

May 24, 2020

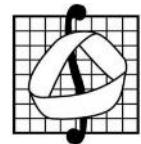
Certified Robustness for modern NN

Aleksandr Petiushko



MSU, PhD in Mechanics and Mathematics

Huawei, principal engineer and AI expert





Intelligence Systems and Data Science Technology Center academic collaborations



St. Petersburg State University



Skolkovo Institute of Science and Technology



LOMONOSOV MOSCOW
STATE UNIVERSITY



Huawei Educational Program

- Starting from 2019, Huawei is presenting **SHARE** educational program: **School of Huawei Advanced Research Education**
 - Школа опережающего научного образования Хуавэй
- Our Intelligence Systems and Data Science Technology Center is doing courses in MSU:
 - 2 year full-educational program
 - 12 semester courses
 - 2 main directions:
 - Computer Vision and Machine Learning specialization
 - Big Data and Information Theory specialization



Sources

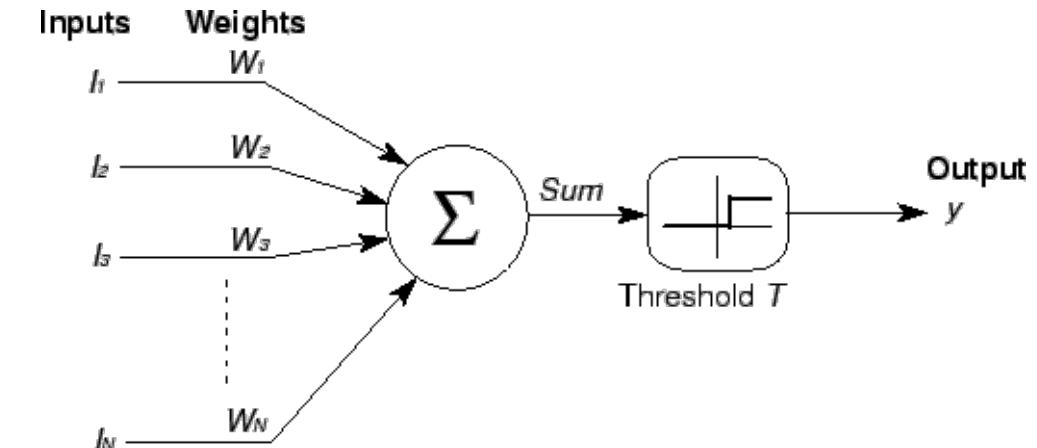
- This presentation is prepared based on:
 - Course “**Modern Computer Vision**”¹ on department Mechanics and Mathematics of MSU
 - **NAUKA-2019**² open lecture
 - Internal discussions inside Intelligence Systems and Data Science Technology Center, Moscow Research Center, Huawei

[1] <https://github.com/mlcoursemm/cvcoursemm2019autumn>

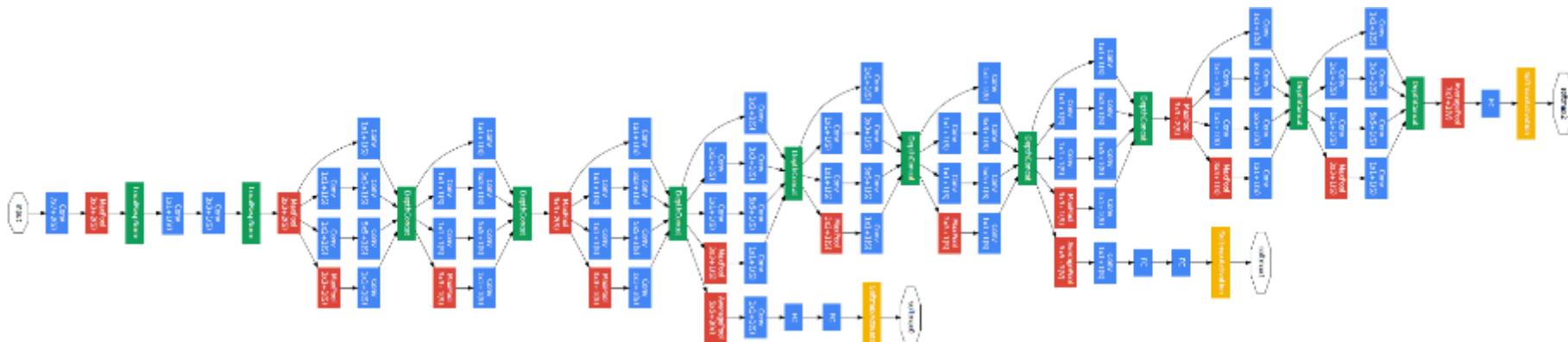
[2] <https://www.youtube.com/watch?v=tKPSqKYuLTc>



Neural Nets progress



- Starting from 1943 – first formal mathematical notion of “artificial neuron” by McCulloch Pitts – Neural Nets became:
 - Wider (more parameters)
 - Deeper (more computational blocks, or layers)
 - Better! (increasing performance on tasks to be solved)



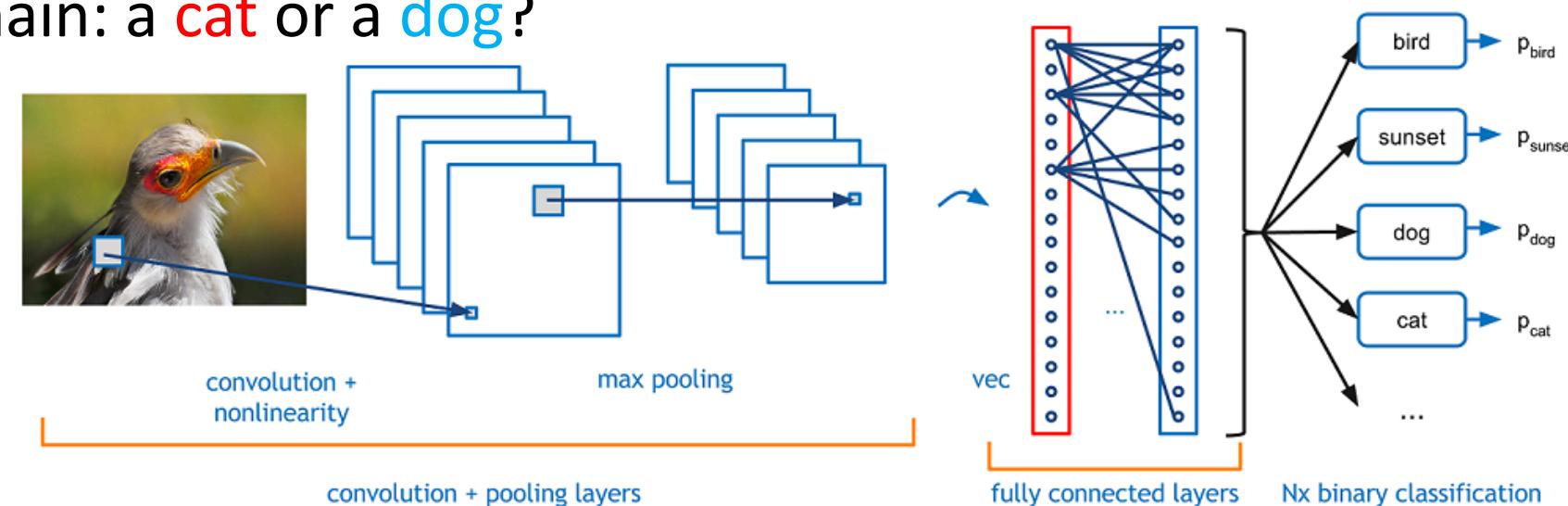


Convolutional Neural Nets



Cat
Duck
Dog

- To work with images and videos **convolutional neural nets (CNN)** are the most suited tool
- E.g. they can detect the object and determine its class
- Can be used to answer the main question in the Computer Vision domain: a **cat** or a **dog**?





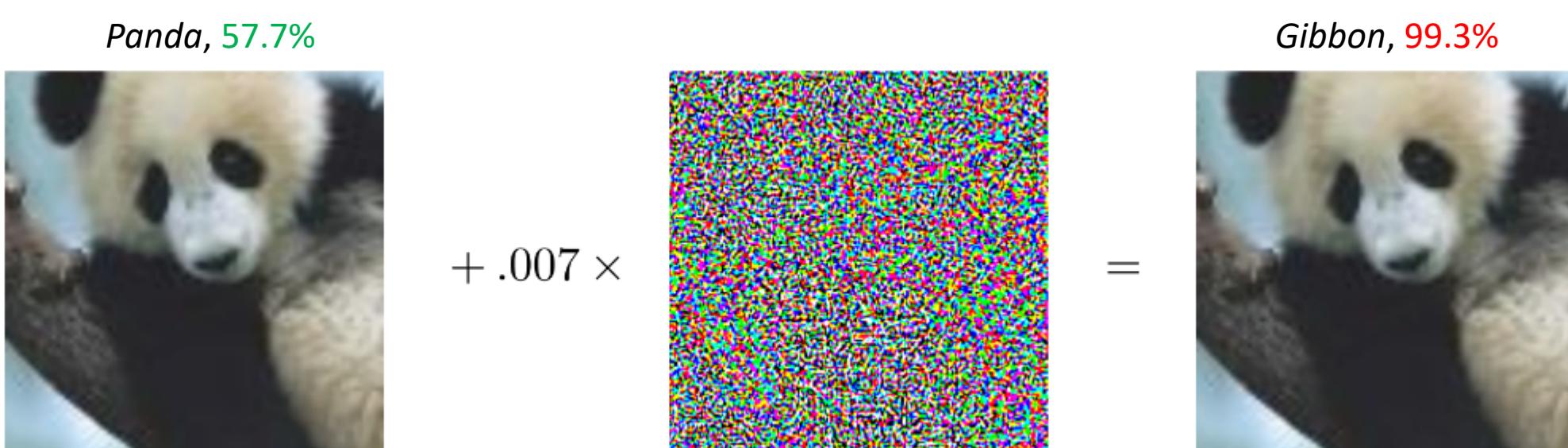
CNNs are better than human!..



- In the most known image recognition benchmark **ImageNet** (classification onto 1000 of classes) **CNNs** now are making top-5 error that is less than **2%**, at the same time for the trained **human** this error is more than **5%**
- For the well known Face Recognition benchmark «**Labeled Faces in the Wild**» **human** error level in **2.5%** was beaten by **CNN** at 2014 by **1%** (and became **1.5%**)
 - At present CNNs are competing between each other using false positive rate $10^{-8} – 10^{-9}$ on much more challenging facial benchmarks

... Or maybe not really?

- It turned out that we can introduce **perceptually invisible** perturbations to the input images, that will lead to the completely different output of the CNN
 - E.g. classification result will change from c «*Panda*» to «*Gibbon*»

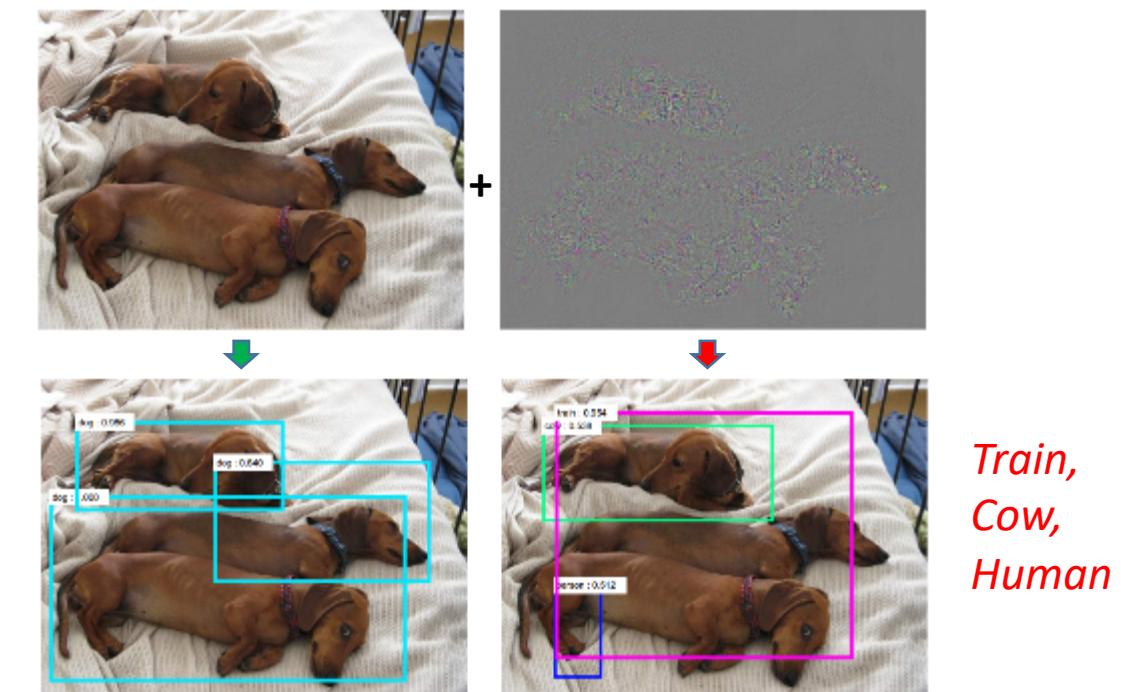


Such a perturbation is called **adversarial example (or attack)**

Other example of adversarials

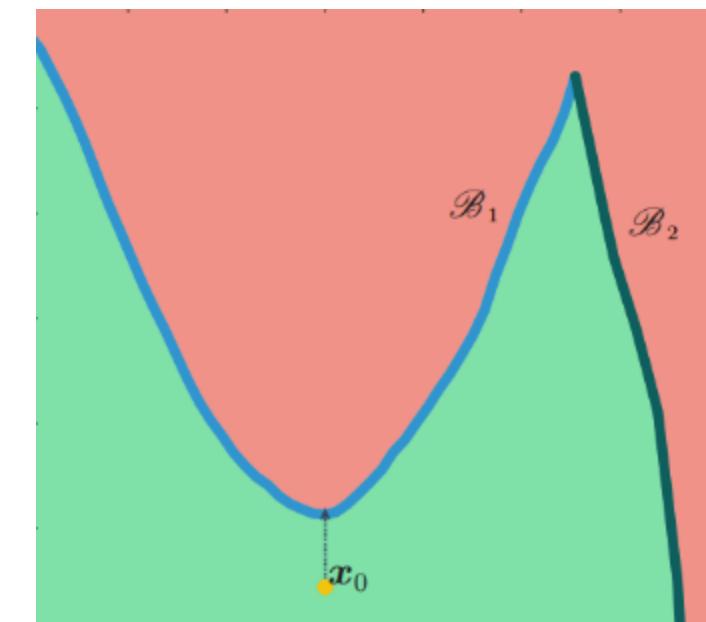
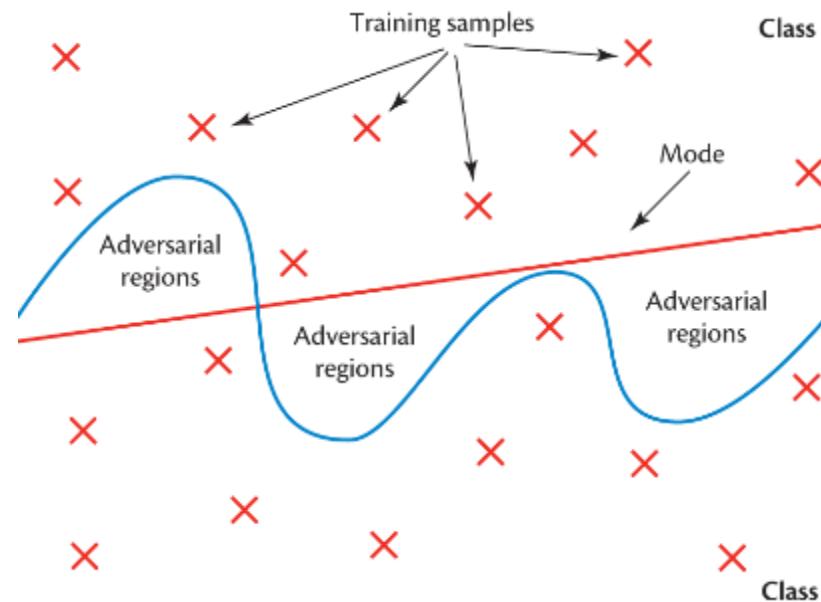
- At the same manner we can fool not only CNNs designed for the classification, but
 - CNN designed for other tasks (detection, segmentation)
 - Recurrent NN for text processing, etc

Article: Super Bowl 50
Paragraph: “Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”
Question: “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”
Original Prediction: John Elway
Prediction under adversary: Jeff Dean



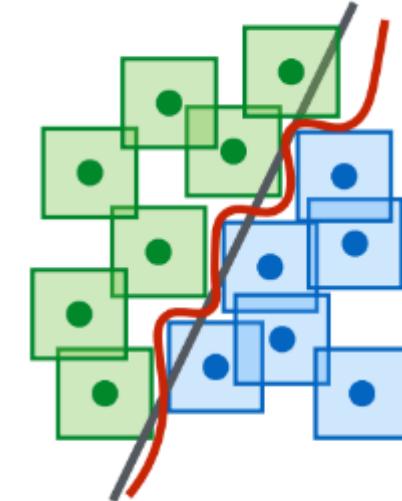
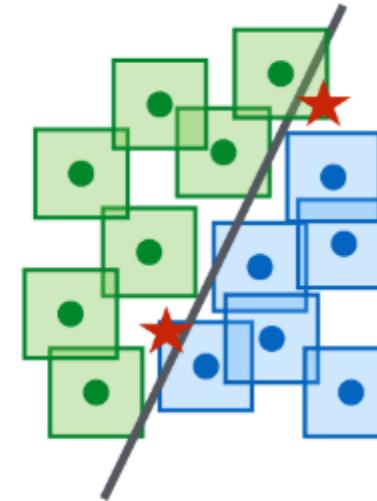
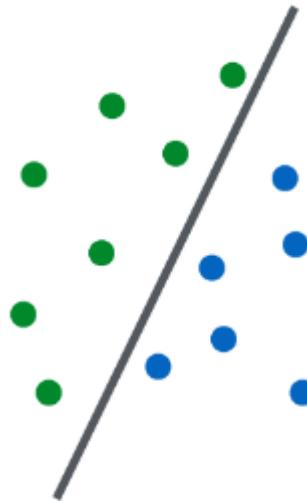
Why it is happening

- One of the main reason of such a behavior by NN on very close inputs
 - the training procedure of NN:
 - The common situation is that classes separating borderlines could be quite close to the training samples and it is very easy to move to the wrong class area by a very small step



Some ways of improving it

- A couple of simple ways – during training:
 - For every training example to add all its pixel vicinity
 - Drawback: exponentially increased number of input samples
 - Add only the most hard examples from this vicinity
 - Drawback: additional backpropagation steps





Certified Robustness - definitions

- Suppose our NN function $f(x)$ is the **classifier** to K classes: $f: R^d \rightarrow Y, Y = \{1, \dots, K\}$
 - Usually we have NN $h(x): R^d \rightarrow R^K$, and $f(x) = \text{argmax}_i h(x)_i$
 - Deterministic approach: we want to find provide the class of perturbation $S(x, f)$ so as the classifier output will not change, or more formally:
 - $f(x + \delta) = f(x) \forall \delta \in S(x, f)$
 - Probabilistic approach: having the probability of robustness P , find the class of input perturbations $S(x, f, P)$ s.t.:
 - $\text{Prob}_{\delta \in S(x, f, P)}(f(x + \delta) = f(x)) = P$
- If NN $f(x)$ is the **regressor**: $f: R^d \rightarrow R$
 - Having the upper and lower bounds on the output perturbation, find the class of input perturbations $S(x, f, f_{low}, f_{up})$:
 - $f(x) - f_{low} \leq f(x + \delta) \leq f(x) + f_{up}, \forall \delta \in S(x, f, f_{low}, f_{up})$



Certified Robustness – definitions (2)

- Also, inverse tasks could be considered
- If NN $f(x)$ is the **classifier** to K classes: $f: \mathbb{R}^d \rightarrow Y, Y = \{1, \dots, K\}$
 - Probabilistic approach: we want to measure the probability of not changing the classifier output under some class of input perturbations S :
 - $\text{Prob}_{\delta \in S}(f(x + \delta) = f(x))$
- If NN $f(x)$ is the **regressor**: $f: \mathbb{R}^d \rightarrow \mathbb{R}$
 - We want to find the upper and lower bounds of the output perturbation under some class of input perturbations S in the analytical form:
 - $f(x) - f_{low}(f, x, S) \leq f(x + \delta) \leq f(x) + f_{up}(f, x, S), \forall \delta \in S$
- Can be measured just by analyzing the output of $f(x + \delta)$

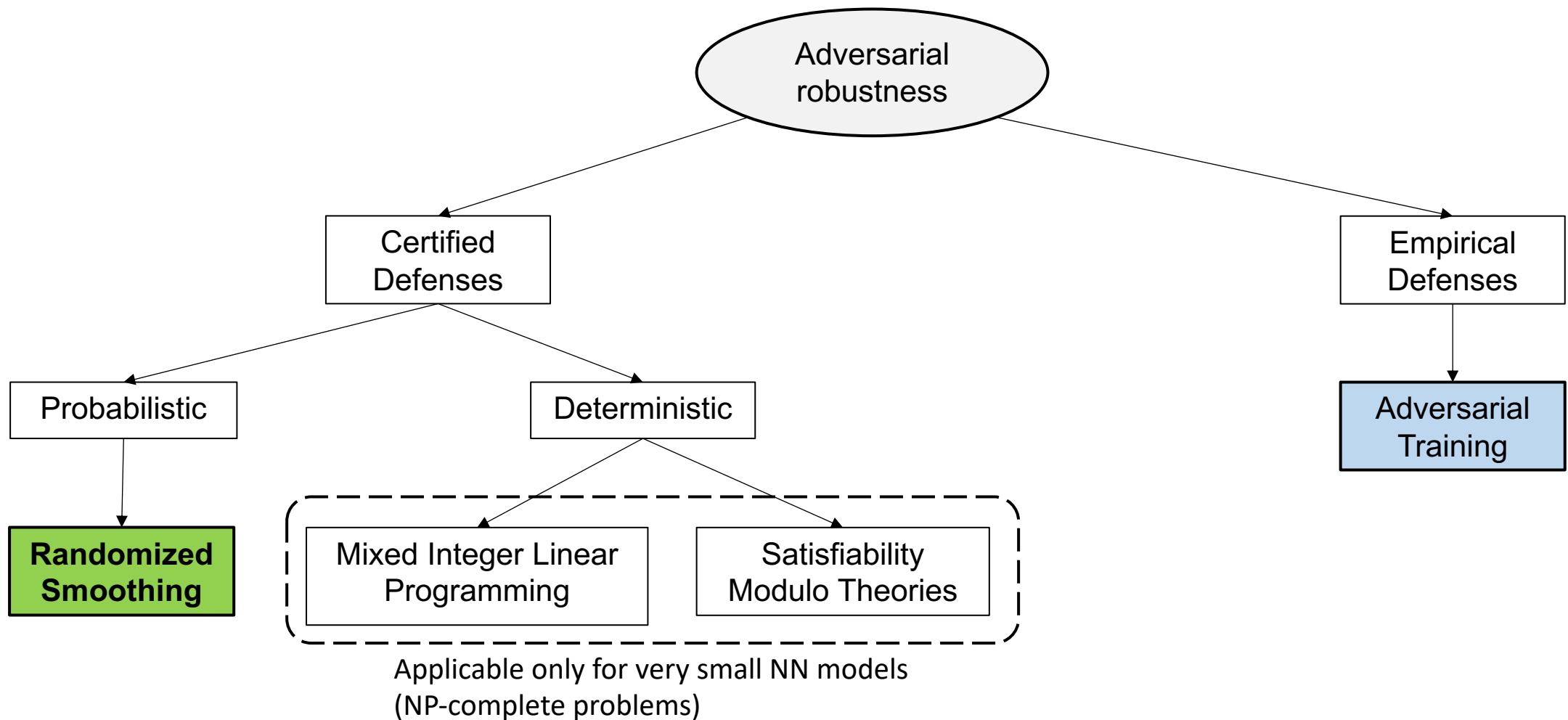


Certified Robustness and Lipschitz Function

- Neural Net output is $h: R^d \rightarrow R^K$, and the classifier itself is $f: R^d \rightarrow Y, Y = \{1, \dots, K\}$, where $f(x) = \text{argmax}_{i \in Y} h(x)_i$
- Consider binary case ($K = 2$), and probabilistic output: $h(x)_1 + h(x)_2 = 1, h(x)_i \geq 0$
- **Lipschitz function** $f: R^d \rightarrow R$ with a Lipschitz constant L : $\forall x_1, x_2$ it is true that $|f(x_1) - f(x_2)| \leq L||x_1 - x_2||$
- **Local Lipschitz function** f with a Lipschitz constant $L(x_0)$: $\forall x \in S(x_0)$ it is true that $|f(x_0) - f(x)| \leq L(x_0)||x_0 - x||$
- Let $j = \text{argmax}_{i \in Y} h(x_0)_i$, and $h(x_0)_j - h(x_0)_{i \neq j} \geq \epsilon \ \forall i \neq j$
- Let $h(x_0)_j$ - local Lipschitz function with a Lipschitz constant $L(x_0)$
- Then if $S(x_0) = \{x: ||x_0 - x|| \leq \frac{\epsilon}{2L(x_0)}\}$, we have $|h(x_0)_j - h(x)_j| \leq L(x_0) \frac{\epsilon}{2L(x_0)} = \frac{\epsilon}{2}$
- As a consequence we'll have $j = \text{argmax}_{i \in Y} h(x)_i$, and $f(x) = f(x_0) = j$ in the vicinity $S(x_0) = \{x: ||x_0 - x|| \leq \frac{\epsilon}{2L(x_0)}\}$, and certified robustness!
- True certified radius can be much bigger than Lipschitz vicinity $S(x_0)$



Adversarial Robustness





Empirical VS Certified

- **Empirical**
 - Upper bound on the true robustness accuracy
 - But only until the new stronger attack appears
- **Certified**
 - Lower bound on the true robustness accuracy
 - It is what has been theoretically proven, and no one attack can beat it



Empirical robustness: adversarial training

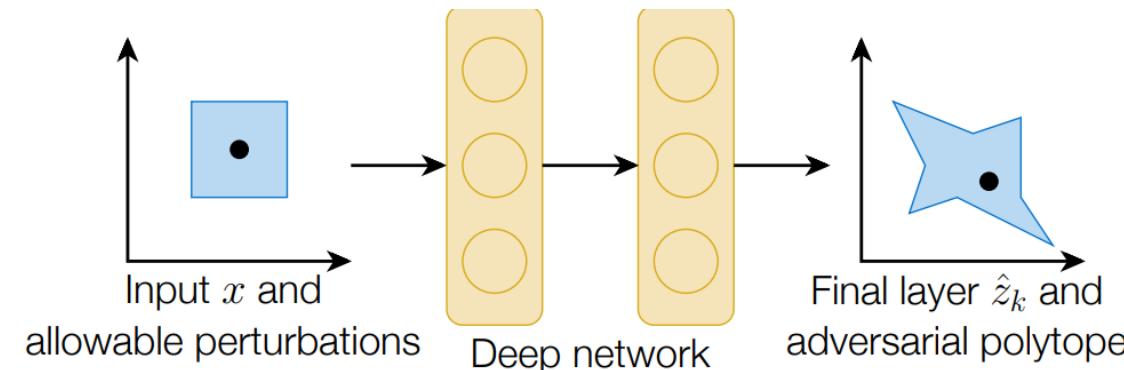
- **Main idea:**
 - train on the most hard examples using some class of perturbations $S (= \Delta)$ around training examples

$$\min_{\theta} \mathbf{E}_{x,y} [\text{Loss}(f_{\theta}(x), y)] \implies \min_{\theta} \mathbf{E}_{x,y} [\max_{\delta \in \Delta} \text{Loss}(f_{\theta}(x + \delta), y)]$$

- **Drawbacks:**
 - Quite inefficient training (longer than usual because of finding of hard examples for every training sample for every iteration)
 - The accuracy on clean samples is lower than for usual training

The problem with l_p -balls

- Usually the robustness is studied under the l_p -balls perturbations of input (S : $\|\delta\|_p \leq \varepsilon$)
- The problem is while the input l_p -ball is convex, for the output it could be of any form and convexity
- That's why it is hard to prove anything



Convex relaxation¹

- **Main idea:**
 - Let's make our regions convex by relaxation!

Corollary 1. For a data point x , label y^* and $\epsilon > 0$, if

$$J_\epsilon(x, g_\theta(e_{y^*} 1^T - I)) \geq 0 \quad (15)$$

(this quantity is a vector, so the inequality means that all elements must be greater than zero) then the model is guaranteed to be robust around this data point. Specifically, there does not exist an adversarial example \tilde{x} such that $\|\tilde{x} - x\|_\infty \leq \epsilon$ and $f_\theta(\tilde{x}) \neq y^*$.

cannot be generalized to ImageNet

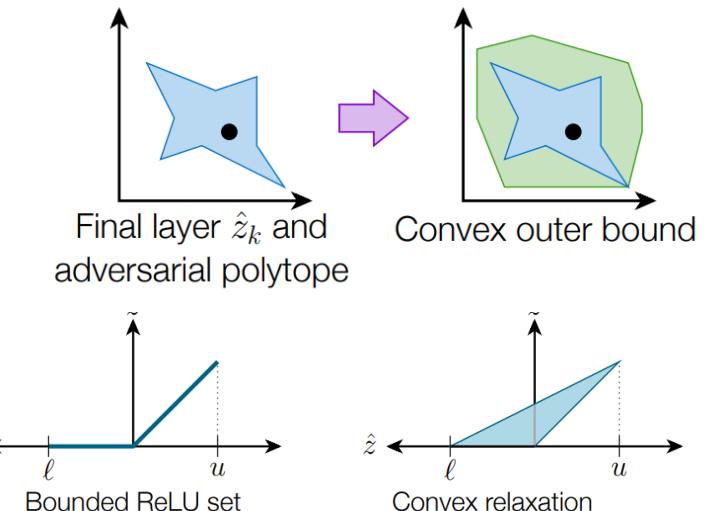


Figure 2. Illustration of the convex ReLU relaxation over the bounded set $[\ell, u]$.

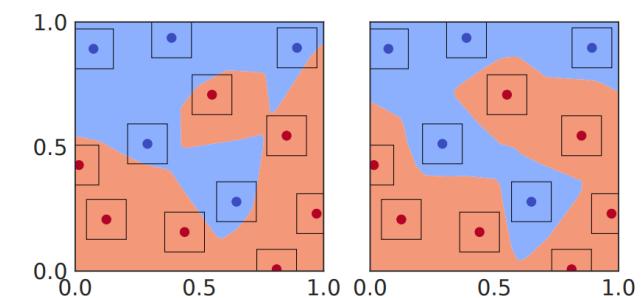
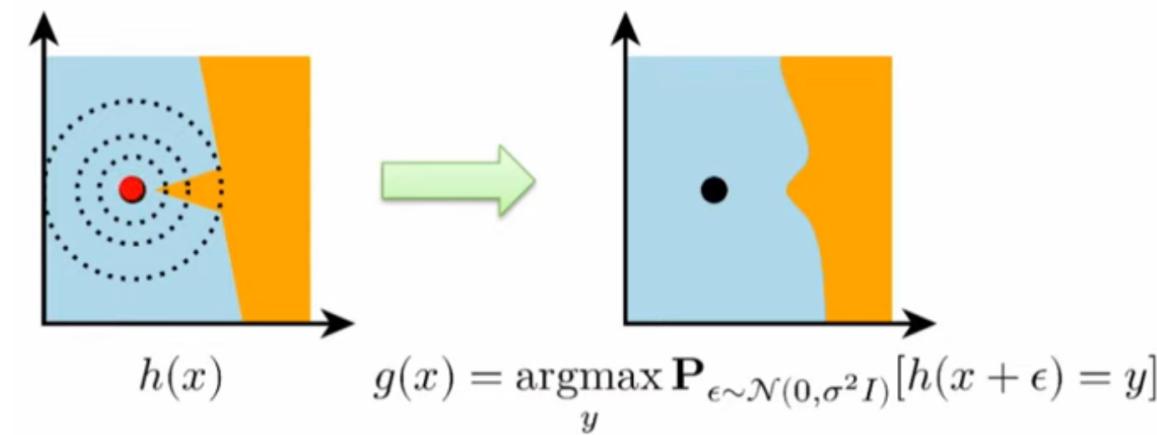


Figure 3. Illustration of classification boundaries resulting from standard training (left) and robust training (right) with ℓ_∞ balls of size $\epsilon = 0.08$ (shown in figure).

Adversarial examples and boundary curvature

- Very curved boundary will lead to adversarial examples looking very similar to the classification boundary
- So let's diminish this curvature spike influence!
 - Different approaches exist e.g. by *Lecuyer et al.*¹ and *Li et al.*², but the most famous one is by *Cohen et al.*



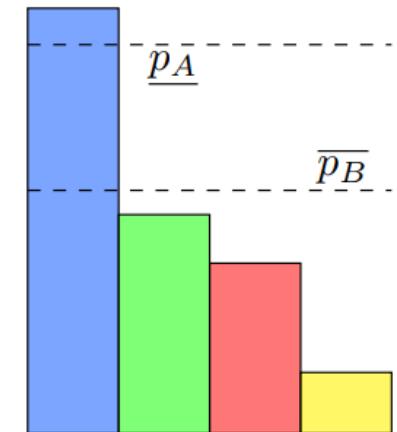
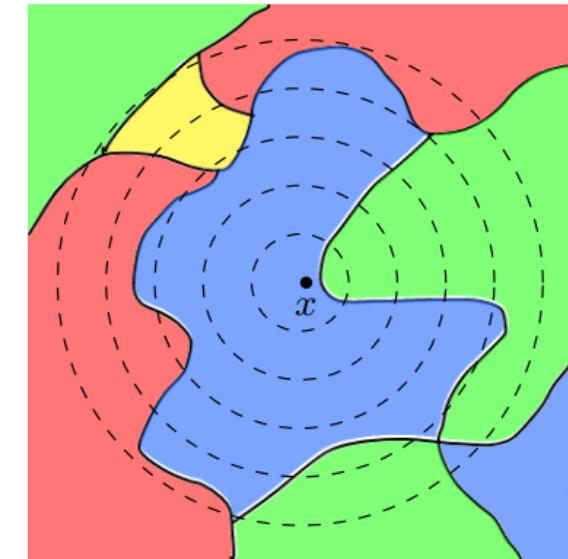
[1] Lecuyer, Mathias, et al. "Certified robustness to adversarial examples with differential privacy."

[2] Li, Bai, et al. "Certified Adversarial Robustness with Additive Noise."

Randomized Smoothing¹

- **Main idea:**
 - Let's use another definition of classifier!
 - New classifier (in fact, sort of TTA):
 - $g(x) = \operatorname{argmax}_{c \in Y} P(f(x + \varepsilon) = c), \varepsilon \sim N(0, \sigma^2 I)$
- **The main robustness result:**
 - If $f(x)$ classifier is robust under Gaussian noise,
 - Then $g(x)$ classifier is robust under **ANY** noise
- The radius R in **Theorem 1** is tight: with the bigger radius there exists an adversarial example

Theorem 2. Assume $\underline{p}_A + \overline{p}_B \leq 1$. For any perturbation δ with $\|\delta\|_2 > R$, there exists a base classifier f consistent with the class probabilities (2) for which $g(x + \delta) \neq c_A$.



Theorem 1. Let $f : \mathbb{R}^d \rightarrow \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let g be defined as in (1). Suppose $c_A \in \mathcal{Y}$ and $\underline{p}_A, \overline{p}_B \in [0, 1]$ satisfy:

$$\mathbb{P}(f(x + \varepsilon) = c_A) \geq \underline{p}_A \geq \overline{p}_B \geq \max_{c \neq c_A} \mathbb{P}(f(x + \varepsilon) = c) \quad (2)$$

Then $g(x + \delta) = c_A$ for all $\|\delta\|_2 < R$, where

$$R = \frac{\sigma}{2} (\Phi^{-1}(\underline{p}_A) - \Phi^{-1}(\overline{p}_B)) \quad (3)$$



Randomized smoothness: some interesting facts

- Some interesting details about linear classifier for two classes: $f(x) = \text{sign}(w^T x + b)$
 - It is the smoothed version of itself:

Proposition 3. *If f is a two-class linear classifier $f(x) = \text{sign}(w^T x + b)$, and g is the smoothed version of f with any σ , then $g(x) = f(x)$ for any x (where f is defined).*

- Certified radius of it is the distance between the point x and the boundary:

Proposition 4. *If f is a two-class linear classifier $f(x) = \text{sign}(w^T x + b)$, and g is the smoothed version of f with any σ , then invoking Theorem 1 at any x (where f is defined) with $\underline{p}_A = p_A$ and $\overline{p}_B = p_B$ will yield the certified radius $R = \frac{|w^T x + b|}{\|w\|}$.*

- But there always exists a classifier with real robustness radius more than R from **Theorem 1**:

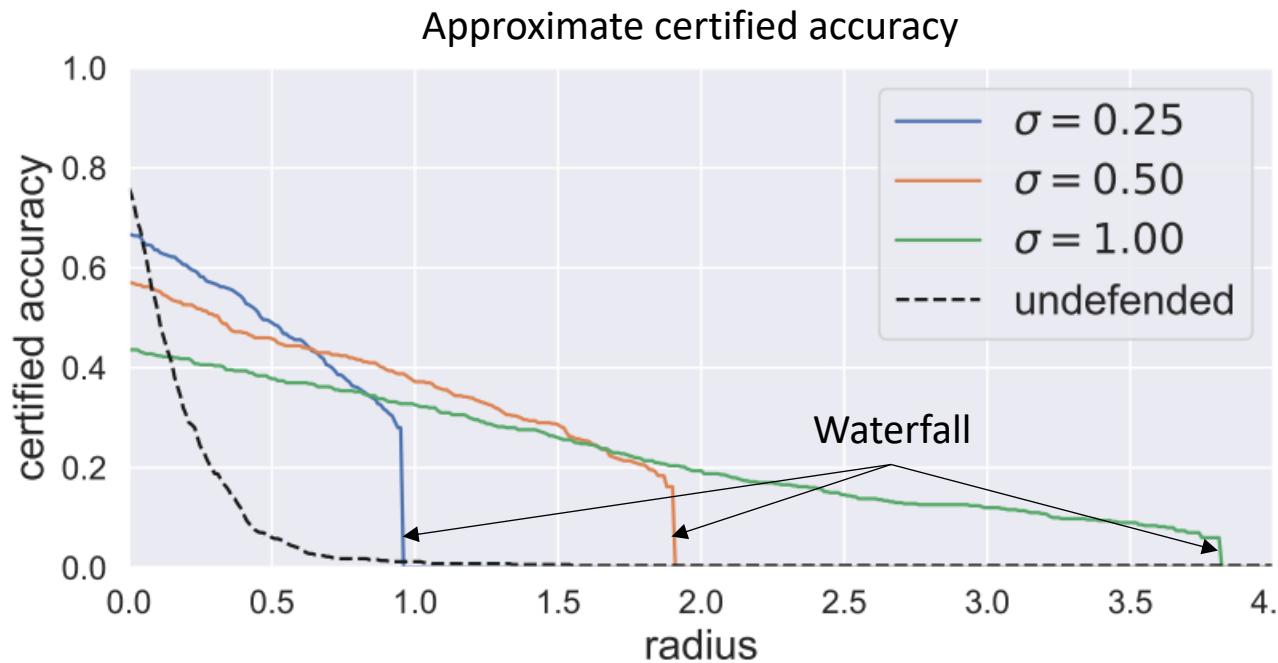
Proposition 6. *For any $\tau > 0$, there exists a base classifier f and an input x_0 for which the corresponding smoothed classifier g is robust around x_0 at radius ∞ , yet Theorem 1 only certifies a radius of τ around x_0 .*



Randomized smoothness: results

- The authors propose the procedure to return the radius R and output class c based on input x and the deviance of noise σ
 - This procedure can even avoid to provide the answer with some probability α
- To certify the classifiers, authors **trained the base models with Gaussian noise from $N(\mathbf{0}, \sigma^2 I)$** – in fact, to make the classifier $f(x)$ more robust to Gaussian noise
- Trained models are compared using “**approximate certified accuracy**”:
 - For each test radius $\delta = r$ the fraction of examples is returned on which CERTIFY
 - Provides the answer
 - Returns the correct class
 - Returns a radius R so as $r \leq R$
 - Also when estimating the $g(x)$ authors run Monte Carlo N times

Randomized smoothness: results on ImageNet (1)



Waterfall just because we the trained model is robust usually under some $r \leq R$

Table 1. Approximate certified accuracy on ImageNet. Each row shows a radius r , the best hyperparameter σ for that radius, the approximate certified accuracy at radius r of the corresponding smoothed classifier, and the standard accuracy of the corresponding smoothed classifier. To give a sense of scale, a perturbation with ℓ_2 radius 1.0 could change one pixel by 255, ten pixels by 80, 100 pixels by 25, or 1000 pixels by 8. Random guessing on ImageNet would attain 0.1% accuracy.

ℓ_2 RADIUS	BEST σ	CERT. ACC (%)	STD. ACC(%)
0.5	0.25	49	67
1.0	0.50	37	57
2.0	0.50	19	57
3.0	1.00	12	44

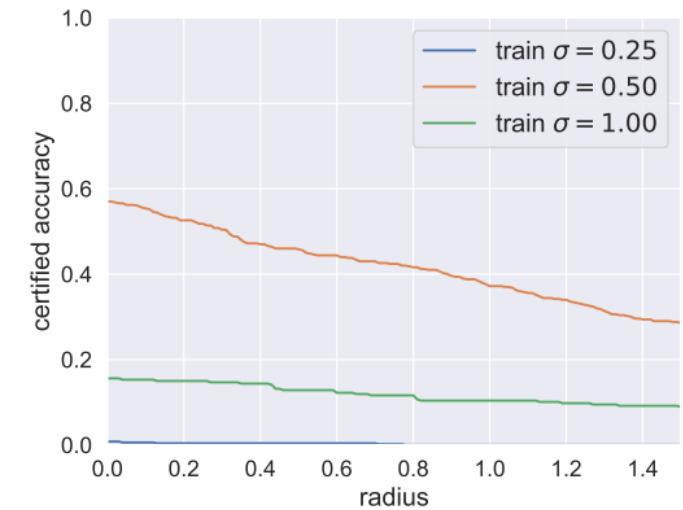
Randomized smoothness: results on ImageNet (2)

- In fact, with NN $f(x)$ we can have larger real robustness radius than R from **Theorem 1**:
 - Authors just tried to find the real adversaries under $r > R$ and measure the success of the attack
 - The lower success the more robust the model
 - $r = 1.5R$: 17% success
 - $r = 2R$: 53% success

Influence of N for Monte Carlo prediction

CORRECT, ACCURATE	
N	
100	0.65
1000	0.68
10000	0.69

Influence of training σ when testing with $\sigma = 0.5$



Improvement: Adversarial training for smoothed classifier¹

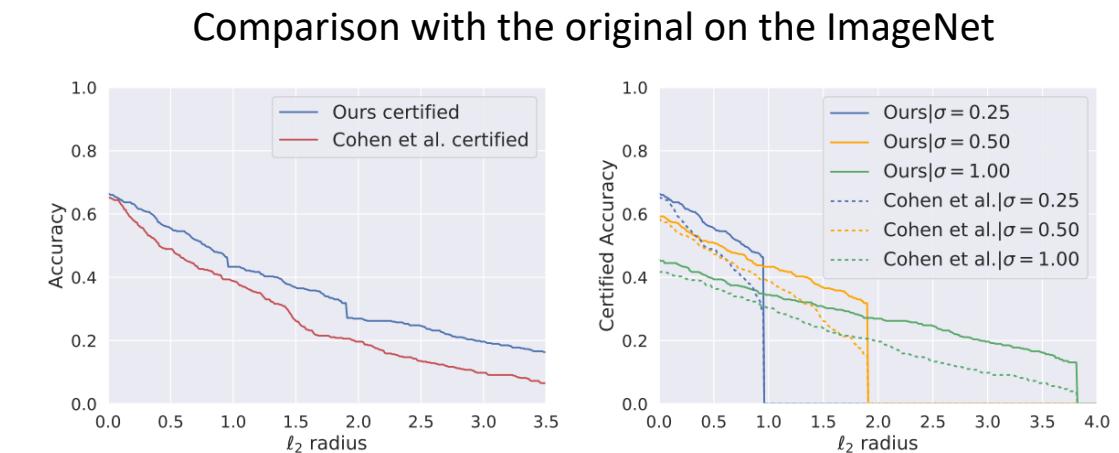
- Instead of simple augmenting the training example with Gaussian noise, let's do in fact adversarial training using attacks on $g(x)$!

$$\hat{x} = \arg \max_{\|x' - x\|_2 \leq \epsilon} \ell_{\text{CE}}(G(x'), y)$$

$$= \arg \max_{\|x' - x\|_2 \leq \epsilon} \left(-\log \mathbb{E}_{\delta \sim \mathcal{N}(0, \sigma^2 I)} [(F(x' + \delta))_y] \right)$$

$$\nabla_{x'} J(x') = \nabla_{x'} \left(-\log \mathbb{E}_{\delta \sim \mathcal{N}(0, \sigma^2 I)} [F(x' + \delta)_y] \right)$$

$$\nabla_{x'} J(x') \approx \nabla_{x'} \left(-\log \left(\frac{1}{m} \sum_{i=1}^m F(x' + \delta_i)_y \right) \right)$$





Improvement: Certified Robustness for Top-k Predictions¹

- The very same idea for Randomized Smoothing, but not only for the top class, but for Top-k classes:
 - $g_k(x) = \operatorname{argmax}_{\substack{c \in Y \\ \epsilon \sim N(0, \sigma^2 I)}} P(f(x + \epsilon) = c),$
- Needed to improve top-5 ImageNet:
 - Certified top-1/top-3/top-5 accuracies = 46.6% / 57.8% / 62.8% when $\|\delta\|_2 = 0.5$

Theorem 1 (Certified Radius for Top- k Predictions). *Suppose we are given an example \mathbf{x} , an arbitrary base classifier f , $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, a smoothed classifier g , an arbitrary label $l \in \{1, 2, \dots, c\}$, and $\underline{p}_l, \bar{p}_1, \dots, \bar{p}_{l-1}, \bar{p}_{l+1}, \dots, \bar{p}_c \in [0, 1]$ that satisfy the following conditions:*

$$\Pr(f(\mathbf{x} + \epsilon) = l) \geq \underline{p}_l \text{ and } \Pr(f(\mathbf{x} + \epsilon) = i) \leq \bar{p}_i, \forall i \neq l, \quad (1)$$

where \underline{p} and \bar{p} indicate lower and upper bounds of p , respectively. Let $\bar{p}_{b_k} \geq \bar{p}_{b_{k-1}} \geq \dots \geq \bar{p}_{b_1}$ be the k largest ones among $\{\bar{p}_1, \dots, \bar{p}_{l-1}, \bar{p}_{l+1}, \dots, \bar{p}_c\}$, where ties are broken uniformly at random. Moreover, we denote by $S_t = \{b_1, b_2, \dots, b_t\}$ the set of t labels with the smallest probability upper bounds in the k largest ones and by $\bar{p}_{S_t} = \sum_{j=1}^t \bar{p}_{b_j}$ the sum of the t probability upper bounds, where $t = 1, 2, \dots, k$. Then, we have:

$$l \in g_k(\mathbf{x} + \delta), \forall \|\delta\|_2 < R_l, \quad (2)$$

where R_l is the unique solution to the following equation:

$$\Phi(\Phi^{-1}(\underline{p}_l) - \frac{R_l}{\sigma}) - \min_{t=1}^k \frac{\Phi(\Phi^{-1}(\bar{p}_{S_t}) + \frac{R_l}{\sigma}))}{t} = 0, \quad (3)$$

where Φ and Φ^{-1} are the cumulative distribution function and its inverse of the standard Gaussian distribution, respectively.

Theorem 2 (Tightness of the Certified Radius). *Assuming we have $\underline{p}_l + \sum_{j=1}^k \bar{p}_{b_j} \leq 1$ and $\underline{p}_l + \sum_{i=1, \dots, l-1, l+1, \dots, c} \bar{p}_i \geq 1$. Then, for any perturbation $\|\delta\|_2 > R_l$, there exists a base classifier f^* consistent with (1) but we have $l \notin g_k(\mathbf{x} + \delta)$.*

BlackBox for Randomized Smoothing¹

- What if we cannot change pretrained classifier, but want to increase its certified robustness?
- Let's train denoiser D used after we add Gaussian noise!
- And then simply apply majority rule
- Denoiser: trained with two losses for every Gaussian σ
 - MSE
 - Stability (CE loss)

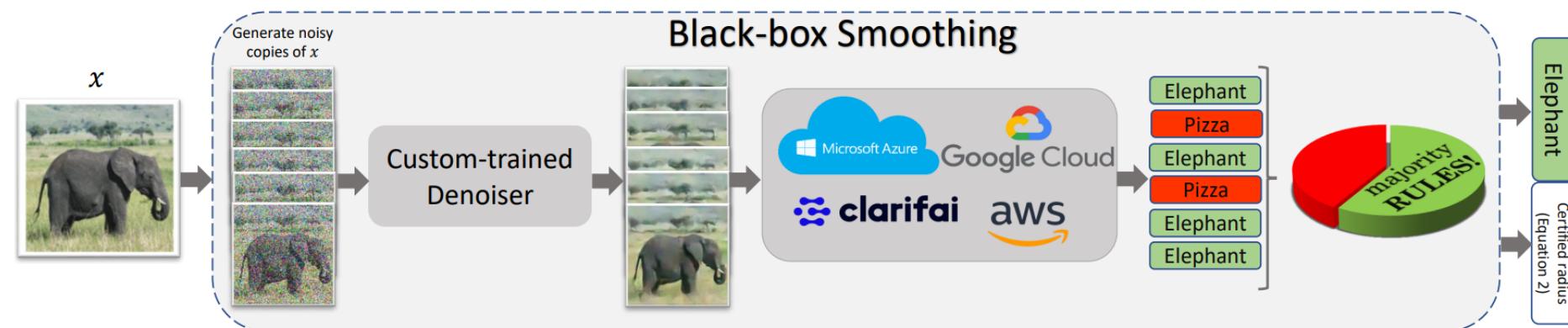
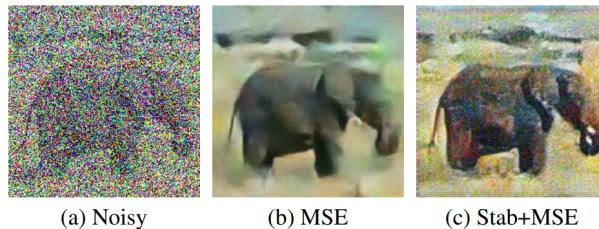
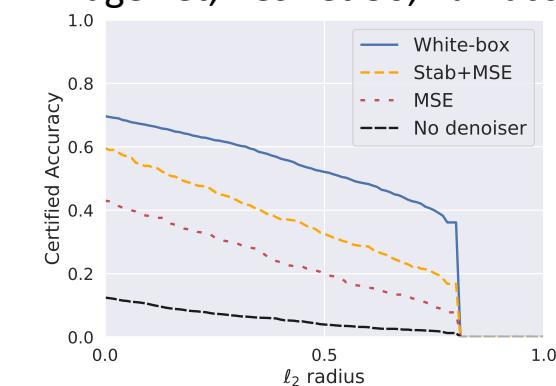


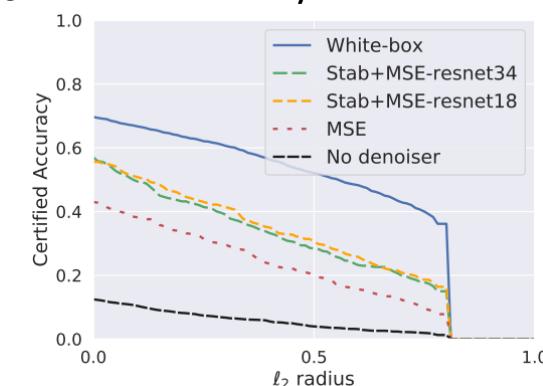
Table 1. Certified top-1 accuracy of ResNet-50 on ImageNet at various ℓ_2 radii (Standard accuracy is in parenthesis).

ℓ_2 RADIUS (IMAGENET)	0.25	0.5	0.75	1.0	1.25	1.5
WHITE-BOX SMOOTHING (COHEN ET AL., 2019) (%)	(70) 62	(70) 52	(62) 45	(62) 39	(62) 34	(50) 29
NO DENOISER (BASELINE) (%)	(49) 32	(12) 4	(12) 2	(0) 0	(0) 0	(0) 0
BLACK-BOX SMOOTHING (QUERY ACCESS) (%)	(69) 48	(56) 31	(56) 19	(34) 12	(34) 7	(30) 4
BLACK-BOX SMOOTHING (FULL ACCESS) (%)	(67) 50	(60) 33	(60) 20	(38) 14	(38) 11	(38) 6

ImageNet, ResNet-50, Full-access

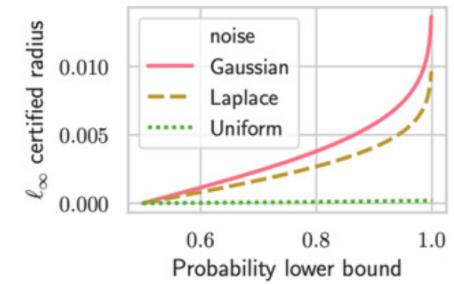
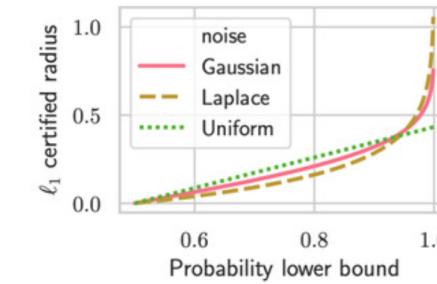


Query-access



Takeaway

- Certification is only for much smaller regions than humans can do
- Certified robustness is better than empirical adversarial training in certification, but worse than clean performance (and too much time to train)
- Using l_p -balls is neither necessary nor sufficient for perceptual robustness
- Other types of randomized smoothing could be taking into account: e.g. *Uniform*¹ or *Laplacian*²
- Randomized smoothing requires multiple inferences ☹
- BTW some note about physical nature of l_p -balls:
 - l_2 : corresponds to the power of signals
 - l_1 : corresponds to the pixel mass
 - l_∞ : corresponds to the noise in camera sensors
 - **l_0 : the most interesting one – corresponds to the practical robustness**





Thank you!

(need to certify everything)