Literature review: Robustness in Deep Learning

Leonard Waldmann*

June 2023

Abstract

We give a brief literature review on the main concepts of robustness in deep learning and more detailed information on robust out-of-distribution detection and robustness to semantic perturbations. We do not aim to present the topics in their full depth, but to present the main concepts and provide references for further reading. This reports is written for a graduate student to quickly familiarize themselves with the topic.

Contents

Disclaimer: robustness of black-box models is a very active research topic and consistent terminology has not yet been established. Moreover, many different viewpoints on the main ideas and definitions are possible. Here, I will present a view that I found intuitive.

0.1 Certified and Adversarial Robustness

Neural network classifiers have showed remarkable performance in image classification tasks on both train and test datasets. However, small additive perturbations indistinguishable to the human eye can easily fool these classifiers into predicting false classes with high probabilities [29]. From a security point of view, we would like to *guarantee* that networks are robust to these kind of small perturbations.

*leonard.waldmann@gmx.de

0.1.1 Definitions and Intuition

In the following, let $\mathcal{Y}$ denote the set of classes and $\|\cdot\|$ denote the euclidean norm on $\mathbb{R}^n$. Let $f : \mathbb{R}^n \to [0,1]^{|\mathcal{Y}|}$ be a *soft classifier* and $h : \mathbb{R}^n \to \mathcal{Y} : x \mapsto \text{argmax}_{c\in\mathcal{Y}} f_c(x)$ be the corresponding *hard classifier*. Let $x \in \mathbb{R}^n$ and $\epsilon > 0$.

Definition 1 (Robustness)**.** We say $h(x) \in \mathbb{R}^n$ is *robust* with *perturbation budget* ϵ if

$$h(x) = h(x+\delta) \quad \forall \delta \in \mathbb{R}^n \text{ with } \|\delta\| < \epsilon. \tag{1}$$

If this is not the case, we call a point $x + \delta$ that violates eq. (1) the *adversary*, *adversarial attack* or *adversarial example*.

Since we want to evaluate the robustness in practice, we are interested in verifying eq. (1). Clearly, robustness of a point x, i.e. eq. (1), is equivalent to the optimization problem [24, Eq. (1)]

$$0 > \max_{\|\delta\|<\epsilon, c\neq h(x)} f_c(x+\delta) - f_{h(x)}(x+\delta). \tag{2}$$

Unfortunately, exactly solving problem (2) is an NP-complete problem [16, Apdx. I]. Hence, verifying or *certifying* the robustness of a sample with any computational *certification method* $\mathcal{M}(x, h, \epsilon)$ is a challenging problem. To make the certification computationally feasible, we usually use approximative certification methods, which can eventually certify a sample, but cannot guarantee that the sample cannot be certified at all (aka. soundness). This essentially corresponds to computing upper bounds on the optimization problem of eq. (2) and checking if the inequality holds, c.f. Section 0.1.2. On the other hand, a computationally feasible approach is *trying to find* an adversary with an *attack model* $a(x, h, \epsilon) = x + \delta$, that violates eq. (2). This essentially corresponds to finding lower bounds on the optimization problem in eq. (2) and checking if the inequality holds. We talk about *certified robustness*, if x can be certified by a method $\mathcal{M}(x, h, \epsilon)$ and we talk about *adversarial robustness*, if an attack model $a(x, h, \epsilon)$ *cannot find* and adversarial attack. The term adversarial robustness, however, is misleading since not being able to find an adversary with some attack model does not imply that there exists no adversary, i.e. does not imply robustness of the sample. We introduce simple metrics to measure adversarial and certified robustness on a dataset.

Definition 2 (Adversarial Accuracy)**.** Let $\mathcal{D} = \{x_i, y_i\}_{i=1,\dots,N}$ be a dataset and $a(\cdot, h, \epsilon) : \mathbb{R}^n \to \mathbb{R}^n$ be an attack model that tries to find an adversary within the perturbation budget ϵ. Then, the *adversarial accuracy* of h on $\mathcal{D}$ under attack $a(\cdot, h, \epsilon)$ is

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(y_i = h(a(x_i, h, \epsilon))) \tag{3}$$

Definition 3 (Certified Accuracy)**.** Let $\mathcal{D} = \{x_i, y_i\}_{i=1,\dots,N}$ be a dataset and $\mathcal{M}(\cdot, h, \epsilon)$ be a method that returns true if it can computationally certify the

robustness of $h(x)$ with perturbation budget ϵ. Then, the *certified accuracy* of h on $\mathcal{D}$ under method $\mathcal{M}(\cdot, h, \epsilon)$ is

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(y_i = h(x_i) \text{ and } h(x_i) \text{ can be certified by } \mathcal{M}(\cdot, h, \epsilon)). \tag{4}$$

Some papers omit the $y_i = h(x_i)$ condition in certified accuracy. Again, we emphasize the interpretation of the accuracies given an approximative certification method $\mathcal{M}(\cdot, h, \epsilon)$ and attack model $a(\cdot, h, \epsilon)$; The following statements are guaranteed conclusions (denoted by $\rightarrow$):

- High certified accuracy $\rightarrow$ it is provably hard to fool the classifier on the dataset within the perturbation budget.
- Low adversarial accuracy $\rightarrow$ it is provably easy to fool the classifier on the dataset within the perturbation budget, e.g. by the attacks computed by $a(\cdot, h, \epsilon)$.

The following results leave room for interpretation (denoted by $\rightsquigarrow$):

- Low certified accuracy $\rightsquigarrow$ $\mathcal{M}(\cdot, h, \epsilon)$ could not certify the samples on the dataset within the perturbation budget, so either, the samples are actually not robust or the approximation of $\mathcal{M}(\cdot, h, \epsilon)$ is too loose.
- High adversarial accuracy $\rightsquigarrow$ $a(\cdot, h, \epsilon)$ could not find a way to fool the classifier on the dataset within the perturbation budget, so either, the samples are actually robust or the attack model generated weak attacks.

From a research perspective, we have the following trade-offs; high certified accuracy *guarantees* a robust classifier but computing certified accuracy is hard. Computing adversarial accuracy is easy given a reasonable attack model a_ϵ, but it does *not guarantee* a robust classifier. Thus, research interests include:

- Certified robustness: developing effective and efficient ways to certify a sample given a classifier and improving a given classifier to increase its certified accuracy
- Adversarial robustness: developing strong attacks and developing defenses against specific attacks

Remark 1 (Different norms)**.** *Instead of the euclidean norm, we can choose any other norm on* $\mathbb{R}^n$*. However, statements about certified and adversarial accuracy will change. On, e.g., an image with values scaled to* $[0, 1]$*, an adversary under budget of* $\epsilon = 1$ *w.r.t.* $\|\cdot\|_2$ *roughly allows changing one pixel to any value but* $\epsilon = 1$ *w.r.t.* $\|\cdot\|_\infty$ *allows changing all pixels of the image to any values.*

Remark 2 (Robustness and accuracy)**.** *Unfortunately, it has been observed that higher robustness usually comes at the cost of lower clean accuracy when the model size is fixed. On a positive note, some papers claim to have avoided this trade-off, see e.g. Carlini et al. [2].*

0.1.2 Methods for Certified Robustness

Given some perturbation budget ϵ and a classifier f, we try to certify a sample x. Since exact methods are not feasible yet, see above, we need to use approximations. In this case, not being able to certify x with some method does not imply that $f(x)$ cannot be certified at ϵ at all, it may just be that the approximation of the method is too loose.

Randomized Smoothing. The current gold standard is *randomized smoothing* by Cohen, Rosenfeld, and Kolter [5]. We smooth a given soft classifier f with Gaussian noise, i.e.

$$g(x) = \operatorname{argmax}_{c \in \mathcal{Y}} \mathbb{P}[f(x+\epsilon) = c] \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I). \tag{5}$$

which turns g into a Lipschitz continuous function [25, Apdx. A]. For a given point x, we can neatly determine the radius of the largest L^2-ball that can be certified around x as [5, Thm. 1]

$$R = \epsilon = \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)) \tag{6}$$

where p_A is the largest and p_B the second largest probability of $g(x)$ and Φ^{-1} is the inverse standard Gaussian cdf. In particular, this radius is tight. To calculate the radius, we need to approximate the values p_A, p_B by sampling and use statistical tests to validate inequalities. Unfortunately, we need many samples to obtain results with high statistical significance [5, c.f. Fig. 5]. Randomized smoothing is the only method yielding good results on large models, but the sampling makes certification resource expensive. The approach has also been extended to other norms [32].

Bound Propagation. Instead of solving the optimization problem in eq. (2) exactly, we compute upper bounds and check whether they satisfy the inequality. Considering the L^∞ norm and ReLU feedforward NNs, a simple and fast method is *Interval Bound Propagation* (IBP) by Gowal et al. [10], where we sequentially propagate upper and lower bounds through the layers using linearity of the neurons and simple properties of the activation functions. Although the bounds are quite weak, simple implementation and fast running time make this a competitive approach. *Reluplex* by Katz et al. [16] can also verify samples in ReLU networks by splitting each ReLU activation into its two linear parts in two separate variables and use the simplex algorithm that simultaneously optimizes and matches the variables. This approach performs well for small networks but is infeasible for large networks. Improving on Reluplex, newer methods like *DeepPoly* by Singh et al. [28] are more scalable and precise but also make more heavy use of optimization theory.

Lipschitz Continuity. The Lipschitz constant of a function expresses how much the change of the output can be controlled by the change of the input.

Definition 4. Define $X = (\mathbb{R}^n, \|\cdot\|_X), Y = (\mathbb{R}^m, \|\cdot\|_Y)$. A function $f : X \to Y$ is *locally Lipschitz continuous* at $x \in X$ if

$$\|f(x) - f(x')\|_Y \leq L_x^{X\to Y}\|x - x'\|_X \quad \forall x' \in X \tag{7}$$

for some $L_x^{X\to Y} \geq 0$. f is *Lipschitz continuous* if

$$\|f(x) - f(x')\|_Y \leq L^{X\to Y}\|x - x'\|_X \quad \forall x, x' \in X \tag{8}$$

for some $L^{X\to Y} \geq 0$. The smallest $L_x^{X\to Y}$ that fulfills eq. (7) is the *local Lipschitz constant* of f at x. The smallest $L^{X\to Y}$ that fulfills eq. (8) is the *Lipschitz constant* of f.

While f being (locally) Lipschitz continuous does not depend on the chosen norms (all norms are equivalent in $\mathbb{R}^n$), the values of the (local) Lipschitz constants do. Lipschitz continuity naturally connects to certified robustness.

Proposition 1. *[34, Prop. B.1] Let f be a soft classifier with Lipschitz constant L_f w.r.t. $\|\cdot\|_p$ and h a hard classifier. We have*

$$h(x + \delta) = h(x) \quad \textit{for all } \|\delta\|_p \leq \frac{c}{L}\,\mathrm{margin}(f(x)) \tag{9}$$

where $L \geq L_f$, $c = 1/2$ if $p = \infty$ and else $c = \sqrt[p]{2}/2$, and $\mathrm{margin}(\cdot)$ *is the difference between the largest and second largest input values. For given x, we can replace L_f by the local Lipschitz constant of f at x.*

It remains to find the exact (local) Lipschitz constant L_f or good upper bounds L. Finding exact Lipschitz constants turns out to be a NP-complete problem [26, Thm. 2] and is practically feasible only for small models. Exact methods for local Lipschitz constants include, e.g., the work of Jordan and Dimakis [15] using *mixed integer linear programming* or the work of Shi et al. [27] using bound propagation. Exact methods for global Lipschitz constants include the work of Fazlyab et al. [7], that describe the estimation as a semi-definite program. In general, producing accurate and fast estimation of (local) Lipschitz constants that scale to large models is still very challenging, where local bounds are tighter but costly to compute and global bounds are loose but more efficient to compute. Since low Lipschitz constant implies good robustness, c.f. Proposition 1, other works experimentally explore the influence of architecture, loss function and optimizer on the Lipschitz constant [17] or design training regularisation terms using upper Lipschitz bounds [9]. Another line of work enforces a low Lipschitz constant by design of the network, e.g. Zhang et al. [34] propose *SortNet* that bound the spectral norm of the weight matrix while sorting the layer inputs to maintain expressiveness.

0.2 Methods for Adversarial Robustness

Following Madry et al. [19], training robust classifiers can be characterized by formulating the objective as saddle point problem

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\|\delta\|<\epsilon} L(\theta, f_\theta(x + \delta), y)]. \tag{10}$$

Attacks try to find a good adversary within a given perturbation budget for a given model, i.e. solve the inner maximization, and *defenses* try to train or finetune a model to have low loss against adversarial attacks, i.e., solve the outer minimization. Note that here, we framed an adversarial attack as targeting the loss while we defined it in Definition 1 as targeting the decision of the classifier. However, since the loss captures miss-classification, the definitions are essentially the same. Hence, computing the inner maximization is NP-complete.

Adversarial Attacks. A straight-forward attack for the L^∞ norm is

$$x^{t+1} = P_{\{s|\|x-s\|<\epsilon\}}(x^t + \alpha \operatorname{sign} \nabla_{x^t} L(\theta, f_\theta(x^t), y)) \tag{11}$$

where α is the step-size, $x = x^0$ the clean input, and P_S the projection onto a set S. For $\alpha = \epsilon$, we call x^1 *fast gradient sign method* (FGSM) and for $t > 1, \alpha \leq \epsilon$, we call x^t *projected gradient descent attack* (PGD attack) [19, Ch. 2.1], since it effectively performs projected gradient descent on the loss function only looking at the sign of the gradient. Madry et al. [19] even argue, that PGD attacks are the strongest attacks utilizing first order information. To build a comparable and parameter-free baseline for evaluating adversarial accuracy, the parameter-free ensemble of five different attacks -including parameter-free versions of PGD attacks- *AutoAttack* has been developed by Croce and Hein [6] and is a metric frequently reported. In general, we also distinguish *white-box* attacks, that have full information of the network at their disposal and can, e.g., calculate gradients, and *black-box* attacks, that can only access input and output of the network.

Adversarial Defenses. An early proposed and still very effective defense is *adversarial training* [8], that adds the loss of adversarial examples as regularization to the clean training loss, i.e.

$$\alpha L(\theta, f_\theta(x), y) + (1 - \alpha) \max_{\|\delta\|<\epsilon} L(\theta, f_\theta(x + \delta), y) \tag{12}$$

where $\alpha \in [0, 1]$. The maximum in eq. (12) can be lower approximated by any adversarial attack, e.g. FGSM due to its fast computation. Instead of lower approximating the minimum, we can also upper approximate the minimum by, e.g., bound propagation yielding *certified training*. Models trained with certified training will be able to better certify robustness of samples using that upper bound. For a blended version of certified and adversarial training see the work of Müller et al. [21].

The development of defenses has started a cat-and-mouse game of developing defenses that defend against some attacks but are again broken by stronger attacks [5, Ch. 2]. This has led to evaluating adversarial robustness with *adaptive attacks*, that consider the full information about the model and its defense to develop attacks adapted to the defense. However, Tramer et al. [30] has found that even adversarial accuracy with adaptive attacks can be very misleading by developing well-thought adaptive attacks for several published defenses that

substantially reduced adversarial accuracy compared to the reported values of the adaptive attacks used in the papers.

Adversarial Purification Another idea is *adversarial purification*, that tries to remove adversarial noise with some denoiser before inputting the cleaned image into the classifier. Nie et al. [22] purified adversaries using pre-trained diffusion models by performing t steps forward and then t steps backward diffusion, where t is a hyperparameter.

0.3 Robust Out-Of-Distribution Detection

Background: out-of-distribution detection. When training models, we assume that the data to be modeled follows some (unknown) distribution $\mathcal{D}_{in}$. Hendrycks and Gimpel [13] found that samples from outside this distribution, called *out-of-distribution* (OOD) samples, can be assigned high probability by trained classifiers. To give an example, if we train a classifier to distinguish cats and dogs but feed it an image of a frog, the latter is out-of-distribution, but may be classified as cat or dog with high probability. Hence, the classifier does not know when it not knows. As a simple but effective method to detect OOD samples, Hendrycks and Gimpel [13] propose thresholding the *maximum softmax probability* (MSP), where a high MSP implies in-distribution and a lower MSP out-of-distribution. This follows the intuition that the probabilities being equal implies that the classifier is least "confident" about its prediction and thus, the sample might be OOD. Moreover, *outlier exposure* (OE) [14] proposes to add a regularization term computed from any OOD dataset $\mathcal{D}_{out}$ during training to obtain OOD aware classifiers, i.e.

$$\min_{\theta} \quad \mathbb{E}_{(x,y)\sim\mathcal{D}_{in}}[L(\theta, f_\theta(x), y) + \lambda \mathbb{E}_{x'\sim\mathcal{D}_{out}}[L_{OE}(\theta, f_\theta(x'), f_\theta(x), y)]] \tag{13}$$

where $\lambda > 0$ and L_{OE} depends on the OOD detection to be trained for, e.g. cross-entropy of $f_\theta(x')$ to the uniform distribution over all classes and neglecting $f_\theta(x), y$. In this case, we can write

$$\min_{\theta} \quad \mathbb{E}_{(x,y)\sim\mathcal{D}_{in}}[L(\theta, f_\theta(x), y)] + \lambda \mathbb{E}_{x'\sim\mathcal{D}_{out}}[L_{OE}(\theta, f_\theta(x'))]. \tag{14}$$

Many other methods to detect out-of-distribution samples and train out-of-distribution-aware classifiers have been proposed, for an overview see [33].

Methods for Robust OOD. It seems natural to ask whether OOD detection is robust, i.e., whether we can adversarially attack an in-distribution sample to make a classifier predict it as out-of-distribution or vice versa. *Adversarial confidence enhancing training* (ACET) [12] proposed the loss

$$\min_{\theta} \quad \mathbb{E}_{(x,y)\sim\mathcal{D}_{in}}[L(\theta, f_\theta(x), y)] + \lambda \mathbb{E}_{(x,y)\sim\mathcal{D}_{out}}[\max_{\|\delta\|<\epsilon} L_{OE}(\theta, f_\theta(x+\delta))], \tag{15}$$

where L_{OE} is the cross-entropy to the uniform distribution over all classes and the maximization is approximated by FGSM. In a slightly different setting,

where the soft classifier f has an additional output class $C+1$ for OOD samples, *adversarial training with informative outlier mining* (ATOM) [3] essentially used the same loss as ACET but with L_{OE} being the cross entropy loss with the target fixed to $C+1$. The main new idea is to only compute the regularization term in eq. (15) for a fraction of the outliers with the lowest predicted probability f_{C+1}, i.e. the most informative outliers, yielding substantial improvements. [4] combined adversarial training and outlier exposure to train a binary OOD detector G that classifies a sample with 1 for OOD and 0 else with

$$\min_{\theta} \quad \mathbb{E}_{x\sim\mathcal{D}_{in}}[\max_{\|\delta\|<\epsilon} \mathbb{1}(G(x+\delta)=0)] + \mathbb{E}_{x\sim\mathcal{D}_{out}}[\max_{\|\delta\|<\epsilon} \mathbb{1}(G(x+\delta)=1)], \quad (16)$$

but only obtained slight improvements while proposing no entirely new techniques. Note, that all of the above methods are empirically motivated and evaluated. Meinke, Bitterwolf, and Hein [20] combine a provably robust binary OOD detector with a robust in-distribution classifier to obtain a provably robust OOD-aware classifier.

0.4 Robustness to Semantic Perturbations

So far, we have looked at norm-bound adversaries, c.f. Definition 1. *Semantic perturbations* have no rigorous mathematical definition, but usually refer to perturbations that preserved human-observed semantics/meaning of the input [18, Ch. 2]. For images, examples include rotation, brightness, and Gaussian blur. Semantic perturbations usually have low-dimensional parameter spaces, e.g. for rotation we only have the rotation angle as parameter, and large changes in the L^p norms that are difficult be verified by the L^p-bound methods considered in Section 0.1. Moreover, they usually are of high practical interest.

Definition 5 (Certified Semantic Robustness)**.** [18, Ch. 3.1] Let $h : \mathcal{X} \to \mathcal{Y}$ be a hard classifier and $\mathcal{Z} \subset \mathbb{R}^p$ the parameter space of a semantic perturbation $\Phi : \mathcal{X} \times \mathcal{Z} \to \mathcal{X}$. We say $x \in \mathcal{X}$ is *robust* for $\mathcal{S} \subset \mathcal{Z}$ w.r.t. Φ if

$$h(x) = h(\Phi(x, \alpha)) \quad \forall \alpha \in \mathcal{S}. \quad (17)$$

Similarly as before, we can define empirical or *adversarial semantic robustness*. *Geometric transformations* are special semantic perturbations that sample new pixel values on a transformed grid from the original image, where pixel values are eventually interpolated, see **??**. They include rotation and scaling. Certifying semantic robustness for perturbations that include interpolation errors, e.g. rotation, is "hard both analytically and computationally" [18, p. 4]. Approaches include *DeepG* [1], that compute a convex relaxation of geometric perturbations that can be fed into any verifier, *TSS* [18], a unified framework for semantic perturbations, and *GSmooth* [11], that utilizes randomized smoothing. Effective and efficient geometric adversarial attacks can be generated by Wang et al. [31], that utilize an algorithm from Lipschitzian optimization theory.

0.5 The Bigger Picture: What do Neural Networks actually learn?

We have seen that we can easily fool a neural network with humanly inconceivable changes (adversaries) or inputs from completely different domains (OOD detection). Moreover, it has been observed that adversarial examples generalize across different architectures [8] and it has been proven that ReLU networks are always over-confident far from the training data [12]. On another note, it has been observed that stronger models with more parameters, like vision transformers, are more robust out-of-the-box learners [23] and that strong diffusion models can remove some adversarial attacks [22]. It seems that stronger models better "understand" the concepts that they are learning, however, we can still develop attacks that fool them. All of this hints at the fact, that we do not really understand what current SOTA models actually learn. Although vulnerability to adversarial attacks an OOD samples might decrease with larger model capacity, pointing out and combating these weaknesses is an important contribution to improving the security and reliability of neural networks.

0.6 Reading Recommendations to Get Started

The key papers I would recommend are; Szegedy et al. [29] to read the first paper that introduces adversarial attacks, Madry et al. [19] to get a good intuition of adversarial attacks and defenses in the saddle point formulation as well as adversarial training, Cohen, Rosenfeld, and Kolter [5] to understand the current SOTA method randomized smoothing, Hendrycks and Gimpel [13] to get an introduction to OOD and Hendrycks, Mazeika, and Dietterich [14] to understand OE as effective training procedure.

References

[1] Mislav Balunovic et al. "Certifying Geometric Robustness of Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[2] Nicholas Carlini et al. *(Certified!!) Adversarial Robustness for Free!* 2023. arXiv: 2206.10550 [cs.LG].

[3] Jiefeng Chen et al. *ATOM: Robustifying Out-of-distribution Detection Using Outlier Mining*. 2021. arXiv: 2006.15207 [cs.LG].

[4] Jiefeng Chen et al. *Robust Out-of-distribution Detection for Neural Networks*. 2021. arXiv: 2003.09711 [cs.LG].

[5] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. *Certified Adversarial Robustness via Randomized Smoothing*. 2019. arXiv: 1902.02918.

[6] Francesco Croce and Matthias Hein. *Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks*. 2020. arXiv: 2003.01690 [cs.LG].

[7] Mahyar Fazlyab et al. *Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks*. 2023. arXiv: 1906.04893 [cs.LG].

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[9] Henry Gouk et al. *Regularisation of Neural Networks by Enforcing Lipschitz Continuity*. 2020. arXiv: 1804.04368 [stat.ML].

[10] Sven Gowal et al. *On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models*. 2019. arXiv: 1810.12715 [cs.LG].

[11] Zhongkai Hao et al. *GSmooth: Certified Robustness against Semantic Transformations via Generalized Randomized Smoothing*. 2022. arXiv: 2206.04310 [cs.LG].

[12] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. *Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem*. 2019. arXiv: 1812.05720 [cs.LG].

[13] Dan Hendrycks and Kevin Gimpel. *A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks*. 2017. arXiv: 1610.02136 [cs.NE].

[14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. *Deep Anomaly Detection with Outlier Exposure*. 2019. arXiv: 1812.04606 [cs.LG].

[15] Matt Jordan and Alexandros G. Dimakis. *Exactly Computing the Local Lipschitz Constant of ReLU Networks*. 2021. arXiv: 2003.01219 [stat.ML].

[16] Guy Katz et al. *Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks*. 2017. arXiv: 1702.01135.

[17] Grigory Khromov and Sidak Pal Singh. *Some Fundamental Aspects about Lipschitz Continuity of Neural Network Functions*. 2023. arXiv: 2302.10886 [cs.LG].

[18] Linyi Li et al. *TSS: Transformation-Specific Smoothing for Robustness Certification*. 2021. arXiv: 2002.12398 [cs.LG].

[19] Aleksander Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2019. arXiv: 1706.06083 [stat.ML].

[20] Alexander Meinke, Julian Bitterwolf, and Matthias Hein. *Provably Robust Detection of Out-of-distribution Data (almost) for free*. 2022. arXiv: 2106.04260 [cs.LG].

[21] Mark Niklas Müller et al. *Certified Training: Small Boxes are All You Need*. 2023. arXiv: 2210.04871 [cs.LG].

[22] Weili Nie et al. *Diffusion Models for Adversarial Purification*. 2022. arXiv: 2205.07460 [cs.LG].

[23] Sayak Paul and Pin-Yu Chen. *Vision Transformers are Robust Learners*. 2021. arXiv: 2105.07581 [cs.CV].

[24] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. *Semidefinite relaxations for certifying robustness to adversarial examples*. 2018. arXiv: 1811.01057 [cs.LG].

[25] Hadi Salman et al. *Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers*. 2020. arXiv: 1906.04584 [cs.LG].

[26] Kevin Scaman and Aladin Virmaux. *Lipschitz regularity of deep neural networks: analysis and efficient estimation*. 2019. arXiv: 1805.10965 [stat.ML].

[27] Zhouxing Shi et al. *Efficiently Computing Local Lipschitz Constants of Neural Networks via Bound Propagation*. 2022. arXiv: 2210.07394 [cs.LG].

[28] Gagandeep Singh et al. "An Abstract Domain for Certifying Neural Networks". In: *Proc. ACM Program. Lang.* 3.POPL (2019). DOI: 10.1145/3290354. URL: https://doi.org/10.1145/3290354.

[29] Christian Szegedy et al. *Intriguing properties of neural networks*. 2014. arXiv: 1312.6199.

[30] Florian Tramer et al. *On Adaptive Attacks to Adversarial Example Defenses*. 2020. arXiv: 2002.08347 [cs.LG].

[31] Fu Wang et al. *Towards Verifying the Geometric Robustness of Large-scale Neural Networks*. 2023. arXiv: 2301.12456 [cs.LG].

[32] Greg Yang et al. *Randomized Smoothing of All Shapes and Sizes*. 2020. arXiv: 2002.08118 [cs.LG].

[33] Jingkang Yang et al. *Generalized Out-of-Distribution Detection: A Survey*. 2022. arXiv: 2110.11334 [cs.CV].

[34] Bohang Zhang et al. *Rethinking Lipschitz Neural Networks and Certified Robustness: A Boolean Function Perspective*. 2022. arXiv: 2210.01787 [cs.LG].