Certifying Neural Network Robustness to Random Input Noise from Samples

Brendon G. Anderson
University of California, Berkeley

Somayeh Sojoudi
University of California, Berkeley

Abstract

Methods to certify the robustness of neural networks in the presence of input uncertainty are vital in safety-critical settings. Most certification methods in the literature are designed for adversarial input uncertainty, but researchers have recently shown a need for methods that consider random uncertainty. In this paper, we propose a novel robustness certification method that upper bounds the probability of misclassification when the input noise follows an arbitrary probability distribution. This bound is cast as a chance-constrained optimization problem, which is then reformulated using input-output samples to replace the optimization constraints. The resulting optimization reduces to a linear program with an analytical solution. Furthermore, we develop a sufficient condition on the number of samples needed to make the misclassification bound hold with overwhelming probability. Our case studies on MNIST classifiers show that this method is able to certify a uniform infinity-norm uncertainty region with a radius of nearly 50 times larger than what the current state-of-the-art method can certify.

1 INTRODUCTION

Real-world data is inherently uncertain. Such uncertainty comes in a variety of forms, including random measurement noise, adversarial attacks, and even geometric transformations such as the morphing of images (Franceschi et al., 2018; Balunovic et al., 2019; Jin et al., 2019). Many neural networks, despite having excellent predictive capabilities within their nominal operating regime, are highly sensitive to these uncertainties in their inputs (Szegedy et al., 2013; Fawzi et al., 2016; Su et al., 2019). This sensitive behavior is absolutely intolerable when using neural networks to operate safety-critical systems, such as the power grid (Kong et al., 2017). As a result, a large emphasis has been placed by researchers on the development of methods that certify the robustness properties of neural networks.

Much of the literature on robustness certification has revolved around adversarial inputs, i.e., inputs that are designed to cause a worst-case prediction (Wong and Kolter, 2017; Weng et al., 2018; Raghunathan et al., 2018; Anderson et al., 2020). However, as argued in Webb et al. (2018) and Mangal et al. (2019), random input uncertainty better models reality in many cases. Additionally, Weng et al. (2019) showed that random input noise poses a valid threat, as it can cause misclassifications in MNIST and CIFAR networks with a high rate. There have been a handful of recent works considering robustness to random inputs. However, many of them make stringent assumptions on the structure of the network or input distribution, or the formal certification guarantees are relaxed or eliminated in order to enhance computational tractability.

1.1 Related Works

In this section, we review the state-of-the-art methods for assessing robustness to random inputs, highlight their usages, and address their limitations. For instance, Mangal et al. (2019) defines robustness as the network output being Lipschitz continuous with high probability when two inputs are chosen randomly. Their proposed method is limited to neural networks composed of conditional affine transformations, e.g., ReLU networks. On the other hand, Weng et al. (2019) analytically bounds the probability that a classifier's margin function exceeds a given value. Although this probabilistic method applies to general neural network models, it assumes that the random input noise is constrained to an ℓ_p-norm ball and is either Gaussian

Author emails: {bganderson,sojoudi}@berkeley.edu.

or has independent coordinates. Furthermore, their bounding technique relies on worst-case analysis methods, making their resulting certificates relatively loose (see Section 5).

In Webb et al. (2018), robustness is measured by the probability that random input noise results in misclassification. The authors propose a sampling-based approximation of the robustness level. However, no theoretical guarantees are given to formally certify the robustness of the network. Contrarily, Dvijotham et al. (2018) formally bounds the probability that a random input maps to an unsafe output. However, the bounding function is nonconvex, and therefore obtaining tight bounds amounts to solving a nonconvex optimization problem. Alternatively, Franceschi et al. (2018) bounds the size of a random input perturbation that causes a classifier prediction error with high probability. Their analytical bounds provide elegant theoretical guarantees, but they depend on knowing the network's worst-case robustness level, which is generally NP-hard to compute without approximation errors or additional assumptions (Weng et al., 2018; Katz et al., 2017).

Finally, Fazlyab et al. (2019) and Couellan (2020) provide methods to guarantee that network outputs do not significantly deviate from the nominal output when the input is subject to random uncertainty. This corresponds to the problem of localizing the network outputs within the output space. Fazlyab et al. (2019) uses the output localization to issue high-probability guarantees for the network's safety. However, their method requires solving a semidefinite program, and their results are demonstrated on small, single-layer networks, so it is not clear whether their method scales to realistic applications. The concentration bounds presented in Couellan (2020) can be applied to deep networks, but their localization results do not immediately translate into a meaningful certificate, e.g., certifying that misclassification occurs with low probability.

1.2 Contributions

In this paper, we focus on classification networks and upper bound the probability of misclassification in the presence of random input noise. To do this, we propose a chance-constrained optimization problem. The approach is direct, without resorting to worst-case analysis and relaxation techniques. Inspired by recent work in the control theory community (Devonport and Arcak, 2020), we show that replacing the probabilistic constraints with hard constraints on samples reduces to a univariate linear program with an analytical solution. We give formal guarantees that the resulting misclassification bound holds with overwhelming probability upon using sufficiently many samples.

The proposed certification method can be applied to networks of arbitrary architecture, and to random inputs with arbitrary probability distribution. We use the certificate to develop an algorithm which computes the largest certifiable uncertainty region when the input is specialized to the commonly used ℓ_p-norm bounded noise regime. We show through numerical simulations on various MNIST networks that our algorithm is able to certify uncertainty regions nearly 50 times larger than what the current state-of-the-art is capable of.

1.3 Outline

We begin by formulating the certification problem in Section 2. In Section 3, we propose an optimization problem to solve for the robustness certificate, and then show that it is sufficient to compute the analytical solution of a surrogate sample-based optimization. In Section 4, an algorithm is presented for computing the largest certifiable ℓ_p-norm ball uncertainty region. Finally, numerical case studies are presented in Section 5, and conclusions are provided in Section 6.

2 PROBLEM FORMULATION

Consider a pre-trained neural network $f\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$. For the sake of concreteness, we assume f is a classification network, where the output dimension n_y equals the number of classes. In this setting, the class assigned to an input $x \in \mathbb{R}^{n_x}$ is given by $i^* = \arg\max_{i\in\{1,2,\ldots,n_y\}} f_i(x)$, where $f_i(x)$ is the ith element of the output vector $f(x)$. In this work, we consider a random input $X \sim \mathbb{P}$ with a known but otherwise arbitrary probability distribution $\mathbb{P}$. As is commonly done in the neural network robustness literature, we assume the input is composed of a known and deterministic *nominal input*, $\bar{x} \in \mathbb{R}^{n_x}$, and a noise term, $\delta \in \mathbb{R}^{n_x}$. Under this model, $X = \bar{x} + \delta$, and all of the randomness in X comes from the random noise δ. We define the *input set* to be $\mathcal{X} = \mathrm{supp}(\mathbb{P}) \subseteq \mathbb{R}^{n_x}$, i.e., the support of the input distribution.

Suppose the true class of the nominal input $\bar{x}$ is i^*. The goal is to certify that the neural network is robust against the noise δ. To this end, the *margin function* associated to class i with respect to class i^* is defined by

$$g_i(x) = f_{i^*}(x) - f_i(x).$$

If $g_i(X) \geq 0$ for all $i \in \{1, 2, \ldots, n_y\}$, then the neural network assigns class i^* to the perturbed input X. In this case, the classification of X correctly matches that of the nominal input. Contrarily, if $g_i(X) < 0$ for some $i \neq i^*$, then the noise δ causes the input X to

be misclassified. If $g_i(x) \geq 0$ for all i and all $x \in \mathcal{X}$, then X is correctly classified with probability one, and f is said to be *deterministically robust*. Since the methods in this paper can be applied to each class $i \in \{1, 2, \ldots, n_y\}$ individually, we restrict our attention to analyzing just one target class $i \neq i^*$ from here on.

Certifying deterministic robustness has been extensively studied in the context of worst-case and adversarial inputs, but for random inputs, deterministic robustness is often too stringent to hold (Webb et al., 2018). Consequently, the problem we focus on is instead upper bounding the probability of misclassification, as studied in works such as Webb et al. (2018) and Weng et al. (2019). The probability that X is misclassified is given by $\mathbb{P}(g_i(X) < 0)$. Therefore, given a small permissible probability level $\epsilon \in (0, 1)$, we seek to certify that

$$\mathbb{P}(g_i(X) \geq 0) \geq 1 - \epsilon. \tag{1}$$

We call the network *probabilistically robust* if (1) holds.

3 MISCLASSIFICATION PROBABILITY BOUND

In Weng et al. (2019), the misclassification probability is bounded in a two step process: first, the margin function is deterministically bounded by an affine function, and second, the affine function is probabilistically bounded with respect to the random input X. The first step relies upon bounding techniques from the adversarial literature that results in a worst-case bound, which may be unnecessarily loose. On the other hand, Webb et al. (2018) estimates the misclassification probability using random samples, but provides no formal bounding guarantees. In what follows, we formulate a chance-constrained optimization to directly bound the misclassification probability, and then use sampling to give an analytical optimal solution that still provides high-probability guarantees.

Since our goal is to certify that $\mathbb{P}(g_i(X) \geq 0) \geq 1 - \epsilon$, we propose solving the following chance-constrained optimization problem:

$$\begin{aligned} \underset{r \in \mathbb{R}}{\text{maximize}} \quad & r \\ \text{subject to} \quad & \mathbb{P}(g_i(X) \geq r) \geq 1 - \epsilon. \end{aligned} \tag{2}$$

If the solution r^* to (2) satisfies $r^* \geq 0$, then $\mathbb{P}(g_i(X) \geq 0) \geq \mathbb{P}(g_i(X) \geq r^*) \geq 1 - \epsilon$, so we obtain the misclassification bound we desire. Unfortunately, the constraint in (2) generally results in a nonconvex optimization, even though the inequality $g_i(x) \geq r$ is a convex constraint on r (Nemirovski and Shapiro, 2007).

In order to reformulate (2) into a tractable convex optimization, we consider drawing independent samples $X_1, X_2, \ldots, X_N$ from $\mathbb{P}$. Intuitively, if $g_i(X_j) \geq r$ for all $j \in \{1, 2, \ldots, N\}$, then for N large enough, we suspect the probability $\mathbb{P}(g_i(X) \geq r)$ for an arbitrary random input $X \sim \mathbb{P}$ to be close to one. As a result, we can consider solving the following optimization in place of (2):

$$\begin{aligned} \underset{r \in \mathbb{R}}{\text{maximize}} \quad & r \\ \text{subject to} \quad & g_i(X_j) \geq r \text{ for all } j \in \{1, 2, \ldots, N\}. \end{aligned} \tag{3}$$

This method of replacing chance constraints with hard constraints on samples is commonly used in the stochastic optimization and control theory literature, and is called the *scenario approach* (Campi et al., 2009; Tempo et al., 2012). The resulting scenario problem (3) is a univariate linear program with a simple analytical solution, as given by the following proposition.

Proposition 1. *The problem* (3) *is solved by*

$$\hat{r} = \min_{j \in \{1, 2, \ldots, N\}} g_i(X_j). \tag{4}$$

Proof. Clearly $\hat{r}$ is feasible for (3). Furthermore, if $r > \hat{r}$, then $r > g_i(X_j)$ for some $j \in \{1, 2, \ldots, N\}$, so r is not feasible. □

We have shown that the sample-based optimization (3) is easy to solve: all one must do is sample N inputs X_j from $\mathbb{P}$, compute their margin function values $g_i(X_j)$ from the network outputs $f(X_j)$, and then find the minimum margin function value amongst the samples. However, the solution $\hat{r}$ is random, and it is unclear whether the bound on the misclassification probability from the original problem (2) still holds using $\hat{r}$ in place of r^*. In the theorem that follows, we show that using sufficiently many samples ensures that the misclassification bound still holds with overwhelming probability.

Theorem 1. *Let* $\delta \in (0, 1)$. *If* $N \geq \frac{2}{\epsilon}\left(\log \frac{1}{\delta} + 1\right)$ *and* $\hat{r} \geq 0$, *then the bound* (1) *on the misclassification probability holds with probability at least* $1 - \delta$.

Proof. Suppose $N \geq \frac{2}{\epsilon}\left(\log \frac{1}{\delta} + 1\right)$. Since the problem (3) is a convex optimization problem with one optimization variable, Theorem 1 in Campi et al. (2009) gives that, with probability at least $1 - \delta$, the scenario optimization solution $\hat{r}$ satisfies the original chance constraint in (2). In this case, $\hat{r} \geq 0$ implies that

$$\mathbb{P}(g_i(X) \geq 0) \geq \mathbb{P}(g_i(X) \geq \hat{r}) \geq 1 - \epsilon,$$

as desired. □

Theorem 1 shows that the analytical solution (4) obtained using sufficiently many samples is feasible for the chance-constrained problem (2) with overwhelming probability. Therefore, the margin function $g_i(X)$ for a random input X is guaranteed to exceed the value $\hat{r}$ in a high-probability sense. As a result, $\hat{r}$ being nonnegative is sufficient to issue a high-probability certificate that the network is probabilistically robust.

Unlike previous approaches to bounding the misclassification probability, such as in Weng et al. (2019), our sample-based method avoids worst-case deterministic relaxation techniques, and instead we have encoded a form of "probabilistic relaxation" into the user-prescribed parameter δ. Since the required number of samples N grows logarithmically in $1/\delta$, the probabilistic relaxation level δ can be taken very small and still result in a moderate number of input-output samples to compute. Another benefit of our approach is that it can be applied to arbitrary input distributions $\mathbb{P}$, input sets $\mathcal{X}$, and neural network architectures, unlike most other methods in the literature that have restrictions such as Gaussian noise, ℓ_p-norm balls, or ReLU networks (see Section 1.1). Finally, it is important to remark that the number of samples required by Theorem 1 is independent of the number and width of layers in the network, making our bounding procedure particularly well suited for deep neural networks.

4 ALGORITHM FOR ℓ_p-BALLS

In this section, we consider applying our certification method to the special case when the input uncertainty is distributed over an ℓ_p-norm ball, i.e., $\mathcal{X} = \mathcal{B}_p(\bar{x}, \alpha) := \{x \in \mathbb{R}^{n_x} : \|x - \bar{x}\| \leq \alpha\}$. This is a common model of uncertainty in both the adversarial input and random input literature. Similar to Weng et al. (2019), we can use our bound to compute the largest radius α such that misclassification occurs with probability less than ϵ. In particular, when $\hat{r}$ from (4) is less than zero, the certificate from Theorem 1 does not hold, indicating that the uncertainty region is too large. In this case, we can decrease the radius α and recompute a larger value of $\hat{r}$. On the other hand, when $\hat{r} > 0$, it is possible that the uncertainty region may be made larger without the misclassification probability exceeding ϵ. This observation leads to the bisection search given in Algorithm 1.

The output of Algorithm 1 gives the largest uncertainty radius such that random inputs X distributed on the ℓ_p-norm ball of radius α are misclassified with probability no more than ϵ. This robustness certificate holds with probability at least $1 - \delta$ according to Theorem 1. In practice, the computational bottleneck of Algorithm 1 is the feedforward computation of the sampled network outputs $f(X_j)$ used to compute the margin function values. These feedforward computations must be performed at each iteration of the bisection with new samples, since the input set changes at each iteration. Despite this, we will see in Section 5 that the algorithm is able to converge within a few minutes on large MNIST networks. Furthermore, the proposed method is able to certify ℓ_p-norm balls with radius nearly 50 times larger than those certifiable using the state-of-the-art method given by Weng et al. (2019). We will now move to discussing these case studies in detail.

Algorithm 1: Maximum radius ℓ_p-norm ball

Input : Probability levels $\epsilon, \delta \in (0, 1)$
Radius interval endpoints $0 < \alpha_1 < \alpha_2$
Tolerance $\beta \in (0, \alpha_2 - \alpha_1)$
Output: Maximum radius α

initialize $\hat{r} = 0$;
set $N = \left\lceil \frac{2}{\epsilon}\left(\log\frac{1}{\delta} + 1\right)\right\rceil$;
while $\alpha_2 - \alpha_1 > \beta$ **or** $\hat{r} < 0$ **do**
  set $\alpha = \frac{\alpha_1 + \alpha_2}{2}$;
  sample $X_1, X_2, \ldots, X_N \overset{\text{i.i.d.}}{\sim} \mathbb{P}$ from $\mathcal{B}_p(\bar{x}, \alpha)$;
  set $\hat{r}$ according to (4);
  if $\hat{r} > 0$ **then**
    set $\alpha_1 = \alpha$;
  else
    set $\alpha_2 = \alpha$;

5 SIMULATIONS

In this section, we numerically validate the effectiveness of our proposed certification method using two experiments. All results are computed using TensorFlow in Python on a standard laptop with a 2.60 GHz dual-core i5 processor.

In the first experiment, we consider a variety of pretrained fully connected MNIST digit classification networks with either ReLU, tanh, sigmoid, or arctan activation functions. A network model with m hidden layers, each having n neurons, is denoted by $m \times [n]$. We compute the largest radius α for an ℓ_∞-norm uncertainty ball, such that misclassification occurs with probability no more than ϵ when the input is uniformly distributed on $\mathcal{B}_\infty(\bar{x}, \alpha)$. We perform this computation for a variety of probability levels ϵ, and for each one we average the computation over 10 nominal inputs $\bar{x}$. The targeted class i, which defines the margin function g_i relative to the nominal input's true class i^*, is randomly chosen for each input being tested. For the set of inputs tested, we perform the computation

in two ways: first, using the state-of-the-art PROVEN algorithm (Weng et al., 2019), and second, using Algorithm 1 with $\delta = 10^{-5}$. We also include the worst-case values corresponding to $\epsilon = 0$ (i.e., the minimum adversarial distortion), which are obtained from the algorithm in Zhang et al. (2018) and provided in Weng et al. (2019). The results are shown in Table 1.

As seen in Table 1, our method is able to certify a much larger input uncertainty set than PROVEN with the same probability level, for every configuration tested here. Observe the percentage increase in the certifiable radius α obtained using our method compared to PROVEN. For the ReLU networks tested in Table 1a, the percentage increase ranges between 620%–4300%, meaning our method certifies uncertainty sets nearly 50 times as large as what PROVEN can. This certification power comes at the cost of slightly higher computation time for $\epsilon = 0.001$. For higher values of ϵ, our method requires much less data, making our computation time even faster than PROVEN's. Since our computational method depends on ϵ, one can obtain a quick initial certificate using Algorithm 1 with a moderately sized ϵ, then afterward decide to perform a full-strength certification with small ϵ. Contrarily, PROVEN takes the same amount of time to solve whether ϵ is large or small, making it perhaps infeasible to obtain quick certificates for very large networks. Since the vast majority of computation time in Algorithm 1 comes from forward passes of the samples through the network, we suspect our method to perform particularly fast on convolutional neural networks.

Note that the percentage increase in Table 1 is highest for larger and deeper networks. This indicates that our method is especially powerful for certifying deep neural networks. The PROVEN algorithm depends on deterministically bounding the margin function using affine functions before developing the high-probability guarantees, and the affine bounds are conjectured to become looser as the network becomes larger (Weng et al., 2019). This deterministic bounding technique comes from the adversarial robustness literature, and therefore it being embedded into PROVEN is likely the reason why the PROVEN radii are so close to the worst-case radii when compared to our results. Our method bypasses this unnecessary preliminary bound altogether.

To verify that our radii computations in Table 1 are accurate, we give for ReLU networks an *a posteriori* estimate of the probability that no misclassification occurs, i.e., $\mathbb{P}(g_i(X) \geq 0)$. This is done by a Monte Carlo approximation using over 48000 inputs distributed uniformly on the uncertainty ball $\mathcal{B}_\infty(\bar{x}, \alpha)$, where α is computed from our method as in Table 1a and $\bar{x}$ is a fixed nominal input. The probability estimates are shown in Table 2. As seen, each probability level exceeds the prescribed lower bound $1 - \epsilon$, showing that our radii do indeed correspond to input uncertainty sets with certifiably low misclassification probability.

We now discuss the second experiment. As mentioned above, PROVEN works by first deriving an affine lower bound on the nonlinear margin function, then probabilistically bounding the affine function. Therefore, if the neural network in question is a linear classifier, i.e., of the form $f(x) = Wx + b$, then the deterministic part of the PROVEN bound becomes tight. In this case, we suspect the PROVEN radii to closer match the radii computed using our purely probabilistic bound. We compute the radii α for a variety of linear classifiers of input size n_x and output size n_y in Table 3. For simplicity, the network weights and biases are chosen with elements uniformly distributed on $[0, 1]$ in this experiment. Since the radius obtained using our method depends on the random input-output samples, we perform the computation 30 times using Algorithm 1 for each configuration, and provide the sample mean and variance of the results. As expected, the PROVEN certificates are much closer to ours in this special case, although our method is still able to outperform PROVEN in every configuration, with very little variance in the computations. These results show that the worst-case bounding techniques used in the adversarial robustness literature may work well for simple models with random inputs, such as linear classifiers, but that for general nonlinear networks, these bounds are too loose, and we require novel probabilistic approaches like the one proposed here.

6 CONCLUSIONS

In this paper, we propose a data-driven method for certifying the robustness of neural networks to random input uncertainty. The optimization-based certificate is shown to have a simple analytical solution, and the resulting bound on the misclassification probability can be applied to neural networks of any size and structure, as well as inputs following any probability distribution. In the case the input noise is uniformly distributed on an ℓ_p-norm ball, an algorithm to compute the largest certifiable uncertainty region is presented. Our numerical experiments show that the proposed method is capable of certifying nearly 50 times larger uncertainty sets than the state-of-the-art method when applied to MNIST networks with various activations. The results suggest that new data-driven probabilistic methods, like that proposed here, should be used to certify networks in the presence of random inputs, instead of re-tooling the worst-case bounding techniques from the adversarial robustness literature.

Table 1: Maximum ℓ_∞-norm uncertainty ball radii α for various networks and probability levels ϵ. Results are for MNIST networks with $\delta = 10^{-5}$, and are averaged over 10 nominal inputs with randomly chosen target classes i. Worst-case values correspond to $\epsilon = 0$ and are computed using the method Zhang et al. (2018). PRVN (PROVEN) is the state-of-the-art method for certifying against random input noise (Weng et al., 2019).

(a) ReLU activations. Computation times are also listed in seconds.

Model	Worst-case	$\epsilon = 0.001$		$\epsilon = 0.25$		$\epsilon = 0.5$		Increase
		PRVN	Ours	PRVN	Ours	PRVN	Ours	
$2 \times [20]$	0.02746	**0.04864**	**0.35044**	0.05111	0.54118	0.05173	0.61983	**620.48%**
	—	0.0440 s	76.8045 s	0.0432 s	0.1641 s	0.0419 s	0.0924 s	
$3 \times [20]$	0.02236	**0.03813**	**0.35082**	0.03925	0.57673	0.03952	0.62028	**820.06%**
	—	0.0901 s	61.2525 s	0.0919 s	0.1404 s	0.0839 s	0.1189 s	
$2 \times [1024]$	0.03158	**0.05531**	**0.60665**	0.05694	0.85456	0.05734	0.93232	**996.82%**
	—	0.9062 s	214.1965 s	0.8888 s	0.4254 s	0.8887 s	0.2569 s	
$3 \times [1024]$	0.02397	**0.03531**	**0.64541**	0.03571	0.87134	0.03581	0.95033	**1727.84%**
	—	21.8928 s	362.8095 s	14.9471 s	0.7712 s	22.7815 s	1.1072 s	
$4 \times [1024]$	0.00962	**0.01289**	**0.56906**	0.01292	0.82702	0.01293	0.88040	**4314.97%**
	—	67.2073 s	535.2870 s	55.3338 s	1.5010 s	54.7098 s	0.8888 s	

(b) tanh activations.

Model	Worst-case	$\epsilon = 0.001$		$\epsilon = 0.25$		$\epsilon = 0.5$		Increase
		PRVN	Ours	PRVN	Ours	PRVN	Ours	
$2 \times [1024]$	0.02232	**0.02915**	**0.49781**	0.03005	0.75077	0.03013	0.82327	**1607.75%**
$3 \times [1024]$	0.01121	**0.01360**	**0.45129**	0.01376	0.66109	0.01378	0.76432	**3218.31%**
$4 \times [1024]$	0.00682	**0.00745**	**0.47325**	0.00750	0.73762	0.00750	0.79554	**6252.35%**

(c) Sigmoid activations.

Model	Worst-case	$\epsilon = 0.001$		$\epsilon = 0.25$		$\epsilon = 0.5$		Increase
		PRVN	Ours	PRVN	Ours	PRVN	Ours	
$2 \times [1024]$	0.02785	**0.03285**	**0.45103**	0.03404	0.73559	0.03419	0.77772	**1273.00%**
$3 \times [1024]$	0.01856	**0.02296**	**0.39569**	0.02342	0.62579	0.02348	0.69109	**1623.39%**
$4 \times [1024]$	0.01778	**0.02170**	**0.38415**	0.02224	0.64037	0.02229	0.69021	**1670.28%**

(d) arctan activations.

Model	Worst-case	$\epsilon = 0.001$		$\epsilon = 0.25$		$\epsilon = 0.5$		Increase
		PRVN	Ours	PRVN	Ours	PRVN	Ours	
$2 \times [1024]$	0.02105	**0.02796**	**0.41446**	0.02907	0.65673	0.02915	0.74568	**1382.33%**
$3 \times [1024]$	0.01250	**0.01462**	**0.47061**	0.01486	0.75402	0.01488	0.78567	**3118.95%**
$4 \times [1024]$	0.00726	**0.00829**	**0.41343**	0.00836	0.64085	0.00837	0.74236	**4887.09%**

Table 2: Empirical estimates of $\mathbb{P}(g_i(X) \geq 0)$ for MNIST ReLU networks using over 48000 random samples. Samples are drawn uniformly from the ℓ_∞-norm ball with radius α computed using our method, as in Table 1.

Model	$\epsilon = 0.001$	$\epsilon = 0.01$	$\epsilon = 0.1$	$\epsilon = 0.25$
$2 \times [20]$	1.00000	0.99988	0.99700	0.99688
$3 \times [20]$	0.99994	0.99988	0.99359	0.99806
$2 \times [1024]$	0.99994	1.00000	0.99925	0.99904
$3 \times [1024]$	0.99996	0.99990	0.99823	0.99844
$4 \times [1024]$	1.00000	0.99994	0.99915	0.99904

Table 3: Maximum ℓ_∞-norm uncertainty ball radii α for linear classifiers (activation functions are identity). The network weights and biases are randomly chosen with elements uniformly distributed on $[0, 1]$. PRVN (PROVEN) is the state-of-the-art method for certifying against random input noise (Weng et al., 2019). Our proposed algorithm is performed 30 times for each configuration, and the sample mean and variance of the resulting α values are reported.

(n_x, n_y)	$\epsilon = 0.001$			$\epsilon = 0.1$			$\epsilon = 0.25$		
	PRVN	Mean	Var.	PRVN	Mean	Var.	PRVN	Mean	Var.
$(50, 10)$	0.2244	0.3738	0.0002	0.3886	0.5363	0.0023	0.5008	0.6070	0.0060
$(100, 50)$	0.2722	0.4344	0.0003	0.4715	0.6405	0.0029	0.6077	0.7244	0.0058
$(500, 100)$	0.3516	0.5517	0.0007	0.6089	0.8177	0.0048	0.7848	0.8931	0.0148
$(1000, 500)$	0.0930	0.1476	0.0000	0.1610	0.2152	0.0003	0.2076	0.2460	0.0006

References

B. G. Anderson, Z. Ma, J. Li, and S. Sojoudi. Tightened convex relaxations for neural network robustness certification. *arXiv preprint arXiv:2004.00570*, 2020.

M. Balunovic, M. Baader, G. Singh, T. Gehr, and M. Vechev. Certifying geometric robustness of neural networks. In *Advances in Neural Information Processing Systems*, pages 15287–15297, 2019.

M. C. Campi, S. Garatti, and M. Prandini. The scenario approach for systems and control design. *Annual Reviews in Control*, 33(2):149–157, 2009.

N. Couellan. Probabilistic Robustness Estimates for Deep Neural Networks. In *ICML workshop on Uncertainty and Robustness in Deep Learning*, Virtual Conference, United States, July 2020. International Conference on Machine Learning (ICML). URL `https://hal-enac.archives-ouvertes.fr/hal-02572277`.

A. Devonport and M. Arcak. Estimating reachable sets with scenario optimization. *Proceedings of Machine Learning Research*, 2020.

K. Dvijotham, M. Garnelo, A. Fawzi, and P. Kohli. Verification of deep probabilistic models. *arXiv preprint arXiv:1812.02795*, 2018.

A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pages 1632–1640, 2016.

M. Fazlyab, M. Morari, and G. J. Pappas. Probabilistic verification and reachability analysis of neural networks via semidefinite programming. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 2726–2731. IEEE, 2019.

J.-Y. Franceschi, A. Fawzi, and O. Fawzi. Robustness of classifiers to uniform ℓ_p and gaussian noise. *arXiv preprint arXiv:1802.07971*, 2018.

M. Jin, J. Lavaei, S. Sojoudi, and R. Baldick. Boundary defense against cyber threat for power system operation. *arXiv preprint arXiv:1908.10315*, 2019.

G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, 10(1):841–851, 2017.

R. Mangal, A. V. Nori, and A. Orso. Robustness of neural networks: a probabilistic and practical approach. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pages 93–96. IEEE, 2019.

A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

A. Raghunathan, J. Steinhardt, and P. S. Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pages 10877–10887, 2018.

J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

R. Tempo, G. Calafiore, and F. Dabbene. *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media, 2012.

S. Webb, T. Rainforth, Y. W. Teh, and M. P. Kumar. A statistical approach to assessing neural network robustness. *arXiv preprint arXiv:1811.07209*, 2018.

L. Weng, P.-Y. Chen, L. Nguyen, M. Squillante, A. Boopathy, I. Oseledets, and L. Daniel. Proven: Verifying robustness of neural networks with a probabilistic approach. In *International Conference on Machine Learning*, pages 6727–6736. PMLR, 2019.

T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

E. Wong and J. Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pages 4939–4948, 2018.