Robust Design of Deep Neural Networks against Adversarial Attacks based on Lyapunov Theory

Arash Rahnama
Modzy
arash.rahnama@modzy.com

Andre T. Nguyen
Booz Allen Hamilton
nguyen_andre@bah.com

Edward Raff
Booz Allen Hamilton
raff_edward@bah.com

Abstract

Deep neural networks (DNNs) are vulnerable to subtle adversarial perturbations applied to the input. These adversarial perturbations, though imperceptible, can easily mislead the DNN. In this work, we take a control theoretic approach to the problem of robustness in DNNs. We treat each individual layer of the DNN as a nonlinear system and use Lyapunov theory to prove stability and robustness locally. We then proceed to prove stability and robustness globally for the entire DNN. We develop empirically tight bounds on the response of the output layer, or any hidden layer, to adversarial perturbations added to the input, or to any preceding hidden layer. We show how the spectral norm of the weight matrix for an individual layer relates to Lyapunov properties of that layer, and consequently to the local and global stability and robustness of the DNN. Our results give new insights into how spectral norm regularization can mitigate the adversarial effects. Finally, we evaluate the power of our approach on a variety of data sets and network architectures and against some of the well-known adversarial attacks.

1. Introduction

The objective of a supervised learning task for the input $u \in \mathbb{R}^d$, and its associated target value y for a given DNN denoted by $H_\theta(u)$, where θ is a set of parameters to be learned during the training, is to classify the instance u correctly such that $y = H_\theta(u)$. Recently, the research community has become interested in adversarial attacks, where the adversary's goal is to introduce a small amount of engineered perturbation $\Delta \in \mathbb{R}^d$ to u, so that $u' = u + \Delta$, while still maintaining its perceptual similarity to u, can deceive the DNN into making a mistake, i.e., $H_\theta(u + \Delta) \neq y$. The adversary is usually assumed to be constrained by an ℓ_p-norm so that $\|u' - u\|_{\ell_p} \leq \epsilon$, where ϵ bounds the adversaries' freedom to alter the input. While this does not capture the full scope of potential adversaries [4], attacks of this form have proven difficult to prevent [32, 5, 2, 31]. While optimal defenses have been developed for simple linear models [3, 24], the over-parameterized nature of DNNs and the complexity of surfaces learned during training make the development of robust solutions against the adversary difficult [14].

In this work, we use Lyapunov theory of stability and robustness of nonlinear systems to develop a new understanding of how DNNs respond to changes in their inputs, and thus, to adversarial attacks under the ℓ_2 norm framework. By treating each layer l in the DNN as a nonlinear system h_l, we develop a new framework which sets tight bounds on the response of the individual layer to adversarial perturbations (maximum changes in $\|h_l(u) - h_l(u + \Delta)\|_2^2$) based on the spectral norm of the weights, $\rho(W_l)$, of the individual layer. We characterize Lyapunov properties of an individual layer and their relationship to local and global stability and robustness of the network.

Since our analysis is based on a sequence of nonlinear transformations $h_{l=1,\ldots,n}$, our method is the first to bound the response to input alterations by the attack parameter Δ_l for any layer l in the DNN. For simple forward networks (fully connected and convolutional), our results show that an attack's success under the ℓ_2 ball is independent of the depth of the DNN and that the spectral norm of the weight matrix for the first and last layer of a network have the largest effect on robustness. Our Lyapunov analysis of Residual Blocks shows that the skip-connections of these blocks contribute to their lack of robustness against adversarial attacks [13] and that these blocks require more restrictive Lyapunov conditions (a tighter spectral norm regularization) for maintaining the same level of robustness. This may compromise the DNN's accuracy on the clean data set [34, 10]. Previous works [11, 28, 37, 8] have used a penalty on the spec-

tral norm to build networks more robust to attack. However, their analysis requires using one penalty for all layers, which over-regularizes the networks (reducing accuracy in all cases [10]), and provides only loose bounds that can not be empirically evaluated. Our approach shows how and why to set different spectral penalties for each layer, improving the accuracy of the model on clean and perturbed inputs. Simultaneously, our bound is empirically tight and has no hidden constants, giving us greater insight to the impact of attacks and how perturbations propagate through the network.

In summary, our contributions revolve around showing that adversarial ML research can leverage control theory to understand and build defenses against strong adversaries. This can accelerate the research progress in this area. We prove that with our proposed training approach, the perturbation to the final activation function ($\Delta_n \in \mathbb{R}^d$) is bounded by $\|\Delta_n\|_2 \leq \sqrt{c} \cdot \epsilon$, where c is a constant determined by the hyper-parameters chosen and ϵ models the adversarial perturbation magnitude. Our bound is empirically tight, and applies to *all* possible inputs, and to attacks applied at *any* layer of the network (input, or hidden), with no distributional assumptions. Our analysis shows how Residual Blocks can aid adversarial attacks, and extensive empirical tests show that our approach's defensive advantages increase with the adversary's freedom ϵ.

2. Related work

To provide certifiable defenses, complex optimization schemes are usually adopted to show that all the data points inside an ℓ_p ball around a sample data point have the same prediction [36, 21, 7, 9]. The bounds provided by these methods are usually loose, and the computational costs associated with them increase exponentially with the size of the network. These approaches are only applicable to parts of the input space for which feasible solutions exist. Works such as [39] have empirically shown that bounding a layer's response to the input generally improves robustness. Works such as [35, 40] focus on certifying robustness for a DNN by calculating the bounds for the activation functions' responses to the inputs. The closest to our work are the results given in [11, 28, 37, 8]. [8] utilizes Lipschitz properties of the DNN to improve robustness against adversarial attacks. Unlike [8], our approach does not require a predetermined set of hyper-parameters to prove robustness. Our analysis provides a range of possible values which determine different levels of robustness and may be selected per application. Similarly, [28] empirically explores the benefits of bounding the response of the DNN by regularizing the spectral norm of layers based on the Lipschitz properties of the DNN. These Lipschitz based approaches may be seen as one subset of our Lyapunov based approach. As we will describe, our Lyapunov based analysis is built upon a more general input-output nonlinear mapping which does not necessarily depend on restricting the networks's Lipschitz property. [37] explores the benefits of training networks with spectral regularization for improving generalizability against input perturbations. However, their work bounds the spectral norm of all layers by 1. As we will show, this approach limits the performance on the clean data set and does not produce the most robust DNN. [11] uses PAC-Bayes generalization analysis to estimate the robustness of DNNs trained by spectral regularization against adversarial attacks. The analysis given in their work however, requires the same regularization condition enforced across all layers of the DNN. Our work provides per layer conditions for robustness, which may be utilized for the selection of the best regularization parameters for each layer independently, given a data set and architecture through cross-validation. Although, we have not empirically shown in the paper, our approach theoretically should provide a level of robustness against intermediate level attacks recently introduced in [18]. Our work provides a theoretical backing for the empirical findings in [28, 36] which state that Leaky ReLu, a modified version of the ReLu function, may be more robust comparatively. Finally, only one prior work has looked at control theory for adversarial defense, but their method covered only a toy adversary constrained to perturbing the input by a constant [29].

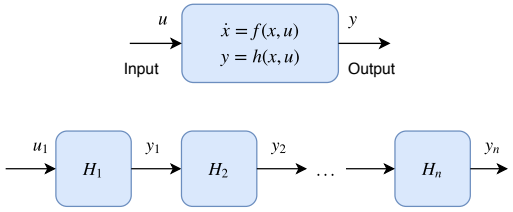

Figure 1: A nonlinear system (top). The DNN modeled as a cascade of nonlinear systems (bottom).

3. Preliminaries and Motivation

In this work, we address the machine learning problem of adversarial attacks on DNNs from a control theoretic perspective. To aid in bridging the gap between these two fields, we will briefly review the core control theory results needed to understand our work, with further exposition in Appendix A for less familiar readers. The design of stable and robust systems that maintain their desired performance in the presence of external noise and disturbance has been studied in the control theory research literature. Our work is based on the Lyapunov theory of stability and robustness of nonlinear systems which dates back to more than a century ago [22]. We treat each layer of the DNN as a nonlinear system and model the DNN as a cascade connection of nonlinear systems. A nonlinear system is defined as a system which produces an output signal for a given input signal

through a nonlinear relationship. More specifically, consider the following general definition for the nonlinear system H (Fig. 1),

$$H : \begin{cases} \dot{x} = f(x, u) \\ y = h(x, u), \end{cases}$$

where $x \in X \subseteq R^n$, $u \in U \subseteq R^m$, and $y \in Y \subseteq R^k$ are respectively the state, input and output of the system, and X, U and Y are the local state, input and output sub-spaces around the current operating points. The nonlinear mappings f and h model the relationship among the input signal u, the internal states of the system x and the output signal y. To help connect these control theory notations and definitions to our analysis of DNN models, see Table 1.

Table 1: A control theory to machine learning mapping.

	Control theoretic definition	Context for our work
u	Input of the nonlinear system	Inputs to the DNN (e.g., image) or the inputs to a hidden layer from the previous layer
y	Output of the nonlinear system	Output of the DNN or the output of any hidden layer
x	States of the nonlinear system	Weights and biases of a layer
$\dot{x}$	Transient changes of the states over discrete or continuous steps	Changes of the weights and biases during t training steps
$f(.)$	Nonlinear function modeling the transient changes of the states of the system based on the previous state and current input	Models the updates applied to the weights' and biases' of a layer over the t training steps
$h(.)$	Nonlinear function modeling the steady-state behavior of the system given the current state and input	Models the input-output relationship of a hidden layer given the current values of weights, biases and input
ρ, ν	Lyapunov parameters modeling the input-output behavior (robustness and stability properties) of the system (explained in Subsection 4.3)	Values determining the extent of spectral regularization enforced at a layer given the desired level of robustness and accuracy (Theorem 3, Corollary 2)

The behavior of a layer inside the DNN may be modeled as a nonlinear system defined above. More specifically, for the layer l, the input signal u takes the size of the layer $l-1$ and stands for the input to the layer l before it is transformed by the weights and biases. y has the size of layer l and may be defined as the output of the layer l after the activation functions. The weights and biases of the DNN are the states of the nonlinear systems. In this vein, h and f are general functions which model the relationship between the states x, and the nonlinear transformation applied to the input of the layer u to produce the output y of the layer after activation functions. The states are dynamically updated through gradient descent, given the inputs from the training data set during the training iterations t. $\dot{x}_l$ is the derivative taken over training iterations indicating that the weights and biases are changing during training, and f models this nonlinear behavior during the training iterations. Please see Appendix A for further details.

Our analysis is based on Lyapunov theory which gives us the freedom to define stability and robustness purely based on the input-output relationship of the layers without the exact knowledge of the internal state changes $\dot{x}$ (i.e., we do not need to know the specific weight and bias values). More specifically, Lyapunov theory combined with the fact that we are showing bounded-input-bounded-output (BIBO) stability and robustness of the layer, allows us to abstract out the transient behavior of the nonlinear systems during the training iterations t, i.e., $\dot{x}_l(t) = f_l(x_l(t), u_l(t))$ where $x_l(t) = \{W_l(t), B_l(t)\}$, from the robustness analysis and focus only on the steady-state behavior of the nonlinear system, i.e., the input-output mapping of the layer, $y_l = h_l(x_l, u_l) = h_l(\{W_l, B_l\}, u_l)$ where $h_l(.)$ models the nonlinear transformation. By analyzing $h_l(.)$, we can derive robust conditions to be enforced during training for the states of a layer $\{W_l, B_l\}$.

Next, we define an input-output stability and robustness Lyapunov criterion for a nonlinear system. A nonlinear system is said to be bounded-input-bounded-output (BIBO) stable, if it produces bounded outputs for bounded input signals. A nonlinear system is said to be stable and robust, if it produces bounded output signals that are close in the Euclidean space, when their respective input signals are also close enough in the Euclidean space [22]. Mathematically, these definitions can be represented as follows,

Definition 1 *[22] System H is instantaneously incrementally finite-gain (IIFG) ℓ_2-stable and robust, if for any two inputs $u_1, u_2 \in U$, there exists a positive gain γ, such that the relation,*

$$||y_2 - y_1||_{\ell_2} \leq \gamma ||u_2 - u_1||_{\ell_2}.$$

locally holds. Here, $||y_2 - y_1||_{\ell_2}$ and $||u_2 - u_1||_{\ell_2}$ represent the Frobenius ℓ_2-norm of the input signals u_1 and u_2 and their respective output signals y_1 and y_2.

If a system is IIFG stable and robust, then the changes in the output of the entire system are bounded by the changes in the

input of the entire system. As a result, if the changes in the input are minuscule, which is the assumption for the majority of ℓ_2 adversarial attacks, then the changes in their respective outputs are also minuscule. This is the exact behavior that we would like to encourage for each layer of the DNN so that the entire network is robust against adversarial attacks.

Remark 1 *It is important to note that the relationship given in Definition 1 is more general than enforcing Lipschitz continuity. In particular, the above relationship should only hold locally for the input signals u_1 and u_2 for the DNN to be IIFG. Further, the above assumption does not place any constraints on the initial conditions of the DNN. Additionally, Lipschitz continuity implies uniform continuity, but the relationship given above potentially allows for discontinuous distributions and does not enforce any continuously differentiable condition on the mapping from the input to the output of the DNN [22, 12, 19]. Finally, we will show that the enforcement of the relationship given in Definition 1 locally at the layer level is not necessary. We will show that by enforcing a much looser condition at each layer, we can encourage the behavior given in Definition 1 globally for the DNN.*

Lyapunov theory of dissipativity provides a fundamental framework for the stability and robustness analysis of systems based on a generalized notion of the "energy" supplied and dissipated in the system [22]. This generalized notion of "energy" is defined based on a relationship that solely relies on the input and output of the system. This theory states that a passive system which dissipates energy is robust and stable. The benefit of this approach is that by only enforcing a relationship between the input and output of the DNN, we can define a measure of stability and robustness for the entire DNN against adversarial attacks and characterize robust conditions for the states of the DNN (weights and biases). This is done by first defining an input-output mapping which characterizes a specific relationship between the input and output of the layers inside the DNN and then enforcing that the relationship should hold for the DNN for all the inputs supplied to the model and the outputs produced by the model during training so that the same will hold during inference. The following definition provides the mathematical representation of the aforementioned concept,

Definition 2 *[38] System H is considered to be instantaneously Incrementally Input Feed-Forward Output Feedback Passive (IIFOFP), if it is dissipative with respect to the input-output mapping $\omega(\cdot,\cdot)$,*

$$\omega(u_2-u_1, y_2-y_1) = (u_2-u_1)^T(y_2-y_1) - \delta(y_2-y_1)^T(y_2-y_1) - \nu(u_2-u_1)^T(u_2-u_1),$$

for some value $\nu \in \mathbb{R}$ and $\delta \in \mathbb{R}$ where $\nu \cdot \delta \leq 0.25$.

Remark 2 *Although ν and δ may take negative or positive values, our goal is to design layers that have positive δ's. Further, a positive set of δ and ν for a layer implies IIFG stability and robustness for that layer. Nevertheless, as we will show, for the entire network to be robust and stable, the ν and δ for an individual layer can take any value as long as certain conditions are met. ν and δ define a range of possible stability and robustness properties for a system. The lack of stability and robustness in one layer may be compensated for by an excess of robustness and stability in another layer [22]. In Subsection 4.3, we provide an interpretation of the above relations and outline the implications behind the selection of ν and δ.*

Theorem 1 *[22] If the nonlinear system H is IIFOFP with $\delta > 0$, then it is IIFG stable and robust with the finite gain $\gamma = \frac{1}{\delta}$.*

Our goal in this paper is to connect Definition 2 to Definition 1 to achieve robustness and the property given in Theorem 1. By enforcing the looser condition given in Definition 2, where ν and δ can take a range of values locally at the layer level, we encourage a robust global behavior for the entire DNN as given in Definition 1. According to Lyapunov theory for a layer to have the instantaneously IIFOFP property in Definition 2, the following condition should hold for the input-output mapping of all the inputs u_1, u_2 fed to the layer and their respective output signals y_1, y_2: $\omega(u_2-u_1, y_2-y_1) > 0$ [22]. If this holds, then the layer is dissipative with respect to the nonlinear relation given in Definition 2. Our results will show how we can reach Definition 1 and robustness globally by enforcing Definition 2 locally. Lastly, we will use the following matrix properties in our proofs,

Definition 3 *[20] A square matrix A is a quasi-dominant matrix (diagonally dominant), if there exists a positive diagonal matrix $P = diag\{p_1, p_2, ..., p_n\}$ such that $a_{ii}p_i \geq \sum_{j\neq i} |a_{ij}|p_j,\ \forall i$, and/or $a_{jj}p_j \geq \sum_{i\neq j} |a_{ji}|p_i,\ \forall j$. If these inequalities are met strictly, then the matrix is said to be strictly row-sum (or column-sum) quasi-dominant. If P can be chosen as the identity matrix, then the matrix is said to be row- or column- diagonally dominant.*

Corollary 1 *[33] Every symmetric quasi-dominant matrix is positive definite.*

4. Theoretical Analysis and Main Results

In our analysis, we treat each layer of the DNN as a nonlinear system as defined in Section 3 i.e., H_i for all layers $i = 1, ..., n$ (Fig. 1). A nonlinear system is defined as a layer in the network that accepts an input vector from the previous layer and produces an output vector with the size of the current layer after the weights, biases and activation functions are applied to the input signal. We prove the conditions under

which each layer H_i is instantaneously IIFOFP with specific hyper-parameters δ_i and ν_i. One can interpret the values of δ_i and ν_i as measures of robustness for the specific layer i. Our results place specific constraints on the weight matrices at the given layer based on the values of the hyper-parameters δ_i and ν_i. We train the model and enforce these conditions during back-propagation. Consequently, we can show that the entire DNN is instantaneously IIFOFP and IIFG stable and robust. This means that the DNN maintains a level of robustness against adversarial changes added to the input for up to a specific adversarial ℓ_2 norm ϵ. This is because the changes in the output of the DNN are now bounded by the changes introduced to the input by the adversary. Robustness in this sense means that the adversary now needs to add a larger amount of noise to the input of the DNN in order to cause larger changes in the output of the DNN and affect the decision making process. Our approach improves robustness against adversarial changes added to the input signal and the output of a hidden layer before it is fed into the next layer.

In our approach, we consider Leaky ReLU activation functions, even though our results hold for ReLU as well. Our results and the Lyapunov theory suggest that Leaky ReLU is a more robust activation function than ReLu. We expand on this in Subsection 4.3. Δ is a measure for intervention. Δ models the extent of adversarial noise introduced to the input by the adversary. The effects of the majority of ϵ-based attacks of different norms such as the fast gradient method (FGM) and projected gradient descent (PGD) method may be modeled by Δ [23, 25, 26]. Given a DNN, we are interested in the (local) robustness of an arbitrary natural example u by ensuring that all of its neighborhood has the same inference outcome. The neighborhood of u may be characterized by an ℓ_2 ball centered at u. We can define an adversarial input as follows,

Definition 4 *Consider the input u of the layer of size n, i.e., $u \in R^n$ and the perturbed input signal $u + \Delta$ where $\Delta \in R^n$ is the attack vector that can take any value. The perturbed input vector $u+\Delta$ is within a Δ^0-bounded ℓ_2-ball centered at u if $u + \Delta \in B_2(u, \Delta^0)$, where $B_2(u, \Delta^0) := \{u + \Delta |\ ||u + \Delta - u||_2 = ||\Delta||_2 \leq \Delta^0\}$.*

Geometrically speaking, the minimum distance of a misclassified nearby instance to u is the smallest adversarial strength needed to alter the DNN's prediction, which is also the largest possible robustness measure for u. We will use the conic behavior of the activation function, spectral norm of the weights and their relation to Lyapunov theory to train DNNs that are BIBO stable and robust against the adversary.

4.1. Robustness analysis of each layer in the DNN

Each layer of a DNN can be modeled as $y_l = h_l(W_l u_l + b_l)$ for $l = 1, ..., n$ for some $n > 2$, where $u_l \in R^{n'_{l-1}}$ is the input of the l-th layer, and $W_l \in R^{n'_l \times n'_{l-1}}$ and $b_l \in R^{n'_l}$ are respectively the layer-wise weight matrix and bias vector applied to the flow of information from the layer $l-1$ to the layer l. $h_l : R^{n'_{l-1}} \rightarrow R^{n'_l}$ models the entire numerical transformation at the l-th layer including the (nonlinear) activation functions. n'_{l-1} and n'_l represent the number of neurons in layers $l-1$ and l. For a set of weight and bias parameters, $\{W_l, b_l\}_{l=1}^n$, we can model the behavior of the entire DNN as $H_{\{W_l,b_l\}_{l=1}^n}(u_1) = y_n$ where $H_{\{W_l,b_l\}_{l=1}^n} : R^{n'_1} \rightarrow R^{n'_n}$ and u_1 is the initial input to the DNN. Given the training data set of size K, $(u_i, y_i)_{i=1}^K$, where $u_i \in R^{n'_1}$ and $y_i \in R^{n'_n}$, the loss function is defined as $\frac{1}{K} L(H_{\{W_l,b_l\}_{l=1}^n}(u_i), y_i)$, where L is usually selected to be cross-entropy or the squared ℓ_2-distance for classification and regression tasks, respectively. The model parameters to be learned is x. We consider the problem of obtaining a model that is insensitive to the perturbation of the input. The goal is to obtain parameters, $\{W_l, b_l\}_{l=1}^n$, such that the ℓ_2-norm of $h(u + \Delta) - h(u)$ is small, where $u \in R^{n'_1}$ is an arbitrary vector and $\Delta \in R^{n'_1}$ is an engineered perturbation vector with a small ℓ_2-norm added by the adversary. To be more general and further investigate the properties of the layers, we assume that each activation function, modeled by h_l is a modified version of element-wise ReLU called the Leaky ReLU: $h_l(y_l) = \max(y_l, ay_l)$, where $0 < a < 1$. It follows that, to bound the variations in the output of the DNN by the variations in the input, it suffices to bound these variations for each $l \in \{1, ..., n\}$. Here, we consider that the attack variations Δ are added by the adversary into the initial input or the input of the hidden layers. This motivates us to consider a new form of regularization scheme, which is based on an individual layer's Lyapunov property. The first question we seek to answer is the following: what are the conditions under which a layer l is IIFOFP with a positive δ_l and a ν_l that may take any value? In practice, it is best to train layers so that both ν_l and δ_l are positive, as this means a tighter bound and a more robust layer (Subsection 4.3), however, this is not a necessary condition for our results. The following theorem relates the spectral norm of the weight matrix to IIFOFP and IIFG stability and robustness of a layer to answer the above question.

Theorem 2 *The numerical transformation at the hidden layer l of the DNN as defined in Subsection 4.1 is instantaneously IIFOFP and consequently IIFG stable and robust, if the spectral norm of the weight matrix for the layer satisfies the following condition,*

$$\rho(W_l) \leq \frac{1}{\delta_l^2} + \frac{2|\nu_l|}{\delta_l}$$

where $\rho(W_l)$ is the spectral norm of the weight matrix at the layer l, and the hyper-parameters $\delta_l > 0$ and ν_l meet the condition $\delta_l \cdot \nu_l \leq 0.25$.

Proof in Appendix B

Remark 3 *It is important to note that the above theorem shows a relationship between the spectral norm of the weight matrix at the layer l and instantaneously IIFG stability and robustness of the layer as defined in the Definition 1 and Theorem 1 through the hyper-parameters δ_l and ν_l. Namely, a larger value for ν_l leads to a larger upper-bound for the spectral norm of the weight matrix at layer l. Larger values of δ_l however, have a reciprocal relation to the spectral norm of the weight matrix. The relationship between parameters δ_l, ν_l and the spectral norm of the weight matrix may be utilized during the robust training of DNNs through the spectral regularization enforced at each layer.*

We can implement the above condition for each layer during the training of the network. Theorem 2 provides a wide range of possible regularization designs for the DNN and commonly used hyper-parameter selection approaches may be used to select the spectral regularization parameters per layer. If the above condition is met for each layer, then we can posit that the DNN is stable and robust in Lyapunov sense. The exact measure of stability and robustness depends on the selection of δ_l and ν_l. The global effects of this choice are outlined in Subsection 4.2. The extension of Theorem 2 to convolutional layers follows in a similar pattern and is given in Appendix C. Appendix D includes the robustness analysis of ResNet building blocks. It is important to note that ν_l and δ_l are design hyper-parameters that are selected before the training starts. The only real conditions placed on the hyper-parameters are that δ_l should be positive and $\delta_l \times \nu_l \leq 0.25$. The exact implications of choosing the hyper-parameters and their effects on the robustness of the layer against adversarial noise are detailed in Subsection 4.3.

4.2. Robustness analysis of the entire DNN

Next, we connect our results from Theorem 2 to the robustness of the entire DNN, i.e., we show how the selection of individual (δ_i, ν_i) for layers $i = 1, ..., n$ affect the robustness of the entire DNN represented by the global set (δ, ν).

Theorem 3 *Consider the cascade interconnection of hidden layers inside the DNN as given in Fig. 1 where $n > 2$, and each layer H_l for $l = 1, \ldots, n$ is instantaneously IIFOFP with their respective ν_l and δ_l as given in Theorem 2, i.e., for any two incremental inputs u_{l1}, u_{l2} for the layer l we have,*

$$\omega(u_{l2} - u_{l1}, y_{l2} - y_{l1}) = (u_{l2} - u_{l1})^T (y_{l2} - y_{l1}) - \delta_l (y_{l2} - y_{l1})^T (y_{l2} - y_{l1}) - \nu_l (u_{l2} - u_{l1})^T (u_{l2} - u_{l1}),$$

as the nonlinear input-output mapping of the layer where $\omega(u_{l2} - u_{l1}, y_{l2} - y_{l1}) > 0$. Then the entire DNN is also instantaneously IIFOFP with the hyper-parameters ν, δ and input-output mapping,

$$\omega(u_2 - u_1, y_2 - y_1) = (u_2 - u_1)^T (y_2 - y_1) - \delta (y_2 - y_1)^T (y_2 - y_1) - \nu (u_2 - u_1)^T (u_2 - u_1),$$

where u_1 and u_2 are the initial inputs to the DNN and y_1 and y_2 are their respective output signals, if the matrix $-A$ is quasi-dominant, where A is defined as,

$$A = \begin{bmatrix} \nu - \nu_1 & \frac{1}{2} & 0 & \cdots & -\frac{1}{2} \\ \frac{1}{2} & -\delta_1 - \nu_2 & \frac{1}{2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{2} & -\delta_{n-1} - \nu_n & \frac{1}{2} \\ -\frac{1}{2} & 0 & \cdots & \frac{1}{2} & \delta - \delta_n \end{bmatrix}.$$

Proof in Appendix E

Remark 4 *For the DNN to be IIFOFP with $\delta > 0$ and $\nu > 0$ and consequently stable and robust, we need to set the hyper-parameters for training such that $\delta_l > 0$ for $l = 1, ..., n$, $\delta_n > \delta > 0$, $\nu_1 > 0$, and $\nu_1 > \nu > 0$. The rest of ν_l's are selected such that the matrix $-A$ is quasi-dominant. Note that theoretically, the δ_l's or ν_l's for some hidden layers may take negative values, as long as the matrix $-A$ stays quasi-dominant, i.e., $\delta_l + \nu_{l+1} > 1$ for $l = 1, ..., n-1$ and $\delta_{n-1} + \nu_n > 1$. By selecting the hyper-parameters according to Theorem 3, one is indirectly setting the spectral regularization rule for each layer. Appendix F details an example on the selection of these hyper-parameters.*

Theorem 3 points to an interesting fact that the first and last hidden layer may have the largest effect on the robustness of the DNN. To keep the matrix $-A$ quasi-dominant, δ and ν have a direct dependence on the values of δ_n and ν_1. Next, we can characterize a relationship between the incremental changes in the input signals of a DNN, i.e., Δ_1, and their effects on the output of the DNN, i.e., Δ_n.

Corollary 2 *Consider the cascade interconnection of hidden layers inside the DNN as given in Fig. 1 where $n > 2$, if each layer H_l is instantaneously IIFOFP with their respective ν_i and δ_i, and the DNN is trained to meet the conditions given in Theorem 3, then the entire DNN is also instantaneously IIFOFP with its respective ν and δ and the input-output mapping $\omega(u_2 - u_1, y_2 - y_1) = (u_2 - u_1)^T (y_2 - y_1) - \delta (y_2 - y_1)^T (y_2 - y_1) - \nu (u_2 - u_1)^T (u_2 - u_1)$ where $\delta > 0$. One can show that the variations in the final output of the entire DNN (Δ_n) are upper-bounded (limited) by the variations in the input signal (Δ_1) through the following relation,*

$$\|\Delta_n\|_2^2 \leq \left(\frac{1}{\delta^2} + \frac{2\nu}{\delta} \right) \|\Delta_1\|_2^2 = \left(\frac{1}{\delta^2} + \frac{2\nu}{\delta} \right) \epsilon^2$$

where the design parameter δ and ν are both positive.

Proof in Appendix G

4.3. Conic interpretation of the proposed approach

[38] was the first work that connected Lyapunov notion of stability and robustness to the conicity (boundedness) behavior of the input-output mapping of a nonlinear

system. According to [38], a nonlinear numerical transformation is stable, if it produces bounded outputs for bounded inputs. The same nonlinear transformation is also robust, if it produces outputs that are insensitive to small changes added to the input. A stable and robust nonlinear numerical transformation then exhibits a conic behavior.

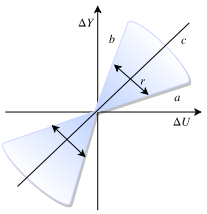

Figure 2: A depiction of the interior conic behavior of a nonlinear system.

According to [38], a nonlinear transformation exhibits a conic behavior, if the mapping between the changes in the input and the respective changes in the output, given the nonlinear transformation, always fits inside a conic sector on the input-output plane i.e., a conic nonlinear numerical transformation in the Hilbert space is one whose input changes Δu and output changes Δy are restricted to some conic sector of the $\Delta U \times \Delta Y$ inner product space as given in Fig. 2. This conic behavior is usually defined by the center line of the cone c and the radius r:

Definition 5 *[38] A relation H is interior conic, if there are real constants $r \geq 0$ and c for which $||\Delta y - c\Delta u||_{\ell_2} \leq r||\Delta u||_{\ell_2}$ is satisfied.*

This is the exact behavior that we are encouraging for each layer of the DNN so that the outputs of each layer become insensitive to small changes in the input. In particular, we have $\Delta u = \Delta_1$, $\Delta y = \Delta_n$, $c = \frac{a+b}{2}$ and $r = \frac{b-a}{2}$ where a and b are the slopes of the lower and upper bounds of the cone, and c and r are the center and radius of the cone. One can show that,

$$
\begin{aligned}
&(\Delta_n - c\Delta_1)^T(\Delta_n - c\Delta_1) \leq r^2\Delta_1^T\Delta_1 \\
&\rightarrow (\Delta_n - (\frac{a+b}{2})\Delta_1)^T(\Delta_n - (\frac{a+b}{2})\Delta_1) \\
&\leq (\frac{b-a}{2})^2\Delta_1^T\Delta_1 \rightarrow 0 \leq \Delta_1^T\Delta_n \\
&- (\frac{1}{a+b})\Delta_n^T\Delta_n - (\frac{ba}{a+b})\Delta_1^T\Delta_1.
\end{aligned}
$$

Hence by selecting $\delta = \frac{1}{a+b}$ and $\nu = \frac{ba}{b+a}$, we are bounding the output changes by the changes in the input as depicted in Fig. 2 and by that, we make the numerical transformations occurring at each layer of the DNN insensitive to small changes in the input. Particularly, a positive δ implies $b > 0$ with a larger δ implying a smaller positive b and a larger distance between the slope of the upper-bound of the cone and the ΔY axis. This implies that ΔY increases in a slower rate with increases in ΔU. A positive ν implies $a > 0$ with a larger ν implying a larger a and a larger distance between the lower-bound of the cone and the ΔU axis [22]. We are encouraging the pair (Δ_1, Δ_n) to be instantaneously confined to a sector of the plane as depicted. The conic interpretation described here combined with the results given in the previous sections support the findings presented in [28, 36] that the use of Leaky Relu activation in the architecture of DNNs may contribute to robustness.

5. Experiments

We validate our results by performing a diverse set of experiments on a variety of architectures (fully-connected, AlexNet, ResNet) and data sets (MNIST, CIFAR10, SVHN and ImageNet). Appendix L contains the details on hyper-parameters and training process for the above architecture and data set combinations. The experiments are implemented in TensorFlow [1] and the code will be made readily available. We test the DNNs against the fast gradient method (FGM) attack [15] with Frobenius ℓ_2 norm of $\epsilon \in [0.1, 0.4]$, and iterative projected gradient descent (PGD) attack [25] with 100 iterations, $\alpha = 0.02\epsilon$ and the same range of epsilons. Further, we show in Appendix J that our approach provides improved robustness against the Carlini & Wagner (C&W) attack [6].

Our robust Lyapunov training method regularizes the spectral norm of a layer l so that, $\rho(W_l) \leq \beta_l$ where $\beta_l = \frac{1}{\delta_l^2} + \frac{2|\nu_l|}{\delta_l}$. Given n layers, one can pick n different combinations of (δ_l, ν_l) for a given data set and architecture as long as the conditions given in Theorem 2, Theorem 3 and Remark 4 are met. With these defined, we also have a global δ and ν. These values are implied by setting $\delta_{1,\ldots,n}$ and $\nu_{1,\ldots,n}$, where any $\delta < \delta_n$ and $\nu < \nu_1$ is valid, so long as $\delta \cdot \nu \leq 1/4$ holds true. As such, we set (δ_1, ν_1), (δ_n, ν_n) tighter to make the network robust, and choose a single set of looser (δ_m, ν_m) for all intermediate layers, to maximize the model's expressive capability, which our theory guarantees is safe and gives the attacker no additional advantage. Appendix F outlines the process for selecting the Lyapunov hyper-parameters according to our proofs, or one could select them by simple cross validation. The different sets of Lyapunov design parameters used in our experiments are detailed in Appendix H. We represent a robust DNN with its global Lyapunov parameters (δ, ν). It is important to note that the greater flexibility (higher expressiveness [10]) allowed for the intermediate layers leads to a better generalization on the clean and adversarial data sets. This is not true for DNNs trained by weight-decay or spectral norm regularization against a single threshold β. All the previous works on this subject [11, 28, 37, 8], keep β constant across layers. These harder constraints over-regularize and thus impair the DNN's ability against attacks. Our results outlined in Appendix K show that our Lyapunov DNNs are more robust and perform better in comparison to the aforementioned works.

Table 2: The incremental output variations (Δ_n) for the 3 layer forward-net given an attack strength and the bounds calculated according to Corollary 2 (MNIST, PGD attack)

	Mean output change $\|\Delta_n\|_2 \leq$ Lyapunov Bound		
Lyapunov Parameters	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$
$\delta = 0.89, \nu = 0.28$	$0.244 \leq 0.435$	$0.490 \leq 0.615$	$0.736 \leq 0.753$
$\delta = 0.95, \nu = 0.26$	$0.202 \leq 0.407$	$0.406 \leq 0.576$	$0.610 \leq 0.706$
$\delta = 1.1, \nu = 0.22$	$0.170 \leq 0.352$	$0.340 \leq 0.497$	$0.511 \leq 0.609$
Baseline	1.233	2.446	3.641

Table 2 details the effectiveness of our approach in bounding the incremental changes in the output of the DNN caused by the attack. Given an adversarial perturbation of size $\|\Delta_1\|_2^2$, our Corollary 1 states the change to the final logits ($\|\Delta_n\|_2^2$) should be bounded by $\|\Delta_n\|_2^2 \leq \sqrt{\left(\frac{1}{\delta^2} + \frac{2\nu}{\delta}\right)}\|\Delta_1\|_2^2$. Baseline represents a DNN with no regularization enforced during training. The results show that our bounds are never violated, however as ϵ gets larger, the output changes get closer to our proposed bounds. Given the empirical tightness of the bounds, one may be able to a-priori determine the worst case vulnerabilities against an attack for a network, i.e., connect the adversarial output changes to decision boundaries between classes.

When we consider the accuracy under attack, our Lyapunov approach dominates prior works. [11] for instance, obtained an accuracy of 62% at $\epsilon = 0.1$ with adversarial training on CIFAR10 under the ℓ_2 PGD adversary with a lower number of iterations. Our approach obtains an accuracy of $\geq 73\%$ at $\epsilon = 0.1$ and still dominates with an accuracy of $\geq 63\%$ at $\epsilon = 0.4$ (Table 11 in Appendix I). Fig. 3 reports the CIFAR10 test accuracy under the iterative PGD and FGM attacks for different values of ϵ. As noted in Section 4, DNNs trained with larger global (δ, ν) maintain their robustness in a more consistent way for larger ϵ's. This is because enforcing a larger global (δ, ν) leads to DNNs with a more restricted conic behavior able to more effectively bound the negative effects of the adversarial noise. ℓ_2 weight-decay training seems ineffective against the attacks and our testing shows that adversarial training does not improve the performance of models trained with weight-decay regularization (Fig. 4 in Appendix I).

Lastly, to show that our approach scales to larger architectures and data sets, Table 3 represents our results for Lyapunov-based robust ResNet50 architectures trained on the ImageNet data set. A comprehensive set of results for a larger set of experiments is given in Appendix I. This appendix also includes our mathematical proofs on how Lyapunov based spectral regularization of the weights can improve the robustness of residual blocks. The bottom five rows of Table 3 show the effectiveness of our Lyapunov based regularization in improving the robustness of the ResNet model

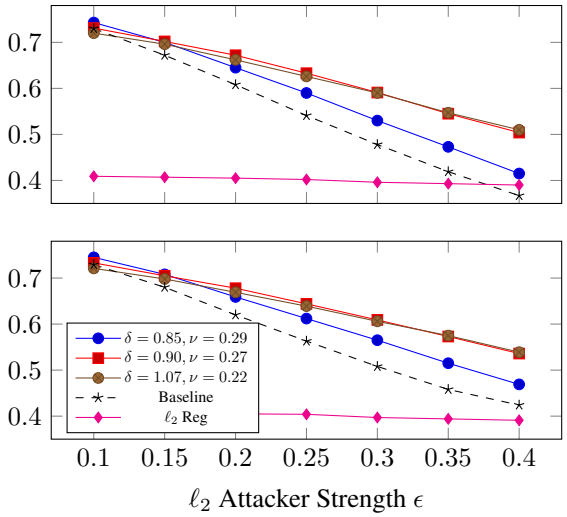

Figure 3: Accuracy of the DNN under PGD attack ($k = 100$ iterations) (top) and FGM attack (bottom) using AlexNet on CIFAR10. Plots share the same legend and axis.

trained on ImageNet against FGM based attacks across a spectrum of ϵ values. The first two rows further show that our approach remains compatible with adversarial training, gaining further improvements in accuracy while under attack.

Table 3: Experiment results for ResNet50 trained on the ImageNet dataset under the FGM attack.

Network Type	Top 1 Accuracy on the Adversarial Test Dataset (Attack Strength ϵ)			
	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$
$\delta = 1.0, \nu = 0.24$ with ℓ_2 FGM Adv. Training	**0.69**	**0.68**	**0.68**	**0.68**
$\delta = 0.89, \nu = 0.26$ with ℓ_2 FGM Adv. Training	0.67	0.65	0.63	0.61
$\delta = 0.95, \nu = 0.26$	**0.61**	**0.51**	**0.44**	**0.38**
$\delta = 0.86, \nu = 0.29$	0.60	**0.51**	0.43	0.37
$\delta = 0.74, \nu = 0.33$	0.60	0.50	0.42	0.36
Baseline Training	0.57	0.45	0.40	0.34
Baseline Training with Freb. ℓ_2 reg., $\lambda = 0.01$	0.58	0.46	0.39	0.32

6. Conclusion

In this paper, we analyzed the robustness of forward, convolutional, and residual layers against adversarial attacks based on Lyapunov theory of stability and robustness. We proposed a new robust way of training which improves robustness and allows for independent selection of the regularization parameters per layer. Our work bounds the layers' response to the adversary and gives insights into how different architectures, activation functions, and network designs behave against attacks.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning (ICML)*, 2018.

[3] Battista Biggio, Giorgio Fumera, and Fabio Roli. Multiple classifier systems for robust classifier design in adversarial environments. *International Journal of Machine Learning and Cybernetics*, 1(1):27–41, 12 2010.

[4] Tom B. Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian J. Goodfellow. Unrestricted Adversarial Examples. *arXiv preprint*, 2018.

[5] Nicholas Carlini and David Wagner. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pages 3–14, New York, NY, USA, 2017. ACM.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[7] Chih-Hong Cheng, Georg Nührenberg, and Harald Ruess. Maximum resilience of artificial neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 251–268. Springer, 2017.

[8] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 854–863. JMLR. org, 2017.

[9] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[10] Nicolas Couellan. The coupling effect of lipschitz regularization in deep neural networks. *arXiv preprint arXiv:1904.06253*, 2019.

[11] Farzan Farnia, Jesse M Zhang, and David Tse. Generalizable adversarial training via spectral normalization. *arXiv preprint arXiv:1811.07457*, 2018.

[12] Alhussein Fawzi, Hamza Fawzi, and Omar Fawzi. Adversarial vulnerability for any classifier. *arXiv preprint arXiv:1802.08686*, 2018.

[13] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

[14] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial Spheres. In *ICLR Workshop*, 2018.

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[16] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Qian Huang, Zeqi Gu, Isay Katsman, Horace He, Pian Pawakapan, Zhiqiu Lin, Serge Belongie, and Ser-Nam Lim. Intermediate level adversarial attack for enhanced transferability. *arXiv preprint arXiv:1811.08458*, 2018.

[19] Ming Jin and Javad Lavaei. Stability-certified reinforcement learning: A control-theoretic perspective. *arXiv preprint arXiv:1810.11505*, 2018.

[20] Eugenius Kaszkurewicz and Amit Bhaya. *Matrix diagonal stability in systems and computation*. Springer Science & Business Media, 2012.

[21] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[22] Hassan K. Khalil. *Nonlinear Systems*, volume 2. Prentice Hall New Jersey, 1996.

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[24] Chang Liu, Bo Li, Yevgeniy Vorobeychik, and Alina Oprea. Robust Linear Regression Against Training Data Poisoning. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pages 91–102, New York, NY, USA, 2017. ACM.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[26] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, and Shin Ishii. Distributional smoothing with virtual adversarial training. *arXiv preprint arXiv:1507.00677*, 2015.

[27] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

[28] Haifeng Qian and Mark N Wegman. L2-nonexpansive neural networks. *arXiv preprint arXiv:1802.07896*, 2018.

[29] Arash Rahnama, Andre T. Nguyen, and Edward Raff. Connecting Lyapunov Control Theory to Adversarial Attacks. In *Proceedings of AdvML'19: Workshop on Adversarial Learning Methods for Machine Learning and Data Mining at KDD*, 2019.

[30] Hanie Sedghi, Vineet Gupta, and Philip M Long. The singular values of convolutional layers. *arXiv preprint arXiv:1805.10408*, 2018.

[31] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. One pixel attack for fooling deep neural networks. *arXiv*, 2017.

[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[33] Olga Taussky. A recurring theorem on determinants. *The American Mathematical Monthly*, 56(10P1):672–676, 1949.

[34] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *stat*, 1050:11, 2018.

[35] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks. *arXiv preprint arXiv:1804.09699*, 2018.

[36] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018.

[37] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[38] George Zames. On the input-output stability of time-varying nonlinear feedback systems part one: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE transactions on automatic control*, 11(2):228–238, 1966.

[39] Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 39–49. ACM, 2017.

[40] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, pages 4939–4948, 2018.