Second-Order Provable Defenses against Adversarial Attacks

Sahil Singla [1] **Soheil Feizi** [1]

Abstract

A robustness certificate against adversarial examples is the minimum distance of a given input to the decision boundary of the classifier (or its lower bound). For *any* perturbation of the input with a magnitude smaller than the certificate value, the classification output will provably remain unchanged. Computing exact robustness certificates for neural networks is difficult in general since it requires solving a non-convex optimization. In this paper, we provide computationally-efficient robustness certificates for neural networks with differentiable activation functions in two steps. First, we show that if the eigenvalues of the Hessian of the network (curvatures of the network) are bounded (globally or locally), we can compute a robustness certificate in the l_2 norm efficiently using convex optimization. Second, we derive a computationally-efficient differentiable upper bound on the curvature of a deep network. We also use the curvature bound as a regularization term during the training of the network to boost its certified robustness. Putting these results together leads to our proposed **C**urvature-based **R**obustness **C**ertificate (CRC) and **C**urvature-based **R**obust **T**raining (CRT). Our numerical results show that CRT leads to significantly higher certified robust accuracy compared to interval-bound propagation (IBP) based training. We achieve certified robust accuracy 69.79%, 57.78% and 53.19% while IBP-based methods achieve 44.96%, 44.74% and 44.66% on 2,3 and 4 layer networks respectively on the MNIST-dataset.

1. Introduction

Modern neural networks achieve high accuracy on tasks such as image classification and speech recognition, but are known to be brittle to small, adversarially chosen perturbations of their inputs (Szegedy et al., 2014). A classifier which correctly classifies an image $\mathbf{x}$, can be fooled by an adversary to misclassify an *adversarial example* $\mathbf{x} + \delta$, such that $\mathbf{x} + \delta$ is indistinguishable from $\mathbf{x}$ to a human. Adversarial examples can also fool systems when they are printed out on a paper and photographed with a smart phone (Kurakin et al., 2016a). Even in a black box threat model, where the adversary has no access to the model parameters, attackers could target autonomous vehicles by using stickers or paint to create an adversarial stop sign that the vehicle would interpret as a 'yield' or another sign (Papernot et al., 2016). This trend is worrisome and suggests that adversarial vulnerabilities need to be appropriately addressed before neural networks can be deployed in security critical applications.

In this work, we propose a new approach for developing provable defenses against ℓ_2-bounded adversarial attacks as well as computing robustness certifications of pre-trained deep networks with differentiable activations. In contrast to the existing certificates (Weng et al., 2018; Zhang et al., 2018b) that use the first-order information (upper and lower bounds on the slope), our approach is based on the second-order information (upper and lower bounds on curvature values i.e. eigenvalues of the Hessian). Our approach is based on two key theoretically-justified steps: First, in Theorems 1 and 2, we show that if the eigenvalues of the Hessian of the network (curvatures of the network) are bounded (globally or locally), we can efficiently compute a robustness certificate and develop a defense method against ℓ_2-bounded adversarial attacks using convex optimization. Second, in Theorem 4, we derive a computationally-efficient differentiable bound on the curvature (eigenvalues of the Hessian) of a deep network. We derive this bound by explicitly characterizing the Hessian of a deep network in Lemma 1.

Although the problem of finding the closest adversarial example to a given point for deep nets leads to a non-convex optimization problem, our proposed Curvature-based Robustness Certificate (CRC), under some verifiable conditions, is able to compute points on the decision boundary that are provably closest to the input. That is, it provides

[1]Department of Computer Science, University of Maryland, College Park. Correspondence to: Sahil Singla <ssingla@cs.umd.edu>, Soheil Feizi <sfeizi@cs.umd.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

the tightest certificate in those cases. For example, for a 2,3,4 layer networks trained on MNIST, we can find provably closest adversarial points for 44.17%, 22.59%, 19.53% cases, respectively (Table 2). To the best of our knowledge, our method is the first approach that can efficiently compute provably closest adversarial examples for a significant fraction of examples in non-trivial neural networks.

We note that un-regularized networks, specially deep ones, can obtain large curvature bounds which can lead to small robustness certificates. However, by using the derived curvature bound as a regularizer during training, we significantly decrease curvature values of the network, with little or no decrease in its performance (Table 5, Figure 1). Using this technique, our method significantly outperforms interval-bound propagation (IBP) (Wong et al., 2018; Zhang et al., 2019a) and achieves state of the art certified accuracy (Tables 3 and 4). In particular, our method achieves certified robust accuracy 69.79%, 57.78% and 53.19% while IBP-based methods achieve 44.96%, 44.74% and 44.66% on 2,3 and 4 layer networks, respectively, on the MNIST-dataset (similar results for Fashion-MNIST).

Other recent works (e.g. Moosavi Dezfooli et al. (2019); Qin et al. (2019)) empirically show that using an *estimate* of curvature at inputs as a regularizer leads to *empirical* robustness on par with the adversarial training. In this work, however, we use a bound on the absolute value of curvature (and not an estimate) as a regularizer and show that it results in high *certified* robustness. Moreover, previous works have tried to certify robustness by bounding the Lipschitz constant of the neural network (Anil et al., 2018; Hein & Andriushchenko, 2017; Peck et al., 2017; Szegedy et al., 2014; Zhang et al., 2018c). Our approach, however, is based on bounding the Lipschitz constant of the gradient which in turn leads to bound on the eigenvalues of the Hessian of deep neural networks.

In summary, we make the following contributions:

- We derive a closed-form expression for the Hessian of a deep network with differentiable activation functions (Lemma 1) and derive bounds on the curvature using this closed-form formula (Theorems 3 and 4).
- We develop computationally efficient methods for both the robustness certification as well as the adversarial attack problems (Theorems 1 and 2).
- We provide verifiable conditions under which our method is able to compute points on the decision boundary that are provably closest to the input. Empirically, we show that this condition holds for a significant fraction of examples (Table 2).
- We show that using our proposed curvature bounds as a regularizer during training leads to improved certified accuracy on 2,3 and 4 layer networks (on the MNIST and Fashion-MNIST datasets) compared to IBP-based adversarial training (Wong & Kolter, 2017; Zhang et al., 2019a) (Tables 3 and 4). Our robustness certificate (CRC) outperforms CROWN's certificate (Zhang et al., 2018b) significantly when trained with our regularizer (Table 5).

To the best of our knowledge, this is the first work that (a) demonstrates the utility of second-order information for provable robustness, (b) derives a framework to find the exact robustness certificates in the l_2 norm and the exact worst case adversarial perturbation in an l_2 ball of given a radius under some conditions, and (c) derives an exact closed form expression for the Hessian and bounds on the curvature values using the same.

2. Related work

In the last couple of years, several *empirical defenses* have been proposed for training classifiers to be robust against adversarial perturbations (Kurakin et al., 2016b; Madry et al., 2018; Miyato et al., 2017; Papernot et al., 2016; Samangouei et al., 2018; Zhang et al., 2019b; Zheng et al., 2016) Although these defenses robustify classifiers to particular types of attacks, they can be still vulnerable against stronger attacks (Athalye & Carlini, 2018; Athalye et al., 2018; Carlini & Wagner, 2017; Laidlaw & Feizi, 2019; Uesato et al., 2018). For example, (Athalye et al., 2018) showed most of the empirical defenses proposed in ICLR 2018 can be broken by developing tailored attacks for each of them.

To end the cycle between defenses and attacks, a line of work on *certified defenses* has gained attention where the goal is to train classifiers whose predictions are *provably* robust within some given region (Bunel et al., 2017; Carlini et al., 2017; Cheng et al., 2017; Croce et al., 2018; Dutta et al., 2018; Dvijotham et al., 2018a;b; Ehlers, 2017; Fischetti & Jo, 2018; Gehr et al., 2018; Gowal et al., 2018; Huang et al., 2016; Katz et al., 2017; Levine & Feizi, 2020a;b;c; Lomuscio & Maganti, 2017; Mirman et al., 2018; Raghunathan et al., 2018a;b; Singh et al., 2018; Wang et al., 2018a;b; Weng et al., 2018; Wong & Kolter, 2017; Wong et al., 2018; Zhang et al., 2018b; 2019a). These methods, however, do not scale to large and practical networks used in solving modern machine learning problems. Another line of defense work focuses on *randomized smoothing* where the prediction is robust within some region around the input with a user-chosen probability (Cao & Gong, 2017; Cohen et al., 2019; Lécuyer et al., 2018; Li et al., 2018; Liu et al., 2017; Salman et al., 2019). Although these methods can scale to large networks, certifying robustness with probability close to 1 often requires generating a large number of noisy samples around the input which leads to high inference-time computational complexity. We discuss existing works in

Table 1. A summary of various primal and dual concepts used in the paper. f denotes the function of the decision boundary, i.e. $\mathbf{z}_y^{(L)} - \mathbf{z}_t^{(L)}$ where y is the true label and t is the attack target. m and M are lower and upper bounds on the smallest and largest eigenvalues of the Hessian of f, respectively.

	Certificate problem $_{(-)\,=\,cert}$	Attack problem $_{(-)\,=\,attack}$
primal problem, $p^*_{(-)}$	$\min_{f(\mathbf{x})=0} 1/2\lVert\mathbf{x}-\mathbf{x}^{(0)}\rVert^2$	$\min_{\lVert\mathbf{x}-\mathbf{x}^{(0)}\rVert\leq\rho} f(\mathbf{x})$
dual function, $d_{(-)}(\eta)$	$\min_{\mathbf{x}} 1/2\lVert\mathbf{x}-\mathbf{x}^{(0)}\rVert^2 + \eta f(\mathbf{x})$	$\min_{\mathbf{x}} f(\mathbf{x}) + \eta/2(\lVert\mathbf{x}-\mathbf{x}^{(0)}\rVert^2 - \rho^2)$
When is dual solvable?	$-1/M \leq \eta \leq -1/m$	$-m \leq \eta$
dual problem, $d^*_{(-)}$	$\max_{-1/M\leq\eta\leq-1/m} d_{cert}(\eta)$	$\max_{-m\leq\eta} d_{attack}(\eta)$
When primal = dual?	$f(\mathbf{x}^{(cert)}) = 0$	$\lVert\mathbf{x}^{(attack)} - \mathbf{x}^{(0)}\rVert = \rho$

more details in Appendix A.

3. Notation

Consider a fully connected neural network with L layers and N_I neurons in the I^{th} layer ($L \geq 2$ and $I \in [L]$) for a multi-label classification problem with C classes ($N_L = C$). The corresponding function of the neural network is $\mathbf{z}^{(L)} : \mathbf{R}^D \to \mathbf{R}^C$ where D is the dimension of the input. For an input $\mathbf{x}$, we use $\mathbf{z}^{(I)}(\mathbf{x}) \in \mathbf{R}^{N_I}$ and $\mathbf{a}^{(I)}(\mathbf{x}) \in \mathbf{R}^{N_I}$ to denote the input (*before* applying the activation function) and output (*after* applying the activation function) of neurons in the I^{th} hidden layer of the network, respectively. To simplify notation and when no confusion arises, we make the dependency of $\mathbf{z}^{(I)}$ and $\mathbf{a}^{(I)}$ to $\mathbf{x}$ implicit. We define $\mathbf{a}^{(0)}(\mathbf{x}) = \mathbf{x}$ and $N_0 = D$.

With a fully connected architecture, each $\mathbf{z}^{(I)}$ and $\mathbf{a}^{(I)}$ is computed using a transformation matrix $\mathbf{W}^{(I)} \in R^{N_I \times N_{I-1}}$, the bias vector $\mathbf{b}^{(I)} \in R^{N_I}$ and an activation function $\sigma(.)$ as follows:

$$\mathbf{z}^{(I)} = \mathbf{W}^{(I)}\mathbf{a}^{(I-1)} + \mathbf{b}^{(I)}, \qquad \mathbf{a}^{(I)} = \sigma\left(\mathbf{z}^{(I)}\right)$$

We use $(\mathbf{z}_i^{(L)} - \mathbf{z}_j^{(L)})(\mathbf{x})$ as a shorthand for $\mathbf{z}_i^{(L)}(\mathbf{x}) - \mathbf{z}_j^{(L)}(\mathbf{x})$.

We use $[p]$ to denote the set $\{1, \ldots, p\}$ and $[p, q],\ p \leq q$ to denote the set $\{p, p+1, \ldots, q\}$. We use small letters i, j, k etc to denote the index over a vector or rows of a matrix and capital letters I, J to denote the index over layers of network. The element in the i^{th} position of a vector $\mathbf{v}$ is given by $\mathbf{v}_i$, the vector in the i^{th} row of a matrix $\mathbf{A}$ is $\mathbf{A}_i$ while the element in the i^{th} row and j^{th} column of $\mathbf{A}$ is $\mathbf{A}_{i,j}$. We use $\lVert\mathbf{v}\rVert$ and $\lVert\mathbf{A}\rVert$ to denote the 2-norm and the operator 2-norm of the vector $\mathbf{v}$ and the matrix $\mathbf{A}$, respectively. We use $|\mathbf{v}|$ and $|\mathbf{A}|$ to denote the vector and matrix constructed by taking the elementwise absolute values. We use $\lambda_{max}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A})$ to denote the largest and smallest eigenvalues of a symmetric matrix $\mathbf{A}$. We use $diag(\mathbf{v})$ to denote the diagonal matrix constructed by placing each element of $\mathbf{v}$ along the diagonal. We use $\odot$ to denote the Hadamard Product, $\mathbf{I}$ to denote the identity matrix. We use $\preccurlyeq$ and $\succcurlyeq$ to denote Linear Matrix Inequalities (LMIs) such that given two symmetric matrices $\mathbf{A}$ and $\mathbf{B}$ where $\mathbf{A} \succcurlyeq \mathbf{B}$ means $\mathbf{A} - \mathbf{B}$ Positive Semi-Definite (PSD).

4. Using duality to solve the attack and certificate problems

Consider an input $\mathbf{x}^{(0)}$ with true label y and attack target t. In the certificate problem, our goal is to find a lower bound of minimum l_2 distance between $\mathbf{x}^{(0)}$ and decision boundary $f(\mathbf{x}) = 0$ where $f(\mathbf{x}) = (\mathbf{z}_y^{(L)} - \mathbf{z}_t^{(L)})(\mathbf{x})$. The problem for solving the exact distance (*primal*) can be written as:

$$p^*_{cert} = \min_{f(\mathbf{x})=0} \left[\frac{1}{2}\left\lVert\mathbf{x} - \mathbf{x}^{(0)}\right\rVert^2\right]$$

$$p^*_{cert} = \min_{\mathbf{x}} \max_{\eta} \left[\frac{1}{2}\left\lVert\mathbf{x} - \mathbf{x}^{(0)}\right\rVert^2 + \eta f(\mathbf{x})\right] \tag{1}$$

However, solving the above problem can be hard in general. Using the minimax theorem (primal ≥ dual), we can write the *dual* of the above problem as follows:

$$p^*_{cert} \geq \max_{\eta} d_{cert}(\eta)$$

$$d_{cert}(\eta) = \min_{\mathbf{x}} \left[\frac{1}{2}\left\lVert\mathbf{x} - \mathbf{x}^{(0)}\right\rVert^2 + \eta f(\mathbf{x})\right] \tag{2}$$

From the theory of duality, we know that $d_{cert}(\eta)$ for each value of η gives a lower bound on the exact certification value (the primal solution) p^*_{cert}. However, since f is non-convex, solving $d_{cert}(\eta)$ for every η can be difficult. In the next section, we will prove that the curvature of the function f is bounded globally:

$$m\mathbf{I} \preccurlyeq \nabla^2_{\mathbf{x}} f \preccurlyeq M\mathbf{I} \qquad \forall \mathbf{x} \in \mathbb{R}^D \tag{3}$$

In this case, we have the following theorem (d^*_{cert} is defined in Table 1):

Theorem 1. *$d_{cert}(\eta)$ is a convex optimization problem for $-1/M \leq \eta \leq -1/m$. Moreover, If $\mathbf{x}^{(cert)}$ is the solution to d^*_{cert} such that $f(\mathbf{x}^{(cert)}) = 0$, then $p^*_{cert} = d^*_{cert}$.*

Below, we briefly outline the proof while the full proof is presented in Appendix E.1. The Hessian of the *objective function* of the dual $d_{cert}(\eta)$, i.e the function inside the $\min_{\mathbf{x}}$ is given by:

$$\nabla^2_{\mathbf{x}}\left[\frac{1}{2}\left\|\mathbf{x}-\mathbf{x}^{(0)}\right\|^2+\eta f(\mathbf{x})\right]=\mathbf{I}+\eta\nabla^2_{\mathbf{x}}f$$

From equation (3), we know that the eigenvalues of $\mathbf{I}+\eta\nabla^2_{\mathbf{x}}f$ are bounded between $(1+\eta m, 1+\eta M)$ if $\eta \geq 0$, and in $(1+\eta M, 1+\eta m)$ if $\eta \leq 0$. In both cases, we can see that for $-1/M \leq \eta \leq -1/m$, all eigenvalues will be non-negative, making the objective function convex. When $\mathbf{x}^{(cert)}$ satisfies $f(\mathbf{x}^{(cert)})=0$, we have $d^*_{cert}=1/2\|\mathbf{x}^{(cert)}-\mathbf{x}^{(0)}\|^2$. Using the duality theorem we have $d^*_{cert}\leq p^*_{cert}$ and from the definition of p^*_{cert}, we have $p^*_{cert}\leq d^*_{cert}$. Combining the two inequalities, we get $p^*_{cert}=d^*_{cert}$.

Next, we consider the attack problem. The goal here is to find an adversarial example inside an l_2 ball of radius ρ such that $f(\mathbf{x})$ is minimized. Using similar arguments, we can get the following theorem for the attack problem (p^*_{attack}, d^*_{attack} and d_{attack} are defined in Table 1):

Theorem 2. *$d_{attack}(\eta)$ is a convex optimization problem for $-m \leq \eta$. Moreover, if $\mathbf{x}^{(attack)}$ is the solution to d^*_{attack} such that $\left\|\mathbf{x}^{(attack)}-\mathbf{x}^{(0)}\right\|=\rho$, $p^*_{attack}=d^*_{attack}$.*

The proof is presented in Appendix E.2. We note that both Theorems 1 and 2 hold for any non-convex function with continuous gradients. Thus they can also be of interest in problems such as optimization of neural nets.

Using Theorems 1 and 2, we have the following definitions for certification and attack optimizations:

Definition 1. ***(Curvature-based Certificate Optimization)*** *Given an input $\mathbf{x}^{(0)}$ with true label y, false target t, we define $(\eta^{(cert)}, \mathbf{x}^{(cert)})$ as the solution of the following max-min optimization:*

$$\max_{-1/M\leq\eta\leq-1/m}\min_{\mathbf{x}}\left[\frac{1}{2}\left\|\mathbf{x}-\mathbf{x}^{(0)}\right\|^2+\eta f(\mathbf{x})\right]$$

*We refer to $\left\|\mathbf{x}^{(cert)}-\mathbf{x}^{(0)}\right\|$ as the **C**urvature-based **R**obustness **C**ertificate (CRC).*

Definition 2. ***(Curvature-based Attack Optimization)*** *Given input $\mathbf{x}^{(0)}$ with label y, false target t, and the l_2 ball radius ρ, we define $(\eta^{(attack)}, \mathbf{x}^{(attack)})$ as the solution of the following optimization:*

$$\max_{\eta\geq-m}\min_{\mathbf{x}}\left[\frac{\eta}{2}\left(\left\|\mathbf{x}-\mathbf{x}^{(0)}\right\|^2-\rho^2\right)+f(\mathbf{x})\right]$$

*When $\mathbf{x}^{(attack)}$ is used for training in an adversarial training framework, we call the method the **C**urvature-based **R**obust **T**raining (CRT).*

A direct implication of Theorems 1 and 2 is that the tightness of our robustness certificate crucially depends on the tightness of our curvature bounds, m and M. If m and M are very large compared to the true eigenvalue bounds of the Hessian of the network, the resulting robustness certificate will be vacuous. In Table 5 (and Figure 1), we show that by adding the derived bound as a regularization term during the training, we can significantly decrease curvature bounds of the network, with little or no decrease in its performance. This leads to high robustness certifications against adversarial attacks.

5. Curvature Bounds for deep networks

In this section, we provide a computationally efficient approach to compute the curvature bounds for neural networks with differentiable activation functions. To the best of our knowledge, there is no prior work on finding provable bounds on the curvature values of deep neural networks.

5.1. Closed form expression for the Hessian

Using the chain rule of second derivatives, we can derive $\nabla^2_{\mathbf{x}}\mathbf{z}^{(L)}_i$ as a sum of matrix products:

Lemma 1. *Given an L layer neural network, the Hessian of the i^{th} hidden unit with respect to the input $\mathbf{x}$, i.e $\nabla^2_{\mathbf{x}}\mathbf{z}^{(L)}_i$ is given by the following formula:*

$$\nabla^2_{\mathbf{x}}\mathbf{z}^{(L)}_i=\sum_{I=1}^{L-1}\left(\mathbf{B}^{(I)}\right)^T diag\left(\mathbf{F}^{(L,I)}_i\odot\sigma''\left(\mathbf{z}^{(I)}\right)\right)\mathbf{B}^{(I)}$$

where $\mathbf{B}^{(I)}$ is the Jacobian of $\mathbf{z}^{(I)}$ with respect to $\mathbf{x}$ (dimensions $N_I\times D$), and $\mathbf{F}^{(L,I)}$ is the Jacobian of $\mathbf{z}^{(L)}$ with respect to $\mathbf{a}^{(I)}$ (dimensions $N_L\times N_I$).

The proof is presented in Appendix E.3. Using the chain rule, we can compute $\mathbf{B}^{(I)}$, $\mathbf{F}^{(L,I)}$ matrices in Lemma 1 recursively as follows:

$$\mathbf{B}^{(I)}=\begin{cases}\mathbf{W}^{(1)}, & I=1\\ \mathbf{W}^{(I)}diag\left(\sigma'\left(\mathbf{z}^{(I-1)}\right)\right)\mathbf{B}^{(I-1)}, & I\geq 2\end{cases}$$

$$\mathbf{F}^{(L,I)}=\begin{cases}\mathbf{W}^{(L)}, & I=L-1\\ \mathbf{W}^{(L)}diag\left(\sigma'\left(\mathbf{z}^{(L-1)}\right)\right)\mathbf{F}^{(L-1,I)}, & I\leq L-2\end{cases}$$

This leads to a fast back-propagation like method that can be used to compute the Hessian. Note that Lemma 1 only assumes a matrix multiplication operation from $\mathbf{a}^{(I-1)}$ to $\mathbf{z}^{(I)}$. Since a convolution operation can also be expressed as a matrix multiplication, we can directly extend this lemma to deep convolutional networks. Furthermore, Lemma 1 can also be of independent interest in other related problems such as second-order interpretation methods for deep learning (e.g. (Singla et al., 2019)).

5.2. Curvature bounds for Two Layer networks

For a two-layer network and using Lemma 1, $\nabla^2_{\mathbf{x}}\left(\mathbf{z}^{(2)}_y - \mathbf{z}^{(2)}_t\right)$ is given by:

$$\left(\mathbf{W}^{(1)}\right)^T diag\Bigg(\left(\mathbf{W}^{(2)}_y - \mathbf{W}^{(2)}_t\right) \odot \sigma^{''}\left(\mathbf{z}^{(1)}\right)\Bigg)\mathbf{W}^{(1)}$$

In the above equation, note that only the term $\sigma^{''}(\mathbf{z}^{(1)})$ depends on $\mathbf{x}$. We can maximize and minimize each element in the diag term, $(\mathbf{W}^{(2)}_{y,i} - \mathbf{W}^{(2)}_{t,i})\sigma^{''}(\mathbf{z}^{(1)}_i)$ independently subject to the constraint that $\sigma^{''}(.)$ is bounded. Using this procedure, we construct matrices $\mathbf{P}$ and $\mathbf{N}$ that satisfy properties given in the following theorem:

Theorem 3. *Given a two layer network whose activation function has bounded second derivative:*

$$h_L \leq \sigma^{''}(x) \leq h_U \quad \forall x \in \mathbb{R}$$

(a) We have the following linear matrix inequalities (LMIs):

$$\mathbf{N} \preccurlyeq \nabla^2_{\mathbf{x}}\left(\mathbf{z}^{(2)}_y - \mathbf{z}^{(2)}_t\right) \preccurlyeq \mathbf{P} \qquad \forall \mathbf{x} \in \mathbb{R}^D$$

(b) If $h_U \geq 0$ and $h_L \leq 0$, $\mathbf{P}$ is PSD, $\mathbf{N}$ is a NSD matrix.

(c) This gives the following global bounds on the eigenvalues of the Hessian:

$$m\mathbf{I} \preccurlyeq \nabla^2_{\mathbf{x}}\left(\mathbf{z}^{(2)}_y - \mathbf{z}^{(2)}_t\right) \preccurlyeq M\mathbf{I} \tag{4}$$

where $M = \lambda_{max}(\mathbf{P}),\ m = \lambda_{min}(\mathbf{N})$

$\mathbf{P}$ and $\mathbf{N}$ are independent of $\mathbf{x}$ and defined in equations (51) and (52) in Appendix E.4.

The proof is presented in Appendix E.4. Because power iteration finds the eigenvalue with largest magnitude, we can use it to find m and M only when $\mathbf{P}$ is PSD and $\mathbf{N}$ is NSD. We solve for h_U and h_L for sigmoid, tanh, softplus activation functions in Appendix F and show that this is in fact the case for them.

We note that this result does not hold for ReLU networks since the ReLU function is not differentiable everywhere. However, in Appendix *G*, we devise a method to compute the certificate for a two layer ReLU network by finding a quadratic function that is a provable lower bound for $\mathbf{z}^{(2)}_y - \mathbf{z}^{(2)}_t$. We show that the resulting method significantly outperforms CROWN-Ada (see Appendix Table 9).

5.3. Curvature bounds for Deep networks

Using Lemma 1, we know that $\nabla^2_{\mathbf{x}}\mathbf{z}^{(L)}_i$ is a sum product of matrices $\mathbf{B}^{(I)}$ and $\mathbf{F}^{(L,I)}_i$. Thus, if we can find upper bounds for $\|\mathbf{B}^{(I)}\|$ and $\|\mathbf{F}^{(L,I)}_i\|_\infty$, we can get upper bounds for $\|\nabla^2_{\mathbf{x}}\mathbf{z}^{(L)}_i\|$. Using this intuition (proof is presented in Appendix E.5), we have the following result:

Theorem 4. *Given an L layer neural network whose activation function satifies:*

$$|\sigma^{'}(x)| \leq g,\ |\sigma^{''}(x)| \leq h \qquad \forall x \in \mathbb{R},$$

the absolute value of eigenvalues of $\nabla^2_{\mathbf{x}}\mathbf{z}^{(L)}_i$ is globally bounded by the following quantity:

$$\left\|\nabla^2_{\mathbf{x}}\mathbf{z}^{(L)}_i\right\| \leq h\sum_{I=1}^{L-1}\left(r^{(I)}\right)^2 \max_j\left(\mathbf{S}^{(L,I)}_{i,j}\right), \quad \forall \mathbf{x} \in \mathbb{R}^D$$

where $r^{(I)}$ and $\mathbf{S}^{(L,I)}$ are independent of $\mathbf{x}$ and defined recursively as:

$$r^{(I)} = \begin{cases} \left\|\mathbf{W}^{(1)}\right\|, & I = 1 \\ g\left\|\mathbf{W}^{(I)}\right\| r^{(I-1)}, & I \geq 2 \end{cases} \tag{5}$$

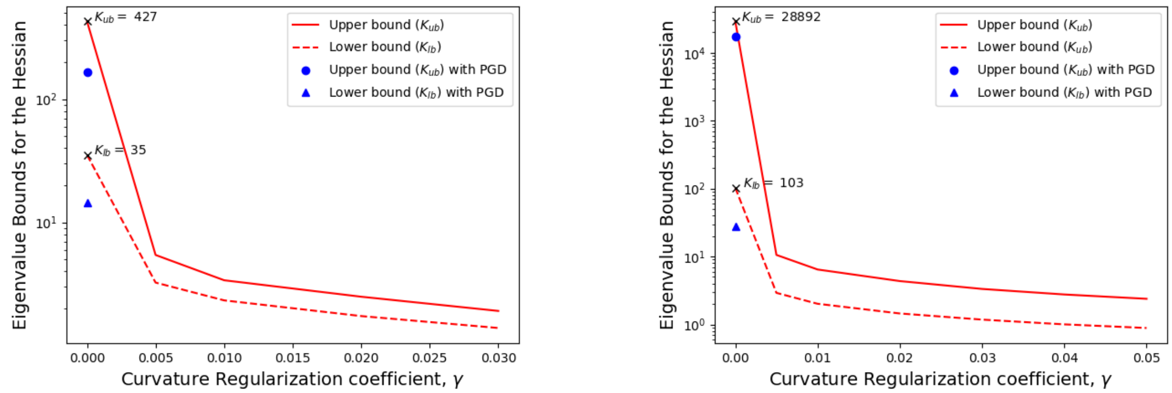
$$\mathbf{S}^{(L,I)} = \begin{cases} \left|\mathbf{W}^{(L)}\right|, & I = L-1 \\ g\left|\mathbf{W}^{(L)}\right| \mathbf{S}^{(L-1,I)}, & I \leq L-2 \end{cases} \tag{6}$$

The above expressions allows for an extremely efficient computation of the curvature bounds for deep networks. We consider simplification of this result for sigmoid, tanh, softplus activations in Appendix F. The curvature bounds for $\mathbf{z}^{(L)}_y - \mathbf{z}^{(L)}_t$ can be computed by replacing $\mathbf{W}^{(L)}_i$ with $\mathbf{W}^{(L)}_y - \mathbf{W}^{(L)}_t$ in Theorem 4. The resulting bound is independent of $\mathbf{x}$, and only depends on network weights $\mathbf{W}^{(I)}$, the true label y, and the target t. We denote it with $K(\mathbf{W}, y, t)$. To simplify notation, when no confusion arises we denote it with K. In our experiments, for two layer networks, we use M, m from Theorem 3 since it provides tighter curvature bounds. For deeper networks ($L \geq 3$), we use $M = K,\ m = -K$.

6. Adversarial training with curvature regularization

Since the term $\mathbf{B}^{(I)}$ in Lemma 1 is the Jacobian of $\mathbf{z}^{(I)}$ with respect to $\mathbf{x}$, $\|\mathbf{B}^{(I)}\|$, it is equal to the lipschitz constant of the neural network constructed from the first I layers of the original network. Finding tight bounds on the lipschitz constant is an active area of research (Fazlyab et al., 2019; Scaman & Virmaux, 2018; Weng et al., 2018) and the product of the operator norm of weight matrices is known to be a loose bound on the lipschitz constant for deep networks. Since we use the same product to compute the bound for $\|\mathbf{B}^{(I)}\|$ in Theorem 4, the resulting curvature bound is likely to be loose for very deep networks.

In Figure 1, we observe the same trend: as the depth of the network increases, the upper bound K_{ub} computed using Theorem 4 becomes significantly larger than the lower



(a) 2 layer network (b) 3 layer network

Figure 1. Illustration of lower (K_{lb}) and upper (K_{ub}) bounds on the curvature of 2 and 3 layer networks with sigmoid activations trained on MNIST. Without any curvature regularization (γ = 0), curvature bounds increase significantly for deeper networks. Similarly with γ = 0, networks adversarially trained with PGD have high curvature as well (note the log scale of the y-axis). However, using our curvature bound as a regularizer, the bound becomes tight and CRC gives high certificate values (Table 5). We report the curvature bounds (K_{lb} and K_{ub}) for networks with different depths in Appendix Table 10.

bound K_{lb} (computed by taking the maximum of the largest eigenvalue of the Hessian across all test images with label y and the second largest logit t, then averaging across different (y, t)). However, by regularizing the network to have small curvature during training, the bound becomes significantly tighter. Interestingly, using curvature regularization, even with this loose curvature bound for deep nets, we achieve significantly higher robust accuracy than the current state of the art methods while enjoying significantly higher standard accuracy as well (see Tables 3 and 4).

To regularize the network to have small curvature values, we penalize the curvature bound K during training. To compute the gradient of K with respect to the network weights, note that using Theorem 4, we can compute K using absolute value, matrix multiplications, and operator norm. Since the gradient of operator norm does not exist in standard libraries, we created a new layer where the gradient of $\|\mathbf{W}^{(I)}\|$ i.e $\nabla_{\mathbf{W}^{(I)}} \|\mathbf{W}^{(I)}\|$ is given by:

$$\nabla_{\mathbf{W}^{(I)}} \|\mathbf{W}^{(I)}\| = \mathbf{u}^{(I)} \left(\mathbf{v}^{(I)}\right)^T$$
$$\text{where} \quad \mathbf{W}^{(I)}\mathbf{v}^{(I)} = \|\mathbf{W}^{(I)}\|\mathbf{u}^{(I)}$$

Note that $\|\mathbf{W}^{(I)}\|$, $\mathbf{u}^{(I)}$ and $\mathbf{v}^{(I)}$ can be computed using power iteration. Since the network weights do not change significantly during a single training step, we can use the singular vectors $\mathbf{u}^{(I)}$ and $\mathbf{v}^{(I)}$ computed in the previous training step to update $\mathbf{W}^{(I)}$ using one iteration of power method. This approach to compute the gradient of the largest singular value of a matrix has also been used in previous published work (Miyato et al., 2018). Thus, the per-sample loss for training with curvature regularization is given by:

$$\ell\left(\mathbf{z}^{(L)}(\mathbf{x}^{(0)}),\ y\right) + \gamma K(\mathbf{W},\ y,\ t) \tag{7}$$

where ℓ denotes the cross entropy loss, y is the true label of the input $\mathbf{x}^{(0)}$, t is the attack target and γ is the regularization coefficient for penalizing large curvature values. Similar to the adversarial training, in CRT, we use $\mathbf{x}^{(attack)}$ instead of $\mathbf{x}^{(0)}$ in equation (7).

7. Experiments

The *certified robust accuracy* means the fraction of correctly classified test samples whose robustness certificates (computed using CRC) are greater than a pre-specified radius ρ. Unless otherwise specified, we use the class with the second largest logit as the attack target (i.e. the class t). The notation ($L \times [1024]$, activation) denotes a neural network with L layers with the specified activation, ($\gamma = c$) denotes standard training with γ set to c, while (CRT, c) denotes CRT training with $\gamma = c$. Certificates are computed over 150 randomly chosen correctly classified images. We use a single NVIDIA GeForce RTX 2080 Ti GPU.

7.1. Fraction of inputs with tightest robustness certificate

Using the verifiable condition of Theorems 1 and 2, our approach is able to (1) find points that are provably the worst case adversarial perturbations (in the l_2 norm) in the attack problem and (2) find points on the decision boundary that are provably closest to the input in the l_2 norm in the certification problem. In particular, in Table 2, we observe that for curvature regularized networks, our approach finds provably worst-case adversarial perturbations for *all* of the inputs with a small drop in the accuracy. Moreover, for 2,3,and 4 layer networks, our method finds provably closest adversarial examples for 44.17%, 22.59% and 19.53% of inputs in the MNIST test set, respectively.

Table 2. Certificate success rate denotes the fraction of points satisfying $\mathbf{z}_y - \mathbf{z}_t = 0$, Attack success rate denotes the fraction of points ($\mathbf{x}^{(0)}$) satisfying $\|\mathbf{x}^{(attack)} - \mathbf{x}^{(0)}\|_2 = \rho = 0.5$ implying *primal=dual* in Theorems 1 and 2 respectively. We use the MNIST dataset.

Network	γ	Accuracy	Attack success	Certificate success
2×[1024], sigmoid	0.	98.77%	5.05%	2.24%
	0.03	98.30%	100%	44.17%
3×[1024], sigmoid	0.	98.52%	0.%	0.12%
	0.05	97.60%	100%	22.59%
4×[1024], sigmoid	0.	98.22%	0.%	0.01%
	0.07	95.24%	100%	19.53%

We note that the technique presented in this work is not applicable to ReLU networks due to the absence of the curvature information. Verifying the robustness property in an l_2 ball around the input is known to be an NP-complete problem for ReLU networks (Katz et al., 2017) — for an arbitrary ReLU network, it is computationally challenging to even verify (let alone find) that a given adversarial perturbation is the worst case perturbation in polynomial time unless P=NP. Using curvature-regularized neural networks with smooth activation functions, we show that it is possible to find (and not just verify) the exact worst case perturbation (and robustness certificate) for a significant fraction of test inputs. We note that in the worst case even for smooth classifiers, the problem of finding the worst adversarial perturbations remain computationally challenging. However, our theoretical and empirical results provide strong evidence that bounding the curvature of the network and using smooth activation functions can be critical to achieve high robustness guarantees for a significant fraction of samples.

7.2. Comparison with existing provable defenses

We compare against certified defense techniques proposed in Wong et al. (2018) and Zhang et al. (2019a) in Table 3 for the MNIST dataset (LeCun & Cortes, 2010) and Table 4 for the Fashion-MNIST dataset (Xiao et al., 2017) with l_2 radius $\rho = 1.58$. Even though our proposed CRT requires fully differentiable activation functions such as softplus, sigmoid, tanh etc, we include comparison with ReLU networks because the methods proposed in Wong et al. (2018); Zhang et al. (2019a) use ReLU. Since CROWN-IBP can be trained using the softplus activation function, we include it in our comparison. Similar comparison with l_2 radius $\rho = 0.5$ is given in Appendix Table 2 (MNIST dataset) and Table 3 (Fashion-MNIST dataset). We observe that CRT (certified with CRC) gives significantly higher certified robust accuracy as well as standard accuracy compared to either of the methods on both MNIST and Fashion-MNIST datasets for both different values of ρ. Since shallow fully connected networks are known to perform poorly on the CIFAR-10 dataset, we do not include those results in our comparison.

Table 3. Comparison with interval-bound propagation based adversarial training methods: COAP i.e Convex Outer Adversarial Polytope (Wong et al., 2018), CROWN-IBP (Zhang et al., 2019a)) and Curvature-based Robust Training (Ours) with attack radius $\rho = 1.58$ on MNIST. For CROWN-IBP, we vary the final_beta hyperparameter between 0.8 and 3, and use the model with best certified accuracy. Results with $\rho = 0.5$ are in Appendix Table 2.

Network	Training	Standard Accuracy	Certified Robust Accuracy
2×[1024], softplus	**CRT, 0.01**	**98.68%**	**69.79%**
	CROWN-IBP	88.48%	42.36%
2×[1024], relu	COAP	89.33%	44.29%
	CROWN-IBP	89.49%	44.96%
3×[1024], softplus	**CRT, 0.05**	**97.43%**	**57.78%**
	CROWN-IBP	86.58%	42.14%
3×[1024], relu	COAP	89.12%	44.21%
	CROWN-IBP	87.77%	44.74%
4×[1024], softplus	**CRT, 0.07**	**95.60%**	**53.19%**
	CROWN-IBP	82.74%	41.34%
4×[1024], relu	COAP	90.17%	44.66%
	CROWN-IBP	84.4%	43.83%

In Appendix Table 5, we compare CRT with Randomized Smoothing (Cohen et al., 2019). For 2 & 3 layer networks, we achieve higher robust accuracy. However, we note that since our certificate is deterministic while the smoothing-based certificate is probabilistic (although with high probability), the results are not directly comparable. As a separate result, we also prove that randomized smoothing bounds the curvature of the network (Theorem 1 in Appendix E.6). We also include comparison with empirical defense methods namely PGD and TRADES in Appendix Table 8.

7.3. Comparison with existing certificates

In Table 5, we compare CRC with CROWN-general (Zhang et al., 2018a). For 2-layer networks, CRC outperforms CROWN significantly. For deeper networks, CRC works better than CROWN when the network is trained with curvature regularization. However, with small $\gamma = 0.01$, we

Table 4. Comparison between COAP (Wong et al., 2018), CROWN-IBP (Zhang et al., 2019a)) and Curvature-based Robust Training (Ours) with attack radius $\rho = 1.58$ on Fashion-MNIST. Results with $\rho = 0.5$ for are in Appendix Table 3.

Network	Training	Standard Accuracy	Certified Robust Accuracy
2×[1024], softplus	**CRT, 0.01**	**80.31%**	**54.39%**
	CROWN-IBP	69.23%	47.19%
2×[1024], relu	COAP	74.1%	46.3%
	CROWN-IBP	70.73%	48.61%
3×[1024], softplus	**CRT, 0.05**	**78.39%**	**53.4%**
	CROWN-IBP	68.72%	46.52%
3×[1024], relu	COAP	73.9%	46.3%
	CROWN-IBP	70.79%	48.69%
4×[1024], softplus	**CRT, 0.07**	**75.61%**	**49.6%**
	CROWN-IBP	68.31%	46.21%
4×[1024], relu	COAP	73.6%	45.1%
	CROWN-IBP	70.21%	48.08%

see a significant increase in CRC but a very small drop in the test accuracy (without any adversarial training). We can see that with $\gamma = 0.01$, non-trivial certified accuracies of 83.53%, 88.33%, 89.61% can be achieved on $2, 3, 4$ layer sigmoid networks, respectively, without any adversarial training. Adversarial training using CRT further boosts certified accuracy to 95.59%, 94.99% and 93.41%, respectively. We show some results on CIFAR-10 dataset in Appendix Table 7. We again observe improvements in the robustness certificate and certified robust accuracy using CRC and CRT.

7.4. Results using local curvature bounds

From Theorems 1 and 2, we can observe that if the curvature is *locally* bounded within a convex region around the input (we call it the "safe" region), then the corresponding dual problems (d^*_{cert}, d^*_{attack}) are again convex optimization problems provided the optimization trajectory does not escape the safe region.

Theorem 3 can be directly extended to compute the local curvature bound using bounds on the second derivatives, i.e. $\sigma''(\mathbf{z}^{(1)})$ in the local region. In Table 6, we show significant improvements for the CRC certificate for two-layer sigmoid networks on the MNIST dataset for $\gamma = 0$. However, with the curvature regularization, the difference is insignificant. We also observe that the certified accuracy for (CRT, 0.0) improves from 95.04% to 95.31% and for standard improves from 54.17% to 58.06%. The certified accuracy remains the same for other cases. Implementation details are in the Appendix Section C.6.

Network	Training	Standard Accuracy	Certified Robust Accuracy
2×[1024], sigmoid	standard	98.37%	54.17%
	$\gamma = 0.01$	98.08%	83.53%
	CRT, 0.01	98.57%	**95.59%**
3×[1024], sigmoid	standard	98.37%	0.00%
	$\gamma = 0.01$	97.71%	88.33%
	CRT, 0.01	97.23%	**94.99%**
4×[1024], sigmoid	standard	98.39%	0.00%
	$\gamma = 0.01$	97.41%	89.61%
	CRT, 0.01	97.83%	**93.41%**

(a) Effect of γ on certified robust accuracy

Network	Training	Certificate (mean)	
		CROWN	CRC
2×[1024], sigmoid	standard	0.28395	**0.48500**
	$\gamma = 0.01$	0.32548	**0.84719**
	CRT, 0.01	0.43061	**1.54673**
3×[1024], sigmoid	standard	**0.24644**	0.06874
	$\gamma = 0.01$	0.39799	**1.07842**
	CRT, 0.01	0.39603	**1.24100**
4×[1024], sigmoid	standard	**0.19501**	0.00454
	$\gamma = 0.01$	0.40620	**1.05323**
	CRT, 0.01	0.40327	**1.06208**

(b) Comparison between CROWN-general (Zhang et al., 2018a) and CRC.

Table 5. Effect of curvature regularization and CRT on certified robust accuracy and robustness certificate

Computing local curvature bounds for deeper networks, however, is more challenging due to the presence of terms involving multiplication of first and second derivatives. A straightforward extension of Theorem 4 4, wherein we compute the upper bound on σ' and σ'' in a local region around the input across all neurons in all layers does not yield significant improvements over the global method, therefore we do not include those results in our comparison.

Table 6. Comparison between Certified Robust accuracy and CRC for 2 layer sigmoid and tanh networks using global and local curvature bounds on MNIST dataset with $\rho = 0.5$

Network	Training	CRC (Global)	CRC (Local)
2×[1024], sigmoid	standard	0.5013	**0.5847**
	CRT, 0.0	1.0011	**1.1741**
	CRT, 0.01	1.5705	**1.6047**
	CRT, 0.02	1.6720	**1.6831**

8. Extension to convolutional neural networks

The formula derived in Lemma 1 is valid even for convolutional neural networks. However, to derive the curvature bound (using Theorem 4), we need to compute a bound on the singular values of the Jacobian of the convolution layer (i.e. $\|\mathbf{W}^{(I)}\|$). In order to do this, one can use spectral bounds for convolution layers derived in (Sedghi et al., 2018). We present some preliminary results using this technique in (Singla & Feizi, 2019) for a single layer convolutional neural network with softplus activations.

9. Conclusion

In this paper, we develop computationally-efficient convex relaxations for robustness certification and adversarial attack problems given the classifier has a bounded curvature. We also show that this convex relaxation is tight under some general verifiable conditions. To be able to use proposed certification and attack convex optimizations, we derive global curvature bounds for deep networks with differentiable activation functions. This result is a consequence of a closed-form expression that we derive for the Hessian of a deep network. Adversarial training using our attack method coupled with curvature regularization results in a significantly higher certified robust accuracy than the existing provable defense methods. Our proposed curvature-based robustness certificate significantly outperforms the CROWN certificate when trained with our regularizer. Scaling up our proposed curvature-based robustness certification and training methods as well as further tightening the derived curvature bounds are among interesting directions for the future work.

10. Acknowledgements

This project was supported in part by NSF CAREER AWARD 1942230, HR 00111990077, HR001119S0026 and Simons Fellowship on ”Foundations of Deep Learning”.

References

Anil, C., Lucas, J., and Grosse, R. B. Sorting out lipschitz function approximation. In *ICML*, 2018.

Athalye, A. and Carlini, N. On the robustness of the cvpr 2018 white-box adversarial example defenses. *ArXiv*, abs/1804.03286, 2018.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Bunel, R., Turkaslan, I., Torr, P. H. S., Kohli, P., and Mudigonda, P. K. A unified view of piecewise linear neural network verification. In *NeurIPS*, 2017.

Cao, X. and Gong, N. Z. Mitigating evasion attacks to deep neural networks via region-based classification. *ArXiv*, abs/1709.05583, 2017.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec ’17, 2017.

Carlini, N., Katz, G., Barrett, C. E., and Dill, D. L. Provably minimally-distorted adversarial examples. 2017.

Cheng, C.-H., Nührenberg, G., and Ruess, H. Maximum resilience of artificial neural networks. In *ATVA*, 2017.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

Croce, F., Andriushchenko, M., and Hein, M. Provable robustness of relu networks via maximization of linear regions. *ArXiv*, abs/1810.07481, 2018.

Dutta, S., Jha, S., Sankaranarayanan, S., and Tiwari, A. Output range analysis for deep feedforward neural networks. In *NFM*, 2018.

Dvijotham, K., Gowal, S., Stanforth, R., Arandjelovic, R., O’Donoghue, B., Uesato, J., and Kohli, P. Training verified learners with learned verifiers. *ArXiv*, abs/1805.10265, 2018a.

Dvijotham, K., Stanforth, R., Gowal, S., Mann, T. A., and Kohli, P. A dual approach to scalable verification of deep networks. In *UAI*, 2018b.

Ehlers, R. Formal verification of piece-wise linear feed-forward neural networks. *ArXiv*, abs/1705.01320, 2017.

Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. J. Efficient and accurate estimation of lipschitz constants for deep neural networks. *CoRR*, abs/1906.04893, 2019. URL http://arxiv.org/abs/1906.04893.

Fischetti, M. and Jo, J. Deep neural networks and mixed integer linear optimization. *Constraints*, 23:296–309, 2018.

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. T. Ai2: Safety and robustness certification of neural networks with abstract interpretation. *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2018.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *ArXiv*, abs/1810.12715, 2018.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 2266–2276. 2017.

Huang, X., Kwiatkowska, M. Z., Wang, S., and Wu, M. Safety verification of deep neural networks. *ArXiv*, abs/1610.06940, 2016.

Katz, G., Barrett, C. W., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *CAV*, 2017.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. *ArXiv*, abs/1607.02533, 2016a.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. *ArXiv*, abs/1611.01236, 2016b.

Laidlaw, C. and Feizi, S. Functional adversarial attacks. In *NeurIPS*, 2019.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Lécuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. K. K. Certified robustness to adversarial examples with differential privacy. In *IEEE S&P 2019*, 2018.

Levine, A. and Feizi, S. (de)randomized smoothing for certifiable defense against patch attacks. *ArXiv*, abs/2002.10733, 2020a.

Levine, A. and Feizi, S. Robustness certificates for sparse adversarial attacks by randomized ablation. In *AAAI*, 2020b.

Levine, A. J. and Feizi, S. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. *ArXiv*, abs/1910.10783, 2020c.

Li, B. H., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive gaussian noise. 2018.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. *ArXiv*, abs/1712.00673, 2017.

Lomuscio, A. and Maganti, L. An approach to reachability analysis for feed-forward relu neural networks. *ArXiv*, abs/1706.07351, 2017.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Mirman, M., Gehr, T., and Vechev, M. T. Differentiable abstract interpretation for provably robust neural networks. In *ICML*, 2018.

Miyato, T., ichi Maeda, S., Koyama, M., and Ishii, S. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41: 1979–1993, 2017.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Moosavi Dezfooli, S. M., Fawzi, A., Uesato, J., and Frossard, P. Robustness via curvature regularization, and vice versa. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, May 2016. doi: 10.1109/SP.2016.41.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016. URL http://arxiv.org/abs/1602.02697.

Peck, J., Roels, J., Goossens, B., and Saeys, Y. Lower bounds on the robustness to adversarial perturbations. In *NIPS*, 2017.

Qin, C., Martens, J., Gowal, S., Krishnan, D., Fawzi, A., De, S., Stanforth, R., and Kohli, P. Adversarial robustness through local linearization. *arXiv preprint arXiv:1907.02610*, 2019.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. *ArXiv*, abs/1801.09344, 2018a.

Raghunathan, A., Steinhardt, J., and Liang, P. Semidefinite relaxations for certifying robustness to adversarial examples. In *NeurIPS*, 2018b.

Salman, H., Yang, G., Li, J., Zhang, P., Zhang, H., Razenshteyn, I. P., and Bubeck, S. Provably robust deep learning via adversarially trained smoothed classifiers. *ArXiv*, abs/1906.04584, 2019.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Scaman, K. and Virmaux, A. Lipschitz regularity of deep neural networks: analysis and efficient estimation, 2018.

Sedghi, H., Gupta, V., and Long, P. M. The singular values of convolutional layers. *arXiv preprint arXiv:1805.10408*, 2018.

Singh, G., Gehr, T., Mirman, M., Püschel, M., and Vechev, M. T. Fast and effective robustness certification. In *NeurIPS*, 2018.

Singla, S. and Feizi, S. Bounding singular values of convolution layers, 2019.

Singla, S., Wallace, E., Feng, S., and Feizi, S. Understanding impacts of high-order loss approximations and features in deep learning interpretation. In *ICML*, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

Uesato, J., O'Donoghue, B., Kohli, P., and van den Oord, A. Adversarial risk and the dangers of evaluating against weak attacks. In *ICML*, 2018.

Wang, S., Chen, Y., Abdou, A., and Jana, S. K. K. Mixtrain: Scalable training of verifiably robust neural networks. 2018a.

Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. K. K. Efficient formal safety analysis of neural networks. In *NeurIPS*, 2018b.

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

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. *ArXiv*, abs/1711.00851, 2017.

Wong, E., Schmidt, F. R., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *NeurIPS*, 2018.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 4939–4948. Curran Associates, Inc., 2018a.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. *ArXiv*, abs/1811.00866, 2018b.

Zhang, H., Zhang, P., and Hsieh, C.-J. Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications. In *AAAI*, 2018c.

Zhang, H., Chen, H., Xiao, C., Li, B., Boning, D. S., and Hsieh, C. Towards stable and efficient training of verifiably robust neural networks. *CoRR*, abs/1906.06316, 2019a. URL http://arxiv.org/abs/1906.06316.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019b.

Zheng, S., Song, Y., Leung, T., and Goodfellow, I. J. Improving the robustness of deep neural networks via stability training. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4480–4488, 2016.