Adversarial Risk Bounds through Sparsity based Compression

Emilio Rafael Balda RWTH Aachen University
Niklas Koep RWTH Aachen University
Arash Behboodi RWTH Aachen University
Rudolf Mathar RWTH Aachen University

Abstract

Neural networks have been shown to be vulnerable against minor adversarial perturbations of their inputs, especially for high dimensional data under ℓ_∞ attacks. To combat this problem, techniques like adversarial training have been employed to obtain models that are robust on the training set. However, the robustness of such models against adversarial perturbations may not generalize to unseen data. To study how robustness generalizes, recent works assume that the inputs have bounded ℓ_2-norm in order to bound the adversarial risk for ℓ_∞ attacks with no explicit dimension dependence. In this work, we focus on ℓ_∞ attacks with ℓ_∞ bounded inputs and prove margin-based bounds. Specifically, we use a compression-based approach that relies on efficiently compressing the set of tunable parameters without distorting the adversarial risk. To achieve this, we apply the concept of effective sparsity and effective joint sparsity on the weight matrices of neural networks. This leads to bounds with no explicit dependence on the input dimension, neither on the number of classes. Our results show that neural networks with approximately sparse weight matrices not only enjoy enhanced robustness but also better generalization. Finally, empirical simulations show that the notion of effective joint sparsity plays a significant role in generalizing robustness to ℓ_∞ attacks[1].

[1]The implementations used in this work are available at `github.com/ebalda/adversarial-risk-bounds`

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

1 Introduction

In recent years, neural networks have been shown to be particularly vulnerable to maliciously designed perturbations of their inputs. Such perturbed inputs are known as adversarial examples and they are often only slightly distorted versions of the original inputs. For example, in image classification, adversarial examples have been shown to be indistinguishable from the original image to the human eye. This phenomena motivated several works aimed at understanding the nature of classifiers, and in particular neural networks, in the presence of adversarial examples. Initial works focused on the linearity (and non-linearity) of classifiers and its implications on the robustness of Deep Neural Networks (DNNs) against adversarial examples (Goodfellow et al. 2015; Tanay et al. 2016; Sabour et al. 2016). Subsequent works shed some light on the nature of adversarial examples by studying the properties of decision boundaries (A. Fawzi, Moosavi-Dezfooli, et al. 2016; Tanay et al. 2016; Rozsa, Gunther, et al. 2018; Rozsa, Günther, et al. 2016; Moosavi-Dezfooli et al. 2018), while others focused on the model capacity of neural networks in relation to the problem difficulty (A. Fawzi, O. Fawzi, et al. 2018; Kurakin et al. 2017; Madry et al. 2018). While these approaches contributed to understanding the nature of adversarial examples, they do not consider whether the robustness of classifiers against adversarial perturbations generalizes to unseen data.

If a classifier is robust to perturbations of the training set, can we guarantee that it will also be robust to perturbations of the test set? This question is not particularly new. The optimization community has studied this problem for quite some time. The work of Xu et al. 2008, studied robust regression in Lasso, while later work (Xu et al. 2009) obtained results for support vector machines. Other works considered the generalization properties of robust optimization in a distributional sense (Sinha et al. 2018), that is when adversarial examples are assumed to be samples from the worst possible distribution within a Wasserstein ball around the original one. These works provide algorithms for training various types of classifiers with robustness

guarantees. Regarding neural networks, for the case where no adversarial perturbations are present, there exists an extensive literature on their generalization properties. Many of these works are based on bounding the Rademacher complexity of the function class (Bartlett et al. 2017; Golowich et al. 2018; Neyshabur, Z. Li, et al. 2018; X. Li et al. 2018), while others make use of the PAC-Bayes framework (Neyshabur, Bhojanapalli, et al. 2017a; Neyshabur, Bhojanapalli, et al. 2017b; Nagarajan et al. 2019). There are other works which rely in different techniques, for instance, Arora et al. 2018 rely on compressing the weights of neural networks. Despite this knowledge, proving robustness guarantees for neural networks remained unstudied till recently. Initial works going into this direction studied neural networks in artificial scenarios. For instance, Attias et al. 2018 proved generalization bounds for the case when the adversary can modify a finite number of entries per input. Following this approach, Diochnos et al. 2018 showed that the number of flipped bits required to fool almost all inputs is less than $\mathcal{O}(\sqrt{n})$, for the case when the input is binary and uniformly distributed. As similar subsequent result (Mahloujifar et al. 2019) for binary inputs, proved the existence of polynomial-time attacks that find adversarial examples of Hamming distance $\mathcal{O}(\sqrt{n})$. Concurrently, the work of Schmidt et al. 2018 showed that the amount of data necessary to classify n-dimensional Gaussian data grows by a factor of $\sqrt{n}$ in the presence of an adversary. However, Cullina et al. 2018 showed that the Vapnik-Chervonenkis (VC)-dimension of linear classifiers does not increase in the adversarial setting. Additionally, they derived generalization guarantees for binary linear classifiers. Moreover, Montasser et al. 2019 showed that VC-classes are learnable in the adversarial setting, but only if one refrains from using standard empirical risk minimization approaches. Later works considered more general scenarios. Using a PAC-Bayes approach, Farnia et al. 2019 proved a generalization bound for neural networks under ℓ_2 attacks. However, deriving bounds for attacks with bounded ℓ_∞-norm (instead of ℓ_2-norm) is of particular interest, since most successful attacks in computer vision are of this type. In addition, such attacks tend to be more effective for scenarios where the input dimension is large, thus deriving generalization bounds without explicit dimension dependence is promising.

Now, let us overview recent works addressing the problem of proving generalization bounds for neural networks in the adversarial setting, where the attacker has bounded ℓ_∞ perturbations. Since these works are closely related to this work, we discuss them in more detail in the following list.

- Yin et al. 2019 bounded the Rademacher complexity for linear classifiers and neural networks in the adversarial setting. This lead to explicit bounds on the notion of adversarial risk for the linear classifier as well as neural networks. Nevertheless, such bound applied only to neural networks with one hidden layer and ReLU activations.
- Concurrent work from Khim et al. 2019 proved bounds on a surrogate of the adversarial test error. In that work, the authors use the so-called tree transform on the function class to derive their results. Under the assumption that the original inputs have ℓ_2 bounded norm, the authors proved generalization bounds with no explicit dimension dependence in the binary classification setting. Yet, the authors extend this to k-class classification by incurring an additional factor k on their bound.
- Later work from Tu et al. 2018 formulated generalization in the adversarial setting as a minimax problem. Their proposed framework is more general than previous ones in the sense that it can be applied to support vector machines and principal component analysis, as well as neural networks. Nonetheless, for neural networks this approach yielded a generalization bound with explicit dimension dependence.

One common assumption shared by these works is that the inputs come from a distribution with bounded ℓ_2-norm, which is a weaker notion than assuming ℓ_∞ bounded inputs.

1.1 Our Contributions

In this work, we study the problem of bounding the generalization error of multi-layer neural networks under ℓ_∞ attacks, where we assume that the original inputs have ℓ_∞ bounded norm. Using a compression approach, we obtain bounds with no explicit dependence on the input dimension or the number of classes. We summarize our contributions as follows.

- We prove generalization bounds in the presence of adversarial perturbations of bounded ℓ_∞-norm under the assumption that the input distribution has bounded ℓ_∞-norm as well. This is an improvement with respect to recent works where the input is assumed to be ℓ_2 bounded.
- We extend the compression approach from (Arora et al. 2018) by incorporating the notion of effective sparsity. Using this technique we prove that the capacity of neural networks, under adversarial perturbations, is bounded by the effective sparsity and effective joint sparsity of its weight matrices.

This result has no explicit dimension dependence, neither it depends on the number of classes. We show that approximately sparse weights not only improve robustness against ℓ_∞ bounded adversarial perturbations, but also provide better generalization as well.

- We corroborate our result with experiments on the MNIST and CIFAR-10 datasets, where the bound correlates with adversarial risk. We observe that adversarial training significantly decreases the bound, while standard training does not. Similarly, adversarial training seems to decrease both, effective sparsity and effective joint sparsity, as predicted by our result. Moreover, in these experiments, effective joint sparsity appears to be the dominant quantity in our bound. This shows the importance of effective joint sparsity for achieving generalization in the adversarial setting, a relation that was not discovered so far.

1.2 Notation

The notation $\mathcal{B}^n_{p,\varepsilon}$ is used to refer to an n-dimensional ℓ_p ball of size ε, that is the set $\mathcal{B}^n_{p,\varepsilon} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_p \leq \varepsilon\}$. We use the compact notation $\widetilde{\mathcal{O}}(n) := \mathcal{O}(n \log n)$ to ignore logarithmic factors.

2 Problem Setup

We start with the standard margin-based statistical learning framework. To that end, let $\mathcal{X}$ be the feature space, $\mathcal{Y}$ the label space, and $\mathcal{D} : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ a probability measure. In this work, it is assumed that all instances $\mathbf{x} \in \mathcal{X}$ have ℓ_∞-norm bounded by 1, that is $\mathcal{X} \subseteq \mathcal{B}^n_{\infty,1} \subset \mathbb{R}^n$. Without loss of generality, let the label space be $\mathcal{Y} = \{1, 2, \ldots, |\mathcal{Y}|\}$. Using these notions, a classifier is defined through its so called score function $f : \mathbb{R}^n \to \mathbb{R}^{|\mathcal{Y}|}$ such that the predicted label is $\operatorname{argmax}_{j \in \mathcal{Y}} f_j(\cdot)$, where $f_j(\cdot)$ is the j-th entry of $f(\cdot)$. Moreover, given an instance $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$, the classification margin is defined as

$$\ell(f; \mathbf{x}, y) = f_y(\mathbf{x}) - \max_{j \neq y} f_j(\mathbf{x}) \,.$$

In this manner, a positive margin implies correct classification. Then, for any distribution $\mathcal{D}$ the expected margin loss with margin $\gamma \geq 0$ is defined as

$$L_\gamma(f) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\ell(f; \mathbf{x}, y) \leq \gamma\right] .$$

We study the case where an adversary is present. This adversary has access to the input $\mathbf{x}$ and is allowed to add a perturbation $\boldsymbol{\eta}$ with ℓ_∞-norm bounded by some $\varepsilon \geq 0$ (*i.e.*, $\boldsymbol{\eta} \in \mathcal{B}^n_{p,\varepsilon}$) such that the classification margin is as small as possible. This perturbed input $\mathbf{x} + \boldsymbol{\eta}$ is usually known as an adversarial example. Furthermore, let us define the margin under adversarial perturbations as

$$\ell_\varepsilon(f; \mathbf{x}, y) = \inf_{\boldsymbol{\eta} \in \mathcal{B}^n_{\infty,\varepsilon}} \ell(f; \mathbf{x} + \boldsymbol{\eta}, y) \,.$$

This leads to the definition of adversarial margin loss, that is

$$L^\varepsilon_\gamma(f) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\ell_\varepsilon(f; \mathbf{x}, y) \leq \gamma\right] .$$

Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ be the training set composed of m instances drawn independently from $\mathcal{D}$. Using these instances we define $\widehat{L}^\varepsilon_\gamma(f) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}_{(\ell_\varepsilon(f;\mathbf{x}_i,y_i)\leq\gamma)}$ as the empirical estimate of $L^\varepsilon_\gamma(f)$, where $\mathbb{1}_{(\cdot)}$ denotes the indicator function. Note that $L^\varepsilon_0(f)$ and $\widehat{L}^\varepsilon_0(f)$ are the expected and training error under adversarial perturbations, respectively.

For many classifiers, such as deep neural networks, the score function f belongs to a complicated function class $\mathcal{F}$, which usually has more sample complexity than the size of the training set. Even without the presence of an adversary, it is challenging to bound the generalization error, given by the difference $L_0(f) - \widehat{L}_\gamma(f)$, of such function classes. The key idea behind the compression framework presented in (Arora et al. 2018) is to show that there exists a finite function class $\mathcal{G}$ with low sample complexity and a mapping that assigns a function $g \in \mathcal{G}$ to every $f \in \mathcal{F}$ such that the empirical loss is not severely degraded. This trick allows us to bound the generalization error using the sample complexity of $\mathcal{G}$ instead of $\mathcal{F}$. A drawback of this method is that we are only able to bound $L_0(g) - \widehat{L}_\gamma(f)$ instead of the original generalization error. Nevertheless, as the authors mentioned in (Arora et al. 2018), a similar issue is present as well in standard PAC-Bayes bounds, where the bound is on a noisy version of f. Moreover, the authors discuss some possible ways to solve this issue, but these approaches were left for future work. In this paper, we leverage such a compression framework by extending it to the case when an adversary is present. Our goal is to bound the generalization error under the presence of an adversary. We start by introducing some formal definitions and theorems, similar to the ones in (Arora et al. 2018). All proofs are deferred to the supplementary material.

Definition 2.1 ($(\gamma, \varepsilon, \mathcal{S})$-compressible)**.** *Given a set of parameter configurations $\mathcal{A}$, let $\mathcal{G}_\mathcal{A} = \{g_A | A \in \mathcal{A}\}$ be a set of parametrized functions g_A. We say that the score function $f \in \mathcal{F}$ is $(\gamma, \varepsilon, \mathcal{S})$-compressible through $\mathcal{G}_\mathcal{A}$ if*

$$\forall \mathbf{x} \in \mathcal{S}, y \in \mathcal{Y} : |\ell_\varepsilon(f; \mathbf{x}, y) - \ell_\varepsilon(g_A; \mathbf{x}, y)| \leq \gamma \,.$$

Theorem 2.2. *Given the finite sets $\mathcal{A}$ and $\mathcal{G}_{\mathcal{A}} = \{g_A | A \in \mathcal{A}\}$, if f is $(\gamma, \varepsilon, \mathcal{S})$-compressible via $\mathcal{G}_{\mathcal{A}}$ then there exists an $A \in \mathcal{A}$ such that with high probability*

$$L_0^{\varepsilon}(g_A) \leq \widehat{L}_{\gamma}^{\varepsilon}(f) + \mathcal{O}\left(\sqrt{\frac{\log|\mathcal{A}|}{m}}\right).$$

Corollary 2.2.1. *In the same setting of Theorem 2.2, if f is compressible only for a fraction $1-\delta$ of the training sample, then with high probability*

$$L_0^{\varepsilon}(g_A) \leq \widehat{L}_{\gamma}^{\varepsilon}(f) + \mathcal{O}\left(\sqrt{\frac{\log|\mathcal{A}|}{m}}\right) + \delta\,.$$

This main definition and following theorems are trivial extensions of the ones used in (Arora et al. 2018) to the adversarial setting. However, even for the linear classifier, the main technique used in that work for compressing f cannot be applied to the setup of this paper without incurring into explicit dimensionality dependencies in the resulting bounds. This will be explained in detail in the next section.

3 Main Results

In this section we introduce our main results. We start with linear classifiers on binary classification and move forward to multi-class neural networks.

3.1 Linear Classifier

We start with a linear classifier for binary labels. Assume that $\mathbf{x} \in \mathcal{B}_{\infty,1}^n$, $y \in \{1, 2\}$ and let $\mathbf{w} = (w_1, \ldots, w_n)^\top$ be a vector of weights of a linear classifier. Then, the score function of the linear classifier is given by

$$f_{\mathbf{w}}(\mathbf{x}) = \begin{pmatrix} 0 \\ \langle \mathbf{w}, \mathbf{x} \rangle \end{pmatrix}.$$

This simplifies the margin to $\ell(f; \mathbf{x}, y) = (2y - 3)\langle \mathbf{w}, \mathbf{x} \rangle$, which leads to

$$\ell_{\varepsilon}(f_{\mathbf{w}}; \mathbf{x}, y) = (2y-3)(\langle \mathbf{w}, \mathbf{x} \rangle - \varepsilon \|\mathbf{w}\|_1)\,.$$

Note that $(2y - 3) \in \{-1, +1\}$. The weight vector $\mathbf{w} \in \mathbb{R}^n$ of this classifier, with margin γ, can be compressed into another $\widehat{\mathbf{w}}$ such that both classifiers make the same predictions with reasonable probability (as we will see in Lemma 3.2). Given $\delta \in (0, 1]$, the compressed classifier $\widehat{\mathbf{w}}$ is constructed entry-wise in the following definition.

Definition 3.1 (CompressVector$(\gamma, \mathbf{w})$)**.** *Given $\mathbf{w} \in \mathcal{B}_{1,1}^n, \delta \in (0, 1], \gamma > 0$ and $\varepsilon > 0$, let us define the random mapping* CompressVector$(\gamma, \cdot)$ *which outputs $\widehat{\mathbf{w}} = (\widehat{w}_1, \ldots, \widehat{w}_n)^\top =$* CompressVector$(\gamma, \mathbf{w})$ *as follows*

$$\widehat{w}_i = z_i w_i / p_i$$

with

$$z_i \sim \text{Bern}(p_i) \;\; \textit{and} \; p_i = \frac{|w_i|}{\delta\gamma^2}(1+\varepsilon)^2\,,$$

where $\text{Bern}(p_i)$ *denotes the Bernoulli distribution with probability p_i.*

The following lemma shows that such classifier $\widehat{\mathbf{w}}$ outputs the same prediction as $\mathbf{w}$, with probability $1-\delta$, and has only $\mathcal{O}\left((\log n)(1+\varepsilon^2)/\delta\gamma^2\right)$ non-zero entries with high probability.

Lemma 3.2. *Given $\mathbf{w} \in \mathcal{B}_{1,1}^n, \delta \in (0, 1], \gamma > 0$ and $\varepsilon > 0$. If $\widehat{\mathbf{w}} =$* CompressVector$(\gamma, \mathbf{w})$ *then for any $\mathbf{x} \in \mathcal{B}_{\infty,1}^n$, $y \in \mathcal{Y}$ it holds*

$$\mathbb{P}_{\widehat{\mathbf{w}}}\left[\left|\ell_{\varepsilon}(f_{\mathbf{w}}; \mathbf{x}, y) - \ell_{\varepsilon}(f_{\widehat{\mathbf{w}}}; \mathbf{x}, y)\right| \geq \gamma\right] \leq \delta\,,$$

and the number of non-zero entries in $\widehat{\mathbf{w}}$ is less than $\mathcal{O}((\log n)(1+\varepsilon)^2/\delta\gamma^2)$ with high probability.

By discretizing $\widehat{\mathbf{w}}$, we obtain a compression setup that maps $\mathbf{w}$ into a discrete set but fails with probability δ. To that end, we handle discretization by clipping and then rounding in the following lemma.

Lemma 3.3. *Let us define*

- $\mathbf{w}'$ *component-wise as* $w_i' = w_i \mathbb{1}_{\left(|w_i| \geq \frac{\gamma}{4n(1+\varepsilon)}\right)}$,
- $\widetilde{\mathbf{w}} =$ CompressVector$(\gamma/2, \mathbf{w}')$,
- $\widehat{\mathbf{w}}$ *is obtained by rounding each entry of $\widetilde{\mathbf{w}}$ to the nearest multiple of $\frac{\gamma}{2n(1+\varepsilon)}$.*

Then, for all $\mathbf{x} \in \mathcal{B}_{\infty,1}^n$ and $y \in \mathcal{Y}$ we have that

$$\mathbb{P}_{\widehat{\mathbf{w}}}\left[\left|\ell_{\varepsilon}(f_{\mathbf{w}}; \mathbf{x}, y) - \ell_{\varepsilon}(f_{\widehat{\mathbf{w}}}; \mathbf{x}, y)\right| \geq \gamma\right] \leq \delta\,.$$

Therefore, we can apply Corollary 2.2.1 and choose $\delta = \left((1+\varepsilon)^2/\gamma^2 m\right)^{1/3}$, which yields a generalization bound of order $\widetilde{\mathcal{O}}\left(((1+\varepsilon)^2/\gamma^2 m)^{1/3}\right)$ as shown in the following theorem.

Theorem 3.4. *With high probability*

$$L_0^{\varepsilon}(f_{\widehat{\mathbf{w}}}) \leq \widehat{L}_{\gamma}^{\varepsilon}(f_{\mathbf{w}}) + \widetilde{\mathcal{O}}\left(\left(\frac{(1+\varepsilon)^2}{\gamma^2 m}\right)^{1/3}\right),$$

where $\widetilde{\mathcal{O}}(\cdot)$ ignores logarithmic factors.

This approach is fairly similar to the original one in the work of Arora et al. 2018, but the p_i values are chosen differently in order to deal with the new term $\varepsilon \|\mathbf{w}\|_1$ that appears in the margin's expression.

This result provides a dimension-free bound[2]. However, that bound scales with $m^{1/3}$ instead of $\sqrt{m}$, since

[2]Except for logarithmic terms.

the compression approach fails with probability δ. To tackle this issue, Arora et al. 2018 proposed a compression algorithm based on random projections. In their setup, this technique works due to a famous corollary of the Johnson-Lindenstrauss lemma that shows that we can construct random projections which preserve the inner product $\langle \mathbf{w}, \mathbf{x} \rangle$. In addition, since the Euclidean inner product can be induced by the ℓ_2-norm, the ℓ_2-norm of $\mathbf{w}$ is preserved as well. However, in this setup we would need a random projection that preserves $\|\mathbf{w}\|_1$ and $\langle \mathbf{w}, \mathbf{x} \rangle$ at the same time, which seems unattainable unless additional assumptions are made. Therefore, we propose to assume an effective sparsity bound on $\mathbf{w}$, which is defined as follows.

Definition 3.5 (Effective $\overline{s}$-sparsity)**.** *A vector $\mathbf{w} \in \mathbb{R}^n$ is effectively $\overline{s}$-sparse, with $\overline{s} \in [1, n]$, if*

$$\|\mathbf{w}\|_{1/2} \leq \overline{s}\, \|\mathbf{w}\|_1 \, .$$

Note that all s-sparse vectors[3] are effectively s-sparse as well, but not vice-versa. Assuming that $\mathbf{w}$ is effectively sparse allows us to compress it by simply setting its lowest entries to zero. The following lemma provides a tight bound on the error, in the ℓ_1 sense, that is caused by this process.

Lemma 3.6 ((Foucart et al. 2013): Theorem 2.5)**.** *For any $\mathbf{w} \in \mathbb{R}^n$ the following inequalities hold:*

$$\inf \{\|\mathbf{w} - \mathbf{z}\|_1 : \mathbf{z} \text{ is } s\text{-sparse}\} \leq \frac{1}{4s} \|\mathbf{w}\|_{1/2} \, ,$$
$$\inf \{\|\mathbf{w} - \mathbf{z}\|_\infty : \mathbf{z} \text{ is } s\text{-sparse}\} \leq \frac{1}{s} \|\mathbf{w}\|_1 \, .$$

In both cases, the infimum is attained when $\mathbf{z}$ is an s-sparse vector whose non-zero entries are the s-largest absolute entries of $\mathbf{w}$.

For any effectively $\overline{s}$-sparse classifier $\mathbf{w}$ with margin γ, this lemma allows us to compress it into a vector $\widehat{\mathbf{w}}$, with only $\mathcal{O}(\overline{s}(1+\varepsilon)/\gamma)$ non-zero entries, such that both classifiers assign the same label to any input. This is carried out by the following lemma.

Lemma 3.7. *Given an effectively $\overline{s}$-sparse vector $\mathbf{w} \in \mathcal{B}^n_{1,1}$, let us define $\mathbf{w}' \in \mathcal{B}^n_{1,1}$ as the s-sparse vector whose non-zero entries are the s-largest absolute entries of $\mathbf{w}$. In addition, the vector $\widehat{\mathbf{w}}$ is obtained by rounding each entry of $\mathbf{w}'$ to the nearest multiple of $\gamma/s(1+\varepsilon)$. If we choose $s = \overline{s}(1+\varepsilon)/2\gamma$ then*

$$\forall \mathbf{x} \in \mathcal{B}^n_{\infty,1}, y \in \mathcal{Y} : \quad \left|\ell_\varepsilon(f_{\mathbf{w}}; \mathbf{x}, y) - \ell_\varepsilon(f_{\widehat{\mathbf{w}}}; \mathbf{x}, y)\right| \leq \gamma \, .$$

Since this compression approach does not fail, we can discretize $\widehat{\mathbf{w}}$ and apply Theorem 2.2. This allows to prove the following generalization bound for the linear classifier in the presence of an adversary.

[3] A vector with no more than s non-zero entries is said to be s-sparse.

Theorem 3.8. *Let $\mathbf{w}$ be any linear classifier with $\|\mathbf{w}\|_{1/2} / \|\mathbf{w}\|_1 \leq \overline{s}$, and margin $\gamma > 0$ on the training set $\mathcal{S}$. Then, if $|\mathcal{S}| = m$, with high probability the adversarial risk is bounded by*

$$L_0^\varepsilon(f_{\widehat{\mathbf{w}}}) \leq \widehat{L}_\gamma^\varepsilon(f_{\mathbf{w}}) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{(1+\varepsilon)\overline{s}}{\gamma m}}\right) ,$$

where $\widetilde{\mathcal{O}}(\cdot)$ ignores logarithmic factors.

This result provides a bound with no explicit dimension dependence. Moreover, we observe that the presence of an adversary only increases the sample complexity by a factor $(1+\varepsilon)$. Note that, for dense $\mathbf{w}$'s like $\mathbf{w} = \mathbf{1}$, the dimension dependence is hidden inside $\overline{s}$. However, it has been observed that the dynamics of adversarial training lead to sparsity structures in the weights of linear classifiers and neural networks (Farnia et al. 2019; Guo et al. 2018; Madry et al. 2018; Wang et al. 2018). Our experimental findings, in Section 4, align with those results.

3.2 Neural Networks

Due to the ℓ_∞-norm bound on the perturbation $\boldsymbol{\eta}$, in this work the mixed $(1,\infty)$-norm of the weight matrices plays a central role. As an example, let us consider a linear classifier in multi-class classification, that is $f(\mathbf{x}) = \boldsymbol{W}^\top \mathbf{x}$. Then, a perturbation $\boldsymbol{\eta}$ can perturb any entry of the vector of score functions (*i.e.*, $f(\mathbf{x})$) at most

$$\sup_{\|\boldsymbol{\eta}\|_\infty \leq 1} \|\boldsymbol{W}^\top \boldsymbol{\eta}\|_\infty = \|\boldsymbol{W}\|_{1,\infty} \, .$$

The last equality comes from the properties of operator norms, see (Tropp 2004) for more details. Similar statements can be made for the layers of a neural network with 1-Lipschitz activation functions. Let us start by defining a d-layered fully connected neural network as

$$\mathbf{x}^i := \phi(\boldsymbol{W}^{i\top} \mathbf{x}^{i-1}) \, , \quad \forall i = 1, 2, \ldots, d \, , \tag{1}$$

where ϕ is a 1-Lipschitz activation function applied entry-wise, $\mathbf{x}^0 := \mathbf{x}$ and $f(\mathbf{x}) := \mathbf{x}^d$. Following the steps of Section 3.1, we now impose some conditions on $\boldsymbol{W}$ that allow us to efficiently compress it into another matrix $\widehat{\boldsymbol{W}}$ which belongs to a potentially small set. To that end, let us start by introducing the notion of effective joint sparsity.

Definition 3.9 (Effective joint sparsity)**.** *A matrix $\boldsymbol{W} \in \mathbb{R}^{n_1 \times n_2}$ is effectively joint $\overline{s}$-sparse, with $\overline{s} \in [1, n_2]$, if*

$$\|\boldsymbol{W}\|_{1,1} \leq \overline{s}\, \|\boldsymbol{W}\|_{1,\infty} \, .$$

Any matrix with s non-zero columns is effectively joint $\overline{s}$-sparse as well. Note that, given a matrix $\boldsymbol{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$, its effectively joint sparsity can be written as a the effective sparsity the vector $(\|\mathbf{w}_1\|_1, \ldots, \|\mathbf{w}_n\|_1)^\top$. A consequence of Lemma 3.6 is that we can compress effectively joint-sparse matrices by setting to zero their columns with lowest ℓ_1-norm. For example, assume that $\boldsymbol{W} \in \mathbb{R}^{n_1 \times n_2}$ is an effective joint $\overline{s}$-sparse matrix and that $\widehat{\boldsymbol{W}}$ is constructed by setting to zero all columns of $\boldsymbol{W}$ except for its s largest in the ℓ_1 sense. Then, by Lemma 3.6, we can bound the $\|\cdot\|_{1,\infty}$ error as

$$\left\|\boldsymbol{W} - \widehat{\boldsymbol{W}}\right\|_{1,\infty} \leq \frac{1}{s}\|\boldsymbol{W}\|_{1,1} \leq \frac{\overline{s}}{s}\|\boldsymbol{W}\|_{1,\infty} .$$

The resulting compressed matrix $\widehat{\boldsymbol{W}}$ would have only s non-zero columns instead of the original n_2. However, every column has potentially n_1 non-zero values. Therefore, in order to compress $\boldsymbol{W}$ further, we assume that each one of its columns has bounded effective sparsity as well. In summary, effective joint sparsity allows us to reduce the number of non-zero columns in a matrix, while effective sparsity of the columns allows us to reduce the number of non-zero elements that each of those columns may have. Finally, discretization is handled using a standard covering number argument. Putting all together into the following compression algorithm (Algorithm 1) allows us to map $\boldsymbol{W}$ into a discrete set while keeping the $\|\cdot\|_{1,\infty}$ error bounded.

By construction, using this algorithm guarantees that the error is bounded, as stated in the following lemma.

Lemma 3.10. *Let $\boldsymbol{W}$ be an effectively joint $\overline{s}_2$-sparse matrix with effectively $\overline{s}_1$-sparse columns, such that $\|\boldsymbol{W}\|_{1,\infty} \leq 1$. If $\widehat{\boldsymbol{W}} = \text{MatrixCompress}\,(\boldsymbol{W}, \gamma)$, then*

$$\left\|\boldsymbol{W} - \widehat{\boldsymbol{W}}\right\|_{1,\infty} \leq \gamma\,,$$

where $\widehat{\boldsymbol{W}}$ belongs to a discrete set $\mathcal{C}$ such that $\log|\mathcal{C}| \leq \widetilde{\mathcal{O}}\left(\|\boldsymbol{W}\|_{1,\infty}^2\,\overline{s}_1\overline{s}_2/\gamma^2\right)$.

From this lemma, we can see that the set of possible compressed matrices has reasonable size. Moreover, approximately sparse matrices can be compressed efficiently. This result leads us to the main contribution of this paper, which is stated in the following theorem.

Theorem 3.11. *Assume $\mathbf{x} \in \mathcal{B}^n_{\infty,1}$. Let $f_{\boldsymbol{W}}$ be a d-layer neural network with ReLU activations, and effectively joint $\overline{s}_2^j$-sparse weight matrices with effectively $\overline{s}_1^j$-sparse columns for $j = 1, \ldots, d$. Let us assume that the network is rebalanced so that $\left\|\boldsymbol{W}^1\right\|_{1,\infty} = \cdots = \left\|\boldsymbol{W}^d\right\|_{1,\infty} = 1$. Then, given $\gamma > 0$ and $\varepsilon < \gamma/4$, there exists a finite function set $\mathcal{G}$ composed of the functions $f_{\widehat{\boldsymbol{W}}}$ such that for any $f_{\boldsymbol{W}}$ the adversarial risk is bounded as*

$$L_0^\varepsilon(f_{\widehat{\boldsymbol{W}}}) \leq \widehat{L}_\gamma^\varepsilon(f_{\boldsymbol{W}}) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{d}{m}\left(\frac{1+\gamma/2-\varepsilon}{\gamma/2-2\varepsilon}\right)^2\left(\sum_{j=1}^d\sqrt{\overline{s}_1^j\overline{s}_2^j}\right)^2}\right)$$

with high probability.

Algorithm 1 MatrixCompress $(\cdot, \gamma)$

Require: $\gamma > 0$ and $\boldsymbol{W} \in \mathbb{R}^{n_1 \times n_2}$ with $\|\boldsymbol{W}\|_{1,\infty} = 1$, effectively $\overline{s}_1$-sparse columns and is effectively joint $\overline{s}_2$-sparse

Ensure:

$$\left\|\boldsymbol{W} - \widehat{\boldsymbol{W}}\right\|_{1,\infty} \leq \gamma\,,$$

where $\widehat{\boldsymbol{W}}$ belongs to a discrete set $\mathcal{C}$ such that $\log|\mathcal{C}| \leq \widetilde{\mathcal{O}}\left(\|\boldsymbol{W}\|_{1,\infty}^2\,\overline{s}_1\overline{s}_2/\gamma^2\right)$

Choose $s_1 = 3\|\boldsymbol{W}\|_{1,\infty}\,\overline{s}_1/4\gamma$ and $s_2 = 3\|\boldsymbol{W}\|_{1,\infty}\,\overline{s}_2/\gamma$

Let $\overline{\boldsymbol{W}} \in \mathbb{R}^{n_1 \times n_2}$ be obtained by setting to zero the columns of $\boldsymbol{W}$ except for the s_2 columns with largest ℓ_1 norm

Let $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{n_1 \times n_2}$ be constructed by keeping the s_1 largest values of every column of $\overline{\boldsymbol{W}}$ and setting to zero the other entries

Let $\mathcal{W}$ be the set all possible $\widetilde{\boldsymbol{W}}$

Let $\mathcal{C}$ be the covering set of $\mathcal{W}$ such that $\forall\widetilde{\boldsymbol{W}} \in \mathcal{W}, \exists\widehat{\boldsymbol{W}} \in \mathcal{C} : \left\|\widetilde{\boldsymbol{W}} - \widehat{\boldsymbol{W}}\right\|_{1,\infty} \leq \gamma/3$

Let $\widehat{\boldsymbol{W}} \in \mathcal{C}$ be the closest matrix in the $\|\cdot\|_{1,\infty}$ sense to $\widetilde{\boldsymbol{W}}$

Return: $\widehat{\boldsymbol{W}}$

This result proves a bound with no explicit dimension dependence, which is also independent from the number of classes. On the other hand, there seems to be an unavoidable dependence with $\sqrt{d}$. Yet, this dependence is also present in the bounds for multilayer neural networks, derived in related works (Khim et al. 2019; Tu et al. 2018). The rebalancing in Theorem 3.11 simplifies the proof by getting rid of the term $\prod_{j=1}^d\|\boldsymbol{W}^j\|_{1,\infty}$, which appears in other works such as (Khim et al. 2019). Note that, for ReLU networks, rebalancing does not affect the labels that $f_{\boldsymbol{W}}$ assigns to the inputs. However, by the definition of ℓ_ε and L_γ^ε, in practice, γ cannot be larger than $2\prod_{j=1}^d\|\boldsymbol{W}^j\|_{1,\infty}$. Then, the requirement $\varepsilon < \gamma/4$, in the setup of Theorem 3.11, limits the use of this result to $\varepsilon < 0.5$, or less for neural networks with smaller classification margins. Nonetheless, considering that $\mathbf{x} \in \mathcal{B}^n_{\infty,1}$, this requirement may not be extremely restrictive, since $\varepsilon = 0.5$ is a rather high value. Despite this shortcom-

Table 1: Comparison of the result in Theorem 3.11 with existing bounds for d-layered neural networks and $\|\boldsymbol{\eta}\|_\infty \leq \varepsilon$. We assume ReLU activations and rebalance the networks such that the bounds are simplified. The input $\mathbf{x} \in \mathbb{R}^n$ is assumed to belong to one of k classes and γ is the classification margin. The term $\lambda_f^+ > 0$ depends on m but may not vanish as m increases, while the term Λ_ε vanishes if $\varepsilon = 0$. For the precise definition of these two terms refer to (Tu et al. 2018).

		Rebalancing	Generalization Bound
Khim et al. 2019	$\|\mathbf{x}\|_2 \leq R$	$\left\|\boldsymbol{W}^j\right\|_{1,\infty} = 1$	$\widetilde{\mathcal{O}}\left(\sqrt{\frac{d}{m}k^2(R\max_j \left\|\boldsymbol{W}^j\right\|_{\mathrm{F}} + \varepsilon)^2}\right)$
Tu et al. 2018	$\|\mathbf{x}\|_2 \leq R$	$\left\|\boldsymbol{W}^j\right\|_2 = 1$	$\widetilde{\mathcal{O}}\left(\lambda_f^+\varepsilon + \sqrt{\frac{R^2}{m}\left(n^2\left(\sum_{j=1}^d \sqrt{\left\|\boldsymbol{W}^j\right\|_{\mathrm{F}}}\right)^2 + \Lambda_\varepsilon\right)^2}\right)$
ours	$\|\mathbf{x}\|_\infty \leq 1$	$\left\|\boldsymbol{W}^j\right\|_{1,\infty} = 1$	$\widetilde{\mathcal{O}}\left(\sqrt{\frac{d}{m}\left(\frac{1+\gamma-\varepsilon}{\gamma-2\varepsilon}\right)^2\left(\sum_{j=1}^d \sqrt{\left\|\boldsymbol{W}^j\right\|_{1/2,\infty}\left\|\boldsymbol{W}^j\right\|_{1,1}}\right)^2}\right)$

ing, Theorem 3.11 improves existing bounds in other aspects, as shown in Table 1. For instance, we observe that Theorem 3.11 improves existing works from requiring $\|\mathbf{x}\|_2 \leq R$ to $\|\mathbf{x}\|_\infty \leq 1$. Note that, in general, knowing that $\|\mathbf{x}\|_1 \leq 1$ only allows to bound R by $\sqrt{n}$, which would add an explicit dimension dependence on existing results. Moreover, our result does not depend on the number of classes as the work from Khim et al. 2019, nor it contains terms that do not vanish with m or an explicit dimension dependence (as Tu et al. 2018).

4 Experiments

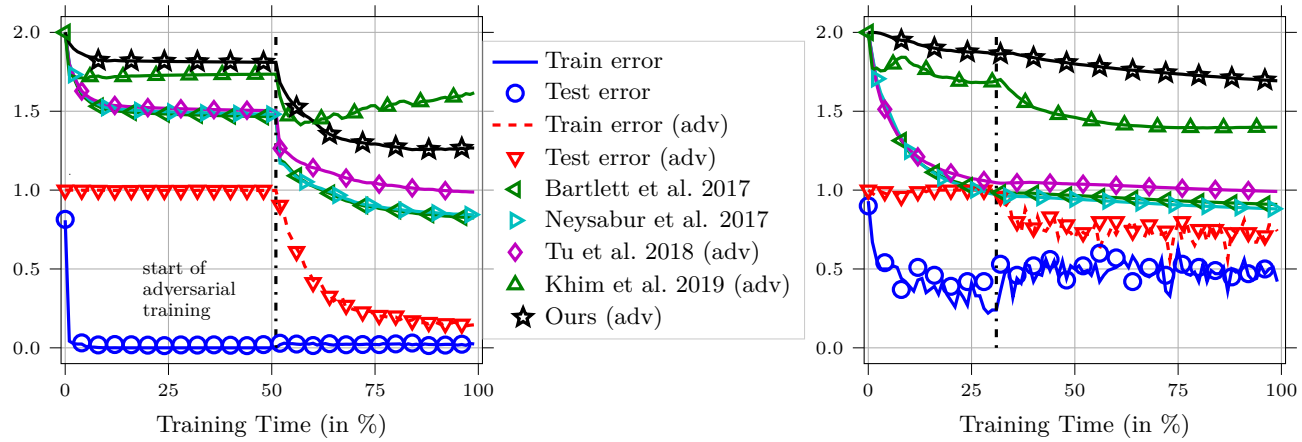
We conduct a experiment to corroborate our findings. To that end, we train a fully connected neural network of 3 layers with ReLU activations on the MNIST and CIFAR-10 datasets. After preprocessing, the inputs are 1024-dimensional vectors with ℓ_∞-norm bounded by one. We split training into two phases to distinguish between bounds that correlate with adversarial error and ones that correlate with standard error. Implementation details are deferred to the supplementary material.

In Figure 1(a), the network is first trained, on the MNIST dataset, without using adversarial examples. Then, after 50% of the training time, we start introducing adversarial examples to the training set. The same procedure is done in Figure 1(b) for the CIFAR-10 dataset, but adversarial examples are introduced after 33% of the training time. We observe that the adversarial error remains unchanged until adversarial training starts, this behavior correlates well with our result. Interestingly, classic (not adversarial) risk bounds (Bartlett et al. 2017; Neyshabur, Bhojanapalli, et al. 2017a) decrease significantly with adversarial training. This agrees with the intuition, from Theorem 3.11, that the effective sparsity induced by adversarial training improves generalization. Additionally, we compute the effective sparsity and effective joint sparsity of the weight matrices. In Figure 2, we see how these quantities correlate well with the adversarial risk as well. Interestingly, the effective joint sparsity of the weight matrices dominates our generalization bound, a property that was overlooked so far in this context. Overall, these findings show that inducing sparsity structures on the weight matrices does not only provide robustness, but also improves generalization of neural networks.

5 Conclusion

We have established adversarial risk bounds for DNNs under ℓ_∞ attacks. Our result has improved existing generalization bounds in terms of dependencies with the number of classes, the input dimension, and the norm of the inputs. This generalization bound has shown that effective sparsity does not only improve robustness, but results in better generalization of DNNs. As a result, this theoretical finding encourages the use of adversarial examples to improve the generalization capabilities of classifiers, even for applications where robustness to perturbations is not a major concern.

While it was already observed by Madry et al. 2018 that adversarial training leads to sparse weights, our empirical simulations found the notion of effective joint sparsity to be specially relevant for providing generalization guarantees under adversarial perturbations. This connection has not been discovered so far in existing works. As consequence, building regularization or optimization schemes based on this notion of joint sparsity seems to be a promising alternative for obtaining robust models, without adversarial training.



(a) MNIST (b) CIFAR-10

Figure 1: Test error (*i.e.*, 1 − accuracy) and rescaled generalization bounds during standard and adversarial training. For aesthetic reasons, these bounds are normalized to be between 0 and 2. Adversarial training improves our bound significantly, while standard training does not.

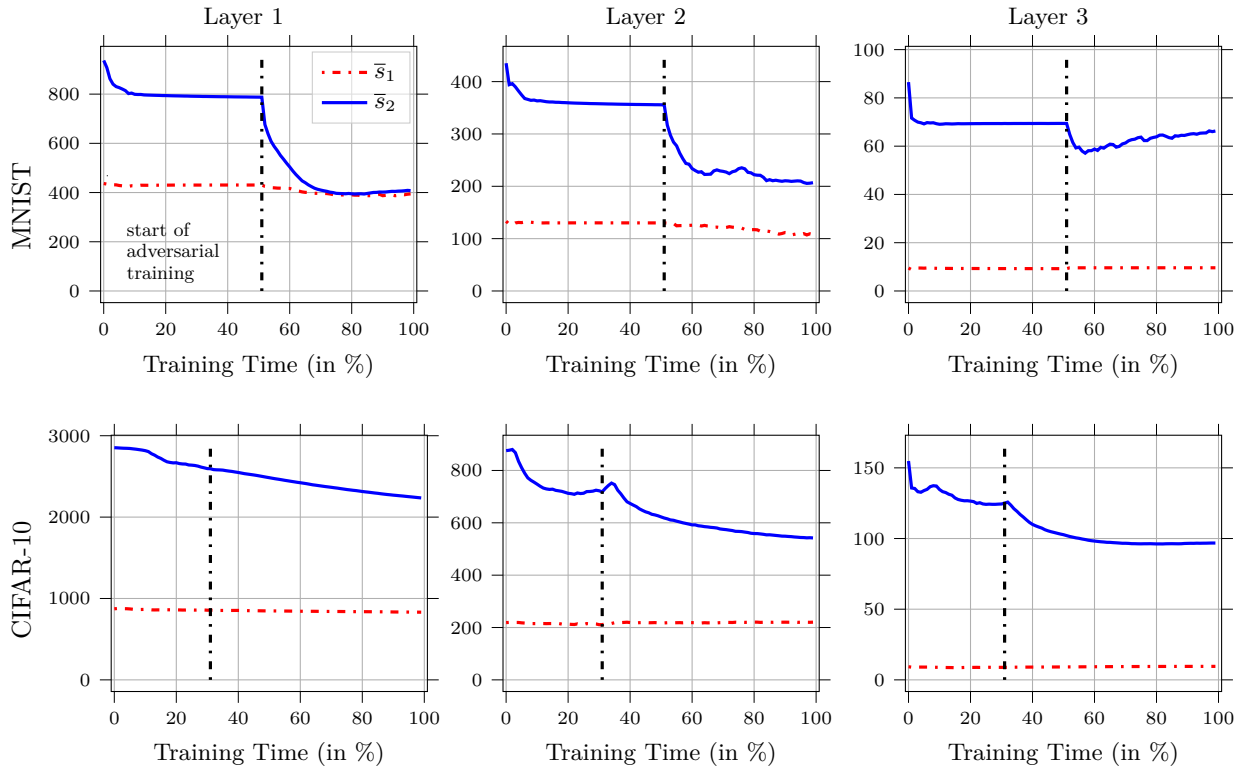


Figure 2: Experiment on the MNIST dataset. Effective sparsity and effective joint sparsity of the weight matrices, at every layer, of a vanilla neural network. These quantities tend to improve with adversarial training.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments, which led to improvements on the presentation of this work.

References

Arora, S. et al. (2018). "Stronger generalization bounds for deep nets via a compression approach". In: *International Conference on Machine Learning (ICML)*.

Attias, I., A. Kontorovich, and Y. Mansour (2018). "Improved Generalization Bounds for Robust Learning". In: *International Conference on Algorithmic Learning Theory (ALT)*.

Bartlett, P. L., D. J. Foster, and M. Telgarsky (2017). "Spectrally-normalized margin bounds for neural networks". In: *Neural Information Processing Systems (NIPS)*.

Cullina, D., A. N. Bhagoji, and P. Mittal (2018). "PAC-learning in the presence of adversaries". In: *Neural Information Processing Systems (NeurIPS)*.

Diochnos, D. I., S. Mahloujifar, and M. Mahmoody (2018). "Adversarial Risk and Robustness: General Definitions and Implications for the Uniform Distribution". In: *Neural Information Processing Systems (NeurIPS)*.

Farnia, F., J. M. Zhang, and D. Tse (2019). "Generalizable Adversarial Training via Spectral Normalization". In: *International Conference on Learning Representations (ICLR)*.

Fawzi, A., O. Fawzi, and P. Frossard (2018). "Analysis of classifiers' robustness to adversarial perturbations". In: *Machine Learning*.

Fawzi, A., S.-M. Moosavi-Dezfooli, and P. Frossard (2016). "Robustness of classifiers: from adversarial to random noise". In: *Neural Information Processing Systems (NIPS)*.

Foucart, S. and H. Rauhut (2013). *A Mathematical Introduction to Compressive Sensing. Applied and Numerical Harmonic Analysis*. Birkhäuser.

Golowich, N., A. Rakhlin, and O. Shamir (2018). "Size-Independent Sample Complexity of Neural Networks". In: *Conference on Learning Theory (COLT)*.

Goodfellow, I., J. Shlens, and C. Szegedy (2015). "Explaining and Harnessing Adversarial Examples". In: *International Conference on Learning Representations (ICLR)*.

Guo, Y. et al. (2018). "Sparse DNNs with Improved Adversarial Robustness". In: *Neural Information Processing Systems (NeurIPS)*.

Khim, J. and P.-L. Loh (2019). "Adversarial Risk Bounds via Function Transformation". In: *arXiv preprint arXiv:1810.09519*.

Kurakin, A., I. J. Goodfellow, and S. Bengio (2017). "Adversarial Machine Learning at Scale". In: *International Conference on Learning Representations (ICLR)*.

Li, X. et al. (2018). "On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond". In: *arXiv preprint arXiv:1806.05159*.

Madry, A. et al. (2018). "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations (ICLR)*.

Mahloujifar, S. and M. Mahmoody (2019). "Can Adversarially Robust Learning Leverage Computational Hardness?" In: *International Conference on Algorithmic Learning Theory (ALT)*.

Montasser, O., S. Hanneke, and N. Srebro (2019). "VC Classes are Adversarially Robustly Learnable, but Only Improperly". In: *arXiv preprint arXiv:1902.04217*.

Moosavi-Dezfooli, S.-M. et al. (2018). "Robustness of Classifiers to Universal Perturbations: A Geometric Perspective". In: *International Conference on Learning Representations (ICLR)*.

Nagarajan, V. and J. Z. Kolter (2019). "Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience". In: *International Conference on Learning Representations (ICLR)*.

Neyshabur, B., S. Bhojanapalli, et al. (2017a). "A pac-bayesian approach to spectrally-normalized margin bounds for neural networks". In: *International Conference on Learning Representations (ICLR)*.

Neyshabur, B., S. Bhojanapalli, et al. (2017b). "Exploring Generalization in Deep Learning". In: *Neural Information Processing Systems (NIPS)*.

Neyshabur, B., Z. Li, et al. (2018). "Towards Understanding the Role of Over-Parametrization in Generalization of Neural Networks". In: *International Conference on Learning Representations (ICLR)*.

Rozsa, A., M. Gunther, and T. E. Boult (2018). "Towards Robust Deep Neural Networks with BANG". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Rozsa, A., M. Günther, and T. E. Boult (2016). "Are Accuracy and Robustness Correlated". In: *IEEE International Conference on Machine Learning and Applications (ICMLA)*.

Sabour, S. et al. (2016). "Adversarial Manipulation of Deep Representations". In: *International Conference on Learning Representations (ICLR)*.

Schmidt, L. et al. (2018). "Adversarially Robust Generalization Requires More Data". In: *Neural Information Processing Systems (NeurIPS)*.

Sinha, A., H. Namkoong, and J. C. Duchi (2018). "Certifying Some Distributional Robustness with Principled Adversarial Training". In: *International Conference on Learning Representations (ICLR)*.

Tanay, T. and L. Griffin (2016). "A boundary tilting persepective on the phenomenon of adversarial examples". In: *arXiv preprint arXiv:1608.07690*.

Tropp, J. A. (2004). "Topics in sparse approximation". PhD thesis.

Tu, Z., J. Zhang, and D. Tao (2018). "Theoretical Analysis of Adversarial Learning: A Minimax Approach". In: *arXiv preprint arXiv:1811.05232*.

Wang, L. et al. (2018). "Adversarial Robustness of Pruned Neural Networks". In: *ICLR Workshop*.

Xu, H., C. Caramanis, and S. Mannor (2008). "Robust Regression and Lasso". In: *IEEE Transactions on Information Theory*.

Xu, H., C. Caramanis, and S. Mannor (2009). "Robustness and regularization of support vector machines". In: *Journal of Machine Learning Research*.

Yin, D., R. Kannan, and P. Bartlett (2019). "Rademacher Complexity for Adversarially Robust Generalization". In: *International Conference on Machine Learning (ICML)*.