Lecture 6: Robust Training

Trustworthy Machine Learning

Spring 2024

robustness: $\|x - x'\|_\infty \leq \epsilon \Rightarrow$ same label

Agenda

- **Feb 3: Adversarial Examples**
- **Today: Defense: Adversarial training and randomized smoothing**
- **Feb 12: Guest lecture by Alex Robey on robustness for LLMs**
- **Feb 14, 19 (and maybe 21): Formal methods for verified robustness**
- **Homework 1 on adversarial robustness**

Today: Training to ensure robustness

- **Key publications:**
  - **Intriguing properties of neural networks; Szegedy et al, 2014**
  - **Explaining and harnessing adversarial examples; Goodfellow et al, 2015**
  - **Certified adversarial robustness via randomized smoothing; Cohen at al, 2019**

- **Acknowledgement for slides:**
  - **Tutorial: Adversarial robustness: Theory and practice; Kolter and Madry**
  - **Lectures on Robustness in machine learning; Hongyang Zhang (Waterloo)**
  - **Notes by Eric Wong for “Debugging Data and Models”**

Supervised Learning

- Given a model f parameterized by θ
- Loss(x, y; θ) denotes the error of f_θ on input x with respect to desired output y
- Learning as optimization:

  Given a training set S of labeled input/output pairs (x, y),

  find θ to minimize the average training loss

Adversarial Example Computation

- Given a (trained) model f with parameters θ
- Fix input x and corresponding output $y = f_\theta(x)$
- $Loss(x+\delta, y; \theta)$ denotes the “change” in output with respect to δ–perturbation in input
- Search for adversarial example:

Given a bound Δ on input perturbation,

find $0 < \delta < \Delta$ to maximize $Loss(x+\delta, y; \theta)$

Adversarial Training

- Given a model f parameterized by θ
- Loss(x, y; θ) denotes the error of f_θ on input x with respect to desired output y
- Given training set S of labeled input/output examples (x,y)
- Goal: Account for adversarial examples during learning (update of parameters θ)
- Adversarial training as optimization:

$$\min_{\theta} \sum_{x,y \in S} \max_{\delta \in \Delta} \mathrm{Loss}\left(x + \delta, y; \theta\right)$$

MinMax Optimization

$$\min_{\theta} \sum_{x,y \in S} \max_{\delta \in \Delta} \text{Loss}\left(x+\delta, y; \theta\right)$$

How to obtain optimal θ by modifying gradient descent?

Danskin's Theorem for solving MinMax problems

A fundamental result in optimization:

$$\nabla_\theta \max_{\delta \in \Delta} \text{Loss}(x + \delta, y; \theta) = \nabla_\theta \text{Loss}(x + \delta^\star, y; \theta)$$

where $\delta^\star = \max_{\delta \in \Delta} \text{Loss}(x + \delta, y; \theta)$

Caveat: Result assumes that we are computing δ^* exactly, but we are not …

Adversarial Training Algorithm

Repeat:

1. Select a minibatch B
2. For each (x,y) in B, compute the adversarial example $\delta^*(x)$

   Recall FGSM method of steepest descent to compute adversarial examples

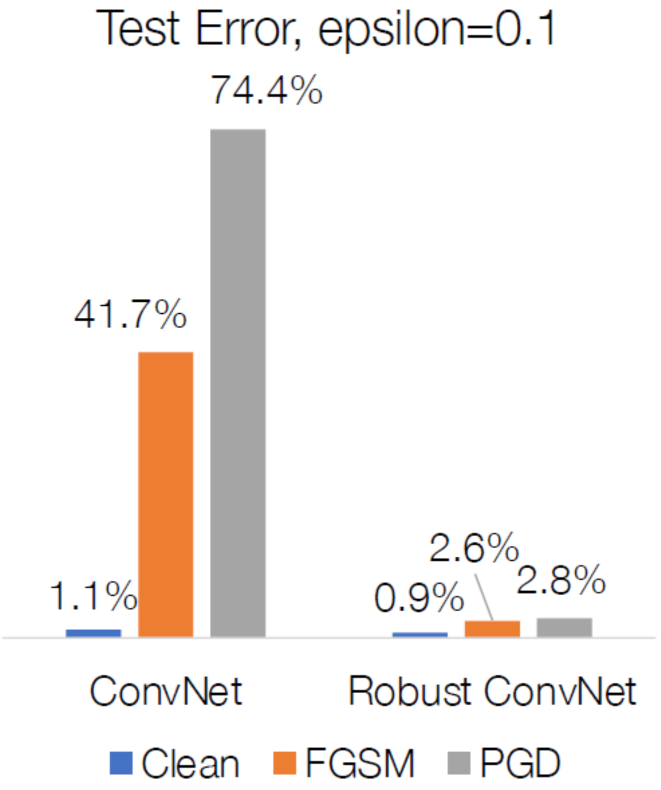
   $$\delta = \epsilon \cdot \operatorname{sign}\big(\nabla_\delta \text{Loss}(x+\delta, y; \theta)\big)$$

3. Update parameters

$$\theta := \theta - \frac{\alpha}{|B|} \sum_{x,y \in B} \nabla_\theta \text{Loss}(x + \delta^\star(x), y; \theta)$$

Note: in practice, one can mix standard updates and adversarial updates

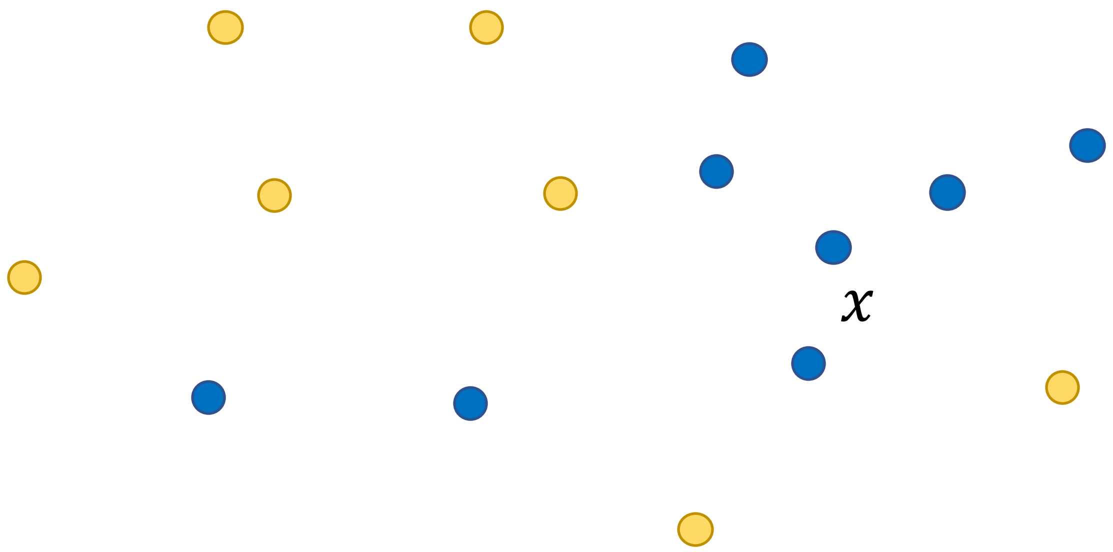
Empirical Evaluation of Robust Training



Beyond Empirical Defenses

- Adversarial training improves robustness empirically
- But adversarial example is only one type of attack, new attacks need new defenses
- Certified robustness: Can we get mathematical guarantees of robustness ?
- Certified Robustness via randomized smoothing [Cohen et al; 2019]

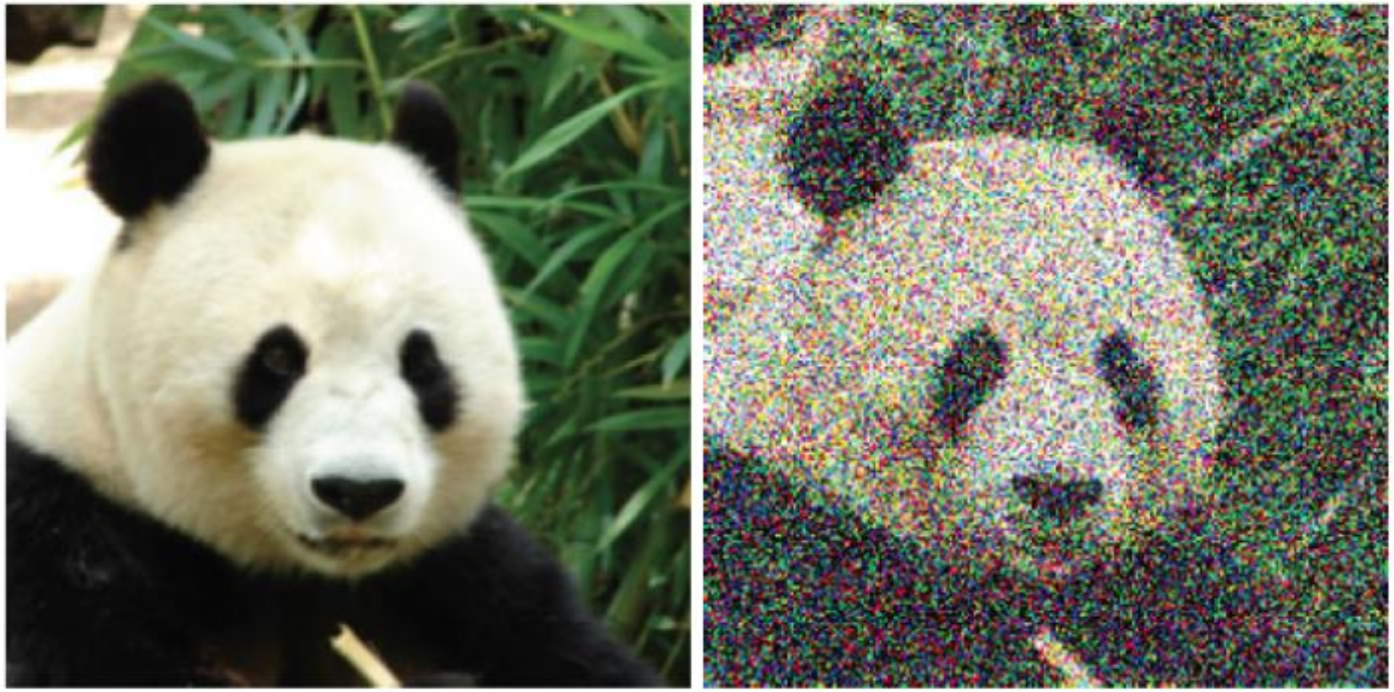
Smoothing of a given classifier, informally



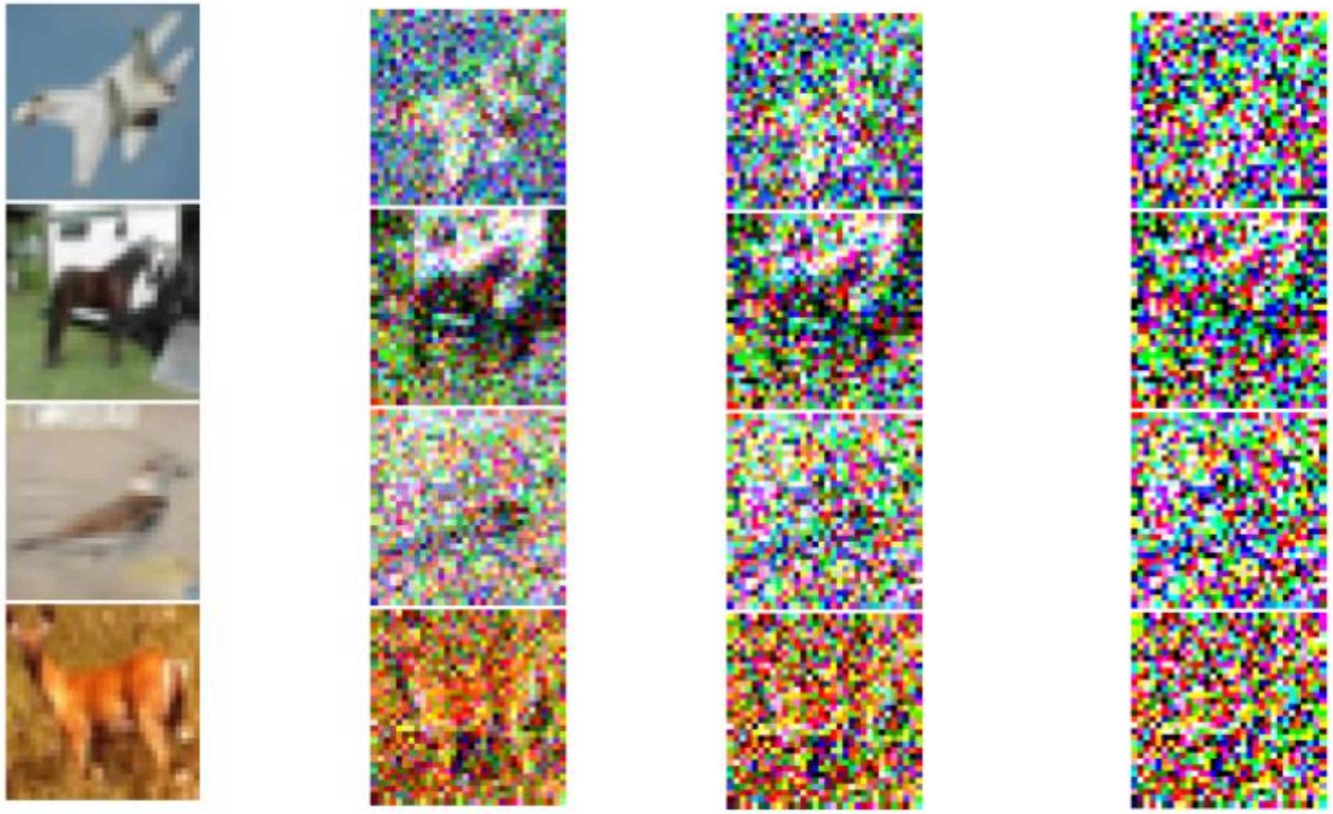
- Sample multiple perturbations x’ of x
- Compute the label f(x’) for each variant
- Set g(x) to the majority vote

Creating Random Perturbations

- Given an input x, consider inputs x+ η, where η is noise sampled from Gaussian distribution with mean 0 and variance σ^2, that is, $\eta \sim \mathcal{N}(0, \sigma^2 I)$

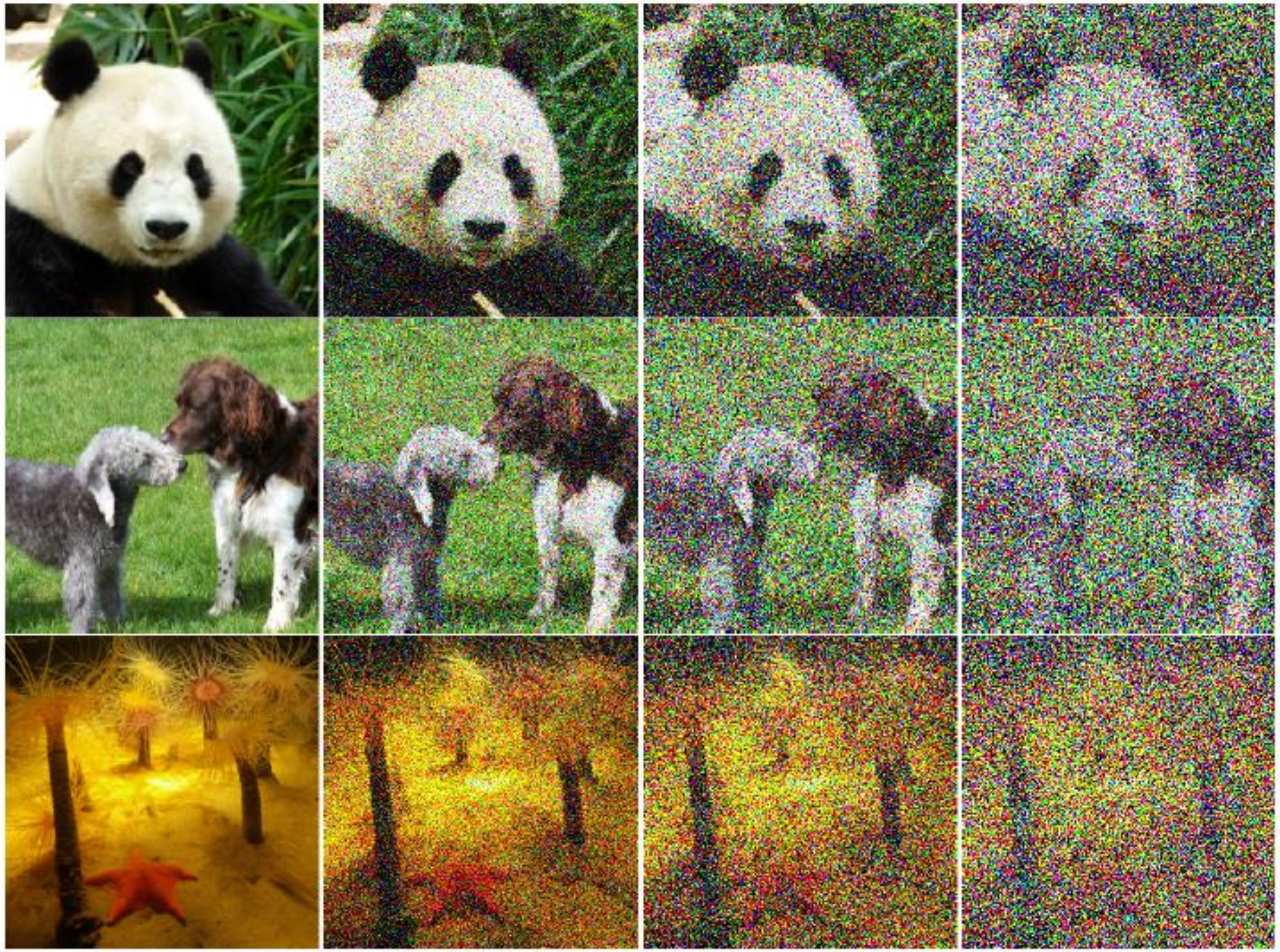


Examples of noisy images from CIFAR-10 with varying levels of Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ from $\sigma = 0$ to $\sigma = 1$



$\sigma = 0.00$ $\sigma = 0.25$ $\sigma = 0.50$ $\sigma = 1.00$

Examples of noisy images from ImageNet with varying levels of Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ from $\sigma = 0$ to $\sigma = 1$



$\sigma = 0.00$ $\sigma = 0.25$ $\sigma = 0.50$ $\sigma = 1.00$

Smoothed classifier

- Given a base classifier f, its smoothed version g maps an input x to the majority prediction of f on many Gaussian-perturbed images x+ η

$$g(x) = \underset{y}{\operatorname{argmax}}\, \mathbb{P}_{\eta}\, [f(x + \eta) = y]$$

Estimation by Monte Carlo Sampling

To design a smoothed classifier *g* at the input sample *x* requires to identify the most likely class $\hat{c}_A$ returned by the base classifier *f* on noisy images

- Step 1: create *n* versions of *x* corrupted with Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$
- Step 2: evaluate the predictions by base classifier for all corrupted images, $f(x+\eta)$
- Step 3: identify the top two classes $\hat{c}_A$ and $\hat{c}_B$ with the highest number of predictions on $f(x+\eta)$
- Step 4: if n_A (number of predictions by *f* for the top class $\hat{c}_A$) is much greater than n_B (number of predictions for the second highest class $\hat{c}_B$), return $\hat{c}_A$ as the prediction by $g(x)$
  - Otherwise, if $n_A - n_B < \alpha$, abstain from making a prediction

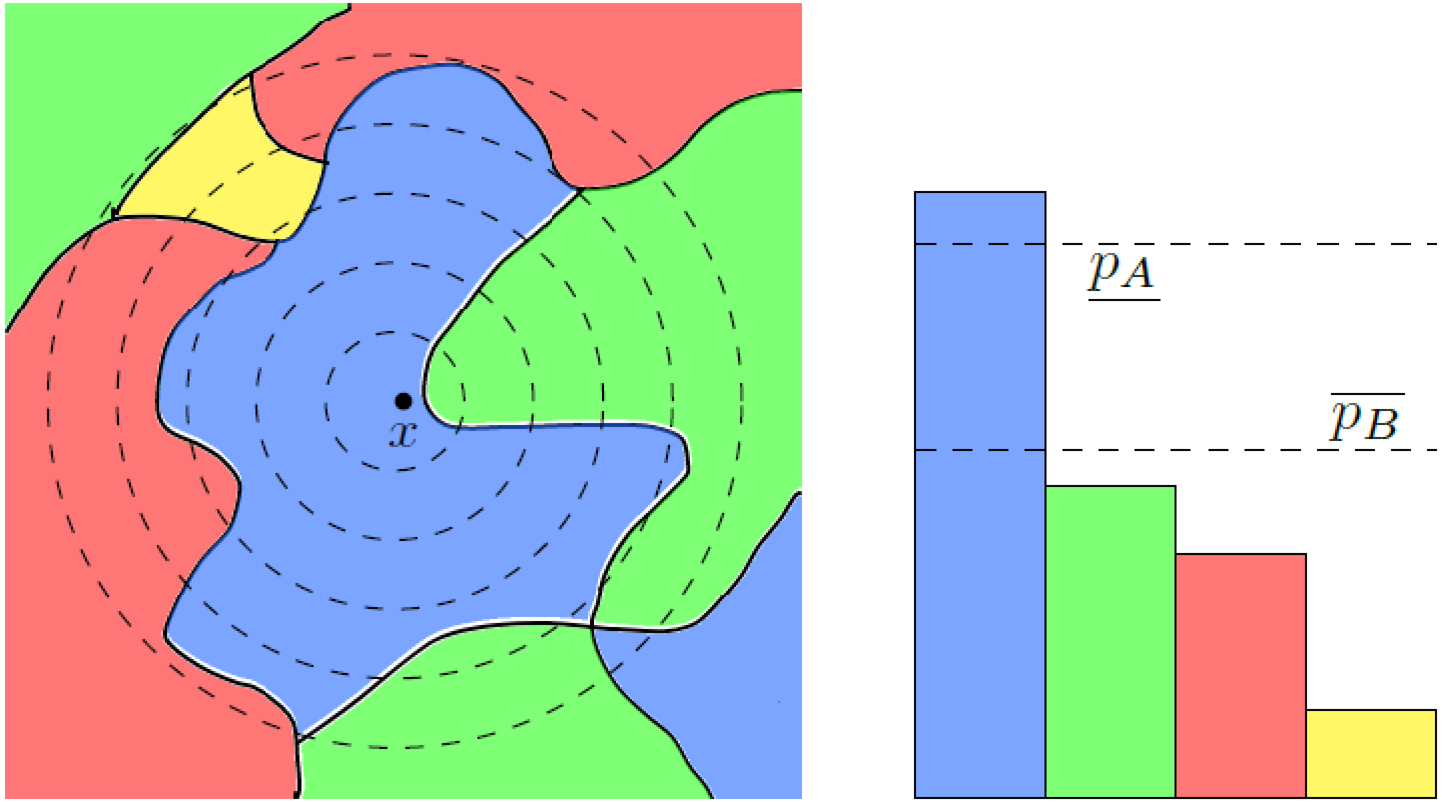
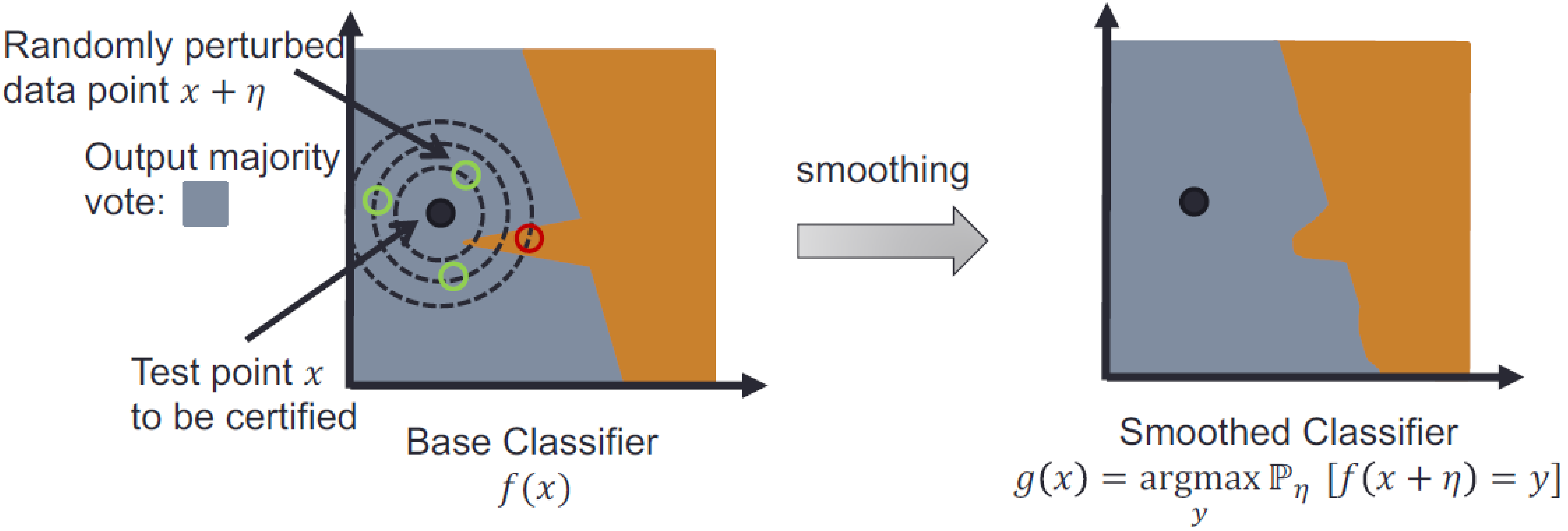


Figure 1. Evaluating the smoothed classifier at an input x. **Left**: the decision regions of the base classifier f are drawn in different colors. The dotted lines are the level sets of the distribution $\mathcal{N}(x, \sigma^2 I)$. **Right**: the distribution $f(\mathcal{N}(x, \sigma^2 I))$. As discussed below, $\underline{p_A}$ is a lower bound on the probability of the top class and $\overline{p_B}$ is an upper bound on the probability of each other class. Here, $g(x)$ is "blue."

Illustrating effect of smoothing



Randomized Smoothing

- Method works for an arbitrary f, including complex neural networks
- The smoothed version g of a given classifier f turns out to be empirically robust
- The bound Δ on adversarial robustness radius is related to the parameter σ in Gaussian noise
- Intuitively: large random noise can be used to drown out small adversarial perturbation
- Key question: can one establish this relationship provably?

Randomized Smoothing Guarantee

Certified robust radius by [Cohen et al.'19]:

Confidence of majority vote

Given any input $x \in \mathbb{R}^d$, let η be Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ and $p = \max_y \mathbb{P}_\eta[f(x+\eta) = y]$. Then $g(x) = g(x+\delta)$ for any δ such that $\|\delta\|_2 \leq \Phi^{-1}(p)\sigma$, where Φ is CDF of standard Gaussian.

Computable certified radius for x

Proof sketch (binary classifier)

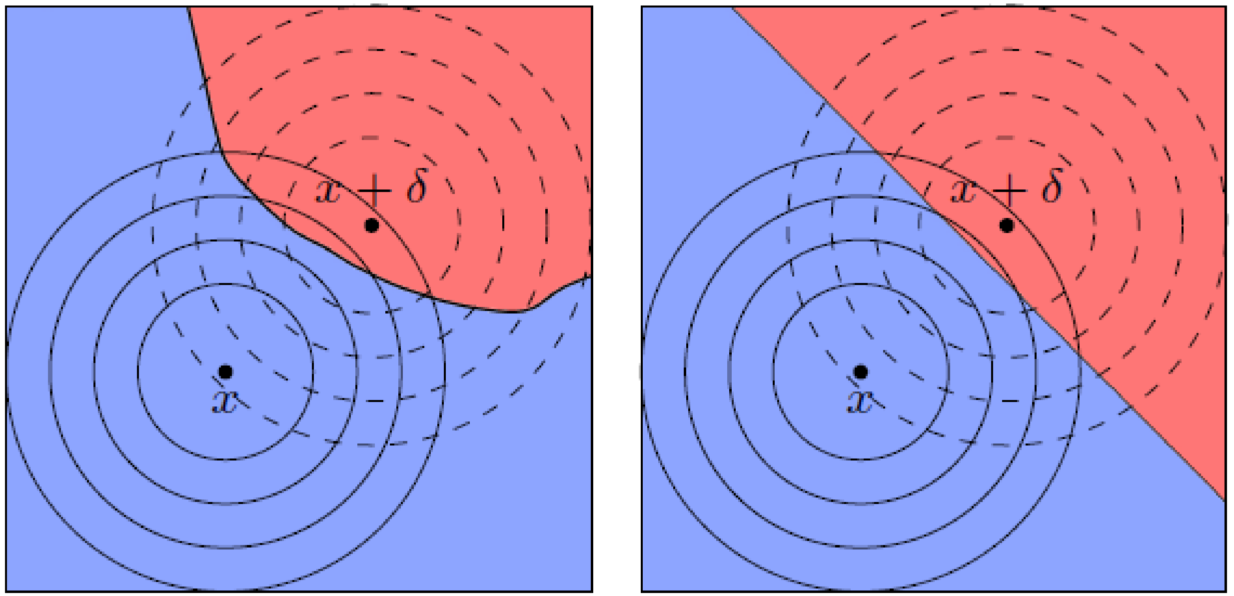
1. Suppose the top class has probability p_A, so f classifies $\mathcal{N}(x, \sigma^2 I)$ as A with probability $\geq p_A$.

   .

2. Consider a fixed perturbation δ. We want the probability that f classifies s $\mathcal{N}(x+\delta, \sigma^2 I)$ as A. If this probability is greater than 1/2 then $g(x+\delta) = A$.

3. We want a statement for all possible f, so consider the worst case f which classifies s $\mathcal{N}(x, \sigma^2 I)$ with probability $\geq p_A$, but minimizes the probability that $\mathcal{N}(x+\delta, \sigma^2 I)$ is A.

Illustrating worst-case classifier in the proof



Among all classifiers f for which g(x) is blue with probability greater than a given threshold, and g(x+δ) is blue with minimal probability, the "worst-case" is linear classifier normal to direction of δ from x

Proof sketch

1. Suppose the top class has probability p_A, so f classifies $\mathcal{N}(x, \sigma^2 I)$ as A with probability $\geq p_A$.

.

2. Consider a fixed perturbation δ. We want the probability that f classifies s $\mathcal{N}(x+\delta, \sigma^2 I)$ as A. If this probability is greater than 1/2 then $g(x+\delta) = A$.

3. We want a statement for all possible f, so consider the worst case f which classifies s $\mathcal{N}(x, \sigma^2 I)$ with probability $\geq p_A$, but minimizes the probability that $\mathcal{N}(x+\delta, \sigma^2 I)$ is A.

4. By a similar argument to the Neyman Pearson lemma, this worst-case classifier is the linear classifier $f(x') = \begin{cases} A \text{ if } \delta^T(x'-x) \leq \sigma\|\delta\|_2 \Phi^{-1}(p_A) \\ B \text{ otherwise} \end{cases}$

5. For this worst case classifier, f classifies $\mathcal{N}(x+\delta, \sigma^2 I)$ as A with probability $\Phi\left(\Phi^{-1}(p_A) - \frac{\|\delta\|_2}{\sigma}\right)$. Solving this for 1/2 we get the condition $\|\delta\|_2 < \sigma\Phi^{-1}(p_A)$.

Randomized Smoothing Guarantee

Certified robust radius by [Cohen et al.'19]:

Confidence of majority vote

Given any input $x \in \mathbb{R}^d$, let η be Gaussian noise $\mathcal{N}(0, \sigma^2 I)$ and $p = \max_y \mathbb{P}_\eta[f(x+\eta) = y]$. Then $g(x) = g(x+\delta)$ for any δ such that $\|\delta\|_2 \leq \Phi^{-1}(p)\sigma$, where Φ is CDF of standard Gaussian.

Computable certified radius for x

If we can estimate that the probability g(x)=A is at least p_1 and the probability that g(x)=B is at most p_2, where A is the most likely class and B is the "runner-up" class, then above bound holds with $\Phi^{-1}(p)$ replaced by $(\Phi^{-1}(p_1) - \Phi^{-1}(p_2))/2$

Implementing Certified Robustness

 # certify the robustness of g around x

function CERTIFY(f, σ, x, n_0, n, α)

  counts0 ← SAMPLEUNDERNOISE(f, x, n_0, σ)

  $\hat{c}_A$ ← top index in counts0

  counts ← SAMPLEUNDERNOISE(f, x, n, σ)

  $\underline{p_A}$ ← LOWERCONFBOUND(counts[$\hat{c}_A$], n, $1 - \alpha$)

  if $\underline{p_A} > \frac{1}{2}$ **return** prediction $\hat{c}_A$ and radius $\sigma\, \Phi^{-1}(\underline{p_A})$

  else return ABSTAIN

SampleUnderNoise(f,x,n, σ) samples n values of noise from the distribution $\eta \sim \mathcal{N}(0, \sigma^2 I)$

evaluates f (x + η), and returns a vector of class counts

Implementing Certified Robustness

 # certify the robustness of g around x

function CERTIFY(f, σ, x, n_0, n, α)

`counts0` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n_0, σ)

$\hat{c}_A \leftarrow$ top index in `counts0`

`counts` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n, σ)

$\underline{p_A} \leftarrow$ LOWERCONFBOUND(`counts`[$\hat{c}_A$], n, $1 - \alpha$)

if $\underline{p_A} > \frac{1}{2}$ **return** prediction $\hat{c}_A$ and radius $\sigma\, \Phi^{-1}(\underline{p_A})$

else return ABSTAIN

LowerConfBound(k, n, $1-\alpha$) returns one-sided (1-α) lower interval for the Binomial parameter p given the sample k ~ Binomial(n,p)

Implementing Certified Robustness

```
# certify the robustness of g around x
function CERTIFY(f, σ, x, n_0, n, α)
    counts0 ← SAMPLEUNDERNOISE(f, x, n_0, σ)
    ĉ_A ← top index in counts0
    counts ← SAMPLEUNDERNOISE(f, x, n, σ)
    p_A ← LOWERCONFBOUND(counts[ĉ_A], n, 1 − α)
    if p_A > 1/2 return prediction ĉ_A and radius σ Φ^{-1}(p_A)
    else return ABSTAIN
```

Proposition 2. *With probability at least $1 - \alpha$ over the randomness in* CERTIFY, *if* CERTIFY *returns a class $\hat{c}_A$ and a radius R (i.e. does not abstain), then g predicts $\hat{c}_A$ within radius R around x: $g(x + \delta) = \hat{c}_A \;\; \forall \; \|\delta\|_2 < R$.*

Certified Robustness via Randomized Smoothing

- First method to give mathematical guarantees of robustness

- Robustness radius R depends on noise parameter σ and separation between top two classes in prediction of x

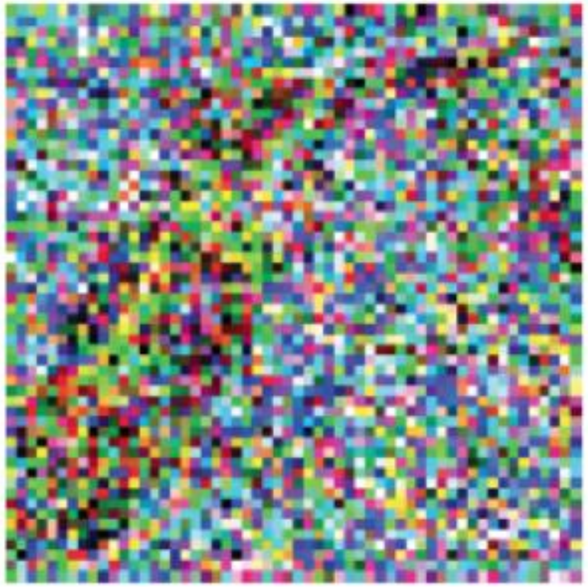
- There is accuracy – robustness trade-off

- Follow-up work studies theoretical limits of robustness guarantees

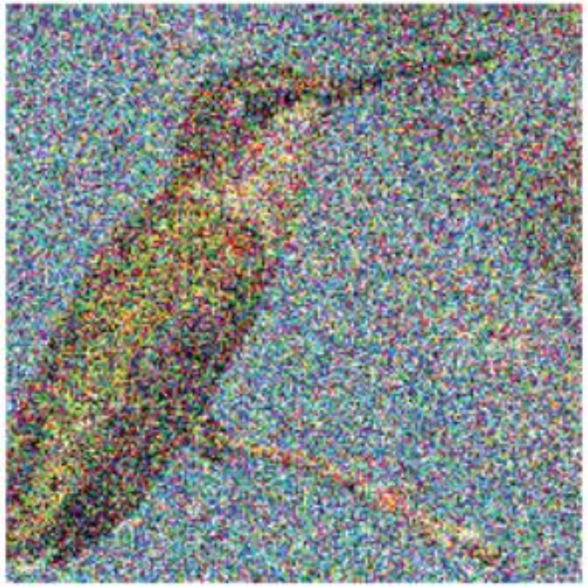
Noise vs Resolution

Clean 56×56 image

Clean 224×224 image



Noisy 56×56 image
($\sigma = 0.5$)

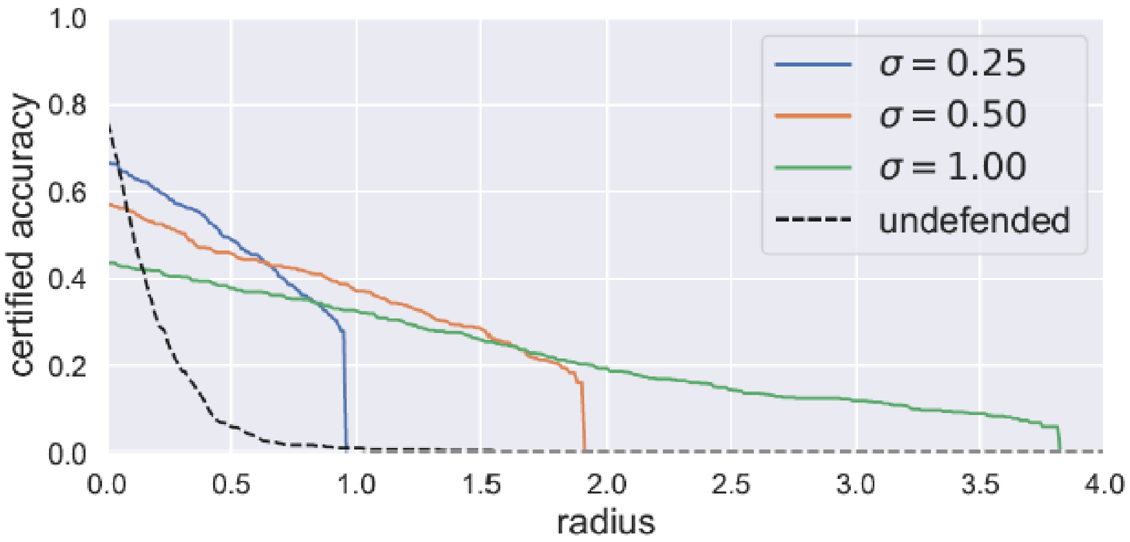


Noisy 224 ×224 image
($\sigma = 0.5$)

Certified Robustness: Empirical Evaluation

Plot of the certified top-1 accuracy by ResNet50 on ImageNet by the randomized smoothing

- As the radius R increases, the certified accuracy decreases
- The noise level σ controls the tradeoff between accuracy and robustness
  - When σ is small (e.g., $\sigma = 0.25$), perturbations with small radius R (e.g. $R = 0.5$) can be certified with high accuracy
  - However, for small σ (e.g., $\sigma = 0.25$), perturbations with $R > 1.0$ cannot be certified
  - Increasing σ (e.g., $\sigma = 1.0$) will enable robustness to larger perturbations ($R > 3.0$ and higher), but will result in decreased certified accuracy



Agenda

- **Feb 3: Adversarial Examples**
- **Today: Defense: Adversarial training and randomized smoothing**
- **Feb 12: Guest lecture by Alex Robey on robustness for LLMs**
- **Feb 14, 19 (and maybe 21): Formal methods for verified robustness**
- **Homework 1 on adversarial robustness**