Certifiable Robustness and Defenses

Machine Learning Security

Fabio Brau

Dipartimento di Ingegneria Elettrica e Elettronica, Cagliari

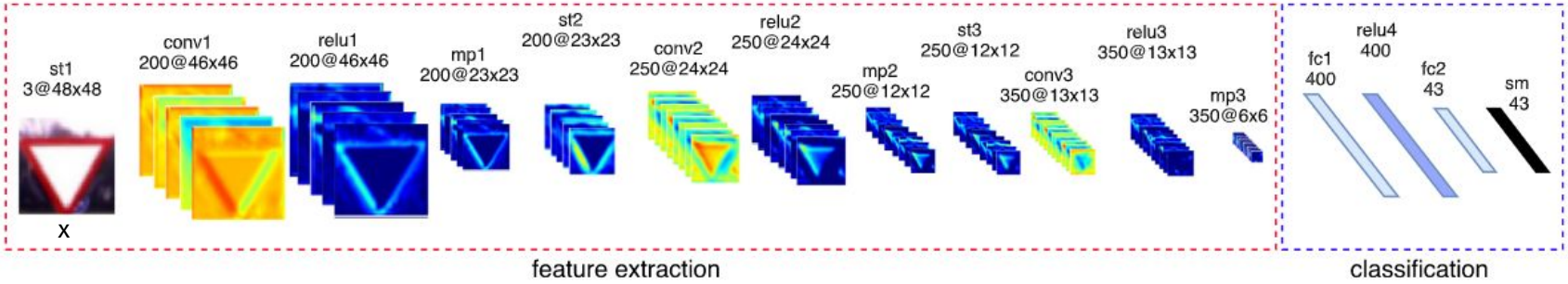
Introduction and Definition of Certifiable Robustness

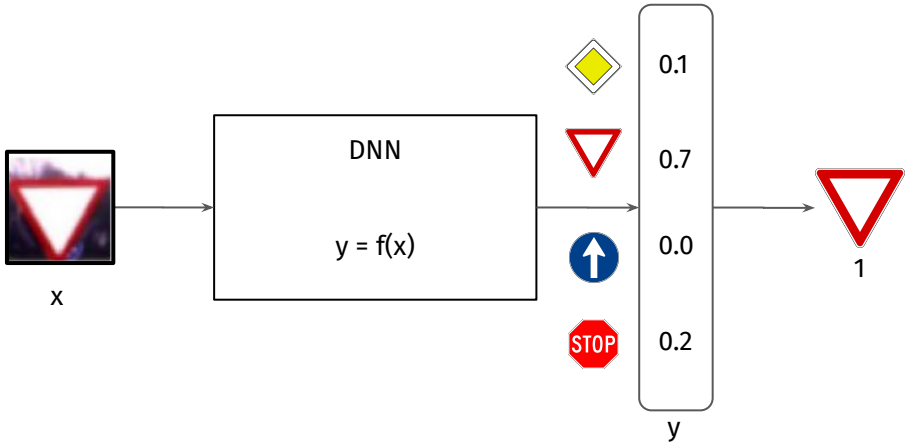
Verification Methods

Lipschitz Bounded Neural Networks

Randomized Smoothing

Neural Networks for Image Classification

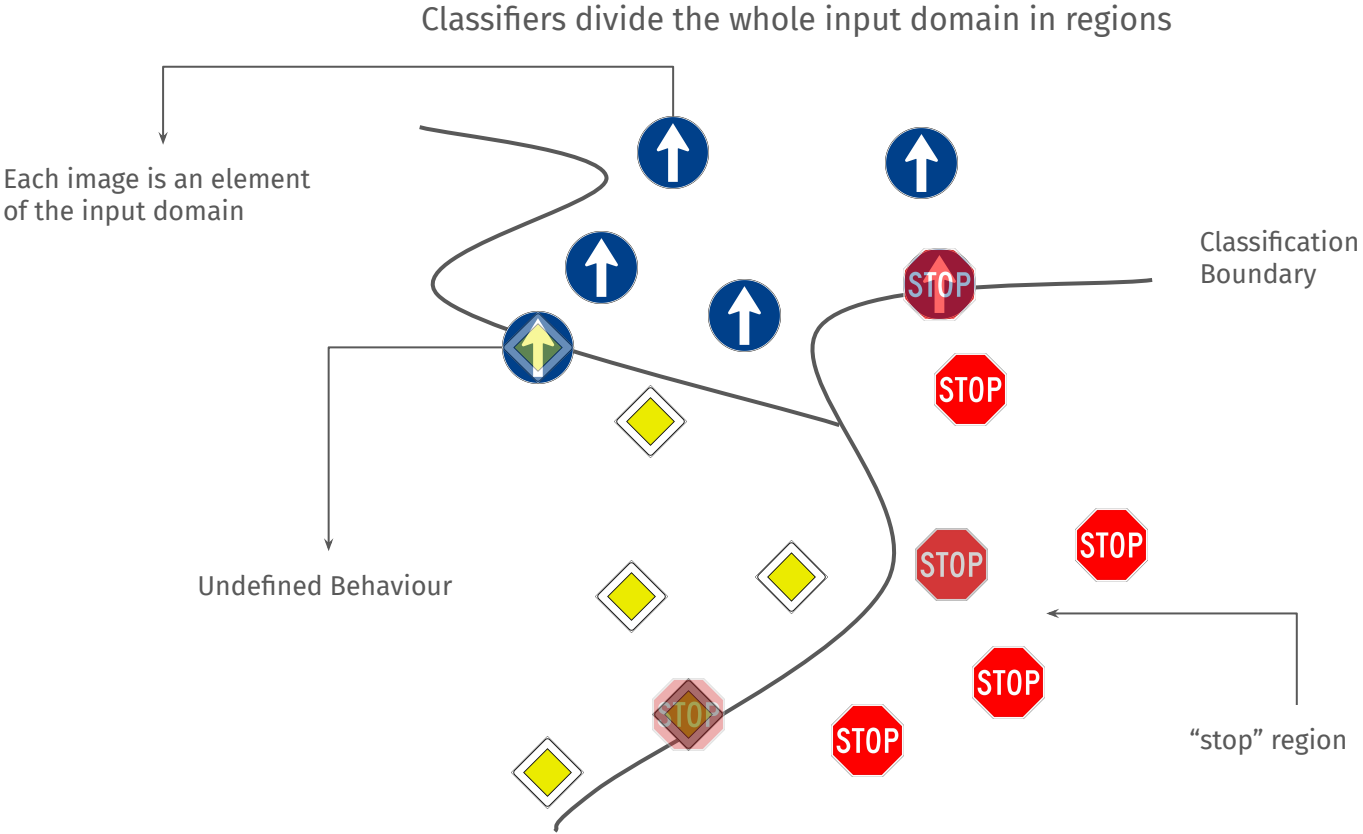




Encoding Table

0	
1	
2	
4	STOP

Geometrical Intuition



How to evaluate a classifier?

Definition (classification)
Given a Neural Network $f : \mathbb{R}^n \to \mathbb{R}^C$
the predicted label is the largest component

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

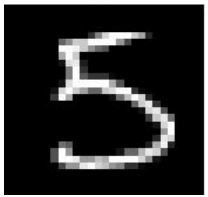
Definition (Accuracy)
Given a distribution of images $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$

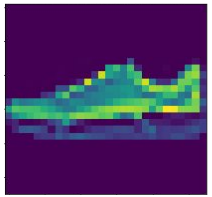
and an *oracle* $\mathcal{O}(\mathbf{x}) \in \{1, \ldots, C\}$

the accuracy is defined as the ratio of good predictions

$$\mathcal{A}(f) = \mathbb{P}\left(\mathcal{K}_f(\mathbf{x}) = \mathcal{O}(\mathbf{x})\right)$$

Accuracy on a few known datasets


MNIST Acc >99%


FMNIST Acc >99%


CIFAR10 Acc >94%


GTSRB Acc >97%


ImgNet Acc >94%

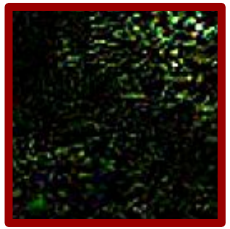
Accuracy and trustworthiness are not interchangeable concepts!

Sensitivity to Input Perturbation

Average Minimal Adversarial Perturbation*
Average noise (dB) sufficient to fool the model for different dataset

MNIST -27 dB , **FMNIST** -40 dB

CIFAR-10 -41 dB, **GTSRB** -32 dB

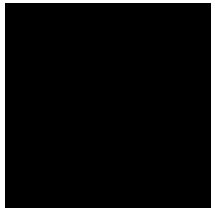


Luminosity improved by 50 %

Clean Input

SHARK

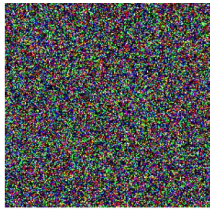
No Perturbation



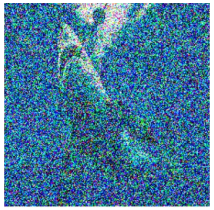
-∞ dB

SHARK

Gaussian Noise (σ = .5)

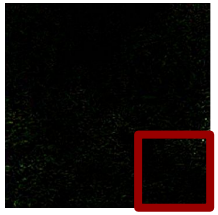


-6 dB



PLANE

Adversarial Perturbation



-30 dB

SHIP

Certifiable Robustness

Minimal Adversarial Perturbation (Binary Case)

Definition (Binary Classification)
Given a scalar continuous function $f : \mathbb{R}^n \to \mathbb{R}$
the binary classifier is defined on the sign of f

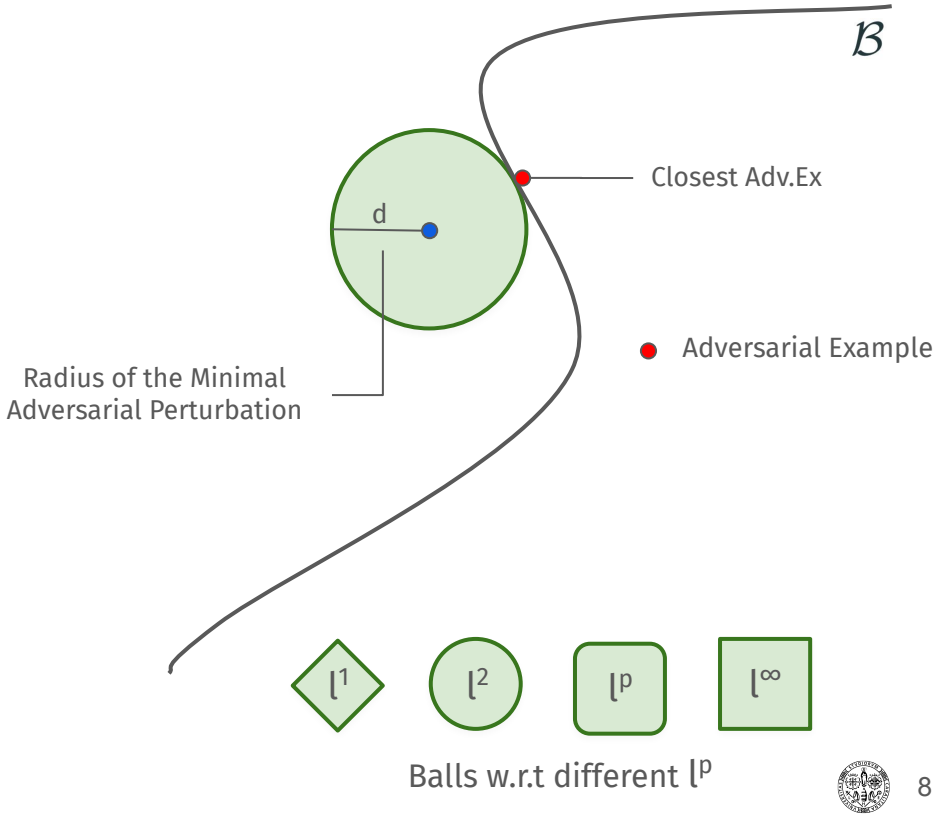
$$\mathcal{K}_f(x) = \begin{cases} 1 & \text{if} \quad f(x) \geq 0 \\ -1 & \text{if} \quad f(x) < 0 \end{cases}$$

Definition (Decision Boundary)
Is the region of zeros of the function

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

Definition (Minimal Adversarial Perturbation)
Is the closest point in the decision boundary

$$d_f(x) = \inf_{\mathcal{B}} \|p - x\|$$



Minimal Adversarial Perturbation (General Case)

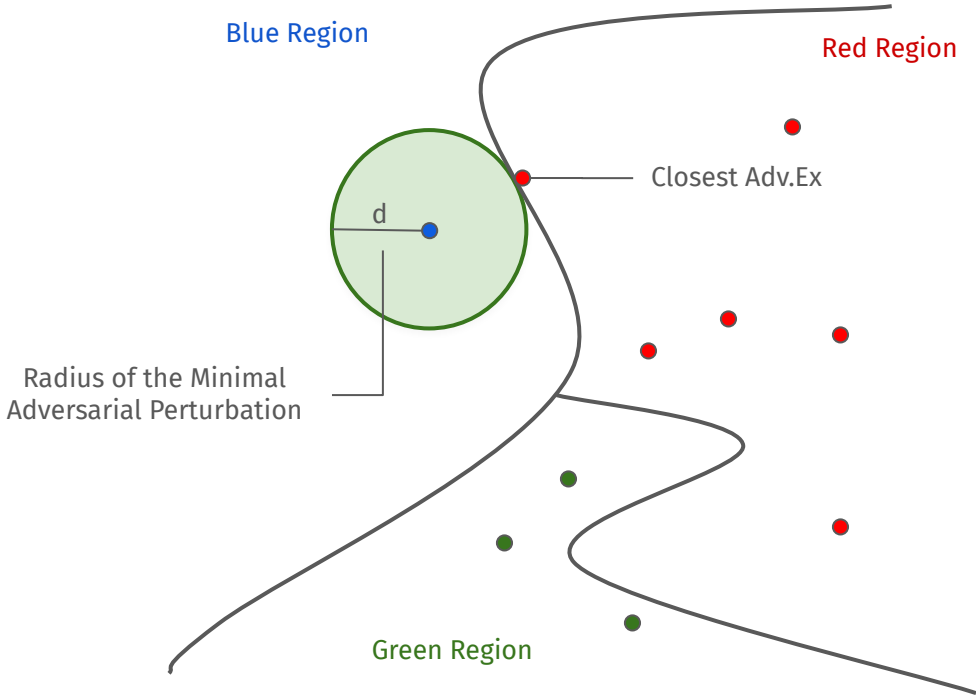
Definition (Multiclass classification)
Given a continuous function $f : \mathbb{R}^n \to \mathbb{R}^C$. A C-classes classifier is given by the index of the largest component

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

Definition (Minimal Adversarial Perturbation)
Is the distance from the closest adversarial examples

$$d_f(x, l) = \inf_{\delta \in \mathbb{R}^n} \|\delta\| \quad \text{s.t.} \quad \mathcal{K}_f(x+\delta) \neq l$$

where l is the correct class of x.



Minimal Adversarial Perturbation

Observation

The MAP of the multi-class classifier can be reduced to the MAP of a binary classifier. Let $F(y) = f_l(y) - \max_{j \neq l} f_j(y)$ and l the label of x, then the following equality holds,

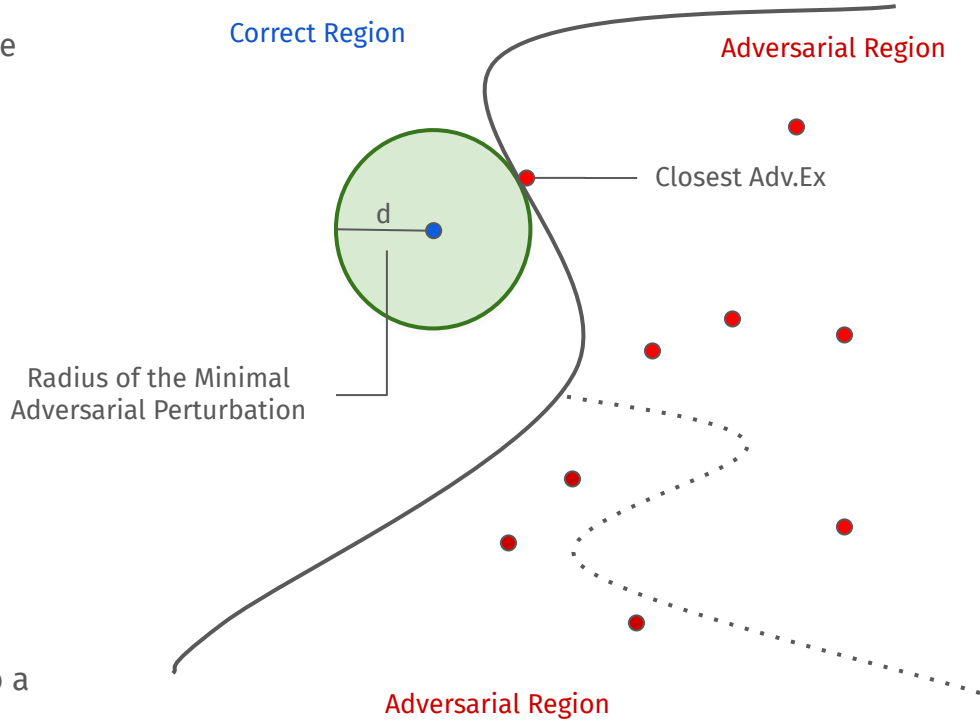
$$d_f(x, l) = d_F(x).$$

Proof

Prove first that $\mathcal{K}_f(y) \neq l \iff F(y) \leq 0$.

Second, prove that $d_f(x, l) \leq d_F(x)$.

Finally, prove that assuming the strict inequality brings to a contradiction.



Certifiable ε-Robust Classification

Definition (Robustness in l^p norm)
Classification does not change under perturbation of bounded magnitude. In formulas, a classification $\mathcal{K}(x)$ is ε-robust if

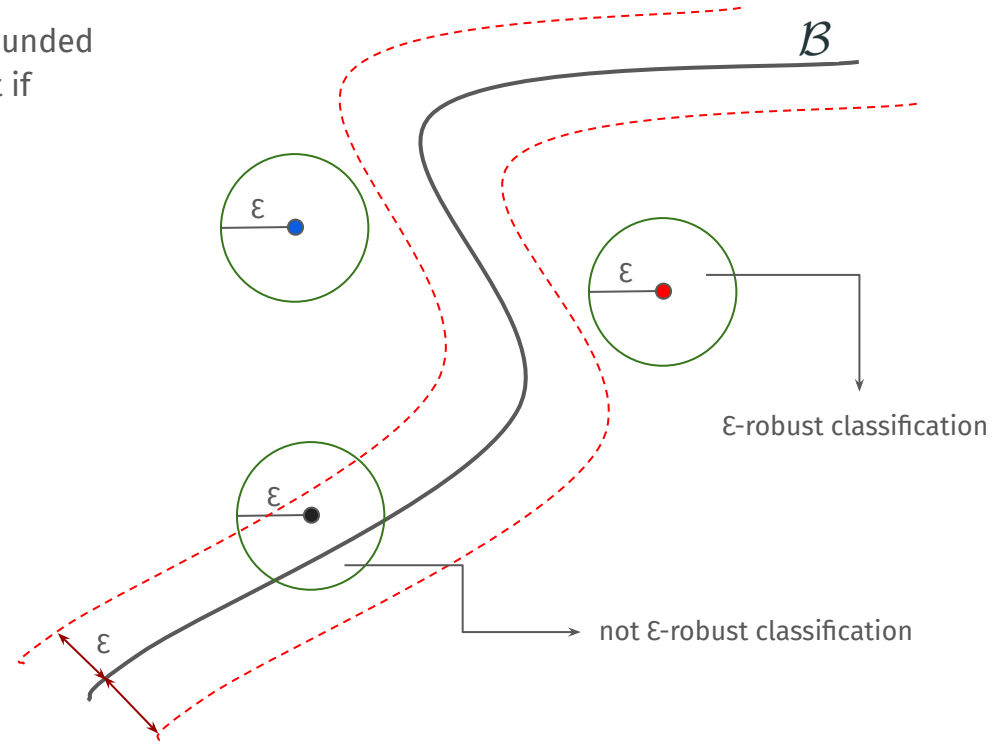
$$\|\delta\| < \varepsilon \;\Rightarrow\; \mathcal{K}(x) = \mathcal{K}(x+\delta)$$

Definition (ε-robust accuracy)
Is the ratio of correct ε-robust classifications

$$\mathcal{A}_R(f,\varepsilon) = \mathbb{P}(\mathcal{K}_f(\mathbf{x}+\delta) = \mathcal{O}(\mathbf{x}),\ \forall\|\delta\| < \varepsilon)$$

Remark

- "The input is robust..." 🤔
- "The classifier is robust..." ❌
- "The classification in x is robust..." ✅



Certifiable ε-Robust Classification by MAP computation

Upper Bound of the MAP

All the practical solution of the map provide an adversarial example, which constitutes by construction an **upper bound** of the MAP

$$d_f(x, l) < \|x_{adv} - x\| = \overline{d_f(x, l)}$$

Observation

The upper-bound of the MAP provides a certification of NOT-robustness.

The classification $\mathcal{K}_f(x)$ is $d_f(x, l)$- robust.

The classification $\mathcal{K}_f(x)$ is **not** $\overline{d_f(x, l)}$- robust.

Algorithms for MAP estimation

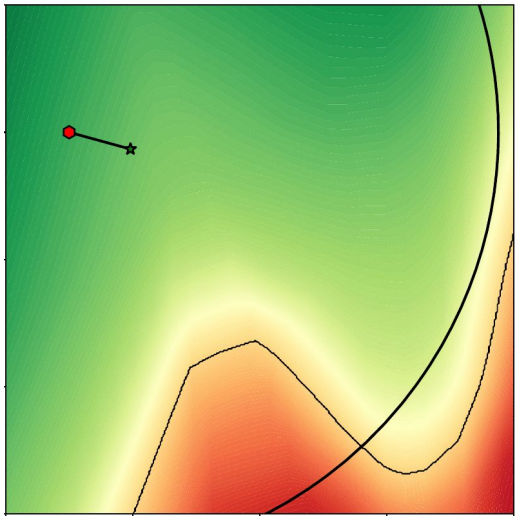
Method	Solution	lp norm	# Inferences
L-BFGS [a]	Accurate	2	> 10k (slow)
CW [b]	Accurate	2, ∞	≈ 10k (slow)
DeepFool [c]	Approximated	2, ∞	≈ 20 (flash)
DDN [d]	Approximated	2	≈ 1k (fast)
FMN [e]	Approximated	0,1,2,∞	≈ 1k (fast)

[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.
[e] Maura Pintor et al. "**Fast minimum-norm adversarial attacks through adaptive norm constraints**". NeurIPS, 2021.

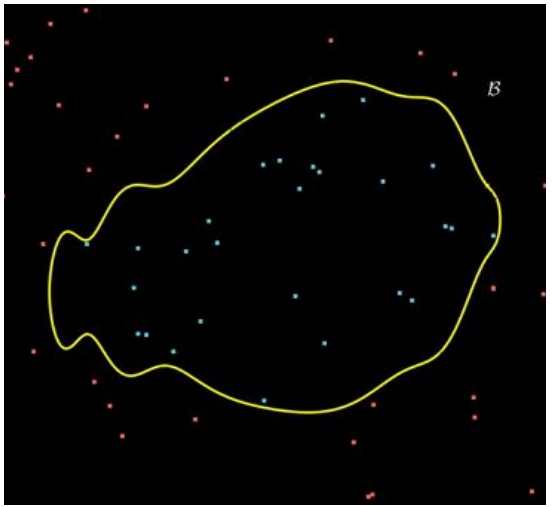
Certifiable ε-Robust Classification by MAP computation

MAP can be estimated (upper bounded) by following the gradient direction.

FMN Strategy [a]



Fast Bisection [b]



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

[b] Fabio Brau et al. *"On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error"*.

Verification Methods

Definitions and Introduction

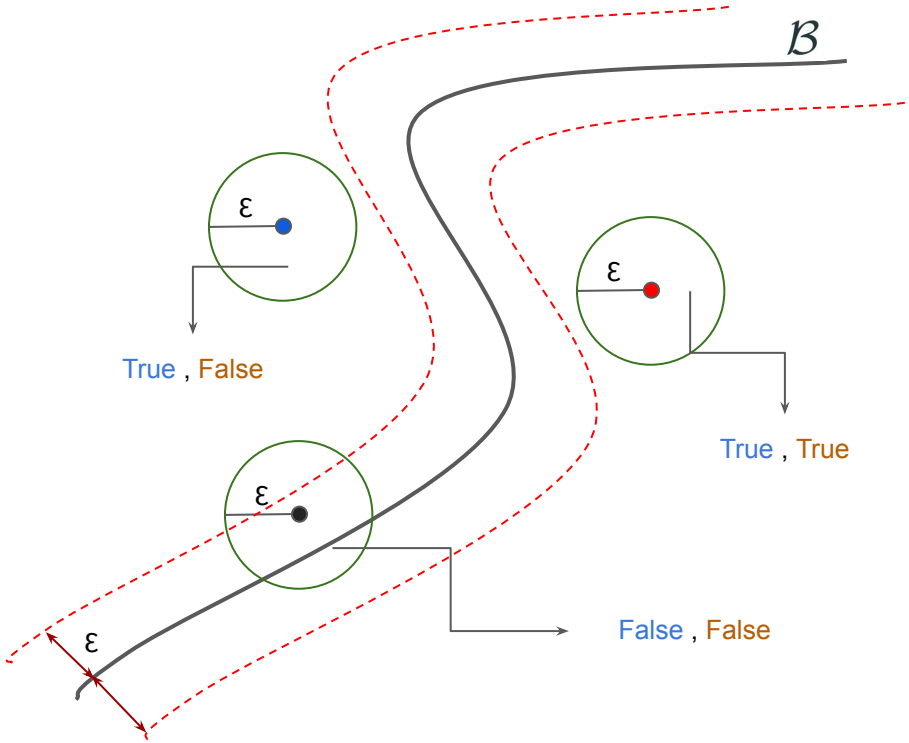
Definition (Verification of the robustness)
Given a classifier $\mathcal{K}$ and a sample x, check whether

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

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

Definition (Complete and Incomplete Verifier)

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



Complete verification is NP-Hard

Theorem (Guy Katz et al.)
Let us assume f a ReLU Deep Neural Network, and

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

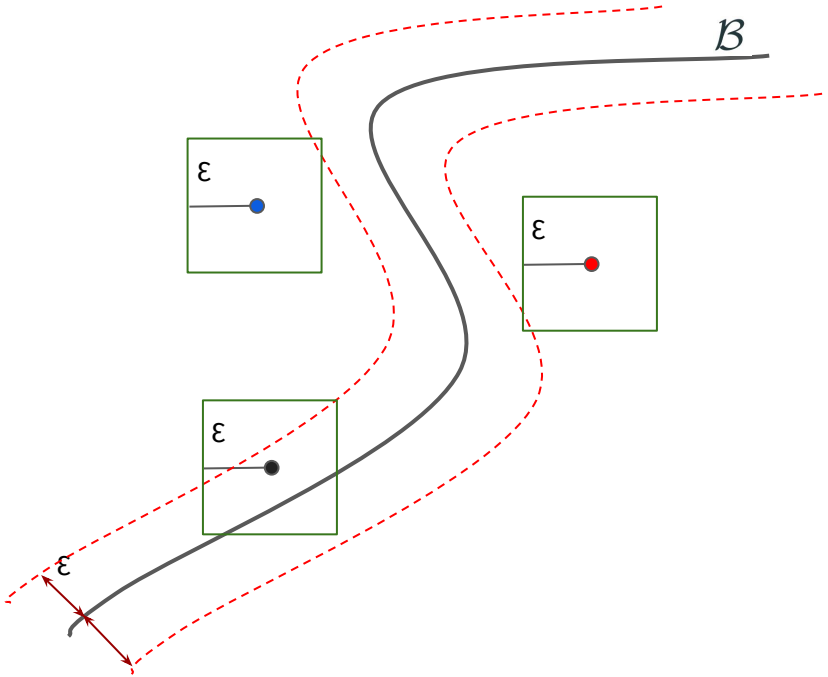
then completely check $\zeta(x)$ is **NP-HARD**

Definition (Minimum Problem Formulation)
Verification can be deduced by solving a minimum problem

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

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

Linear Constraint



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

Complete verification is NP-Hard

Definition (Minimum Problem Formulation)
Verification can be deduced by solving a minimum problem

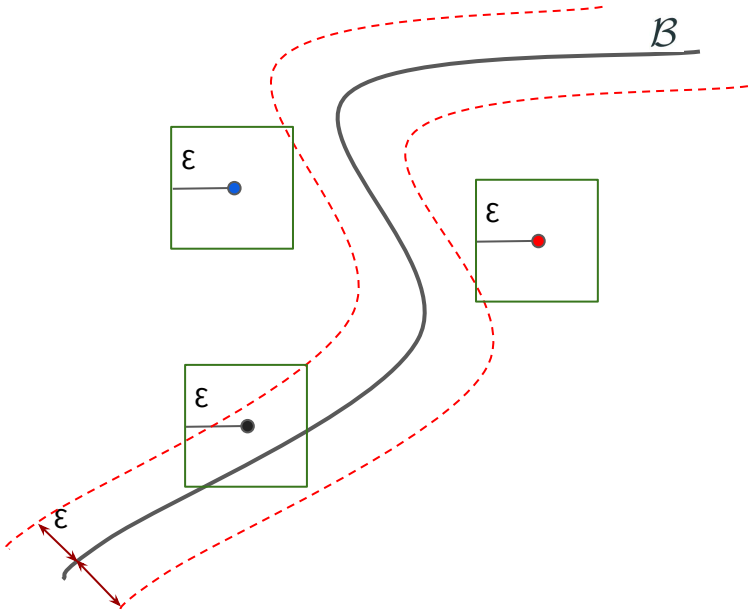
$$P(x) = \min_{j \neq l} \min_{y \in \mathbb{R}^n} \quad f_l(y) - f_j(y)$$
$$\text{s.t.} \quad -\varepsilon \leq x_i - y_i \leq \varepsilon, \forall i$$

Observation
The complete verification of the robustness for ∞ norm

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

is deduced by observing that

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

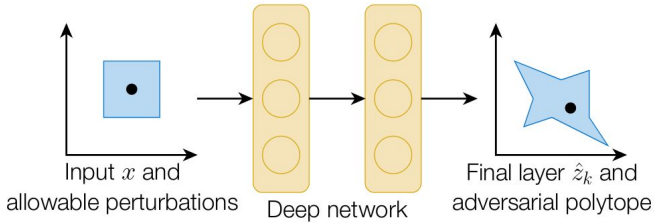


*Guy Katz et al. "*Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks*"

Incomplete verification through relaxation

Deep Neural Network with ReLU

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



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

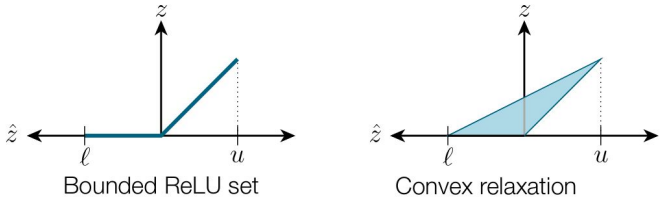
$$P(x) = \min_{j \neq l} \min_{y \in \mathbb{R}^n} \quad \hat{z}_l^{(L)} - \hat{z}_j^{(L)}$$
$$\text{subject to} \quad -\varepsilon \leq x - z^{(0)} \leq \varepsilon$$
$$\hat{z}^{(i)} = W_i z^{(i-1)} + b_i, \quad i = 1, \cdots, L$$
$$z^{(i)} = \max\{0, \hat{z}^{(i)}\}, \quad i = 1, \cdots, L-1$$

NON Linear Constraint

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

Incomplete verification through relaxation

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

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

subject to

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

Relaxed Linear Constraints

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

Incomplete verification through relaxation

Observation (Relaxation gives Incompleteness)

The relaxed problem $\tilde{P}(x)$ provides an incomplete verification of the robustness. In formulas, the robustness statement

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

can be proved by checking the sign of the relaxed problem

$$\tilde{P}(x) > 0 \quad \Rightarrow \quad \zeta(x)$$

Remark The opposite implication is False

$$\tilde{P}(x) \leq 0 \quad \not\Rightarrow \quad \neg\zeta(x)$$

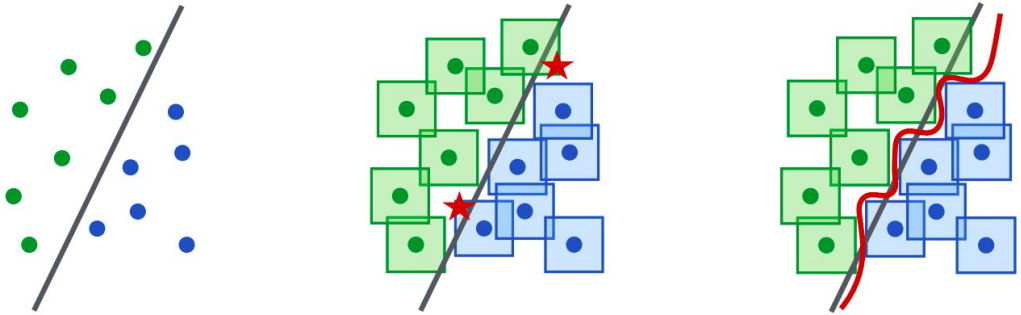
Proof

Let $\mathcal{C}_\varepsilon$ be the set of feasible point with the non linear constraint, and let $\tilde{\mathcal{C}}_\varepsilon$ the relaxed linear constraints. Since $\mathcal{C}_\varepsilon \subseteq \tilde{\mathcal{C}}_\varepsilon$ the following inequality holds $\tilde{P}(x) \leq P(x)$.

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.

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

Robust Loss Function

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

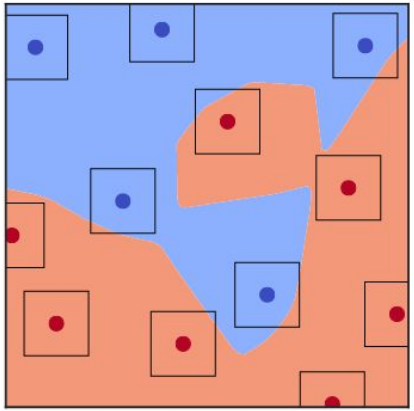
Convex Relaxed Robust Minimum Problem

Observation (No Proof)
Robust loss function can be upper bounded by

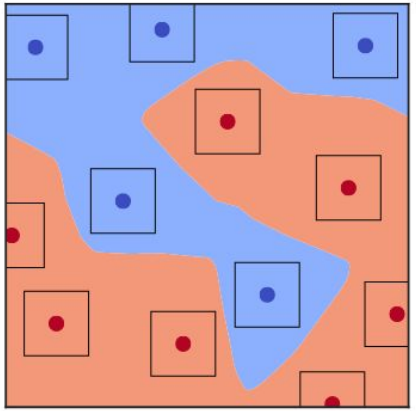
$$\max_{\|\delta\|<\varepsilon} L(f_\theta(x+\delta), l) \leq \tilde{L}_\varepsilon(x, l; \theta)$$

and the solution of the the (RP) is approximated with a **sub-optimal** solution.

$$\min_\theta \frac{1}{N}\sum_{i=1}^{N} \max_{\|\delta\|<\varepsilon} L(f_\theta(x_i+\delta), l_i) \leq \min_\theta \frac{1}{N}\sum_{i=1}^{N} \tilde{L}_\varepsilon(x_i, l_i; \theta)$$



Standard Training



Robust Training

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

Verification by Estimating the Lipschitz Constant

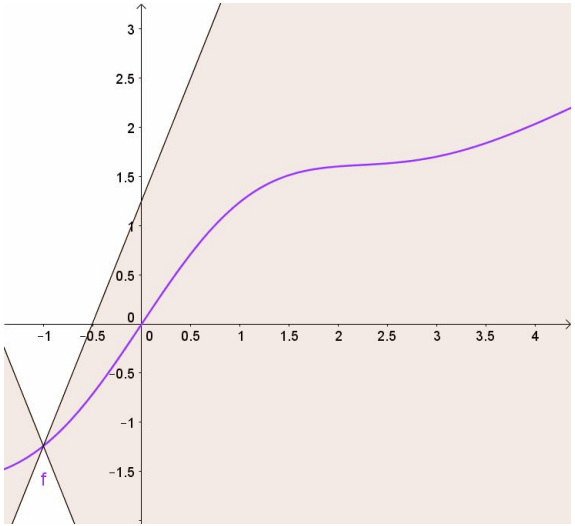
Definition (L-Lipschitz)
A function f is L-lipschitz with respect to the l^p norm if

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

Definition (Local L-Lipschitz property)
A function f is locally L-lipschitz with respect to the l^p norm

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

Observation
The curve's slope is lower than L



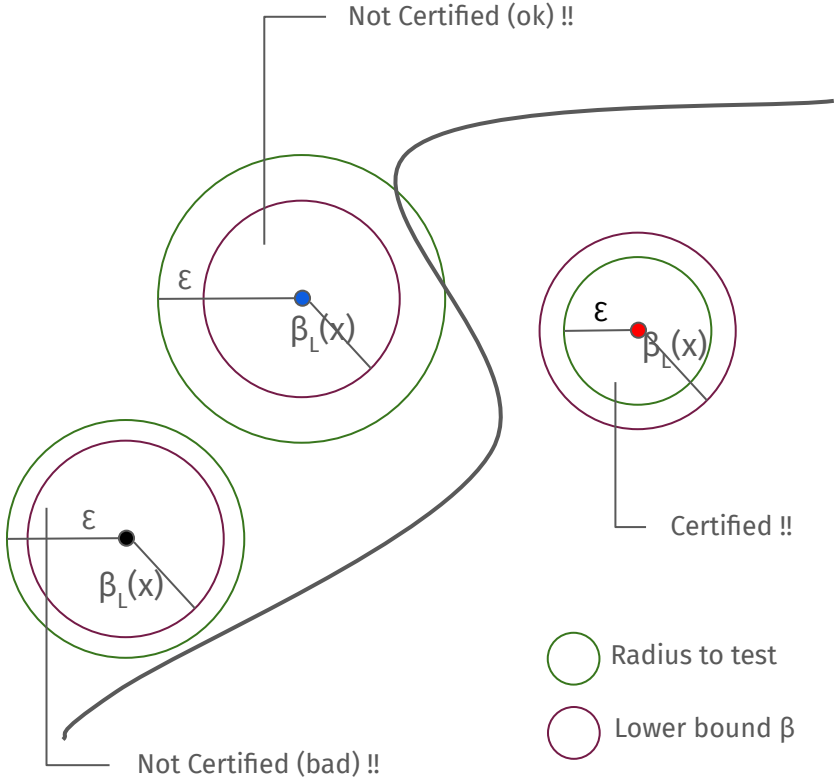
Gif taken from https://en.wikipedia.org/wiki/Lipschitz_continuity#/media/File:Lipschitz_Visualisierung.gif

Verification by knowing the (local) lipschitz constant

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

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

is a bound of the *Minimal Adversarial Perturbation* in l^p norm.



Remark
A lower bound of the MAP provides an incomplete verification. If $\zeta(x)$: "$\forall y \in \mathcal{N}(x) \quad \mathcal{K}(x) = \mathcal{K}(y)$"

$$\text{"}\varepsilon < \beta_L(x)\text{"} \Rightarrow \zeta(x)$$

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

Estimating the lipschitz constant by the gradient

Theorem (Cross Lipschitz Bound)

Let $f : \mathbb{R}^n \to \mathbb{R}^C$, and let x classified as l, if $f_l - f_j$ is L_j - Lipschitz in the neighborhood $B_p(x, R)$, then

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

is still a lower bound of the minimal adversarial perturbation in the l^p norm.

Cross Lipschitz Constant

Proof

Let δ bounded in norm by $\beta_L(x)$. Consider the inequality given by the definition of lipschitz function for any j and consider the left side.

$$(f_l - f_j)(x + \delta) \geq (f_l - f_j)(x) - L_j \beta_L(x) \quad (1)$$

The right hand of Equation (1) is positive by definition.

Estimating the lipschitz constant by the gradient

Theorem (Lipschitz upper bound)
The maximum of the gradient is an upper bound of the cross-lipschitz constants. Let L_j be defined by

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

where q is the dual number $\frac{1}{p} + \frac{1}{q} = 1$, then $f_l - f_j$ are L_j- lipschitz locally in a radius R.

Proof
Apply the fundamental theorem of integrals to $g(t) = (f_l - f_j)(x + t\delta)$ to deduce the following equality

$$|(f_l - f_j)(x+\delta) - (f_l - f_j)(x)| = \int_0^1 \frac{d}{dt} g(t)\, dt \quad .$$

Use the Cauchy-Schwarz inequality to deduce that

$$\left| \int_0^1 \nabla(f_l - f_j) \cdot \delta \, dt \right| < \|\delta\|_p \int_0^1 \|\nabla(f_l - f_j)\|_q \, dt$$

and take the maximum in the neighborhood to conclude.

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

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

Keydea

Estimate the maximum with multiple samplings.

2 **for** $i \leftarrow 1$ **to** N_b **do**
3 **for** $k \leftarrow 1$ **to** N_s **do**
4 randomly select a point $\boldsymbol{x}^{(i,k)} \in B_p(\boldsymbol{x_0}, R)$
5 compute $b_{ik} \leftarrow \|\nabla g(\boldsymbol{x}^{(i,k)})\|_q$ via back propagation
6 **end**
7 $S \leftarrow S \cup \{\max_k\{b_{ik}\}\}$
8 **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

Remark

1. Computationally expensive.
2. Not certifiable since only a lower bound of the maximum can be found.

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

Summary: Verification Methods

Advantages

1. Verification methods are **highly reliable** since they are based on the solution of well founded MPs
2. Can be involved in a training process to improve the (empirical) robustness of a model classification

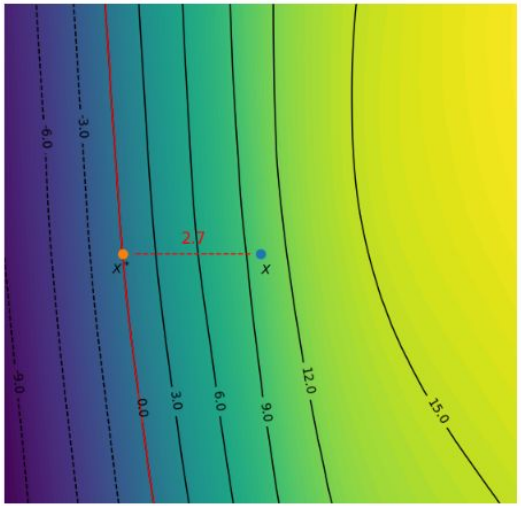
Disadvantages

1. Verifications methods do not scale to larger networks or are typically computational expensive
2. Can require a complete knowledge of the model's architecture and hidden states.

Lipschitz Bounded Neural Networks

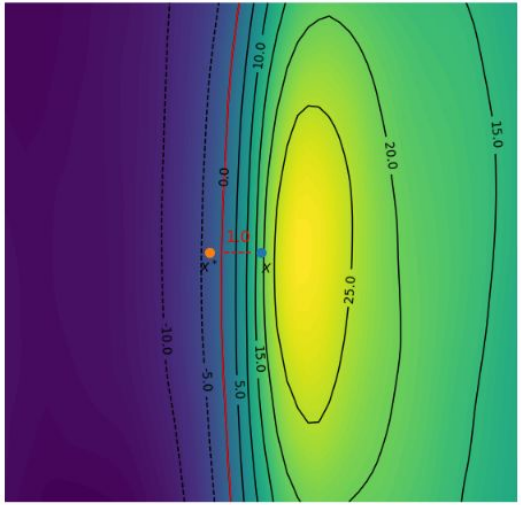
Lipschitz constant of Neural Networks

LeNet-5



(a) MNIST

ResNet-32



(d) CIFAR10

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

Lipschitz constant of Neural Networks

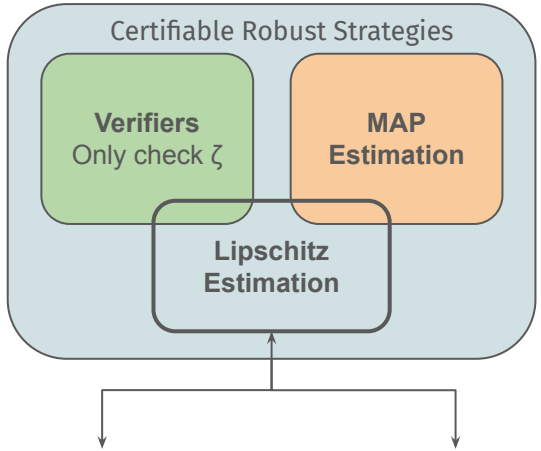
Observation.
Feedforward Neural Networks with *linear*, *convolutional* and *residual* layers are L-Lipschitz for some constant L.

MLP
LeNet
ResNet
U-Net
} L-Lipschitz

Standard trainings don't care about L

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



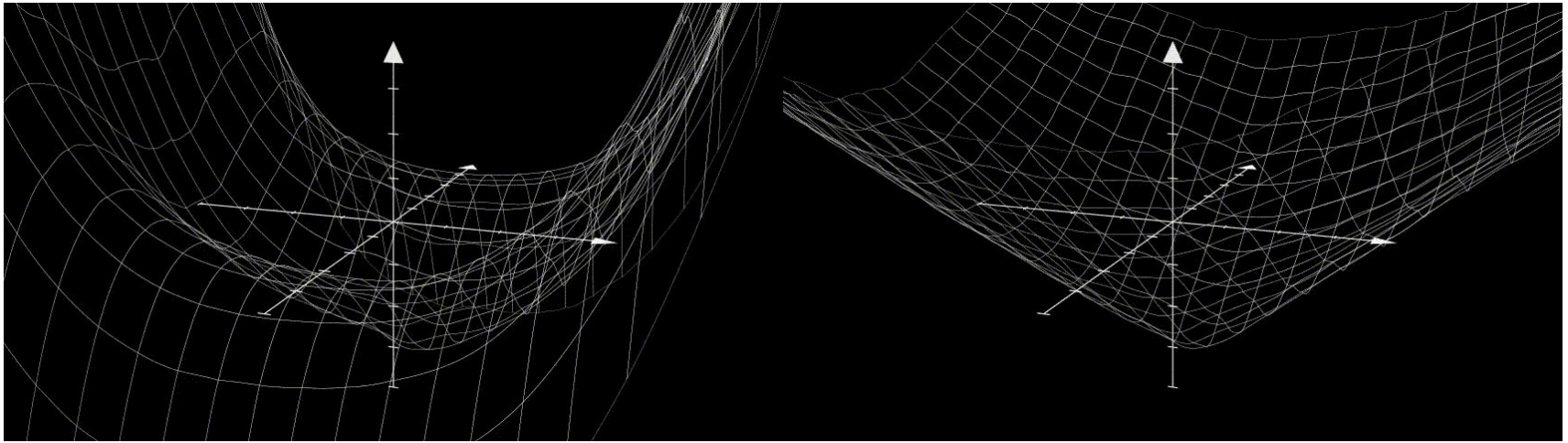
Regularized training

Search for f that is L-lipschitz bounded

Aim of the Section

Are Lipschitz bounded DNN good classifier?

Graphical representation of level curves of a 1-lipschitz function.



Observation The 0-level curve (white) is the decision boundary and it is the same for both functions. In formulas,

$$\forall x \in \mathbb{R}^n, \quad \mathcal{K}_f(x) = \mathcal{K}_{\tilde{f}}(x) \qquad \text{where} \qquad \tilde{f}(x) = \frac{f(x)}{L} \text{ is 1-Lipschitz.}$$

Common deep neural networks are lipschitz

Definition (L-Lipschitz)
A function f is L-lipschitz with respect to the l^p norm

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

Observation (Composition)

Composition of lipschitz functions is lipschitz

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

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

Examples of Lipschitz Layers

Fully connected, Convolutional, Residual, Pooling

Remark.
Common Deep Neural Networks are Lipschitz

Linear Layers are lipschitz

Definition (Operatorial Norm)

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

When p=∞, then norm is called *uniform*, if p=2 the norm is called spectral norm.

Observation

Affine functions expressed by $f(x) = Wx + b$ are $\|W\|_p$- lipschitz

where $\|W\|_p$ is the operatorial norm of the weight matrix.

Proof

Consider the following chain of equalities

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

$$(y - x = v)$$

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

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

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

1-Lipschitz Linear Layers

Observation
The spectral norm is the largest singular value of the matrix

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

Observation (Spectral Normalization)
The following layer is 1-lipschitz.

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

Note that the applied weight is parameterized through W.

Normalized linear layer

```python
def normalized_linear(x: Tensor, weight: Tensor, b: Tensor)
    r"""Compute a 1-lipschitz fully connected operation."""
    m, n = weight.shape
    v = torch.randn(n, 1)  # Random initial vector
    v = v / torch.norm(v, p=2)  # Normalize to have unitary
    for _ in range(MAX_NUM_ITERS):
        u = weight @ v
        u = u / torch.norm(u, p=2)

        v = weight.T @ u
        v = v / torch.norm(v, p=2)
    sigma = u.T @ weight @ v
    weight = weight / sigma
    return linear(x, weight, b)
```

Remark
Power method can be used for estimating the

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

Orthogonal linear layers

Definition (Orthogonal Matrix)
A square matrix Q is orthogonal if and only if

$$QQ^T = Q^TQ = I$$

Observation
An fully connected layer with an orthogonal weight

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

is 1-Lipschitz with respect to the euclidean norm.

Bjorck Orthogonalization
The following iterative method converge to an orthogonal matrix starting from $Q_0 = W$ if $\|W\|_2 \leq 1$,

$$A_k = I - Q_k^T Q_k$$
$$Q_{k+1} = Q_k \left(I + \frac{1}{2}A_k + \frac{3}{8}A_k^2 + \cdots + (-1)^p \binom{-\frac{1}{2}}{p} \right).$$

Remark
The parameterized weight Q_k is orthogonal for k~20, and depends in a differentiable manner from W.

Cem Anil et al. "*Sorting Out Lipschitz Function Approximation*". (2019)
Björck and Bowie. *"An iterative algorithm for computing the best estimate of an orthogonal matrix"*. (1971)

Orthogonal linear layers

Cayley Transformation[a]

$$A = W - W^T$$

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

Exponential Map[b]

$$A = W - W^T$$

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

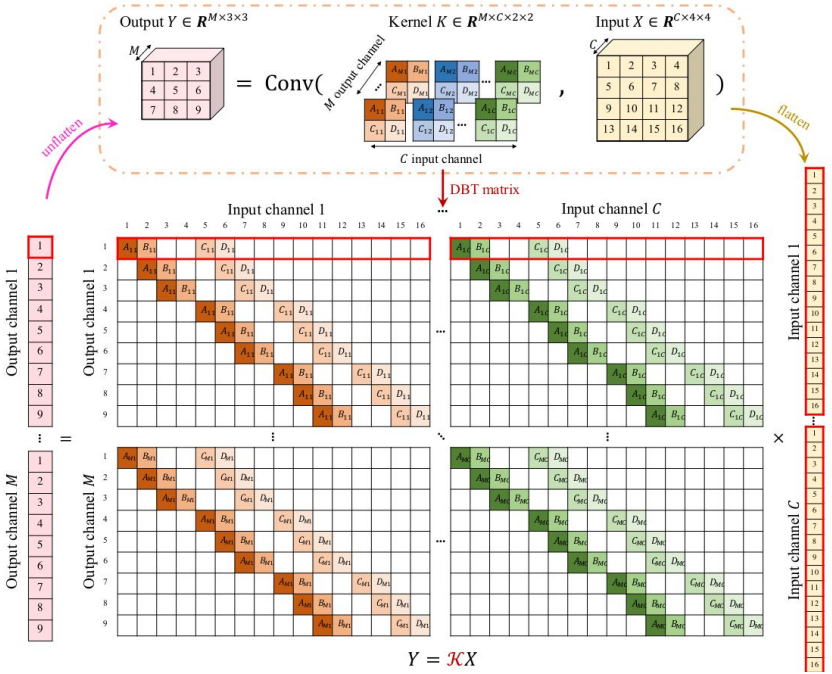
[a] Asher Trockmann et al. "*Orthogonalizing Convolutional Layers with the Cayley Transform*"

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

1-Lipschitz Convolutions (no details)

Observation
Convolutions are Lipschitz functions.

Proof
Convolutions are particular case of (sparse) linear layers, that can be represented through a **double-block Toeplitz** matrix.



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

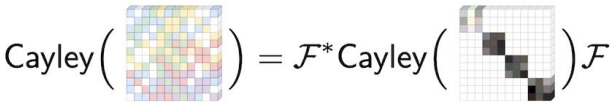
Orthogonal Convolutions (no details)

Cayley Transformation[a]

$$A = W - W^T$$

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

Orthogonalizing a
multi-channel convolution...

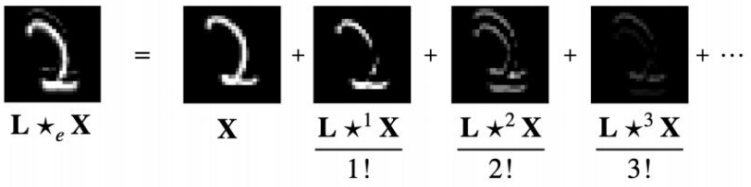


...can be done *efficiently* by orthogonalizing
a Fourier-domain **block-diagonal matrix**.

Exponential Map [b]

$$A = W - W^T$$

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

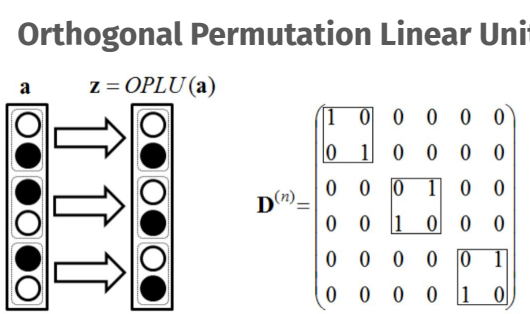


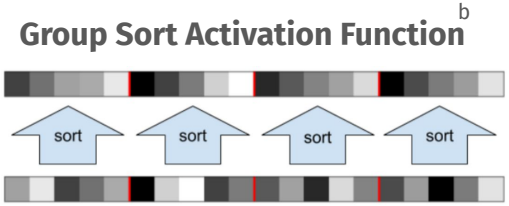
(c) Convolution exponential $(\mathbf{L} \star_e \mathbf{X})$

[a] Asher Trockmann et al. *"Orthogonalizing Convolutional Layers with the Cayley Transform"*
[b] Sahil Singla. *"Skew Orthogonal Convolutions"*.

1-Lipschitz activation functions

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]		$\begin{cases} \alpha\left(e^x - 1\right) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ 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





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

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

Evaluation of the CRA

Definition (ε-robust accuracy)
Is the ratio of correct ε-robust classifications

$$\mathcal{A}_R(f, \varepsilon) = \mathbb{P}(\mathcal{K}_f(\mathbf{x} + \delta) = \mathcal{O}(\mathbf{x}),\ \forall \|\delta\| < \varepsilon)$$

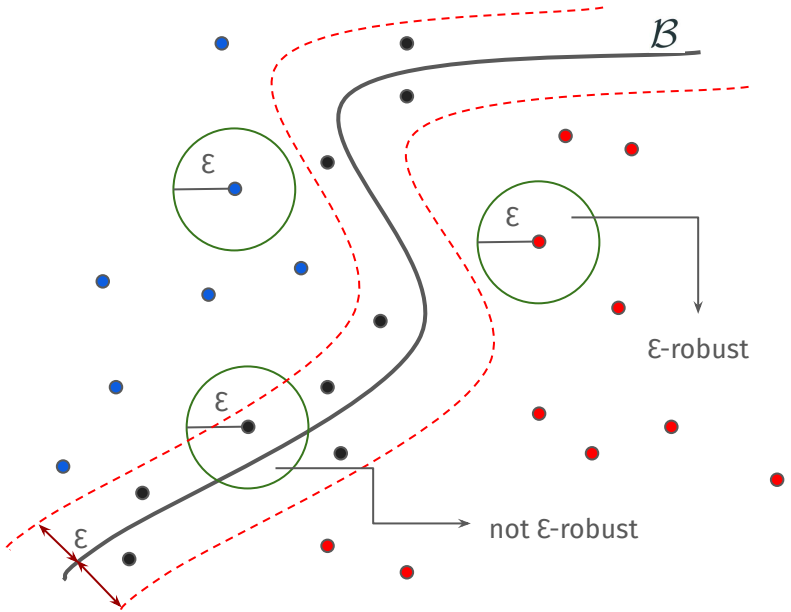
Reminder (Lower bound of MAP)
Let us assume f be local L-lipschitz in a large radius R, then

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

is a bound of the *Minimal Adversarial Perturbation* in l^p norm.

Definition (ε-robust accuracy - operative -)
Is the ratio of correct classifications far from the boundary

$$\tilde{\mathcal{A}}_R(f, \varepsilon) = \mathbb{P}\left(\mathcal{K}_f(\mathbf{x}) = \mathcal{O}(\mathbf{x}),\ \beta_L(\mathbf{x}) > \varepsilon\right)$$



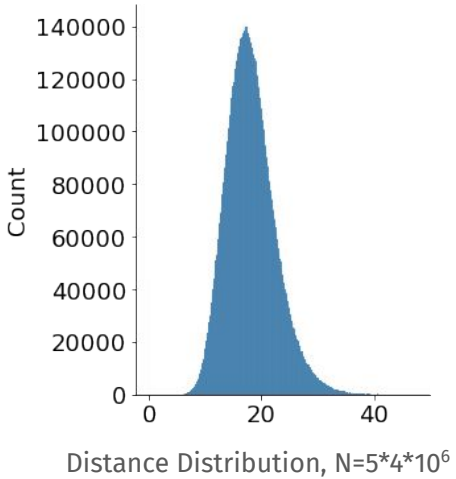
Theoretical maximum CRA for CIFAR-10

	plane	auto	bird	cat	deer	dog	frog	horse	ship	truck
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 closest classes

Two farthermost classes

Euclidean pairwise distances measured on the CIFAR-10 dataset. Theoretical 100% accuracy is possible for **ε = 1.8** , since is half of the distance between the two closest images.



Distance Distribution, N=5*4*10^6

Evaluating the CRA on CIFAR-10

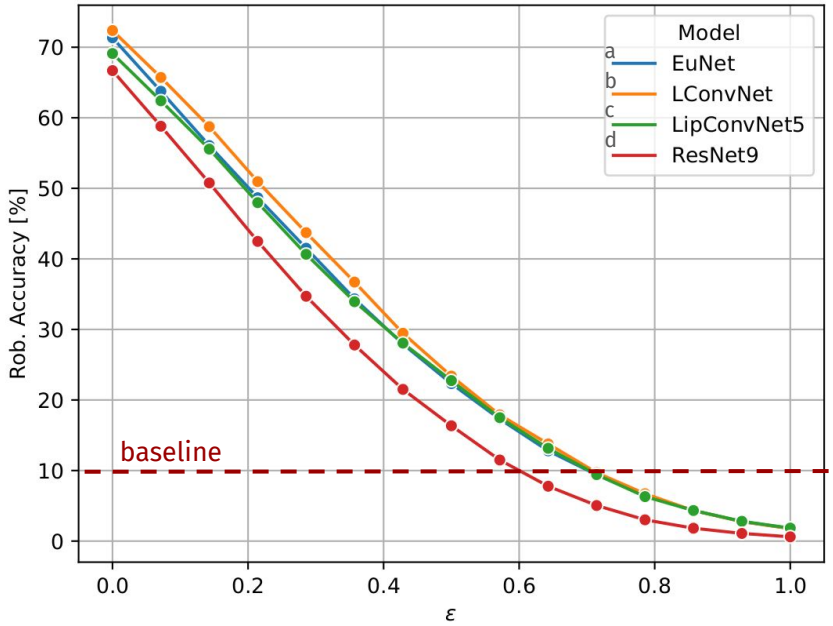
CRA on CIFAR 10

1. Increasing ε, the CRA [%] drops
2. Even with small values of ε, the cra of Lipschitz models is particularly lower than then accuracy

Definition (ε-robust accuracy - operative -)
Is the ratio of correct classifications far from the boundary

$$\tilde{\mathcal{A}}_R(f, \varepsilon) = \mathbb{P}\left(\mathcal{K}_f(\mathbf{x}) = \mathcal{O}(\mathbf{x}),\ \beta_L(\mathbf{x}) > \varepsilon\right)$$

Update (January 2024)
CRA on CIFAR-10 has been increased up to 78 % !!



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

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

Summary: Lipschitz Bounded Neural Networks

Advantages

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

Disadvantages

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 still the SOTA of

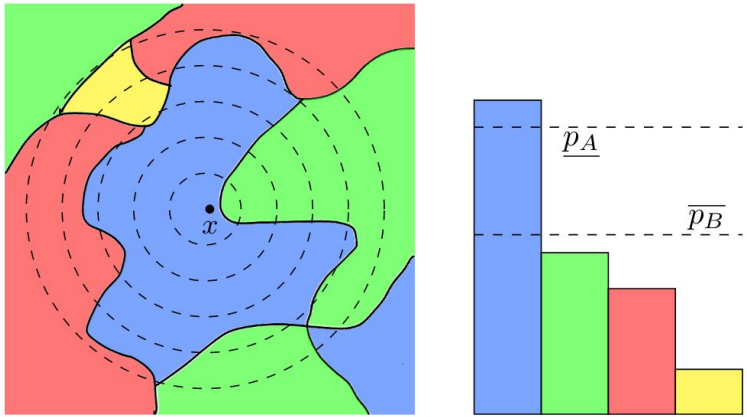
Randomized Smoothing

Randomized Smoothing Strategy

Definition (Smooth Classifier)
Given a base classifier $f : \mathbb{R}^n \to \{1, \cdots, C\}$, and a value σ, the *smooth classifier* g_σ is defined by

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

where $\mathcal{N}(0,\sigma I)$ is the gaussian distribution.



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 of the RS strategy

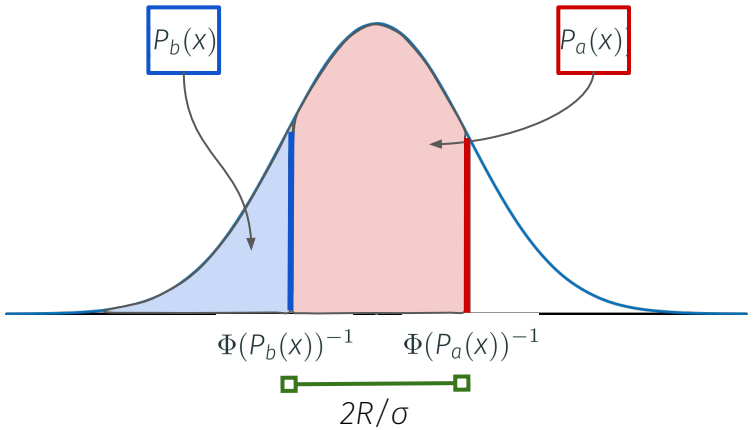
Theorem (Certification radius of RS)
Let $P(x)$ be the vector of probabilities defined by

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

and let a and b the top-2 most probable classes. Then $g_\sigma(x)$ is certifiable $R(x)$-robust for

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

where Φ is the cumulative gaussian distribution function.



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

Certifiable robust classification proof (sketch)

Proof (Part I) (for σ = 1)
Let $P(x)$ be the vector of probabilities defined by

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

Observe that by definition of density function

$$P_c(x) = \int_{\mathbb{R}^n} \mathbb{1}_{\{f(x)=c\}}(x+\varepsilon)\,\mathcal{N}(0,I)(\varepsilon)d\varepsilon$$

and by changing variable

$$P_c(x) = \int_{\mathbb{R}^n} \mathbb{1}_{\{f(x)=c\}}(t)\,\mathcal{N}(0,I)(x-t)dt$$

that is the convolution with the gaussian density function of the base classifier f

$$P_c(x) = (\mathbb{1}_{\{f(x)=c\}} * \mathcal{N}(0,I))(x)\ .$$

Lemma (Salmann)
Convolving with the gaussian kernel produces a lipschitz function, from which we deduce that

$$\forall c, \quad G_c(x) = \Phi^{-1}(P_c(x)) \text{ is 1-Lipschitz}$$

Proof (Part II)
Since G is 1-lipschitz for each component, then the cross lipschitz constant are L_j = 2. By applying the Hein theorem for the certifiable robustness we deduce that

$$\begin{aligned}\beta_L(x) &= \min_{j\neq a} \frac{G_a(x) - G_j(x)}{2} \\ &= \frac{1}{2}\left(G_a(x) - \max_{j\neq a} G_j(x)\right) \\ &= \frac{1}{2}\left(G_a(x) - G_b(x)\right) = R(x)\end{aligned}$$

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

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

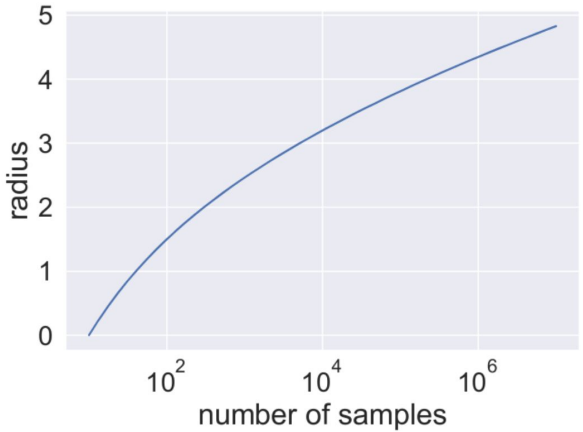
Let ε_1,..., ε_n sampled from $\mathcal{N}(0,\sigma I)$

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

with a confidence level of 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



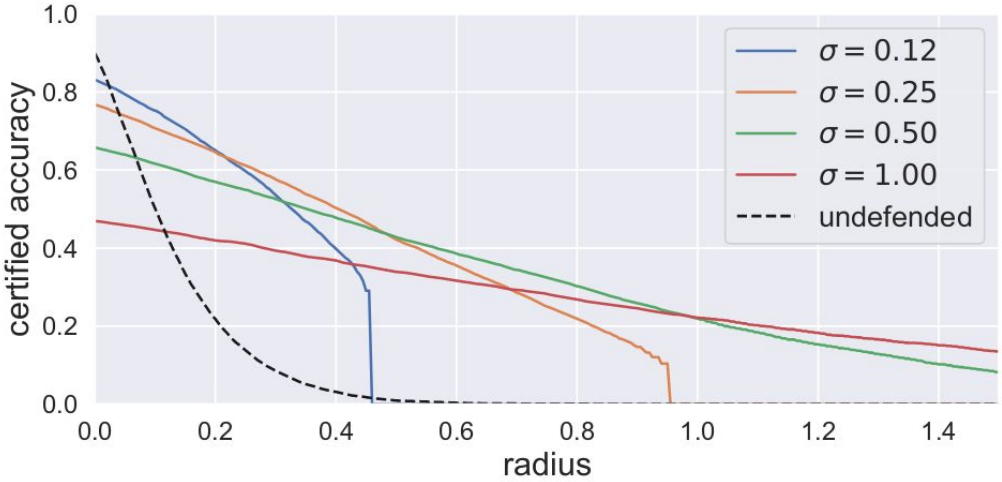
Jeremy Cohen et al. "*Certified Adversarial Robustness via Randomized Smoothing*"

Evaluation of certified robust accuracy for RS

Remind (robust accuracy - operative)

$$\tilde{\mathcal{A}}_R(f,\varepsilon) = \mathbb{P}\left(\mathcal{K}_f(\mathbf{x}) = \mathcal{O}(\mathbf{x}),\ \beta_L(\mathbf{x}) > \varepsilon\right) \qquad \text{where} \qquad \beta_L(x) = R = \frac{\sigma}{2}\left(\Phi(P_a(x))^{-1} - \Phi(P_b(x))^{-1}\right)$$

CIFAR10, confidence 99.999%, samples 10^5



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

Smooth Adversarial Training

Definition (Base Classifier)
In the case of classifier deduced by a DNN

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

Definition (Soft Smooth Classifier)
The (hard) smoothed classifier can be substituted by

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

from which classes are deduced by argmax

Definition (Smooth Attack)

The attacker can leverage the smoot classifier to deduced an attack, where the expectation

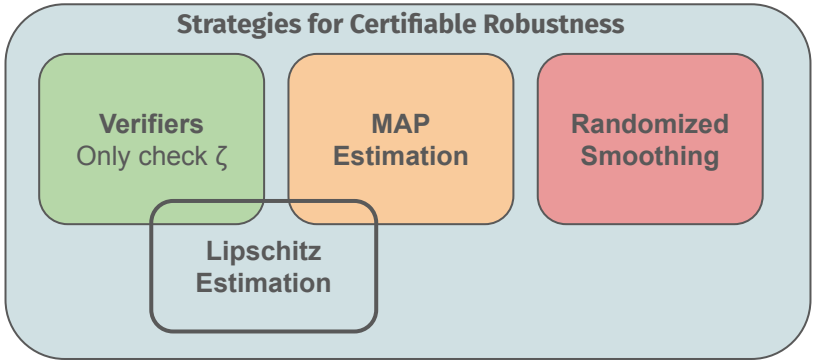
$$\begin{aligned}\hat{x} &= \operatorname*{argmax}_{\|z-x\| \leq \rho} \mathcal{L}_{CE}\left(G_\sigma(z), c\right) \\ &= \operatorname*{argmax}_{\|z-x\| \leq \rho} \left(-\log \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma I)}\left[F(x+\varepsilon)_c\right]\right)\end{aligned}$$

is approximated with a monte carlo sampling.

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

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

 DIEE, Università di Cagliari

 fabio.brau@unica.it

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

 retis.santannapisa.it/~f.brau