SIAM J. MATH. DATA SCI.
Vol. 5, No. 4, pp. 920–948

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Adversarial Robustness of Sparse Local Lipschitz Predictors*

Ramchandran Muthukumar† and Jeremias Sulam‡

Abstract. This work studies the adversarial robustness of parametric functions composed of a linear predictor and a nonlinear representation map. Our analysis relies on *sparse local Lipschitzness* (SLL), an extension of local Lipschitz continuity that better captures the stability and reduced effective dimensionality of predictors upon local perturbations. SLL functions preserve a certain degree of structure, given by the sparsity pattern in the representation map, and include several popular hypothesis classes, such as piecewise linear models, Lasso and its variants, and deep feedforward ReLU networks. Compared with traditional Lipschitz analysis, we provide a tighter robustness certificate on the minimal energy of an adversarial example, as well as tighter data-dependent nonuniform bounds on the robust generalization error of these predictors. We instantiate these results for the case of deep neural networks and provide numerical evidence that supports our results, shedding new insights into natural regularization strategies to increase the robustness of these models.

Key words. adversarial robustness, robustness certificates, generalization guarantees, deep neural networks

MSC codes. 68Q32, 26A16, 68T07

DOI. 10.1137/22M1478835

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

1. Introduction.

During the past decade, deep learning has proven a successful model for a variety of real-world data-driven tasks, such as image classification [27], language modeling [19], and more. Modern deep learning architectures compose a learned representation map with a linear classifier, where the former can be feedforward, convolutional, recurrent, or attention maps, sequentially combined with nonlinear activation functions. Despite the strong empirical success of these models, a complete understanding of important properties, such as generalization [29] and robustness [34], is lacking. Importantly, state-of-the-art deep learning models are vulnerable to adversarially crafted small perturbations to input, called *adversarial examples* [53]. This vulnerability limits the deployment of these models in safety-critical tasks such as autonomous driving [12] and healthcare [30].

Adversarial examples are easy to generate, are hard to detect [26, 14], are able to be deployed in the physical world [21, 32], and are often transferable across predictors for the same task [33, 44]. This has led to significant empirical research to defend models against attacks [16, 60] as well subsequent work on improving these attacks to compromise the

*Received by the editors February 17, 2022; accepted for publication (in revised form) May 11, 2023; published electronically October 30, 2023.
https://doi.org/10.1137/22M1478835
Funding: This work was partially supported by NSF grant CCF 2007649, DARPA GARD award HR00112020004, and NSF TRIPODS CCF-1934979 via the MINDS Fellowship award.

†Department of Computer Science & Mathematical Institute of Data Science, Johns Hopkins University, Baltimore, MD 21218 USA (rmuthuk1@jhu.edu).
‡Department of Biomedical Engineering & Mathematical Institute of Data Science, Johns Hopkins University, Baltimore, MD 21218 USA (jsulam1@jhu.edu).

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

performance of defended models [3, 10, 39]. Several works have explored strategies to either improve or evaluate the robustness of modern deep learning models. For instance, *adversarial training* improves robustness by injecting adversarial attacks during the training phase [34, 46], while other works focus on certifying the level of corruption a model can withstand [23, 28, 45, 50, 59, 62].

Amidst the rapidly evolving empirical insights, there has been concurrent research aimed at providing theoretical guarantees on adversarial robustness for different hypothesis classes [4, 5, 31, 61]. Some of these works study the computational and statistical limits of adversarial attacks [11, 22, 35]. Others study trade-offs between robustness and natural (or benign) performance [15, 56, 63], provable guarantees for adversarial training [2, 65], or the analysis of optimal levels of provable adversarial defenses [17, 48].

In this work, we focus on two central questions of adversarial robustness: *certified robustness* and *robust generalization.* Our analysis for both of these questions will rely on the sensitivity of the model to changes in both its input and its parameters, a quantity that is naturally characterized by its Lipschitz constant. This view can be quite limited, however: for general nonlinear functions, such sensitivity to perturbations can greatly vary across the input space (for different samples) and across the hypothesis space (for different predictors). In this work, we show that local measures of sensitivity that additionally account for structural invariance in the outputs lead to tighter stability bounds and more informative results.

1.1. Outline. The paper is organized as follows. We elaborate on the formal task of supervised learning and adversarial robustness in section 2. Our contributions are summarized in section 3. The next two sections collect our main results, section 4 for certified robustness and section 5 for robust generalization. Finally, we demonstrate experimental results in section 6 and conclude with future directions in section 7.

1.2. Notation. Throughout this work, scalar quantities are denoted by lower case or upper case (not bold) letters, and vectors are denoted by bold lower case letters. Matrices are denoted by upper case letters: $\mathbf{W}$ is a matrix with *rows* $\mathbf{w}_i$. The Frobenius and operator norms are denoted by $\|\cdot\|_F$ and $\|\cdot\|_2$, respectively. For any matrix $\mathbf{W} \in \mathbb{R}^{p\times d}$ with *rows* $\mathbf{w}_i$, for $u, v \geq 1$, the group (u, v) norm[1] is defined as $\|\mathbf{W}\|_{u,v} := \left\| \left(\|\mathbf{w}_1\|_u, \ldots, \|\mathbf{w}_p\|_u \right) \right\|_v$. We informally refer to the Euclidean norm of a vector as its *energy.* We denote by $\succeq$ the elementwise $\geq$ operator for vectors. Sets and spaces are denoted by capital (and often calligraphic) letters, with the exception of the set $[p] = \{1, \ldots, p\}$. For a Banach space $\mathcal{W}$ embedded with norm $\|\cdot\|_{\mathcal{W}}$, we denote by $\mathcal{B}_r^{\mathcal{W}}(\mathbf{w})$ a bounded ball centered around point $\mathbf{w}$ with radius r. When describing a composition of affine functions, such as deep neural networks, $\mathbf{W}^k$ refers to the parameters corresponding to layer k. More generally, outside of norms, superscripts indicate layer index. We denote by $\mathcal{P}_I$ the index selection operator that restricts an input to the coordinates specified in the set I. For a vector $\mathbf{x} \in \mathbb{R}^d$ and $I \subset [d]$, $\mathcal{P}_I : \mathbb{R}^d \to \mathbb{R}^{|I|}$ is defined as $\mathcal{P}_I(\mathbf{x}) := \mathbf{x}[I]$. For a matrix $\mathbf{W} \in \mathbb{R}^{p\times d}$ and $I \subset [p]$, $\mathcal{P}_I(\mathbf{W}) \in \mathbb{R}^{|I|\times d}$ restricts $\mathbf{W}$ to the *rows* specified by I. For row and column index sets $I \subset [p]$ and $J \subset [d]$, $\mathcal{P}_{I,J}(\mathbf{W}) \in \mathbb{R}^{|I|\times|J|}$ restricts $\mathbf{W}$ to the corresponding submatrix.

[1] Defined over rows rather than columns.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

2. Robust supervised learning. Consider the task of multiclass classification with a bounded input space $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 \leq 1\}$ and labels $\mathcal{Y} = \{1, \ldots, C\}$ from an unknown distribution $\mathcal{D}_\mathcal{Z}$ over $\mathcal{Z} := (\mathcal{X} \times \mathcal{Y})$. We search for a hypothesis in $\mathcal{H} := \{h : \mathcal{X} \to \mathcal{Y}'\}$ that is an accurate predictor of label y given input $\mathbf{x}$. Note that $\mathcal{Y}$ and $\mathcal{Y}'$ need not be the same. In this work, we consider $\mathcal{Y}' = \mathbb{R}^C$ and consider the predicted label of the hypothesis h as $\hat{y}(\mathbf{x}) := \operatorname{argmax}_j [h(\mathbf{x})]_j$.[2] Throughout this work, the quality of a predictor $h \in \mathcal{H}$ at a sampled data point $\mathbf{z} = (\mathbf{x}, y) \in \mathcal{Z}$ is measured by a b-bounded L_{loss}-Lipschitz loss function $\ell : (\mathcal{H} \times \mathcal{Z}) \to [0, b]$. With these elements, the population risk of a hypothesis $R : \mathcal{H} \to [0, b]$ is the expected loss it incurs on a randomly sampled data point, $R(h) := \mathbb{E}_{\mathbf{z} \sim \mathcal{D}_\mathcal{Z}}[\ell(h, \mathbf{z})]$. The goal of supervised learning is to obtain a hypothesis with low risk. While the true distribution $\mathcal{D}_\mathcal{Z}$ is unknown, we assume access to a training set $\mathsf{S}_T = \{\mathbf{z}_1, \ldots, \mathbf{z}_m\}$ such that $\mathbf{z}_i = (\mathbf{x}_i, y_i) \overset{\text{i.i.d}}{\sim} \mathcal{D}_\mathcal{Z}$, and we instead minimize the *empirical risk,* i.e., the average loss on the training sample S_T, i.e., $\hat{R}(h) := \frac{1}{m} \sum_{i=1}^m \ell(h, (\mathbf{x}_i, y_i))$. We note two canonical choices of loss function for classification tasks: the zero-one loss $\ell^{[0/1]}$, and the margin loss ℓ^γ with threshold $\gamma > 0$ [38]. The zero-one loss is 1 for incorrect prediction and zero otherwise. The margin loss ℓ^γ is based on a margin operator $\mathcal{M} : \mathcal{Y}' \times \mathcal{Y} \to \mathbb{R}$, $\mathcal{M}(\mathbf{t}, y) := [\mathbf{t}]_y - \max_{j \neq y} [\mathbf{t}]_j$,[3]

$$\ell^\gamma(h, \mathbf{z}) := \min\left\{1, \max\left\{0, 1 - \frac{\mathcal{M}(h(\mathbf{x}), y)}{\gamma}\right\}\right\}.$$

The margin loss is 0 only for correct prediction with sufficient margin $\mathcal{M}(h(\mathbf{x}), y) \geq \gamma$. The margin loss with threshold $\gamma > 0$ is $\frac{2}{\gamma}$-Lipschitz w.r.t. change in predictor output [38].

The sensitivity of a predictor to changes in inputs or parameters is characterized by their global Lipschitz constants. For a predictor $h \in \mathcal{H}$, we let L_{inp} denote the maximal change in the output of the predictor h upon a change in its input. Similarly, for a suitable norm defined on the hypothesis class $\mathcal{H}$, we denote by L_{par} the global Lipschitz constant measuring the sensitivity of the output to changes in the parameters of the predictor. Formally, for all $\mathbf{x}, \tilde{\mathbf{x}} \in \mathcal{X}$ and all $\hat{h}, h \in \mathcal{H}$, we have that

$$\|h(\tilde{\mathbf{x}}) - h(\mathbf{x})\|_2 \leq \mathsf{L}_{\text{inp}} \|\tilde{\mathbf{x}} - \mathbf{x}\|_2, \qquad \left\|\hat{h}(\mathbf{x}) - h(\mathbf{x})\right\|_2 \leq \mathsf{L}_{\text{par}} \left\|\hat{h} - h\right\|_\mathcal{H}.$$

2.1. Adversarial robustness. To evade test-time adversarial attacks, we seek predictors that are robust to adversarial corruption in the bounded set $\mathcal{B}_\nu^\mathcal{X}(\mathbf{0}) := \{\boldsymbol{\delta} \in \mathcal{X} \mid \|\boldsymbol{\delta}\|_2 \leq \nu\}$. The robust loss $\ell_{\text{rob}}(h, \mathbf{z}) := \max_{\boldsymbol{\delta} \in \mathcal{B}_\nu^\mathcal{X}(\mathbf{0})} \ell(h, (\mathbf{x} + \boldsymbol{\delta}, y))$ captures the quality of a predictor h under an attack. We call the population (resp., empirical) risk evaluated on the robust loss the robust population (resp., empirical) risk,

$$R_{\text{rob}}(h) := \mathop{\mathbb{E}}_{\mathbf{z} \sim \mathcal{D}_\mathcal{Z}} \left[\ell_{\text{rob}}(h, \mathbf{z})\right], \quad \hat{R}_{\text{rob}}(h) := \frac{1}{m} \sum_{i=1}^m \ell_{\text{rob}}(h, \mathbf{z}_i).$$

In this case, the *robust* global Lipschitz constant $\mathsf{L}_{\text{par},\nu}$ measures parameter sensitivity on corrupted inputs,

$$\forall h, \hat{h} \in \mathcal{H}, \forall \mathbf{x} \in \mathcal{X}, \quad \max_{\boldsymbol{\delta} \in \mathcal{B}_\nu^\mathcal{X}(\mathbf{0})} \left\|\hat{h}(\mathbf{x} + \boldsymbol{\delta}) - h(\mathbf{x} + \boldsymbol{\delta})\right\| \leq \mathsf{L}_{\text{par},\nu} \left\|\hat{h} - h\right\|.$$

[2] The argmax here breaks ties deterministically.

[3] The predicted label is correct if $\mathcal{M}(h(\mathbf{x}), y) \geq 0$.

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Note that since $(\mathbf{x}+\boldsymbol{\delta})$ might not be in the original input domain $\mathcal{X}$, $\mathsf{L}_{\mathrm{par},\nu}$ can differ from $\mathsf{L}_{\mathrm{par}}$.

In this work, we focus on two central problems: *certified robustness,* providing a guarantee that a predictor h that correctly classifies an input $\mathbf{x}$, will not be changed if contaminated with an adversarial perturbation of bounded norm $\|\boldsymbol{\delta}\|_2 \leq \bar{r}(\mathbf{x})$; and *robust generalization,* seeking to understand when a predictor h learned on a collection of samples $\mathtt{S}$ can generalize to corrupted unseen data, i.e., when $R_{\mathrm{rob}}(h)$ is low. Last, while we focus on the widely studied constraint set of ℓ_2-bounded perturbations [53], most of the derived analysis is directly extendable to general ℓ_p norms, and we will comment on these whenever relevant.

2.2. Representation-linear hypothesis class. In this work, we consider a class of structured hypotheses $\mathcal{H}_{\mathcal{A},\mathcal{W}}$ called *representation-linear* hypotheses, with parameters $(\mathbf{A},\mathbf{W})$, where classification weights $\mathbf{A} \in \mathcal{A} \subset \mathbb{R}^{C\times p}$ act upon a learned representation map $\Phi_{\mathbf{W}} : \mathcal{X} \to \mathbb{R}^p$ with representation weights $\mathbf{W} \in \mathcal{W}$,

$$\mathcal{H}_{\mathcal{A},\mathcal{W}} := \{h_{\mathbf{A},\mathbf{W}} : \mathcal{X} \to \mathbb{R}^C,\ h_{\mathbf{A},\mathbf{W}}(\mathbf{x}) := \mathbf{A}\Phi_{\mathbf{W}}(\mathbf{x})\ \forall\ \mathbf{A} \in \mathcal{A} \text{ and } \mathbf{W} \in \mathcal{W}\}.$$

The parameters of each hypothesis $(\mathbf{A},\mathbf{W})$ are learned based on sample data $\mathtt{S}_T$. We assume that the representation space (image of $\Phi_{\mathbf{W}}$) is embedded with the Euclidean norm $\|\cdot\|_2$.[4] Naturally, each choice of a representation map Φ results in a corresponding hypothesis class. For the discussion in this paper, we assume a consistent choice of parameterizing functions in $\mathcal{H}$, and thus functions with different parameters are considered to be different. For the sake of simplicity, we denote $\mathcal{H} = \mathcal{H}_{\mathcal{A},\mathcal{W}}$, $h(\mathbf{x}) = h_{\mathbf{A},\mathbf{W}}(\mathbf{x})$, and $\Phi(\mathbf{x}) = \Phi_{\mathbf{W}}(\mathbf{x})$ when clear from the context. We discuss common representation maps.

Linear representations. The simplest case is that of representation maps that are linear; i.e., $\Phi(\mathbf{x}) = \mathbf{W}\mathbf{x}$ for some $\mathbf{W} \in \mathcal{W} \subset \mathbb{R}^{p\times d}$. One could require additional structure, such as taking $p < d$ and taking $\mathcal{W}$ as the set of projection matrices, thus computing low dimensional projections of the data. Linear low dimensional representations have proved to be beneficial in the context of adversarial robustness [6].

Supervised sparse coding. Here, for a dictionary $\mathbf{W} \in \mathcal{W}_{\mathrm{rip}}$,[5] the representation map Φ encodes input $\mathbf{x}$ as $\Phi(\mathbf{x}) \in \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x}-\mathbf{W}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$, the solution to a LASSO problem [54]. In this case, the representation Φ is nonlinear and encourages sparsity. This hypothesis class, denoted by $\mathcal{H}_{\mathrm{SSC}}$, is called task driven dictionary learning in [36], is frequently used in computer vision [20], and is analyzed in the context of adversarial robustness in [52].

Feedforward neural networks. The representation map implemented by a depth-$(K+1)$ neural network, $\Phi^{[K]}$, is a sequence of K affine maps composed with a nonlinear activation function. The most common choice for this activation is the rectifying linear unit, or ReLU, $\sigma : \mathbb{R} \to \mathbb{R}^{\geq 0}$, defined by $\sigma(x) = \max\{x, 0\}$, acting entrywise in an input vector. Each layer[6]

[4] We assume that the norm $\|\cdot\|_{\mathcal{A}}$ is submultiplicative and consistent with the Euclidean vector norm $\|\cdot\|_2$.

[5] $\mathcal{W}_{\mathrm{rip}}$ is the oblique manifold of matrices with unit-norm columns satisfying the restricted isometry property. A matrix $\mathbf{W}$ is RIP with constant η_s if this is the smallest value so that, for any s-sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^p$: $\|\boldsymbol{\alpha}\|_0 = s$, $\mathbf{W}$ is close to an isometry, i.e., $(1-\eta_s)\|\boldsymbol{\alpha}\|_2^2 \leq \|\mathbf{W}\boldsymbol{\alpha}\|_2^2 \leq (1+\eta_s)\|\boldsymbol{\alpha}\|_2^2$.

[6] Recall that superscripts in parameters and variables for neural network classes index the layer.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

has a weight $\mathbf{W}^k \in \mathcal{W}^k \subset \mathbb{R}^{d^k \times d^{k-1}}$ and bias $\mathbf{b}^k \in \mathfrak{B}^k \subset \mathbb{R}^{d^k}$. For this family of functions, the representation map $\Phi^{[K]}$ has parameters $\{\mathbf{W}^k, \mathbf{b}^k\}_{k=1}^K$ and is formally given by

$$\Phi^{[K]}(\mathbf{x}) := \sigma\left(\mathbf{W}^K \sigma\left(\mathbf{W}^{K-1} \cdots \sigma\left(\mathbf{W}^1 \mathbf{x} + \mathbf{b}^1\right) \cdots + \mathbf{b}^{K-1}\right) + \mathbf{b}^K\right).$$

We denote the hypothesis class $\mathcal{H}_{\mathrm{FNN,K}}$ with parameter space $\mathcal{A} \times \prod_{k=1}^K (\mathcal{W}^k \times \mathfrak{B}^k)$.

3. Contributions. In this work, we present results for robustness certificates and generalization guarantees based on a novel tool we call *sparse local Lipschitzness* (SLL). SLL measures the *local* sensitivity of a predictor h while additionally requiring the preservation of sparsity patterns in the representation Φ within a *sparse local radius*. The additional structural constraint enables us to express any SLL predictor by an equivalent *simpler* function with fewer *active* degrees of freedom or parameters. Importantly, our definition of SLL is flexible, allowing for any degree of sparsity levels and for a controlled, "tunable" trade-off between sparsity and local sensitivity, as measured by the sparse local radius. SLL predictors are a subclass of local Lipschitz predictors and include common representation-linear hypothesis classes such as the ones mentioned previously.

We present a certified radius for any SLL predictor w.r.t. input that improves standard local (and global) analysis (see Theorem 4.3 and Corollary 4.8). Compared to traditional Lipschitz analysis, we then demonstrate a tighter data-dependent nonuniform bound on the robust generalization error for predictors that are (robust) SLL w.r.t. parameters (see Theorem 5.2, Theorem SM1.7, Corollary 5.8). Our bounds depend mildly on the power of the adversary: the adversarial energy ν only impacts the *fast* term in the upper bound, i.e., $\mathcal{O}\left(\frac{\nu}{m}\right)$, an improvement from recent results [5, 61], which are $\mathcal{O}(\frac{\nu}{\sqrt{m}})$.

We instantiate these results for deep neural networks as a particular case. Figure 1 shows that for a trained neural network, at any test input, the number of active neurons in each layer is *at most* half the width. Here, as the corruption to inputs increases in energy, the number of neurons that flip activation states increases. Our analysis quantifies the stability of the strongly inactive neurons at each input to corruption in input and/or parameters. For

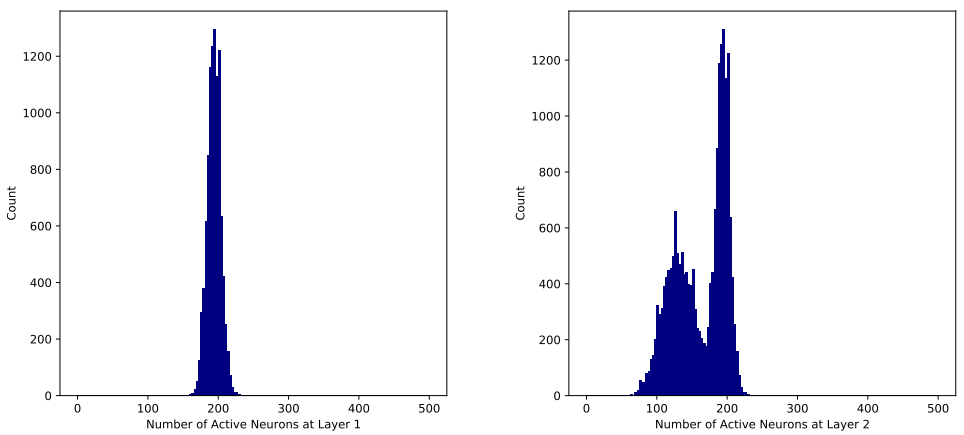


Figure 1. *Distribution of active neurons across validation data in each layer of a 2-layer feedforward network trained on MNIST.*

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

both certified robustness and robust generalization, this highlights the reduced dimensionality of the predictor to minor corruption. We show that the sparse local radius at each layer, for either input or parameter sensitivity, depends on the alignment between the pre-activation layer inputs and the rows of the layer weight matrix. For input sensitivity, we show that the sparse local Lipschitz scale of a depth-$(K+1)$ network is the product of operator norms of reduced linear maps at each layer, resulting in 2–3 fold improvement over the global Lipschitz constant for typical settings (see section 6). In turn, for parameter sensitivity, we demonstrate that the sparse local Lipschitz scale is given by an upper bound on the operator norms of any reduced linear maps by incorporating a measure of coherence between rows. This sparse local Lipschitz scale thus has a better dependence on depth than the global Lipschitz constant. Finally, we note that while other neural network architectures are part of the representation-linear hypothesis class, and can be shown to be sparse local Lipschitz functions (e.g., for convolutional networks), we refrain from instantiating our results for these architectures in this work and defer these interesting questions to future work.

4. Certified robustness. For a fixed predictor h (potentially data dependent), input $\mathbf{x}$, and perturbation $\boldsymbol{\delta}$, we denote by $\hat{y}(\mathbf{x})$ and $\hat{y}(\mathbf{x}+\boldsymbol{\delta})$ the predicted labels before and after corruption, respectively. A pointwise certified radius function $r_{\text{cert}} : \mathcal{X} \to \mathbb{R}^{\geq 0}$ is a guarantee that for any bounded perturbation $\boldsymbol{\delta}$ the predicted label remains unchanged, that is,

$$\|\boldsymbol{\delta}\|_2 \leq r_{\text{cert}}(\mathbf{x}) \implies \hat{y}(\mathbf{x}+\boldsymbol{\delta}) = \hat{y}(\mathbf{x}).$$

We develop a pointwise certified radius that relies on the local sensitivity of the predictor h.

4.1. Sparse local Lipschitz w.r.t. inputs. We start by characterizing a class of functions that preserve sparsity in their output for bounded perturbations to inputs. Throughout this manuscript, we refer to sparsity as *the number of zero entries* of a vector. We say that a vector $\mathbf{t} \in \mathbb{R}^d$ is s-sparse if it has an inactive set I of size s, that is, if there exists I so that $\mathcal{P}_I(\mathbf{t}) = \mathbf{0} \in \mathbb{R}^s$. Naturally, $\mathbf{t}$ is s-sparse only when $s \leq d - \|\mathbf{t}\|_0$. With these elements, we are now ready to define the sparse local Lipschitzness of a function.

Definition 4.1 (sparse local Lipschitzness w.r.t. input). *Let $\mathbf{x}$ and $\Phi(\mathbf{x})$ be s_{in}- and s_{out}-sparse, respectively, and let $\mathbf{s} = (s_{in}, s_{out})$. The representation map Φ is $\mathbf{s}$-sparse local Lipschitz at $\mathbf{x}$ (w.r.t. inputs) if there exist an index set I_{out} of size s_{out}, a local radius $r \geq 0$, and a Lipschitz scale $l \geq 0$ such that for any perturbed input $\tilde{\mathbf{x}} \in \mathcal{B}_r^{\mathcal{X}}(\mathbf{x})$ with a common inactive set I_{in} of size s_{in}, i.e., $\mathcal{P}_{I_{in}}(\tilde{\mathbf{x}}) = \mathcal{P}_{I_{in}}(\mathbf{x}) = \mathbf{0}$, one has that*

$$\|\Phi(\tilde{\mathbf{x}}) - \Phi(\mathbf{x})\|_2 \leq l\,\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \wedge \mathcal{P}_{I_{out}}(\Phi(\tilde{\mathbf{x}})) = \mathcal{P}_{I_{out}}(\Phi(\mathbf{x})) = \mathbf{0}.$$

In words, Φ is sparse local Lipschitz at $\mathbf{x}$ if, for points in a neighborhood of $\mathbf{x}$ that preserve a certain input sparsity pattern, the function is local Lipschitz *and* preserves a certain representation sparsity pattern. It is important to note that the sparse local sensitivity, i.e., the trio of index set I_{out}, radius r, and Lipschitz scale l, is dependent on the specific input $\mathbf{x}$ as well as the sparsity levels $\mathbf{s}$ which can range through all possible sizes of index sets in the inputs and in the representations, i.e., $\mathbf{s} \in \mathcal{S} := [d] \times [p]$. However, Φ will only be SLL at $\mathbf{x}$ for $\mathbf{s} \in \{0, \ldots, d - \|\mathbf{x}\|_0\} \times \{0, \ldots, p - \|\Phi(\mathbf{x})\|_0\}$. More generally, for a fixed representation map Φ,

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

we can define a local radius function $r_{\text{inp}} : \mathcal{X} \times \mathcal{S} \to \mathbb{R}^{\geq 0}$ and local Lipschitz scale functions,[7] $l_{\text{inp}} : \mathcal{X} \times \mathcal{S} \to \mathbb{R}^{\geq 0}$, and extend our definition.

Definition 4.2 (sparse local Lipschitz function). *We say that the representation map Φ is sparse local Lipschitz w.r.t. inputs if for any $\mathbf{x} \in \mathcal{X}$, for all appropriate*[8] *sparsity levels $\mathbf{s}$, Φ is an $\mathbf{s}$-sparse local Lipschitz at $\mathbf{x}$ with associated radius $r_{\text{inp}}(\mathbf{x}, \mathbf{s})$ and local Lipschitz scale $l_{\text{inp}}(\mathbf{x}, \mathbf{s})$.*

In this way, if Φ is sparse locally Lipschitz w.r.t. inputs, for an appropriate $\tilde{\mathbf{x}}$,

$$\|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \leq r_{\text{inp}}(\mathbf{x}, \mathbf{s}) \implies \|\Phi(\tilde{\mathbf{x}}) - \Phi(\mathbf{x})\|_2 \leq l_{\text{inp}}(\mathbf{x}, \mathbf{s}) \|\tilde{\mathbf{x}} - \mathbf{x}\|_2 .$$

Note that SLL representations Φ are also local Lipschitz with radius $r_{\text{inp}}(\mathbf{x}, \mathbf{0})$ and local Lipschitz scale $l_{\text{inp}}(\mathbf{x}, \mathbf{0})$. Additionally, if $r_{\text{inp}}(\mathbf{x}, \mathbf{0}) = \infty$ for all $\mathbf{x} \in \mathcal{X}$, then the representation is global Lipschitz with constant $\max_{\mathbf{x} \in \mathcal{X}} l_{\text{inp}}(\mathbf{x}, \mathbf{0})$. There could be multiple radius functions r_{inp} and l_{inp} that meet the requirements of Definitions 4.1 and 4.2. For each hypothesis class, we assume a fixed choice of these functions. For representation-linear hypothesis classes $\mathcal{H}$, it is natural to couple this notion of sensitivity with the classification weight $\mathbf{A}$. More precisely, under appropriate conditions on $\tilde{\mathbf{x}}$, we can have that

$$\|h(\tilde{\mathbf{x}}) - h(\mathbf{x})\|_2 = \|\mathbf{A}\Phi(\tilde{\mathbf{x}}) - \mathbf{A}\Phi(\mathbf{x})\|_2 \leq \|\mathbf{A}\|_2 \, l_{\text{inp}}(\mathbf{x}, \mathbf{s}) \cdot \|\tilde{\mathbf{x}} - \mathbf{x}\|_2 .$$

We are now ready to present our first result that develops a certified radius for SLL predictors.

Theorem 4.3. *Consider a predictor h with classification weight $\mathbf{A}$, sparse local Lipschitz representation map Φ, and fixed sparsity level $\mathbf{s} = (0, s_{\text{out}})$. Let h classify an input $\mathbf{x}$ as label $\hat{y}(\mathbf{x})$ with classification margin $\mathcal{M}(h(\mathbf{x}), \hat{y}(\mathbf{x})) > 0$. Upon bounded perturbations $\boldsymbol{\delta}$, the predicted label of the classifier remains unchanged, i.e., $\hat{y}(\mathbf{x} + \boldsymbol{\delta}) = \hat{y}(\mathbf{x})$ when the energy of the perturbation is below the certificate, $\|\boldsymbol{\delta}\|_2 \leq r_{\text{cert}}(\mathbf{x}, \mathbf{s})$, where*

$$r_{\text{cert}}(\mathbf{x}, \mathbf{s}) := \min \left\{ r_{\text{inp}}(\mathbf{x}, \mathbf{s}), \frac{\mathcal{M}(h(\mathbf{x}), \hat{y}(\mathbf{x}))}{2 \|\mathbf{A}\|_2 \cdot l_{\text{inp}}(\mathbf{x}, \mathbf{s})} \right\} .$$

Proof. Observe that the predicted labels remain unchanged when $\mathcal{M}(h(\mathbf{x} + \boldsymbol{\delta}), \hat{y}(\mathbf{x})) \geq 0$. Since the margin operator $\mathcal{M}(\cdot, j)$ is 2-Lipschitz in $\mathcal{Y}'$, thus $\mathcal{M}(h(\mathbf{x}), \hat{y}(\mathbf{x})) - \mathcal{M}(h(\mathbf{x} + \boldsymbol{\delta}), \hat{y}(\mathbf{x})) \leq 2 \|h(\mathbf{x} + \boldsymbol{\delta}) - h(\mathbf{x})\|_p$. On the other hand, note that for $\mathbf{s} = (0, s_{\text{out}})$ there are no sparsity constraints on the perturbed input. Hence,

$$\|\boldsymbol{\delta}\|_2 \leq r_{\text{inp}}(\mathbf{x}, \mathbf{s}) \implies \|h(\mathbf{x} + \boldsymbol{\delta}) - h(\mathbf{x})\|_2 \leq \|\mathbf{A}\|_2 \cdot l_{\text{inp}}(\mathbf{x}, \mathbf{s}) \|\boldsymbol{\delta}\|_2 .$$

As a result, $\mathcal{M}(h(\mathbf{x}), \hat{y}(\mathbf{x})) - \mathcal{M}(h(\mathbf{x} + \boldsymbol{\delta}), \hat{y}(\mathbf{x})) \leq 2 \|\mathbf{A}\|_2 \cdot l_{\text{inp}}(\mathbf{x}, \mathbf{s}) \|\boldsymbol{\delta}\|_2 .$ ∎

The first constraint on the certificate ensures that the perturbation is within the local radius. The second constraint, on the other hand, ensures that the effect of the perturbation

[7] For any input $\mathbf{x}$ and a choice of sparsity levels $\mathbf{s} = (s_{in}, s_{out}) \in \mathcal{S}$ such that $s_{in} > d - \|\mathbf{x}\|_0$ or $s_{out} > p - \|\Phi(\mathbf{x})\|_0$, for simplicity we let the corresponding radius $r_{\text{inp}}(\mathbf{x}, \mathbf{s}) := 0$ and the local Lipschitz scale $l_{\text{inp}}(\mathbf{x}, \mathbf{s}) := \infty$.

[8] Valid sparsity levels are in $\{0, \ldots, d - \|\mathbf{x}\|_0\} \times \{0, \ldots, p - \|\Phi(\mathbf{x})\|_0\}$.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

does not exceed the classification margin in the representation space. In the second constraint, the distance between the original and perturbed representation is bounded using the local Lipschitz scale. The input sparsity level s_{in} is naturally fixed to 0 to ensure that all ℓ_2 bounded perturbations are covered in the certified radius guarantee. This theorem encompasses local or global Lipschitz representations as the special case when $\mathbf{s} = \mathbf{0}$ (thus, it is more flexible than global analyses), and it is a generalization of the result in [57, Proposition 1].

4.1.1. Composition of sparse local Lipschitz functions. Useful representation maps can often be obtained as the composition of several intermediate maps, as in the case of feedforward neural networks and multilayered sparse coding [51]. More generally, consider K intermediate layer representation maps $\Phi^{(k)} : \mathbb{R}^{d^{k-1}} \to \mathbb{R}^{d^k}$ for $1 \le k \le K$, which are then composed to obtain $\Phi^{[K]}$,

$$\Phi^{[K]}(\mathbf{x}) := \Phi^{(K)} \circ \Phi^{(K-1)} \circ \cdots \circ \Phi^{(1)}(\mathbf{x}). \tag{4.1}$$

Let $(s^0, s^1, \ldots, s^K)$ denote an appropriate choice of sparsity level for each intermediate representation from $\mathbf{x}$ to $\Phi^{[k]}(\mathbf{x})$.[9] By defining the layerwise input-output sparsity levels $\mathbf{s}^{(k)} := (s^{k-1}, s^k)$ and the cumulative input-output sparsity levels $\mathbf{s}^{[k]} := (s^0, s^1, \ldots, s^k)$, we now show that one can compose sparse local Lipschitz functions to obtain a function of the same class.

Lemma 4.4. *Assume that each $\Phi^{(k)}$ in (4.1) is SLL w.r.t. inputs with local radius functions $r_{\text{inp}}^{(k)}$ and local Lipschitz scale $l_{\text{inp}}^{(k)}$. Then the composed representation maps $\Phi^{[k]}$ are also SLL w.r.t. inputs with local radius functions $r_{\text{inp}}^{[k]}$ and local Lipschitz scale $l_{\text{inp}}^{[k]}$ given by*

$$r_{\text{inp}}^{[k]}\left(\mathbf{x}, \mathbf{s}^{[k]}\right) := \min_{1 \le n \le k} \frac{r_{\text{inp}}^{(n)}\left(\Phi^{[n-1]}(\mathbf{x}), \mathbf{s}^{(n)}\right)}{l_{\text{inp}}^{[n-1]}\left(\mathbf{x}, \mathbf{s}^{[n-1]}\right)}, \qquad l_{\text{inp}}^{[k]}\left(\mathbf{x}, \mathbf{s}^{[k]}\right) := \prod_{n=1}^{k} l_{\text{inp}}^{(n)}\left(\Phi^{[n-1]}(\mathbf{x}), \mathbf{s}^{(n)}\right).$$

Proof sketch. For the base case $k = 1$, by definition, $r_{\text{inp}}^{[1]}(\mathbf{x}, \mathbf{s}^{[1]}) = r_{\text{inp}}^{(1)}(\mathbf{x}, \mathbf{s}^{(1)})$ and $l_{\text{inp}}^{[1]}(\mathbf{x}, \mathbf{s}^{[1]}) = l_{\text{inp}}^{(1)}(\mathbf{x}, \mathbf{s}^{(1)})$. Consider the case when $k = 2$, $\Phi^{[2]}(\mathbf{x}) := \Phi^{(2)} \circ \Phi^{(1)}(\mathbf{x})$. Consider a perturbation $\boldsymbol{\delta}$ in the initial input. By the definition of SLL, if $\|\boldsymbol{\delta}\|_2 \le r_{\text{inp}}^{[1]}(\mathbf{x}, \mathbf{s}^{[1]})$, then $\|\Phi^{[1]}(\mathbf{x}+\boldsymbol{\delta}) - \Phi^{[1]}(\mathbf{x})\|_2 \le l_{\text{inp}}^{[1]}(\mathbf{x}, \mathbf{s}^{[1]})\|\boldsymbol{\delta}\|_2$. Note that the perturbation in the first layer outputs $\Phi^{[1]}(\mathbf{x}+\boldsymbol{\delta}) - \Phi^{[1]}(\mathbf{x})$ is a perturbation in the second layer inputs to the map $\Phi^{(2)}$. Hence, if $\|\Phi^{[1]}(\mathbf{x}+\boldsymbol{\delta}) - \Phi^{[1]}(\mathbf{x})\|_2 \le r_{\text{inp}}^{(2)}(\Phi^{[1]}(\mathbf{x}), \mathbf{s}^{(2)})$, then

$$\begin{aligned}
\left\|\Phi^{(2)} \circ \Phi^{[1]}(\mathbf{x}+\boldsymbol{\delta}) - \Phi^{(2)} \circ \Phi^{[1]}(\mathbf{x})\right\|_2 &\le l_{\text{inp}}^{(2)}\left(\Phi^{[1]}(\mathbf{x}), \mathbf{s}^{(2)}\right) \left\|\Phi^{[1]}(\mathbf{x}+\boldsymbol{\delta}) - \Phi^{[1]}(\mathbf{x})\right\|_2 \\
&\le l_{\text{inp}}^{(2)}\left(\Phi^{[1]}(\mathbf{x}), \mathbf{s}^{(2)}\right) \cdot l_{\text{inp}}^{(1)}\left(\mathbf{x}, \mathbf{s}^{(1)}\right) \|\boldsymbol{\delta}\|_2 . \\
&=: l_{\text{inp}}^{[2]}\left(\mathbf{x}, \mathbf{s}^{[2]}\right) \|\boldsymbol{\delta}\|_2 .
\end{aligned}$$

One can similarly extend this logic to the case $k > 2$. If each of the intermediate representation maps is SLL w.r.t. inputs, then by appropriately weaving the sparsity levels at each layer we

[9] That is, one where $s^k \le d^k - \|\Phi^{[k]}(\mathbf{x})\|_0$.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

can show that the composed representation map is also SLL. The complete detailed proof by induction can be found in subsection SM1.2. ∎

As before, local Lipschitz and global Lipschitz compositions are special cases of the result above. One can readily use this local Lipschitz scale function $l_{\text{inp}}^{[K]}$ to obtain a certified radius as per Theorem 4.3 for functions that are compositions of sparse-local Lipschitz functions.

4.1.2. Optimal certified radius. For any sparse local Lipschitz predictors, the certified radius in Theorem 4.3 at the trivial choice of sparsity $r_{\text{cert}}(\mathbf{x}, \mathbf{0})$ is computed using $l_{\text{inp}}^{[K]}(\mathbf{x}, \mathbf{0})$, the global Lipschitz constant for $\Phi^{[K]}$. Increasing the sparsity vector $\mathbf{s}$ entrywise can result in a smaller local Lipschitz scale $l_{\text{inp}}^{[K]}(\mathbf{x}, \mathbf{s})$ at the expense of a smaller local radius $r_{\text{inp}}^{[K]}(\mathbf{x}, \mathbf{s})$. Hence, for a given input $\mathbf{x}$, the best robustness certificate $r^*(\mathbf{x})$ is generated by a specific choice of sparsity level $\mathbf{s}^*(\mathbf{x})$ that achieves a low Lipschitz scale in a sufficiently large neighborhood,

$$\mathbf{s}^*(\mathbf{x}) := \operatorname*{argmax}_{\mathbf{s}} r_{\text{cert}}(\mathbf{x}, \mathbf{s}), \quad r^*(\mathbf{x}) := r_{\text{cert}}(\mathbf{x}, \mathbf{s}^*(\mathbf{x})). \tag{4.2}$$

The complexity of this optimization problem will depend on the specific hypothesis class and on the function r_{cert}. For a composition of SLL predictors, the number of feasible sparsity levels in (4.2) is $\mathcal{O}(\prod_{k=1}^{K} d^k)$, exponential in the number of intermediate maps K. Thus, rather than search for the optimal sparsity vector $\mathbf{s}^*(\mathbf{x})$, we propose to approximate this solution (sometimes exactly) by a binary search over the space of the certified radius instead. More precisely, it is easy to see that $r^*(\mathbf{x}) \in [0, \|\mathbf{x}\|_2]$. Thus, if one has access to an algorithm $\mathcal{A}(\mathbf{x}, \nu)$ that can return, for any given input and energy level ν, an appropriate sparsity vector $\hat{\mathbf{s}}$ so that $\nu \leq r_{\text{inp}}^{[K]}(\mathbf{x}, \hat{\mathbf{s}})$, one can implement a binary search over $\nu \in [0, \|\mathbf{x}\|_2]$ by checking whether

$$\nu \leq \frac{\mathcal{M}(h(\mathbf{x}), \hat{y}(\mathbf{x}))}{2\,\|\mathbf{A}\|_2 \cdot l_{\text{inp}}^{[K]}(\mathbf{x}, \hat{\mathbf{s}})}. \tag{4.3}$$

If this is satisfied, such a level of sparsity is safe. This allows us to carry out a binary refinement over ν until a tolerance level, *tol*, is satisfied, reducing the complexity of this search to $\mathcal{O}(\log_2(\frac{\|\mathbf{x}\|_2}{tol}))$.[10] Naturally, the quality of this solution depends on the algorithm $\mathcal{A}(\mathbf{x}, \nu)$. We will see that for functions that satisfy a notion of *monotonicity,* such an algorithm can be easily instantiated and can be, in a specific sense, optimal.

Definition 4.5 (monotone ordering). *A sparse local Lipschitz representation map Φ with radius function r_{inp} and Lipschitz scale l_{inp} is said to have monotone ordering if the following hold:*

$$\textit{Lipschitz condition: } (s_1^{in}, s_1^{out}) \preceq (s_2^{in}, s_2^{out}) \implies l_{\text{inp}}(\mathbf{x}, (s_2^{in}, s_2^{out})) \leq l_{\text{inp}}(\mathbf{x}, (s_1^{in}, s_1^{out})), \tag{4.4}$$

$$\textit{radius condition 1: } s_1^{in} \leq s_2^{in}\ \forall\, s \in [d^{out}],\ r_{\text{inp}}(\mathbf{x}, (s_1^{in}, s)) \leq r_{\text{inp}}(\mathbf{x}, (s_2^{in}, s)), \tag{4.5}$$

$$\textit{radius condition 2: } s_1^{out} \leq s_2^{out}\ \forall\, s \in [d^{in}],\ r_{\text{inp}}(\mathbf{x}, (s, s_1^{out})) \geq r_{\text{inp}}(\mathbf{x}, (s, s_2^{out})). \tag{4.6}$$

[10] Note that we suppress the complexity of the algorithm $\mathcal{A}(\mathbf{x}, \nu)$, which will depend on the specific hypothesis class and refer simply to the search complexity.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Algorithm 4.1. $\mathcal{A}(\mathbf{x}, \nu)$ to generate valid sparse vector $\hat{\mathbf{s}}$ for $\Phi^{[K]} = \Phi^{(K)} \circ \cdots \circ \Phi^{(1)}$.

Require : Input $\mathbf{x}$ and perturbation energy level ν.
Require : Intermediate sparse local radius $r_{\text{inp}}^{(k)}$ and local Lipschitz scale $l_{\text{inp}}^{(k)}$ $\forall\, k \in [K]$.
Ensure : Sparsity vector $\hat{\mathbf{s}}$ such that $\nu \le r_{\text{inp}}^{[K]}(\mathbf{x}, \hat{\mathbf{s}})$.
Initialize : $\hat{\mathbf{s}} := \{0, \ldots, 0\} \in \mathbb{R}^{K+1}$.
Initialize : Perturbation level, $\hat{\nu}^0 := \nu$.
for k = 1 to K **do**
  $\hat{s}^k \leftarrow$ maximum s such that $\hat{\nu}^{k-1} \le r_{\text{inp}}^{(k)}\left(\Phi^{[k-1]}(\mathbf{x}), \{\hat{s}^{k-1}, s\}\right)$.
  $\hat{\nu}^k \leftarrow \hat{\nu}^{k-1} \cdot l_{\text{inp}}^{(k)}\left(\Phi^{[k-1]}(\mathbf{x}), \{\hat{s}^{k-1}, \hat{s}^k\}\right)$.
end for
Return : $\hat{\mathbf{s}} = \{0, \hat{s}^1, \ldots, \hat{s}^K\}$.

Algorithm 4.1 implements the rule $\mathcal{A}(\mathbf{x}, \nu)$ described above. This algorithm is correct, as we now make precise.

Lemma 4.6. *The sparsity vector $\hat{\mathbf{s}}$ generated by Algorithm 4.1 is correct, i.e., $r_{\text{inp}}^{[K]}(\mathbf{x}, \hat{\mathbf{s}}) \ge \nu$. If $\Phi^{[K]}$ is such that each intermediate map $\Phi^{(k)}$ has monotone ordering as per Definition 4.5, then Algorithm 4.1 is maximal; i.e., for any sparsity vector $\bar{\mathbf{s}}$, $\nu \le r_{\text{inp}}^{[K]}(\mathbf{x}, \bar{\mathbf{s}}) \implies \bar{\mathbf{s}} \preceq \hat{\mathbf{s}}$. Additionally, under monotone ordering, if ν is deemed unsafe, i.e., (4.3) is not satisfied, then there exists no sparsity vector $\bar{\mathbf{s}}$ such that $\nu \le r_{\text{cert}}(\mathbf{x}, \bar{\mathbf{s}})$.*

Proof. Define $\hat{\mathbf{s}}$ and $\hat{\nu}^k := \nu \cdot l_{\text{inp}}^{[k]}(\mathbf{x}, \hat{\mathbf{s}})$ as in Algorithm 4.1. For each layer k, the choice of $\hat{s}^k$ ensures that $\hat{\nu}^{k-1} = \nu \cdot l_{\text{inp}}^{[k-1]}(\mathbf{x}, \{0, \hat{s}^1, \ldots, \hat{s}^{k-1}\}) \le r_{\text{inp}}^{[k]}(\mathbf{x}, \{0, \hat{s}^1, \ldots, \hat{s}^k\})$, thus ensuring correctness. Consider any sparsity vector $\bar{\mathbf{s}}$ such that $\nu \le r_{\text{inp}}^{[K]}(\mathbf{x}, \bar{\mathbf{s}})$, and let $\bar{\nu}^k := \nu \cdot l_{\text{inp}}^{[k]}(\mathbf{x}, \bar{\mathbf{s}})$. We aim to show that $\bar{\mathbf{s}} \le \hat{\mathbf{s}}$ necessarily. For $k = 1$, $\nu \le r_{\text{inp}}^{[1]}(\mathbf{x}, \{0, \bar{s}^1\})$, and by construction,

$$\hat{s}^1 := \max s \text{ such that } \nu \le r_{\text{inp}}^{(1)}(\mathbf{x}, \{0, s\}).$$

Hence, $\bar{s}^1 \le \hat{s}^1$. Furthermore, by the monotone ordering assumption,

$$\hat{\nu}^1 \overset{\text{(a)}}{\le} \bar{\nu}^1 \overset{\text{(b)}}{\le} r_{\text{inp}}^{(2)}\left(\Phi^{[1]}(\mathbf{x}), \{\bar{s}^1, \bar{s}^2\}\right) \overset{\text{(c)}}{\le} r_{\text{inp}}^{(2)}\left(\Phi^{[1]}(\mathbf{x}), \{\hat{s}^1, \bar{s}^2\}\right). \tag{4.7}$$

In the above statement, (a) follows from monotone ordering of the Lipschitz scale, (b) follows from assumption on $\bar{\mathbf{s}}$, and (c) follows from monotone ordering of the radius w.r.t. the input sparsity level. Further, $\hat{s}^2$ is defined as the maximum sparsity level such that

$$\hat{s}^2 := \max s \text{ such that } \hat{\nu}^1 \le r_{\text{inp}}^{(2)}\left(\Phi^{[1]}(\mathbf{x}), \{\hat{s}^1, s\}\right).$$

From (4.7), $\bar{s}^2$ is also feasible for the above optimization, and hence $\bar{s}^2 \le \hat{s}^2$. One can repeat these arguments until layer K to show that $\bar{\mathbf{s}} \le \hat{\mathbf{s}}$. Hence, Algorithm 4.1 chooses the maximal sparsity vector. Now consider the set of sparsity vectors $\mathcal{S}_{\text{good}}$ such that $\nu \le r_{\text{inp}}^{[K]}(\mathbf{x}, \mathbf{s})$. If there exists a witness vector $\mathbf{s}$ such that $r_{\text{cert}}(\mathbf{x}, \mathbf{s}) \ge \nu$, then certainly $\mathbf{s} \in \mathcal{S}_{\text{good}}$. We have just shown that for all $\mathbf{s} \in \mathcal{S}_{\text{good}}$, we have $\mathbf{s} \preceq \hat{\mathbf{s}}$, and hence by the monotone ordering property,

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

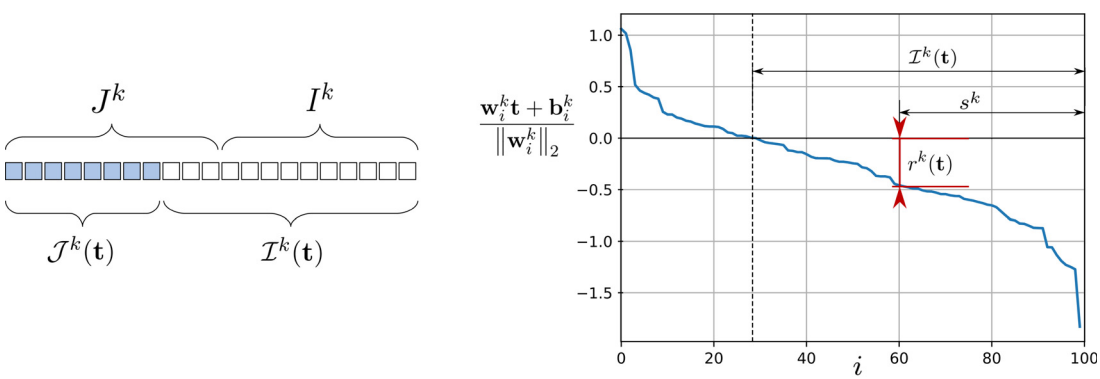
$l^{[K]}_{\text{inp}}(\mathbf{x}, \hat{\mathbf{s}}) \leq l^{[K]}_{\text{inp}}(\mathbf{x}, \mathbf{s})$, and thus $r_{\text{cert}}(\mathbf{x}, \mathbf{s}) \leq r_{\text{cert}}(\mathbf{x}, \hat{\mathbf{s}})$. Therefore, if $r_{\text{cert}}(\mathbf{x}, \hat{\mathbf{s}}) < \nu$, then for all $\mathbf{s} \in \mathcal{S}_{\text{good}}$ we necessarily have $r_{\text{cert}}(\mathbf{x}, \mathbf{s}) < \nu$ and the conclusion follows. ∎

In what follows, we will study a classes of functions that have monotone ordering.

4.2. Certified robustness for feedforward neural networks. For the remainder of this section, we will refer to feedforward neural networks exclusively as the representation map $\Phi^{[K]}$ given by the composition of K piecewise affine maps $\Phi^{(k)}(\mathbf{t}) := \sigma(\mathbf{W}^k\mathbf{t} + \mathbf{b}^k)$. The presence of the ReLU activation in each feedforward map naturally encourages some degree of sparsity at each layer. For each feedforward map $\Phi^{(k)}$, we denote the inactive set of the representation at any intermediate input $\mathbf{t} \in \mathbb{R}^{d^{k-1}}$ by $\mathcal{I}^k(\mathbf{t}) := \{j \in [d^k] \,|\, \mathbf{w}^k_j\mathbf{t} + \mathbf{b}^k_j \leq 0\}$, while its complement contains the support of layer k, i.e., $\mathcal{J}^k(\mathbf{t}) = \text{Supp}(\Phi^{(k)}(\mathbf{t})) = [d^k] \backslash \mathcal{I}^k(\mathbf{t})$. In what follows, we define index sets I^k as subsets of the full inactive set $\mathcal{I}^k(\mathbf{t})$, i.e., $I^k \subseteq \mathcal{I}^k(\mathbf{t})$, and the corresponding index set $J^k = (I^k)^C$ as supersets of the support set, i.e., $J^k \supseteq \mathcal{J}^k(\mathbf{t})$.[11] Figure 2(a) illustrates these sets for a given layer $\Phi^{(k)}(\mathbf{t})$. Later, in section 6, we will demonstrate typical levels of sparsity achieved by common feedforward models.

Since $\Phi^{(k)}$ is an affine map composed with a ReLU operator, the map $\Phi^{(k)}$ is Lipschitz with constant $\|\mathbf{W}^k\|_2$. In the following lemma, we move beyond this global characterization and define its sparse local radius as the maximum energy of a perturbation $\boldsymbol{\gamma} \in \mathbb{R}^{d^{k-1}}$ under which there exists at least one common inactive index set I^k of size s^k for both $\Phi^{(k)}(\mathbf{t})$ and $\Phi^{(k)}(\mathbf{t} + \boldsymbol{\gamma})$. Along with the local radius function, we also define a specific inactive set in the output that is guaranteed to withstand any input perturbation.

Lemma 4.7. *Let $\mathbf{s}^{(k)} = (s^{k-1}, s^k)$ denote the input-output sparsity levels for $\Phi^{(k)}$, so that $s^{k-1} \leq d^{k-1} - \|\mathbf{t}\|_0$ and $s^k \leq d^k - \|\Phi^{(k)}(\mathbf{t})\|_0$. The representation map $\Phi^{(k)}$ is SLL w.r.t. its*



(a) (b)

Figure 2. (a) *Illustration of the sets $\mathcal{J}^k(\mathbf{t})$, $\mathcal{I}^k(\mathbf{t})$, as well as I^k and J^k, for a given intermediate input $\sigma(\mathbf{W}^k\mathbf{t} + \mathbf{b}^k)$. Colored squares represent nonzero elements, ordered here without loss of generality.* (b) *Illustration of the radius $r^{(k)}_{\text{inp}}(\mathbf{t}, \mathbf{s}^{(k)})$ for a one layer neural network, given the (sorted) values of the normalized pre-activations.*

[11] For ease of notation, in the following discussion we denote $J^0 := [d^0]$.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

input with stable inactive index set I^k*, sparse local radius function* $r_{\text{inp}}^{(k)}$*, and sparse local Lipschitz scale function* $l_{\text{inp}}^{(k)}$ *defined as*

$$I^k(\mathbf{t},\mathbf{s}^{(k)}) := \underset{I\subseteq\mathcal{I}^k(\mathbf{t}),\,|I|=s^k}{\operatorname{argmax}} \min_{i\in I} \frac{\left|\mathbf{w}_i^k\mathbf{t}+\mathbf{b}_i^k\right|}{\left\|\mathbf{w}_i^k\right\|_2}, \quad r_{\text{inp}}^{(k)}(\mathbf{t},\mathbf{s}^{(k)}) := \min_{i\in I^k} \frac{\left|\mathbf{w}_i^k\mathbf{t}+\mathbf{b}_i^k\right|}{\left\|\mathbf{w}_i^k\right\|_2},$$
$$l_{\text{inp}}^{(k)}(\mathbf{t},\mathbf{s}^{(k)}) := \max_{I^{k-1}\subseteq\mathcal{I}^{k-1}(\mathbf{t}),\,|I^{k-1}|=s^{k-1}} \left\|\mathcal{P}_{J^k,J^{k-1}}(\mathbf{W}^k)\right\|_2,$$

where $J^k = (I^k(\mathbf{t},\mathbf{s}^{(k)}))^c, J^{k-1} = (I^{k-1})^c$ *are index sets of sizes* $(d^k - s^k)$ *and* $(d^{k-1} - s^{k-1})$*. Further, the feedforward map has monotone ordering.*

Proof. Fix any layer input $\mathbf{t}$ and sparsity level $\mathbf{s}^{(k)}$. The set I^k defined above is inactive for the representation $\Phi^{(k)}(\mathbf{t})$. Further, by the definition of the local radius function $r_{\text{inp}}^{(k)}$, for each inactive index $i\in I^k$, $\mathbf{w}_i^k\mathbf{t}+\mathbf{b}_i^k \le -\left\|\mathbf{w}_i^k\right\|_2 \cdot r_{\text{inp}}^{(k)}(\mathbf{t},\mathbf{s}^{(k)})$. Let $\tilde{\mathbf{t}}$ be a perturbed input that is within the sparse local radius $\|\tilde{\mathbf{t}}-\mathbf{t}\|_2 \le r_{\text{inp}}^{(k)}(\mathbf{t},\mathbf{s}^{(k)})$ and further shares a sparsity pattern $\mathcal{P}_{I^{k-1}}(\mathbf{t}) = \mathcal{P}_{I^{k-1}}(\tilde{\mathbf{t}}) = \mathbf{0}$ for some I^{k-1}. We shall show that the set I^k of size s^k is also inactive for the perturbed representation $\Phi^k(\tilde{\mathbf{t}})$ for any $i\in I^k$:

$$\mathbf{w}_i^k\tilde{\mathbf{t}}+\mathbf{b}_i^k = \left(\mathbf{w}_i^k\mathbf{t}+\mathbf{b}_i^k\right) + \mathbf{w}_i^k(\tilde{\mathbf{t}}-\mathbf{t}) \le -\left\|\mathbf{w}_i^k\right\|_2 \cdot r_{\text{inp}}^{(k)}(\mathbf{t},\mathbf{s}^{(k)}) + \left\|\mathbf{w}_i^k\right\|_2 \left\|\tilde{\mathbf{t}}-\mathbf{t}\right\|_2 \le 0.$$

Thus, we have shown the existence of a common inactive set for both the original and the perturbed representations. To bound the distance between the representations, note that

$$\left\|\Phi^{(k)}(\tilde{\mathbf{t}}) - \Phi^{(k)}(\mathbf{t})\right\|_2 = \left\|\sigma\left(\mathbf{W}^k\tilde{\mathbf{t}}+\mathbf{b}^k\right) - \sigma\left(\mathbf{W}^k\mathbf{t}+\mathbf{b}^k\right)\right\|_2 \le \left\|\mathcal{P}_{J^k,J^{k-1}}(\mathbf{W}^k)\left(\tilde{\mathbf{t}}-\mathbf{t}\right)\right\|_2.$$

The second inequality above stems from ignoring common inactive sets. To see that the r_{inp} and l_{inp} have monotone ordering, for input sparsity levels $s_1^{k-1} \le s_2^{k-1}$ and any fixed output sparsity level $s\in[d^k]$, we have, $I^k(\mathbf{t},(s_1^{k-1},s)) = I^k(\mathbf{t},(s_2^{k-1},s))$ by definition, and hence

$$r_{\text{inp}}\left(\mathbf{t},(s_1^{k-1},s)\right) = r_{\text{inp}}\left(\mathbf{t},(s_2^{k-1},s)\right), \quad l_{\text{inp}}\left(\mathbf{t},(s_1^{k-1},s)\right) \ge l_{\text{inp}}\left(\mathbf{t},(s_1^{k-1},s)\right).$$

Similarly, for a fixed $s\in[d^{k-1}]$, and $s_1^k \le s_2^k$, note that $I^k(\mathbf{t},(s,s_1^k)) \subseteq I^k(\mathbf{t},(s,s_2^k))$, and hence

$$r_{\text{inp}}\left(\mathbf{t},(s,s_1^k)\right) \ge r_{\text{inp}}\left(\mathbf{t},(s,s_2^k)\right), \quad l_{\text{inp}}\left(\mathbf{t},(s,s_1^k)\right) \ge l_{\text{inp}}\left(\mathbf{t},(s,s_1^k)\right). \qquad \blacksquare$$

It is worthwhile to stress the implications of the local radius function $r_{\text{inp}}^{(k)}$ for the intermediate feedforward layer $\Phi^{(k)}$. An analogous expression for this quantity can be obtained by considering the (normalized) vector $\mathbf{q}^k(\mathbf{t}) := [\frac{\mathbf{w}_i^k\mathbf{t}+\mathbf{b}_i^k}{\|\mathbf{w}_i^k\|_2}]_{i=1}^{d^k}$ of pre-activations,

$$I^k(\mathbf{t},\mathbf{s}^{(k)}) := \textsc{Top-k}(-\mathbf{q}^k(\mathbf{t}),s^k), \quad r_{\text{inp}}^{(k)}(\mathbf{t},\mathbf{s}^{(k)}) = \text{ReLU}\left(\textsc{sort}(-\mathbf{q}^k(\mathbf{t}),s^k)\right). \tag{4.8}$$

Here, for a vector $\mathbf{t}$ and index j, $\textsc{Top-k}(\mathbf{t},j)$ is the index set of the top j entries and $\textsc{sort}(\mathbf{t},j)$ is the j*th* largest entry in $\mathbf{t}$. The above expression also reveals that the computation of the sparse local radius can be easily incorporated into the forward pass of neural networks

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

(with an additional normalization and sort operation). The radius quantifies the minimal distance of a neuron in the "most inactive set" I^k (of size s^k) for becoming active. As illustrated in Figure 2(b), $r^{(k)}_{\text{inp}}(\mathbf{t}, \mathbf{s}^{(k)})$ is a *decreasing* function of s^k. While Lemma 4.7 studies sensitivity to ℓ_2 perturbations, one can easily extend this to any ℓ_p-norm perturbation. To do so, it suffices to compute the vector $\mathbf{q}^k(\mathbf{t})$ by normalizing with an ℓ_q-norm (with $1/q+1/p=1$). The corresponding Lipschitz scale is then $\left\|\mathcal{P}_{J^k,J^{k-1}}(\mathbf{W}^k)\right\|_{p\to p}$. We now present the robustness certificate that combines Theorem 4.3 and Lemmas 4.4 and 4.7.

Corollary 4.8. *Consider a trained depth-K feedforward neural network $h \in \mathcal{H}_{FNN,K}$. Let $\mathbf{s}$ be a fixed choice of sparsity levels at each layer so that $s^k \le d^k - \|\Phi^{(k)}(\mathbf{t})\|_0$, and let $\mathbf{s}^{(k)}$ be the corresponding layerwise input-output sparsity levels. The cumulative sparse local radius $r^{[K]}$ and local Lipschitz scale $l^{[K]}$ are defined as*

$$r^{[K]}_{\text{inp}}(\mathbf{x},\mathbf{s}) := \min_{1\le k\le K} \frac{\text{ReLU}\left(\text{SORT}(-\mathbf{q}^k(\Phi^{[k-1]}(\mathbf{x})), s^k)\right),}{\prod_{n=1}^{k}\left\|\mathcal{P}_{J^n,J^{n-1}}(\mathbf{W}^n)\right\|_2}, \qquad l^{[K]}_{\text{inp}}(\mathbf{x},\mathbf{s}) := \prod_{k=1}^{K}\left\|\mathcal{P}_{J^k,J^{k-1}}(\mathbf{W}^k)\right\|_2.$$

The predicted label remains unchanged, i.e., $\hat{y}(\mathbf{x}+\boldsymbol{\delta}) = \hat{y}(\mathbf{x})$, whenever $\|\boldsymbol{\delta}\|_2 \le r_{\text{cert}}(\mathbf{x},\mathbf{s})$:

$$r_{\text{cert}}(\mathbf{x},\mathbf{s}) := \min\left\{ r^{[K]}_{\text{inp}}(\mathbf{x},\mathbf{s}),\ \frac{\mathcal{M}(h(\mathbf{x}),\hat{y}(\mathbf{x}))}{2\,\|\mathbf{A}\|_2 \cdot l^{[K]}_{\text{inp}}(\mathbf{x},\mathbf{s})} \right\}.$$

The proof can be found in subsection SM1.4. Since feedforward networks have monotone ordering, for each $\nu > 0$, Algorithm 4.1 provides optimal sparsity vector $\hat{\mathbf{s}}$ for certification.

4.3. Discussion. For each input $\mathbf{x}$, the inactive index sets I^k of size s^k in Lemma 4.7 are chosen to maximize the local radius $r^{(k)}_{\text{inp}}(\mathbf{x})$. The certificate in Corollary 4.8 reflects a trade-off between the sparsity of the representations at each intermediate layer s^k (via the reduced operator norms) and $\mathcal{M}(h(\mathbf{x}),\hat{y}(\mathbf{x}))$, the classification margin in feature space. Before moving on, we summarize a few key remarks about our approach.

Reduced model perspective. At each input $\mathbf{x}$, for sparsity levels s^k and their chosen index sets J^k of size $d^k - s^k$, we can define a reduced neural network $\Phi^{[K]}_{\text{red}} : \mathbb{R}^{d^0} \to \mathbb{R}^{d^K - s^K}$,

$$\Phi^{[K]}_{\text{red}}(\mathbf{x}) := \sigma\left(\mathbf{W}^K_{\text{red}}\,\sigma\left(\mathbf{W}^{K-1}_{\text{red}}\cdots\sigma\left(\mathbf{W}^1_{\text{red}}\,\mathcal{P}_{J^0}(\mathbf{x}) + \mathbf{b}^1_{\text{red}}\right)\cdots + \mathbf{b}^{K-1}_{\text{red}}\right) + \mathbf{b}^K_{\text{red}}\right), \tag{4.9}$$

where $\mathbf{W}^k_{\text{red}} \in \mathbb{R}^{(d^k-s^k)\times(d^{k-1}-s^{k-1})}$ are defined as the submatrices of the parameters of $\Phi^{(k)}$ at specific active sets, i.e., $\mathbf{W}^k_{\text{red}} = \mathcal{P}_{J^k,J^{k-1}}(\mathbf{W}^k)$, and similarly for the biases. Corollary 4.8 essentially shows that, at each input $\mathbf{x}$, the feedforward neural network $\Phi^{[K]}$ is equivalent to a particular reduced feedforward neural network $\Phi^{[K]}_{\text{red}}$ in a local neighborhood around input $\mathbf{x}$. That is, for all perturbations $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_2 \le r^{[K]}_{\text{inp}}(\mathbf{x},\mathbf{s}^{[k]})$, the following holds:

$$\Phi^{[K]}(\mathbf{x}+\boldsymbol{\delta}) = \Phi^{[K]}_{\text{red}}(\mathbf{x}+\boldsymbol{\delta}). \tag{4.10}$$

The reduction of the active weights in the network locally can be seen as a form of input-specific pruning of the neural network. Importantly, this observation goes beyond the statement that a feedforward neural network is locally *linear* in a neighborhood of $\mathbf{x}$. Making this observation

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

would be too stringent, as the size of this neighborhood could be arbitrarily small. Instead, the equivalence in (4.10) holds for the *nonlinear* function $\Phi_{\text{red}}^{[K]}$. Within the specified neighborhood, the activation patterns might change but only in the complement of the sets I^k. The definition of these sets provides a controllable knob via the sparsity requirements.

Comparison to related work. These observations are related to the analysis based on max-affine operators in [7], providing a partitioning of the input space $\mathcal{X}$ based on successive feedforward layers. That work shows an effective way to compute the distance to the partition boundary, and this can be seen as a version of the local radius function we have defined *only* when the sparsity level is set to the exact number of inactive neurons $|\mathcal{I}^k(\mathbf{x})|$ for each input in each layer. When there exists a row $\mathbf{w}_j^k$ that is nearly active, i.e., $-\xi < \mathbf{w}_j^k \Phi^{[k-1]}(\mathbf{x}) + \mathbf{b}_j^k < 0$ for small $\xi \approx 0$, the distance from $\mathbf{x}$ to the input partition is near zero. In this case, the flexibility for different levels of sparsity in our analysis is crucial, allowing us to expand the active set and increase the allowable radius.

The work in [16] makes the empirical observation that *Parseval networks*—loosely speaking, those with operator norms close to 1—result in better robustness. Our results show that, in fact, this is not necessary as long as the operator norms of the *reduced* matrices are close to one. In particular, increasing depth can have a dramatic impact on the local Lipschitz scale l_{inp} if the reduced linear map is contractive while the original map is not, i.e., $\|\mathcal{P}_{J^k,J^{k-1}}(\mathbf{W}^k)\| < 1 < \|\mathbf{W}^k\|_2$. More generally, our results expose an inefficiency in approaches that directly compute the Lipschitz constant of the full feedforward network. Our measure of sensitivity that accounts for *locality* and *sparsity* are at least as good as global measures (and potentially much tighter). Note that one could use any algorithm for estimating the Lipschitz constant of a neural network [18, 23, 25, 47, 59, 62] applied to the reduced model, $\Phi_{\text{red}}^{[K]}$, in order to estimate $l_{\text{inp}}^{[K]}(\mathbf{x}, \mathbf{s}^{[K]})$ efficiently. Last, the authors of [37, 52] study the case of supervised sparse coding, performing a similar sensitivity analysis for the hypothesis class $\mathcal{H}_{\text{SSC}}$ focusing on a local radius threshold (or *encoder gap*) that preserves the support, or sparsity level, of the representation obtained under corruption. The robustness certificate developed in [52, Theorem 5.1] is equivalent to the application of Theorem 4.3 for the class $\mathcal{H}_{\text{SSC}}$ (see Lemma SM1.1 for full details). Thus, our work generalizes results in [52].

Dependence on input. The results presented above are input-specific and require the computation of the operator norms of the reduced submatrices. In many settings, it might be more relevant to have a similar notion for a set of inputs instead. For each layer, this can be done by searching for the worst case submatrix of $\mathbf{W}^k$ of size $(d^k - s^k, d^{k-1} - s^{k-1})$ via an extension of the well-studied notion of *babel function* [55].

Definition 4.9 (reduced babel function). *For any matrix $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$, we define the reduced babel function at row sparsity level $s_1 \in \{0, \dots, d_1 - 1\}$ and column sparsity level $s_2 \in \{0, \dots, d_2 - 1\}$ as*

$$\mu_{s_1,s_2}(\mathbf{W}) := \max_{\substack{J_1 \subset [d_1], \\ |J_1| = d_1 - s_1}} \max_{j \in J_1} \left[\sum_{\substack{i \in J_1, \\ i \neq j}} \max_{\substack{J_2 \subseteq [d_2] \\ |J_2| = d_2 - s_2}} \frac{|\mathcal{P}_{J_2}(\mathbf{w}_i) \mathcal{P}_{J_2}(\mathbf{w}_j)^T|}{\|\mathcal{P}_{J_2}(\mathbf{w}_i)\|_2 \, \|\mathcal{P}_{J_2}(\mathbf{w}_j)\|_2} \right],$$

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

the maximum cumulative mutual coherence between a reference row in J_1 of size $(d_1 - s_1)$, and any other row in J_1, each restricted to any subset of columns J_2 of size[12] $(d_2 - s_2)$.

The reduced babel function is computationally tractable,[13] albeit more expensive to compute than the babel function. The additional flexibility is showcased in the following result.

Lemma 4.10. *For any matrix $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$, the operator norm of any nontrivial*[14] *submatrix indexed by sets $J_1 \subseteq [d_1]$ of sizes $(d_1 - s_1)$ and $J_2 \subseteq [d_2]$ of size $(d_2 - s_2)$ can be bounded as $\|\mathcal{P}_{J_1,J_2}(\mathbf{W})\|_2 \le \sqrt{1 + \mu_{s_1,s_2}(\mathbf{W})} \cdot \|\mathbf{W}\|_{2,\infty}$.*

Proof. By the Gerschgorin disk theorem, for any eigenvalue λ of $\mathcal{P}_{J_1,J_2}(\mathbf{W})$, there exists index j such that λ_i lies in the Gerschgorin disks centered at $\langle \mathcal{P}_{J_2}(\mathbf{w}_j), \mathcal{P}_{J_2}(\mathbf{w}_j)\rangle$ with radius $\sum_{i \neq j} \langle \mathcal{P}_{J_2}(\mathbf{w}_j), \mathcal{P}_{J_2}(\mathbf{w}_i)\rangle$. The conclusion follows by bounding the radius using the reduced babel function $\mu_{s_1,s_2}(\mathbf{W})$. For a complete proof, see subsection SM1.6. ∎

A key feature of the above lemma is the bound on the operator norm of a submatrix that only depends on the size of the chosen index sets. For nontrivial sparsity levels, the proposed bound often improves on the naive bound $\|\mathcal{P}_{J_1,J_2}(\mathbf{W})\|_2 \le \|\mathbf{W}\|_2$. Similar notions have been proposed in [1]. The upper bounds from Lemma 4.10 can be computed offline and used in place of $\left\|\mathcal{P}_{J^k,J^{k-1}}(\mathbf{W}^k)\right\|_2$ for quick certification for a new sample with appropriate sparsity.

5. Robust generalization.

In this section, we move beyond deterministic robustness certificates and provide a generalization bound for the robust risk of SLL hypotheses that only has a mild dependence on the energy of the adversary. We do so by studying the sensitivity of a predictor to simultaneous changes in input *and* parameters.

Recall that a representation-linear hypothesis class $\mathcal{H}$ contains predictors of the form $h(\cdot) = \mathbf{A}\Phi_{\mathbf{W}}(\cdot)$, where the parameters $(\mathbf{A}, \mathbf{W}) \in \mathcal{A} \times \mathcal{W}$. Until further instantiated, we assume that the parameter sets $\mathcal{A}$ and $\mathcal{W}$ are bounded with respect to embedded norms, such that $\|\mathbf{A}\|_2 \le \mathsf{M}_{\mathcal{A}}$ and $\|\mathbf{W}\|_{\mathcal{W}} \le \mathsf{M}_{\mathcal{W}}$. We define the norm of a representation-linear predictor h to be $\|h\|_{\mathcal{H}} := \max\{\frac{\|\mathbf{A}\|_2}{\mathsf{M}_{\mathcal{A}}}, \frac{\|\mathbf{W}\|_{\mathcal{W}}}{\mathsf{M}_{\mathcal{W}}}\}$. For any pair predictors h and $\hat{h}$ with weights $(\mathbf{A}, \mathbf{W})$ and $(\hat{\mathbf{A}}, \hat{\mathbf{W}})$, we let Φ and $\hat{\Phi}$ be their corresponding representation maps. The distance between predictors is measured by the induced distance metric.[15] For a (globally) Lipschitz predictor h, one can obtain a uniform bound (see subsection SM1.8) on the generalization error that is $\mathcal{O}(\sqrt{\frac{\ln(\mathcal{N}(\frac{1}{m},\mathcal{H}))+\ln(\frac{1}{\alpha})}{m}} + \frac{\mathsf{L}_{\text{par}}}{m})$, with probability $1 - \alpha$, and $\mathcal{N}(\epsilon, \mathcal{H})$ is the proper covering number of $\mathcal{H}$ w.r.t. induced distance metric at resolution ϵ. In the analysis that follows, we refine the second term by exploiting data-dependent properties of a trained predictor.

5.1. Sparse local Lipschitz w.r.t. parameters.

We start by characterizing a class of functions that preserve sparsity in the representations.

Definition 5.1. *Let h be a representation-linear hypothesis such that $\Phi(\mathbf{x})$ is s-sparse at $\mathbf{x}$. The hypothesis h is s-sparse local Lipschitz w.r.t. parameters at $\mathbf{x}$ if there exist an inactive*

[12] When $s_1 = d_1 - 1, |J_1| = 1$, we simply define $\mu_{(s_1,s_2)}(\mathbf{W}) := 0$.

[13] Replacing the $\|\mathcal{P}_J(\mathbf{w}_i)\|_2$ in the denominators of the definition with $\|\mathbf{W}\|_{2,\infty}$ vastly reduces the complexity of evaluating the reduced babel function, and all subsequent lemmas still hold.

[14] That is, $0 \le s_1 \le d_1 - 1$ and $0 \le s_2 \le d_2 - 1$.

[15] The induced distance metric is $\|\hat{h} - h\|_{\mathcal{H}} := \max\{\frac{\|\hat{\mathbf{A}} - \mathbf{A}\|_2}{\mathsf{M}_{\mathcal{A}}}, \frac{\|\hat{\mathbf{W}} - \mathbf{W}\|_{\mathcal{W}}}{\mathsf{M}_{\mathcal{W}}}\}$.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

index set I of size s, a local radius $r \geq 0$, and a local Lipschitz scale $l \geq 0$ such that, for any perturbed predictor $\hat{h} \in \mathcal{B}_r^{\mathcal{H}}(h)$,

$$\mathcal{P}_I(\hat{\Phi}(\mathbf{x})) = \mathcal{P}_I(\Phi(\mathbf{x})) = \mathbf{0} \quad \wedge \quad \left\|\hat{h}(\mathbf{x}) - h(\mathbf{x})\right\|_2 \leq l \left\|\hat{h} - h\right\|_{\mathcal{H}}.$$

Furthermore, h is sparse local Lipschitz w.r.t. parameters if for every $\mathbf{x} \in \mathcal{X}$ and all appropriate sparsity levels s, h is s-sparse local Lipschitz (w.r.t. parameters) at $\mathbf{x}$, with corresponding local radius $r_{\text{par}}(\mathbf{x}, s)$ and local Lipschitz scale $l_{\text{par}}(\mathbf{x}, s)$.

As before, the local neighborhood is defined both in terms of a norm radius and a sparsity level, and both $r(\mathbf{x}, s)$ and $l(\mathbf{x}, s)$ decrease with increasing s. Importantly, for Lipschitz hypotheses h, there always exist s so that h is s-sparse local Lipschitz. As the reader can likely foresee, the utility of Definition 5.1 is that, when $r_{\text{par}}(\mathbf{x}, s) > 0$, the sparsity level s indicates the existence of a stable inactive set of indices I of size s, so that one can restrict the analysis to the (extended) active set $J = [d] \backslash I$ in such a way that

$$\|\hat{h} - h\|_{\mathcal{H}} \leq r_{\text{par}}(\mathbf{x}, s) \implies \hat{h}(\mathbf{x}) = \hat{\mathbf{A}}\hat{\Phi}(\mathbf{x}) = \mathcal{P}_{[C],J}(\hat{\mathbf{A}})\, \mathcal{P}_J(\hat{\Phi}(\mathbf{x})).$$

The range of sparsity levels for each input is in $\{0, \ldots, p - \|\Phi(\mathbf{x})\|_0\}$, and this recovers Lipschitz functions at the trivial choice of sparsity level $s = 0$ with $r_{\text{par}}(\mathbf{x}, 0) = \infty$ and $l_{\text{par}}(\mathbf{x}, 0) := \mathsf{L}_{\text{par}}$.

In order to study generalization, one must extend this property over a finite set of samples $\mathcal{V} \subset \mathcal{X}$. For a certain radius threshold $\epsilon > 0$, among all feasible sparsity levels, we choose the optimal s that minimizes the worst-case local Lipschitz scale across the set $\mathcal{V}$ while guaranteeing a sufficiently large local radius, that is,

$$s^*(\mathcal{V}, \epsilon) := \operatorname*{argmin}_s \max_{\mathbf{x} \in \mathcal{V}} l_{\text{par}}(\mathbf{x}, s) \quad \text{s.t.} \quad \epsilon \leq \min_{\mathbf{x} \in \mathcal{V}} r_{\text{par}}(\mathbf{x}, s). \tag{5.1}$$

Note that $s = 0$ is always feasible for the optimization problem defined. We can now define the *sparse regularity* of a predictor h w.r.t. reference set $\mathcal{V}$ and a fixed radius threshold ϵ as

$$\mathcal{L}(h, \mathcal{V}, \epsilon) := \max_{\mathbf{x} \in \mathcal{X}} l_{\text{par}}(\mathbf{x}, s^\star(\mathcal{V}, \epsilon)) \text{ s.t. } \epsilon \leq r_{\text{par}}\left(\mathbf{x}, s^\star(\mathcal{V}, \epsilon)\right).$$

This sparse regularity measures the worst-case local Lipschitz scale at any input in $\mathcal{X}$ with a sufficiently large local radius at the reference sparsity level $s^*(\mathcal{V}, \epsilon)$. In the unfavorable (*not sparse*) case $s^*(\mathcal{V}, \epsilon) = 0$, the corresponding sparse regularity $\mathcal{L}(h, \mathcal{V}, \epsilon) = \mathsf{L}_{\text{par}}$, the global Lipschitz constant. Thus, a generalization bound that relies on $\mathcal{L}(h, \mathcal{V}, \epsilon)$ is, at worst, dependent on the global Lipschitz constant L_{par} but potentially much tighter. We now present our generalization bound for SLL predictors, which makes use of *unlabeled* samples, $\mathtt{S}_U$, in addition to the training set $\mathtt{S}_T$, both with m samples. The former will be used to inform the sparse regularity of the predictor, while the latter is used to fit the parameters of the models.

Theorem 5.2. *With probability at least $(1 - \alpha)$ over the choice of i.i.d. training sample $\mathtt{S}_T$ and unlabeled data $\mathtt{S}_U$ each of size m, for any predictor $h \in \mathcal{H}$ with parameters $(\mathbf{A}, \mathbf{W})$, the generalization error, with a b-bounded and L_{loss}-Lipschitz loss, is bounded by*

$$R(h) \leq \hat{R}(h) + \mathcal{O}\left(b\sqrt{\frac{\ln\left(\mathcal{N}(\frac{1}{m}, \mathcal{H})\right) + \ln(\frac{2}{\alpha})}{2m}} + \frac{\mathsf{L}_{\text{loss}} \cdot \mathcal{L}(h, \mathtt{S}_T \cup \mathtt{S}_U, \frac{1}{2m})}{m}\right). \tag{5.2}$$

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

This result follows from standard arguments by constructing an ϵ cover of the hypothesis space and bounding the stability of the function's outputs on this cover, and it can be considered a generalization of the bound presented in [52]. The complete proof can be found in subsection SM1.9. Note that to compute the sparse regularity of h, r_{par} must be easy to compute at each input, and l_{par} must be regular enough to optimize over. The requirement of additional unlabeled data can be seen as a limitation; however, this dependence is mild, as it incurs in a linear increase in the number of training samples.

5.2. Robust sparse local Lipschitz. To extend Theorem 5.2 to the robust setting, one needs to characterize the parameter sensitivity of the predictor under corrupted inputs. Let $\nu > 0$ be the adversarial energy, and recall that $\mathsf{L}_{\text{par},\nu}$ is the global Lipschitz constant,

$$\forall\, \hat{h}, h, \mathbf{x}, \quad \max_{\boldsymbol{\delta} \in \mathcal{B}_\nu^{\mathcal{X}}(\mathbf{0})} \left\| \hat{h}(\mathbf{x}+\boldsymbol{\delta}) - h(\mathbf{x}+\boldsymbol{\delta}) \right\|_2 \leq \mathsf{L}_{\text{par},\nu} \left\| \hat{h} - h \right\|_{\mathcal{H}}.$$

Based on this, a uniform generalization bound, analogous to Theorem SM1.5 but for the robust risk $R_{\text{rob}}(h)$, can be readily established with a dependence of $\mathcal{O}(\mathsf{L}_{\text{par},\nu}/m)$. To move beyond the global analysis, we extend the sparse local Lipschitz property.

Definition 5.3. *Let h be a representation-linear hypothesis so that $\Phi(\mathbf{x})$ is s-sparse at $\mathbf{x}$. The hypothesis h is robust s-sparse local Lipschitz w.r.t. parameters at $\mathbf{x}$ if there exist an inactive index set I of size s, a local radius $r \geq 0$, and a local Lipschitz scale $l \geq 0$, such that, for any perturbed predictor $\hat{h} \in \mathcal{B}_r^{\mathcal{H}}(h)$ and any corruption $\boldsymbol{\delta} \in \mathcal{B}_\nu^{\mathcal{X}}(\mathbf{0})$, the index set I remains inactive after input and parameter perturbations, and the distance between the predictor outputs is bounded, that is,*

$$\mathcal{P}_I(\hat{\Phi}(\mathbf{x}+\boldsymbol{\delta})) = \mathcal{P}_I(\Phi(\mathbf{x}+\boldsymbol{\delta})) = \mathbf{0} \quad \wedge \quad \left\| \hat{h}(\mathbf{x}+\boldsymbol{\delta}) - h(\mathbf{x}+\boldsymbol{\delta}) \right\|_2 \leq l \left\| \hat{h} - h \right\|_{\mathcal{H}}.$$

Additionally, the hypothesis h is robust sparse local Lipschitz w.r.t. parameters if h is sparse local Lipschitz (w.r.t. parameters) for every $\mathbf{x} \in \mathcal{X}$ and any appropriate sparsity level s, with corresponding local radius $r_{\text{par},\nu}(\mathbf{x}, s)$ and local Lipschitz scale $l_{\text{par},\nu}(\mathbf{x}, s)$.

For a robust sparse local Lipschitz h, at any input $\mathbf{x}$ and sparsity level s where the predictor has a nontrivial robust local radius (i.e., $r_{\text{par},\nu}(\mathbf{x}, s) \geq 0$), there exists an inactive index set I of size s for the representation $\Phi(\mathbf{x})$ that withstands simultaneous perturbations to inputs and parameters. Note that the chief difference between this and Definition 5.1 is that here the sensitivity is evaluated at the point $(\mathbf{x}+\boldsymbol{\delta})$ while being a property of h at $\mathbf{x}$. Indeed, a sparse local Lipschitz predictor is also robust sparse local Lipschitz,

$$r_{\text{par},\nu}(\mathbf{x}, s) = \min_{\boldsymbol{\delta} \in \mathcal{B}_\nu^{\mathcal{X}}(\mathbf{0})} r_{\text{par}}(\mathbf{x}+\boldsymbol{\delta}, s), \quad l_{\text{par},\nu}(\mathbf{x}, s) = \max_{\boldsymbol{\delta} \in \mathcal{B}_\nu^{\mathcal{X}}(\mathbf{0})} l_{\text{par}}(\mathbf{x}+\boldsymbol{\delta}, s).$$

At the trivial sparsity level $s = 0$, we simply let $r_{\text{par},\nu}(\mathbf{x}, 0) = \infty$ and the $l_{\text{par},\nu}(\mathbf{x}, 0) = \mathsf{L}_{\text{par},\nu}$ for any Lipschitz h. Leveraging the controllable trade-off between sparsity levels and the local sensitivity, we define the robust optimal level $s^*_{\text{rob}}(\mathcal{V}, \epsilon)$ and robust sparse regularity,

$$s^*_{\text{rob}}(\mathcal{V}, \epsilon) := \operatorname*{argmin}_s \max_{\mathbf{x} \in \mathcal{V}} l_{\text{par},\nu}(\mathbf{x}, \mathbf{s}) \text{ s.t. } \epsilon \leq \min_{\mathbf{x} \in \mathcal{V}} r_{\text{par},\nu}(\mathbf{x}, \mathbf{s}), \tag{5.3}$$

$$\mathcal{L}_{\text{rob}}(h, \mathcal{V}, \epsilon) := \max_{\mathbf{x} \in \mathcal{X}} l_{\text{par},\nu}(\mathbf{x}, s^\star_{\text{rob}}(\mathcal{V}, \epsilon)) \text{ s.t. } \epsilon \leq r_{\text{par},\nu}\left(\mathbf{x}, s^\star_{\text{rob}}(\mathcal{V}, \epsilon)\right).$$

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Using these ideas, the result in Theorem 5.2 can be extended to the robust setting (see Theorem SM1.7). We omit this and move on to our analysis for feedforward neural networks.

5.3. Robust generalization for feedforward neural networks. For simplicity, we only consider networks with zero bias (as in [9, 40, 41, 43]).[16] We consider depth-$(K+1)$ feedforward neural networks where the representation map Φ has layer weights $\{\mathbf{W}^k\}_{k=1}^K$. For notational convenience, we denote the classification weight in the final linear layer as $\mathbf{W}^{K+1}$ (in lieu of $\mathbf{A}$). For the reminder of this section, we consider a fixed set of constants $\{\mathsf{M}_{\mathcal{W}}^k, \{\mathsf{M}_s^k\}\}_{k=1}^{K+1}$ that defines a hypothesis space $\mathcal{H}^{K+1}$ with parameters in $\prod_{k=1}^{K+1}\mathcal{W}^k$, where

$$\mathcal{W}^k := \left\{\mathbf{W} \in \mathbb{R}^{d^k \times d^{k-1}} \;\middle|\; \|\mathbf{W}\|_{2,\infty} \le \mathsf{M}_{\mathcal{W}}^k \;\; \forall\, (s^k, s^{k-1}),\;\; \mu_{s^k,s^{k-1}}(\mathbf{W}) \le \mathsf{M}_{s^k}^k\right\},$$

while $\mathcal{W}^{K+1} := \{\mathbf{W} \in \mathbb{R}^{d^K \times C} \;\big|\; \|\mathbf{W}\|_{2,\infty} \le \mathsf{M}_{\mathcal{W}}^{K+1} \quad \forall\, s^K,\;\; \mu_{s^K,0}(\mathbf{W}) \le \mathsf{M}_{s^K}^{K+1}\}$. The final classification weight space[17] accounts for the sparsity in the representation output, as opposed to the output of the predictor. In this manner, a predictor $h \in \mathcal{H}^{K+1}$ is defined as $h(\mathbf{x}) = (\mathbf{W}^{K+1})^T\Phi^{[K]}(\mathbf{x})$, where the representation map $\Phi^{[K]}$ is the composition of K feedforward maps, so that $\Phi^{[K]}(\mathbf{x}) = \sigma(\mathbf{W}^K\sigma(\mathbf{W}^{K-1}\cdots\sigma(\mathbf{W}^1\mathbf{x})))$. While the weight spaces are constrained in the group norm, we define the following scaled norm $\|\cdot\|_{\mathcal{W}^k}$ fit to the purpose of measuring parameter perturbations $\|\mathbf{W}\|_{\mathcal{W}^k} := \frac{\sqrt{d^k}}{\mathsf{M}_{\mathcal{W}}^k}\cdot\|\mathbf{W}\|_{2,\infty}$ such that $\left\|\mathbf{W}^k\right\|_{\mathcal{W}^k} \le \sqrt{d^k}$ for any $\mathbf{W}^k \in \mathcal{W}^k$. Based on the scaled norms, we define the norm of any feedforward network, $\|h\|_{\mathcal{H}^{K+1}} := \max_{1\le k\le K+1}\|\mathbf{W}^k\|_{\mathcal{W}^k}$. Additionally, the predictors in $\mathcal{H}^{K+1}$ are constrained by the reduced babel function[18] at each layer for all appropriate sparsity levels. As before, at each layer $1 \le k \le K$, $\mathcal{I}^k(\mathbf{x}) := \{j \in [d^k] : \mathbf{w}_j^k\Phi^{[k-1]}(\mathbf{x}) \le 0\}$ and $\bar{s}^k(\mathbf{x}) := |\mathcal{I}^k(\mathbf{x})|$ denote the index set of all inactive rows and their sizes, respectively. We further let $\bar{\mathbf{s}}(\mathbf{x}) := \{\bar{s}^1(\mathbf{x}), \ldots, \bar{s}^K(\mathbf{x})\}$. Although Definition 5.3 only requires a scalar sparsity level corresponding to the representation output, $\Phi(\mathbf{x})$, for the case of multilayered neural networks, we will refine this definition by a vector of layerwise sparsity levels $\mathbf{s} = \{s^0, s^1, \ldots, s^K\}$ that achieve the same goal of sparsity in each layer representation at the level s^k. Note that for a representation at a given point, $\Phi(\mathbf{x})$, this latter vector denotes potential sparsity levels, whereas the previous $\bar{\mathbf{s}}(\mathbf{x})$ denotes the maximal possible sparsity; i.e., $s^k \le \bar{s}^k(\mathbf{x})$ for all k.

The intermediate sparsity levels can improve the robust properties further. To quantify this phenomenon, for any sparsity vector $\mathbf{s}$, we define $\zeta^0(\mathbf{s}) := 1$ and for $1 \le k \le K+1$, $\zeta^k(\mathbf{s}) := \prod_{n=1}^k \mathsf{M}_{\mathcal{W}}^n\sqrt{1+\mathsf{M}_{s^n}^n}$. Indeed, as per Lemma 4.10, $\zeta^k(\mathbf{s})$ provides an upper bound on the product of operator norms of reduced linear maps, i.e.,

(5.4)
$$\zeta^k(\mathbf{s}) \ge \prod_{n=1}^k \sup_{\mathbf{W}^n\in\mathcal{W}^n} \|\mathbf{W}^n\|_{2,\infty}\sqrt{1+\mu_{s^n,s^{n-1}}(\mathbf{W}^n)} \ge \prod_{n=1}^k \sup_{\substack{\mathbf{W}^n\in\mathcal{W}^n,\\ |J^n|=d^n-s^n,\\ |J^{n-1}|=d^{n-1}-s^{n-1}}} \left\|\mathcal{P}_{J^n,J^{n-1}}(\mathbf{W}^n)\right\|_2.$$

[16] Results for networks with nonzero bias can also be derived from our analysis.

[17] For convenience, $\mathcal{W}^{K+1}$ has been defined as a subset of $\mathbb{R}^{d^K\times C}$ rather than $\mathbb{R}^{C\times d^K}$.

[18] Naturally, we only consider constraints that match the properties of the reduced babel function: since by definition, $\mu_{(d^k-1,s)}(\mathbf{W}) = 0$, we require $\mathsf{M}_{d^k-1}^k = 0$ for all layers. Furthermore, for any $a, b : a \ge b$, we require $\mathsf{M}_a^k \le \mathsf{M}_b^k$ to mirror the fact that $\mu_{a,s}(\mathbf{W}^k) \le \mu_{b,s}(\mathbf{W}^k)$.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

For any $\tilde{\mathbf{s}} \succeq \mathbf{s} \succeq \mathbf{0}$, $\zeta^k(\tilde{\mathbf{s}}) \leq \zeta^k(\mathbf{s}) \leq \zeta^k(\mathbf{0})$, where $\zeta^k(\mathbf{0})$ is an upper bound on the product of operator norms of the full linear maps $\mathbf{W}^k$. For the induced distance metric[19] corresponding to $\|\cdot\|_{\mathcal{H}}$, we note the following *robust* (global) Lipschitz of a neural network.

Lemma 5.4. *For a fully connected neural network with K layers, $\Phi^{[K]}(\mathbf{x})$, its robust global Lipschitz constant can be upper bounded by $\mathsf{L}_{\mathrm{par},\nu} \leq (K+1)\zeta^{K+1}(\mathbf{0}) \cdot (1+\nu)$.*

Proof sketch. The proof is a simple application of the definitions above and operator norm inequalities. Given predictors $h, \hat{h} \in \mathcal{H}^{K+1}$ with weights $\{\mathbf{W}^k\}$ and $\{\hat{\mathbf{W}}^k\}$, we note that for $1 \leq k \leq K$, for any layer weight matrix $\mathbf{W}^k \in \mathcal{W}^k$, $\left\|\mathbf{W}^k\right\|_2 \leq \sqrt{1+\mathsf{M}_0^k} \cdot \mathsf{M}_{\mathcal{W}}^k$, and further, $\|\hat{\mathbf{W}}^k - \mathbf{W}^k\|_2 \leq \mathsf{M}_{\mathcal{W}}^k \|\hat{h} - h\|_{\mathcal{H}^{K+1}}$. Similar inequalities hold for $K+1$. We then show that, at any layer $k \leq K$, the distance between the perturbed representations is bounded: $\|\hat{\Phi}^{[k]}(\mathbf{x}+\boldsymbol{\delta}) - \Phi^{[k]}(\mathbf{x}+\boldsymbol{\delta})\| \leq k\zeta^k(\mathbf{0}) \cdot (1+\nu) \cdot \|\hat{h} - h\|_{\mathcal{H}^{K+1}}$. The final proof follows by employing the upper bound to the operator norms, given by ζ^k, and this latter bound of the representations at every layer. The full proof can be found in subsection SM1.11. ∎

Note that $\mathsf{L}_{\mathrm{par},\nu}$ is exponential in the network's depth and captures the worst-case interaction between layer matrices, inputs, and adversarial perturbations. Frequently, generalization bounds measure the sensitivity of a hypothesis class using $\mathsf{L}_{\mathrm{par},\nu}$ and hence $\zeta^K(\mathbf{0})$ [9, 41]. To instead measure sensitivity using $\zeta^k(\mathbf{s})$, we need to characterize the inactive set at each layer, and hence we identify a *critical angle* between the rows of $\mathbf{w}^k$ and the layer input.

Definition 5.5 (critical angular distance). *Let $\mathsf{M}_{\mathcal{W}}$ and $\mathsf{M}_{\mathcal{T}}$ be domain hyperparameters. Consider a matrix $\mathbf{W} \in \mathcal{W} \subset \mathbb{R}^{p \times q}$ such that $\|\mathbf{W}\|_{2,\infty} \leq \mathsf{M}_{\mathcal{W}}$ and a vector $\mathbf{t} \in \mathcal{T} \subset \mathbb{R}^q$ such that $\|\mathbf{t}\|_2 \leq \mathsf{M}_{\mathcal{T}}$. The angular distance[20] between the matrix $\mathbf{W}$ and vector $\mathbf{t}$ is defined to be the vector function $\beta : \mathcal{W} \times \mathcal{T} \to [0,1]^p$,*

$$[\beta(\mathbf{W}, \mathbf{t})]_i := \frac{1}{\pi} \cdot \arccos\left(\frac{\langle \mathbf{w}_i, \mathbf{t} \rangle}{\mathsf{M}_{\mathcal{W}} \mathsf{M}_{\mathcal{T}}}\right) \quad \forall\, i \in [p].$$

The critical angular distance $\theta : \mathcal{W} \times \mathcal{T} \times [p] \to [0,1]$ at sparsity level s is the sth-largest entry,

$$\theta(\mathbf{W}, \mathbf{t}, s) := \mathrm{SORT}\left(\beta(\mathbf{W}, \mathbf{t}), s\right).$$

Each component of $\beta(\mathbf{W}, \mathbf{t})$ quantifies the angular distance between a row $\mathbf{w}_i$ and $\mathbf{t}$.[21] In turn, the critical angular distance $\theta(\mathbf{W}, \mathbf{t}, s)$ represents the sth-largest angular distance

[19] For any two networks h and $\hat{h}$ with weights $\{\mathbf{W}^k\}_{k=1}^K$ and $\{\mathbf{W}^k\}_{k=1}^{K+1}$, respectively,

$$\left\|\hat{h} - h\right\|_{\mathcal{H}^{K+1}} := \max_{1 \leq k \leq K+1} \left\|\hat{\mathbf{W}}^k - \mathbf{W}^k\right\|_{\mathcal{W}^k}.$$

[20] The term "distance" here is an abuse of notation. More precisely, the vector $\beta(\mathbf{W}, \mathbf{t})$ contains a distance value in each component.

[21] Note that, naturally, $\langle \mathbf{w}_i, \mathbf{t} \rangle = \mathsf{M}_{\mathcal{W}} \mathsf{M}_{\mathcal{T}} \cos\left(\pi [\beta(\mathbf{W}, \mathbf{t})]_i\right)$. Furthermore, our definition represents a scaled version of the true angular distance since

$$\left|\frac{\langle \mathbf{w}_i, \mathbf{t} \rangle}{\mathsf{M}_{\mathcal{W}} \mathsf{M}_{\mathcal{T}}}\right| \leq \left|\frac{\langle \mathbf{w}_i, \mathbf{t} \rangle}{\|\mathbf{w}_i\|_2 \|\mathbf{t}\|_2}\right|.$$

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

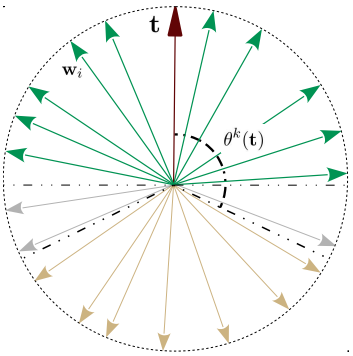


Figure 3. *Illustration of the critical angular distance for layer matrix* $\mathbf{W}^k$ *and input* $\mathbf{t}$*. Green weights are active, while grey and orange ones are inactive. The critical angle,* $\pi \cdot \theta^k(\mathbf{t})$*, is denoted by the black dashed line.*

formed by any row of matrix $\mathbf{W}$ with the input $\mathbf{t}$. A larger critical angular distance indicates that some set of s inactive indices in $\sigma(\mathbf{W}\mathbf{t})$ is resilient to bounded perturbations in the input $\mathbf{t}$ or weight vectors $\mathbf{W}$, as illustrated in Figure 3.

For the case of multilayered neural networks, a layerwise angular distance and critical angular distance can be evaluated at each layer with domain hyperparameters $\mathsf{M}^k_{\mathcal{W}}$ and $\zeta^{k-1}(\mathbf{0})$, which we denote by $\beta^k(\mathbf{x}) := \beta(\mathbf{W}^k, \Phi^{[k-1]}(\mathbf{x}))$ and $\theta^k(\mathbf{x}, \mathbf{s}) := \textsc{sort}(\beta^k(\mathbf{x}), s^k)$. Associated with the critical angular distance $\theta^k(\mathbf{x}, \mathbf{s})$, at layer k, there is an index set $I^k = \textsc{Top–k}(\beta^k(\mathbf{x}), s^k)$ such that, for each $i \in I^k$,

$$\frac{\mathbf{w}^k_i \Phi^{[k-1]}(\mathbf{x})}{\mathsf{M}^k_{\mathcal{W}} \zeta^{k-1}(\mathbf{0})} = \cos\Big(\pi \cdot \beta^k(\mathbf{x})_i\Big) \le \cos\Big(\pi \cdot \theta^k(\mathbf{x}, \mathbf{s})\Big).$$

Therefore, if $\cos\big(\pi\theta^k(\mathbf{x}, \mathbf{s})\big) < 0$ (i.e., if $\theta^k(\mathbf{x}, \mathbf{s}) \ge \frac{1}{2}$), then the set I^k is inactive for $\Phi^{[k]}(\mathbf{x})$.[22] Thus, critical angular distance can capture the existence of a *stable* inactive index sets.

Lemma 5.6. *A feedforward neural network* $h \in \mathcal{H}^{K+1}$ *is* $\mathbf{s}$*-robust sparse local Lipschitz w.r.t. parameters with scale* $l_{\mathrm{par},\nu}(\mathbf{x}, \mathbf{s}) := (K+1)\zeta^{(K+1)}(\mathbf{s}) \cdot (1+\nu)$ *and radius*

$$r_{\mathrm{par},\nu}(\mathbf{x}, \mathbf{s}) := \min_{1 \le k \le K} \frac{\iota(\mathbf{s}) + \max\big\{0, -\cos(\pi\theta^k(\mathbf{x}, \mathbf{s}) - \nu\big\}}{k(1+\nu)}.$$

Here $\iota(\cdot)$ *is the extended indicator*[23] *function of the positive orthant* $\mathbb{R}^K_+$*.*

Proof sketch. In a nutshell, this result can be shown by noting that for the defined radius $r_{\mathrm{par},\nu}(\mathbf{x}, \mathbf{s})$, the critical angular distance $\Theta^k(\mathbf{x}, \mathbf{s})$ at each layer is sufficiently large so that rows corresponding to strongly inactive set I^k remain inactive upon perturbations to the model

[22] In the presence of appropriately scaled and nonzero bias $\mathbf{b}^k$, the lower bound here becomes

$$\theta^k(\mathbf{x}) \ge \frac{1}{\pi} \cdot \arccos\left(\frac{-\mathbf{b}^k_i}{\mathsf{M}^k_{\mathcal{W}} \zeta^{k-1}(\mathbf{0})}\right).$$

[23] For all $\mathbf{s} \succ \mathbf{0}$, $\iota(\mathbf{s}) = 0$ and at the trivial choice sparsity levels, $\iota(\mathbf{0}) = \infty$.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

weights (by no more than $r_{\mathrm{par},\nu}(\mathbf{x},\mathbf{s})$). One can then notice that since the strongly inactive sets at each layer are maintained, one can follow logic similar to Lemma 5.4 to obtain a Lipschitz scale dependent on the operator norms of the reduced weights at each layer. The scaling factor $k\cdot(1+\nu)$ stems from requiring larger critical angle distance in the last few layers to withstand the multiplicative effect of perturbation in composed predictors such as $\Phi^{[K]}$. The proof is simple, albeit somewhat long, and so we defer the full version to subsection SM1.12. ∎

To interpret Lemma 5.6, consider a fixed parameter radius ϵ. If the angular distances of the farthest s^k vectors at each layer k is sufficiently large—i.e., if the s^k vectors have a sufficiently negative correlation with the input—then they can withstand input perturbation of magnitude ν, and parameter perturbation of magnitude ϵ, to still remain inactive. If at input $\mathbf{x}$ the robust local radius $r_{\mathrm{par}}(\mathbf{x},s)<\epsilon$ for all nontrivial sparsity levels $s>0$, then one cannot guarantee the preservation of a (nontrivial) inactive set at any layer. In this case, the distance between predictors outputs is bounded by $\mathsf{L}_{\mathrm{par},\nu}=l_{\mathrm{par},\nu}(\mathbf{x},\mathbf{0})$.

Reduced model perspective. If the robust sparse local radius $r_{\mathrm{par},\nu}(\mathbf{x},\mathbf{s})>0$, Lemma 5.6 establishes the existence of stable inactivity of index sets $I^k=\textsc{Top–k}(\beta^k(\mathbf{x}),s^k)$ at each layer. In such an event, the effect of $\hat{h}\in\mathcal{H}^{K+1}$ with perturbed parameters (within the radius) on any perturbed input $\mathbf{x}+\boldsymbol{\delta}$ can be reduced to a predictor $\hat{h}_{\mathrm{red}}(\mathbf{x}+\boldsymbol{\delta})$ using only the reduced weight matrices $\hat{\mathbf{W}}^k_{\mathrm{red}}=\mathcal{P}_{J^k,J^{k-1}}(\hat{\mathbf{W}}^k)\in\mathbb{R}^{(d^k-s^k)\times(d^{k-1}-s^{k-1})}$:

$$\hat{h}(\mathbf{x}+\boldsymbol{\delta})=(\hat{\mathbf{W}}^{K+1}_{\mathrm{red}})^T\sigma\left(\hat{\mathbf{W}}^{K}_{\mathrm{red}}\cdots\sigma\left(\hat{\mathbf{W}}^{1}_{\mathrm{red}}\,(\mathbf{x}+\boldsymbol{\delta})\right)\cdots\right)=:\hat{h}_{\mathrm{red}}(\mathbf{x}+\boldsymbol{\delta}).$$

As in subsection 4.3, the reduced predictor is still nonlinear and the index sets $J^k=(I^k)^c$ are determined by the input $\mathbf{x}$ and unperturbed predictor h. The above reduction is of course also true for the original predictor. At a fixed sparsity vector $\mathbf{s}$, although the index sets vary across inputs, we can analyze and bound the Lipschitz constant of such a worst-case reduction. This explains the independence of the robust local Lipschitz scale on a specific input and the utilization of $\zeta^{[K+1]}(\mathbf{s})$, an upper bound on the worst-case product of operator norms of reduced weights.

As per (5.1), let the robust optimal sparsity level be $\mathbf{s}:=s^*_{\mathrm{rob}}(\mathcal{V},\frac{1}{|\mathcal{V}|})$,[24] and the robust sparse regularity for a feedforward network predictor h is

$$\mathcal{L}_{\mathrm{rob}}\left(h,\mathcal{V},\frac{1}{|\mathcal{V}|}\right)=l_{\mathrm{par},\nu}(\mathbf{x},\mathbf{s})=(K+1)\zeta^{K+1}(\mathbf{s})\cdot(1+\nu). \tag{5.5}$$

This quantifies the worst-case local Lipschitz scale of the predictor h at any input $\mathbf{x}$ where it has a sufficiently large local radius, is a data-dependent norm-based regularity measure that can be much smaller than $l_{\mathrm{par},\nu}(\mathbf{x},\mathbf{0})$, depending on the set $\mathcal{V}$, and scales linearly with the adversarial energy ν. We are finally ready to present the main result bounding the robust generalization error of feedforward networks.

Theorem 5.7. *With probability at least $(1-\alpha)$ over the choice of i.i.d. training sample S_T and unlabeled data S_U, each of size m, for any feedforward network predictor $h\in\mathcal{H}^{K+1}$,*

[24] Note here that the sparsity levels are vectors, and $s^\star_{\mathrm{rob}}(\mathcal{V},\epsilon)$ searches over layerwise sparsity levels.

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

$$R_{\mathrm{rob}}(h) \le \hat{R}_{\mathrm{rob}}(h) + \mathcal{O}\left(b\sqrt{\frac{\mathcal{E}_{\mathcal{N}}(\mathcal{H},m)+\ln(\frac{2}{\alpha})}{2m}} + \frac{\mathsf{L}_{\mathrm{loss}}\cdot\mathcal{L}_{\mathrm{rob}}(h,\mathtt{S}_T\cup\mathtt{S}_U,\frac{1}{2m})}{m(K+1)}\right),$$

where $\mathcal{E}_{\mathcal{N}}(\mathcal{H},m) = \ln(\mathcal{N}(\frac{1}{m(K+1)},\mathcal{H}^{K+1}))$ *is the log of the covering number.*

The robust sparse regularity $\mathcal{L}_{\mathrm{rob}}(h,\mathtt{S}_T\cup\mathtt{S}_U,\frac{1}{2m})$ for a feedforward neural network is solely determined by the sparsity levels $\mathbf{s}$ via $\zeta^k(\mathbf{s})$, the worst-case operator norm of **any** reduced layer weight in $\mathcal{W}^k$. One can tune this result to be dependent on a specific trained predictor (see Theorem SM1.11). We state a specific instance of such an improved result.

Corollary 5.8. *With probability at least* $(1-\alpha)$ *over the choice of training data* $\mathtt{S}_T$ *and unlabeled data* $\mathtt{S}_U$ *each of size* m*, for any soft-margin threshold* $\gamma > 0$*, for any feedforward neural network* $h \in \mathcal{H}^{K+1}$*, there exist layerwise sparsity levels* $\mathbf{s} = [s^1,\dots,s^K]$ *dependent on data* $\mathtt{S}_T\cup\mathtt{S}_U$ *so that the probability of robust misclassification is bounded:*

$$R^{[0/1]}_{rob}(h) \le \hat{R}^{\gamma}_{rob}(h) + \tilde{\mathcal{O}}\left(\sqrt{\frac{\mathcal{E}_{\mathcal{N}}(\mathcal{H},m)+\ln(\frac{2}{\alpha})}{m}} + \frac{(1+\nu)}{\gamma m}\prod_{k=1}^{K+1}\left\|\mathbf{W}^k\right\|_{2,\infty}\sqrt{1+\mu_{s^k,s^{k-1}}(\mathbf{W}^k)}\right),$$

where $R^{[0/1]}_{rob}(h)$ *denotes the robust risk for the zero-one loss* $\ell^{[0/1]}$ *(i.e., probability of robust misclassification at any input)*[25]*, and* $\hat{R}^{\gamma}_{rob}(h)$ *is the robust empirical risk for the margin loss* ℓ^{γ}*.*

In the above theorem, $\tilde{\mathcal{O}}(g)$ denotes complexity that suppresses log factors[26] [9]. We have thus characterized the robust generalization ability of predictors that are SLL w.r.t. parameters.

Comparison to related work. Unlike prior work in robust generalization [61, 5], our bound has a milder dependence on the adversarial energy, $\mathcal{O}(\frac{\nu}{m})$, rather than $\mathcal{O}(\frac{\nu}{\sqrt{m}})$. The full proof found in subsection SM1.14 employs logic akin to structural risk minimization inspired from [9]. The second term in the bound is a predictor-dependent instantiation of the robust sparse regularity normalized by margin threshold, bearing resemblance to other spectrally normalized margin bounds [5, 9, 24, 31, 42, 43, 61]. The reduced babel function captures the coherence between any row and a subset of other rows (with an additional column restriction). Coupled with the group norms, as per Lemma 4.10, the second term in the above bound scales as the product of operator norms of reduced-linear maps, rather than the full weight matrices $\mathbf{W}^k$. The sparsity level $\mathbf{s}$ above is determined by the training sample $\mathtt{S}_T$ and the unlabeled data $\mathtt{S}_U$, as $\mathbf{s} = s^*_{\mathrm{rob}}(\mathtt{S}_T\cup\mathtt{S}_U,\frac{1}{2m})$. Only for a worst-case choice of data distribution and trained network are the sparsity levels trivial ($\mathbf{s}=\mathbf{0}$), in which case one recovers a result that only depends on the global Lipschitz constants.

Our analysis studies robust generalization using both the favorable properties of a training data and the local sensitivity of a trained predictor and is closest in spirit to results on standard (benign) generalization [8, 40, 49, 58]. We note that the bounds in [8, 40, 58] do not have

[25] Formally, $R^{[0/1]}_{\mathrm{rob}}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\mathcal{Z}}}[\max_{\boldsymbol{\delta}\in\mathcal{B}^{\mathcal{X}}_{\nu}(\mathbf{x})}\mathbf{1}\{y\neq\operatorname{argmax}_j h(\mathbf{x}+\boldsymbol{\delta})\}]$.

[26] For functions f and g with the same arguments, $f=\tilde{\mathcal{O}}(g)$ if there exists a constant C such that for any sequence of arguments $\{t^j\}_{j\ge\infty}$ such that $t^j\to\infty$ we have that $\limsup_{j\to\infty}\frac{f(t^j)}{g(t^j)\mathrm{polylog}(g(t^j))}\le C$.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

explicit dependence on the number of parameters. In contrast, a bound on the covering number term $\mathcal{N}(\frac{1}{m(K+1)}, \mathcal{H}^{K+1})$ in Corollary 5.8 can be obtained by parameter counting (see Lemma SM1.17), i.e., $\mathcal{E}_{\mathcal{N}}(\mathcal{H}^{K+1}, m) \propto \mathcal{O}(\log(m) + \sum_{k=1}^{K+1} d^k d^{k-1})$. This renders our result vacuous in the overparameterized regime when $\sum_{k=1}^{K+1} d^k d^{k-1} \gg m$. We believe this is a limitation of the current proof technique, rather than intrinsic to our sensitivity analysis, and conjecture that combining SLL as per Lemma 5.6 with the other learning-theoretic approaches could remove this dependence on dimensions. We leave this extension to future work.

Similar to our analysis, the authors of [40] capture a reduced dimensionality of neural networks and bound the generalization error of the original deterministic network by derandomizing the standard PAC-Bayesian bound. Their analysis weakens the exponential dependence on depth (i.e., the global Lipschitz constant) and does not require an additional unlabeled dataset. However, their bound depends inversely on the minimum absolute pre-activation level in each layer—which can be arbitrarily small in practice. The sensitivity of the predictor is quantified assuming that the original active sets $\mathcal{I}^k(\mathbf{x})$ at each layer remain unchanged upon a Gaussian perturbation (equivalent to requiring $r_{\text{par}}(\mathbf{x}, \bar{\mathbf{s}}(\mathbf{x})) > 0$ in our notation). This presents a rather strong condition, and our analysis moves past this limitation. Additionally, our results also hold for the robust adversarial setting. The analysis in [58], on the other hand, links the parameter sensitivity of predictors to generalization using an augmented loss function that encourages favorable data-dependent properties, such as low Jacobian norms. Their bound also avoids exponential dependence on depth but is restricted to smooth activations (and the benign, nonadversarial setting). The work in [8] presents an alternative approach by studying the curvature of the loss as given by the Hessian. While their bound avoids explicit dependencies on the global Lipschitz constant, it is unclear whether all dependence on the latter is avoided in their characterization of the failure probability.

6. Experimental results. In this section, we showcase the potential benefits of sparse local Lipschitz analysis. We compute as sparse the certified radius and sparse regularity for feedforward neural networks trained for classification on MNIST and SVHN datasets. For more complex tasks, such as ImageNet, one needs additional architectural choices like convolution and pooling, which we regard as future extensions to our work.

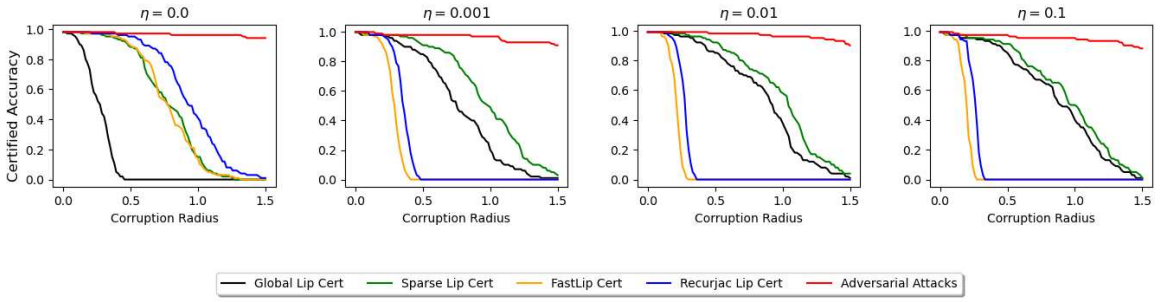
Training setup. We train feedforward networks h with weights $\{\mathbf{W}^k\}_{k=1}^{K+1}$, where $\mathbf{W}^k \in \mathbb{R}^{d^k \times d^{k-1}}$ using the cross-entropy loss with stochastic gradient descent (SGD) with default hyperparameter settings in PyTorch for 2,000 steps with a batch size of 100. Each network is trained with *orthogonal frame regularization,* a measure suggested in [16] for improving robustness that encourages normalized layer weights to be near orthogonal. All trained models[27] h_η are described as follows:

$$h_\eta \in \underset{\{\mathbf{W}^k\}_{k=1}^{K+1}}{\operatorname{argmin}} \frac{1}{m} \sum_{i=1}^{m} \ell\Big(h, (\mathbf{x}_i, y_i)\Big) + \frac{\eta}{K+1} \sum_{k=1}^{K+1} \Big\| I - \tilde{\mathbf{W}}^k (\tilde{\mathbf{W}}^k)^T \Big\|_F^2.$$

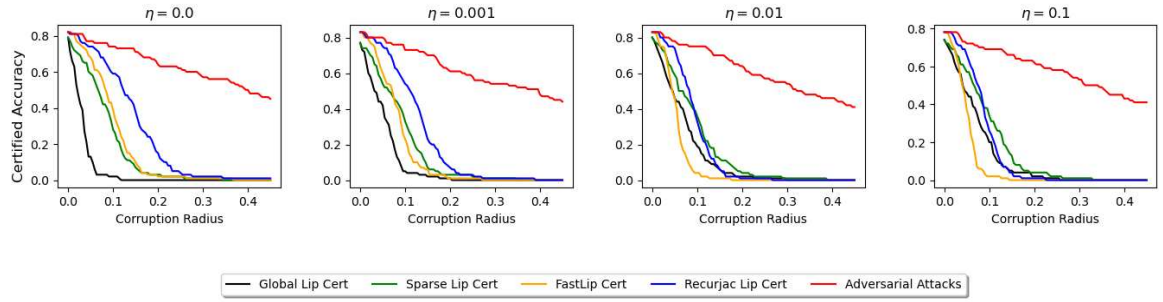
Here $\tilde{\mathbf{W}}^k$ has normalized rows $\tilde{\mathbf{w}}_i^k := \frac{\mathbf{w}_i^k}{\|\mathbf{w}_i^k\|_2}$. We study models trained with four different choices of $\eta \in \{0, 0.001, 0.01, 0.1\}$. For both MNIST and SVHN datasets, the official training

[27] The extra regularization term does not increase the computational cost of training.

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license



(a) MNIST



(b) SVHN

Figure 4. *Security curves for feedforward neural networks with layer widths* [500,500].

sets are randomly split into train and validation data (55,000:5,000 for MNIST, 61,257:12,000 for SVHN). The models are optimized on the training data, and the resulting measures are computed on validation data.

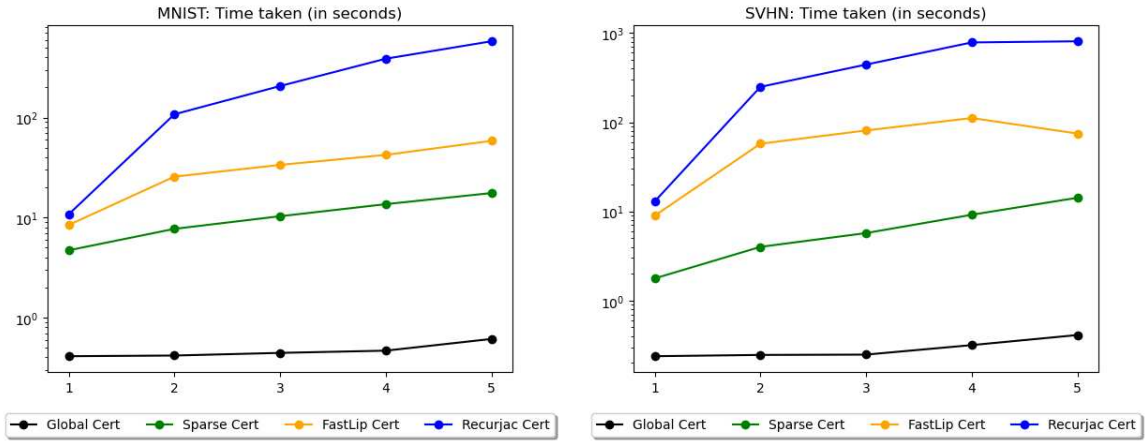
Certified radius. For any network, at each input $\mathbf{x}$ there exists a true robust radius $\bar{r}(\mathbf{x})$—the minimal energy required for a successful adversarial perturbation. The naive lower bound for the certified radius, $r_{\text{global}}(\mathbf{x}) := r_{\text{cert}}(\mathbf{x}, \mathbf{0})$, is computed from Corollary 4.8 at the trivial choice of sparsity. The SLL certificate, $r_{\text{sparse}}(\mathbf{x})$, is obtained by binary search (with tolerance 10^{-6}), and using the optimal sparsity, $\mathbf{s}$ is computed by Algorithm 4.1. The SLL certificate relies on the product of operator norms of specific reduced linear maps. These estimates provide a lower bound $r_{\text{global}}(\mathbf{x}) \leq r_{\text{sparse}}(\mathbf{x}) \leq \bar{r}(\mathbf{x})$. While the value of $\bar{r}(\mathbf{x})$ is not computable,[28] one can obtain a surrogate upper bound $r_{\text{adv}}(\mathbf{x}) \geq \bar{r}(\mathbf{x})$ by measuring the minimal size of an adversarial example found via an ensemble of popular (and effective) adversarial attack strategies, such as PGD [34], Carlini and Wagner [13], etc.

Benchmark via security curves. Using a certified radius, one can compute the *certified accuracy* of a collection of inputs, measured as the fraction of samples that are certified to predict faithfully against a specified size of corruption. The robustness of the trained networks (with two hidden layers of dimension 500) on both MNIST and SVHN datasets are shown in Figures 4(a) and 4(b) via *security curves,* which plot the obtained certified accuracy

[28] Indeed, this is NP hard, as it involves the optimization of a non-convex loss.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license



(a) MNIST (b) SVHN

Figure 5. *Time taken (y-axis) for certification on batch of* 100 *inputs for models with layer dimensions* $[100] * (K + 1)$ *with varying depth* K *(x-axis).*

for increasing size of adversarial perturbations. We compare our "SparseLip" certification based on r_{sparse} with other state-of-the-art certification algorithms that are based on Lipschitz constant estimation: FastLip [59] and RecurJac [64]. Both methods upper bound the Jacobian norm using entrywise bound propagation.

The results in Figures 4(a) and 4(b) demonstrate that orthogonal frame regularization gives improved robustness, as seen via the certified accuracy and the robust accuracy under attack. RecurJac provides the best certification for models trained without regularization, while SparseLip is comparable to FastLip for the same setting. For regularized models, SparseLip provides the best certified accuracy for MNIST and is among the best for SVHN. The performance of both RecurJac and FastLip drops significantly for regularized models. Finally, Figure 5 depicts the considerable computational benefit of our approach, SparseLip.

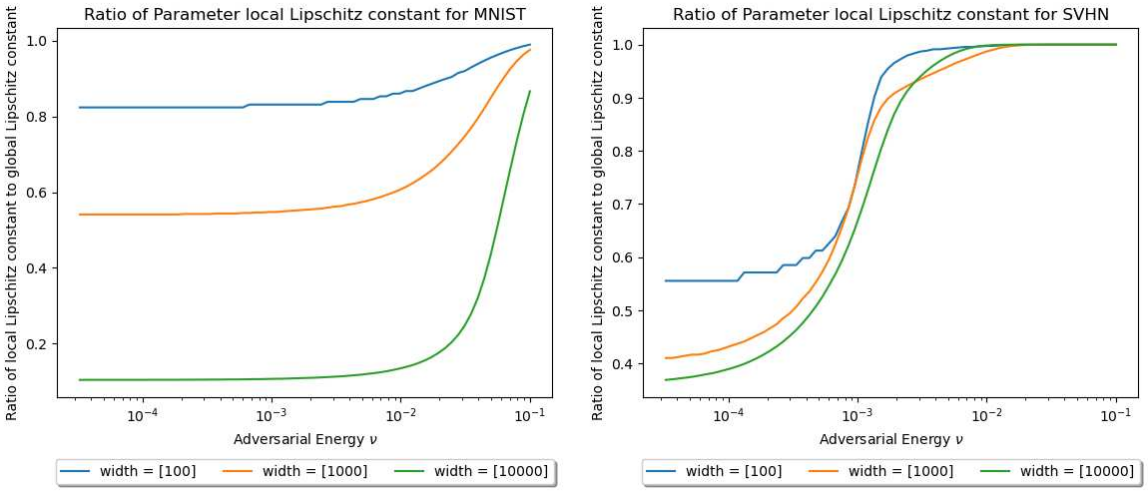
Sparse local Lipschitzness w.r.t. parameter. We finally study how SLL analysis can aid the study of generalization by considering a 1-hidden layer feedforward network of different widths trained via SGD without regularization. Across varying adversarial energy levels ν, Figures 6 and 7 plot the robust sparse regularity $\mathcal{L}_{\text{rob}}(h, \mathcal{V}, \epsilon)$ and robust optimal sparsity level $s^*_{\text{rob}}(h, \mathcal{V}, \epsilon)$ w.r.t. validation data $\mathcal{V}$ as defined in (5.3) at $\epsilon = \frac{1}{|\mathcal{V}|(K+1)}$. We note that for large enough ν the SLL sensitivity is equivalent to global Lipschitz analysis (and correspondingly the optimal sparsity level approaches 0), but for moderate values of ν the robust sparse regularity can be significantly better. Figures 6 and 7 demonstrate the observation in networks with large widths and the ability of SLL analysis to capture the reduced local sensitivity. Importantly, the wider the network, the more significant the reduction in the equivalent Lipschitz scale of the model.

7. Conclusions.

In this work, we study adversarial robustness via the lens of sparse local Lipschitzness (SLL). We show that feedforward neural networks are SLL and equivalent to a reduced *nonlinear* mapping with decreased sensitivity in a local neighborhood around each

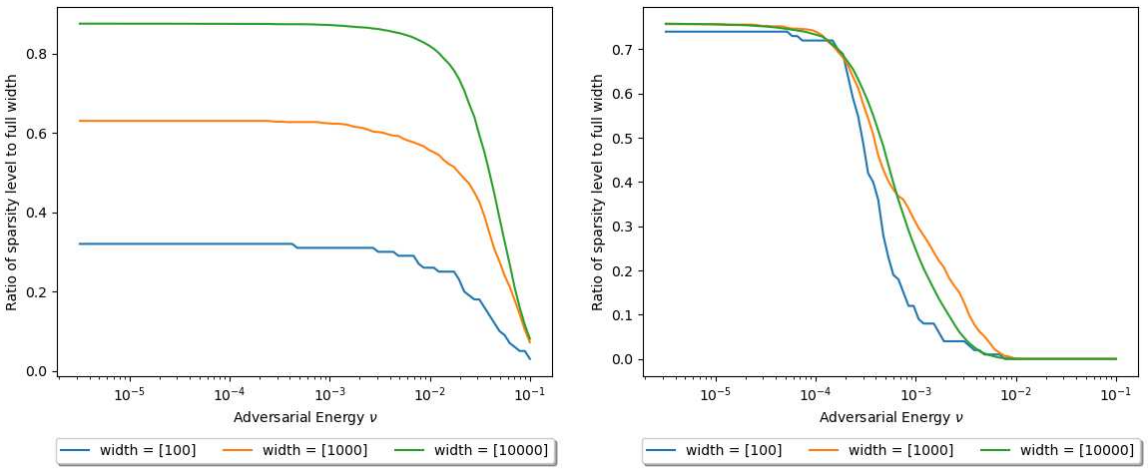
Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license



(a) MNIST (b) SVHN

Figure 6. *Ratio of Lipschitz constants,* $\frac{\mathcal{L}_{\mathrm{rob}}(h,\mathcal{V},\frac{1}{|\mathcal{V}|(K+1)})}{\mathsf{L}_{\mathrm{param},\nu}}$ *for networks of varying widths.*



(a) MNIST (b) SVHN

Figure 7. *Ratio of optimal sparsity to width* d*,* $\frac{s^*_{\mathrm{rob}}(h,\mathcal{V},\frac{1}{|\mathcal{V}|(K+1)})}{d}$ *for networks of varying widths.*

input. In using Lipschitzness properties locally rather than globally, and benefiting from sparse structures, our approach provides an improved certified radius at any input and bounds on the robust generalization error with only a mild dependence on the adversarial corruption. Our work is a step towards producing data-dependent nonuniform bounds that leverage the favorable properties of a trained predictor on a sample datum. We believe that the ideas presented here are extensible to other hypothesis classes that encourage other structural priors, such as convolutional, attention, or graph neural networks. The identification of the particular reduced models for each class presents an intriguing topic of future research.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

Acknowledgment. The authors thank Raman Arora for useful comments in early stages of this work.

REFERENCES

[1] A. Aberdam, D. Simon, and M. Elad, *When and How Can Deep Generative Models Be Inverted?*, https://arxiv.org/abs/2006.15555, 2020.

[2] Z. Allen-Zhu and Y. Li, *Feature Purification: How Adversarial Training Performs Robust Deep Learning*, https://arxiv.org/abs/2005.10190, 2020.

[3] A. Athalye, N. Carlini, and D. A. Wagner, *Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial example*, in Proceedings of the 35th Internatonal Conference on Machine Learning (ICML), 2018.

[4] I. Attias, A. Kontorovich, and Y. Mansour, *Improved Generalization Bounds for Robust Learning*, https://arxiv.org/abs/1810.02180, 2019.

[5] P. Awasthi, N. Frank, and M. Mohri, *Adversarial Learning Guarantees for Linear Hypotheses and Neural Networks*, https://arxiv.org/abs/2004.13617, 2020.

[6] P. Awasthi, H. Jain, A. Rawat, and A. Vijayaraghavan, *Adversarial Robustness via Robust Low Rank Representations*, https://arxiv.org/abs/2007.06555, 2020.

[7] R. Balestriero, R. Cosentino, B. Aazhang, and R. Baraniuk, *The geometry of deep networks: Power diagram subdivision*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2019.

[8] A. Banerjee, T. Chen, and Y. Zhou, *De-randomized BAC-Bayes Margin Bounds: Applications to Non-convex and Non-smooth Predictors*, https://arxiv.org/abs/2002.09956, 2020.

[9] P. L. Bartlett, D. J. Foster, and M. Telgarsky, *Spectrally-normalized margin bounds for neural networks*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2017.

[10] W. Brendel, J. Rauber, and M. Bethge, *Decision-Based Adversarial Attacks: Reliable Attacks against Black-Box Machine Learning Models*, https://arxiv.org/abs/1712.04248, 2018.

[11] S. Bubeck, E. Price, and I. P. Razenshteyn, *Adversarial Examples from Computational Constraints*, https://arxiv.org/abs/1805.10204, 2019.

[12] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, *Adversarial sensor attack on LiDAR-based perception in autonomous driving*, in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019.

[13] N. Carlini and D. Wagner, *Towards Evaluating the Robustness of Neural Networks*, preprint, https://arxiv.org/abs/1608.04644, 2017.

[14] N. Carlini and D. A. Wagner, *Adversarial examples are not easily detected: Bypassing ten detection methods*, in Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017.

[15] L. Chen, Y. Min, M. Zhang, and A. Karbasi, *More data can expand the generalization gap between adversarially robust and standard models*, in Proceedings of the 37th Internatonal Conference on Machine Learning (ICML), 2020.

[16] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, *Parseval networks: Improving robustness to adversarial examples*, in Proceedings of the 34th International Conference on Machine Learning, Proc. Mach. Learn. Res. 70, D. Precup and Y. W. Teh, eds., PMLR, Cambridge, MA, 2017, pp. 854–863, https://proceedings.mlr.press/v70/cisse17a.html.

[17] J. Cohen, E. Rosenfeld, and Z. Kolter, *Certified adversarial robustness via randomized smoothing*, in Proceedings of the 36th International Conference on Machine Learning, Proc. Mach. Learn. Res. 97, K. Chaudhuri and R. Salakhutdinov, eds., PMLR, Cambridge, MA, 2019, pp. 1310–1320, https://proceedings.mlr.press/v97/cohen19c.html.

[18] P. L. Combettes and J.-C. Pesquet, *Lipschitz certificates for layered network structures driven by averaged activation operators*, SIAM J. Math. Data Sci., 2 (2020), pp. 529–557, https://doi.org/10.1137/19M1272780.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, in Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 2019.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

[20] I. Diamant, E. Klang, M. Amitai, E. Konen, J. Goldberger, and H. Greenspan, *Task-driven dictionary learning based on mutual information for medical image classification*, IEEE Trans. Biomed. Eng., 64 (2016), pp. 1380–1392.
[21] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, T. Kohno, and D. X. Song, *Physical Adversarial Examples for Object Detectors*, https://arxiv.org/abs/1807.07769, 2018.
[22] A. Fawzi, H. Fawzi, and O. Fawzi, *Adversarial vulnerability for any classifier*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2018.
[23] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas, *Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks*, https://arxiv.org/abs/1906.04893, 2019.
[24] N. Golowich, A. Rakhlin, and O. Shamir, *Size-independent sample complexity of neural networks*, in Proceedings of the 31st Annual Conference on Learning Theory (COLT), 2018.
[25] F. L. Gómez, P. Rolland, and V. Cevher, *Lipschitz Constant Estimation of Neural Networks via Sparse Polynomial Optimization*, https://arxiv.org/abs/2004.08688, 2020.
[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, https://arxiv.org/abs/1412.6572, 2015.
[27] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[28] M. Hein and M. Andriushchenko, *Formal guarantees on the robustness of a classifier against adversarial manipulation*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2017.
[29] Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, *Fantastic Generalization Measures and Where to Find Them*, https://arxiv.org/abs/1912.02178, 2020.
[30] D. S. Kermany, M. H. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. L. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. N. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. R. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, *Identifying medical diagnoses and treatable diseases by image-based deep learning*, Cell, 172 (2018), pp. 1122–1131.
[31] J. Khim and P.-L. Loh, *Adversarial Risk Bounds via Function Transformation*, preprint, https://arxiv.org/abs/1810.09519, 2018.
[32] A. Kurakin, I. J. Goodfellow, and S. Bengio, *Adversarial Examples in the Physical World*, https://arxiv.org/abs/1607.02533, 2017.
[33] Y. Liu, X. Chen, C. Liu, and D. X. Song, *Delving into Transferable Adversarial Examples and Black-Box Attacks*, https://arxiv.org/abs/1611.02770, 2017.
[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards Deep Learning Models Resistant to Adversarial Attacks*, https://arxiv.org/abs/1706.06083, 2018.
[35] S. Mahloujifar, D. I. Diochnos, and M. Mahmoody, *The Curse of Concentration in Robust Learning: Evasion and Poisoning Attacks from Concentration of Measure*, preprint, https://arxiv.org/abs/1809.03063, 2018.
[36] J. Mairal, F. Bach, and J. Ponce, *Task-driven dictionary learning*, IEEE Trans. Pattern Anal. Mach. Intell., 34 (2011), pp. 791–804.
[37] N. A. Mehta and A. G. Gray, *On the Sample Complexity of Predictive Sparse Coding*, https://arxiv.org/abs/1202.4050, 2012.
[38] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, MIT Press, Cambridge, MA, 2018.
[39] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, *DeepFool: A simple and accurate method to fool deep neural networks*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2574–2582.
[40] V. Nagarajan and J. Z. Kolter, *Deterministic PAC-Bayesian Generalization Bounds for Deep Networks via Generalizing Noise-Resilience*, https://arxiv.org/abs/1905.13344, 2019.
[41] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, *Exploring generalization in deep learning*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2017.
[42] B. Neyshabur, S. Bhojanapalli, D. A. McAllester, and N. Srebro, *A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks*, https://arxiv.org/abs/1707.09564, 2018.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license

[43] B. Neyshabur, R. Tomioka, and N. Srebro, *Norm-based capacity control in neural networks*, in Proceedings of the 28th Annual Conference on Learning Theory (COLT), 2015.
[44] N. Papernot, P. McDaniel, and I. J. Goodfellow, *Transferability in Machine Learning: From Phenomena to Black-Box Attacks Using Adversarial Samples*, https://arxiv.org/abs/1605.07277, 2016.
[45] A. Raghunathan, J. Steinhardt, and P. Liang, *Certified Defenses against Adversarial Examples*, https://arxiv.org/abs/1801.09344, 2018.
[46] H. Salman, G. Yang, J. Li, P. Zhang, H. Zhang, I. P. Razenshteyn, and S. Bubeck, *Provably robust deep learning via adversarially trained smoothed classifiers,* in Proceedings of the Confeence on Neural Information Processing Systems (NIPS), 2019.
[47] K. Scaman and A. Virmaux, *Lipschitz regularity of deep neural networks: Analysis and efficient estimation*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2018.
[48] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, *Are Adversarial Examples Inevitable?*, https://arxiv.org/abs/1809.02104, 2019.
[49] J. Shawe-Taylor and R. C. Williamson, *A pac analysis of a bayesian estimator*, in Proceedings of the 10th Annual Conference on Learning Theory (COLT), 1997.
[50] A. Sinha, H. Namkoong, and J. C. Duchi, *Certifiable Distributional Robustness with Principled Adversarial Training*, https://arxiv.org/abs/1710.10571, 2017.
[51] J. Sulam, A. Aberdam, A. Beck, and M. Elad, *On multi-layer basis pursuit, efficient algorithms and convolutional neural networks*, IEEE Trans. Pattern Anal. Mach. Intell., 42 (2019), pp. 1968–1980.
[52] J. Sulam, R. Muthukumar, and R. Arora, *Adversarial robustness of supervised sparse coding*, Adv. Neural Inf. Process. Syst., 33 (2020), pp. 2110–2121.
[53] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, *Intriguing Properties of Neural Networks*, https://arxiv.org/abs/1312.6199, 2014.
[54] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.
[55] J. A. Tropp, A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss, *Improved sparse approximation over quasiincoherent dictionaries*, in Proceedings of the IEEE International Conference on Image Processing (ICIP), Vol. 1, 2003, pp. 1–37.
[56] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, *Robustness May Be at Odds with Accuracy*, https://arxiv.org/abs/1805.12152, 2019.
[57] Y. Tsuzuku, I. Sato, and M. Sugiyama, *Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks*, Adv. Neural Inf. Process. Syst., 31 (2018), pp. 6541–6550.
[58] C. Wei and T. Ma, *Data-dependent sample complexity of deep neural networks via lipschitz augmentation*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2019.
[59] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. S. Boning, I. S. Dhillon, and L. Daniel, *Towards fast computation of certified robustness for ReLU networks*, in Proceedings of the 37th Internatonal Conference on Machine Learning (ICML), 2018.
[60] E. Wong and Z. Kolter, *Provable defenses against adversarial examples via the convex outer adversarial polytope*, in Proceedings of the 35th International Conference on Machine Learning, Proc. Mach. Learn. Res. 80, J. Dy and A. Krause, eds., PMLR, Cambridge, MA, 2018, pp. 5286–5295, https://proceedings.mlr.press/v80/wong18a.html.
[61] D. Yin, K. Ramchandran, and P. Bartlett, *Rademacher Complexity for Adversarially Robust Generalization*, preprint, https://arxiv.org/abs/1810.11914, 2018.
[62] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, *Efficient neural network robustness certification with general activation functions*, in Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2018.
[63] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, *Theoretically principled trade-off between robustness and accuracy*, in Proceedings of the 36th International Conference on Machine Learning, Proc. Mach. Learn. Res. 97, K. Chaudhuri and R. Salakhutdinov, eds., PMLR, Cambridge, MA, 2019, pp. 7472–7482, https://proceedings.mlr.press/v97/zhang19p.html.
[64] H. Zhang, P. Zhang, and C.-J. Hsieh, *RecurJac: An Efficient Recursive Algorithm for Bounding Jacobian Matrix of Neural Networks and its Applications*, https://arxiv.org/abs/1810.11783, 2019.
[65] Y. Zhang, O. Plevrakis, S. S. Du, X. Li, Z. Song, and S. Arora, *Over-parameterized Adversarial Training: An Analysis Overcoming the Curse of Dimensionality*, https://arxiv.org/abs/2002.06668, 2020.

Downloaded 10/09/24 to 73.201.213.137 . Redistribution subject to CCBY license

© 2023 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license