Certified Adversarial Robustness via Randomized Smoothing

迟智名

July 29, 2022

目录

Contents

目录

Contribution

- Prove a tight robustness guarantee in L_2 norm for randomized smoothing with Gaussian noise.
- We can have no knowledge about the base classifier beyond the distribution of $f(x + \epsilon)$
- The smoothed classifier is not itself a neural network, though it leverages the discriminative ability of a neural network base classifier.

目录

Definition

定义 (Randomized Smoothing)

Let $f: \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ the smoothed classifier g returns:

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x + \varepsilon) = c) \tag{1}$$

$$\text{i.e.}\, g(x) = \arg\max_{c \in \mathcal{Y}} \; m(c), m(c) = m(\{x' \in \mathbb{R}^d : f(x') = c\})$$

Example

Figure 2. The smoothed classifier's prediction at an input x (left) is defined as the most likely prediction by the base classifier on random Gaussian corruptions of x (right; $\sigma = 0.5$). Note that this Gaussian noise is much larger in magnitude than the adversarial perturbations to which g is provably robust. One interpretation of randomized smoothing is that these large random perturbations "drown out" small adversarial perturbations.

One problem: How to measure the possibility? Sample

Robustness Guarantee

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

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

Illustration of the proof of Theorem 1

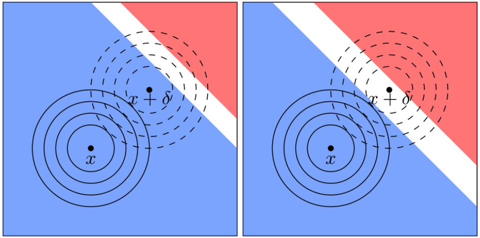


Figure 9. Illustration of the proof of Theorem 1. The solid line concentric circles are the density level sets of $X := x + \varepsilon$; the dashed line concentric circles are the level sets of $Y := x + \delta + \varepsilon$. The set A is in blue and the set B is in red. The figure on the left depicts a situation where $\mathbb{P}(Y \in A) > \mathbb{P}(Y \in B)$, and hence $g(x + \delta)$ may equal c_A. The figure on the right depicts a situation where $\mathbb{P}(Y \in A) < \mathbb{P}(Y \in B)$ and hence $g(x + \delta) \neq c_A$.

properties

- Theorem 1 assumes nothing about f
- The certified radius R is large when: (1) the noise level σ is high, (2) the probability of the top class c_A is high, and (3) the probability of each other class is low.
- The certified radius R goes to ∞ as $\underline{p_A} \to 1$ and $\overline{p_B} \to 0$. This should sound reasonable: the Gaussian distribution is supported on all of $\mathbb{R}^d$, so the only way that $f(x+\varepsilon) = c_A$ with probability 1 is if $f = c_A$ almost everywhere.

How, if $\delta > \|R\|$

Theorem 2 (restated). *Assune $\underline{p_A} + \overline{p_B} \leq 1$. For any perturbation $\delta \in \mathbb{R}^d$ with $\|\delta\|_2 > R$, there exists a base classifier f^* consistent with the observed class probabilities (6) such that if f^* is the base classifier for g, then $g(x+\delta) \neq c_A$.*

Proof. We re-use notation from the preceding proof.

Pick any class c_B arbitrarily. Define A and B as above, and consider the function

$$f^*(x) := \begin{cases} c_A & \text{if } x \in A \\ c_B & \text{if } x \in B \\ \text{other classes} & \text{otherwise} \end{cases}$$

This function is well-defined, since $A \cap B = \emptyset$ provided that $\underline{p_A} + \overline{p_B} \leq 1$.

By construction, the function f^* satisfies (6) with equalities, since

$$\mathbb{P}(f^*(x+\varepsilon) = c_A) = \mathbb{P}(X \in A) = \underline{p_A} \qquad \mathbb{P}(f^*(x+\varepsilon) = c_B) = \mathbb{P}(X \in B) = \overline{p_B}$$

It follows from (13) and (14) that

$$\mathbb{P}(Y \in A) < \mathbb{P}(Y \in B) \iff \|\delta\|_2 > R$$

By assumption, $\|\delta\|_2 > R$, so $\mathbb{P}(Y \in A) < \mathbb{P}(Y \in B)$, or equivalently,

$$\mathbb{P}(f^*(x+\delta+\varepsilon) = c_A) < \mathbb{P}(f^*(x+\delta+\varepsilon) = c_B)$$

Therefore, if f^* is the base classifier for g, then $g(x+\delta) \neq c_A$. $\square$

Binary Case

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* $\|\delta\|_2 < \sigma\Phi^{-1}(\underline{p_A})$.

Linear base classifier

For two-class linear classifier

$$f(x) = sign(w^T x + b)$$

we can get

- the distance from any input x to the decision boundary is

$$|w^T x + b| / \|w\|^2$$

- the smoothed classifier g is identical to the base classifier f.
- the true robust radius is $|w^T x + b| / \|w\|$

Noise level can scale with image resolution

Figure 4. Left to right: clean 56 x 56 image, clean 224 x 224 image, noisy 56 x 56 image ($\sigma = 0.5$), noisy 224 x 224 image ($\sigma = 0.5$).

目录

details

 # evaluate g at x
function PREDICT(f, σ, x, n, α)
`counts` $\leftarrow$ SAMPLEUNDERNOISE(f, x, n, σ)
$\hat{c}_A, \hat{c}_B \leftarrow$ top two indices in `counts`
$n_A, n_B \leftarrow$ `counts`[$\hat{c}_A$], `counts`[$\hat{c}_B$]
if BINOMPVALUE(n_A, $n_A + n_B$, 0.5) $\leq \alpha$ **return** $\hat{c}_A$
else return ABSTAIN

- SAMPLEUNDERNOISE(f, x, num, σ) works as follows:
  1. Draw num samples of noise, $\varepsilon_1 \dots \varepsilon_{\text{num}} \sim \mathcal{N}(0, \sigma^2 I)$.
  2. Run the noisy images through the base classifier f to obtain the predictions $f(x + \varepsilon_1), \dots, f(x + \varepsilon_{\text{num}})$.
  3. Return the counts for each class, where the count for class c is defined as $\sum_{i=1}^{\text{num}} \mathbf{1}[f(x + \varepsilon_i) = c]$.
- BINOMPVALUE(n_A, $n_A + n_B$, p) returns the p-value of the two-sided hypothesis test that $n_A \sim$ Binomial($n_A + n_B$, p). Using `scipy.stats.binom_test`, this can be implemented as: `binom_test(nA, nA + nB, p)`.

[1]Hung, K. and Fithian, W. Rank verification for exponential families. The Annals of Statistics, (2):758–782, 04 2019.

details

 # 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 a one-sided $(1-\alpha)$ lower confidence interval for the Binomial parameter p given that $k \sim \text{Binomial}(n, p)$. In other words, it returns some number $\underline{p}$ for which $\underline{p} \leq p$ with probability at least $1-\alpha$ over the sampling of $k \sim \text{Binomial}(n, p)$. Following Lecuyer et al. (2019), we chose to use the Clopper-Pearson confidence interval, which inverts the Binomial CDF (Clopper & Pearson, 1934). Using `statsmodels.stats.proportion.proportion_confint`, this can be implemented as

```
proportion_confint(k, n, alpha=2*alpha, method="beta")[0]
```

typical:the mass of $f(x+\varepsilon)$ not allocated to c_A entirely to one runner-up class.

目录

Gaussian data augmentation

- In order for g to classify the labeled example (x, c) correctly and robustly, f needs to consistently classify $\mathcal{N}(x, \sigma^2 I)$ as c
- In high dimension, the Gaussian distribution $\mathcal{N}(x, \sigma^2 I)$ places almost no mass near its mode x.
- As a consequence, when is moderately high, the distribution of natural images has virtually disjoint support from the distribution of natural images corrupted by $\mathcal{N}(x, \sigma^2 I)$
- Therefore, if the base classifier f is trained via standard supervised learning on the data distribution, it will see no noisy images during training

mnist&ImageNet

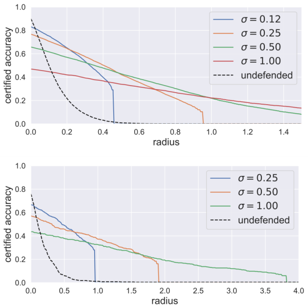


Figure 6. Approximate certified accuracy attained by randomized smoothing on CIFAR-10 (**top**) and ImageNet (**bottom**). The hyper-parameter σ controls a robustness/accuracy tradeoff. The dashed black line is an upper bound on the empirical robust accuracy of an undefended classifier with the base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

- $\alpha = 0.001, n_0 = 100, n = 10^5$
- there is a hard upper limit to the radius
- CIFAR-10 :110-layer residual network; certifying each example took 15 seconds on an NVIDIA RTX 2080 Ti.
- ImageNet :base classifier - ResNet-50; took 110 seconds.
- Full CIFAR-10 test set and 500 examples from the ImageNet test set.
- Black line:DeepFool l_2 adversarial attack
- Blance between accuracy and radii

Result

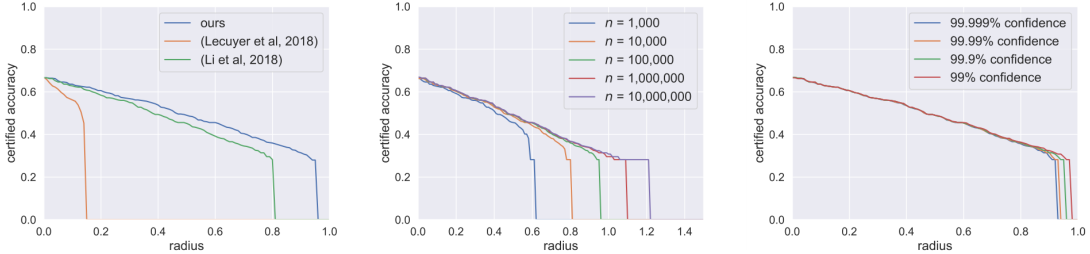
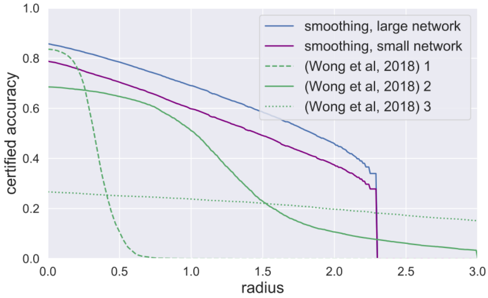


Figure 8. Experiments with randomized smoothing on ImageNet with $\sigma = 0.25$. **Left**: certified accuracies obtained using our Theorem 1 versus those obtained using the robustness guarantees derived in prior work. **Middle**: projections for the certified accuracy if the number of samples n used by CERTIFY had been larger or smaller. **Right**: certified accuracy as the failure probability α of CERTIFY is varied.

Comparison to baselines



2 Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In Advances in Neural Information Processing Systems 31, 2018

Prediction

N	CORRECT, ACCURATE	CORRECT, INACCURATE	INCORRECT, ACCURATE	INCORRECT, INACCURATE	ABSTAIN
100	0.65	0.00	0.23	0.00	0.12
1000	0.68	0.00	0.28	0.00	0.04
10000	0.69	0.00	0.30	0.00	0.01

Table 4. Performance of PRECICT as n is varied. The dataset was ImageNet and $\sigma = 0.25$, $\alpha = 0.001$. Each column shows the fraction of test examples which ended up in one of five categories; the prediction at x is "correct" if PREDICT returned the true label, while the prediction is "accurate" if PREDICT returned $g(x)$. Computing $g(x)$ exactly is not possible, so in order to determine whether PREDICT was accurate, we took the gold standard to be the top class over n =100,000 Monte Carlo samples.

Attack

- PGD:empirically assess the tightness of our bound
- If the example was correctly classified and certified robust at radius R, we tried finding an adversarial example for g within radius $1.5R$ and within radius $2R$. We succeeded 17% of the time at radius $1.5R$ and 53% of the time at radius $2R$.

Thank you