(a) MNIST

ResNet-32



(d) CIFAR10

Contour plot of $F_l(x) = f_l(x) - \max_{j \neq l} f_j(x)$ generated with two random orthogonal directions in the input domain.

Lipschitz constant of Neural Networks

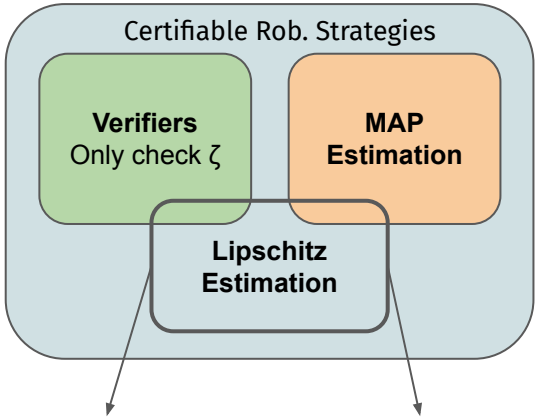
Observation.
Feedforward Neural Networks with *linear*, *convolutional* and *residual* layers are L-Lipschitz for some constant L.

MLP
LeNet
ResNet
U-Net
} L-Lipschitz

Standard trainings don't care about the Lipschitz const.

Neural Model	Random Init.	Trained
ResNet32 (CIFAR10)	$2.39\,10^8$	`inf`
AlexNet (ImageNet)	0.78	$3.68\,10^7$
LeNet (MNIST)	2.13	$3.09\,10^2$
LeNet (FMNIST)	2.13	$4.88\,10^3$
MicronNet (GTSRB)	0.84	`inf`

Concept schema



1. Regularized training
2. Search for f, L-lipschitz

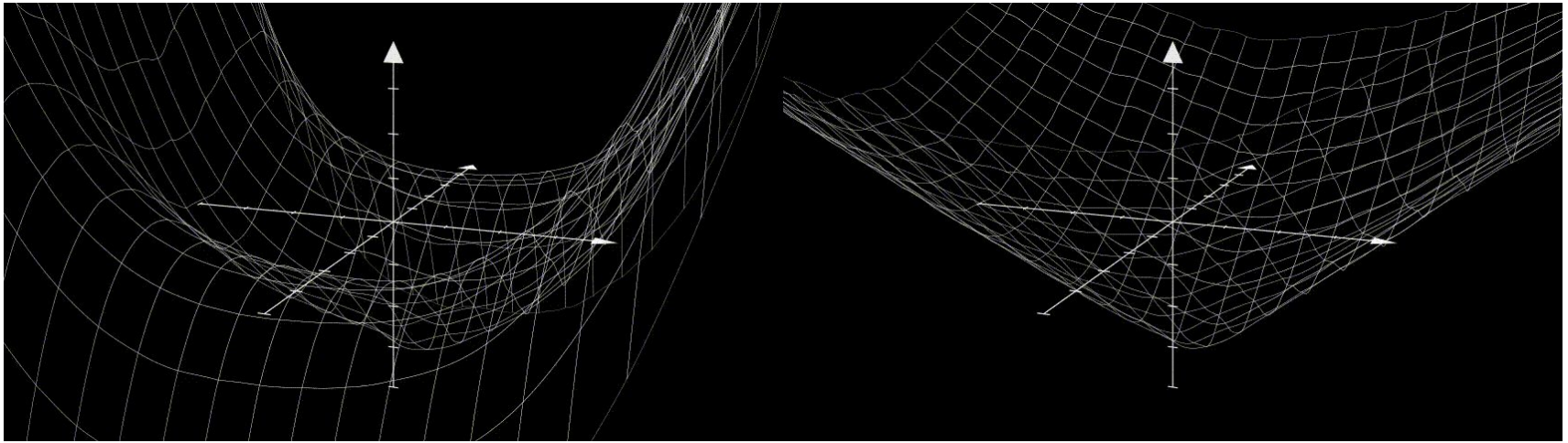
A. Exact computation
B. Local Estimation

Aim of the Section

Can 1-Lipschitz Neural be good classifier?

Vanilla Binary Classifier

1 Lipschitz Neural Network



Graphical representation of level curves of a 1-lipschitz function.
The 0-level curve is the decision boundary and it is the same for both functions

Lipschitz Functions

Definition. (L-Lipschitz)
A function f is L-lipschitz w.r.t. p-norm if

$$\forall x, y \in \mathbb{R}^n, \quad \|f(x) - f(y)\|_p \leq L\|x - y\|_p$$

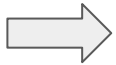
Examples of Lipschitz Layers.*

Fully connected
Convolutional
Residual
Pooling

Property. (Composition)

Composition is L-lipschitz

$$f = \underbrace{f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}}_{L = \prod_{i=1}^{L} L_i}$$



Common Deep Neural Networks* are (globally) Lipschitz

Remark.
Composition of 1-Lipschitz layers is 1-Lipschitz

Linear Layers are Lipschitz

Lipschitz Constant of Affine Functions

$$f(x) = Wx + b$$

Explicit computation.

$$\|f(y) - f(x)\|_p = \|Wy - Wx + b - b\|_p$$

$$(y - x = v)$$

$$\|f(y) - f(x)\|_p = \|Wv\|_p$$

$$\frac{\|f(y) - f(x)\|_p}{\|y - x\|_p} = \frac{\|Wv\|_p}{\|v\|_p} \leq \sup_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|Wv\|_p}{\|v\|_p}$$

$$\|f(y) - f(x)\|_p = \|W\|_p \, \|y - x\|_p$$

Definition. (Operatorial Norm)

$$W \in \mathbb{R}^{m \times n}, \quad \|W\|_p := \sup_{v \in \mathbb{R}^n \setminus \{0\}} \frac{\|Wv\|_p}{\|v\|_p}$$

1. $p=\infty$, Uniform Norm
2. $p=2$, Spectral Norm

Conclusion.

The layer f is $\|W\|_p$ -lipschitz

1-Lipschitz Linear Layers

Spectral Norm

$$\|A\|_2 = \sqrt{\lambda_{max}(A^T A)} = \sigma_{max}(A)$$

The largest singular value of A

Spectral Normalization

$$f_W(x) = \tilde{W}x + b, \quad \text{where} \quad \tilde{W} = \frac{W}{\|W\|_2}$$

The weight is **parameterized** through W.

The layer is 1-Lipschitz.

Power method

```
def NormalizedFC(x, W, b):
    ## Normalization Procedure
    # Pick a random vector
    aux = torch.randn(W.dim(-1))

    #Iterate few times
    for _ in range(max_iterations):
        aux = matmul(W, aux)
        sigma = aux.norm(2)
        aux /= sigma

    # Parameterize the weight
    W_tilde = W / sigma
    return matmul(W_tilde, x ) + b
```

NormalizedFC is differentiable in W

Takeru Miyato et al. "*Spectral Normalization for Generative Adversarial Networks*"

Orthogonal Linear Layers

Orthogonal Matrix

$$Q \in \mathbb{R}^{n \times n} \quad QQ^T = Q^T Q = I$$

Observation.
By construction,

$$f_Q(x) = Qx + b$$

is 1-Lipschitz w.r.t euclidean norm.

Remark 1.
A fully-connected DNN with orthogonal layers is 1-Lipschitz

Bjorck Orthogonalization

$$Q_k = I - W_k^T W_k$$

$$W_{k+1} = W_k \left(I + \frac{1}{2} Q_k + \frac{3}{8} Q_k^2 + \cdots + (-1)^p \binom{-\frac{1}{2}}{p} Q_k^p \right)$$

where $W_0 = W$.

Remark 2.
The parameterized weight Q_k, depends in a differentiable manner from W.

Cem Anil et al. "*Sorting Out Lipschitz Function Approximation*"

Orthogonal Linear Layers

Cayley Transformation[a]

$$A = W - W^T$$

$$Q = (I - A)(I + A)^{-1}$$

Exponential Map[b]

$$A = W - W^T$$

$$Q = \exp(A) := \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

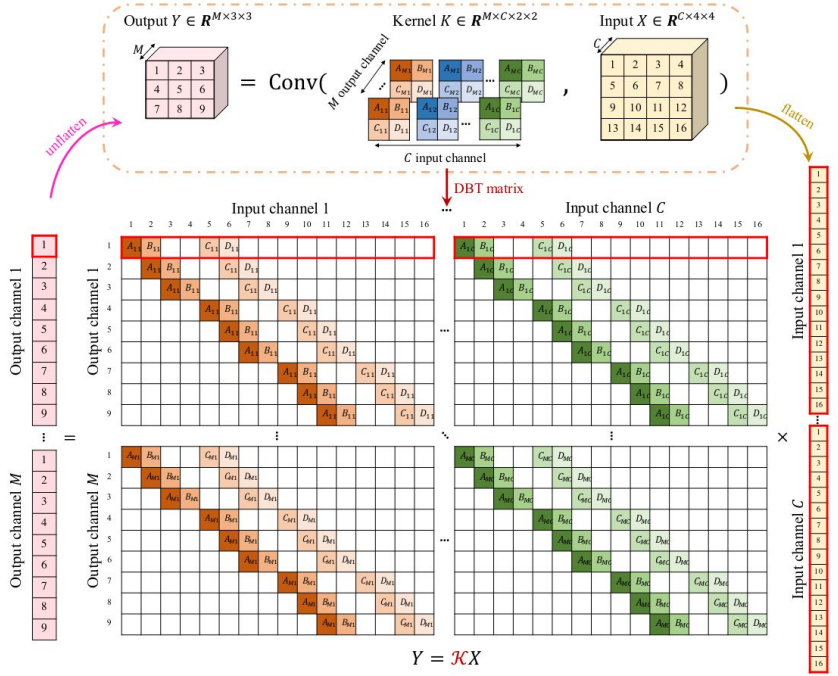
[a] Asher Trockmann et al. *"Orthogonalizing Convolutional Layers with the Cayley Transform"*

[b] Sahil Singla. *"Skew Orthogonal Convolutions"*.

1-Lipschitz Convolutions

Remark.
Convolutions are particular (sparse) Linear Layers. Can be represented through a **double-block Toeplitz** matrix structure.

Observation.
Convolutions are Lipschitz.



Wang et al. "*Orthogonal Convolutional Neural Networks*"

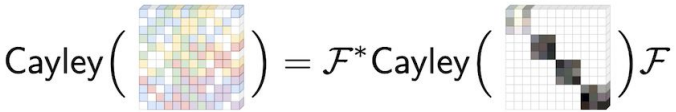
Orthogonal Convolutions (no details)

Cayley Transformation[a]

$$A = W - W^T$$

$$Q = (I - A)(I + A)^{-1}$$

Orthogonalizing a **multi-channel convolution**...

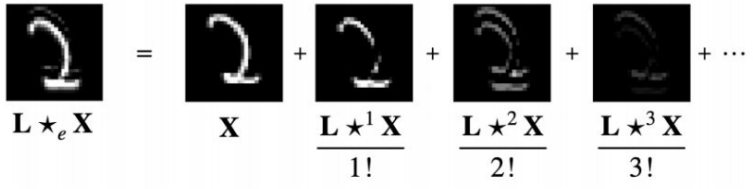


...can be done *efficiently* by orthogonalizing a Fourier-domain **block-diagonal matrix**.

Exponential Map[b]

$$A = W - W^T$$

$$Q = \exp(A) := \sum_{k=0}^{\infty} \frac{A^k}{k!}$$



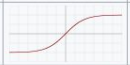
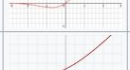
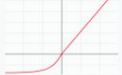
(c) Convolution exponential $(\mathbf{L} \star_e \mathbf{X})$

[a] Asher Trockmann et al. *"Orthogonalizing Convolutional Layers with the Cayley Transform"*

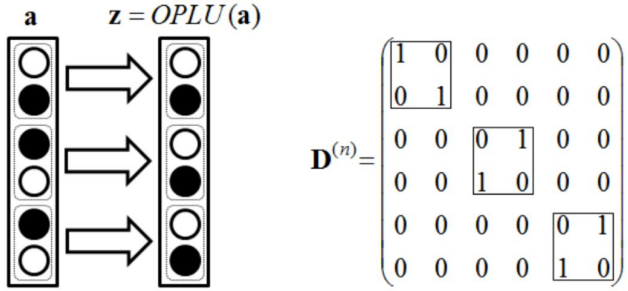
[b] Sahil Singla. *"Skew Orthogonal Convolutions"*.

1-Lipschitz Activation Functions

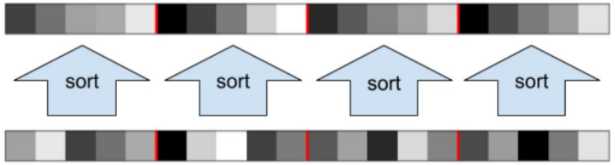
Component-wise Activations

Name	Plot	Function, $g(x)$	Derivative of g, $g'(x)$	Range	Order of continuity
Identity		x	1	$(-\infty, \infty)$	C^∞
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	0	$\{0, 1\}$	C^{-1}
Logistic, sigmoid, or soft step		$\sigma(x) \doteq \frac{1}{1+e^{-x}}$	$g(x)(1-g(x))$	$(0, 1)$	C^∞
Hyperbolic tangent (tanh)		$\tanh(x) \doteq \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - g(x)^2$	$(-1, 1)$	C^∞
Rectified linear unit (ReLU)[8]		$(x)^+ \doteq \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} = \max(0, x) = x\mathbf{1}_{x>0}$	$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$[0, \infty)$	C^0
Gaussian Error Linear Unit (GELU)[5]		$\frac{1}{2}x\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right) = x\Phi(x)$	$\Phi(x) + x\phi(x)$	$(-0.17\ldots, \infty)$	C^∞
Softplus[9]		$\ln(1 + e^x)$	$\frac{1}{1+e^{-x}}$	$(0, \infty)$	C^∞
Exponential linear unit (ELU)[10]		$\begin{cases} \alpha(e^x - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ with parameter α	$\begin{cases} \alpha e^x & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ 1 & \text{if } x = 0 \text{ and } \alpha = 1 \end{cases}$	$(-\alpha, \infty)$	$\begin{cases} C^1 & \text{if } \alpha = 1 \\ C^0 & \text{otherwise} \end{cases}$
Scaled exponential linear unit (SELU)[11]		$\lambda\begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$ with parameters $\lambda = 1.0507$ and $\alpha = 1.67326$	$\lambda\begin{cases} \alpha e^x & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$(-\lambda\alpha, \infty)$	C^0
Leaky rectified linear unit (Leaky ReLU)[12]		$\begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	$\begin{cases} 0.01 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$(-\infty, \infty)$	C^0

Not-component wise Activation



Orthogonal Permutation Linear Unit [a]



Group Sort Activation Function [b]

[a] Chernodub et al. "*Norm-preserving Orthogonal Permutation Linear Unit Activation Functions (OPLU)*"

[b] Cem Anil et al. "*Sorting Out Lipschitz Function Approximation*"

Evaluations

Definition. (Certifiable ε-Accuracy)
Given a dataset $(x_i, y_i) \in \mathcal{X}, \quad i = 1, \cdots, N$

$$\mathcal{A}_\varepsilon(\mathcal{X}) = \frac{1}{N} \# \{i \, : \, \mathcal{K}_f(x_i) = y_i, \, d(x_i) \geq \varepsilon\}$$

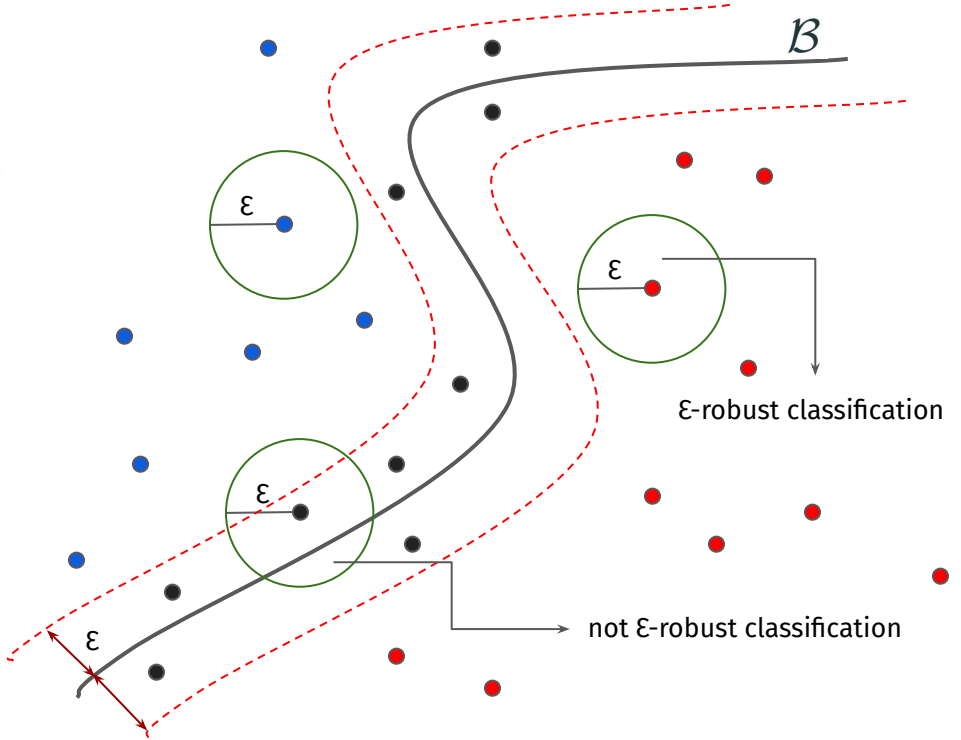
where d is the MAP.

Reminder. (Lower Bound of MAP)

$$\beta_L(x) = \min_{j \neq l} \frac{f_l(x) - f_j(x)}{L \, 2^{\frac{p-1}{p}}}$$

hence for lipschitz neural networks

$$\tilde{\mathcal{A}}_\varepsilon(\mathcal{X}) = \frac{1}{N} \# \{i \, : \, \mathcal{K}_f(x_i) = y_i, \, \beta(x_i) \geq \varepsilon\}$$



Theoretical Maximum ε

	plane	auto	bird	cat	deer	dog	frog	horse	ship
plane	0.00								
auto	7.56	0.00							
bird	3.93	6.98	0.00						
cat	4.97	7.43	5.12	0.00					
deer	4.43	6.93	4.22	5.22	0.00				
dog	5.88	7.84	4.73	6.03	4.64	0.00			
frog	4.53	7.45	3.60	5.15	4.64	5.51	0.00		
horse	5.48	7.85	5.00	6.29	5.59	6.23	5.51	0.00	
ship	4.22	6.82	5.13	5.78	4.85	5.63	4.68	6.22	0.00
truck	7.10	8.09	6.87	6.52	6.05	7.47	6.96	7.83	6.98

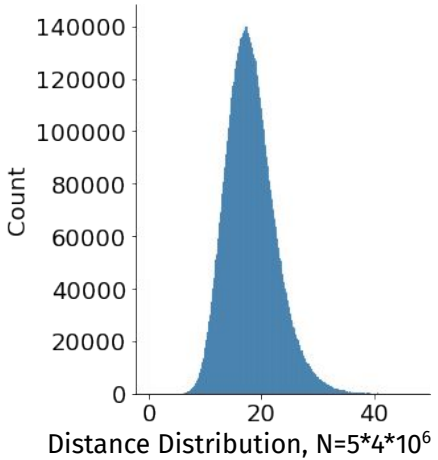
Two most close classes

Two most farther classes

Euclidean distances CIFAR10
Pairwise euclidean distances between testset images

Theoretical 100% Accuracy for

ε = 3.6/2



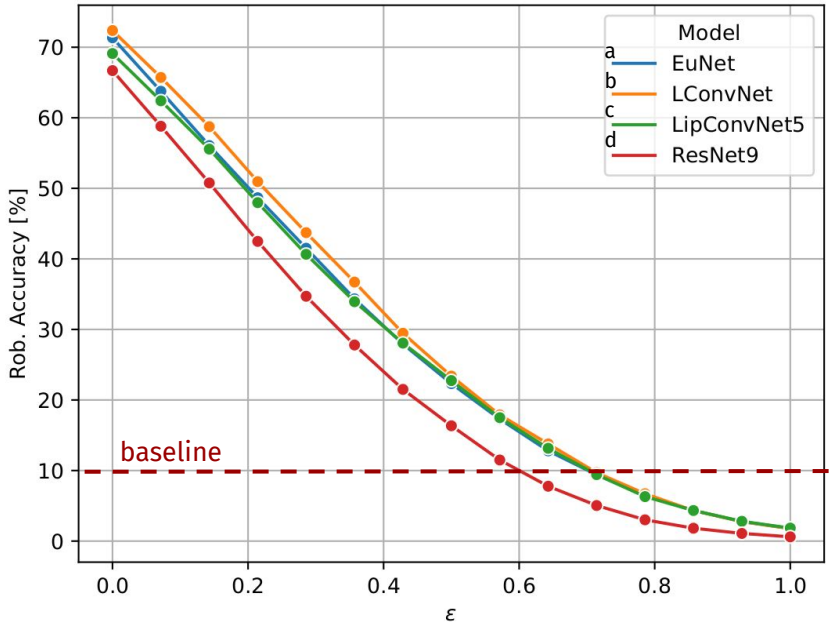
Experimental Results

Results on CIFAR 10

1. Increasing ε, the Certifiable Robust Accuracy drops
2. Even with small values of ε, the accuracy of Lipschitz models is particularly lower than the SOTA

Certifiable ε-Accuracy

$$\tilde{\mathcal{A}}_\varepsilon(\mathcal{X}) = \frac{1}{N}\#\{i : \mathcal{K}_f(x_i) = y_i,\ \beta(x_i) \geq \varepsilon\}$$



Robust Accuracy w.r.t 2-norm for different values of ε

[a] Fabio Brau, Giulio Rossolini, Alessandro Biondi and Giorgio Buttazzo., "*Robust-by-Design Classification with Unitary-Gradient Neural Networks*".
[b] Qiyang Li et al. "*Preventing Gradient Attenuation in Lipschitz Constrained Convolutional Networks*"
[c] Sahil Singla. "*Skew Orthogonal Convolutions*".
[d] Asher Trockmann et al. "*Orthogonalizing Convolutional Layers with the Cayley Transform*"

Conclusions

Pros.

1. Lipschitz Bounded Neural Networks allow certifiable classification at the cost of a **single forward step**

2. The forward of a model is not slower than a vanilla unbounded Neural Network

Cons.

1. Training of the models with orthogonal layers is **slower** than vanilla unbounded models

2. Accuracy is particularly low even with small ε, and does not match the SOTA

Randomized Smoothing

Randomized Smoothing Strategy

Base Classifier

$$f : \mathbb{R}^n \to \{1, \cdots, C\}$$

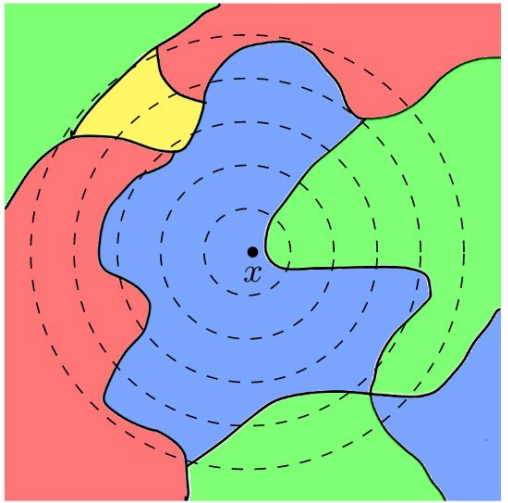
DNN, Decision Tree...

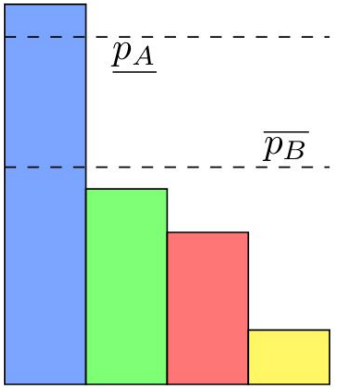
Smooth Classifier

$$g_\sigma : \mathbb{R}^n \to \{1, \cdots, C\}$$

$$g_\sigma(x) := \underset{c}{\operatorname{argmax}}\, \mathbb{P}_{\sim\mathcal{N}(0,\sigma I)}\{f(x+\varepsilon) = c\}$$

Most Probable perturbed sample's class





Left. Classification Regions of the base classifier
Right. Class frequency of perturbed sample x.

Jeremy Cohen et al. "*Certified Adversarial Robustness via Randomized Smoothing*"

Certifiable Robust Classification

Theorem.
Let consider the vector of probabilities P

$$P_c(x) = \mathbb{P}_{\sim\mathcal{N}(0,\sigma I)}\{f(x+\varepsilon) = c\}$$

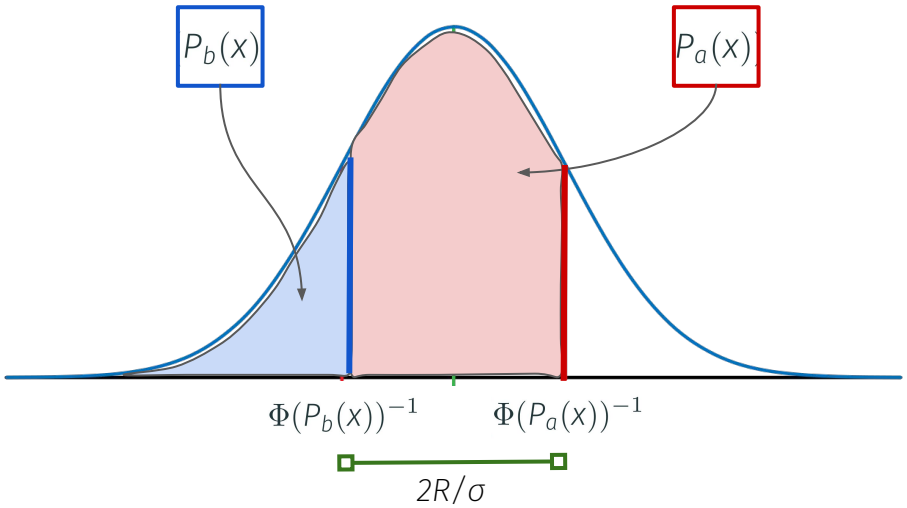
and let a and b the top-2 most probable classes.

Then $g_\sigma(x)$ is R-robust,

$$\forall \|\delta\| \leq R, \quad g_\sigma(x+\delta) = g_\sigma(x)$$

where

$$R = \frac{\sigma}{2}\left(\Phi(P_a(x))^{-1} - \Phi(P_b(x))^{-1}\right)$$



Φ Gaussian Cumulative Density Function

How to estimate the Smooth Classifier?

$$P_c(x) = \mathbb{P}_{\sim\mathcal{N}(0,\sigma I)}\{f(x+\varepsilon) = c\}$$ has no an explicit expression !!

Montecarlo Approach

Let $\varepsilon_1, \ldots, \varepsilon_n$ sampled from $\mathcal{N}(0,\sigma I)$

$$P_a(x) \geq \underline{P_a(x)} := \frac{\#\{i : f(x+\varepsilon_i) = a\}}{n}$$

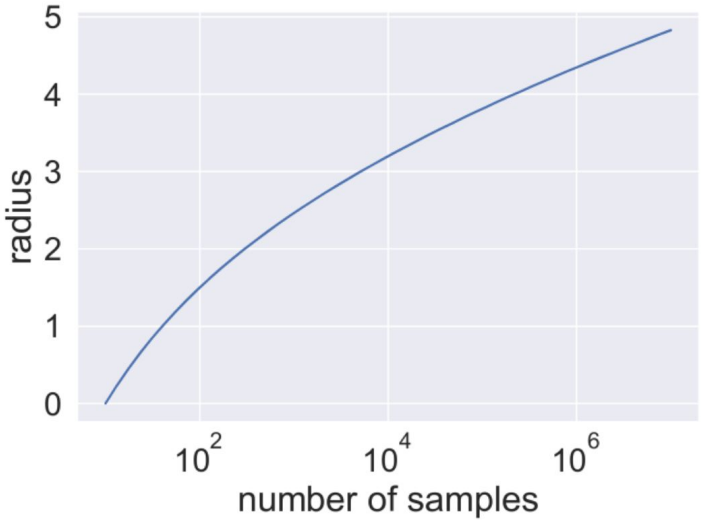
with confidence α

Computational Complexity

Larger radius require huge amount of samples

$R = 0.5 \approx 2\sigma$ with a confidence of 99.90%

requires evaluating ≈ 1000 samples



Jeremy Cohen et al. "*Certified Adversarial Robustness via Randomized Smoothing*"

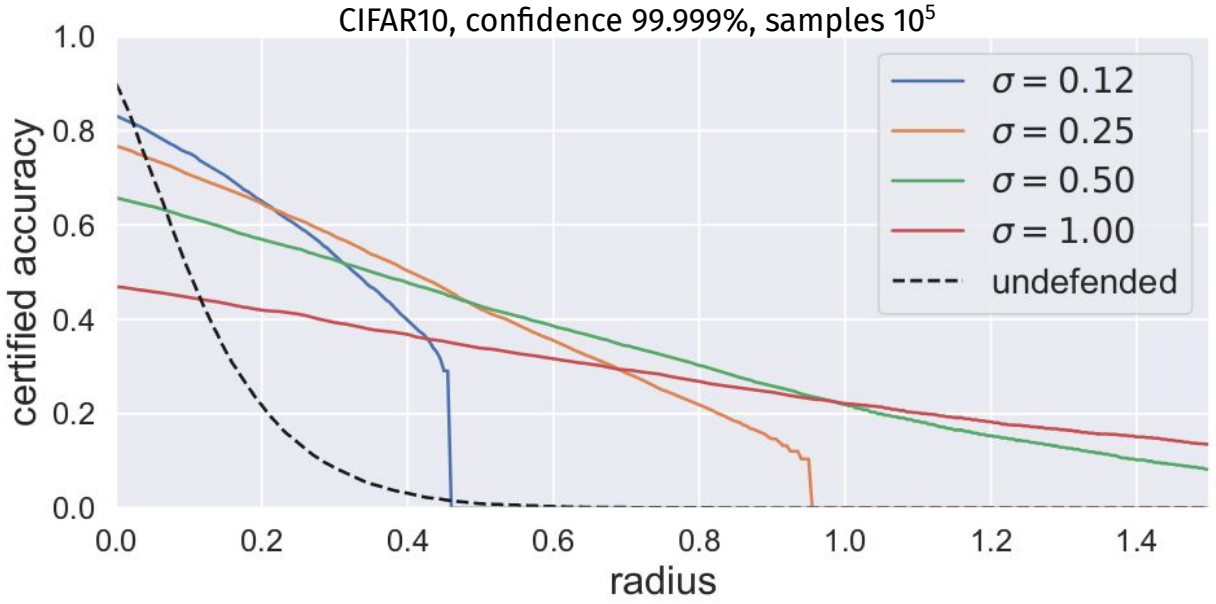
Certified Accuracy Evaluation

Memo (Certified Accuracy)

$$\mathcal{A}_\varepsilon(\mathcal{X}) = \frac{\#\{i : g_\sigma(x_i) = y_i, \quad R(x) \leq \varepsilon\}}{N}$$

where

$$R = \frac{\sigma}{2}\left(\Phi(P_a(x))^{-1} - \Phi(P_b(x))^{-1}\right)$$



Jeremy Cohen et al. "*Certified Adversarial Robustness via Randomized Smoothing*"

Smooth Adversarial Training

Base Classifier

$$f(x) := \operatorname{argmax}_i F_i(x)$$

Soft Smooth Classifier

$$G_\sigma(x) := \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma I)}[F(x+\varepsilon)]$$

Smooth Attack

$$\begin{aligned}\hat{x} &= \operatorname*{argmax}_{\|z-x\| \le \rho} \mathcal{L}_{CE}\left(G_\sigma(z), c\right) \\ &= \operatorname*{argmax}_{\|z-x\| \le \rho} \left(-\log \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma I)}\left[F(x+\varepsilon)_c\right]\right)\end{aligned}$$

Monte Carlo Gradient Estimation

Hadi Salman et al. "*Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers*"

Smooth Adversarial Training

Smooth Attack

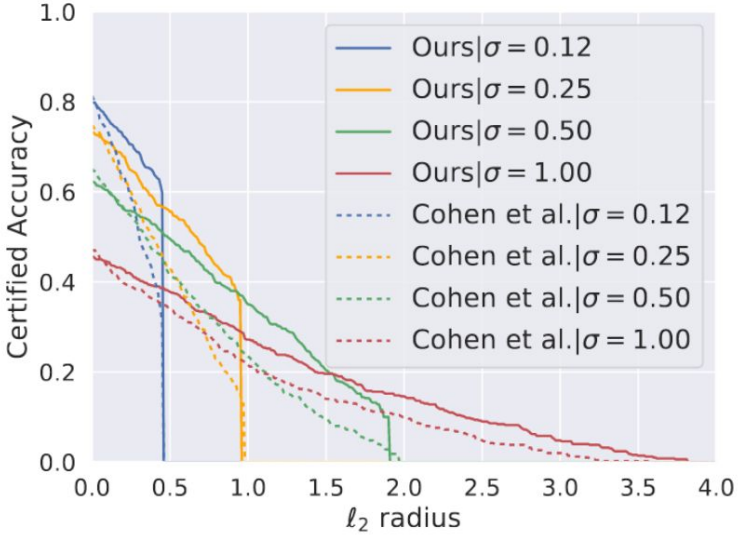
$$\hat{x} = \underset{\|z-x\| \le \rho}{\operatorname{argmax}} \mathcal{L}_{CE}\left(G_\sigma(z), c\right)$$

$$= \underset{\|z-x\| \le \rho}{\operatorname{argmax}} \left(-\log \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma I)}\left[F(z+\varepsilon)_c\right]\right)$$

Monte Carlo Gradient Estimation

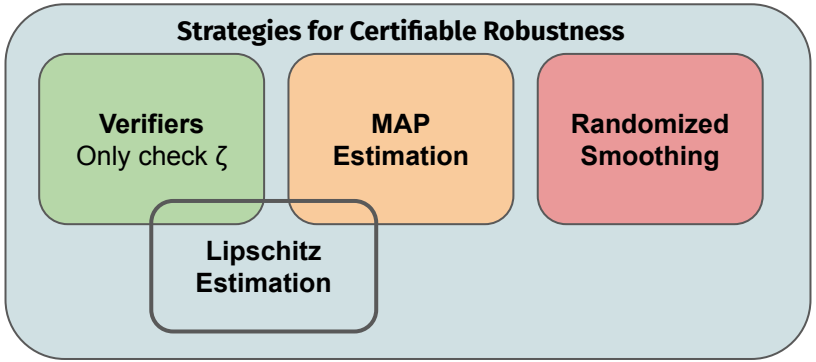
Gradient Estimation

$$\nabla_z \mathcal{L}_{CE}\left(G_\sigma(z), c\right) \approx -\nabla_z \log\left(\frac{1}{m}\sum_i F(z+\varepsilon_i)_c\right)$$



Hadi Salman et al. "*Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers*"

Conclusion



- Verification
- Local Lipschitz Estimation
- Lipschitz Bounded DNNs
- Randomized Smoothing

Thanks for the attention

Fabio Brau

Scuola Superiore Sant'Anna, Pisa

fabio.brau@santannapisa.it

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

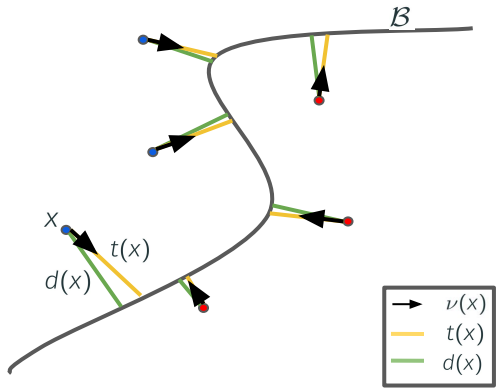
http://retis.santannapisa.it/~f.brau

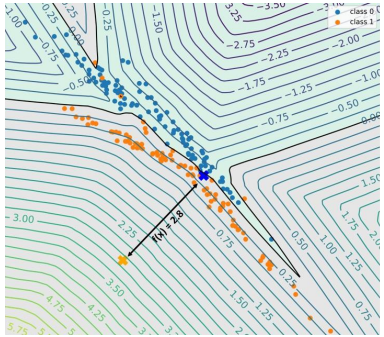
Retis
Real-Time Systems Laboratory

ISTITUTO
DI TELECOMUNICAZIONI,
INFORMATICA
E FOTONICA

Sant'Anna
Scuola Universitaria Superiore Pisa

Conclusions





Online Estimation of MAP via Minimal Root Problem[a]

Signed Distance Classifier[b]

Online Certifiable Robustness

[a] **Brau F.**, Rossolini G., Biondi A., Buttazzo G., *"**On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error**"*. TPAMI, 2022.
[b] **Brau F.**, Rossolini G., Biondi A., Buttazzo G., *"**Robust-by-Design Classification with Unitary-Gradient Neural Networks**"*. AAAI, 2023.

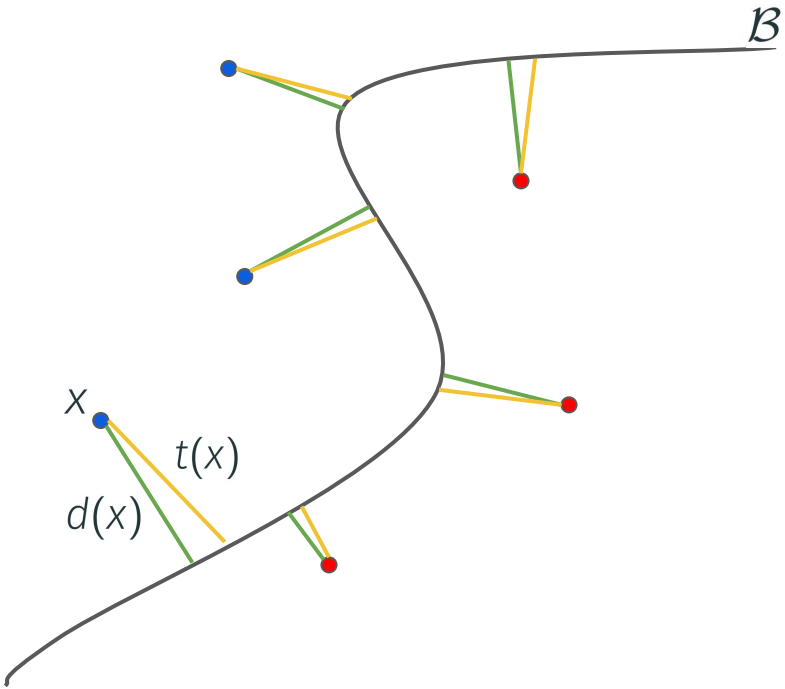
Diffusion applied to Randomized Smoothing?

Estimation of the Minimal Adversarial Perturbation

Estimation of the
Minimal Adversarial Perturbation

$$\frac{1}{\rho} t(x) < d(x) \leq t(x)$$

$\mathcal{K}(x)$ is a $\frac{1}{\rho} t(x)$-robust classification



In our work, we prove that the estimation above is valid in a neighborhood of the classification boundary

Brau F., Rossolini G., Biondi A., Buttazzo G., "***On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error***". TPAMI, 2022.

Estimation through Gradient Direction

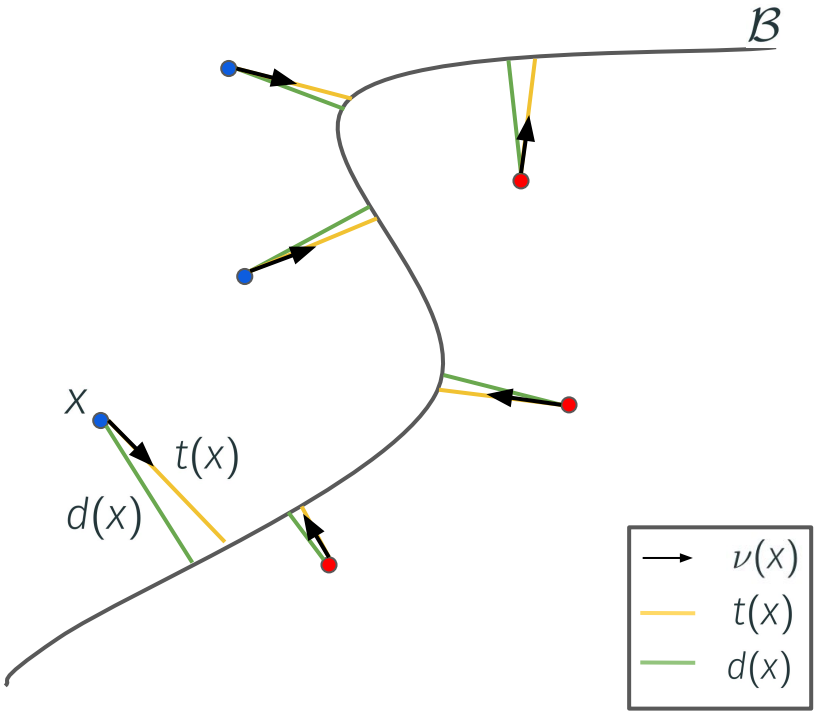
Minimal Root Problem

$$t(x) = \min_{t \in \mathbb{R}_+} \quad |t|$$
$$\text{s.t.} \quad f(x + t\nu(x)) = 0$$
$$\nu(x) = -\text{sgn}(f)(x)\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

Theorem. There exists a *certification radius* σ

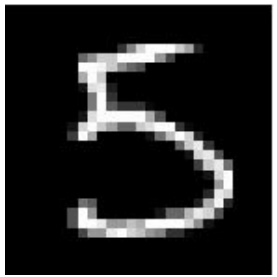
$$\frac{1}{\rho} t(x) < d(x) \leq t(x), \quad \forall x \in \Omega_\sigma$$

where $\mathcal{B} \subseteq \Omega_\sigma$

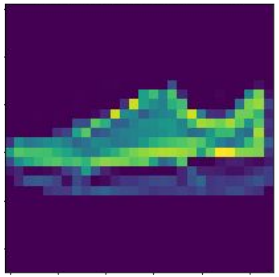


Brau F., Rossolini G., Biondi A., Buttazzo G., "***On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error***". TPAMI, 2022.

Experimental Results



MNIST



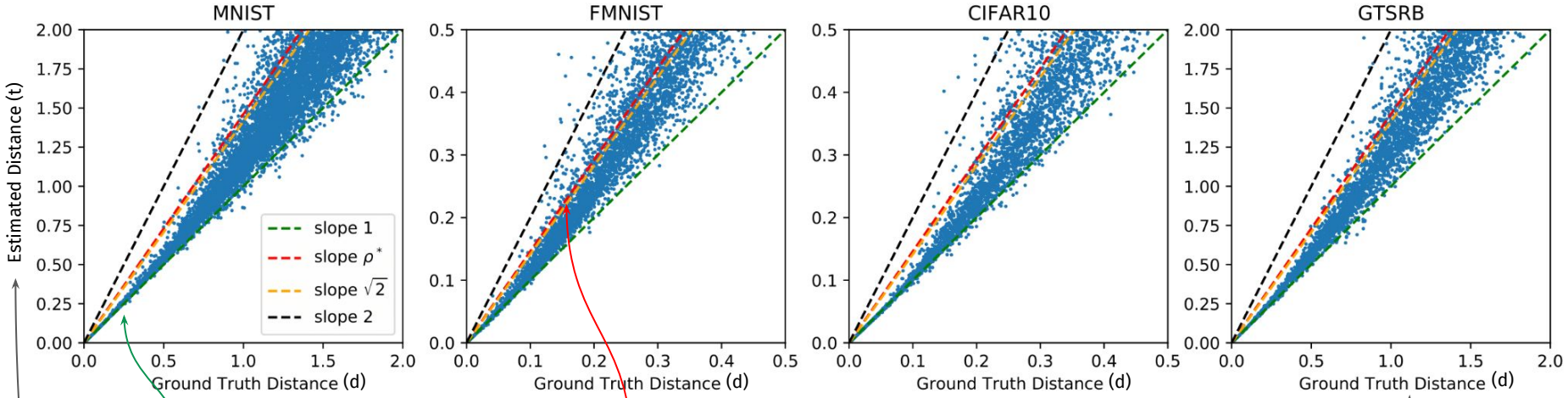
Fashion MNIST

CIFAR10

GTSRB

Brau F., Rossolini G., Biondi A., Buttazzo G., "***On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error***". TPAMI, 2022.

Distance Estimation



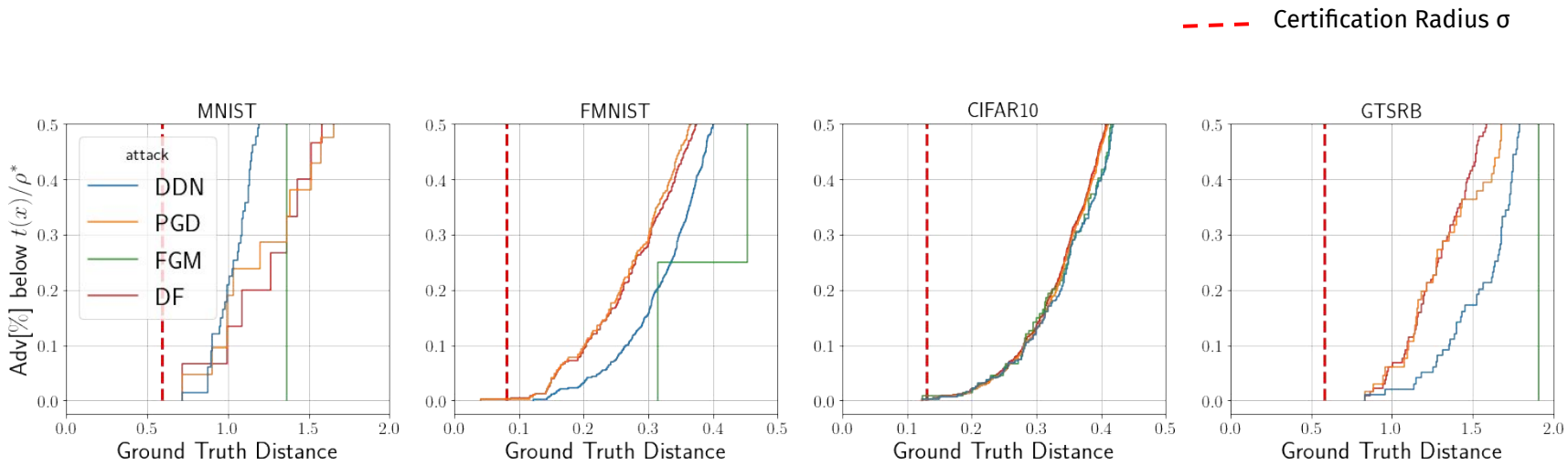
$$d(x) < t(x)$$

$$\frac{1}{\rho} t(x) < d(x)$$

Bisection and Newton method

Ground Truth Distance = min(IP,DF,DDN)

A Bound of the Minimal Adversarial Perturbation



On the left side of the certification line, the samples which satisfy the estimation.

On the y-axis the amount of successful attacks of magnitude $\frac{1}{\rho}t(x)$

Estimation of the Minimal Adversarial Perturbation

$$\frac{1}{\rho}t(x) < d(x) < t(x)$$

Brau F., Rossolini G., Biondi A., Buttazzo G., "***On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error***". TPAMI, 2022.

Summary and Motivations

Achievement

Aim	Solution	Guarantees	# Inferences
Computation of MAP	Approximated	✓	≈ 20 (fast)

Applications

1. Online Robustness Certification
2. Online Verification (below σ)
3. Fast Adv.Ex Generation

Brau F., Rossolini G., Biondi A., Buttazzo G., "***On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error***". TPAMI, 2022.

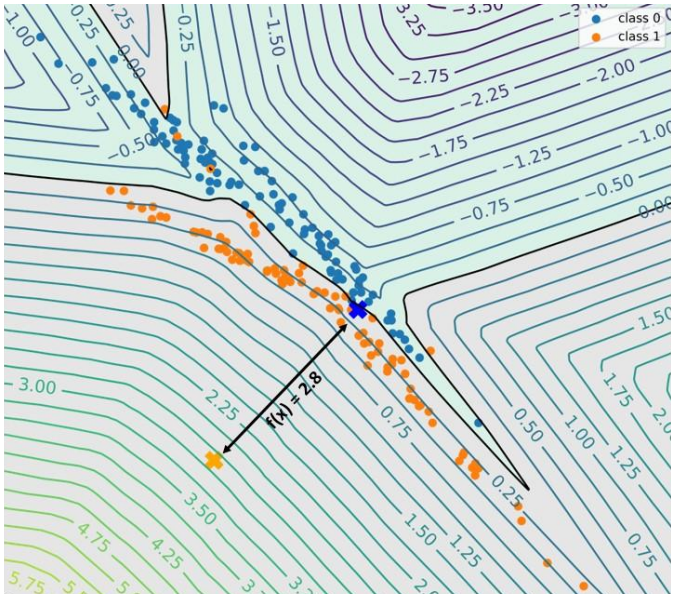
Signed Distance Classifier

Motivations

Aim	Estimation of MAP	Guarantees	# Inferences
Search for f	Exact	✓	1

?

The output of f must provide the MAP.



Models for Certifiable Robust Classification

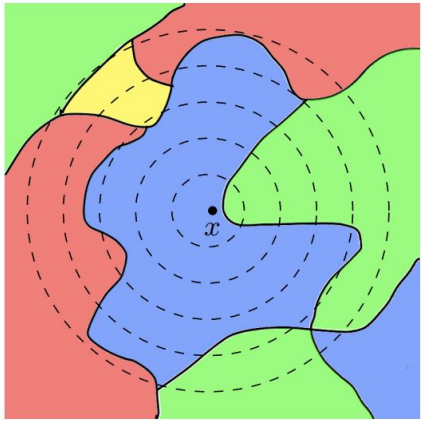
L-Lipschitz Neural Networks

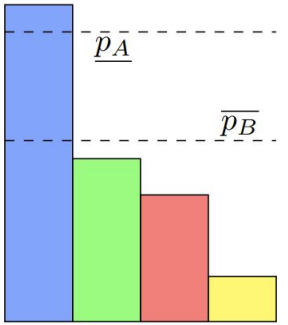
$$\|f(x) - f(y)\| \leq L\|x - y\|$$

$$R := \frac{\max(0, f_l(x) - \max_{j \neq l} f_j(x))}{L\sqrt{2}}$$

$\mathcal{K}(x)$ is a R -robust classification

Classification via Randomized Smoothing





$$R := \frac{\sigma}{2}\left(\Phi(p_A)^{-1} - \Phi(p_B)^{-1}\right)$$ [b]

[a] Li et al, "***Preventing Gradient Attenuation in Lipschitz Constrained Convolutional Networks***". NIPS, 2019.
[b] Cohen et al, **"Certified Adversarial Robustness via Randomized Smoothing".** PMLR, 2021.

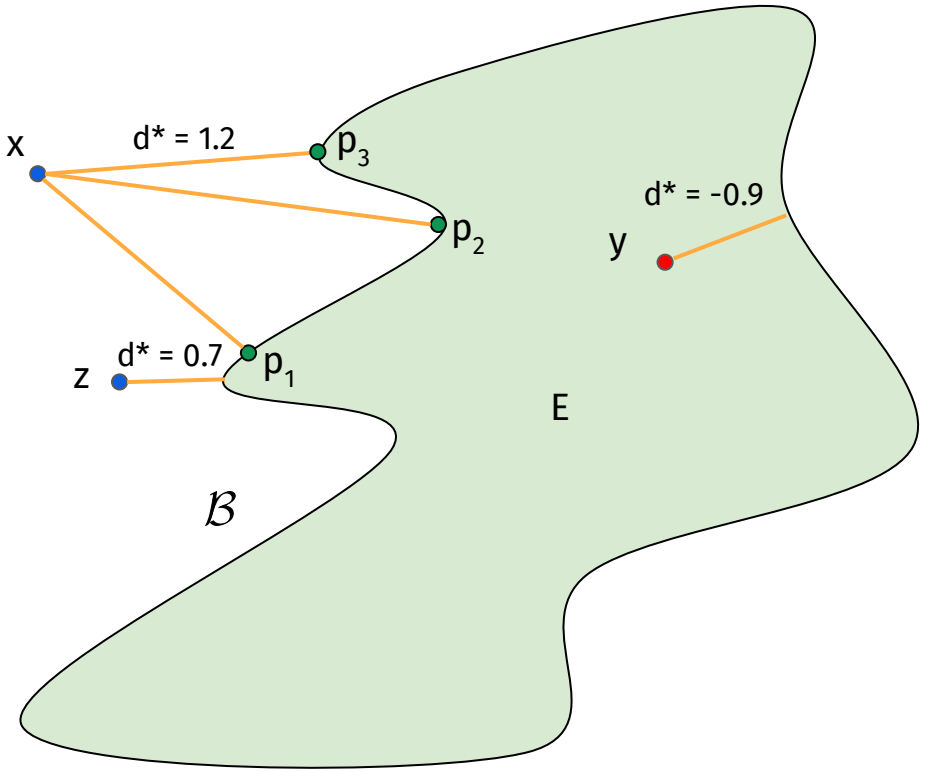
Signed Distance Function

Distance Function

$$d(x, \mathcal{B}) = \min_{p \in \mathcal{B}} d(x, p)$$

Signed Distance Function

$$d^*(x) = \begin{cases} -d(x, \mathcal{B}) & x \in E \\ d(x, \mathcal{B}) & x \notin E \end{cases}$$



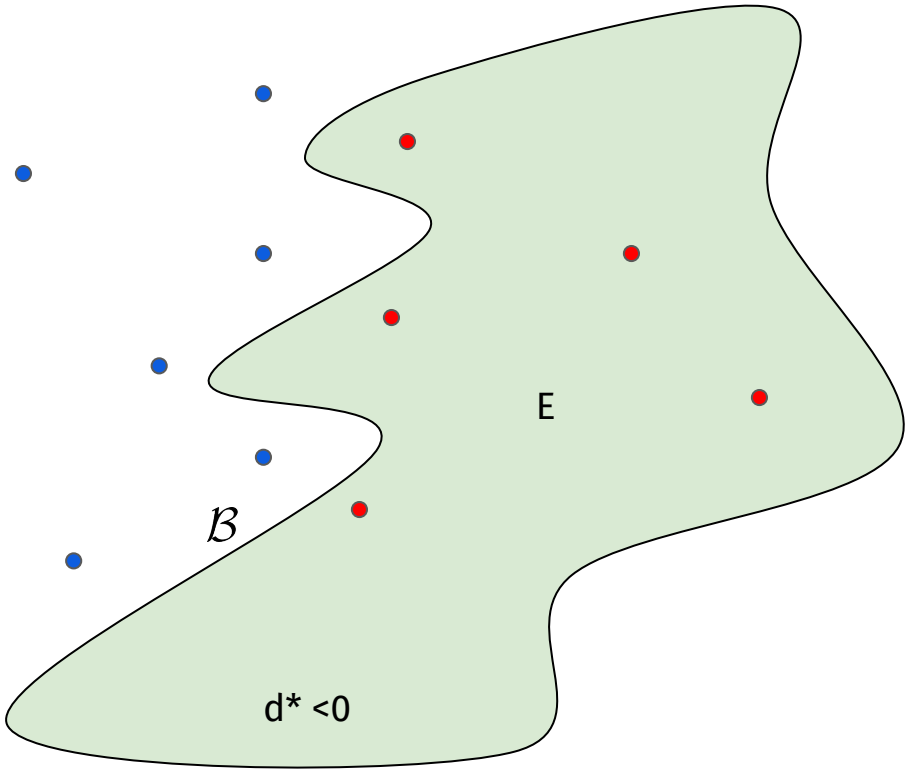
SDF are Binary Classifiers

Binary classification from f

$$\forall x, \quad \mathcal{K}_f(x) = \begin{cases} -1, & \text{if } f(x) < 0 \\ 1, & \text{if } f(x) > 0 \end{cases}$$

d* provides the same classification

$$E = \{x : f(x) \leq 0\} \quad \Rightarrow \quad \mathcal{K}_f = \mathcal{K}_{d*}$$



SDF are robust by Design

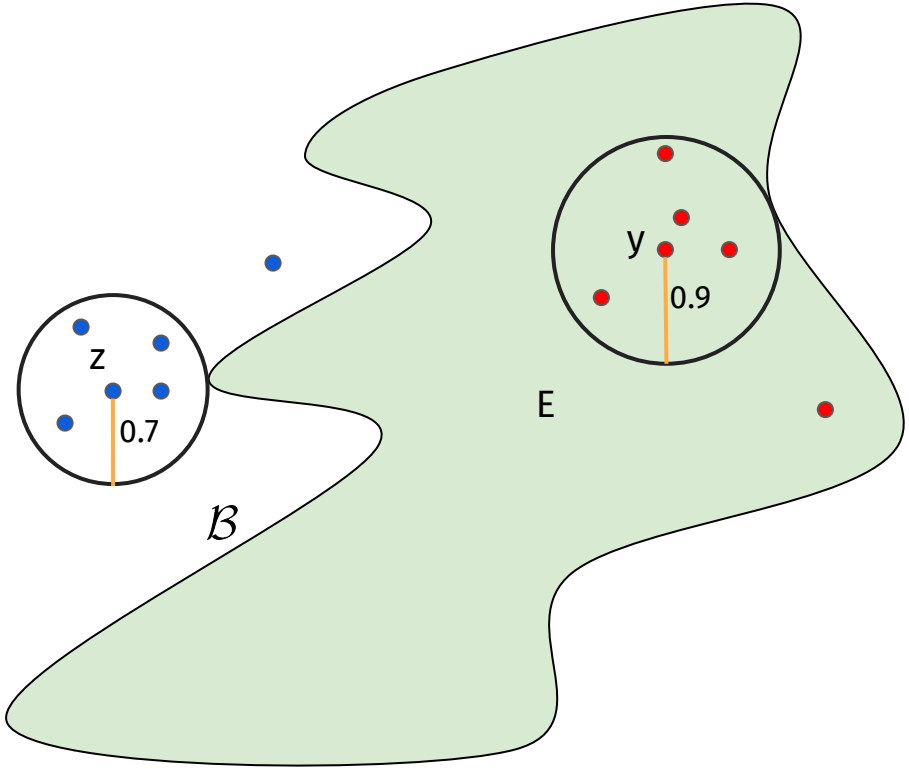
Robustness Property for $R=|d^*(x)|$

$$d(x, x+\delta) < R$$

$$\Downarrow$$

$$\mathcal{K}_{d^*}(x) = \mathcal{K}_{d^*}(x+\delta)$$

Remark. The robustness radius R is obtained at the cost of a single inference of the model



Unitary Gradient Property

Signed Distance Function d*

$$\nabla d^*(x) = \frac{x - x^*}{d^*(x)}$$

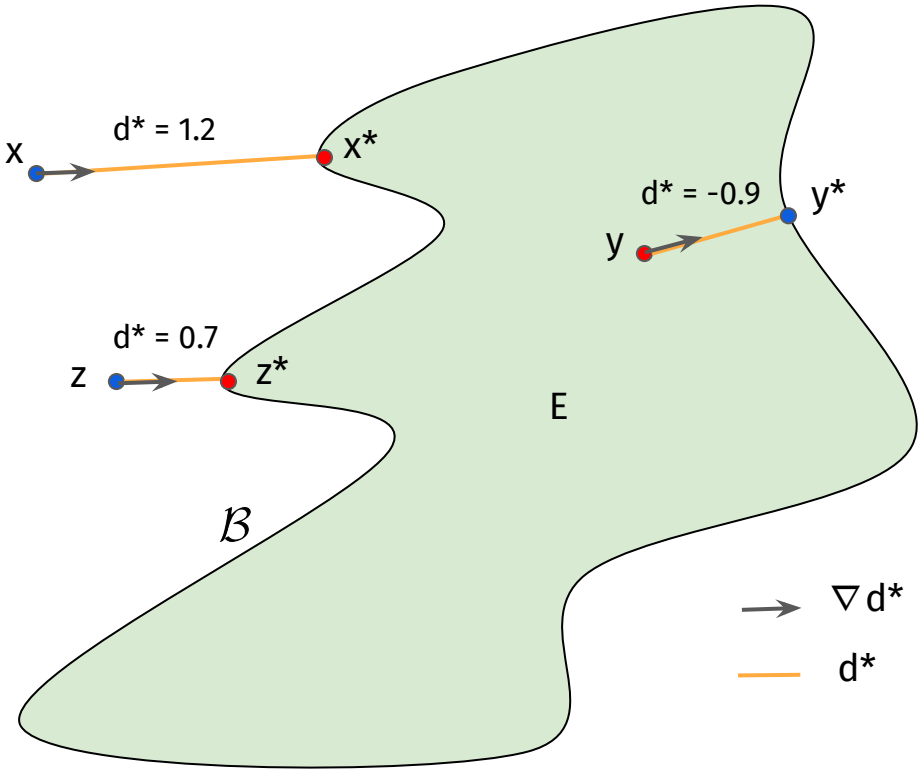
Necessary Conditions

Unitary Gradient

$$\|\nabla d^*(x)\| = 1$$

Closest Adversarial Example

$$x^* = x - d^*(x)\nabla d^*(x)$$

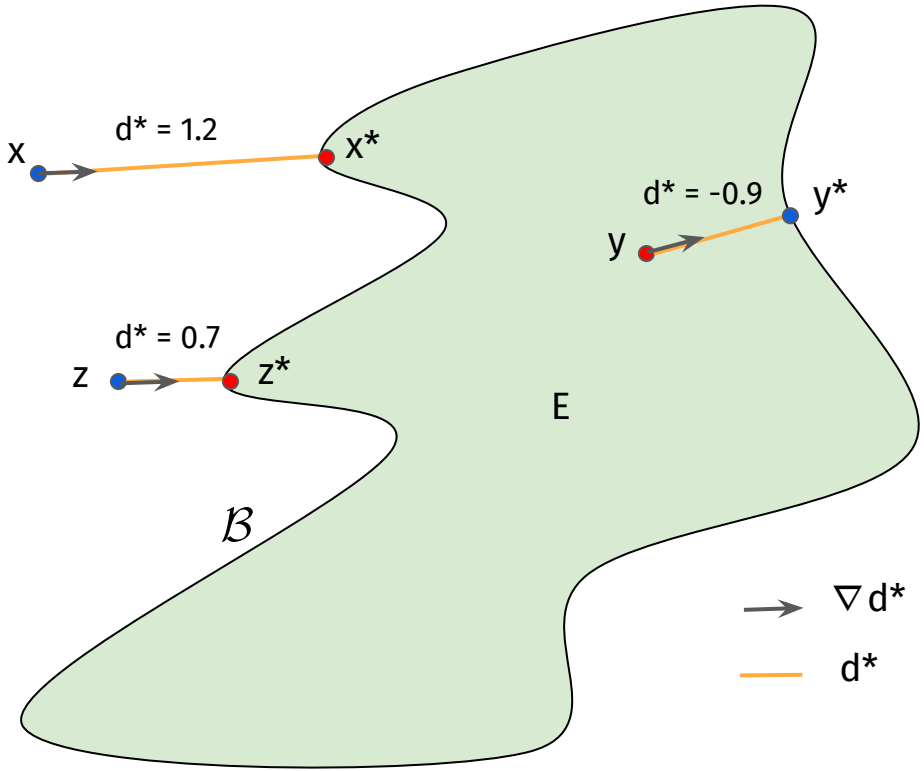


Unitary Gradient Property

Theorem. (Characterization)
Let $f : \mathcal{U} \to \mathbb{R}$, *where* $\mathcal{U}$ *open set. Then*

$$\|\nabla f\| = 1 \Rightarrow \exists \Omega \subseteq \mathcal{U}, \quad f(x) = d^*(x)$$

where $E = \{x : f(x) \leq 0\}$.



Unitary Gradient Neural Network

Feed Forward Neural Network

$$f = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}$$

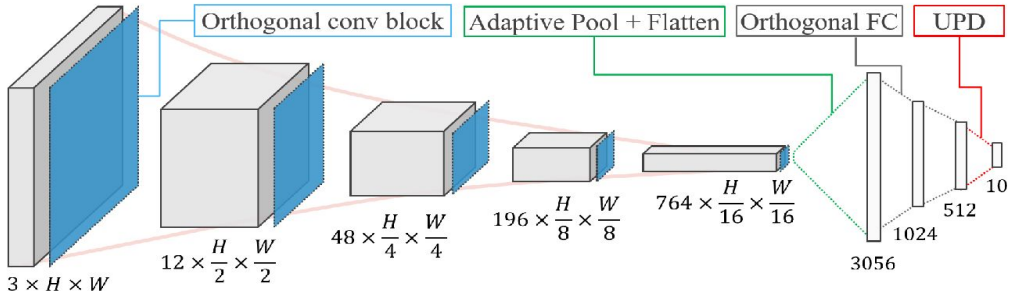
Sufficient Conditions

Gradient Norm Preserving

$$\mathrm{Jac} f^{(i)} \, \mathrm{Jac} f^{(i)T}(\cdot) = I$$

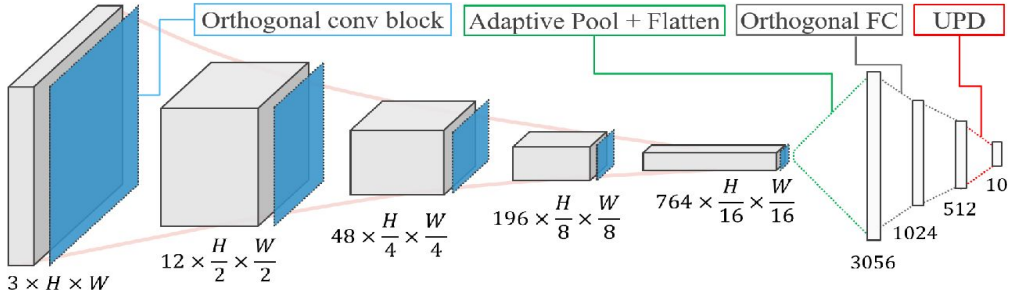
Unitary Pair Difference

$$\|(\nabla f_i^{(L)} - \nabla f_j^{(L)})(\cdot)\| = 1$$

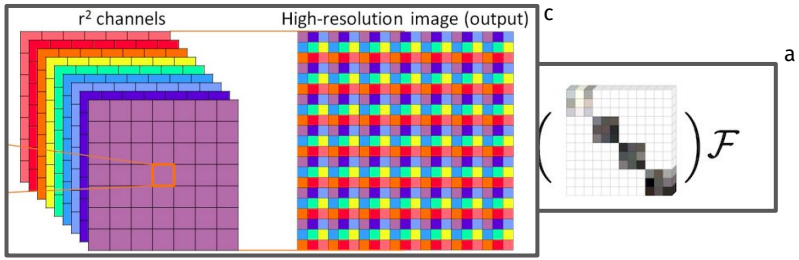


Brau F., Rossolini G., Biondi A., Buttazzo G., "***Robust-by-Design Classification with Unitary-Gradient Neural Networks***". AAAI, 2023.

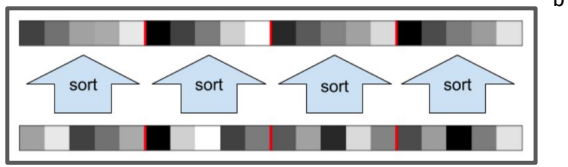
Unitary Gradient Neural Network



- [a] Orthogonal Conv/Linear Layers
- [c] Pixel Unshuffle Layer



- [b] GNP Activation Function (MaxMin, Abs)
- Unitary Pair Difference



[a] Asher Trockman and Zico Kolter. **Orthogonalizing Convolutional Layers with the Cayley Transform.** ICLR 2021.
[b] Cem Anil et al. **Sorting Out Lipschitz Function Approximation.** ICML 2019.
[c] Wenzhe Shi et al. **Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network.** CVPR 2017.

Unitary Pair Difference Layer

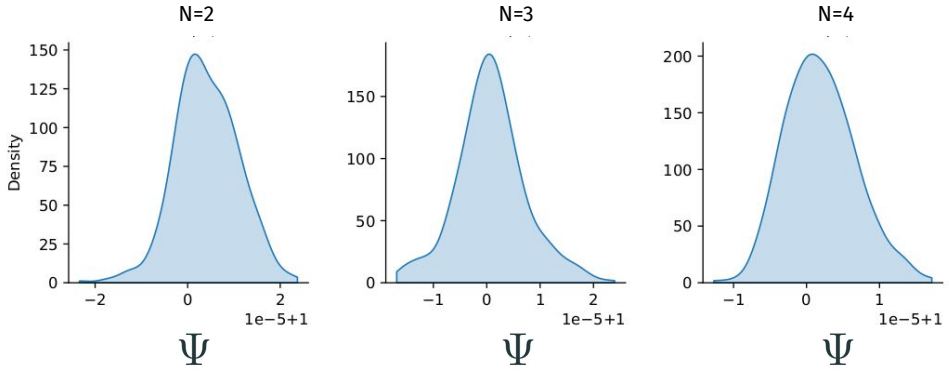
Definition

$$f(x) = U(W)x + b$$

$$\|U_{i:} - U_{j:}\| = 1$$

Parameterization via Minimization

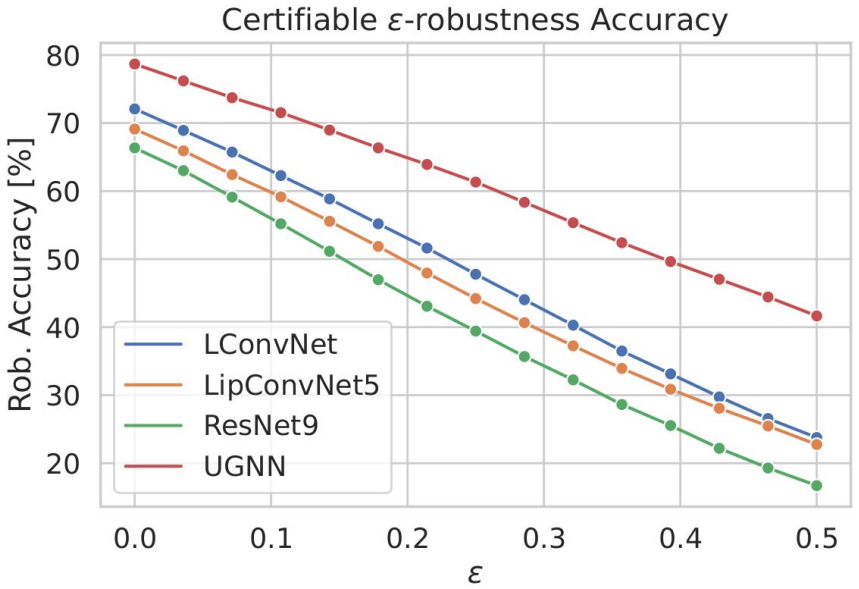
$$\Psi(A) = \sum_{h<k} (\|A_{h:} - A_{k:}\|^2 - 1)^2$$




```
def U(W: Tensor, N: int):
  U = W
  for in range(N):
    U = L-BFGS(psi(U), U)
  return U
```

Certified Robust Accuracy

Input Size	Accuracy [%] Std.Norm	Raw
32	72.1±0.54	68.9±0.81
64	72.6±0.69	72.8±0.61
128	74.9±0.45	76.2±0.30
256	**76.8**±0.29	**78.5**±0.22



Certified accuracy on $\mathbb{X} = \{(x_i, y_i)\}_{i \leq N}$

$$\mathcal{A}_\varepsilon(\mathbb{X}) = \frac{1}{N} \# \{x_i \,:\, \mathcal{K}_{d^*}(x_i) = y_i,\ |d^*(x)| \geq \varepsilon\}$$

Brau F., Rossolini G., Biondi A., Buttazzo G., "***Robust-by-Design Classification with Unitary-Gradient Neural Networks***". AAAI, 2023.

Estimation of the Minimal Adversarial Perturbation

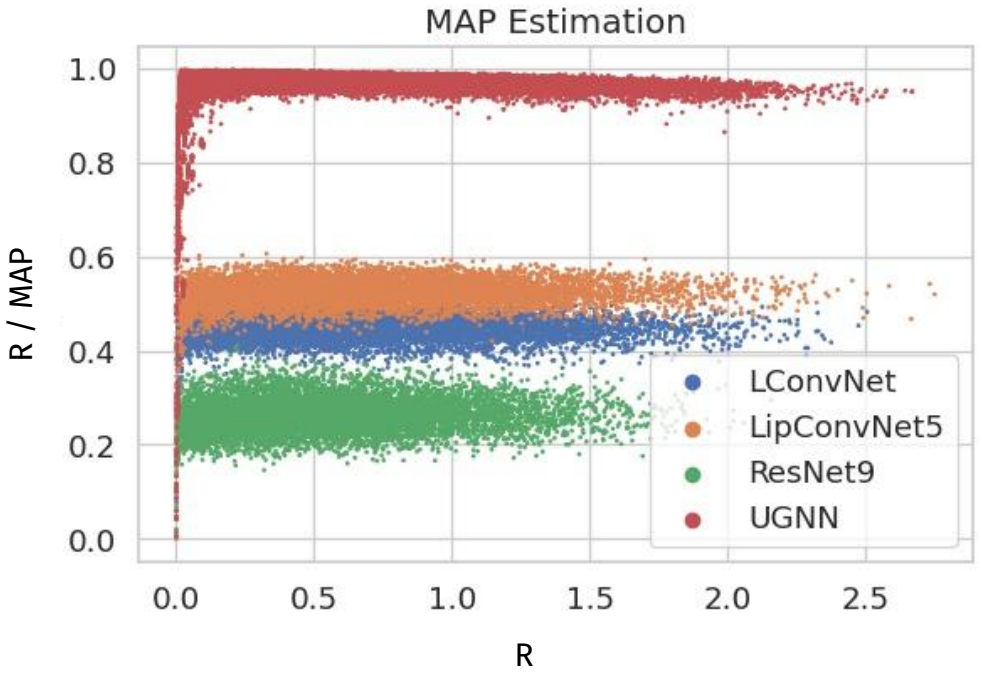
Lower Bound of MAP

L-Lipschitz Model *f*

$$R = \frac{f_l(x) - \max_{j \neq l} f_j(x)}{L\sqrt{2}}$$

UGNN Model *f*

$$R = f_l(x) - \max_{j \neq l} f_j(x)$$



Brau F., Rossolini G., Biondi A., Buttazzo G., "***Robust-by-Design Classification with Unitary-Gradient Neural Networks***". AAAI, 2023.

Backup: Theoretical Certification Radius

Theoretical Bound of σ

$$\sigma \propto \frac{\inf_x \|\nabla_x f(x)\|}{\sup_x \|\nabla^2 f(x)\|}$$

Gradient of f

Hessian of f

where $\rho \approx 1.461$

Theorem. There exists a *certification radius* σ

$$\frac{1}{\rho} t(x) < d(x) \leq t(x), \quad \forall x \in \Omega_\sigma$$

where $\mathcal{B} \subseteq \Omega_\sigma$

Backup: Empirical Certification Radius

Empirical Bound of σ

$$\mathbb{X} = \{(x_i, y_i)\}_{i \leq N}$$

Validation dataset

$$\hat{\sigma}(\rho) = \min \left\{ d(x) \; : \; \frac{t(x)}{d(x)} > \rho, \, (x, l) \in \mathbb{X} \right\}$$

	MNIST	FMNIST	CIFAR10	GTSRB
σ	0.59	0.12	0.13	0.58
#n	722	1198	849	576

Theorem. There exists a *certification radius* σ

$$\frac{1}{\rho} t(x) < d(x) \leq t(x), \quad \forall x \in \Omega_\sigma$$

where $\mathcal{B} \subseteq \Omega_\sigma$

Brau F., Rossolini G., Biondi A., Buttazzo G., "***On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error***". TPAMI, 2022.

Backup: False Positive

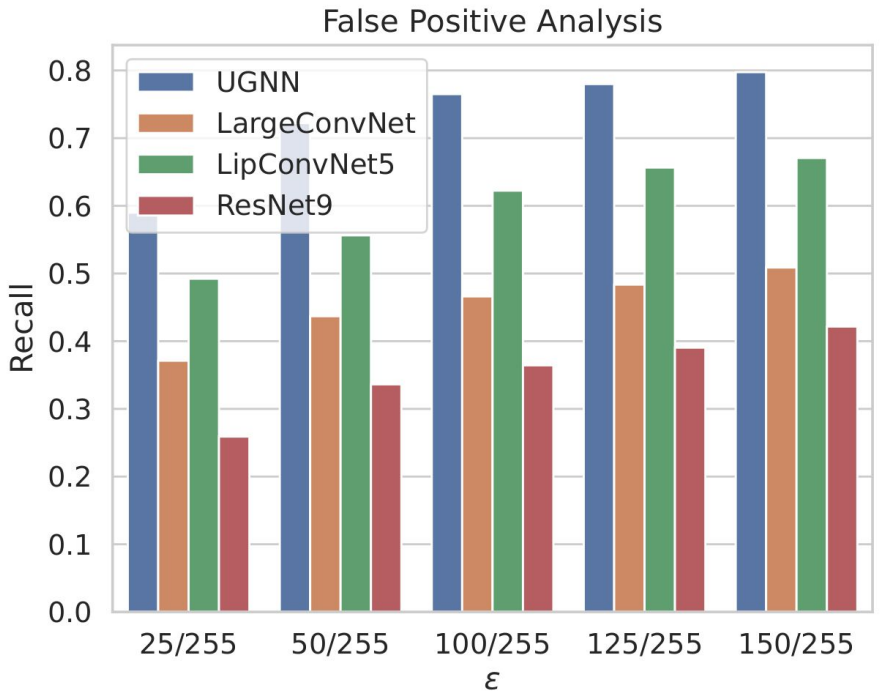
Recall

$$r = \frac{\#TP_\varepsilon}{\#(TP_\varepsilon \cup FP_\varepsilon)}$$

$$TP_\varepsilon := \{x : MAP(x) \leq \varepsilon \wedge LB(x) \leq \varepsilon\}$$

$$FP_\varepsilon := \{x : MAP(x) > \varepsilon \wedge LB(x) \leq \varepsilon\}$$

Ratio of correctly rejected samples over the whole rejections

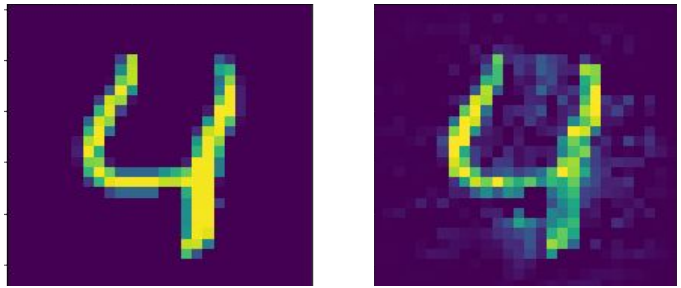


Ideal Signed Distance Classifier, Recall = 1.0

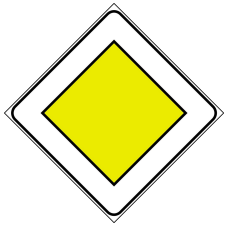
Brau F., Rossolini G., Biondi A., Buttazzo G., "***Robust-by-Design Classification with Unitary-Gradient Neural Networks***". AAAI, 2023.

Thought Experiment

Mand.Left: 97.0 % , d : 1.2



Four: 99.0 % , d : 2.8



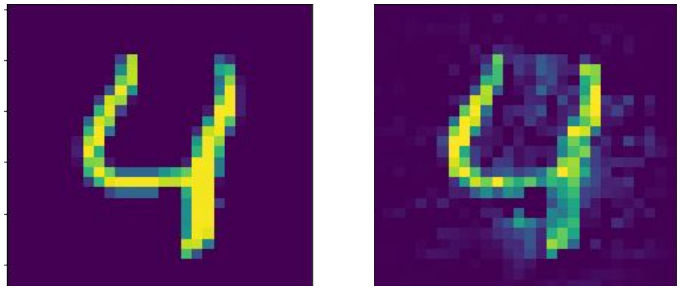
Stop 98.9 % ✗

d = 1.5

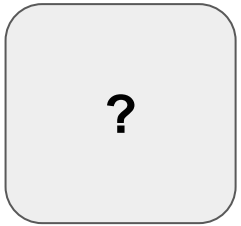
Security Threshold : ε = 2.5

Thought Experiment

Mand.Left: 97.0 % , d : 1.2



Four: 99.0 % , d : 2.8



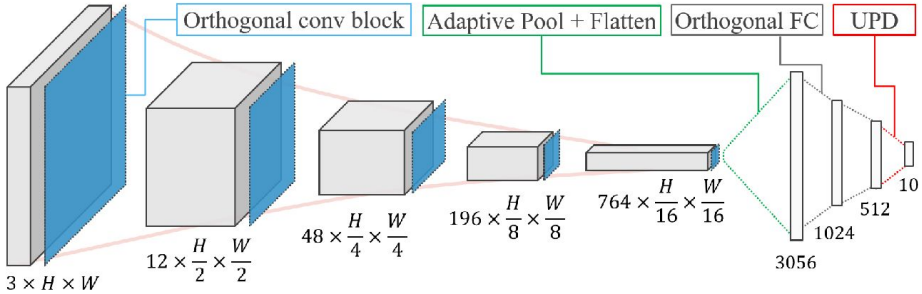
Stop 98.9 %

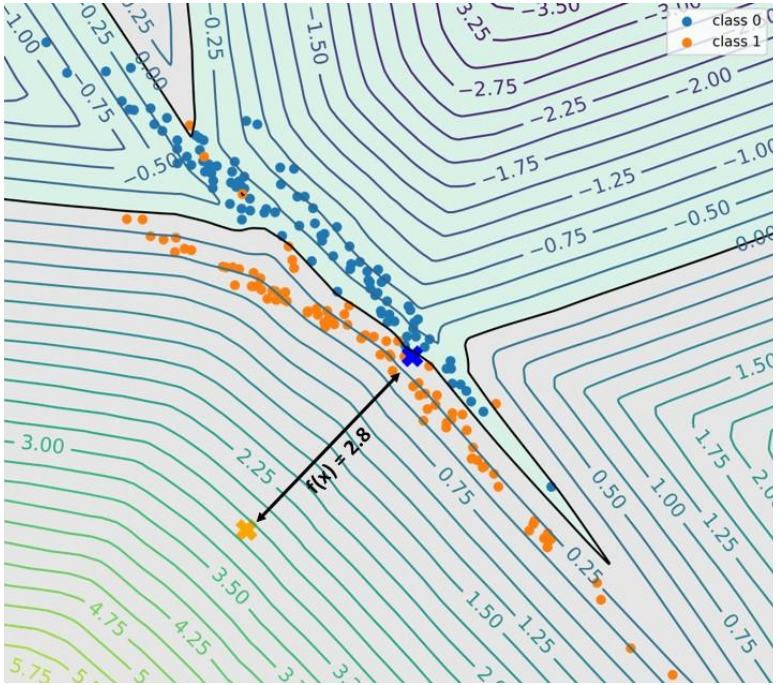
Security Threshold : ε = 2.5

Signed Distance Classifier

Method	Solution	Guarantees	# Inferences
New	Accurate	✓	= 1

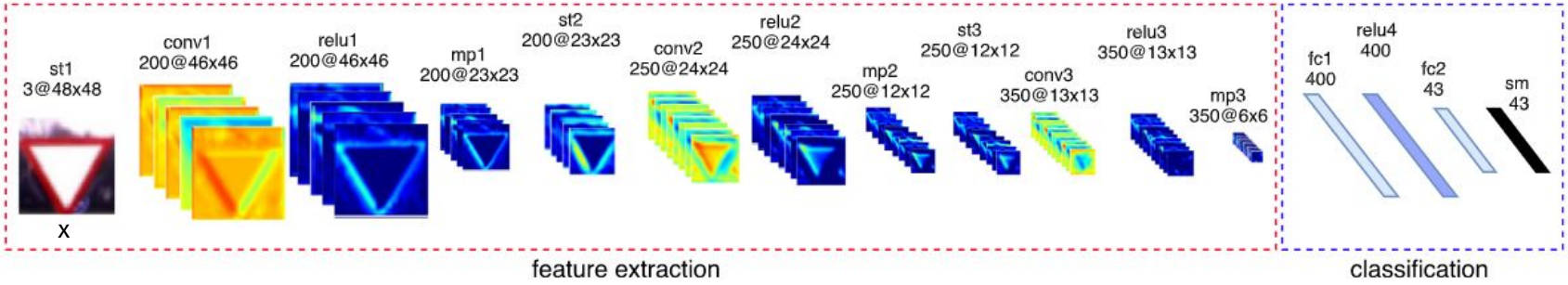
?

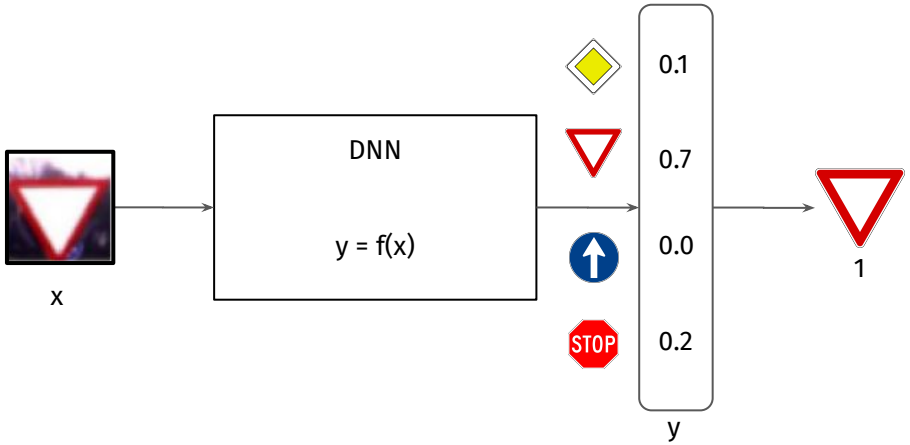




Brau Fabio, Rossolini Giulio, Biondi Alessandro, Buttazzo Giorgio. ***Robust-by-Design Classification with Unitary-Gradient Neural Networks***. Arxiv (preprint) arxiv.org/abs/2209.04293 . September 09, 2022.

The Classification Problem





Encoding Table

0	
1	
2	
4	

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