CY 7790

Special Topics in Security and Privacy: Machine Learning Security and Privacy Fall 2021

Alina Oprea
Associate Professor
Khoury College of Computer Science

October 7 2021

Class Project

- Project
  - Any topic in security or privacy of ML
  - Two types of projects:
    - SoK: Systematization of Knowledge; literature review with some implementation component (comparing different attacks or defenses, etc.)
    - Research project: Evaluate a security attack or defense, a privacy attack or defense, or an ML fairness issue
- Project proposal
  - Prepare short presentations in class (10 minutes) to get feedbackon Oct. 21
  - One-page proposal submitted to Gradescope on Oct. 20
  - Team of two members, can also have individual contributors

Project Proposal

- Title
- Team members
- Problem statement
- Approach / Methodology
- Evaluation (datasets, models, metrics)
- Project milestone
  - About 3 weeks after proposal

Project Topics

- Evasion attacks
  - Attacks for specific applications and datasets (cyber security, healthcare, etc.)
  - What objectives and distance metrics can be used (use domain expertise to guide the attacks and defenses)
  - Efficient black-box attacks
  - Defenses (system-level or ML-level); application-specific defenses
  - Multiple data modalities: images with text
  - Explainability methods to understand attack or defense
- Poisoning attacks
  - Data or model poisoning for specific settings
  - Clean-label attacks vs get access to labels
  - Poisoning to facilitate evasion attacks
  - Code-poisoning attacks: what if a component of the ML code is corrupted?
  - Defenses: data sanitization (outlier detection), system-level defenses
  - Evaluate transferability across models / datasets
  - Poisoning anomaly detection / unsupervised learning

Project Topics

- ML security analysis in realistic conditions
  - What if certain components fail (e.g., random number generator)
  - If model is pruned for practical reasons (to fit on a mobile device), how is robustness impacted
  - Side channels in prediction
- Privacy attacks and defenses
  - Membership inference, property inference, model extraction attack, memorization s in different settings
  - What can an adversary learn about training data and model?
  - Decision-based versus confidence-based attacks
  - Defenses: differential privacy (DP) or system-level; compare different DP methods
  - New models of privacy (label-DP)
  - Connection between privacy and robustness
- Fairness in ML
  - Subgroup analysis – which groups are incorrectly classified?
  - How to add constraints to classifiers to ensure certain fairness conditions (e.g., equal false-positive rates)
  - Are fairness, robustness, and privacy at odds?

Adversarial Examples are Not Bugs, They are Features

Harsh Chaudhari

Threat Model

- Evasion Attack: Attacker is finding a perturbation, that leads to misclassification of the model.
- Can be untargeted or targeted depending on the setting.

Are Models *“always”* at Fault?

Roadmap

- (The Takeaway) High-Level Results and Observations
- (Not so fun) Formal Definitions
- More Details

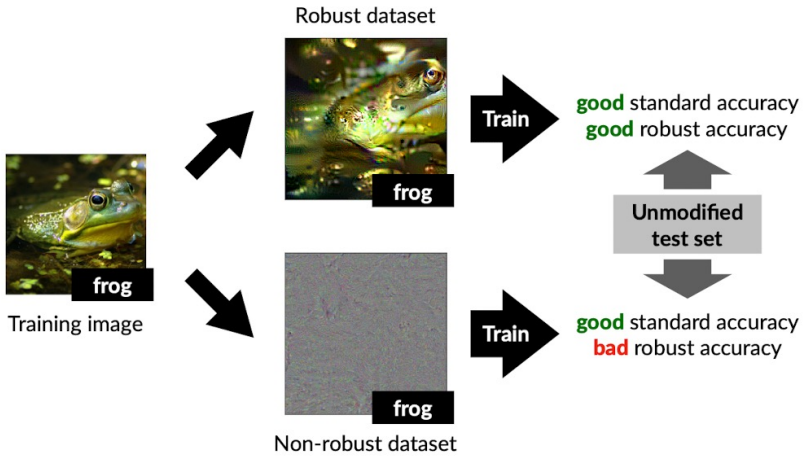
High Level Results and Observations

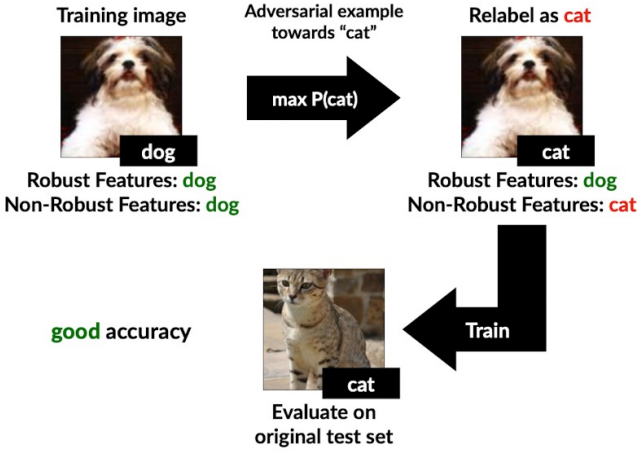
- Visualization of a dataset in terms of Robust and Non-Robust features
- Observations on the behavior of these two sets of features
- Applying this concept to explaining transferability of attacks

Methodology

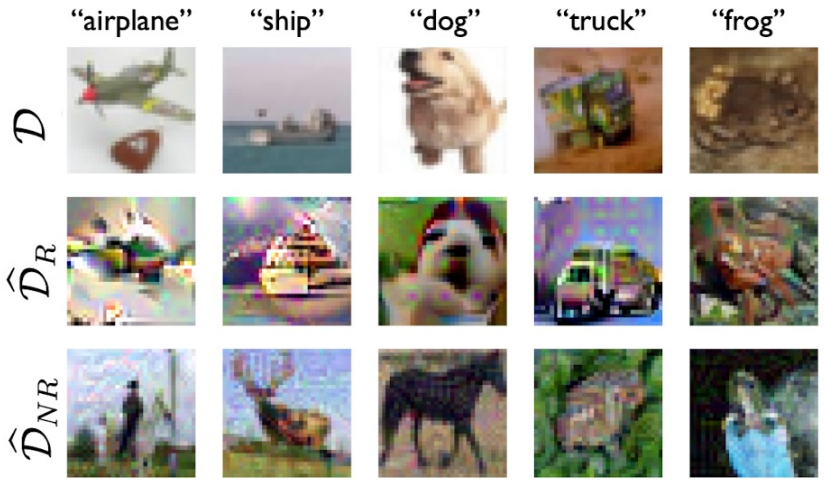
- Definitions of robust and useful features
  - Useful features are correlated to label
  - Robust features remain correlated to label under perturbation
- Construct a dataset with robust features
  - Show that standard training achieves robustness
  - Adversarial vulnerability is a property of the dataset and features
  - Non-robust features create adversarial examples
- Create a dataset with non-robust features
  - To humans, this looks wrongly labeled
  - Training on this yields good accuracy on original test set
  - Non-robust features are useful for generalization

Robust v/s Non Robust Features





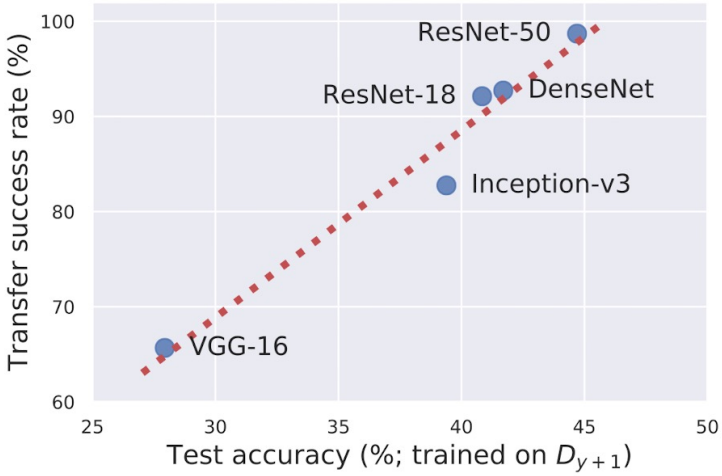
Further Evidence



- Standard training with robust features provides some resilience to attacks
- Standard training with non-robust features does not provides resilience to attacks

Transfer Rate of Adversarial Examples

- Adversarial Transferability arises from utilizing similar non-robust features



Formal Definitions

- Feature f: A function mapping from the input space X to the real numbers, with the set of all features thus being F = { f : X → R}
- ρ-useful features:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[y \cdot f(x)] \geq \rho.$$

More Definitions

- γ-robustly useful features:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\inf_{\delta\in\Delta(x)} y\cdot f(x+\delta)\right]\geq\gamma.$$

- Useful, Non-Robust features: A feature which is ρ-useful for some ρ bounded away from zero, but is not a γ-robust feature for any γ ≥ 0.

Training Methods

- Standard Training:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}_\theta(x,y)\right] = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[y\cdot\left(b+\sum_{f\in F} w_f\cdot f(x)\right)\right].$$

- Robust Training:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\Delta(x)}\mathcal{L}_\theta(x+\delta,y)\right],$$

Search for Robust and Non-Robust Features

- Recipe for Robust Dataset Generation:

$$\mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}_R}[f(x)\cdot y] = \begin{cases} \mathbb{E}_{(x,y)\sim\mathcal{D}}[f(x)\cdot y] & \text{if } f \in F_C \\ 0 & \text{otherwise,} \end{cases} \qquad \min_{x_r} \|g(x_r) - g(x)\|_2,$$

- Recipe for Non-Robust Generation: Repeat the same process but on a non-robust standard model.

Importance Non-Robust Features

- Behavior of model on Non-Robust Dataset:
  - D_{rand} : labels are assigned uniformly random
  - D_{det} : labels assigned deterministically based on the source class

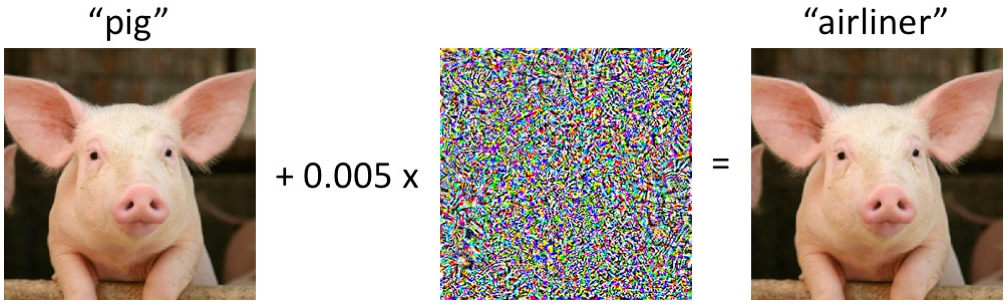
Source Dataset	Dataset	
	CIFAR-10	ImageNet$_R$
$\mathcal{D}$	95.3%	96.6%
$\widehat{\mathcal{D}}_{rand}$	63.3%	87.9%
$\widehat{\mathcal{D}}_{det}$	43.7%	64.4%

Certified Adversarial Robustness via Randomized Noise

by Jeremy Cohen, Elan Rosenfeld, Zico Kolter

John Abascal

Threat Model



- Attacker is attempting to force an incorrect prediction by the model

  –This is achieved by finding a perturbation, δ, such that:

$$\max_{\delta} \text{loss}(x + \delta, \theta)$$

Problem Statement

- Defenses against adversarial attacks are a cat-and-mouse game
  - Whenever a new defense is made, another paper is published within a few months that breaks it

Are there provable robustness guarantees for defenses against adversarial attacks?

Problem Statement

- **Are there provable robustness guarantees for defenses against adversarial attacks?**
  - Yes! The authors of this paper show that using ***randomized smoothing***, as a defense technique nets us guarantees for *any* classifier's ***robustness***

What is Robustness?

- A classifier is said to be ***certifiably robust*** if for any input x, one can obtain a guarantee that the classifier's prediction is constant within some set around x [Cohen et al.]
    - This set is often the ℓ_2 or ℓ_∞ ball around x

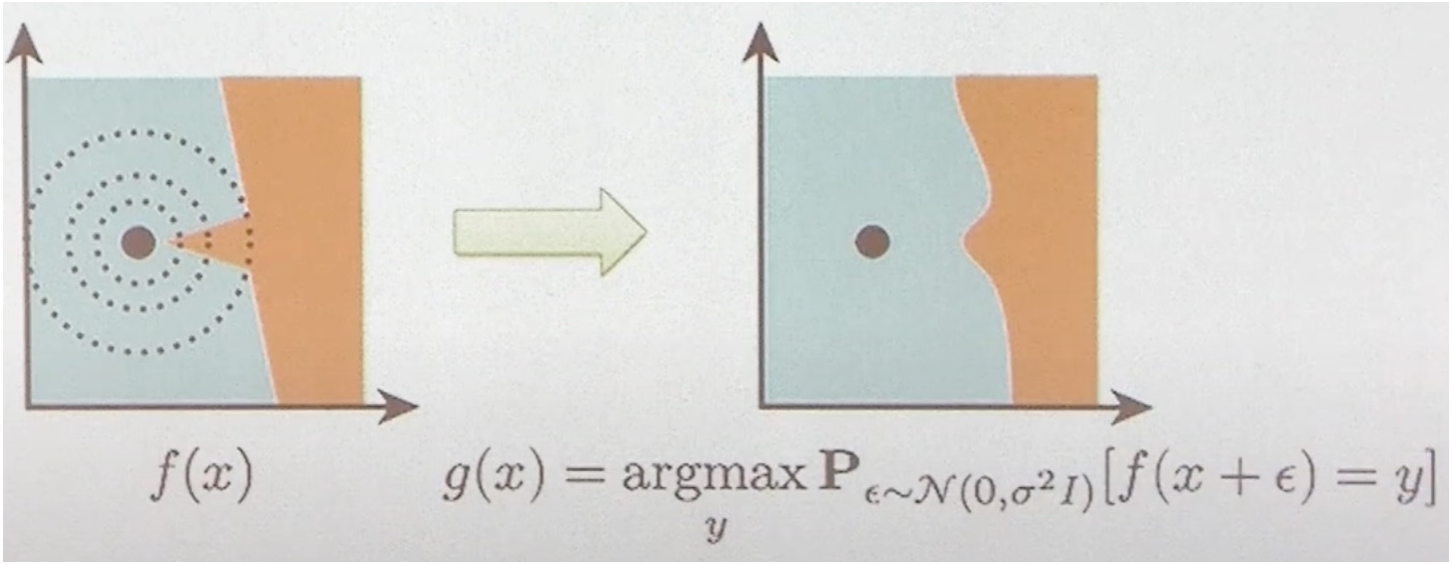
What is Randomized Smoothing?

- Consider a classification problem from $\mathbb{R}^d$ to $\mathcal{Y}$, where we have found some arbitrary base classifier f
  - We define the ***"smoothed" classifier*** (at test time) as

$$g(x) = \underset{c \in \mathcal{Y}}{argmax} \mathbb{P}[f(x + \epsilon) = c]$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$

What is Randomized Smoothing?



[Kolter]

What is Randomized Smoothing?

- The randomized smoothing procedure (for a neural network) would be defined as the following

1. $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
2. Find y such that $\mathbb{P}[f(x + \epsilon) = y]$ is maximized
3. Store y
4. **Repeat**
5. Take majority y as the label for $f(x)$

What is Randomized Smoothing?

- Observation
  - Randomized smoothing requires the base classifier to be able to correctly classify noisy images

Theorem 1 (Binary Case)

- Given an input, x, let $\hat{y} = g(x)$ be the prediction of the smooth classifier, and let $p > \frac{1}{2}$ be the associated probability under the smoothing distribution

$$p = \mathbb{P}_{\epsilon \sim \mathcal{N}(0,\sigma^2 I)}[f(x+\epsilon) = y]$$

- Then, $g(x+\delta) = \hat{y} \quad \forall \delta$ such that

$$\| \delta \|_2 < \sigma \cdot \Phi^{-1}(p) = R$$

where Φ^{-1} is the Gaussian inverse CDF

Theorem 1 (Binary Case)

- Observation
  - This theorem assumed nothing about f
  - R is large when
    - p is large (in fact, as $p \to 1, \Phi^{-1}(p) \to \infty$)
    - noise, σ, is high

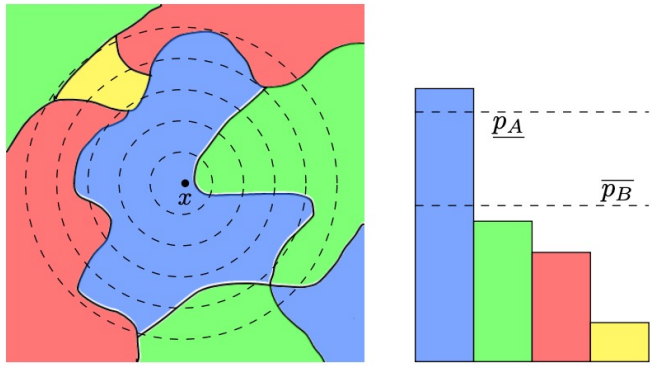
Theorem 1 (General Case)

- Let p_A be the probability of the most probable class, and p_B be the probability of the "runner up" (output by "smoothed" classifier)
- Suppose $c_A \in \mathcal{Y}$ and $p_A, \overline{p_B} \in [0,1]$ satisfy

$$\mathbb{P}[f(x+\epsilon) = c_A] \geq p_A \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}[f(x+\delta = c]$$

- Then $g(x+\delta) = c_A \quad \forall \| \delta \|_2 < R,$

where $R = \frac{\sigma}{2} \cdot (\Phi^{-1}(p_A) - \Phi^{-1}(\overline{p_B}))$

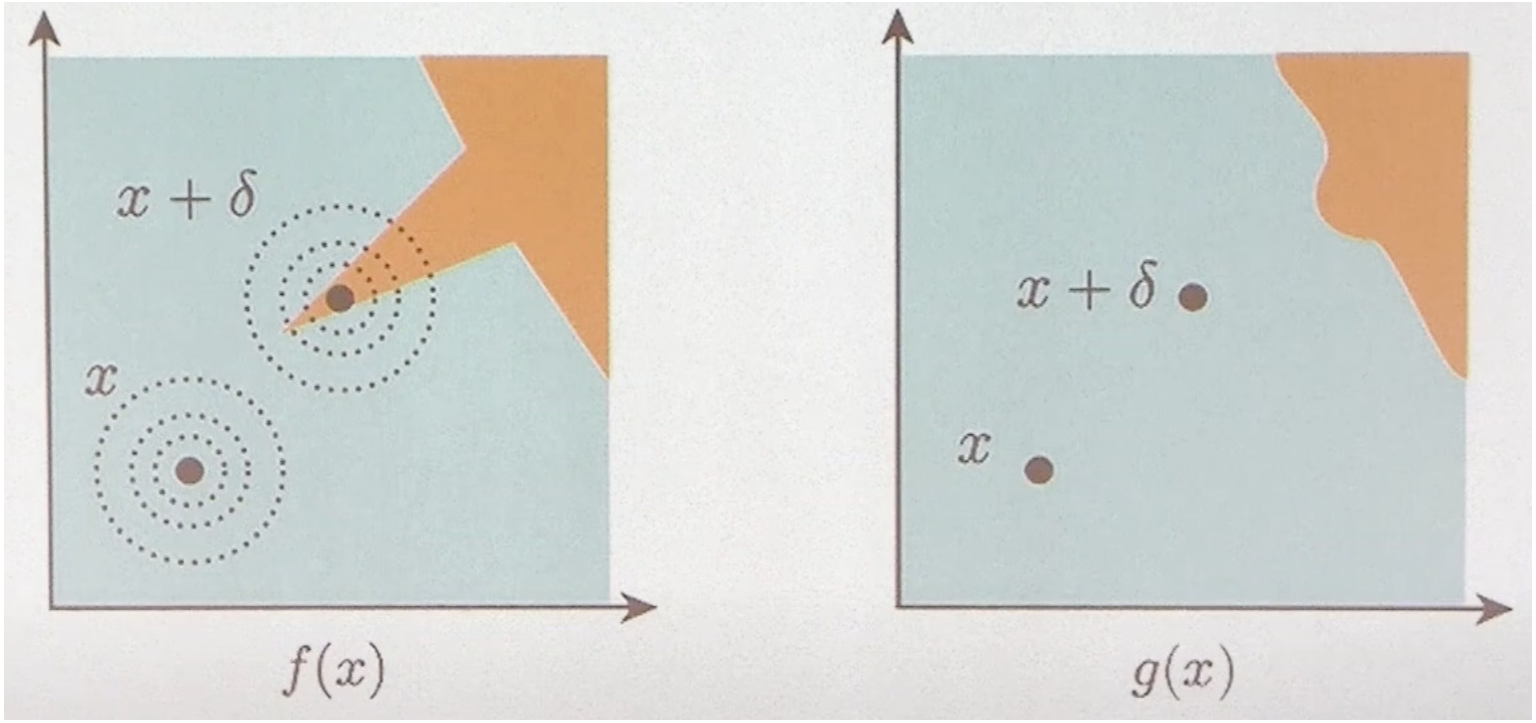


[Cohen et al.]

Corollary

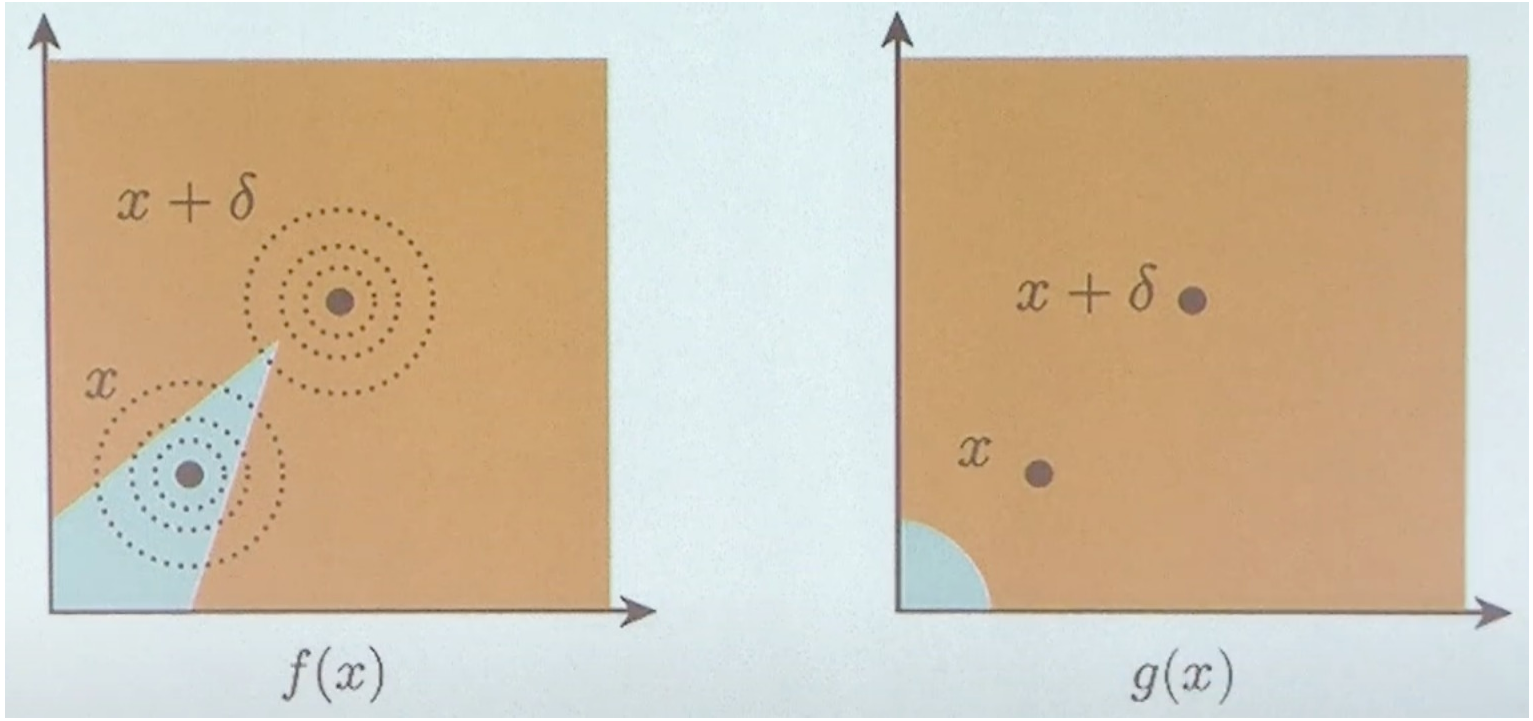
- Assume $\underline{p_A} + \overline{p_B} \leq 1$. For any perturbation δ with $\| \delta \|_2 > R$, $\exists$ a base classifier f consistent with the class probabilities for which $g(x + \delta) \neq c_A$

“Proof” of Theorem 1



[Kolter]

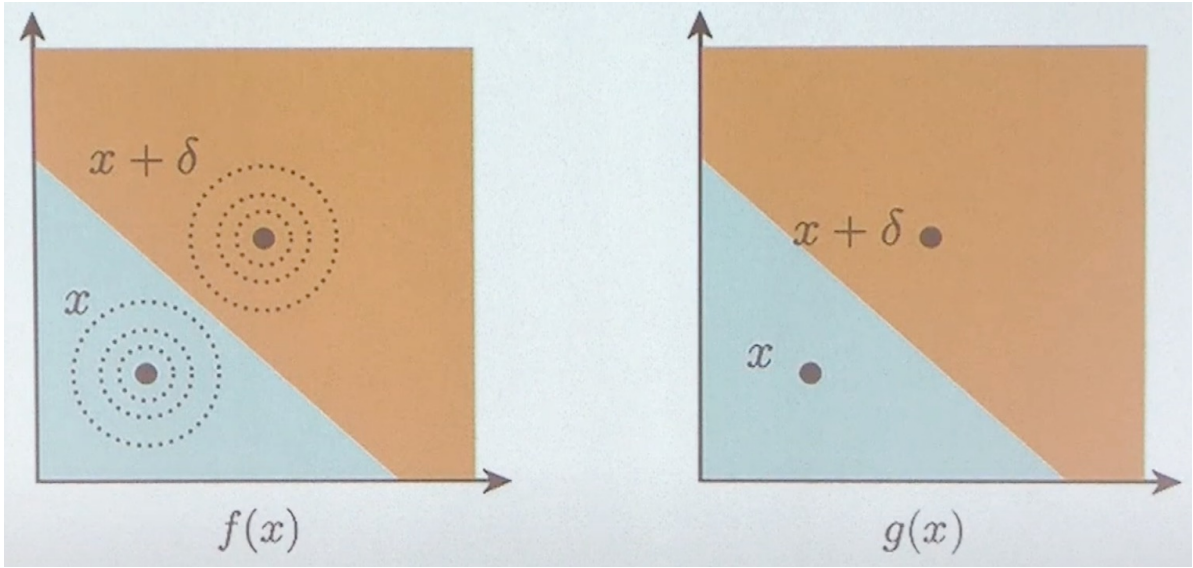
“Proof” of Theorem 1



[Kolter]

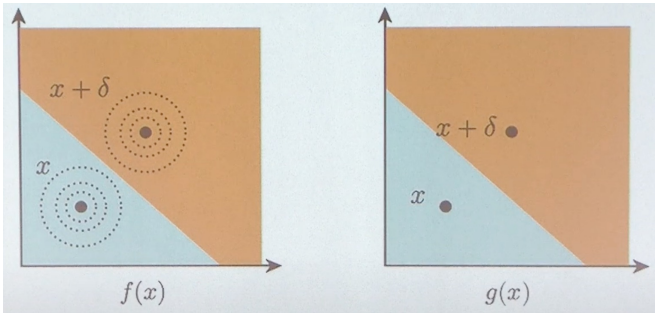
“Proof” of Theorem 1

- What is the worst case decision-boundary?



“Proof” of Theorem 1

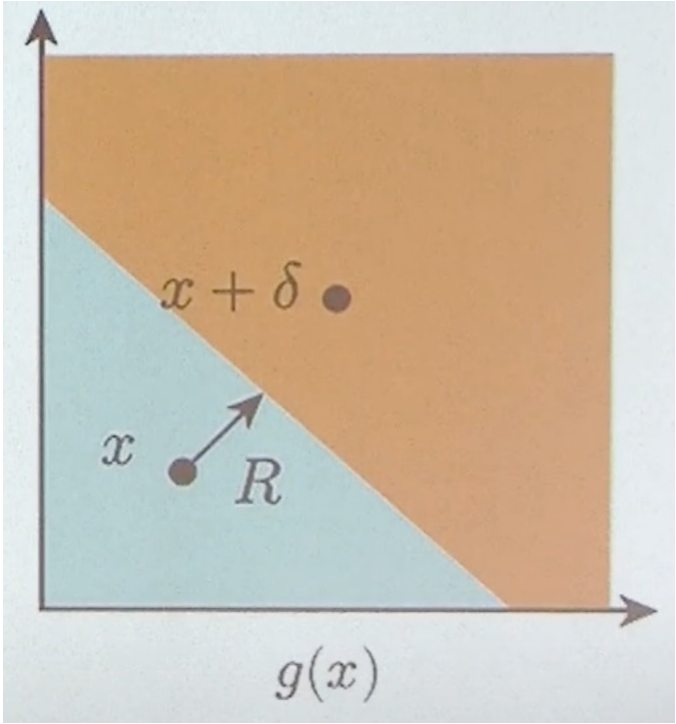
- Worst case for our classifier is when x and $x + \delta$ are separated by a linear decision boundary (Neyman-Pearson Lemma)
- When we apply Gaussian smoothing to a linear separator, it remains linear



“Proof” of Theorem 1

- For a linear classifier, we can directly compute the ℓ_2 distance to the decision boundary

$R = \sigma \cdot \Phi^{-1}(p)$ (binary case)



Empirical Results

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

Empirical Results

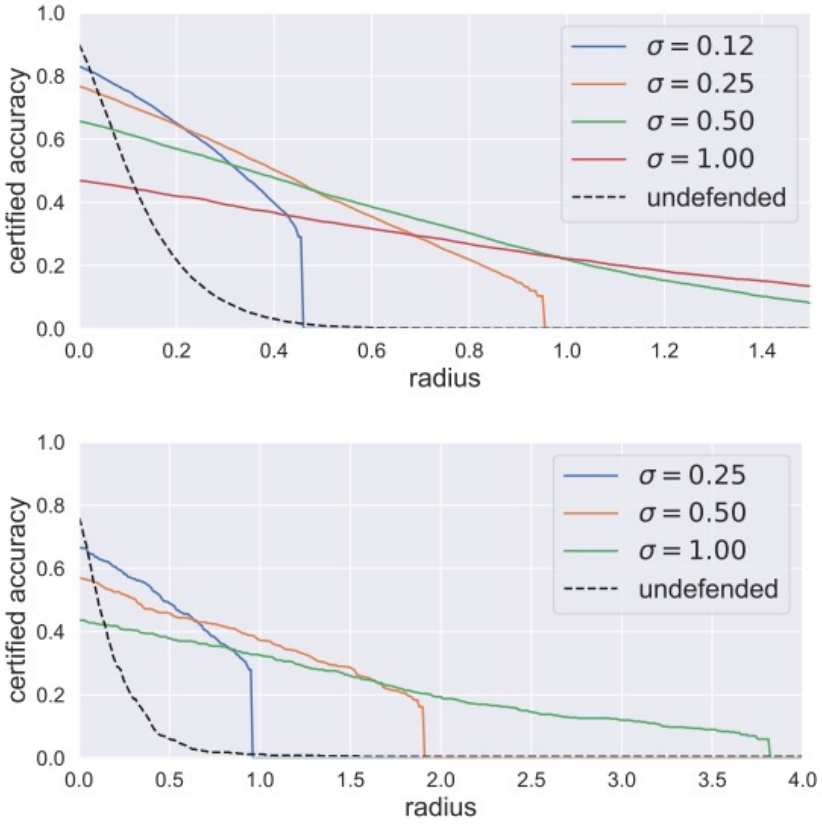


Figure 6. Approximate certified accuracy attained by randomized smoothing on CIFAR-10 (**top**) and ImageNet (**bottom**). The hyperparameter σ controls a robustness/accuracy tradeoff. The dashed black line is an upper bound on the empirical robust accuracy of an undefended classifier with the base classifier's architecture.

Strengths

- Gives provable guarantees for the robustness of *any* classifier
- Does not require a layer-by-layer decomposition of a model to analyze its robustness

Limitations

- Randomized smoothing guarantees robustness for the "smoothed" classifier, not the base one
- Since the probability of correct classification under smoothing can't be computed for a neural network, we have to use sampling
  - These bounds hold w.h.p., though
- The R that is certified is *very* small compared to noise distribution

$$\| \epsilon \|_2 = O(\sigma \cdot \sqrt{d}),\ R = O(\sigma)$$