Adversarial Risk Bounds through Sparsity based Compression

Emilio Rafael Balda Niklas Koep Arash Behboodi Rudolf Mathar
RWTH Aachen University RWTH Aachen University RWTH Aachen University 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 $\ell_\infty$ 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 $\ell_2$-norm in order to bound the adversarial risk for $\ell_\infty$ attacks with no explicit dimension dependence. In this work, we focus on $\ell_\infty$ attacks with $\ell_\infty$ 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 $\ell_\infty$ attacks.

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. 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.

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

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).
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. 2017b; 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 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 $O(\sqrt{n})$, for the case when the input is binary and uniformly distributed. As similar subsequent result (Mahlouji-far et al. 2019) for binary inputs, proved the existence of polynomial-time attacks that find adversarial examples of Hamming distance $O(\sqrt{n})$. 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 $\ell_2$ attacks. However, deriving bounds for attacks with bounded $\ell_\infty$-norm (instead of $\ell_2$-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 $\ell_\infty$ 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 complex-
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 $\ell_\infty$ 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 $\mathcal{B}_{p,\varepsilon}^n$ is used to refer to an $n$-dimensional $\ell_p$ ball of size $\varepsilon$, that is the set $\mathcal{B}_{p,\varepsilon}^n = \{x \in \mathbb{R}^n : \|x\|_p \leq \varepsilon\}$. We use the compact notation $\tilde{O}(n) := \mathcal{O}(n \log n)$ 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 $\mathcal{D} : \mathcal{X} \times \mathcal{Y} \rightarrow [0, 1]$ a probability measure. In this work, it is assumed that all instances $x \in \mathcal{X}$ have $\ell_\infty$-norm bounded by 1, that is $\mathcal{X} \subseteq \mathcal{B}_{\infty,1}^n \subset \mathbb{R}^n$. 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 $f : \mathbb{R}^n \rightarrow \mathbb{R}^{|\mathcal{Y}|}$ such that the predicted label is $\text{argmax}_{j \in \mathcal{Y}} f_j(\cdot)$, where $f_j(\cdot)$ is the $j$-th entry of $f(\cdot)$. Moreover, given an instance $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the classification margin is defined as

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

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

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

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

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

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

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

Let $\mathcal{S} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be the training set composed of $m$ instances drawn independently from $\mathcal{D}$. Using these instances we define $\hat{L}_\gamma^\varepsilon(f) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}_{(\ell_\varepsilon(f; x_i, y_i) \leq \gamma)}$ as the empirical estimate of $L_\gamma^\varepsilon(f)$, where $\mathbb{1}_{(\cdot)}$ denotes the indicator function. Note that $L_0^\varepsilon(f)$ and $\hat{L}_0^\varepsilon(f)$ 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 $L_0(f) - \hat{L}_\gamma(f)$, 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 $L_0(g) - \hat{L}_\gamma(f)$ 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 $\mathcal{G}_A = \{g_A | A \in \mathcal{A}\}$ 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 $\mathcal{G}_A$ if

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

$$L_0^*(g_A) \leq \hat{L}_2^*(f) + \mathcal{O}\left(\frac{\log |\mathcal{A}|}{m}\right).$$

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

$$L_0^*(g_A) \leq \hat{L}_2^*(f) + \mathcal{O}\left(\frac{\log |\mathcal{A}|}{m}\right) + \delta.$$

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 $x \in \mathbb{B}_n^1$, $y \in \{1, 2\}$ and let $w = (w_1, \ldots, w_n)\top$ be a vector of weights of a linear classifier. Then, the score function of the linear classifier is given by

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

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

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

Note that $(2y - 3) \in \{-1, +1\}$. The weight vector $w \in \mathbb{R}^n$ of this classifier, with margin $\gamma$, can be compressed into another $\tilde{w}$ 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 $\tilde{w}$ is constructed entry-wise in the following definition.

Definition 3.1 (CompressVector($\gamma, w$)). Given $w \in \mathbb{B}_n^1$, $\delta \in (0, 1), \gamma > 0$ and $\varepsilon > 0$, let us define the random mapping CompressVector($\gamma, \cdot$) which outputs $\tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_n)\top = \text{CompressVector}(\gamma, w)$ as follows

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

with

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

where Bern($p_i$) denotes the Bernoulli distribution with probability $p_i$.

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

Lemma 3.2. Given $w \in \mathbb{B}_n^1$, $\delta \in (0, 1), \gamma > 0$ and $\varepsilon > 0$. If $\tilde{w} = \text{CompressVector}(\gamma, w)$ then for any $x \in \mathbb{B}_n^1, y \in \mathcal{Y}$ it holds

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

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

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

Lemma 3.3. Let us define

- $w'$ component-wise as $w'_i = w_i \mathbb{1}_{\{|w_i| \geq \frac{\gamma}{\delta (1 + \varepsilon)}\}}$,
- $\tilde{w} = \text{CompressVector}(\gamma/2, w')$,
- $\tilde{w}$ is obtained by rounding each entry of $\tilde{w}$ to the nearest multiple of $\frac{\gamma}{\delta (1 + \varepsilon)}$.

Then, for all $x \in \mathbb{B}_n^1$ and $y \in \mathcal{Y}$ we have that

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

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

Theorem 3.4. With high probability

$$L_0(f_{\tilde{w}}) \leq \hat{L}_2(f_w) + \tilde{O}\left(\frac{(1 + \varepsilon)^2}{\gamma^2 m}\right)$$

where $\tilde{O}(\cdot)$ 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 $\varepsilon \|w\|_1$ 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 $\delta$. 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 $\langle w, x \rangle$. In addition, since the Euclidean inner product can be induced by the $\ell_2$-norm, the $\ell_2$-norm of $w$ is preserved as well. However, in this setup we would need a random projection that preserves $\|w\|_1$ and $\langle w, x \rangle$ at the same time, which seems unattainable unless additional assumptions are made. Therefore, we propose to assume an effective sparsity bound on $w$, which is defined as follows.

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

$$
\|w\|_{1/2} \leq \bar{s} \|w\|_1.
$$

Note that all $s$-sparse vectors are effectively $s$-sparse as well, but not vice-versa. Assuming that $w$ 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 $\ell_1$ sense, that is caused by this process.

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

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

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

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

For any effectively $\bar{s}$-sparse classifier $w$ with margin $\gamma$, this lemma allows us to compress it into a vector $\tilde{w}$, with only $O(\bar{s}(1 + \varepsilon)/\gamma)$ 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 $\bar{s}$-sparse vector $w \in \mathcal{B}_{\bar{s}, 1}^n$, let us define $w' \in \mathcal{B}_{\bar{s}}, 1$ as the $s$-sparse vector whose non-zero entries are the $s$-largest absolute entries of $w$. In addition, the vector $\tilde{w}$ is obtained by rounding each entry of $w'$ to the nearest multiple of $\gamma/s(1 + \varepsilon)$. If we choose $s = \bar{s}(1 + \varepsilon)/2\gamma$ then

$$
\forall x \in \mathcal{B}_{\infty, 1}^n, y \in \mathcal{Y} : \|\ell_z(f_w; x, y) - \ell_z(f_{w'}; x, y)\| \leq \gamma.
$$

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

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

$$
L_0^*(f_w) \leq \hat{L}_0^*(f_w) + \tilde{O}(\sqrt{(1 + \varepsilon)\log m}),
$$

where $\tilde{O}(\cdot)$ 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 $w$’s like $w = 1$, the dimension dependence is hidden inside $\bar{s}$. 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 $\ell_\infty$-norm bound on the perturbation $\eta$, 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 $f(x) = W^\top x$. Then, a perturbation $\eta$ can perturb any entry of the vector of score functions (i.e., $f(x)$) at most

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

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

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

where $\phi$ is a 1-Lipschitz activation function applied entry-wise, $x^0 := x$ and $f(x) := x^d$. Following the steps of Section 3.1 we now impose some conditions on $W$ that allow us to efficiently compress it into another matrix $\tilde{W}$ 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 $W \in \mathbb{R}^{n_1 \times n_2}$ is effectively joint $\bar{s}$-sparse, with $\bar{s} \in [1, n_2]$, if

$$
\|W\|_{1, 1} \leq \bar{s} \|W\|_{1, \infty}.
$$
Any matrix with $s$ non-zero columns is effectively joint $\pi$-sparse as well. Note that, given a matrix $W = (w_1, \ldots, w_n)$, its effectively joint sparsity can be written as a the effective sparsity the vector $(\|w_1\|_1, \ldots, \|w_n\|_1)$. A consequence of Lemma 3.6 is that we can compress effectively joint-sparse matrices by setting to zero their columns with lowest $\ell_1$-norm. For example, assume that $W \in \mathbb{R}^{n_1 \times n_2}$ is an effective joint $\pi$-sparse matrix and that $\tilde{W}$ is constructed by setting to zero all columns of $W$ except for its $s$ largest in the $\ell_1$ sense. Then, by Lemma 3.6, we can bound the $\|\cdot\|_{1,\infty}$ error as

$$\|W - \tilde{W}\|_{1,\infty} \leq \frac{1}{8} \|W\|_{1,1} \leq \frac{\pi}{\delta} \|W\|_{1,\infty}.$$

The resulting compressed matrix $\tilde{W}$ 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 $W$ 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 $W$ 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 $W$ be an effectively joint $\pi_2$-sparse matrix with effectively $\pi_1$-sparse columns, such that $\|W\|_{1,\infty} \leq 1$. If $\tilde{W} = \text{MatrixCompress}(W, \gamma)$, then

$$\|W - \tilde{W}\|_{1,\infty} \leq \gamma,$$

where $\tilde{W}$ belongs to a discrete set $C$ such that

$$\log |C| \leq \tilde{O} \left( \|W\|_{1,\infty}^2 \pi_1 \pi_2 / \gamma^2 \right).$$

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 $x \in \mathcal{B}^{d,\infty}_{\pi,1}$. Let $f_W$ be a d-layer neural network with ReLU activations, and effectively joint $\pi_j$-sparse weight matrices with effectively $\pi_i$-sparse columns for $j = 1, \ldots, d$. Let us assume that the network is relaxed so that $\|W_i\|_{1,\infty} = \cdots = \|W_d\|_{1,\infty} = 1$. Then, given $\gamma > 0$ and $\varepsilon < \gamma/4$, there exists a finite function set $\mathcal{G}$ composed of the functions $f_{W}$ such that for any $f_W$ the adversarial risk is bounded as

$$L_0^d(f_{\tilde{W}}) \leq \tilde{L}_0^d(f_W) + \tilde{O} \left( \frac{d}{m} \left( 1 + \frac{\gamma/2 - \varepsilon}{\gamma/2 - 2\varepsilon} \right)^2 \left( \sum_{j=1}^d \sqrt{\pi_1^j \pi_2^j} \right)^2 \right),$$

with high probability.

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 multi-layer 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 $\prod_{j=1}^d \|W_j\|_{1,\infty}$, which appears in other works such as (Khim et al. 2019). Note that, for ReLU networks, rebalancing does not affect the labels that $f_{W}$ assigns to the inputs. However, by the definition of $\ell_1$ and $L_1^\gamma$, in practice, $\gamma$ cannot be larger than $2 \prod_{j=1}^d \|W_j\|_{1,\infty}$. 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 $x \in \mathcal{B}^{d,\infty}_{\pi,1}$, this requirement may not be extremely restrictive, since $\varepsilon = 0.5$ is a rather high value. Despite this shortcom-

---

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

**Require:** $\gamma > 0$ and $W \in \mathbb{R}^{n_1 \times n_2}$ with $\|W\|_{1,\infty} = 1$, effectively $\pi_1$-sparse columns and is effectively joint $\pi_2$-sparse

**Ensure:**

$$\|W - \tilde{W}\|_{1,\infty} \leq \gamma,$$

where $\tilde{W}$ belongs to a discrete set $C$ such that

$$\log |C| \leq \tilde{O} \left( \|W\|_{1,\infty}^2 \pi_1 \pi_2 / \gamma^2 \right).$$

Choose $s_1 = 3 \|W\|_{1,\infty} \pi_1 / 4\gamma$ and $s_2 = 3 \|W\|_{1,\infty} \pi_2 / \gamma$

Let $W \in \mathbb{R}^{n_1 \times n_2}$ be obtained by setting to zero the columns of $W$ except for the $s_2$ columns with largest $\ell_1$ norm

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

Let $\tilde{W}$ be the set all possible $\tilde{W}$

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

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

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

<table>
<thead>
<tr>
<th>Rebalancing</th>
<th>Generalization Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$|x|_2 \leq R$</td>
<td>$|W_j|_{1,\infty} = 1$</td>
</tr>
<tr>
<td>$|x|_2 \leq R$</td>
<td>$|W_j|_2 = 1$</td>
</tr>
<tr>
<td>$|x|_\infty \leq 1$</td>
<td>$|W_j|_{1,\infty} = 1$</td>
</tr>
</tbody>
</table>

5 Conclusion

We have established adversarial risk bounds for DNNs under $\ell_\infty$ 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.
Figure 1: Test error (i.e., $1 - \text{accuracy}$) 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**


