Smoothed Robustness Analysis: Bridging worst- and average-case robustness analyses via smoothed analysis

Anonymous authors
Paper under double-blind review

Abstract

Understanding the robustness of neural networks has attracted significant attention due to its sensitivity to adversarial and noise attacks being still a major drawback. In one extreme, the worst-case approach gives a region in the input space robust to any perturbation, i.e., a worst-case region. On the other, the average-case approach describes robustness against random perturbations. Several studies have attempted to bridge these two extremes. Among them, randomized smoothing became a prominent approach by certifying a worst-case region of a classifier subject to random noise. Here, inspired by smoothed analysis of algorithmic complexity, which bridges the worst- and average-case analyses of algorithms, we propose a new framework of robustness analysis of classifiers, which contains randomized smoothing as a special case. Then, starting from the framework's requirements, we propose a margin loss-based robustness analysis. This analysis, different from randomized smoothing, in case of having access to the classifier's Lipschitz constant, gives a certified radius that doesn't scale with the input noise variance, making this robustness analysis suitable even when the noise variance is small. To validate our approach, we evaluated the robustness of 1-Lipschitz neural networks with the margin-based certified radius as objective function for the MNIST classification task. We found that with the margin-based loss both adversarial and noise robustness were improved in comparison to the randomized smoothing-based one. Attempting to capture empirical robustness, we also compare the trained neural networks to previously reported human-level robustness. The code used in all experiments as well as the data and code for plotting the images from the results section are available in the supplementary material.

1 Introduction

In the last decade, we witnessed the biggest advance of Machine Learning (ML) for real-world tasks due to the practical instantiation of Neural Networks (NN) through Deep Learning (DL). Some well-known tasks excelled by and widely deployed with DL are object detection and classification in computer vision, machine translation and dialogue systems in natural language processing, and recent technological advances, such as protein structure prediction, photo-realistic image generation, and control of self-driving vehicles.

Despite its fast development to solve increasingly complex tasks, one of DL's weak spots is the existence of small perturbations in the input information that results in faulty responses. These tampered inputs are known as Adversarial Example (AE) and were first demonstrated by Szegedy et al. (2014). In their work, they generated adversarial examples in an image classification task by taking a gradient step w.r.t. the input in the direction that maximizes the loss towards an incorrect class and reported two features that make AE potentially harmful: their possible imperceptibility to humans, and perturbations generated with one NN causing misclassification in another.

Since Szegedy et al. (2014) work, an entire branch of ML has been developed to deal with Adversarial Robustness (AR), i.e., the sensitivity to adversarial examples. In its applied axis, adversarial defenses attempt to improve the AR, while adversarial attacks attempt to break these defenses. On the theoretical axis there are two major branches of research, one concerned with the origins of its sensitivity, and the other with the development of analytical tools to quantify it.

One of the approaches to analytically quantify this sensitivity is AR certification. In this approach, the goal is to yield a set of instances in the input space to which the model is guaranteed to perform correctly, and the larger this set, the more robust the model is. For real-valued inputs, as in the case of image classification, the radius of a ball centered in a correctly classified input that doesn't contain any AE is named Certified Radius (CR). AR certification quantifies the sensitivity of classifiers to adversarial perturbations by providing a guaranteed safe region against any perturbation within it, which makes it a *worst-case* robustness analysis. However, even though the number of sensitive directions, i.e., the worst-case directions, can be large since the data space lies in a low dimensional manifold in a high-dimensional space, as is the case of image classification (Khoury & Hadfield-Menell, 2018; Gilmer et al., 2018), evidence suggests that these regions have small volume, since it was shown that adding small noise to AEs significantly reduces its harm (Cao & Gong, 2019; Shi et al., 2022). This means that a worst-case robustness analysis of a classifier can give a pessimistic portrayal of its robustness observed empirically, yielding a CR smaller than the standard deviation of noise that naturally occurs in image processing.

On the opposite direction of the worst-case description from adversarial robustness, a classifier's robustness can be quantified with respect to random perturbations (Franceschi et al., 2018; Weng et al., 2019; Couellan, 2021; Tit & Furon, 2021; Anderson & Sojoudi, 2023). In this setting, given a probability distribution, the goal is to obtain the probability of correct classification, i.e., its average-case performance. Since this can only be obtained exactly for simpler classifiers, such as the linear binary classifier, in practice one obtains lower and upper bounds, with a higher lower bound for larger variances indicating higher robustness to noise. Different from CR, the expectation over all directions mitigates the effect of regions with wrong class but small volume, which makes it a better measure of empirical performance. Nonetheless, to illustrate how this analysis can fall short, let's consider a linear classifier for binary classification. In this task, the goal is to find the hyperplane that best separates the two classes. The adversarial robustness of an input is easily computed as its distance to the hyperplane. On the other hand, the noise robustness of any sample is infinite regardless of how close it is to the decision boundary since the probability of correct and incorrect classification becomes 0.5 only in the infinite variance limit.

Given the previous points, adversarial and noise robustness can be interpreted as two extremes in a robustness spectrum, one that can be pessimistic and the other optimistic. Motivated by this, several approaches attempted to bridge them (Fawzi et al., 2016; Rice & Bair, 2021; Robey et al., 2022), with Randomized Smoothing (RS) (Cohen et al., 2019) being the most prominent. In RS, a deterministic classifier $f(x)$ that returns a 0-1 vector, with 1 in the correct class element, is turned into a probabilistic one $g(x)$, the smoothed classifier, that returns the probability of each class by injecting noise into the input. Then, by using the probabilities of the correct and largest incorrect classes, one can obtain the certified radius of this smoothed classifier, i.e., the radius of a ball around input x within which all points are correctly classified on average. Two properties of RS made it an attractive approach for robustness certification. First, the smoothing by the input noise acts in the same way as in the average-case robustness, mitigating the effect of close, but small, misclassification regions. Second, it is scalable to arbitrarily complex NNs: while most certification methods rely on information from the NN's architecture, such as its Lipschitz constant for Lipschitz-based CR (Cisse et al., 2017; Tsuzuku et al., 2018; Weng et al., 2018; Leino et al., 2021), or its activation functions as in convex relaxation-based CR (Weng et al., 2018; Wong & Kolter, 2018), RS depends only on the input noise distribution and on the method to estimate the classification probabilities.

In the work by Hayes (2020), he compiles and compares the different expressions of CR for randomized smoothing, where the one by Cohen et al. (2019) is shown to yield the tighter l_2-norm certification. However, a common factor to all CR when using Gaussian or Laplace distributions is that it is scaled by the variance parameter, resulting in vanishing CR for small variances. This vanishing effect was pointed out by Mohapatra et al. (2021), who also showed that the range of input variances that yields the largest certified radius results in a loss of information, which degrades the "clean" accuracy. These points suggest that an alternative smoothed radius certification approach, without the scaling by the input variance, might be useful for the small variance smoothing case.

The contributions of the present work are twofold. First, inspired by Smoothed Analysis (SA), originally proposed in the context of algorithmic complexity analysis, we provide a framework of robustness of classifiers that combines both worst- and average-case, the Smoothed Robustness Analysis (SRA). In SA, the goal is to obtain a more faithful description of empirical results as compared to worst- and average-cases, by obtaining the worst-case performance when subject to random perturbations. In their seminal work, Spielman & Teng (2004) proved through SA that the simplex algorithm for linear programming has a polynomial smoothed time complexity, despite having an exponential worst-case time complexity. Even though this analysis was developed in the context of algorithmic time complexity, it can be applied to other performance metrics, as long as a worst-case analysis can be carried after an averaging step. SA served as inspiration for previous neural networks research (Blum & Dunagan, 2002; Haghtalab et al., 2020) and more recently in classification robustness (Robey et al., 2022), but its connection to existing robustness analysis hasn't been made explicit. Hence, by applying this analysis to a classification metric under input perturbations, we provide an expression of SRA, from which randomized smoothing emerges as a special case.

As a second contribution, starting from the framework's requirements, we propose a margin loss-based smoothed robustness analysis for the case of Lipschitz-constrained NNs. For a NN with Lipschitz constant K, the margin, defined as the difference between the largest incorrect and the correct outputs, allows simple computation of the certified radius. Then, using the fact that the smoothed version of a K-Lipschitz function is also K-Lipschitz, we obtain the certified radius of the smoothed NN by employing its smoothed margin. This CR, different from the one obtained with RS, doesn't have the input noise variance as a scaling factor and therefore can certify a safe region even in the zero-limit of the variance.

To validate the smoothed margin as a robustness inducing loss for the case of Lipschitz NNs, we carried out experiments with the MNIST dataset on 1-Lipschitz Multilayer Perceptron (MLP). By measuring the robustness according to adversarial (l_2 and l_∞-norms), noisy (isotropic Gaussian) and progressive linear corruptions, we show that under the considered settings the smoothed margin yields higher robustness to both adversarial and random noises when compared to the (non-smoothed) margin and the randomized smoothing losses.

1.1 Related works

Bridging adversarial and noise robustness

The idea of adversarial and noise robustness as two extremes in a robustness spectrum has been explored in previous works with different approaches to bridge them. Randomized smoothing was first proposed by Lecuyer et al. (2019) within the realms of differential privacy as PixelDP, providing radius certification of ℓ_1 and l_2 balls of smoothed classifiers by employing the Laplace and Gaussian distributions as input noise, respectively. Cohen et al. (2019) proved a tight l_2 CR for the case of Gaussian input noise and coined the name randomized smoothing. The advantages of RS radius certification compared to other certification methods are the smoothing effect, which alleviates the effect of a low-volume sensitive input directions, yielding larger radii compared to the unsmoothed classifier, and that it scales to inputs of larger dimensions and arbitrary NN architectures. However, as pointed out by Mohapatra et al. (2021), in most image classification tasks the certified radius vanishes for small variances, as well as in the interval of maximal certified radius there's a classification accuracy drop due to information loss caused by larger variances.

While RS bridges worst- and average-cases by explicit use of one noise and one adversarial mechanism, other works proposed a spectrum that connects them. Fawzi et al. (2016) considered the case of the expected distance to the decision boundary over randomly sampled d-dimensional input sub-spaces, with $d \in [1, n_{in}]$ and n_{in} the total input dimension. In the extreme cases of $d = 1$ and $d = n_{in}$ one has the average- and worst-case, respectively. Rice & Bair (2021) proposed the l_q-norm of a *smooth loss* w.r.t. a measure $\mu(x)$ as a robustness metric. For the case of a uniform ball as the measure, when $q = \infty$ one has the largest adversarial loss within the ball, and when $q = 1$ one obtains the usual expected loss within the ball, with any q in the middle describing a combination of adversarial and noise robustness.

More recently, motivated by either the lack of interpretability or impracticality of existing attempts to bridge worst- and average-cases, Robey et al. (2022) proposed the *probabilistically robust* learning framework. In this framework, the goal is to interpolate between adversarial training, where for each training sample an adversarial example is used, and average training, where for each sample the average performance under some perturbation is used. For the interpolation, they use the misclassification tolerance parameter ρ, which controls the stringency of the worst loss inside a feasible region by setting the tolerance probability of maximal loss. In the case of the 01 loss with a uniform distribution in the input, if $\rho = 0$ one has the adversarial training. To see this, the loss for one sample is zero only if the probability of correct classification inside the uniform ball is 1, i.e., if there's no AE in the ball. When $\rho = 0.5$, the average-case robustness is obtained, since a training sample only contributes to the loss in case of the classification probability being less than 0.5.

Different from these works, where they propose a continuous with adversarial and noise robustness as extremes, our framework combines both by two different mechanisms. To account for noise robustness, a choice of input perturbation distribution is required, with a higher variance while still classifying correctly indicating greater robustness to noise. On the other hand, adversarial robustness requires the evaluation of a region in the input space, as is the case of l_p-balls, that doesn't contain any misclassification. In this case, the larger the region the more adversarially robust it is. Akin to complexity analysis of algorithms, which has worst-, average-case, and smoothed analyses, this framework contains their robustness analysis equivalents depending on the choice of distribution and region, thus we denote it *smoothed robustness analysis*. Here we stress that randomized smoothing is a special instance of SRA, more precisely when the classification loss is the 01 loss.

Certified radius and margin maximization

Previous works have attempted to use the certified radius as regularization to improve the NN robustness to adversarial attacks. In the case of knowing the NN's Lipschitz constant, Tsuzuku et al. (2018) proposed summing a bias term to the incorrect outputs before passing them to the softmax such that it correctly classifies only if the certified radius is larger than a predefined value. Croce et al. (2019) used the locally linear property of Rectified Linear Unit (ReLU) activation to show that the minimal distance to the polytope around a correctly classified input is either the exact distance to the decision boundary or a lower bound. They then proposed using this value as a regularizer for correctly classified samples in a way that the distance to the decision boundary is maximized.

Under the randomized smoothing approach, Zhen et al. (2021) proposed to use a closed-form approximation for the classification probability to compute the approximate certified radius, and use it as a regularizer to be maximized. Similar to Zhen et al. (2021), in the present work we want to maximize the certified radius, but differently we use the smoothed margin instead of the classification probability to obtain the CR and use it directly as the objective and not as a regularizer. We also implement the probability-based CR from Zhen et al. (2021) to compare the robustness improvements of each.

Lipschitz constrained NNs

Lipschitz-constrained NNs yield a simple certified radius that can be used for margin maximization, which is the motivation for its use in the present work, but it also owns other useful properties. Even though previously believed that they lacked expressiveness, i.e., arbitrary function approximation capacity, as compared to non-constrained ones, thanks to the work of Béthune et al. (2022) we now know that by correctly selecting the loss hyperparameters one can have arbitrarily complex decision boundaries for classifiers. They also show that 1-Lipschitz NNs can be *consistent* depending on the loss, i.e., in the infinite limit of training data it is guaranteed to converge to the Bayes optimal classifier.

Additionally, in the special case of the affine transformation operators (the weight matrices) being orthogonal, and therefore 1-Lipschitz w.r.t. to the l_2-norm, exploding and vanishing gradients are mitigated since orthogonal matrices are norm preserving. These kinds of matrices when paired with 1-Lipschitz element-wise activation functions, such as ReLU or tanh, lose expressiveness, but GroupSort, a non-element-wise activation function that sorts the pre-activations, has been proposed as an activation function that avoids this expressiveness loss (Anil et al., 2019).

Closed form output statistics estimation

In the experiments on smoothed certified radius maximization from the present work, the losses require estimates of the output means and variances, as well as symmetry and independence assumptions. The architectural choice of 1-Lipschitz constraint NN with orthogonal weights and ReLU activation allows these to be approximately achieved. The orthogonality allows the independence from the isotropic Gaussian input to be propagated through the layers, while the ReLU activation allows for simple computation of its mean and variance since under noise it approximately follows a rectified Normal distribution, which in turn can be used to calculate the mean and variance of the (approximately normal) pre-activations of the following layer due to the Central Limit Theorem (CLT).

Similar techniques, some of them accounting for more complex distributions and architectures, have been extensively used in the literature on uncertainty quantification of NNs and Bayesian NNs (Kingma et al., 2015; Bibi et al., 2018; Shekhovtsov & Flach, 2019; Wu et al., 2019; Haußmann et al., 2020; Tsiligkaridis, 2021). In the context of randomized smoothing, similar to the present work, Zhen et al. (2021) used the approximate normality of pre-activations, due to the CLT for wide-layered NNs, to compute the moments of ReLU activation on normal pre-activations. Differently, they don't constrain the weights to be orthogonal and attempt to approximate the covariance matrix of each layer. Even though widely assumed, to the best of our knowledge the CLT for independent but not identically distributed rectified Normal random variables has never been proved. In the appendix we present a proof and, knowing that it is indeed satisfied, we restrict our experiments to only the case of independent hidden units by using orthogonal weight matrices.

Expected margin

Different from our work, which uses the average margin of a classifier as a proxy for the distance to the decision boundary on the smoothed classifier, Kumar et al. (2020) used it in the randomized smoothing setting to quantify the radius certification confidence. However, here we point out that their interpretation of a positive average margin implying correct average classification relies on their definition of correct average classification as the average correct output being larger than the largest average incorrect output. This is guaranteed to be the case when the correct output is symmetrical or negatively skewed and the incorrect outputs are symmetrical or positively skewed, which is likely to happen for most NNs given their wide layers. But still, in the case of positively skewed correct and negatively skewed incorrect outputs, there is a possibility of the median of the largest incorrect being larger than the median of the correct output, even if the mean of the former is less than the mean of the latter. In this case, it would result in an average misclassification, since the larger incorrect median would imply more than 50% of the trials being misclassified. Additionally, to the best of our knowledge the present work is the first time the expected margin is used to quantify the CR of a smoothed classifier.

2 Smoothed Robustness Analysis: bridging adversarial and average robustness analyses

As previously discussed, several works attempted to bridge the worst- and average-case robustness of classifiers, with randomized smoothing being its most prominent. The certified radius obtained via randomized smoothing guarantees that any perturbation within the ball with such radius will not result in misclassification of the classifier under noise. In other words, it measures the worst-case robustness of an averaged, or *smoothed*, classifier. In this description of randomized smoothing, the two steps taken for its implementation are the same used in the Smoothed Analysis (SA) framework from the analysis of algorithms literature (Spielman & Teng, 2004; 2009).

In this section, we show that the similarities between classification robustness and algorithmic analysis go beyond the terminologies by presenting how the SA framework can be applied to robustness analysis in a framework we name Smoothed Robustness Analysis (SRA). We then move forward to show that under SRA one obtains adversarial, noise and randomized smoothing robustness as special cases, and finish the section connecting it to other robustness analyses that bridge worst- and average-case.

2.1 Algorithmic analysis: worst-case, average-case and smoothed

Analysis of algorithms is the branch of computer science that deals with algorithm comparison. Given a task, a performance metric, and two or more algorithms, the goal is to, under the same analysis framework, evaluate which has the best performance. Among the different frameworks, the most widely used is *worst-case* analysis. As the name says, it gives the worst possible performance of an algorithm given some constraints, such as the input size. Hence, if an algorithm has a good worst-case, it is guaranteed to never perform poorly and, intuitively should be preferable than one with worse worst-case. Two reasons for its success are that in many classical algorithms the worst-case is a common instance, as in database search when the key does not exist (Cormen, 2009, pp. 27), or that the number of steps taken is close to worst-case, as in dynamic programming (Roughgarden, 2021, pp. 3).

Despite its popularity, worst-case analysis is known to be too pessimistic in more complex tasks (Spielman & Teng, 2009; Roughgarden, 2021, pp. 3). In the case of numerical analysis, algorithms can deal with high dimensional or non-linear problems where the worst-case comprises instances that might be rare or non-existent in its practical use. To help analyse them in a more realistic way, average-case analysis has been one of the main approaches (Ritter, 2000). In average-case analysis, by assigning a measure to the instance space, one can weigh them and obtain the performance metric of interest (e.g., time complexity, approximation error, etc.) as a weighted average. Nevertheless, the biggest downside of average-case analysis is its strong dependence on the chosen distribution, which, to be tractable, usually requires simpler distributions not corresponding to the real instance distribution (Roughgarden, 2021, pp. 168).

Motivated by the shortcomings of worst- and average-case analyses, Spielman & Teng (2004) proposed Smoothed Analysis (SA). In SA, roughly speaking, one first considers some source of randomness in the algorithm, which yields an averaged ("smoothed") version of it, and then proceeds to apply a worst-case analysis in this smoothed algorithm. In the averaging stage, different from average-case analysis, the distribution has a small variance on the state under evaluation instead of trying to average over all possible states. Spielman & Teng (2009) argue that this perturbation should be small, as if mimicking sensing noise or random faults, and for being small, simpler distributions can be used without carrying biases about the real underlying distribution, as happens in the average-case. The worst-case step then depends on the algorithm in hand as well as the type of data, and the use of simpler distributions allows the application of methods from the usual worst-case analysis.

2.2 Smoothed robustness analysis

In the introduction we gave a high-level description of worst- and average-case analysis of robustness. To move forward to applying the smoothed analysis framework and connect them, we first need to formalize some concepts and requirements. For the aforementioned analyses from analysis of algorithms, one needs to pick the performance metric on which the analysis is applied. In the case of classification tasks, if the loss function $T(\boldsymbol{x})$ satisfies the following condition, then it is a suitable performance metric for the application of smoothed analysis in the robustness setting.

Condition (Average classification condition)**.** *There is some constant* $a \in \mathbb{R}$ *s.t.* $\mathbb{E}_{\boldsymbol{x}' \sim \rho(\boldsymbol{x},\boldsymbol{\theta})}[T(\boldsymbol{x}')] < a$ *implies in the probability of correct classification* $\mathcal{P}_c$ *of* $\boldsymbol{x}$ *under noise distributed as* $\rho(\boldsymbol{x},\boldsymbol{\theta})$ *being greater than* 0.5 *for all* $\boldsymbol{x} \in \mathbb{X}$. *This constant is the average classification condition constant.*

Remark. *When* $\rho(\boldsymbol{x},\boldsymbol{\theta})$ *is the Dirac measure, the average loss is simply the loss* $T(\boldsymbol{x})$. *Thus, if* $T(\boldsymbol{x}) < a$, *then* $\boldsymbol{x}$ *is correctly classified.*

It's straightforward to see that this condition is required for the average-case (noise) robustness analysis, since the robustness evaluation of a classifier w.r.t. some distribution $\rho(\boldsymbol{x},\boldsymbol{\theta})$ depends on the average correct classification under such noise. From this perspective, the larger the variance of the distribution while correctly classified, the more robust to noise it is. To see that this condition is required for the worst-case (adversarial) robustness analysis, let's assume that an oracle obtains the maximal value $T'(\boldsymbol{x})$ of the loss for all $\boldsymbol{x} \in \Omega(\boldsymbol{x})$, where $\Omega(\boldsymbol{x})$ is a subset of the input space containing the input $\boldsymbol{x}$. If this oracle returns $T'(\boldsymbol{x}) < a$, due to condition 2.2 we know that the region is safe to any adversarial perturbation.

Now in order to apply the SA framework, we combine the averaging step from the average-case with the maximum search step from the worst-case analysis. Through this, we arrive in the following definition of *smoothed robustness analysis of classifiers*:

Definition 1 (Smoothed robustness analysis of classifiers)**.** *Let* $\tilde{T}(\boldsymbol{x})$ *denote the maximal average loss within the region* $\Omega(\boldsymbol{x}) \subset \mathbb{X}$ *around input* $\boldsymbol{x}$ *over the distribution* $\rho(\tilde{\boldsymbol{x}}; \boldsymbol{\theta})$*, parametrized by* $\tilde{\boldsymbol{x}} \in \Omega(\boldsymbol{x})$ *and* $\boldsymbol{\theta}$*, i.e.,*

$$\tilde{T}(\boldsymbol{x}) = \max_{\tilde{\boldsymbol{x}} \in \Omega(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{x}' \sim \rho(\tilde{\boldsymbol{x}}; \boldsymbol{\theta})}[T(\boldsymbol{x}')]. \tag{1}$$

If $\tilde{T}(\boldsymbol{x}) < a$*, then there is no perturbation within* $\Omega(\boldsymbol{x})$ *that causes a misclassification on average. Such classifier we call SRA robust.*

Depending on the choices of $\Omega(\boldsymbol{x})$ and $\rho(\tilde{\boldsymbol{x}}, \boldsymbol{\theta})$ this expression yields the worst-, average-case, or a combination of both. If $\Omega(\boldsymbol{x}) = \boldsymbol{x}$, then the max operation is restricted to $\boldsymbol{x}$ itself, and we obtain the average-case. Similarly, if $\rho(\tilde{\boldsymbol{x}}; \boldsymbol{\theta}) = \delta(\tilde{\boldsymbol{x}})$, then the expectation operator results in the pointwise value of the loss with no smoothing effect, hence the worst-case. In figure 1 we illustrate the three different cases.

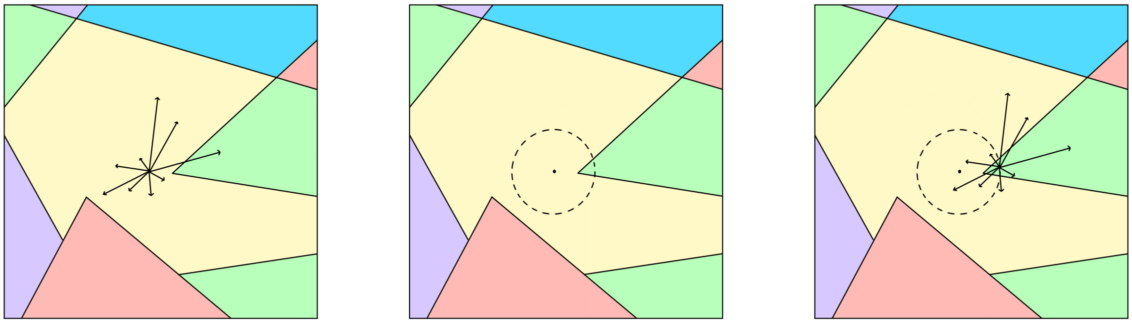

Figure 1: Illustrations of average-case (left), worst-case (center), and smoothed (right) robustness analyses, where different colors correspond to different classes in the input space according to the classifier and the central dot is the analysed input. For average-case, the input is perturbed in random directions. For worst-case, there is a region around the input that either crosses or not the decision boundary. For the smoothed RA, each point inside the region is perturbed randomly and either contains or not instances with incorrect average classification.

2.2.1 SRA with 01 loss

To illustrate the application of SRA we first consider the most canonical loss in machine learning, the 01 loss.

Definition 2 (01 loss)**.** *Given a classification score vector defined as the mapping from the input space* $\boldsymbol{o}(\boldsymbol{x}) : \mathbb{X} \mapsto \mathbb{R}^{|\mathbb{C}|}$*, where* $|\mathbb{C}|$ *is the number of classes, the 01 loss* $\mathcal{L}_{01}(\boldsymbol{x}, c)$ *is defined as*

$$\mathcal{L}_{01}(\boldsymbol{x}, c) = \begin{cases} 0, & \textit{if } o_c(\boldsymbol{x}) > o_m(\boldsymbol{x}) \\ 1, & \textit{otherwise,} \end{cases} \tag{2}$$

where c *is the correct class index and* m *the largest incorrect index, i.e.,* $m = \arg\max_{i \neq c} o_i(\boldsymbol{x})$.

We start with the average-case by setting the worst-case region $\Omega(\boldsymbol{x}) = \boldsymbol{x}$ and noting that $\mathbb{E}_{\boldsymbol{x}' \sim \rho(\boldsymbol{x}, \boldsymbol{\theta})}[\mathcal{L}_{01}(\boldsymbol{x}', c)] = \mathcal{P}_{i \neq c}(\boldsymbol{x}; \rho(\cdot)) = 1 - \mathcal{P}_c(\boldsymbol{x}; \rho(\cdot))$, i.e., the expected loss is the probability of incorrect classification $\mathcal{P}_{i \neq c}(\boldsymbol{x}; \rho(\cdot))$, which in turn is one minus the probability of correct classification $\mathcal{P}_c(\boldsymbol{x}; \rho(\cdot))$. Thus, if its value is less than 0.5, we have that the probability of misclassification is less than 0.5, and therefore correctly classified and robust under $\rho(\boldsymbol{x}, \boldsymbol{\theta})$. From the average-case, we see that the average classification condition constant for the 01 loss is $a = 0.5$.

Different from the average-case, even though the worst-case for $\mathcal{L}_{01}(\boldsymbol{x}, c)$ has a simple interpretation - if inside the region $\Omega(\boldsymbol{x})$ the maximal loss is 0, then it is adversarially robust in $\Omega(\boldsymbol{x})$ - its analysis is difficult due to its non-continuity in the decision boundary. Randomized Smoothing (RS) was proposed as a way of circumventing this problem by considering a proxy classifier that classifies the input subject to random noise, the *smoothed classifier*, in which case the 01 loss becomes the probability of misclassification. The general idea of RS is that depending on the choice of input noise one can use the classification probabilities to obtain the radius of a l_p-ball around an input $\boldsymbol{x}$ which is guaranteed to not contain any adversarial example under the smoothed classifier. In the case of isotropic Gaussian input noise with variance σ_x^2, RS yields the radius ϵ_{01,l_2} of an l_2-ball around the input that is guaranteed to be safe in this proxy classifier (Cohen et al., 2019),

$$\epsilon_{01,l_2}(\boldsymbol{x}; \mathcal{N}(0, \sigma_x^2 \boldsymbol{I})) = \frac{\sigma_x}{2}\left(\Phi^{-1}(\mathcal{P}_c) - \Phi^{-1}(\max_{i \neq c} \mathcal{P}_i)\right), \tag{3}$$

where $\mathcal{P}_c(\boldsymbol{x}; \sigma_x^2)$ is the probability of correct classification and $\Phi^{-1}(\cdot)$ is the inverse Cumulative Distribution Function (CDF) of the standard Normal distribution.

Given this example, here we stress two ways of applying SRA. The first is as a *robustness test* method by setting an arbitrary input region $\Omega(\boldsymbol{x})$ and distribution $\rho(\boldsymbol{x}, \boldsymbol{\theta})$, then verifying whether it satisfies the SRA condition 1. The second and more practical way is as *robustness certification.* By setting a distribution $\rho(\boldsymbol{x}, \boldsymbol{\theta})$ one then proceeds to obtain a safe region (e.g. l_p-ball) of the classifier smoothed with $\rho(\boldsymbol{x}, \boldsymbol{\theta})$. RS from Cohen et al. (2019) is a special case of the latter, and in terms of SRA the region within the l_2-ball guarantees that the classifier is SRA robust w.r.t. the 01 loss and isotropic Gaussian noise, and the radius of this ball we name Smoothed Certified Radius (SCR).

2.2.2 SRA with margin loss

One major drawback of RS is that the radius of the l_p-ball has the input noise variance as scaling factor, as in equation 3, which means that for small variances the certified region vanishes. Once we interpret RS as an SRA approach, by drawing a parallel with SA for analysis of algorithms - where the use of small variance distributions is recommended - one might ask whether the impossibility of certifying regions for small variances is problematic. Motivated by this, we propose the use of the margin loss in SRA.

Definition 3 (Margin loss)**.** *Given a classification score vector defined as the mapping from the input space* $\boldsymbol{o}(\boldsymbol{x}) : \mathbb{X} \mapsto \mathbb{R}^{|\mathbb{C}|}$, *where* $|\mathbb{C}|$ *is the number of classes, the margin loss* $\mathcal{L}_M(\boldsymbol{x}, c)$ *is defined as*

$$\mathcal{L}_M(\boldsymbol{x}, c) = o_m(\boldsymbol{x}) - o_c(\boldsymbol{x}), \tag{4}$$

where c is the correct class index and m the largest incorrect index, i.e., $m = \arg\max_{i \neq c} o_i(\boldsymbol{x})$.

The margin loss $\mathcal{L}_M(\boldsymbol{x}, c)$ of a L_p-Lipschitz classifier, i.e., the difference between the largest incorrect and correct values from the score vector, for being $\alpha_p L_p$-Lipschitz, gives the quickest method for radius certification of such classifier,

$$\epsilon_{M,l_p}(\boldsymbol{x}) = -\frac{\mathcal{L}_M(\boldsymbol{x}, c)}{\alpha_p L_p} \tag{5}$$

where α_p is a constant that depends on the choice of norm ($\alpha_p = \sqrt{2}$ and 2 for $p = 2$ and ∞, respectively) and L_p is the classifier's Lipschitz constant w.r.t. the l_p-norm. Applying SRA to this loss, first we need to obtain the average loss under some input noise $\rho(\boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the distribution's parameters, which we call Smoothed Margin (SM). In the case of symmetrically distributed outputs [1], we have that, if the SM is negative, then the probability of correct classification is greater than 0.5.[2] Hence, the margin loss satisfies the condition 2.2 with constant $a = 0$.

Now for the worst-case step of SRA, we use the fact that a smoothed K-Lipschitz scalar function by convolution with a probability density function is also K-Lipschitz. Then, we obtain a certified radius expression

---

[1] We found this to be an adequate assumption both empirically, as we show in appendix B.1, and theoretically, since independent rectified Gaussian random variables satisfy the CLT, as shown in appendix A.4

[2] The proof can be found on appendix A.1.

just like for the usual margin loss [3],

$$\epsilon_{SM,l_p}(\boldsymbol{x};\rho(\cdot)) = -\frac{\bar{\mathcal{L}}_M(\boldsymbol{x},c;\rho(\cdot))}{\alpha_p L_p}, \tag{6}$$

where $\bar{\mathcal{L}}_M(\boldsymbol{x},c;\rho(\cdot)) = \mathbb{E}_{\boldsymbol{x}\sim\rho(\cdot)}\left[\mathcal{L}_M(\boldsymbol{x},c)\right]$ is the SM.

Then, if $\epsilon_{SM,l_p}(\boldsymbol{x};\rho(\cdot)) > 0$, the input $\boldsymbol{x}$ is SRA robust w.r.t. the margin loss. As a final note, different from the Randomized Smoothing Certified Radius (RSCR), which vanishes in the zero-variance limit, the margin SCR converges to the usual margin CR. This can make the margin SCR a more suitable loss for SCR maximization training by avoiding the information loss caused by the larger variances required to obtain meaningful SCR under RS. According to our experiments on the MNIST dataset, this seems to be the case, where we found that the margin SCR loss improves both adversarial and noise robustness compared to the RSCR and the usual margin CR losses.

2.2.3 Relationship to other beyond worst- and average-case robustness analyses

We already mentioned how SRA connects to randomized smoothing in the case of the 01 loss . The work by Rice & Bair (2021) proposes a method to bridge these two cases based on the l_q-norm of a *smooth* loss $\ell(\boldsymbol{x},c)$ over a measure $\mu(\boldsymbol{x})$ that depends on input $\boldsymbol{x}$,

$$\|\ell(\boldsymbol{x},c)\|_{\mu(\boldsymbol{x}),q} = \mathbb{E}_{x'\sim\mu(\boldsymbol{x})}[|\ell(\boldsymbol{x}',c)|^q]^{\frac{1}{q}}. \tag{7}$$

Their method considers the usual definition of losses as non-negative scalar functions, so $|\ell(\boldsymbol{x}',c)| = \ell(\boldsymbol{x}',c)$. When $q\to\infty$ and $\mu(\boldsymbol{x})$ is a uniform ball, one obtains the worst-case, i.e., the maximal loss within the ball. When $q=1$, this expression gives the average loss over the measure $\mu(\boldsymbol{x})$. The connection of SRA with this method is that the averaging step from SRA generates a smooth function $\ell(\boldsymbol{x},c) = \mathbb{E}_{\tilde{\boldsymbol{x}}\sim\rho(\boldsymbol{x})}[\ell'(\tilde{\boldsymbol{x}},c)]$, $\ell'(\boldsymbol{x},c)$ being a loss not necessarily smooth, and then SRA (for non-negative losses) can be written in this l_q-norm approach as,

$$\|\ell(\boldsymbol{x},c)\|_{\Omega(\boldsymbol{x}),l_\infty} = \mathbb{E}_{x'\sim\Omega(\boldsymbol{x})}[\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\rho(\boldsymbol{x}')}[\ell'(\tilde{\boldsymbol{x}},c)]^\infty]^{\frac{1}{\infty}}. \tag{8}$$

The main difference of SRA and the method by Rice & Bair (2021) is that, while the former incorporates the worst- and average-case as two axis, one controlled by the size of the worst-case region $\Omega(\boldsymbol{x})\subset\mathbb{X}$ and the other by the variance of the smoothing distribution $\rho(\boldsymbol{x})$, the latter combines them as two extremes in a spectrum. Additionally, SRA characterizes the robustness of a classifier by stating whether it is “safe” under a choice of noise distribution and adversarial region given some loss. Then, robustness comparison is done by evaluating which classifier is safe for larger regions (e.g. larger certified radius) and input variances. The approach based on l_q-norm acts as a robustness metric, in which by choosing the norm and measure, in theory, a classifier with larger l_q-norm is more robust.

3 Methods

3.1 Smoothed loss approximations

The goal in our experiments is to compare SCR maximization losses in their improvement of classification robustness. The general form of a SCR loss is

$$\ell_{\cdot,l_2}\left(\boldsymbol{x},c,d,\sigma_x^2\right) = \mathrm{ReLU}(d-\epsilon_{\cdot,l_2}\left(\boldsymbol{x},c,\sigma_x^2\right)) \tag{9}$$

where $\epsilon_{\cdot,l_2}\left(\boldsymbol{x},c,\sigma_x^2\right)$ is the l_2-ball CR under $\cdot$ classification loss (e.g., "01" or "M" for 01 or margin losses, respectively) subject to smoothing of variance σ_x^2, and d is a hyperparameter that controls the maximal CR that contributes to the loss, i.e., if $\epsilon_{\cdot,l_2}\geq d$ the loss is zero.

---

[3] This expression is derived in appendix A.2

The baseline for our comparisons is the CR of the margin loss 5 of a K-Lipschitz classifier, shown in equation 10, since it is the zero-variance limit of the margin SCR, thus not requiring any output estimates, and also for being employed in other works for adversarial robustness improvement.

$$\ell_{M,l_2}\left(\boldsymbol{x}, c, d, \sigma_x^2 = 0\right) = \mathrm{ReLU}(d - \epsilon_{M,l_2}\left(\boldsymbol{x}, c, 0\right)) = \mathrm{ReLU}\left(d + \frac{\mathcal{L}_M(\boldsymbol{x}, c)}{\sqrt{2}K}\right) \tag{10}$$

For non-zero variances, the margin becomes the smoothed margin, and the SCR is given by 6. This expression has the same form as 5 since the margin loss of a K-Lipschitz classifier is $\sqrt{2}K$-Lipschitz, and also because a smoothed K-Lipschitz scalar function due to convolution with a probability density function is also K-Lipschitz (Appendix A.2). However, the computation of the smoothed margin isn't trivial, since it involves a multi-dimensional integral over an unknown distribution of outputs. To obtain a closed-form estimate, we make the assumption that the outputs follow a multi-dimensional Normal with mean vector $\boldsymbol{\mu}(\boldsymbol{o})$ and covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{o}) = \mathrm{diag}(\mathbf{var}(\boldsymbol{o}))$, where $\mathrm{diag}(\boldsymbol{x})$ denotes the diagonal matrix with the vector $\boldsymbol{x}$ in the diagonal elements and $\mathbf{var}(\boldsymbol{o})$ is the variance vector of the output vector $\boldsymbol{o}$.

For a multivariate normal output $\boldsymbol{o}(\boldsymbol{x})$ we have the smoothed margin,

$$\begin{aligned}\bar{\mathcal{L}}_M\left(\boldsymbol{x}, c, \sigma_x^2\right) &= \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\left[\max_{i\neq c} o_i\left(\boldsymbol{x}\right) - o_c\left(\boldsymbol{x}\right)\right] = \mathbb{E}_{\boldsymbol{o}\sim p(\boldsymbol{o})}\left[\max_{i\neq c} o_i\left(\boldsymbol{x}\right)\right] - \mu_c \\ &= \sum_{i\neq c}\int_{-\infty}^{\infty} o_i \rho\left(o_i\right)\prod_{j\neq i}\frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{o_i - \mu_j}{\sqrt{2}\sigma_j}\right)\right] do_i - \mu_c,\end{aligned} \tag{11}$$

where in the second line we took the expectation over the output instead of the input, and the indices are sorted in descending order, with 1 being the index of the largest mean among the incorrect outputs.

We approximate the product of CDF to just the CDF of the largest mean among the ones involved in the product, which gives the following expression,

$$\begin{aligned}\bar{\mathcal{L}}_M\left(\boldsymbol{x}, c, \sigma_x^2\right) \cong & \frac{\sigma_1}{\sqrt{2\pi\left({\sigma_1}^2 + {\sigma_2}^2\right)}}\exp\left(-\frac{\left(\mu_1 - \mu_2\right)^2}{2\pi\left({\sigma_1}^2 + {\sigma_2}^2\right)}\right) + \frac{\mu_1}{2}\left(1 + \mathrm{erf}\left(\frac{\mu_1 - \mu_2}{\sqrt{2\pi\left({\sigma_1}^2 + {\sigma_2}^2\right)}}\right)\right) \\ & \sum_{i=2}^{|C|-1}\frac{\sigma_i}{\sqrt{2\pi\left({\sigma_1}^2 + {\sigma_i}^2\right)}}\exp\left(-\frac{\left(\mu_1 - \mu_i\right)^2}{2\pi\left({\sigma_1}^2 + {\sigma_i}^2\right)}\right) + \frac{\mu_i}{2}\left(1 - \mathrm{erf}\left(\frac{\mu_1 - \mu_i}{\sqrt{2\pi\left({\sigma_1}^2 + {\sigma_i}^2\right)}}\right)\right) - \mu_c.\end{aligned}$$

Empirically, we found that using the terms involving the two largest means are enough to give a close approximation (Appendix B.2). Then, by leaving the two terms, we arrive at the following expression as the smoothed margin,

$$\bar{\mathcal{L}}_M\left(\boldsymbol{x}, c, \sigma_x^2\right) = \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2\pi}}\exp\left(-\frac{\left(\mu_1 - \mu_2\right)^2}{2\pi\left(\sigma_1^2 + \sigma_2^2\right)}\right) + \mu_2 + \frac{\mu_1 - \mu_2}{2}\left(1 + \mathrm{erf}\left(\frac{\mu_1 - \mu_2}{\sqrt{2\pi\left(\sigma_1^2 + \sigma_2^2\right)}}\right)\right) - \mu_c. \tag{12}$$

The margin SCR loss induces a SRA robust classifier w.r.t. the margin loss. To compare the robustness improvement given by it, we also implement a SCR loss-based on the 01 loss, i.e., a RSCR maximization loss. The RSCR depends on the probability of correct classification,

$$\mathcal{P}_c\left(\boldsymbol{x}, \sigma_x^2\right) = \mathcal{P}\left(o_c(\boldsymbol{x}) > o_i(\boldsymbol{x}), \forall i \neq c\right) = \int_{-\infty}^{\infty}\rho\left(o_c\right)\prod_{i\neq c}\frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{o_c - \mu_i}{\sqrt{2}\sigma_i}\right)\right] do_c, \tag{13}$$

which, similar to our approach for the smoothed margin estimate, Zhen et al. (2021) used the following expression as estimate by considering only the term of largest mean μ_i in the product,

$$\bar{\mathcal{P}}_c\left(\boldsymbol{x}, \sigma_x^2\right) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{\mu_c - \mu_1}{\sqrt{2(\sigma_c^2 + \sigma_1^2)}}\right)\right]. \tag{14}$$

Then, applying this expression for the probability of correct classification on the randomized smoothing CR from Cohen et al. (2019), and using $1 - \mathcal{P}_c$ instead of $\max\limits_{i \neq c} \mathcal{P}_i$, one obtains a lower bound on the RSCR

$$\begin{aligned}\epsilon_{01,l_2} &= \frac{\sigma_x}{2}\left(\Phi^{-1}(\mathcal{P}_c) - \Phi^{-1}(\max_{i\neq c}\mathcal{P}_i)\right) \geq \frac{\sigma_x}{2}\left(\Phi^{-1}(\mathcal{P}_c) - \Phi^{-1}(1-\mathcal{P}_c)\right) \\ &= \sigma_x \Phi^{-1}(\mathcal{P}_c) = \sigma_x \Phi^{-1}\left(\Phi\left(\frac{\mu_c - \mu_1}{\sqrt{\sigma_c^2 + \sigma_1^2}}\right)\right) = \sigma_x \frac{\mu_c - \mu_1}{\sqrt{\sigma_c^2 + \sigma_1^2}},\end{aligned}$$

finally yielding the RSCR loss (which for the remaining we'll refer as *Zhen loss*)

$$\ell_{01,l_2}\left(\boldsymbol{x}, c, d, \sigma_x^2\right) = \mathrm{ReLU}\left(d - \sigma_x \frac{\mu_c - \mu_1}{\sqrt{\sigma_c^2 + \sigma_1^2}}\right). \tag{15}$$

3.2 Weight matrix orthogonalization

In section 2, we argued that using the smoothed margin for certified radius maximization training could be beneficial, given that for small input noise variances, as recommended in the smoothed analysis framework, the radius doesn't vanish. To obtain a certified radius from the margin one needs to know the Lipschitz constant of the classifier, which can be done either by constraining it to some value during training or by its evaluation as required. In our experiments, we resort to constraining the Lipschitzness to 1 by employing ReLU activations with orthogonal weight matrices, i.e., $\boldsymbol{W}_i \boldsymbol{W}_i^{\mathrm{T}} = \boldsymbol{I}_{n_i}$, where $\boldsymbol{W}_i$ is the i-th layer weight matrix, $\boldsymbol{I}_{n_i}$ is the identity matrix of size n_i, and n_i is the number of units in layer i. The reason for this is that, because each weight vector is orthogonal to the others, for an isotropic Gaussian noise in the input the covariances within the same hidden layer are zero, which allows the application of the central limit theorem. This is required to obtain closed form estimates of the smoothed margin and probability of correct classification by assuming independent and normally distributed outputs.

Instead of using a regularization term to gradually induce the orthogonalization during the training, as is common practice in the Lipschitz constrained NN literature (Bansal et al., 2018), we adopted the Björck orthogonalization (BO) algorithm (Björck & Bowie, 1971) to guarantee an approximate orthogonalization of weight matrices for every training step. In BO there are two parameters, the order of the approximation p and the number of iterations q. By setting the order $p = 1$ and starting with a matrix $\boldsymbol{A}_0 = \boldsymbol{A}$ the algorithm computes an approximate orthogonal matrix iteratively as

$$\boldsymbol{A}_{k+1} = \boldsymbol{A}_k \left(\frac{3}{2} I - \frac{1}{2} \boldsymbol{A}_k^{\top} \boldsymbol{A}_k\right).$$

This sequence of operations, even though resulting in large overhead for more iterations, is amenable to backpropagation, where the entries of the 0-th matrix $\boldsymbol{A}$ are updated, and easily implemented.

For the implementation of BO we used the code from Anil et al. (2019) as base. However, different from their work, in which a progressive orthogonalization is acceptable, i.e., increasing the iteration number to better approximate an orthogonal matrix as the training converges, and similar to Berg et al. (2019), where they had to enforce orthogonality for their Normalizing flows approach, we start the training with 30 iterations and every 10 epochs the layerwise number of iterations q_l is decreased to the minimum that satisfies $\frac{\sum_{i\neq j} \|\tilde{\boldsymbol{W}}_{l,i} \cdot \tilde{\boldsymbol{W}}_{l,j}^{\top}\|}{n_l * (n_l - 1)} \leq \varepsilon$, i.e., the mean pairwise dot product between different rows of the weight matrix from l-th layer obtained after BO, $\tilde{\boldsymbol{W}}_l$, should be less than ε, n_l being the number of units in the l-th layer. We found that in general this mean isn't less than 10^{-9}, even for large number of iterations, so we set $\varepsilon = 10^{-8}$ in all experiments.

3.3 ReLU moment propagation

In the estimates of the smoothed margin and probability of correct classification we assumed independent normally distributed outputs. The orthogonal matrices are required to guarantee the independence of pre-activations, which for wide layers are approximately normal with mean $\mu_l = \boldsymbol{W}_l \bar{\boldsymbol{h}}_{l-1}$ and covariance matrix

$\mathbf{\Sigma}_l = \boldsymbol{W}_l \mathrm{diag}(\mathrm{Var}(\boldsymbol{h}_{l-1}))\boldsymbol{W}_l^\top$, where $\bar{\boldsymbol{h}}_{l-1}$ is the mean vector of the (l-1)-th hidden layer and $\mathrm{Var}(\boldsymbol{h}_{l-1})$ its variance vector. In the training with smoothed losses we implemented the moment propagation (Wu et al., 2019) in a MLP architecture with ReLU activation function.

Moment propagation consists of computing the mean vector μ_l and covariance matrix Σ_l of the pre-activations $\boldsymbol{s}_l$ of the l-th layer given by the previous hidden layer's mean activation $\bar{h}_{l-1}$ and covariance $\mathrm{cov}\, h_{l-1}$, which in turn are estimated using μ_{l-1} and Σ_{l-1}. Even though the ReLU activation with Normally distributed pre-activation has simple closed forms for its mean and variance[4], the covariance doesn't have a closed form expression thus requiring an approximation (Wu et al., 2019). Fortunately, by using row-orthogonal weight matrices, which doesn't create any covariance from the isotropic input noise, we don't need to worry about the covariances and need to compute just the mean and variance of ReLU layers.

3.4 Robustness metrics

Adversarial robustness

To assess the worst-case robustness, i.e., robustness to adversarial attacks, we implemented the Projected Gradient Descent (PGD) attack algorithm for the l_2 and l_∞ cases as in Madry et al. (2018). PGD iteratively uses the gradient w.r.t. the input to maximize the loss on a correctly classified input with the target class different from the correct. The loss used for the attacks is the 6-th loss evaluated by Carlini & Wagner (2017),

$$L(\boldsymbol{x}, c) = \mathrm{ReLU}\left(\max_{i \neq t} o_i(\boldsymbol{x}) - o_t(\boldsymbol{x})\right),$$

where $o_i(\boldsymbol{x})$ is the i-th output and t is the index of the attack's target class. This loss has been extensively used due to its reportedly strong attacks to the extent of being named C&W loss, and, when t is the index of the largest incorrect output m and the input $\boldsymbol{x}$ is correctly classified, in case of a K-Lipschitz classifier it can be interpreted as the margin CR loss (equation 10) with $d = 0$ scaled by $\sqrt{2}K$,

$$L(\boldsymbol{x}, y) = \mathrm{ReLU}(o_c(\boldsymbol{x}) - o_m(\boldsymbol{x})) = \sqrt{2}K\, \mathrm{ReLU}\left(\frac{\mathcal{L}_M(\boldsymbol{x}, m)}{\sqrt{2}K}\right),$$

where we set the target class m in the margin CR loss' $\mathcal{L}_M(\boldsymbol{x}, \cdot)$ second argument. In our experiments, we picked the target class to be the one with largest incorrect output, and found that a step size of $0.08 \cdot \varepsilon$ with 200 iterations was enough to guarantee convergence, where ε is the radius of adversarial attack.

For each trained NN the following routine was carried to measure its adversarial robustness:

1. Measure the adversarial error for radius $\varepsilon = 0$ (i.e., the clean error), $\mathrm{AdvErr}(\mathbb{X}_{test}, \varepsilon = 0)$, of all test samples in the set $\mathbb{X}_{test}$, and remove all incorrectly classified samples from $\mathbb{X}_{test}$
2. Increase the radius ε by $\Delta\varepsilon = 0.025$ for l_∞ and 0.32 for l_2 attacks
3. For each input sample $\boldsymbol{x} \in \mathbb{X}_{test}$, compute PGD attacks for five uniformly distributed random initializations inside the ball of radius ε around input $\boldsymbol{x}$
4. If at least one attack is successful for an individual sample $\boldsymbol{x}$, remove it from $\mathbb{X}_{test}$, and store the cumulative number of removed samples divided by the initial number of samples as the adversarial error $\mathrm{AdvErr}(\mathbb{X}_{test}, \varepsilon)$ for radius ε
5. Repeat from step 2 until $\mathbb{X}_{test} = \emptyset$

To quantify the overall robustness for each attack case, we propose the use of the Average Successful Radius (ASR). The $\mathrm{AdvErr}(\mathbb{X}_{test}, \varepsilon)$ curve as a function of ε ranges from the clean error to 1 and, due to the large

---

[4] We derive both in the appendix A.3.

number of test samples (10000), can be interpreted as the CDF of the minimal radii required for successful attacks. Then, using the expression for the expected value $\mathbb{E}[x] = \sum x \, \frac{\Delta \mathrm{CDF}(x)}{\Delta x} \Delta x$, we have

$$\mathrm{ASR}(\mathbb{X}_{test}) = \sum_{j=0} (j + 0.5)\, \Delta\varepsilon \, \left(\mathrm{AdvErr}(\mathbb{X}_{test}, (j+1) \cdot \Delta\varepsilon) - \mathrm{AdvErr}(\mathbb{X}_{test}, j \cdot \Delta\varepsilon)\right),$$

where the summation is over the integer j and $\mathrm{AdvErr}(\mathbb{X}_{test}, j \cdot \Delta\varepsilon)$ is the adversarial error for radius $j \cdot \Delta\varepsilon$.

Noise robustness

To evaluate the average-case robustness, i.e., robustness to noise, we measured how the performance is gradually impaired by increasing the standard deviation of an additive isotropic Gaussian noise. For each trained NN the following routine was carried to measure its noise robustness:

1. Measure the noise error for zero variance (i.e., the clean error) $\mathrm{NoiseErr}(\mathbb{X}_{test}, \sigma_x = 0)$ of all test samples $\mathbb{X}_{test}$
2. Increase the input standard deviation σ_x by $\Delta\sigma_x = 0.32$
3. For each input, sample five noise vectors δ from $\mathcal{N}(0, \sigma_x^2 \boldsymbol{I}_{n_x})$, where $\boldsymbol{I}_{n_x}$ is the identity matrix with n_x diagonal elements, and n_x is the input dimension
4. Calculate the noise error for each of the noise samples and store its average in $\mathrm{NoiseErr}(\mathbb{X}_{test}, \sigma_x)$
5. Repeat from step 2 until the average noise error is less than 0.87 ($\frac{|C|-1}{|C|}$ minus a small cutoff value, in this case 0.03)

To quantify the robustness to noise, we propose the Average Misclassification Standard deviation (AMS), where we define misclassification standard deviation of an input $\boldsymbol{x}$ the minimal σ that causes this input to be misclassified on average. While for the ASR with PGD few attack attempts allows us to find a successful one, for the noise robustness we require an exponential number of samples to obtain with high confidence the minimal σ that causes misclassification the larger this value gets, which can make it too costly to compute and exclude the samples individually like we did for ASR. However, similar to ASR, since we have a large number of samples, the noise error curve acts as a proxy to the CDF of the minimal misclassification σ allowing us to use it to estimate the average misclassification σ.

Because the interval of noise error varies from around $2 \sim 5\%$, for zero noise, to 90% in the infinite limit ($\frac{|C|-1}{|C|}$, where $|C|$ is the number of classes, and equals to 10 in our case), we normalized these values to keep the same error for zero noise but increase the 90% to 100% so it becomes a proper measure from which the average can be obtained.

Then, we have the normalized noise error $\mathrm{NoiseErr}'(\mathbb{X}_{test}, \sigma_{x,j})$ with $\sigma_{x,j} = j \cdot \Delta\sigma_x$,

$$\mathrm{NoiseErr}'(\mathbb{X}_{test}, \sigma_{x,j}) = \frac{\mathrm{NoiseErr}(\mathbb{X}_{test}, \sigma_{x,j})}{(1 - \alpha_j) + \alpha_j \mathrm{NoiseErr}(\mathbb{X}_{test}, \sigma_{x,j})}$$

where α_j is a $[0, 1]$ mapping, being 0 for $j = 0$ and 1 for $j \to \infty$,

$$\alpha_j = 1 - \frac{\max_k \mathrm{NoiseErr}(\mathbb{X}_{test}, k \cdot \Delta\sigma_x) - \mathrm{NoiseErr}(\mathbb{X}_{test}, j \cdot \Delta\sigma_x)}{\max_k \mathrm{NoiseErr}(\mathbb{X}_{test}, k \cdot \Delta\sigma_x) - \mathrm{NoiseErr}(\mathbb{X}_{test}, 0)}.$$

The meaning of this expression is that for $\sigma_x = 0$ the error remains the same, while the maximal error becomes 1 for $\sigma_x \gg 0$.

Then, using this normalized noise error we can calculate the AMS as

$$\mathrm{AMS} = \sum_{j=0} (j+0.5)\ \Delta\sigma_x\ \left(\mathrm{NoiseErr}'\left(\mathbb{X}_{test}, (j+1)\cdot\Delta\sigma_x\right) - \mathrm{NoiseErr}'(\mathbb{X}_{test}, j\cdot\Delta\sigma_x)\right).$$

The values of $\Delta\varepsilon$ and $\Delta\sigma_x$ were picked in such a way that it takes around 10 of these increments to completely degrade the performance.

Linear corruption robustness

The previous two metrics give information about both worst- and average-case robustness. However, they don't tell how robust a classifier is in real-world scenarios, or how it compares to human subjects. Motivated by this, we implemented the linear image corruption method proposed by Jang et al. (2021). In their work, they were interested on how robust are Deep NNs when compared to humans in the Imagenet dataset with 10 classes.

In their method, given an image $\boldsymbol{x} \in [0,255]^{n_x}$ and a "gray" noisy image $\boldsymbol{n}$, i.e., all pixels valued with 255/2 plus a small additive noise $\delta \sim \mathcal{N}(0, 6/255)$, the final image to be presented is a linear combination

$$\boldsymbol{I} = \gamma\boldsymbol{x} + (1-\gamma)\boldsymbol{n}$$

where the parameter $\gamma \in [0,1]$ can be interpreted as the signal-to-signal-plus-noise ratio, which when equals to 1 there's pure signal, and when 0 there's pure noise. Then, by presenting images with different values of γ in 0.05 increments, they measured the classification performance drop in both human and NN classifiers.

With this method, not only we can compare the different trained NNs, but also have a qualitative idea of how far from human robustness they are, since one of the results from Jang et al. (2021) is that human subjects in general reached a 50% accuracy drop around the interval of $\gamma \in [0.2, 0.3]$, with average close to 0.25.

3.5 Experimental settings

Loss functions

The goal of our experiments is the proof-of-concept of the Margin Smoothed Certified Radius (MSCR) loss as a suitable loss for robustness improvement by comparing it with randomized smoothing CR (Zhen) loss and Margin Certified Radius (MCR) loss, the latter being the σ_x^2 zero-limit of the margin SCR, as described in section 3.1. For the sake of completeness, we carried out additional trainings with losses that we display in table 1. Additional to the MCR loss, in the Lipschitz constrained NN literature (Anil et al., 2019; Li et al., 2019b) another margin maximization loss is the Multiclass Hinge (MH) $\ell_{\mathrm{MH}}(\boldsymbol{x}, c, d) = \sum_{i\neq c}\max(0, d - (o_c(\boldsymbol{x}) - o_i(\boldsymbol{x})))$, where one tries to maximize the difference not only from the largest incorrect class but to all incorrect. Because both have been used for margin maximization and robustness improvement, we also implement MH loss to compare their robustness improvements. Among the used losses, the Softmax Cross-Entropy (SCE) $\ell_{\mathrm{SCE}}(\boldsymbol{x}, c) = -\log\frac{\exp o_c(\boldsymbol{x})}{\sum_i \exp o_i(\boldsymbol{x})}$ is the only that doesn't enforce margin maximization nor acts on a smoothed classifier, and was used as reference since it is the most employed loss for classifiers. The Smoothed Classifier Softmax Cross-Entropy (SC-SCE) $\ell_{\text{SC-SCE}}(\boldsymbol{x}, c) = -\log\frac{\exp \mu_c(\boldsymbol{x})}{\sum_i \exp \mu_i(\boldsymbol{x})}$ is the SCE using the mean of the output layer $\mu(\bar{\boldsymbol{x}}) = \mathbb{E}_{\boldsymbol{x}'\sim\rho(\boldsymbol{x})}[\boldsymbol{o}(\boldsymbol{x}')]$ to calculate the softmax.

Dataset

For all experiments we employed the MNIST handwritten digits dataset (LeCun & Cortes, 2010), which consists of 60000 training and 10000 test 28x28 8-bit images, labeled from 0 to 9 with the number contained in the respective image. The only pre-processing was the scaling of each pixel from the range 0-255 to 0-1 by dividing all image elements by 255.

Unless otherwise stated, all measurements and results presented were evaluated on the test data.

Table 1: Losses used in this work. *Margin-based* are the ones that explicitly tries to maximize the margin. *Smoothed* are the ones that use a smoothed classifier.

	Loss					
	MSCR	**Zhen**	**MCR**	**MH**	**SCE**	**SC-SCE**
Margin-based	✓	✓	✓	✓	✗	✗
Smoothed	✓	✓	✗	✗	✗	✓

Loss hyperparameters search

The margin-based losses have a hyperparameter d that sets the maximal certified radius of a sample that counts for the loss (i.e., if $\epsilon_{\cdot,l_2} \geq d$ then $\ell_{\cdot,l_2} = 0$), thus controlling the stringency of the margin maximization. For losses of smoothed classifiers, the hyperparameter σ_x^2 is the variance of the input Gaussian noise and controls the smoothing intensity. For each loss we carried a grid search over the hyperparameters, training three NNs for each configuration, so that we could find the most robust configurations with similar clean accuracies in order to compare them. The details of hyperparameters and training are in appendix C.

4 Proof-of-concept: smoothed margin

In this section, we present the proof-of-concept of the MSCR loss as a suitable loss for improving both the worst- (adversarial) and average-case (noise) robustness of Lipschitz constrained NN. First, we show that the worst- and average-case robustness are controlled by, respectively, the hyperparameters d and σ_x^2. Then, by picking the hyperparameters with best l_∞ ASR among the pairs with clean accuracy within the range 98.5 ± 0.3 and 95.5 ± 0.3 of the MSCR, Zhen and MCR losses, we compare their robustness curves. We close the section by plotting the robustness curves for all losses described in section 3.5 for the case 95.5 ± 0.3.

4.1 Hyperparameter search

Figure 2 shows the test data clean accuracy, l_∞ ASR (worst-case robustness) and Gaussian noise AMS (average-case robustness) for different combinations of hyperparameters d and σ_x^2.

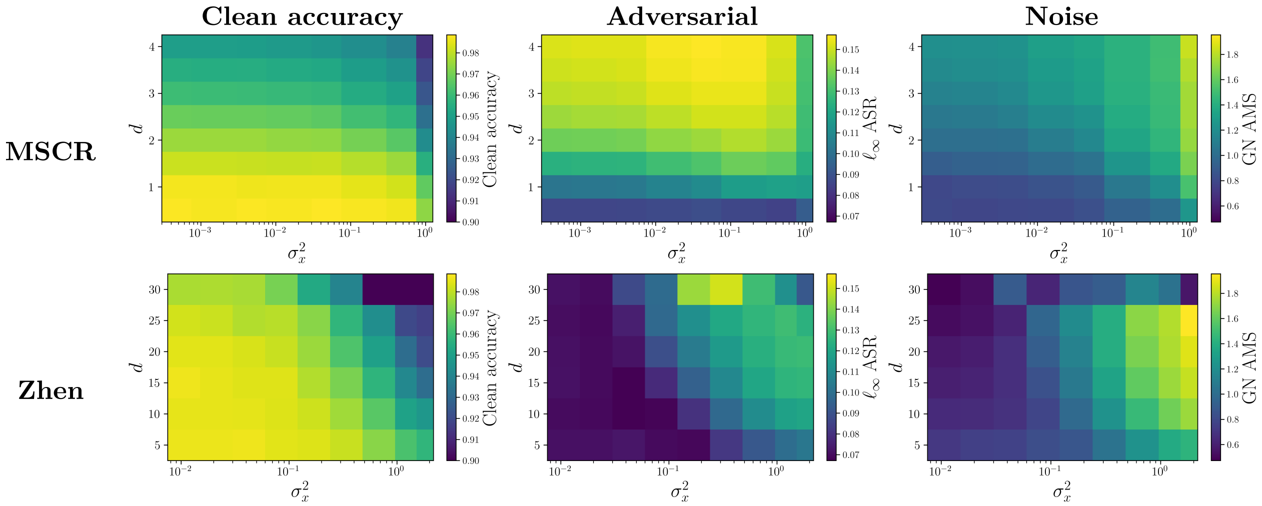

Figure 2: Heatmap of clean accuracy, l_∞ ASR, and Gaussian noise AMS for MSCR and Zhen loss. For each column, the colormaps share the maximum and minimum from both plots.

The first thing to notice from these plots is the well documented generalization-robustness trade-off. For both losses, we can see that hyperparameter pairs with lower clean accuracy (darker areas in first column) are inversely correlated with higher robustness (lighter areas in second and third columns). Second, we point that the two parameters contribute to the robustness in complementary ways: while a larger d in general improves the adversarial robustness, by controlling the worst-case distance to the decision boundary, a larger σ_x^2 induces a higher robustness to noise, since a classifier trained in this case learns to correctly classify on average even for stronger random perturbations.

4.2 Robustness comparison

Since a highly robust NN has smaller clean accuracy, and vice-versa, to compare the robustness improvement we restrict the parameter pairs with clean accuracy within a range $\text{Acc} \pm \Delta$ and select the ones with highest l_∞ ASR, the kind of attack of highest threat. In figures 3 and 4 we show the robustness curves, from the robustness metrics described in section 3.4, for the conditions 98.5 ± 0.3 (i.e., non-clean accuracy degradation case) and 95.5 ± 0.3 (i.e., clean accuracy degradation case), respectively.

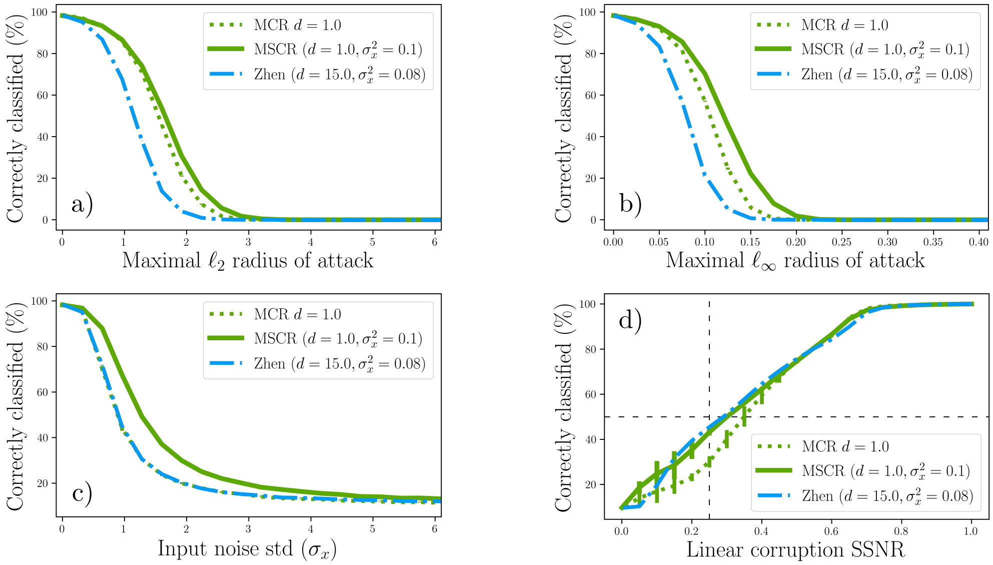

Figure 3: Robustness comparison between MCR, MSCR and Zhen losses for the highest l_∞ robustness within 98.5 ± 0.3 (%) clean accuracy. The vertical axis is the accuracy, horizontal axis is the attack intensity for (a) PGD l_2 adversarial attack; b) PGD l_∞ adversarial attack; c) Gaussian noise; d) Image linear corruption. Dashed horizontal and vertical lines in d) indicate the approximate 50% performance drop point in humans from Jang et al. (2021).

In figure 3, we see that the MSCR loss achieves the highest adversarial and noise robustness, and a similar linear corruption robustness to Zhen loss. Interestingly, while the usual margin CR loss achieves higher adversarial robustness compared to Zhen loss, it has a similar noise robustness and worse linear corruption robustness. The main takeaways here are, first, how even the non-smooth margin (MCR) loss yields a more adversarially robust loss compared to the Zhen loss. Second, not only the smoothing of the margin significantly improves its noise robustness, as one would expect, but also slightly improves its adversarial robustness. Third, even though the Zhen loss showed lower adversarial and noise robustness compared to the MSCR, it showed a slightly better image linear corruption robustness, close to what we would expect human subjects to achieve. These differences demonstrate that the evaluation of a classifier's robustness isn't a trivial subject, with no single metric that rules out a classifier as the more robust one.

Now from figure 4, different from the previous case, we observe that all losses have more similar robustness, with an overall slight improvement of robustness for the MSCR loss, which only the l_2 robustness being virtually the same as the one from the MCR. Particularly, the Zhen loss showed a significant improvement

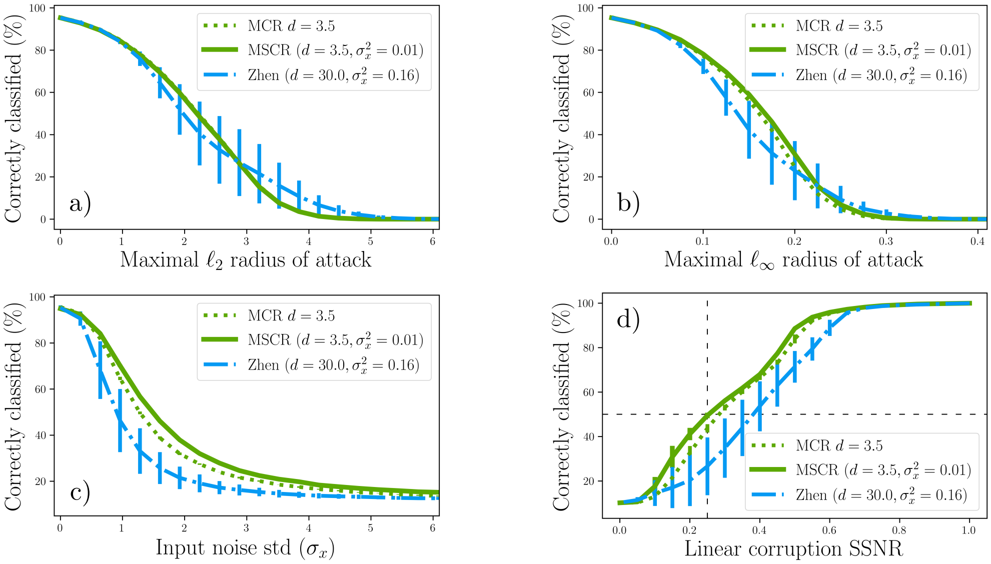

Figure 4: Robustness comparison between MCR, MSCR and Zhen losses for the highest l_∞ robustness within 95.5 ± 0.3 (%) clean accuracy. The vertical axis is the accuracy, horizontal axis is the attack intensity for (a) PGD l_2 adversarial; b) PGD l_∞ adversarial; c) Gaussian noise; d) Image linear corruption. Dashed horizontal and vertical lines in d) indicate the approximate 50% performance drop point in humans from Jang et al. (2021).

in its adversarial robustness, sometimes even surpassing the others. However, we found its training to be more unstable, as shown by the error bars over three trainings. For the sake of completeness, we show in figure 5 the curves obtained for the other losses introduced in section 3.5 as the continuous lines.

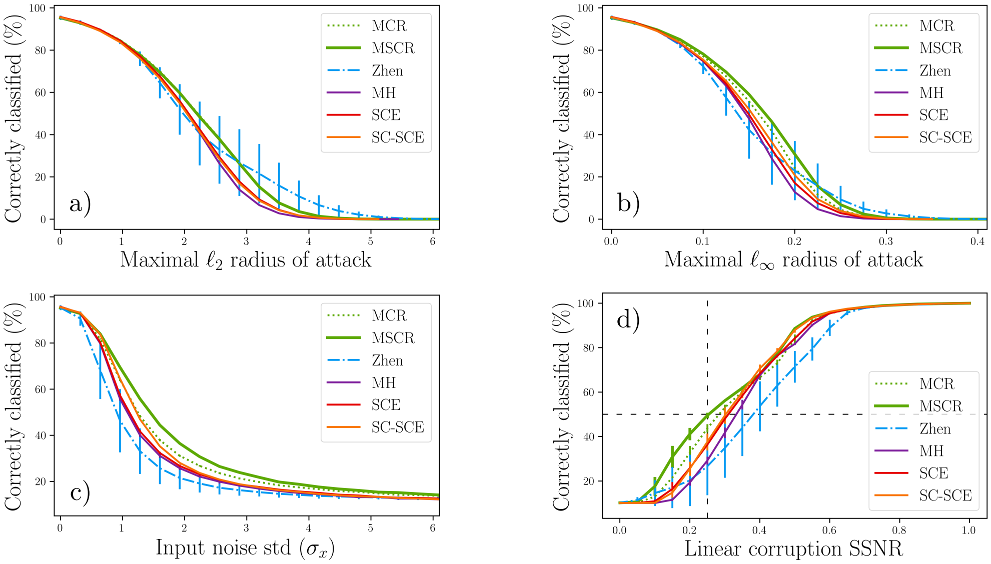

Figure 5: Robustness comparison between all losses described in section 3.5 for the highest l_∞ robustness within 95.5 ± 0.3 (%) clean accuracy. The vertical axis is the accuracy, horizontal axis is the attack intensity for (a) PGD l_2 adversarial; b) PGD l_∞ adversarial; c) Gaussian noise; d) Image linear corruption. Dashed horizontal and vertical lines in d) indicate the approximate 50% performance drop point in humans from Jang et al. (2021).

The robustness obtained with these other losses (given similar clean accuracy) were for all cases smaller than or similar to the ones obtained with MCR (dotted green) or MSCR (dashed green) losses. Curiously, we found that the MC hinge loss, widely used in the robustness literature, yields a performance similar to the SCE, which doesn't enforce any kind of robustness, and much smaller than the MCR loss. The SC-SCE loss is one that has been used as a simple alternative to probability-based losses (as in randomized smoothing), but we found that, at least under the 1-Lipschitz constraint, there's a big performance drop in clean accuracy, not making it a suitable choice.

5 Discussion

Motivated by recent attempts of bridging worst- and average-case robustness analyses of classifiers, in the present work we employed the well established smoothed analysis framework from analysis of algorithms to introduce the Smoothed Robustness Analysis (SRA) framework. In section 2.2.1 we showed that randomized smoothing is a special case of SRA when the loss is the 01 and the input region is an l_p-ball, where the norm p depends on the choice of input distribution. Considering the argument from smoothed analysis literature that the variance of the noise distribution should be a small fraction of the input value, which can't be done for randomized smoothing since the certified radius scales with the variance, in section 2.2.2 we proposed the margin loss as a more suitable loss for SRA. To validate the margin loss as suitable for SRA we used an approximation of the smoothed margin for smoothed certified radius maximization training, which led to improvements in adversarial and noise robustness compared to the usual margin and the randomized smoothing certified radius losses. Here we discuss interpretations and limitations of the present work as well as outline possible future directions.

5.1 Closed-form estimates of $\mathcal{P}_c$ and SM

In the appendix B.2 we show that the estimates of $\mathcal{P}_c$ and SM agree well for trained NNs. However, as the variance increases, approximating the product of CDFs by the one with largest mean in the integrals 11 and 13 becomes less precise resulting in a bigger difference between the numerical integration and approximation values. This happens because as the variance increases the difference between the correct and incorrect outputs means decreases. Still, the approximation of the SM is more accurate than of the $\mathcal{P}_c$ since the former incorporates the statistics of the largest and second largest incorrect outputs, while the latter depends only on the statistic of the largest incorrect output.

We stress here that these estimates are not lower bounds, being a strict upper-bound for $\mathcal{P}_c$ and just an approximation for SM. However, since SCR maximization losses induce an increase in the difference between the correct and largest incorrect output means, hence yielding better estimates as the training proceeds, even though these estimates aren't suitable for SRA comparison between trained NNs they might be for SRA maximization training. Thus we leave for future works using strict lower-bounds or sample estimates with statistical guarantees for SRA comparison.

5.2 Multi-class hinge losses

The smoothed margin from this work uses the definition of margin as the difference between the largest incorrect and correct outputs. For providing a simple proxy of the distance to the decision boundary, it has been widely used in the context of margin maximization as a regularization (Tsuzuku et al., 2018; Zhang et al., 2021). Other works proposed the use of a summation over all the pairwise differences between the incorrect and correct outputs as the loss for robustness improvement of Lipschitz constrained NNs (Anil et al., 2019; Li et al., 2019b). Since in our results the NNs trained with the latter (MH) showed lower robustness than the ones trained with the former (MCR) (fig. 5), here we discuss the reasons for this difference and possible future directions related to the use of other more suitable losses.

Both margin losses are part of a bigger group of large margin losses from the Support Vector Machine (SVM) literature referred as multi-class hinge losses. Vapnik (1998) and Weston & Watkins (1999) independently proposed extending the binary hinge loss to the multi-class case by taking the sum of pair-wise differences between correct and incorrect outputs, which the MH loss is a special case. Before them, multi-class SVMs were treated as $|C| \cdot (1 - |C|)$ binary SVMs between all the outputs, where $|C|$ is the number of classes, which resulted in larger computational overhead. Crammer & Singer (2001) then proposed a simplified multi-class hinge loss that depends only on the difference between correct class output and the largest incorrect, i.e., the margin loss in our work.

To better understand the statistical learning properties of these losses, Zhang (2004) showed that these two losses aren't Fisher consistent, i.e., in the infinite sample limit the index of largest output isn't necessarily the index of the correct posterior classification probability of the Bayes classifier. More precisely, the MH loss is consistent if the probability of the correct output is greater than 0.5 or if the probability of the second largest is less than $\frac{1}{|C|}$, and the margin loss is consistent only if the probability of correct classification is greater than 0.5. This is related to our result that as long as the smoothed margin is negative the probability of correct classification is greater than 0.5 but undetermined otherwise.

Since this consistency is in general a desirable feature, Liu (2007) proposed using a modification of the margin loss to make it consistent by truncating the margin loss so that it stops increasing at some value. The truncated margin loss can be written as the difference between the usual margin loss and a shifted margin loss $\ell_{TM}(\boldsymbol{x}, c, d) = \ell_M(\boldsymbol{x}, c, d) - \ell_M(\boldsymbol{x}, c, -d) = \mathrm{ReLU}(d + (o_m(\boldsymbol{x}) - o_c(\boldsymbol{x})) - \mathrm{ReLU}(-d + (o_m(\boldsymbol{x}) - o_c(\boldsymbol{x}))$, and it has a maximum of $2d$ when the margin for input $\boldsymbol{x}$ is $o_m(\boldsymbol{x}) - o_c(\boldsymbol{x}) \geq d$. Wu & Liu (2007) used this loss to improve the accuracy of multi-class SVMs by ignoring sample outliers, hence a possible direction is to obtain an approximation of the smoothed truncated margin loss, similar to the one from the present work, and verify whether it helps in the robustness-accuracy trade-off by ignoring few samples that might contribute heavily to the loss.

Here we add that different from Dogan et al. (2016), who showed empirically in several datasets that the MH loss, in the case of SVM, has better accuracy compared to other hinge losses, we found the margin (MCR) loss to perform better. Therefore examining potential optimization issues of the MH loss as well as more thorough experiments is needed to confirm the differences observed in our results.

5.3 Robustness-accuracy trade-off

In section 4.1 we showed that by tuning the hyperparameters d and σ_x^2 we can achieve different degrees of adversarial and noise accuracy, respectively, at the cost of clean accuracy. Even though this robustness-accuracy trade-off has been widely demonstrated (e.g. Su et al., 2018; Cohen et al., 2019; Li et al., 2019a), to the best of our knowledge, only Laugros et al. (2019) showed evidence of adversarial and noise robustness corresponding to two different kind of robustness, i.e., a model robust to AEs isn't necessarily robust to random perturbations, and vice-versa. By using losses with parameters that account explicitly for these two robustness, and plotting different robustness metrics as a colormap against them, we provide additional evidence to Laugros et al. (2019) findings.

We also point that, while early works defended that this trade-off is unavoidable (Tsipras et al., 2019; Zhang et al., 2019), recent accounts favor the idea that it is a consequence of model limitations (Stutz et al., 2019; Olfat & Aswani, 2020; Yang et al., 2020; Leino et al., 2021; Li et al., 2022). More specifically, recent works argue that enforcing a tight global Lipschitz constant (Richardson & Weiss, 2021) and proper loss hyperparameter selection (Béthune et al., 2022) in Lipschitz constrained NNs can suffice to avoid this trade-off. Given the 1-Lipschitzness of the NNs in the present work, and that methods for exact calculation of Lipschitz constants of feed-forward ReLU NNs are known (Jordan & Dimakis, 2020; Bhowmick et al., 2021), we leave for a future work the computation of the constants of the trained NNs to verify whether they correlate with the robustness-accuracy gap.

As a last observation, despite intensive efforts, we still don't have a definitive solution for the problem of adversarial robustness. From the NNs side, recent works show evidences that top-down information about context and expectation improves general classification robustness (Huang et al., 2020; Alamia et al., 2023), indicating that it might play a central role in achieving the robustness observed in biological brains. From the neuroscience side, a recent work by Guo et al. (2022) shows that robust NNs show sensitivity to adversarial attacks similar to the ones measured from primate brains, hypothesizing that an error-correction mechanism might be involved. Thus, future works that deal with this trade-off should take in consideration this possible limitation of feed-forward NNs.

5.4 Limitations and other future directions

Given the nature of this work as a first exposition of what we named smoothed robustness analysis, the experimental part was limited only to 1-Lipschitz MLP with ReLU activations under isotropic Gaussian noise, as well as dealing exclusively with l_2-norm radius certification. That said, similar to the randomized smoothing literature, future works with margin SRA should consider other architectures and noise distributions and estimation methods of the smoothed margin that don't require the assumptions of symmetry and independence of outputs. Another limitation is the use of only the MNIST dataset. Therefore, using other dataset with more complex structure, such as cifar-10, is something we leave for future works.

As a final note, since smoothed analysis is a framework originally proposed to better capture the complexity of algorithms observed in practice, another possible direction is to test it as a NN complexity measure. Intuitively, when comparing multiple classifiers under the same loss given same input region $\Omega(\boldsymbol{x})$ and distribution $\rho(\boldsymbol{x}; \boldsymbol{\theta})$, the one with smallest average (over test samples) SRA has a decision boundary that better separates unseen data even under small noise. This might be related to the notion of compression from information bottleneck (Shwartz-Ziv & Tishby, 2017), in which NNs that correctly classify inputs with more noise better compresses the relevant information from these inputs, resulting in lower model complexity.

References

Andrea Alamia, Milad Mozafari, Bhavin Choksi, and Rufin VanRullen. On the role of feedback in image recognition under noise and adversarial attacks: A predictive coding perspective. *Neural Networks*, 157: 280–287, 2023.

Brendon G. Anderson and Somayeh Sojoudi. Data-Driven Certification of Neural Networks With Random Input Noise. *IEEE Transactions on Control of Network Systems*, 10(1):249–260, 2023.

Cem Anil, James Lucas, and Roger Grosse. Sorting out Lipschitz function approximation. *Proceedings of the 36th International Conference on Machine Learning,*, 2019.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can We Gain More from Orthogonality Regularizations in Training Deep CNNs? *32nd Conference on Neural Information Processing Systems*, 2018.

Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. Sylvester Normalizing Flows for Variational Inference. In *Proceedings of the 34th Conference in Uncertainty in Artificial Intelligence*, 2019.

Aritra Bhowmick, Meenakshi D'Souza, and G. Srinivasa Raghavan. LipBaB: Computing Exact Lipschitz Constant of ReLU Networks. In Igor Farkaš, Paolo Masulli, Sebastian Otte, and Stefan Wermter (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2021*, pp. 151–162, 2021.

Adel Bibi, Modar Alfadly, and Bernard Ghanem. Analytic Expressions for Probabilistic Moments of PL-DNN with Gaussian Input. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9099–9107, 2018.

Patrick Billingsley. *Probability and measure*. Wiley series in probability and mathematical statistics. Wiley, New York, 3rd ed edition, 1995.

Å. Björck and C. Bowie. An Iterative Algorithm for Computing the Best Estimate of an Orthogonal Matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

Avrim Blum and John Dunagan. Smoothed Analysis of the Perceptron Algorithm for Linear Programming. In *Proceedings of the 13th annual ACM-SIAM symposium on Discrete algorithms*, pp. 10, 2002.

Louis Béthune, Thibaut Boissin, Mathieu Serrurier, Franck Mamalet, Corentin Friedrich, and Alberto González-Sanz. Pay attention to your loss: understanding misconceptions about 1-Lipschitz neural networks. *36th Conference on Neural Information Processing Systems*, 2022.

Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification. In *Annual Computer Security Applications Conference*, 2019.

Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy*, pp. 39–57, 2017.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval Networks: Improving Robustness to Adversarial Examples. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified Adversarial Robustness via Randomized Smoothing. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Thomas H. Cormen (ed.). *Introduction to algorithms*. MIT Press, Cambridge, Mass, 3rd ed edition, 2009.

Nicolas Couellan. Probabilistic robustness estimates for feed-forward neural networks. *Neural Networks*, 142:138–147, 2021.

Koby Crammer and Yoram Singer. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. *Journal of Machine Learning Research*, 2001.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable Robustness of ReLU networks via Maximization of Linear Regions. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

Urun Dogan, Tobias Glasmachers, and Christian Igel. A Unified View on Multi-class Support Vector Classification. *Journal of Machine Learning Research*, 2016.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. *30th Conference on Neural Information Processing Systems*, 2016.

Jean-Yves Franceschi, Alhussein Fawzi, and Omar Fawzi. Robustness of classifiers to uniform $\ell\_p$ and Gaussian noise. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, 2018.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial Spheres, September 2018. URL `http://arxiv.org/abs/1801.02774`. arXiv:1801.02774 [cs].

Chong Guo, Michael J Lee, Guillaume Leclerc, Joel Dapello, Yug Rao, Aleksander Madry, and James J DiCarlo. Adversarially trained neural representations may already be as robust as corresponding biological neural representations. *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Nika Haghtalab, Tim Roughgarden, and Abhishek Shetty. Smoothed Analysis of Online and Differentially Private Learning. *34th Conference on Neural Information Processing Systems*, 2020.

Manuel Haußmann, Fred A Hamprecht, and Melih Kandemir. Sampling-Free Variational Inference of Bayesian Neural Networks by Variance Backpropagation. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, 2020.

Jamie Hayes. Extensions and limitations of randomized smoothing for robustness guarantees. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3413–3421, 2020.

Yujia Huang, James Gornet, Sihui Dai, Zhiding Yu, Tan Nguyen, Doris Y Tsao, and Anima Anandkumar. Neural Networks with Recurrent Generative Feedback. *34th Conference on Neural Information Processing Systems*, 2020.

Hojin Jang, Devin McCormack, and Frank Tong. Noise-trained deep neural networks effectively predict human vision and its neural responses to challenging images. *PLOS Biology*, 19(12):e3001418, 2021.

Matt Jordan and Alexandros G. Dimakis. Exactly Computing the Local Lipschitz Constant of ReLU Networks. *34th Conference on Neural Information Processing Systems*, 2020.

Marc Khoury and Dylan Hadfield-Menell. On the Geometry of Adversarial Examples, December 2018. URL `http://arxiv.org/abs/1811.00525`. arXiv:1811.00525 [cs, stat].

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *5th International Conference on Learning Representations*, 2017.

Durk P Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. *28th Advances in Neural Information Processing Systems*, 2015.

Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Certifying Confidence via Randomized Smoothing. *34th Conference on Neural Information Processing Systems*, pp. 13, 2020.

Alfred Laugros, Alice Caplier, and Matthieu Ospici. Are Adversarial Robustness and Common Perturbation Robustness Independent Attributes ? In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified Robustness to Adversarial Examples with Differential Privacy. In *2019 IEEE Symposium on Security and Privacy*, pp. 656–672, 2019.

Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-Robust Neural Networks. *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified Adversarial Robustness with Additive Noise. *33rd Conference on Neural Information Processing Systems*, 2019a.

Binghui Li, Jikai Jin, Han Zhong, John E Hopcroft, and Liwei Wang. Why Robust Generalization in Deep Learning is Difficult: Perspective of Expressive Power. *36th Conference on Neural Information Processing Systems*, 2022.

Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger Grosse, and Jörn-Henrik Jacobsen. Preventing Gradient Attenuation in Lipschitz Constrained Convolutional Networks. *33rd Conference on Neural Information Processing Systems*, 2019b.

Yufeng Liu. Fisher Consistency of Multicategory Support Vector Machines. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 2007.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. *5th International Conference on Learning Representations*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. *6th International Conference on Learning Representations*, 2018.

Jeet Mohapatra, Ching-Yun Ko, Tsui-Wei, Weng, Sijia Liu, Pin-Yu Chen, and Luca Daniel. Hidden Cost of Randomized Smoothing. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 2021.

Matt Olfat and Anil Aswani. Average Margin Regularization for Classifiers. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3194–3199, 2020.

Leslie Rice and Anna Bair. Robustness between the worst and average case. *35th Conference on Neural Information Processing Systems*, pp. 12, 2021.

Eitan Richardson and Yair Weiss. A Bayes-optimal view on adversarial examples. *Journal of Machine Learning Research*, 2021.

Klaus Ritter. *Average-case analysis of numerical problems*. Number 1733 in Lecture notes in mathematics. Springer, Berlin; New York, 2000.

Alexander Robey, Luiz F. O. Chamon, George J. Pappas, and Hamed Hassani. Probabilistically Robust Learning: Balancing Average- and Worst-case Performance. *Proceedings of the 39 th International Conference on Machine Learning*, 2022.

Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, Cambridge, 2021.

Alexander Shekhovtsov and Boris Flach. Feed-forward propagation in probabilistic neural networks with categorical and max layers. *7th International Conference on Learning Representations*, pp. 21, 2019.

Lin Shi, Teyi Liao, and Jianfeng He. Defending Adversarial Attacks against DNN Image Classification Models by a Noise-Fusion Method. *Electronics*, 11(12):1814, 2022.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the Black Box of Deep Neural Networks via Information, April 2017. URL `http://arxiv.org/abs/1703.00810`. arXiv:1703.00810 [cs].

Daniel A. Spielman and Shang-Hua Teng. Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time. *Journal of the ACM*, 2004.

Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52(10):76–84, 2009. ISSN 0001-0782, 1557-7317. doi: 10.1145/1562764.1562785.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling Adversarial Robustness and Generalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6969–6980, 2019.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is Robustness the Cost of Accuracy? – A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, volume 11216, pp. 644–661, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, February 2014. URL `http://arxiv.org/abs/1312.6199`. arXiv:1312.6199 [cs].

Karim Tit and Teddy Furon. Efficient Statistical Assessment of Neural Network Corruption Robustness. *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, pp. 11, 2021.

Theodoros Tsiligkaridis. Information Aware max-norm Dirichlet networks for predictive uncertainty estimation. *Neural Networks*, 135:105–114, 2021.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy. *Proceedings of the 37th International Conference on Machine Learning*, 2019.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks. *32nd Conference on Neural Information Processing Systems*, 2018.

Vladimir Naumovich Vapnik. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.

Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S. Dhillon, and Luca Daniel. Towards Fast Computation of Certified Robustness for ReLU Networks. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Tsui-Wei Weng, Pin-Yu Chen, Lam M Nguyen, Mark S Squillante, Akhilan Boopathy, Ivan Oseledets, and Luca Daniel. PROVEN: Verifying Robustness of Neural Networks with a Probabilistic Approach. *Proceedings of the 36th International Conference on Machine Learning*, pp. 10, 2019.

Jason Weston and Christopher Watkins. Support Vector Machines for Multi-Class Pattern Recognition. In *The European Symposium on Artificial Neural Networks*, 1999.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, José Miguel Hernández-Lobato, and Alexander L. Gaunt. Deterministic Variational Inference for Robust Bayesian Neural Networks. *7th International Conference on Learning Representations*, 2019.

Yichao Wu and Yufeng Liu. Robust Truncated Hinge Loss Support Vector Machines. *Journal of the American Statistical Association*, 102(479):974– 983, 2007.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A Closer Look at Accuracy vs. Robustness. *34th Conference on Neural Information Processing Systems*, 2020.

Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards Certifying Linf Robustness using Neural Networks with Linf-dist Neurons. *Proceedings of the 38th International Conference on Machine Learning*, pp. 12, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Tong Zhang. Statistical Analysis of Some Multi-Category Large Margin Classification Methods. *Journal of Machine Learning Research*, 2004.

Xingjian Zhen, Rudrasis Chakraborty, and Vikas Singh. Simpler Certified Radius Maximization by Propagating Covariances. *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, pp. 10, 2021.

A Proofs and derivations

A.1 Relation of smoothed margin and classification probability

Proposition 1. *Let the output layer $\boldsymbol{o}(\boldsymbol{x})$ of a classifier be independent and symmetrically distributed, i.e., the i-th dimension $\boldsymbol{o}_i(\boldsymbol{x})$ is symmetrically distributed with zero covariance with the other outputs $\boldsymbol{o}_{j\neq i}(\boldsymbol{x})$. If $\mathbb{E}_{\boldsymbol{x}'\sim\rho(\boldsymbol{x},\boldsymbol{\theta})}[\mathcal{L}_M(\boldsymbol{x}',c)] < 0$, then the probability of correct classification $\mathcal{P}_c(\boldsymbol{x};\rho(\cdot)) > 0.5$.*

Proof. We have the following inequality for the smoothed margin $\bar{\mathcal{L}}_M(\boldsymbol{x},c;\rho(\cdot))$ of independent and symmetrically distributed outputs,

$$\bar{\mathcal{L}}_M(\boldsymbol{x},c;\rho(\cdot)) = \mathbb{E}_{\boldsymbol{x}'\sim\rho(\boldsymbol{x},\boldsymbol{\theta})}[\max_{i\neq c} o_i(\boldsymbol{x}')] - \mu_c > \text{med}[\max_{i\neq c} o_i(\boldsymbol{x})] - \mu_c = \text{med}[\max_{i\neq c} o_i(\boldsymbol{x})] - \text{med}[o_c(\boldsymbol{x})],$$

where μ_c is the mean correct output, med$[\cdot]$ refers to the median, the inequality step holds in the case of symmetrically distributed outputs since the distribution of $\max\limits_{i\neq c} o_i(\boldsymbol{x})$ is positively skewed, which means that $\mathbb{E}_{\boldsymbol{x}'\sim\rho(\boldsymbol{x},\boldsymbol{\theta})}[\max\limits_{i\neq c} o_i(\boldsymbol{x}')] > \text{med}[\max\limits_{i\neq c} o_i(\boldsymbol{x}')]$, and $\mu_c = \text{med}[o_c(\boldsymbol{x})]$ since $o_c(\boldsymbol{x})$ is symmetrically distributed.

If the difference between the two medians is negative, then the probability of $o_c(\boldsymbol{x}) > \max\limits_{i\neq c} o_i(\boldsymbol{x})$ is greater than 0.5. Therefore, if $\bar{\mathcal{L}}_M(\boldsymbol{x}, c; \rho(\cdot)) < 0$, the probability of correct classification is greater than 0.5.

□

A.2 Smoothed radius certification with margin function

Here we derive a certified radius of a smoothed K-Lipschitz classifier based on the margin function. First we show that the margin of such classifier is $\sqrt{2}K$-Lipschitz, then we use the result that the smoothed form of a scalar K-Lipschitz function is also K-Lipschitz to arrive at the certified radius of a smoothed classifier $\epsilon_{l_2}(\boldsymbol{x}; \rho(\cdot)) = -\frac{\bar{\mathcal{L}}_M(\boldsymbol{x},c;\rho(\cdot))}{\sqrt{2}K}$.

Proposition 2. *The margin function $\mathcal{L}_M(\boldsymbol{x}, c)$ of a K-Lipschitz classifier with fixed c is $\sqrt{2}K$-Lipschitz.*

Proof. To prove this proposition we first need the definition of margin function 3. From the definition of Lipschitz continuity, we need to find an upper bound on $|\mathcal{L}_M(\boldsymbol{x}_1, c) - \mathcal{L}_M(\boldsymbol{x}_2, c)|\ \forall\ \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{X}$ that is proportional to $\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2$.

Because the classifier is K-Lipschitz, the following inequality holds,

$$\begin{aligned} K^2 \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 \geq \|\boldsymbol{o}(\boldsymbol{x}_1) - \boldsymbol{o}(\boldsymbol{x}_2)\|_2^2 &= \sum_i (o_i(\boldsymbol{x}_1) - o_i(\boldsymbol{x}_2))^2 \\ &\geq (o_c(\boldsymbol{x}_1) - o_c(\boldsymbol{x}_2))^2 + (o_{m1}(\boldsymbol{x}_1) - o_{m1}(\boldsymbol{x}_2))^2 \\ &\geq (o_c(\boldsymbol{x}_1) - o_c(\boldsymbol{x}_2))^2 + (o_{m1}(\boldsymbol{x}_1) - o_{m2}(\boldsymbol{x}_2))^2, \end{aligned}$$

where $mj = \arg\max\limits_{i\neq c} o_i(\boldsymbol{x}_j)$. Note that $m1$ is the index of maximal output for $\boldsymbol{x}_1$ but not necessarily for $\boldsymbol{x}_2$, which means that $o_{m1}(\boldsymbol{x}_2) \leq o_{m2}(\boldsymbol{x}_2)$, yielding the last inequality. The last term can be seen as the product of the squared norms of the vectors $\boldsymbol{v} = [o_{m1}(\boldsymbol{x}_1) - o_{m2}(\boldsymbol{x}_2), o_c(\boldsymbol{x}_1) - o_c(\boldsymbol{x}_2)]$ and $\boldsymbol{u} = \frac{\sqrt{2}}{2}[1, -1]$, which from Cauchy-Schwartz inequality yields

$$\begin{aligned} (o_c(\boldsymbol{x}_1) - o_c(\boldsymbol{x}_2))^2 + (o_{m1}(\boldsymbol{x}_1) - o_{m2}(\boldsymbol{x}_2))^2 &\geq \left(\begin{pmatrix} o_{m1}(\boldsymbol{x}_1) - o_{m2}(\boldsymbol{x}_2) \\ o_c(\boldsymbol{x}_1) - o_c(\boldsymbol{x}_2) \end{pmatrix} \cdot \begin{pmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{pmatrix} \right)^2 \\ &= \frac{1}{2}\left((o_{m1}(\boldsymbol{x}_1) - o_c(\boldsymbol{x}_1)) - (o_{m2}(\boldsymbol{x}_2) - o_c(\boldsymbol{x}_2))\right)^2 \\ &= \frac{1}{2}\left|\mathcal{L}_M(\boldsymbol{x}_1, c) - \mathcal{L}_M(\boldsymbol{x}_2, c)\right|^2. \end{aligned}$$

Therefore,

$$|\mathcal{L}_M(\boldsymbol{x}_1, c) - \mathcal{L}_M(\boldsymbol{x}_2, c)| \leq \sqrt{2}K \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2.$$

□

Corollary 2.1 (Margin-based certified radius)**.** *For a correctly classified input $\boldsymbol{x}$, $\varepsilon = -\frac{\mathcal{L}_M(\boldsymbol{x},c)}{\sqrt{2}K}$ is a lower bound on the norm of the minimal perturbation $\boldsymbol{\delta} \in \mathbb{X}$ required to make $\mathcal{L}_M(\boldsymbol{x} + \boldsymbol{\delta}) = 0$. In other words, the classifier correctly classifies $\boldsymbol{x}$ to all perturbations $\boldsymbol{\delta} \in \mathbb{X}$ s.t. $\|\boldsymbol{\delta}\|_2 < \varepsilon$.*

To see this, we first note that $\mathcal{L}_M(\boldsymbol{x}, c) < 0$ since $\boldsymbol{x}$ is correctly classified. Second, a minimal successful adversarial perturbation $\boldsymbol{\delta}$ implies $\mathcal{L}_M(\boldsymbol{x} + \boldsymbol{\delta}, c) = 0$. By substituting both in the Lipschitz inequality we have $|\mathcal{L}_M(\boldsymbol{x}, c)| = -\mathcal{L}_M(\boldsymbol{x}, c) \leq \sqrt{2}K \left\|\boldsymbol{\delta}\right\|_2$, which implies that the minimal successful perturbation has norm $\left\|\boldsymbol{\delta}\right\|_2 \geq -\frac{\mathcal{L}_M(\boldsymbol{x},c)}{\sqrt{2}K}$. Conversely, any perturbation less than that lower bound is guaranteed to not result in misclassification.

To proceed, we need to prove that the smoothed form of a K-Lipschitz function due to the application of random noise is also K-Lipschitz. For this we use the following definition of smoothed function:

Definition 4. *The smoothed form of a function $F : \mathbb{X} \to \mathbb{Y}$ by convolution with a probability density function $\rho(\boldsymbol{x}; \boldsymbol{\theta})$ parametrized by $\boldsymbol{\theta}$ is*

$$\hat{F}\left(\boldsymbol{x}; \boldsymbol{\theta}\right) := \int_{\mathbb{R}^{n_x}} d\boldsymbol{t} \rho(\boldsymbol{t}; \boldsymbol{\theta}) F(\boldsymbol{x} - \boldsymbol{t}).$$

Then we have the following proposition about the lipschitzness of a smoothed scalar function:

Proposition 3. *Let $F(\boldsymbol{x})$ be a K_p-Lipschitz scalar function of $\boldsymbol{x} \in \mathbb{X}$. Then its smoothed form $\hat{F}(\boldsymbol{x}; \boldsymbol{\theta})$ is K_p-Lipschitz, where the subscript p stands for the l_p-norm.*

Proof.

$$\begin{aligned}
\left|\hat{F}\left(\boldsymbol{x}_1; \boldsymbol{\theta}\right) - \hat{F}\left(\boldsymbol{x}_2; \boldsymbol{\theta}\right)\right|_2 &= \left|\int_{\mathbb{R}^{n_x}} d\boldsymbol{t} \rho(\boldsymbol{t}; \boldsymbol{\theta}) (F(\boldsymbol{x}_1 + \boldsymbol{t}) - F(\boldsymbol{x}_2 + \boldsymbol{t}))\right| \\
&\leq \int_{\mathbb{R}^{n_x}} d\boldsymbol{t} \rho(\boldsymbol{t}; \boldsymbol{\theta}) \left|(F(\boldsymbol{x}_1 + \boldsymbol{t}) - F(\boldsymbol{x}_2 + \boldsymbol{t}))\right| \\
&\leq \int_{\mathbb{R}^{n_x}} d\boldsymbol{t} \rho(\boldsymbol{t}; \boldsymbol{\theta}) K_p \left\|\boldsymbol{x}_1 - \boldsymbol{x}_2\right\| = K_p \left\|\boldsymbol{x}_1 - \boldsymbol{x}_2\right\|,
\end{aligned}$$

where the first inequality is because absolute value of an integral is always less than or equal to the integral of an absolute value, and the second inequality is because $F(x)$ is K_p-Lipschitz. ☐

Finally, using the previous results, we obtain a certified radius of a smoothed classifier based on its smoothed margin.

Theorem 4. *Let the* smoothed margin $\hat{\mathcal{L}}_M\left(\boldsymbol{x}, c, \sigma_x^2\right)$ *be defined according to definitions 3 and 4. If $\hat{\mathcal{L}}_M\left(\boldsymbol{x}, c, \sigma_x^2\right) < 0$, then $\hat{\mathcal{L}}_M\left(\boldsymbol{x} + \delta, c, \sigma_x^2\right) < 0 \;\forall\; \delta \in \mathbb{R}^{n_x}$ s.t. $\|\delta\| < -\frac{\hat{\mathcal{L}}_M\left(\boldsymbol{x},c,\sigma_x^2\right)}{\sqrt{2}K}$, where K is the Lipschitz constant of the classifier.*

Proof. From Proposition 2 we know that the Lipschitz constant of the margin function is $\sqrt{2}K$. Using Proposition 3, we have that the smoothed margin is also $\sqrt{2}K$-Lipschitz. Then, using the same argument from corollary 2.1, we have that the minimal perturbation $\delta \in \mathbb{X}$ s.t. $\hat{\mathcal{L}}_M\left(\boldsymbol{x} + \delta, c, \sigma_x^2\right) = 0$ has norm $\left\|\boldsymbol{\delta}\right\|_2 \geq -\frac{\hat{\mathcal{L}}_M\left(\boldsymbol{x},c,\sigma_x^2\right)}{\sqrt{2}K}$. Conversely, any perturbation less than that lower bound is guaranteed to not result in misclassification on the smoothed classifier. ☐

A.3 ReLU moments

Let the pre-activation s of a ReLU unit $h = \mathrm{ReLU}(s)$ follow a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Then, h will follow a rectified Gaussian distribution $\mathcal{N}_{\mathrm{R}}(\mu, \sigma^2)$ with density function

$$\rho(h; \mu, \sigma^2) = \Phi\left(\frac{-\mu}{\sigma}\right)\delta(h) + \frac{\exp(-\frac{(h-\mu)^2}{2\sigma^2})}{\sqrt{2\pi}\sigma}\,\mathrm{H}(h)$$

where $\Phi\left(x\right) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$ is the cumulative distribution function of the standard normal distribution; $\delta(x)$ is the Dirac delta function, and $\mathrm{H}(x)$ is the Heaviside step function. The expected value and variance of $h \sim \mathcal{N}_{\mathrm{R}}(\mu, \sigma^2)$ are denoted as follows:

Expected value of rectified Gaussian distributed variable

The expected value $\bar{h}$ of a unit $h \sim \mathcal{N}_{\mathrm{R}}(\mu, \sigma^2)$ is

$$\bar{h} = \mathbb{E}_{h\sim\mathcal{N}_{\mathrm{R}}(\mu,\sigma^2)}[h] = \int_{-\infty}^{\infty} h\,\rho(h;\mu,\sigma^2)\;dh = \int_{0}^{\infty} h\frac{\exp(-\frac{(h-\mu)^2}{2\sigma^2})}{\sqrt{2\pi}\sigma}\;dh$$

Substituting $\frac{h-\mu}{\sigma} = z$,

$$\begin{aligned}\bar{h} = \int_{-\frac{\mu}{\sigma}}^{\infty} (\sigma z + \mu)\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\;dz &= -\sigma\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\Bigg|_{-\frac{\mu}{\sigma}}^{\infty} + \;\mu\int_{-\frac{\mu}{\sigma}}^{\infty}\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\;dz \\ &= \sigma\frac{\exp(-\frac{\mu^2}{2\sigma^2})}{\sqrt{2\pi}} + \frac{\mu}{2}\left[1+\mathrm{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right)\right],\end{aligned} \tag{16}$$

where the second integral is the complementary cumulative distribution function $1 - \Phi\left(\frac{-\mu}{\sigma}\right)$.

Variance of rectified Gaussian distributed variable

For the variance $\mathrm{Var}(h) = \mathbb{E}_{h\sim\mathcal{N}_{\mathrm{R}}(\mu,\sigma^2)}[h^2] - \bar{h}^2$, we need the second moment $\mathbb{E}_{h\sim\mathcal{N}_{\mathrm{R}}(\mu,\sigma^2)}[h^2]$,

$$\begin{aligned}\mathbb{E}_{h\sim\mathcal{N}_{\mathrm{R}}(\mu,\sigma^2)}[h^2] &= \int_{-\frac{\mu}{\sigma}}^{\infty} (\sigma z + \mu)^2\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\;dz \\ &= \int_{-\frac{\mu}{\sigma}}^{\infty} (\sigma^2 z^2 + 2\mu\sigma z + \mu^2)\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\;dz \\ &= \sigma^2\int_{-\frac{\mu}{\sigma}}^{\infty} z^2\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\;dz + 2\mu\sigma\frac{\exp(-\frac{\mu^2}{2\sigma^2})}{\sqrt{2\pi}} + \frac{\mu^2}{2}\left[1+\mathrm{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right)\right],\end{aligned}$$

obtained substituting $\frac{h-\mu}{\sigma} = z$, and using the previous two integrals from the expected value. The integral with the z^2 term is then calculated with integration by parts,

$$\begin{aligned}\sigma^2\int_{-\frac{\mu}{\sigma}}^{\infty} z^2\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\;dz &= -\sigma^2 z\frac{\exp(-\frac{z^2}{2})}{\sqrt{2\pi}}\Bigg|_{-\frac{\mu}{\sigma}}^{\infty} + \sigma^2\int_{-\frac{\mu}{\sigma}}^{\infty}\frac{\exp\left(-\frac{z^2}{2}\right)}{\sqrt{2\pi}}\;dz \\ &= -\mu\sigma\frac{\exp(-\frac{\mu^2}{2\sigma^2})}{\sqrt{2\pi}} + \frac{\sigma^2}{2}\left[1+\mathrm{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right)\right].\end{aligned}$$

Adding the three terms yields the second moment of h,

$$\mathbb{E}_{h\sim\mathcal{N}_{\mathrm{R}}(\mu,\sigma^2)}[h^2] = \mu\bar{h} + \frac{\sigma^2}{2}\left[1+\mathrm{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right)\right],$$

and its variance,

$$\mathrm{Var}(h) = \frac{\sigma^2}{2}\left[1+\mathrm{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right)\right] - \bar{h}(\bar{h}-\mu). \tag{17}$$

A.4 CLT for independent rectified Gaussian random variables

In section 2.2.2, to obtain the average classification condition constant for the margin loss we assumed the output layer to be symmetrically distributed, which we show in appendix B.1 to be empirically satisfied. For wide enough layers this is the case, since for the case of independent, but not necessarily identically distributed, rectified Gaussian random variables, the Central Limit Theorem (CLT) applies, which we prove in this section.

Proposition 5. *Let $h_1, \dots, h_n$ be a sequence of independent rectified Gaussian random variables, where h_k has mean $\bar{h}_k$ and variance $\mathrm{Var}(h_k) = \sigma_k^2$. Then, the sum $\sum_{k=1}^{n} h_k$ satisfies the CLT.*

Proof. For the proof we use the following theorem and condition:

Theorem 6 (Lindeberg theorem (Billingsley, 1995))**.** *Assume a sequence $x_1, \dots, x_n$ of n independent random variables, where the random variable x_k has mean $\mathbb{E}[x_k] = 0$ and variance $\sigma_k^2 = \mathbb{E}[x_k^2]$. Then, the sum $\sum_{k=1}^{n} x_k$ converges to a normal distribution $\mathcal{N}(0, s_n^2)$, where $s_n^2 = \sum_{k=1}^{n} \sigma_k^2$, if the Lindeberg condition holds for all $\epsilon > 0$.*

Condition (Lindeberg condition)**.** *Assume a sequence $x_1, \dots, x_n$ of n independent random variables, where the random variable x_k has mean $\mathbb{E}[x_k] = 0$ and variance $\sigma_k^2 = \mathbb{E}[x_k^2]$. Then,*

$$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{s_n^2} \int_{|x_k| \geq \epsilon s_n} x_k^2 dP_k = 0 \tag{18}$$

is the Lindeberg condition, where $s_n^2 = \sum_{k=1}^{n} \sigma_k^2$.

Let's consider a sequence $h_1, \dots, h_n$ of independent rectified Gaussian random variables, where h_k has mean $\bar{h}_k$ and variance $\mathrm{Var}(h_k) = \sigma_k^2$. Without loss of generality, let $\sigma_1^2 \leq \sigma_2^2 \leq \dots \leq \sigma_n^2$ and $x_k = h_k - \bar{h}_k$ so that it satisfies[5] $\mathbb{E}[x_k] = 0$. We also define

$$\ell = \arg\max_k \left[\int_{|x_k| \geq \sqrt{n}\epsilon\sigma_1} x_k^2 dP_k \right]. \tag{19}$$

Then, for all $\epsilon > 0$,

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{s_n^2} \sum_{k=1}^{n} \int_{|x_k| \geq \epsilon s_n} x_k^2 dP_k &\leq \lim_{n\to\infty} \frac{1}{n\sigma_1^2} \sum_{k=1}^{n} \int_{|x_k| \geq \sqrt{n}\epsilon\sigma_1} x_k^2 dP_k \\
&\leq \lim_{n\to\infty} \frac{1}{n\sigma_1^2} n \int_{|x_\ell| \geq \sqrt{n}\epsilon\sigma_1} x_\ell^2 dP_\ell \\
&= \frac{1}{\sigma_1^2} \lim_{n\to\infty} \int_{|x_\ell| \geq \sqrt{n}\epsilon\sigma_1} x_\ell^2 dP_\ell \\
&= 0,
\end{aligned}$$

where we used $n\sigma_1^2 \leq s_n^2$ for the first and 19 for the second inequality. The last limit is zero because $P_\ell(x_\ell)$ is a shifted rectified Gaussian distribution, which means that $\lim_{n\to-\infty} P_\ell(x_\ell) = 0$ and $P_\ell(x_\ell)$ rapidly decays for large x_ℓ, i.e., it decays exponentially while the x_ℓ^2 part of the integrand grows quadratically.

---

[5] This change of variable shifts the mean of the sum to 0 without changing the variance and the second moment integral 18

Therefore, Lindeberg's condition is satisfied, so the sum of independent but not necessarily identically distributed rectified Gaussian random variables follows the CLT.

$\square$

B Complementary experimental results

B.1 Approximate output normality

In section 3.1, we assumed that the output layer is distributed according to a multi-variate Normal distribution with independent outputs to obtain closed-form estimates of the smoothed margin and probability of correct classification. Here we compare the values of the empirical smoothed margin and probability of correct classification with the ones obtained through numerical integration to show that for the case of orthogonal weight matrices with wide enough layers (512 units) this assumption is fairly satisfied.

We carried the numerical integration of the integral equations 11 and 13 used for the estimates with `integrate.quad` function from python's `scipy` package. The integration interval was selected to be $\mu_j \pm 4\sigma_j$, where j is the index of the output variable which the integration is taken, since smaller values would result in larger integration errors while larger ones would increase the computational cost with small improvements in precision.

For the empirical evaluation, we obtained sample estimates of the probability of correct classification,

$$\mathcal{P}_c'\left(\boldsymbol{x}, \sigma_x^2\right) = \frac{\sum_{i=1}^{N} \mathbf{1}_{o_c(\boldsymbol{x}_i) > o_j(\boldsymbol{x}_i), \forall j \neq c}}{N},$$

where $N = 10^6$ was the number of samples taken by forwarding different samples of input perturbation from an isotropic Gaussian noise.

Both sample and numerical integration values were obtained by increments of $\Delta\sigma_x = 0.032$ on the input standard deviation, using a NN with settings from section C after 30 training epochs. In figure 6 we see that the approximate output normality assumption is satisfied, given that the numerical and empirical values agree well.

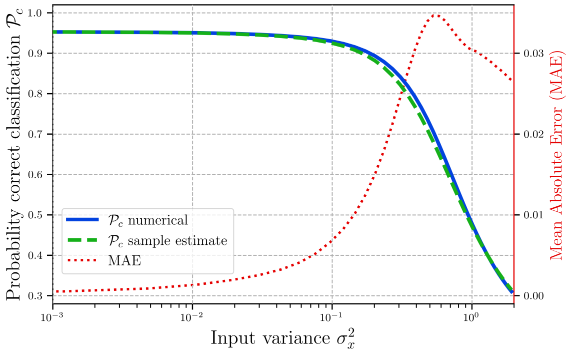

Figure 6: Comparison of correct classification probability $\mathcal{P}_c$ through numerical integration (continuous blue) and sample estimate (dashed green) for different values of input variance σ_x^2. The second axis shows the Mean Absolute Error (MAE) between the numerical and sample estimate values (dotted red). Each curve corresponds to the average over test samples after 30 epochs.

B.2 Closed form approximations

In figure 7 we present how close the approximations (dashed purple lines) of the smoothed margin (eq. 12) and probability of correct classification (eq. 14) are, respectively, to their integral equations 11 and 13 (continuous blue lines). We can see that, overall, the approximations are close to their numerical values, starting to diverge only for larger values of input variance. We also point that the maximal mean absolute error happens when half the samples are misclassified on average, i.e., when $\mathcal{P}_c = 0.5$ and $\mathcal{L}_M = 0$ on average.

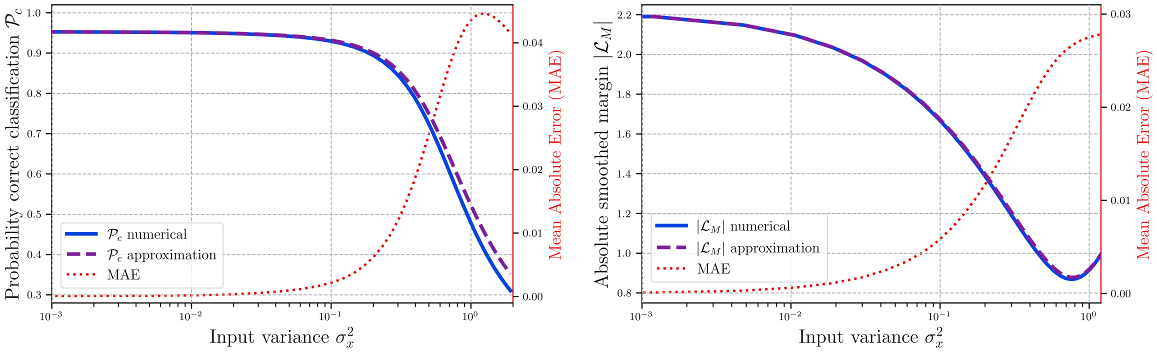

Figure 7: Comparison of approximation (purple) and numerical integration (blue) values of probability of correct classification (left) and absolute smoothed margin (right) for increasing input variance. The second axis shows the Mean Absolute Error (MAE) between the numerical and approximation values (red line). Each curve corresponds to the average over test samples after 30 epochs, and we plot the absolute smoothed margin for better visualization, since as the variance increases, the signs of the SM start to change making the average SM close to zero.

C Additional experimental settings

Loss hyperparameters search

We used $\sigma_x^2 \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$ for the SC-SCE loss. For the MCR and MH losses, we used $d \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$. For the Zhen loss we did a grid search over $\sigma_x^2 \in \{0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.28, 1.75\}$ and $d \in \{5, 10, 15, 20, 25, 30\}$. For the MSCR, over $\sigma_x^2 \in \{10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 10^{0}\}$ and $d \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$.

Training

For the NN architecture we used a three hidden-layer MLP with 512 units each, orthogonal initialization of weight matrices and ReLU activations. The weight update algorithm was Adam (Kingma & Ba, 2017) with learning rate of 0.01 and the usual hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, starting the weight updates only after 2000 steps of updating the adaptive parameters $\boldsymbol{m}$ and $\boldsymbol{v}$ for learning stability. To improve even more the stability, specially for the case of margin-based losses without the Lipschitzness constraint, since it helps in stabilizing the gradients, we normalized the gradient before passing it to Adam.

Finally, we found that using cosine annealing as proposed by Loshchilov & Hutter (2017) with warm restarts resulted in generally better and faster convergences. We implemented the method described in their appendix, so we'll not go into details, but the method works by decaying the scale of the learning rate from 1 to 0 in the first n_0 epochs according to the cosine function and, after reaching the n_0-th epoch, the scaling goes back to 1 and starts the decay again. When the scaling goes back to 1 this is called a warm restart and helps avoid poor local minima and, to improve stability, the decay takes more time in-between warm restarts, with $n_{k+1} = m \cdot n_k$ epochs from the k-th to the $(k+1)$-th warm restart, where m is a constant integer greater than 1. By choosing $n_0 = 20$ and $m = 2$, in the epoch 300 it finishes the third warm restart, with 160 epochs, which we found to result in stable training.

Acronyms

AE Adversarial Example.

AMS Average Misclassification Standard deviation.

AR Adversarial Robustness.

ASR Average Successful Radius.

BO Björck orthogonalization.

CDF Cumulative Distribution Function.

CLT Central Limit Theorem.

CR Certified Radius.

DL Deep Learning.

MAE Mean Absolute Error.

MCR Margin Certified Radius.

MH Multiclass Hinge.

ML Machine Learning.

MLP Multilayer Perceptron.

MSCR Margin Smoothed Certified Radius.

NN Neural Networks.

PGD Projected Gradient Descent.

ReLU Rectified Linear Unit.

RS Randomized Smoothing.

RSCR Randomized Smoothing Certified Radius.

SA Smoothed Analysis.

SC-SCE Smoothed Classifier Softmax Cross-Entropy.

SCE Softmax Cross-Entropy.

SCR Smoothed Certified Radius.

SM Smoothed Margin.

SRA Smoothed Robustness Analysis.

SVM Support Vector Machine.