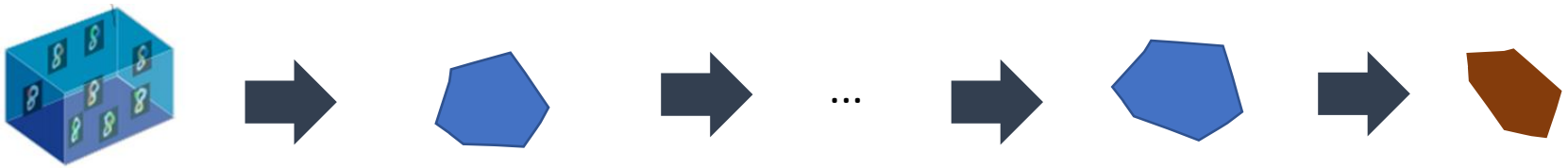
Reliable and Interpretable Artificial Intelligence

Lecture 12: Randomized Smoothing for Robustness

Martin Vechev
ETH Zurich

Fall 2020

Deterministic Certification: reminder



1. Infers (typically convex) shapes capturing intermediate invariants. Usually, relaxations are variants of Polyhedra to balance analysis scalability and precision.
2. The method is general. It can handle any safety property (not just robustness).
3. Deterministic guarantees are provided.
4. Very active research area constantly pushing the size of networks.

Key challenge: scaling to large networks

Randomized Smoothing

Key idea: construct a classifier **g** out of an existing classifier **f**, in a way which ensures that **g** has certain statistical robustness guarantees.

The construction does not assume knowledge of **f** and can scale to large networks. The method focuses on restricted robustness-like properties, and requires sampling at inference time, not required by convex methods. The usual standard accuracy vs. robustness trade-off is present here as well.

Certified Adversarial Robustness via Randomized Smoothing, ICML 2019
Cohen, Rosenfeld, Kolter https://arxiv.org/pdf/1902.02918.pdf

Constructing classifier g

Given a base classifier $f: \mathbb{R}^d \to \mathcal{Y}$,

construct a smoothed classifier g as follows:

$$g(x) := \mathbf{argmax}_{c \in \mathcal{Y}} \ \mathbb{P}_\epsilon(f(x+\epsilon) = c)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$

σ controls the amount of noise

Isotropic Gaussian: restricted co-variance matrix

Robustness Guarantee

Suppose that: $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:

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

Then:

$$g(x+\delta) = c_A \text{ for all } \| \boldsymbol{\delta} \|_2 < R \qquad \text{where:}$$

certification radius $R := \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$

and Φ^{-1} is the inverse of the standard Gaussian CDF.

Robustness Guarantee

Suppose that: $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:

c_A is the most likely class

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

A **lower bound** on the true highest probability p_A

An **upper bound** on the true second-highest probability p_B

In theory, we could potentially compute the true exact probabilities p_A, p_B using for instance exact probabilistic inference solvers such as PSI [https://github.com/eth-sri/psi]. However, exact inference solvers do not scale to realistic networks and we will approximate the probabilities (with certain statistical guarantees).

Robustness Guarantee

If $x \sim \mathcal{N}(0,1)$ and probability $p \in [0,1]$, then $\Phi^{-1}(p) = v$ s.t. $\mathbb{P}_x(x \leq v) = p$

Φ^{-1} is **monotone**: higher values of p produce higher values for $\Phi^{-1}(p)$

For fixed noise σ, to increase radius R, we want higher $\underline{p_A}$ and lower $\overline{p_B}$.

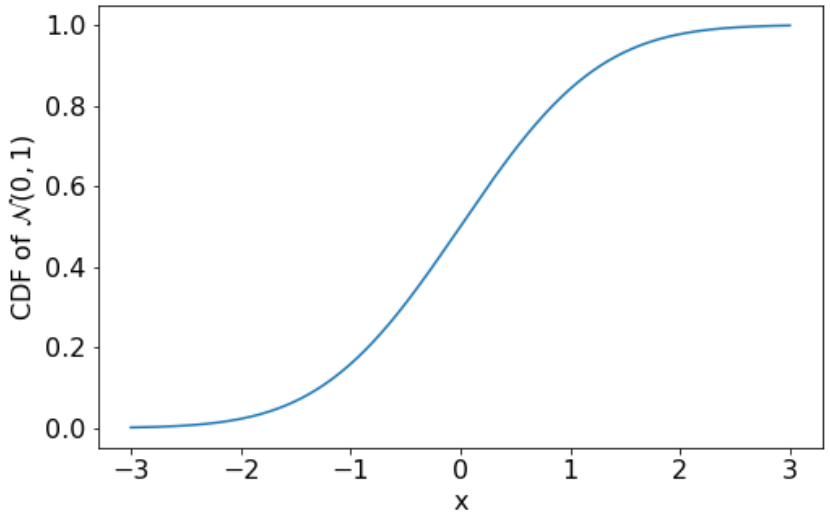
Thus, it is important that classifier f is pre-trained to perform well under Gaussian noise.

Increasing noise σ can increase certified R but can reduce accuracy.

certification radius $R := \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$

Φ^{-1} is the inverse of the standard Gaussian CDF.

Visualization of the normal $\mathcal{N}(0,1)$ CDF



Note: result of $\Phi^{-1}(p)$ can be negative but radius R is always positive due to Φ^{-1} being monotone and the theorem requiring $\underline{p_A} \geq \overline{p_B}$

certification radius $R := \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$

Φ^{-1} is the inverse of the standard Gaussian CDF.

Certified and Standard Accuracy

Note: the certified radius R we obtain may differ between different input x's because the true probabilities p_A and p_B and correspondingly their lower and upper bounds, depend on the input x.

Thus, to compute **certified accuracy**, we pick a target radius T and count the number of points in the test set whose certified radius $R \geq T$ and where the predicted c_A matches the test set label. **Standard accuracy** is instantiated with $T = 0$.

Then:

$$g(x + \delta) = c_A \text{ for all } \| \delta \|_2 < R \qquad \text{where:}$$

certification radius $R := \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$

and Φ^{-1} is the inverse of the standard Gaussian CDF.

Key challenge:

To compute **certified accuracy** of g, we need to get the probabilities p_A and p_B or their bounded versions. However, doing so analytically is not possible due to inherent costs. Thus, we resort to **sampling which will provide statistical guarantees** on the probabilities.

Certification Procedure

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

Certification Procedure

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

To prevent selection bias, sample first to find top label, then sample again with the number of samples $n >> n_0$

Certification Procedure

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

To prevent selection bias, sample first to find top label, then sample again with the number of samples $n >> n_0$

SampleUnderNoise(f, x, n, σ):

evaluates f at $x + \epsilon_i$ for $i \in \{1, \ldots, n\}$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$ and returns a dictionary of class counts.

Certification Procedure

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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$(k, n, 1 - \alpha)$:

assuming $k \sim$ Binomial(n, p) for some unknown p, it returns probability p_l such that $p_l \leq p$ with probability $1 - \alpha$. That is, it finds a lower bound on this unknown probability of success p.

There are many methods to compute confidence intervals, the smoothing paper uses Clopper-Pearson.

Certification: Guarantees

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

Let: $\overline{p_B} = 1 - \underline{p_A}$. Because $\underline{p_A} > \frac{1}{2}$, we know that $\overline{p_B} < \frac{1}{2}$ and therefore $\underline{p_A} \leq \overline{p_B}$. We now can instantiate the theorem to obtain the certified radius.

Certification: Guarantees

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

To get the radius:

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

$$= \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) + \Phi^{-1}(\underline{p_A})\right)$$

$$= \sigma\,\Phi^{-1}(\underline{p_A})$$

Certification: Guarantees

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

Then we get the guarantee from the theorem:

with probability at least $1-\alpha$, if CERTIFY returns class $\widehat{c_A}$ and radius $R = \sigma\,\Phi^{-1}(\underline{p_A})$, then $g(x+\delta) = \widehat{c_A}$ for all $\| \delta \|_2 < R$.

Robustness vs. Accuracy

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

Note: We certify that g returns the same class for all inputs in radius R not that this output is necessarily correct (that is, same label as in the test set)!

There are several reasons why one may obtain an incorrect label (incorrect includes abstentions).

Robustness vs. Accuracy

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

$\sigma = 0$ $\sigma = 0.25$ $\sigma = 0.5$ $\sigma = 1.0$

Reason I:

With increasing noise σ, it is more likely that the perfect smoothed classifier $g(x)$ returns c_A which may not be the label in the test set.

Robustness vs. Accuracy

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
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

$\sigma = 0$ $\sigma = 0.25$ $\sigma = 0.5$ $\sigma = 1.0$

Reason II:

Even if the perfect smoothed classifier returns c_A in the test set, it is possible that because: (i) n_0 is small, or (ii) the true probabilities p_A and the next-best probability are similar, we obtain a label $\widehat{c_A}$ which differs from the c_A. And then, this almost certainly will lead to abstention which will be counted as incorrect label.

Effect of noise σ on Robustness vs. Accuracy

Each entry shows % of images in the test set (in this case ImageNet images), with provable radius $\geq r$ and label as in test set.

	$r = 0.0$	$r = 0.5$	$r = 1.0$	$r = 1.5$	$r = 2.0$	$r = 2.5$	$r = 3.0$
$\sigma = 0.25$	**0.67**	**0.49**	0.00	0.00	0.00	0.00	0.00
$\sigma = 0.50$	0.57	0.46	**0.37**	**0.29**	0.00	0.00	0.00
$\sigma = 1.00$	0.44	0.38	0.33	0.26	**0.19**	**0.15**	**0.12**

Standard Accuracy

We see that as noise increases, the standard accuracy drops but the certified robust radius increases, the same trade-off between accuracy and robustness we discussed before with adversarial training.

Reminder: all of these results are statistical in nature and not deterministic (due to sampling). That is, they hold with **high probability**.

Increasing certified radius for fixed noise σ may require many samples

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

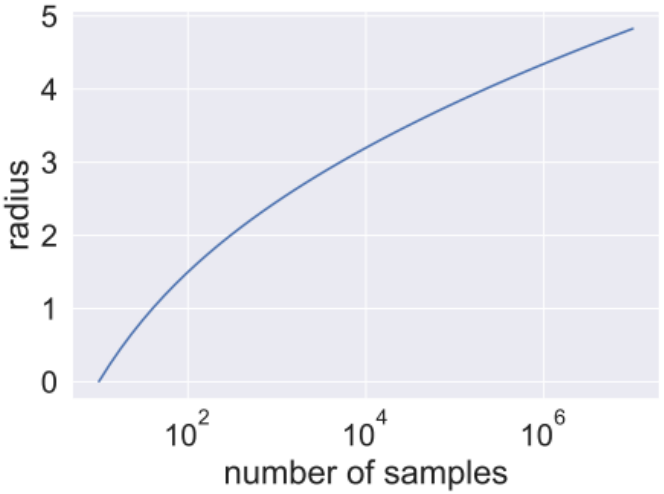
As Φ^{-1} is monotone, increasing $\underline{p_A}$ will increase the radius. To increase $\underline{p_A}$ we need to get the base classifier f to classify more points as $\widehat{c_A}$.

However, even in the extreme case where all points are always classified as $\widehat{c_A}$ by f, increasing the number of samples will only slowly grow the radius.

Increasing certified radius for fixed noise σ may require many samples

function CERTIFY($f, \sigma, x, n_0, n, \alpha$)
  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

In the best case scenario where f always classifies to c_A, we have that with confidence $1-\alpha$, a tight $\underline{p_A}$ lower bound is $\alpha^{\frac{1}{n}}$. Plotting the resulting radius $\sigma \cdot \Phi^{-1}(\alpha^{\frac{1}{n}})$ for $\alpha = 0.001$ and $\sigma = 1$, we see that increasing the number of samples will only slowly grow the radius.



Once a classifier is certified on the test set (via sampling as discussed so far), we need to actually use this classifier at inference time, and again we resort to sampling (with statistical guarantees).

Inference

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

Additional work needed at inference time, which can be expensive, depending on the number of samples

The **null hypothesis** is: the true probability of success of a Bernoulli trial is q.

BinomialPValue(i, n, q): returns the p-value of the null hypothesis, evaluated on n statistically independent samples with i successes.

In our case, the null hypothesis: the true probability of f returning $\widehat{c_A}$ is $q = 0.5$ (meaning the classes are indistinguishable).

Inference

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

We accept the null hypothesis if the returned p-value is $> \alpha$

We reject the null hypothesis if the returned p-value is $\leq \alpha$

If α is small (typically 0.001) , then we may often accept the null hypothesis and ABSTAIN, but we will be more confident in our predictions. If α is higher , then we may make prediction more often, but make more mistakes.

Inference: Guarantees

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

We can prove that:

it returns the wrong class $\widehat{c_A} \neq c_A$ with probability at most α

Inference Guarantees: Proof Sketch

$$\mathbb{P}(\widehat{c_A} \neq c_A, \text{no abstain})$$

$$= \mathbb{P}(\widehat{c_A} \neq c_A) \cdot \mathbb{P}(\text{no abstain} \mid \widehat{c_A} \neq c_A)$$

$$\leq \underbrace{\mathbb{P}(\text{no abstain} \mid \widehat{c_A} \neq c_A)}_{= \alpha}$$

see *Rank verification for exponential families*, Hung & Fithian The Annals of Statistics, 2019 https://arxiv.org/abs/1610.03944

$$\leq \alpha$$

Generalizing Smoothing

Certified Defense to Image Transformations via Randomized Smoothing, NeurIPS'2020
Fischer, Baader, Vechev https://www.sri.inf.ethz.ch/publications/fischer2020smoothing

Given a base classifier $f: \mathbb{R}^d \to \mathcal{Y}$,
and image transformation ψ_α: $\mathbb{R}^d \to \mathbb{R}^d$
construct a smoothed classifier g as follows:

standard smoothing is:

$\psi_\epsilon(x) = x + \epsilon$

$$g(x) := \mathbf{argmax}_{c \in \mathcal{Y}} \ \mathbb{P}_\epsilon(f(\psi_\epsilon(x)) = c)$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{1})$ and requiring **composition**: $\psi_\alpha(\psi_\beta) = \psi_{\alpha+\beta}$

- One obtains the same guarantees as standard smoothing but also further generalizations (e.g., relaxed distributional guarantees and individual guarantees).
- We instantiate ψ with geometric transformations (e.g., rotation, translation) as in earlier lectures. A key challenge here is handling interpolation [see paper].

Summary

- We introduced randomized smoothing, a method which constructs robust classifiers by introducing Gaussian noise which induces a robustness radius. A benefit of smoothing is that it scales to large networks.

- Smoothing relaxes the standard deterministic guarantees into statistical guarantees on the robustness of the classifier.

- To obtain higher certified radius, one may need many samples. It also requires sampling at inference time which convex methods do not. The classic trade-off of accuracy vs. robustness is also present here and is controlled by the amount of noise.

- Generalizing smoothing to different properties is harder than convex methods.