CY 7790, Lecture 8 Notes
Evasion Attacks: Explaining adversarial examples; certified defenses

Hye Sun Yun and Jaydeep Borkar

October 7, 2021

1 Ilyas et al. Adversarial Examples Are Not Bugs, They Are Features

Presented by Harsh Chaudhari

Problem Statement. This paper tries to argue that the existence of adversarial examples are not "always" due to the models being at fault. The authors of the paper argue that adversarial examples exist due to the presence of non-robust features in the data. They set out to explain these features within a theoretical framework and show them in standard datasets. They also demonstrate that it is possible to get a robust model by doing just standard training (and not adversarial training) on the robust features. In addition, they seek to explain the phenomenon of adversarial transferability within the same framework by arguing that transferability is the property of the dataset and not model. Adversarial transferability is the phenonmenon that adversarial perturbations computed for one model often transfer to other, independently trained models.

Threat Model. The threat model for this paper is on evasion attacks. The adversary is trying to find a perturbation for input images that leads to misclassification of the model during inference time. The attacks can be either untargeted or targeted depending on the setting. The adversary can have different levels of knowledge. In some cases, they might know the training data or the type of model or even the model's hyperparameters. It can be white-box, gray-box, or black-box attack based on the setting.

Methodology. The authors of the paper first formally define what robust and non-robust features in a dataset mean. They further propose a method to disentangle the robust features from non-robust ones using an already adversarially trained classifier. This gives separate datasets with robust and non-robust features. They observe the interesting behaviors of these two features in image classification task, and apply this concept to explain adversarial transferability.

A feature is defined to be a function mapping from the input space X to the real numbers, with the set of all features thus being $F = f : X -> \mathbb{R}$.

The paper defines useful, robust, and non-robust features in the following way:

ρ-robust features: For a given distribution D, a feature f is $\rho - useful(\rho > 0)$ if it is correlated with the true label in expectation. Figure 1(a) shows this definition. Also, $\rho_D(f)$ is defined as the largest ρ for which feature f is ρ-useful under distribution D.

γ-robustly useful features: A feature f is considered a robust feature if under adversarial perturbation (δ), f remains γ-useful (for any $\gamma >= 0$). Figure 1(b) shows this definition.

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 $\gamma >= 0$.

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

(a)

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

(b)

Figure 1: Formal definitions of ρ-robust features (a) and γ robustly useful features (b).

To generate the dataset with robust features, the authors used a robust classifier and looked at the last hidden layer in the neural network as all previous layers provide a compact representation of the image while the robust features lie in the representation layer (last hidden layer). They picked a sample from the dataset at random and made it as close to the original as possible in the space by minimizing the distance $\min \|g(x_r) - g(x)\|_2$. To optimize this min distance, the gradient was computed. The dataset with robust features only have robust features which basically means that features perceptible to humans. To generate the dataset with non-robust features, a similar process was done with a standard classifier to only capture the non-robust features of a sample. The dataset with non-robust features have features that humans cannot understand but are features that the models find to be very important.

Using the dataset with robust features, a non-robust, standard model was trained. This model showed to have good robust accuracy and good standard accuracy shown in Figure 2(a). Then, they trained a standard model on the dataset with non-robust features and showed the it will provide good standard accuracy but bad robust accuracy (Figure 2(a)). This showed that doing a standard training on robust features can generate robust classifiers and that non-robust features are enough for the model to generalize.

To demonstrate how adversarial transferability arises from utilizing similar non-robust features, a new dataset was created (one with robust features of one image overlaid with non-robust features of another unrelated image). This dataset created an image so that it looks misclassified to humans and has a label for non-robust features. A model was trained on this new dataset (containing non-robust features) and then was tested on a regular, unmodified test data. The results of the model was good standard accuracy by being able to accurately classify the regular test data. Figure 2(b).

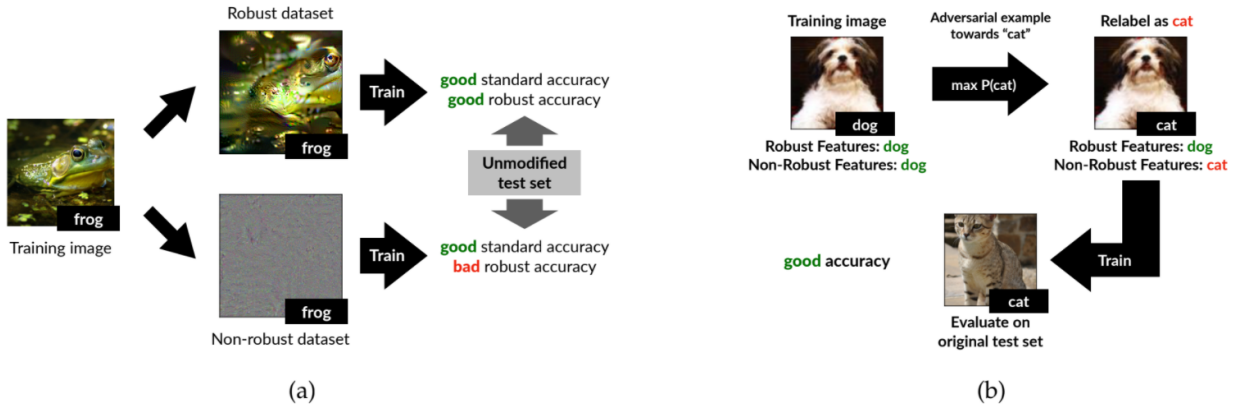


Figure 2: The two models trained to show the importance of robust and non-robust features. Model (a) is used to show how a model with standard trained using only robust features can produce a robust model. Model (b) is used to show how non-robust features can be very important for image classification tasks.

Class Discussion.

The second model with both robust and non-robust features: Does the model recognize the non-robust features as noise and try to optimize against noise? No, that is not necessarily the case. This second model has been trained with mismatched robust and non-robust feature labels. The importance of this model is to show that non-robust features are important for models when it comes to prediction/classification. Even the features that are considered "noise" help with predicting unmodified images.

Adversarial training: What would happen if we do adversarial training with either robust features or non-robust features? The answer seems to be "not sure."

Comparison with Szegedy paper: Do this paper's argument and observations contradict Szegedy paper on how individual neurons do not have semantic value? This paper doesn't contradict because we are talking about features in dataset and not neurons in neural networks.

Generating the non-robust dataset: The paper was not too clear on how the non-robust dataset was generated.

Using an already robust classifier to disentangle features: The paper seems to be a bit hand-wavy when it comes to needing a robust classifier to begin with to train a robust model. You can use robust dataset and train it with a standard model and get a robust model. However, to generate a robust dataset, you would have had a robust classifier to begin with so this method for training a robust model doesn't seem to be that helpful. If there could be some other cheaper way (apart from using an already adversarially robust classifier) to disentangle robust features from non-robust ones, it could turn out to be a cheaper alternative to make models adversarially robust.

2 Cohen et al. Certified Adversarial Robustness via Randomized Smoothing

Presented by John Abascal

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. This paper seeks to prove robustness guarantees for defenses against adversarial attacks. The authors of the paper show that using randomized smoothing as a defense technique provides us guarantees for any classifier's robustness. This paper proves the first tight robustness guarantee for random smoothing as a defense against adversarial attacks.

Threat Model. The threat model for this paper is on evasion attacks. The adversary is trying to force an incorrect prediction by the model by adding some perturbation to the input image during inference time. This objective is achieved by finding a perturbation, δ such that: $\max_{\delta} \text{loss}(x + \delta, \theta)$. The types of attacks this paper tries to defend against are probably gray-box or black-box attacks.

Methodology.

What is robustness? The authors of the paper define a classifier to be *certifiably robust* if for any input x, one can obtain a guarantee that the classifier's prediction is constant within some set often the l_2 or l_∞ ball around x.

What is randomized smoothing? To create a robust classifier, the authors of this paper proposes a technique called randomized smoothing. A smoothed classifier (at test time) is defined as $g(x) = \arg\max_{c \epsilon Y} \mathbb{P}[f(x + \epsilon) = c]$ where $\epsilon \sim N(0, \sigma^2 I)$ for a classification problem from $\mathbb{R}^d$ to Y with some arbitrary base classifier f (Figure 3). In other words, smoothed classifier maximizes the probability of a label on taking many points with some perturbation sampled from a Gaussian distribution. On the other hand, a regular classifier can have some jagged areas in the decision boundary where we can find adversarial examples very easily. However, with smoothing it will be harder to find them. For randomized smoothing to work, we need a base classifier that is able to correctly classify images with noise.

The randomized smoothing procedure for a neural network is as follows:

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

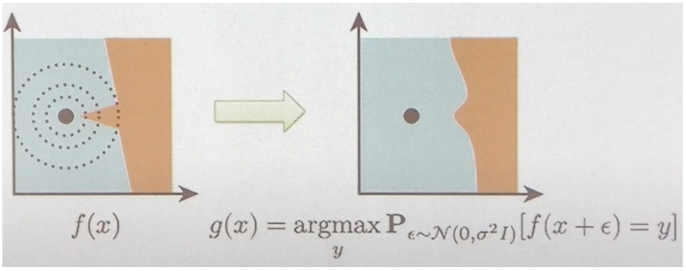


Figure 3: Randomized smoothing explained with a visual of the decision boundary.

Figure 4 provides the details of Theorem 1 which is key to proving that randomized smoothing is certifiably robust. There are several observations of Theorem 1:

1. Theorem 1 assumes nothing about f
2. The certified radius R is large when (1) the noise σ is high, (2) the probability of the top class c_A is high, and (3) the probability of each other class is low.
3. The certified radius R goes to ∞ as $p_A - > 1$, $\Phi^{-1}(p) - > \infty$

The l_2 robustness guarantee is tight if (2) is all that is known about f. It is impossible to certify any superset of the l_2 ball with radius R based on Theorem 2 (Figure 5. Figure 6 provides Theorem 1 for the special case when there are only two classes.

Theorem 1. *Let $f : \mathbb{R}^d \to \mathcal{Y}$ 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

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \quad (3)$$

Figure 4: Details of Theorem 1.

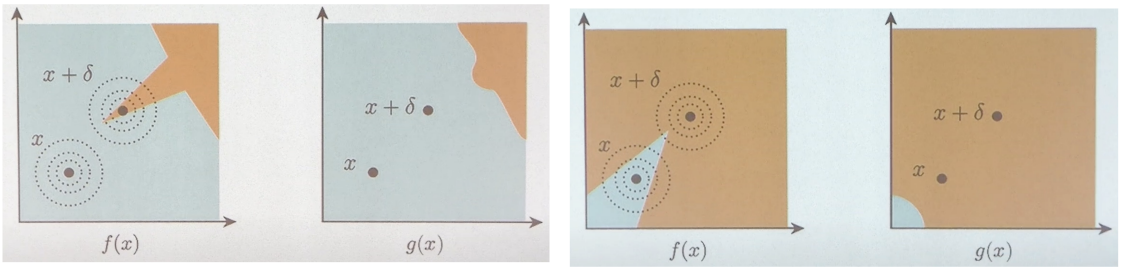
Figures 7 and 8 provide an intuitive, informal proof of the Theorem 1.

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 $g(x + \delta) \neq c_A$.*

Figure 5: Details of Theorem 2.

Theorem 1 (binary case). *Suppose $\underline{p_A} \in (\frac{1}{2}, 1]$ satisfies $\mathbb{P}(f(x + \varepsilon) = c_A) \geq \underline{p_A}$. Then $g(x + \delta) = c_A$ for all $\|\delta\|_2 < \sigma\Phi^{-1}(\underline{p_A})$.*

Figure 6: Details of Theorem 1 for the binary classification case.



(a) (b)

Figure 7: An intuitive, informal proof of Theorem 1. $g(x)$ being certified means that $g(x)$ and $f(x)$ are the same. Therefore, the adversarial example would not be successful with randomized smoothing. (a) shows the case when the adversarial example $x + \delta$ becomes the same class as x. (b) shows the case when x becomes the same class as $x + \delta$ class.

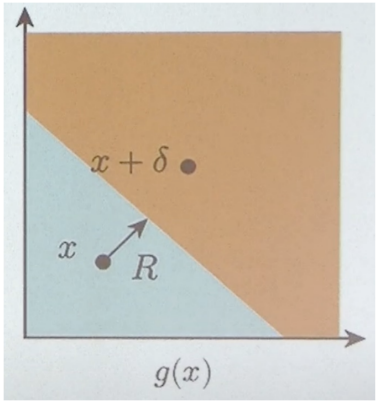


Figure 8: For a linear classifier, we can directly compute the l_2 distance to the decision boundary: $R = \sigma \cdot \Phi^{-1}(p)$ for the binary classification case.

Class Discussion.

Theorem 1 (General Case): P_B is the second ranked class. P_B will lose it's meaning if the upper bound is removed because it would be the same as the maximum of all the classes.

"Proof" of Theorem 1: $g(x)$ being certified doesn't necessarily mean that it has good classification, but it means that

$g(x)$ and $f(x)$ are the same. In most cases, $f(x)$ and $g(x)$ can have the same outputs but accuracy will be very low if we have more non-linearity.

Radius as the constraint: Wouldn't we not have any other classes if we were to follow this randomized smoothing? No, because we can only create adversarial examples within some radius around the original sample. The distance from the decision boundary depends on the confidence of the prediction of x so we will still have different classes and not just 1 big class.

Certifiably based on geometrics: Does certifiability change based on geometrics? The worst case for a classifier is when x and $x + \delta$ are separated by a linear decision boundary (Neyman-Pearson Lemma). Even when we apply Gaussian smoothing to a linear separator, it remains linear.

Robustness for "smoothed" classifier: Randomized smoothing guarantees robustness for the "smoothed" classifier, not the base one. Randomized smoothing cannot be also used to certify robustness for defenses such as defensive distillation.

R that is certified is very small: The R that is certified is VERY small compared to the noise distribution: $\|\epsilon\|_2 = O(\sigma \cdot \sqrt{d}), R = O(\sigma)$.

Certified accuracy drop: It was observed that as we increase the amount of noise in the data (stronger perturbation), the certified accuracy decreases.

Tradeoff: There is a tradeoff between robustness and accuracy with the proposed approach for defense.