# L-Lipschitz Gershgorin ResNet Network

Marius F. R. Juston<sup>1</sup>, William R. Norris<sup>2</sup>, Dustin Nottage<sup>3</sup>, Ahmet Soylemezoglu<sup>3</sup>

Abstract—Deep residual networks (ResNets) have demonstrated outstanding success in computer vision tasks, attributed to their ability to maintain gradient flow through deep architectures. Simultaneously, controlling the Lipschitz bound in neural networks has emerged as an essential area of research for enhancing adversarial robustness and network certifiability. This paper uses a rigorous approach to design  $\mathcal{L}$ -Lipschitz deep residual networks using a Linear Matrix Inequality (LMI) framework. The ResNet architecture was reformulated as a pseudo-tri-diagonal LMI with off-diagonal elements and derived closed-form constraints on network parameters to ensure  $\mathcal{L}$ -Lipschitz continuity. To address the lack of explicit eigenvalue computations for such matrix structures, the Gershgorin circle theorem was employed to approximate eigenvalue locations, guaranteeing the LMI's negative semi-definiteness. Our contributions include a provable parameterization methodology for constructing Lipschitz-constrained networks and a compositional framework for managing recursive systems within hierarchical architectures. These findings enable robust network designs applicable to adversarial robustness, certified training, and control systems. However, a limitation was identified in the Gershgorin-based approximations, which over-constrain the system, suppressing non-linear dynamics and diminishing the network's expressive capacity.

Index Terms—Linear Matrix Inequalities, Lipschitz continuity, Deep residual networks, Adversarial robustness, Gershgorin circle theorem, semi-definite programming

# I. INTRODUCTION

THE robustness of deep neural networks (DNNs) is a critical challenge, mainly when applied in safety-sensitive domains where small adversarial perturbations can lead to dangerous situations such as the misclassification of important objects. One approach to address this issue is by enforcing Lipschitz constraints on the network architectures. These constraints guarantee that small changes in the input will not cause significant changes in the output. This property is vital for certifying robustness against adversarial attacks, which involve introducing slight noise to modify the expected classification output result [1], [2]. The Lipschitz constant is a key measure to bound the network's sensitivity to input perturbations. Specifically, a  $\mathcal{L}$ -Lipschitz network can be theoretically guaranteed that the output remains stable within a defined "stability

Marius F. R. Juston<sup>1</sup> is with The Grainger College of Engineering, Industrial and Enterprise Systems Engineering Department, University of Illinois Urbana-Champaign, Urbana, IL 61801-3080 USA (email: mjuston2@illinois.edu).

William R Norris<sup>2</sup> is with The Grainger College of Engineering, Industrial and Enterprise Systems Engineering Department, University of Illinois Urbana-Champaign, Urbana, IL 61801-3080 USA (email: wrnorris@illinois.edu).

Construction Engineering Research Laboratory<sup>3</sup>, U.S. Army Corps of Engineers Engineering Research and Development Center, IL, 61822, USA

This research was supported by the U.S. Army Corps of Engineers Engineering Research and Development Center, Construction Engineering Research Laboratory.

sphere" around each input, making it resistant to adversarial attacks up to a certain magnitude [3].

To achieve this, several methods have been proposed to enforce Lipschitz constraints on neural networks, including spectral normalization [4], [5], orthogonal parameterization [6], and more recent approaches such as Convex Potential Layers (CPL) and Almost-Orthogonal Layers (AOL) [6], [7]. The previous works have been shown to be formulated under a unifying semi-definitive programming architecture, which possesses the constraints on the networks as LMIs [8]. However, ensuring Lipschitz constraints in deep architectures, particularly residual networks (ResNets), presents unique challenges due to their recursive structure. While prior work has made strides in constraining individual layers [8], [9] and generating a unifying semi-definite programming approach, the generalized deep residual network formulation presents issues in the pseudo-tri-diagonal structure of its imposed LMI.

Furthermore, multi-layered general Feedforward Neural Networks (FNN) have been shown to generate block tridiagonal matrix LMi formulations [10] due to their inherent network structure, which in contrast to the residual formulation, yield explicit solutions [11], [12]. However, due to the off-diagonal structure of the network, the direct application of the exact eigenvalue computation is not feasible, making the solution process significantly more complex.

Previous work has also demonstrated an iterative approach by utilizing projected gradient descent optimization or a regularization term on the estimated Lipschitz constant to ensure a constraint on the Lipschitz constraint [13]–[15]. While this guarantees an iterative enforcement of the Lipschitz constraint, it does not ensure a theoretical Lipschitz guarantee across the entire network until this convergence. However, the advantage of this technique is its generalizability, which allows for the utilization of more general network structures.

#### A. Contributions

This paper introduces the formulation of deep residual networks as Linear Matrix Inequalities (LMI). It derives closed-form constraints on network parameters to ensure theoretical  $\mathcal{L}$ -Lipschitz constraints. The LMI was structured as a tridiagonal matrix with off-diagonal components, which inherently complicates the derivation of closed-form eigenvalue computations. To address this limitation, the Gershgorin circle theorem was employed to approximate the eigenvalue locations. The Gershgorin circles enabled the derivation of closed-form constraints that guaranteed the negative semi-definiteness of the LMI.

Additionally, this paper demonstrates a significant limitation of the Gershgorin circle theorem in this context: the derived approximations lead to over-constraining the system, effectively suppressing the network's non-linear components. This, in turn, makes the network act as a simple linear transformation instead.

Moreover, while [8]'s work generates a closed-form solution for a residual network, it is limited to considering a single inner layer. In contrast, this paper presents a more general formulation that accommodates a more expressive inner layers system within the residual network system, offering greater flexibility and broader applicability.

#### II. LMI FORMULATION

Following the works for [8], who defined a Lipschitz neural network as a constrained LMI problem to define a residual network, limitations in their approach were identified. Specifically, their formulation resulted in a single-layered residual network, which is inherently less expressive compared to the generalized deep-layered residual network popularized by architectures such as ResNet and its variants [16]-[20]. These deeper networks perform better due to the multiple inner layers that compose the modules, which allows for more complex latent space transformations and thus increases the network's expressiveness. This research focuses on establishing constraints for the inner layers to maintain the  $\mathcal{L}$ -Lipschitz condition while maximizing the expressiveness of the residual network for larger inner layers. As such, the inner layers of the residual network were represented as a recursive system of linear equations:

$$x_{k+1} = A_k x_k + B_k w_{k,n}$$

$$w_{k,n} = \sigma_n (C_n w_{k,n-1} + b_n)$$

$$\vdots$$

$$w_{k,1} = \sigma_1 (C_1 x_k + b_1). \tag{1}$$

Where each of the layer parameters were defined as  $C_l \in$  $\mathbb{R}^{d_l \times d_{l-1}}, b_l \in \mathbb{R}^{d_l}$  for  $l \in \{1, \dots, n\}$ . When n = 1, the formulation reduced to the one presented in [8], rendering it redundant in its derivation. The goal of the LMI was to maintain the Lipschitz constraint formulated as  $\|x'_{k+1} - x_{k+1}\| \le \mathcal{L} \|x'_k - x_k\|$ .

Given that this system could be represented as a large recursive system, it was possible to split all the constraints of the inner layers as a set of LMI conditions similar to [8], [10], [21]. For the most general LMI constraint definition, the activation functions were assumed to not necessarily be the ReLU function but a general element-wise activation function, which were L-smooth and m-strongly convex, where  $L_i \ge m_i$ . Thus, the general activation function quadratic constraint was used [8], [10]:

$$\begin{bmatrix} v_k - v_k' \\ w_{k,i}' - w_{k,i} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} -2L_i m_i \Lambda_i & (m_i + L_i) \Lambda_i \\ (m_i + L_i) \Lambda_i & -2\Lambda_i \end{bmatrix} \begin{bmatrix} v_k - v_k' \\ w_{k,i}' - w_{k,i} \end{bmatrix} \le 0$$

where  $\Lambda_n$  must be a positive definitive diagonal matrix. Given that  $v_k - v'_k = C_n \left( w_{k,n-1} - w'_{k,n-1} \right)$  the inequality thus becomes the following quadratic constraints, where  $\Delta w_{k,i}$  was defined as  $\Delta w_{k,i} = w'_{k,i} - w_{k,i}$ ,

$$\begin{bmatrix} C_1 \begin{pmatrix} x_k' - x_k \end{pmatrix} \\ \Delta w_{k,1} \end{bmatrix}^\top \begin{bmatrix} -2L_1 m_1 \Lambda_1 & (m_1 + L_1) \Lambda_1 \\ (m_1 + L_1) \Lambda_1 & -2\Lambda_1 \end{bmatrix} \begin{bmatrix} C_1 \begin{pmatrix} x_k' - x_k \end{pmatrix} \\ \Delta w_{k,1} \end{bmatrix} \leq 0,$$

$$\begin{bmatrix} C_2 \begin{pmatrix} \Delta w_{k,1} \end{pmatrix} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} -2L_2 m_2 \Lambda_2 & (m_2 + L_2) \Lambda_2 \\ (m_2 + L_2) \Lambda_2 & -2\Lambda_2 \end{bmatrix} \begin{bmatrix} C_2 \begin{pmatrix} \Delta w_{k,1} \end{pmatrix} \\ \Delta w_{k,2} \end{bmatrix} \leq 0,$$

$$\vdots$$

$$\begin{bmatrix} C_n \begin{pmatrix} \Delta w_{k,n-1} \end{pmatrix} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} -2L_n m_n \Lambda_n & (m_n + L_n) \Lambda_n \\ (m_n + L_n) \Lambda_n & -2\Lambda_n \end{bmatrix} \begin{bmatrix} C_n \begin{pmatrix} \Delta w_{k,n-1} \end{pmatrix} \\ \Delta w_{k,n} \end{bmatrix} \leq 0. \quad (3)$$

To combine the LMIs, a concatenated vector of all the  $w_{k,n}$ and  $x_k$  was created to sum all the conditions and solve them all together. The following LMI could thus be formulated as the summation in Equation (4). Where,

$$D_{l} = \sum_{i=1}^{l} d_{i},$$

$$E_{l} : \{0,1\}^{(d_{l}+d_{l-1}) \times D_{n}},$$
(5)

$$E_l: \{0,1\}^{(d_l+d_{l-1})\times D_n},\tag{6}$$

$$[E_l]_{ij} = \begin{cases} 1 & \text{if } j - D_l = i \\ 0 & \text{else} \end{cases},$$

moreover, i and j represented the row and column, respectively. The  $E_l$  matrix represented a "selection" vector to ensure that the proper variables were used for the parameterization. Which gave the following resultant LMI in Equation (7). The question then became what parameterization of  $\{\Lambda_1, \dots, \Lambda_n\}, \{C_1, \dots, C_n\}$ , and B would be needed to ensure that the LMI was indeed negative semi-definitive to satisfy the Lipschitz constraint where ideally  $\{C_1, \dots, C_n\}$  would be as unconstrained as possible to ensure expressive inner layers. From the LMI, it could be noticed that it was exceedingly complex to derive the constraint of the network explicitly based on the eigenvalues of the network. As such, although it only provided loose bounds on the eigenvalues, the Gershgorin circle theorem could be used to derive bounds on the network.

**Theorem II.1.** Let A be a complex matrix  $n \times n$  matrix, with entries  $a_{ij}$ . For  $i \in \{1, \dots, n\}$  let  $R_i$  be the sum of the absolute values of the non-diagonal entries of the i-th row:

$$R_i = \sum_{j \neq i} |a_{ij}|. \tag{8}$$

Let  $D(a_{ii}, R_i) \subseteq \mathbb{C}$  be a closed disc centered at  $a_{ii}$  with radius  $R_i$ , every eigenvalue of A lies within at least one of the Gershgorin discs  $D(a_{ii}, R_i)$ .

The following corollary was thus derived to generate conditions to ensure the LMI would be negative semi-definitive.

**Corollary 1.** If all the Gershgorin discs of a matrix A are defined in the negative real plane,  $\mathbb{R}_{-}$ , for  $i \in \{1, \dots, n\}$  $\Re(a_{ii} + R_i) \leq 0$ , then the matrix A must be negative semidefinitive.

The conditions necessary to ensure that the overall LMI matrix M was negative semi-definite were derived by analyzing its Gershgorin discs. The analysis required demonstrating that all Gershgorin discs were entirely contained within the left half-plane, ensuring that the eigenvalues of M were nonpositive. Given the structure of the LMI, the matrix could be decomposed into three distinct sections: the first block, the middle blocks, and the last block. For each block, a corresponding set of constraints on the desired parameters was determined to ensure the feasibility of the problem.

As the LMI matrix was symmetric, the Gershgorin discs derived from the rows were shown to coincide with those

$$\begin{bmatrix} x'_{k,1} - x_{k} \\ w'_{k,1} - w_{k,1} \\ w'_{k,2} - w_{k,2} \\ \vdots \\ w'_{k,n-1} - w_{k,n-1} \\ w'_{k,n} - w_{k,n} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 0_{d_{x}} & I_{d_{x}} \\ \vdots & 0_{d_{1}} \\ \vdots & \vdots \\ 0_{d_{n-1}} & \vdots \\ I_{d_{x}} & 0_{d_{x}} \end{bmatrix} \begin{bmatrix} A_{k}^{\mathsf{T}} A_{k} - \mathcal{L}^{2} I & A_{k}^{\mathsf{T}} B_{k} \\ B_{k}^{\mathsf{T}} A_{k} & B_{k}^{\mathsf{T}} B_{k} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} x'_{k} - x_{k} \\ w'_{k,1} - w_{k,1} \\ \vdots & \vdots \\ 0_{d_{n-1}} & \vdots \\ I_{d_{x}} & 0_{d_{x}} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} x'_{k} - x_{k} \\ w'_{k,1} - w_{k,2} \\ \vdots \\ w'_{k,n-1} - w_{k,n-1} \\ w'_{k,2} - w_{k,2} \\ \vdots \\ w'_{k,2} - w_{k,2} \\ \vdots \\ w'_{k,n-1} - w_{k,n-1} \\ w'_{k,2} - w_{k,2} \end{bmatrix}^{\mathsf{T}} E_{i}^{\mathsf{T}} \begin{bmatrix} C_{l} & 0 \\ 0 & I \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} -2L_{l}m_{l}\Lambda_{l} & (m_{l} + L_{l})\Lambda_{i} \\ (m_{l} + L_{l})\Lambda_{l} & -2\Lambda_{l} \end{bmatrix} \begin{bmatrix} C_{l} & 0 \\ 0 & I \end{bmatrix} E_{i} \begin{bmatrix} x'_{k} - x_{k} \\ w'_{k,1} - w_{k,1} \\ w'_{k,2} - w_{k,2} \\ \vdots \\ w'_{k,n-1} - w_{k,n-1} \\ w'_{k,n} - w_{k,n} \end{bmatrix} \leq 0, \tag{4}$$

$$\begin{bmatrix} A^{\mathsf{T}}A - I - 2L_{1}m_{1}C_{1}^{\mathsf{T}}\Lambda_{1}C_{1} & (L_{1} + m_{1})C_{1}^{\mathsf{T}}\Lambda_{1} & 0 & 0 & 0 & A^{\mathsf{T}}B \\ (L_{1} + m_{1})\Lambda_{1}C_{1} & -2L_{2}m_{2}C_{2}^{\mathsf{T}}\Lambda_{2}C_{2} - 2\Lambda_{1} & \ddots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \ddots & -2L_{n}m_{n}C_{n}^{\mathsf{T}}\Lambda_{n}C_{n} - 2\Lambda_{n-1} & (L_{n} + m_{n})C_{n}^{\mathsf{T}}\Lambda_{n} \\ 0 & B^{\mathsf{T}}A & 0 & 0 & 0 & (L_{n} + m_{n})\Lambda_{n}C_{n} & B^{\mathsf{T}}B - 2\Lambda_{n} \end{bmatrix} \leq 0, \tag{7}$$

derived from the columns. This symmetry allowed the analysis to be carried out equivalently from either perspective without losing generality.

# III. GENERAL LMI SOLUTION

For notation, the parameters  $S_a$  and  $P_a$  were defined as  $S_a = L_a + m_a$  and  $P_a = L_a m_a$  to help reduce the notation size.

# A. Last block

Below is the derivation of the constraints for the parameters defined in the last block portion of the LMI.

**Theorem III.1.** For the parameter  $C_n$ , the norm of the rows must be upper bounded by,

$$\|C_{n,i}\|_1 < \frac{2}{|L_n + m_n|},$$
 (9)

while  $\lambda_{n,i}$  must be lower bounded by,

$$\lambda_{n,i} \ge \frac{b_i^2 + |a_i||b_i|}{2 - |L_n + m_n||C_{n,i}||_1}.$$
 (10)

*Proof.* The final matrix row block was represented through the parameters where l = n:

$$\{B^{\mathsf{T}}A,(L_n+m_n)\Lambda_nC_n,B^{\mathsf{T}}B-2\Lambda_n\}.$$

Which gave the following Gershgorin discs for  $\forall i \{1, \dots, m_n\}$  (where  $m_x = m_n$ ):

$$D\left(b_i^2 - 2\lambda_{n,i}, |a_i||b_i| + \sum_{j=1}^{d_{n-1}} |(L_n + m_n)c_{n,i,j}\lambda_{n,i}|\right), \quad (11)$$

For which the upper bound constraint was thus:

$$\epsilon_{3,n,i,\max} = b_i^2 - 2\lambda_{n,i} + |a_i||b_i| + |S_n| \sum_{j=1}^{d_{n-1}} |c_{n,ij}\lambda_{n,i}|, \quad (12)$$

Applying the negative-semi definitive constraint, the following constraint was derived:

$$0 \ge b_i^2 - 2\lambda_{n,i} + |a_i||b_i| + |L_n + m_n| \sum_{j=1}^{d_{n-1}} |c_{n,i,j}\lambda_{n,i}|,$$

$$\ge b_i^2 + |a_i||b_i| + \lambda_{n,i} \left( |L_n + m_n| \sum_{j=1}^{d_{n-1}} |c_{n,ij}| - 2 \right),$$

$$\lambda_{n,i} \ge \frac{b_i^2 + |a_i||b_i|}{2 - |L_n + m_n| \sum_{j=1}^{d_{n-1}} |c_{n,ij}|}.$$
(13)

Given that all  $\lambda_{n,i}$  must be positive definitive, the only way to ensure this was to ensure that:

$$\sum_{j=1}^{d_{n-1}} |c_{n,ij}| = \|C_{n,i}\|_1 < \frac{2}{|L_n + m_n|}.$$
 (14)

Thus enforcing that all the rows of  $C_n$  must be strictly upper bound by  $\frac{2}{|L_n+m_n|}$ . QED

# B. Middle blocks

Below is the derivation of the constraints for the parameters defined in the middle blocks of the LMI.

**Theorem III.2.** For all  $l = 1, \dots, n-1$  the parameter  $C_l$  must have its row norm be upper bounded by,

$$\|C_{n,i}\|_1 < \frac{2}{|L_n + m_n|},$$
 (15)

and element-wise upper bounded by

$$|c_{l+1,ji}| \le \frac{|S_{l+1}|^2 + 4|P_{l+1}|}{2(|P_{l+1}| + P_{l+1})|S_{l+1}|}.$$
 (16)

while  $\lambda_{l,i}$  must be lower bounded by,

$$\lambda_{l,i} \ge \frac{\sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} \left( |S_{l+1}| |c_{l+1,ji}| - 2P_{l+1} c_{l+1,ji}^2 \right)}{2 - |S_l| \sum_{j=1}^{d_{l-1}} |c_{l,ij}|} + \frac{2|P_{l+1}| \sum_{z=1,z\neq i}^{d_l} \left| \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} c_{l+1,ji} c_{l+1,jz} \right|}{2 - |S_l| \sum_{j=1}^{d_{l-1}} |c_{l,ij}|}.$$
 (17)

$$\{S_{l}\Lambda_{l}C_{l}, -2P_{l+1}C_{l+1}^{\mathsf{T}}\Lambda_{l+1}C_{l+1} - 2\Lambda_{l}, S_{l+1}C_{l+1}^{\mathsf{T}}\Lambda_{n}\}.$$

Which gave the following Gershgorin discs,  $D(a_{ii}, R_i)$ , for  $\forall i\{1\cdots m_l\}\forall l\{1\cdots n-1\}$ :

$$a_{ii} = -2(L_{l+1}m_{l+1}) \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} c_{l+1,ji}^{2} - 2\lambda_{l,i},$$

$$R_{i} = \lambda_{l,i} |L_{l} + m_{l}| \sum_{j=1}^{d_{l-1}} |c_{l,ij}|$$

$$+ |L_{l+1} + m_{l+1}| \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} |c_{l+1,ji}|$$

$$+ 2|L_{l+1}m_{l+1}| \sum_{z=1,z\neq i}^{d_{l}} \left| \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} c_{l+1,ji} c_{l+1,ji} \right|.$$

$$(18)$$

For which the upper bound constraint was thus:

$$\epsilon_{2,l,i,\max} = a_{ii} + R_i,$$

$$= \lambda_{l,i} \left( |S_l| \sum_{j=1}^{d_{l-1}} |c_{l,ij}| - 2 \right)$$

$$+ \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} \left( |S_{l+1}| |c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^2 \right)$$

$$+ 2|P_{l+1}| \sum_{z=1}^{d_l} \sum_{z=i}^{d_{l+1}} \lambda_{l+1,j} c_{l+1,ji} c_{l+1,jz} \right|.$$
(21)

Applying the negative-semi definitive constraint, the following constraint was derived:

$$\lambda_{l,i} \ge \frac{\sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} \left( |S_{l+1}| |c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^2 \right)}{2 - |S_l| \sum_{j=1}^{d_{l-1}} |c_{l,ij}|} + \frac{2|P_{l+1}| \sum_{z=1,z\neq i}^{d_l} \left| \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j}c_{l+1,ji}c_{l+1,jz} \right|}{2 - |S_l| \sum_{j=1}^{d_{l-1}} |c_{l,ij}|}, \quad (22)$$

Given that all  $\lambda$  must be positive definitive and that the numerator and denominator are independent, the following constraints could thus be derived:

$$\sum_{j=1}^{d_{n-1}} |c_{l,ij}| = \|C_{l,i}\|_1 < \frac{2}{|L_l + m_l|}.$$
 (23)

Thus enforcing that all the rows of  $C_l$  had to be strictly upper bound by  $\frac{2}{|L_l+m_l|}$ .

In addition, the numerator was analyzed to ensure that the system remained positive and definite. A simplified approach was adopted by imposing the following conditions:

$$|S_{l+1}||c_{l+1,ji}| \ge 2P_{l+1}c_{l+1,ji}^2,$$

$$|S_{l+1}| \ge 2P_{l+1}|c_{l+1,ji}|,$$

$$|c_{l+1,ji}| \le \frac{|S_{l+1}|}{2P_{l+1}}.$$
(24)

The off-diagonal terms were examined to derive a less restrictive upper bound for the variable  $C_l$ . The radius  $R_i$  was increased to produce a more conservative estimate, which,

although broader, facilitated the inclusion of the off-diagonal terms within the inequality framework.

$$= \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} \left( |S_{l+1}| |c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^{2} \right) 
+ 2|P_{l+1}| \sum_{z=1,z\neq i}^{d_{l}} \sum_{j=1}^{|d_{l+1}} \lambda_{l+1,j}c_{l+1,ji}c_{l+1,ji} \right) 
+ 2|P_{l+1}| \sum_{z=1,z\neq i}^{d_{l}} \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j}|c_{l+1,ji}c_{l+1,ji}| 
+ 2|P_{l+1}| \sum_{z=1,z\neq i}^{d_{l}} \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j}|c_{l+1,ji}c_{l+1,ji}| 
= \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j} \left( |S_{l+1}| |c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^{2} \right) 
+ 2|P_{l+1}| \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j}|c_{l+1,ji}| \sum_{z=1,z\neq i}^{|d_{l}|} |c_{l+1,jz}| 
= \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j} \left( |S_{l+1}| |c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^{2} \right) 
+ 2|P_{l+1}| \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j}|c_{l+1,ji}| \left( \sum_{z=1}^{|d_{l}|} |c_{l+1,jz}| - |c_{l+1,ji}| \right) 
+ 2|P_{l+1}| \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j}|c_{l+1,ji}| \left( \frac{2}{|S_{l+1}|} - |c_{l+1,ji}| \right) 
= \sum_{j=1}^{|d_{l+1}|} \lambda_{l+1,j} \left( |S_{l+1}| |c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^{2} \right) 
+ 2|P_{l+1}| |c_{l+1,ji}| \left( \frac{2}{|S_{l+1}|} - |c_{l+1,ji}| \right) \right]. \quad (25)$$

Where the inner term needed to be constrained:

$$0 \leq |S_{l+1}||c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^{2}$$

$$+ 2|P_{l+1}||c_{l+1,ji}| \left(\frac{2}{|S_{l+1}|} - |c_{l+1,ji}|\right),$$

$$= |S_{l+1}||c_{l+1,ji}| - 2P_{l+1}c_{l+1,ji}^{2}$$

$$+ 2|P_{l+1}| \left(\frac{2|c_{l+1,ji}|}{|S_{l+1}|} - c_{l+1,ji}^{2}\right),$$

$$= |S_{l+1}||c_{l+1,ji}| + \frac{4|P_{l+1}|}{|S_{l+1}|}|c_{l+1,ji}|$$

$$- 2(P_{l+1} + |P_{l+1}|)c_{l+1,ji}^{2},$$

$$0 \leq |S_{l+1}| + \frac{4|P_{l+1}|}{|S_{l+1}|} - 2(P_{l+1} + |P_{l+1}|)|c_{l+1,ji}|$$

$$|c_{l+1,ji}| \leq \frac{|S_{l+1}|^{2} + 4|P_{l+1}|}{2(|P_{l+1}| + P_{l+1}|)|S_{l+1}|}.$$
(26)

Given the additional element-wise constraint, it was observed that there were two situations in which the constraint became irrelevant. The first scenario occurred when  $L_{l+1}m_{l+1} \le 0$ , and the second arose when  $L_{l+1}+m_{l+1}=0$ . This condition was satisfied, for instance, in the case of a ReLU activation function, where L=1 and m=0.

# C. First layer

Finally, below is the derivation of the constraints for the parameters defined in the first-row block of the LMI.

**Theorem III.3.** The parameter  $C_1$  must be element-wise upper bounded by

$$|c_{1,ji}| \le \frac{\left(\mathcal{L}^2 - a_i^2 - |a_i||b_i|\right)|S_1|}{d_1\lambda_{1,j}\left(|S_1|^2 + 4|P_1|\right)},$$
 (27)

*Proof.* This block represented the parameters where l = 1:

$$\{A^{\mathsf{T}}A - \mathcal{L}^2I - 2L_1m_1C_1^{\mathsf{T}}\Lambda_1C_1, (L_1+m_1)C_1^{\mathsf{T}}\Lambda_1, A^{\mathsf{T}}B\}.$$

Which gave the following Gershgorin discs for  $\forall i \{1 \cdots m_x\}$  (where  $m_x = m_n$ ):

$$a_{ii} = a_i^2 - \mathcal{L}^2 - 2(L_1 m_1) \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji}^2,$$

$$R_i = |a_i| |b_i| + |L_1 + m_1| \sum_{j=1}^{d_1} \lambda_{1,j} |c_{1,ji}|$$

$$+ 2|L_1 m_1| \sum_{z=1, z \neq i}^{d_x} \left| \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji} c_{1,jz} \right|.$$
(29)

For which the upper bound constraint was thus:

$$\epsilon_{1,1,i,\max} = a_i^2 - \mathcal{L}^2 - 2P_1 \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji}^2 + |a_i| |b_i|$$

$$+ |S_1| \sum_{j=1}^{d_1} \lambda_{1,j} |c_{1,ji}|$$

$$+ 2|P_1| \sum_{z=1,z\neq i}^{d_x} \left| \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji} c_{1,jz} \right|. \tag{30}$$

Applying the negative-semi definitive constraint, the following constraint was derived:

$$\begin{split} 0 \geq & a_i^2 - \mathcal{L}^2 - 2P_1 \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji}^2 + |a_i||b_i| + |S_1| \sum_{j=1}^{d_1} \lambda_{1,j}|c_{1,ji}| \\ & + 2|P_1| \sum_{z=1,z\neq i}^{d_x} \left| \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji} c_{1,jz} \right|, \\ \leq & a_i^2 - \mathcal{L}^2 - 2P_1 \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji}^2 + |a_i||b_i| + |S_1| \sum_{j=1}^{d_1} \lambda_{1,j}|c_{1,ji}| \\ & + 2|P_1| \sum_{z=1,z\neq i}^{d_x} \sum_{j=1}^{d_1} \lambda_{1,j}|c_{1,ji}| c_{1,jz}|, \\ = & a_i^2 - \mathcal{L}^2 - 2P_1 \sum_{j=1}^{d_1} \lambda_{1,j} c_{1,ji}^2 + |a_i||b_i| + |S_1| \sum_{j=1}^{d_1} \lambda_{1,j}|c_{1,ji}| \\ & + 2|P_1| \sum_{j=1}^{d_1} \lambda_{1,j}|c_{1,ji}| \sum_{z=1,z\neq i}^{d_x} |c_{1,jz}|, \\ = & a_i^2 - \mathcal{L}^2 + |a_i||b_i| + \sum_{j=1}^{d_1} \lambda_{1,j} \left( |S_1||c_{1,ji}| - 2P_1 c_{1,ji}^2 + 2|P_1||c_{1,ji}| \sum_{z=1,z\neq i}^{d_x} |c_{1,jz}| \right), \\ \leq & a_i^2 - \mathcal{L}^2 + |a_i||b_i| + \sum_{j=1}^{d_1} \lambda_{1,j} \left| |S_1||c_{1,ji}| - 2P_1 c_{1,ji}^2 \right| \\ \leq & a_i^2 - \mathcal{L}^2 + |a_i||b_i| + \sum_{j=1}^{d_1} \lambda_{1,j} \left| |S_1||c_{1,ji}| - 2P_1 c_{1,ji}^2 \right| \end{split}$$

$$+2|P_{1}||c_{1,ji}|\left(\frac{2}{|S_{1}|}-|c_{l+1,ji}|\right)\right],$$

$$=\sum_{j=1}^{d_{1}}\left[\frac{a_{i}^{2}-\mathcal{L}^{2}+|a_{i}||b_{i}|}{d_{1}}+\lambda_{1,j}\left(|S_{1}||c_{1,ji}|-2P_{1}c_{1,ji}^{2}\right)+2|P_{1}||c_{1,ji}|\left(\frac{2}{|S_{1}|}-|c_{1,ji}|\right)\right)\right]. (31)$$

The constraint  $L_1m_1 \le 0$  was enforced to ensure the solvability of the system. Consequently, the system of equations was formulated as follows:

$$0 \ge \frac{a_{i}^{2} - \mathcal{L}^{2} + |a_{i}||b_{i}|}{d_{1}} + \lambda_{1,j} \left( |S_{1}||c_{1,ji}| - 2P_{1}c_{1,ji}^{2} + 2|P_{1}||c_{1,ji}| \left( \frac{2}{|S_{1}|} - |c_{1,ji}| \right) \right),$$

$$= \frac{a_{i}^{2} - \mathcal{L}^{2} + |a_{i}||b_{i}|}{d_{1}} + \lambda_{1,j} \left( |S_{1}||c_{1,ji}| + \frac{4|P_{1}|}{|S_{1}|} |c_{1,ji}| \right),$$

$$|c_{1,ji}| \le \frac{\mathcal{L}^{2} - a_{i}^{2} - |a_{i}||b_{i}|}{d_{1}\lambda_{1,j}} \frac{1}{|S_{1}| + \frac{4|P_{1}|}{|S_{1}|}},$$

$$= \frac{\left(\mathcal{L}^{2} - a_{i}^{2} - |a_{i}||b_{i}|\right) |S_{1}|}{d_{1}\lambda_{1,j} \left( |S_{1}|^{2} + 4|P_{1}| \right)}.$$

$$(32)$$

Which then induced the inequality that  $\mathcal{L}^2 - a_i^2 - |a_i||b_i| \ge 0$ , which in turn gave the constraints that  $a_i \in (-\mathcal{L}, \mathcal{L})$  with  $|b_i| < \frac{\mathcal{L}^2 - a_i^2}{|a_i|}$ . This thus completed the LMI constraints. QED

Given all the derived constraints, the complete set of constraints of the neural network was listed in Table II.

The generalized versions of the equations could additionally be presented in matrix forms in Table II, where the absolute value function is applied element-wise, i.e.,  $|A| = \{|a_{ij}|\}$ , for simplified computation and practicality, the diagonal matrices  $\Lambda_l$ , A, B are represented as column vectors where the elements are the diagonal values.

# D. Weighted norm constraint

For a weighted  $\ell_1$ -norm, it was desired to derive an unparameterized optimization formulation scheme for  $x_i$  while ensuring the system remained upper bounded. Where  $v_i > 0$ ,  $\forall i$ .

$$a \le \|x\|_{1,v} = \sum_{i=1}^{n} v_i |x_i| \tag{33}$$

The reparameterization  $|x_i| \le \pm \frac{a}{n} \frac{1}{v_i}$  was introduced, such that:

$$a \le \sum_{i=1}^{n} v_i |x_i| \le \sum_{i=1}^{n} v_i \left| \frac{a}{n} \frac{1}{v_i} \right| = \frac{a}{n} \sum_{i=1}^{n} 1 = a$$
 (34)

Where the constraint,

$$x_i = \frac{a}{n} \frac{1}{v_i} p_i, \tag{35}$$

where  $p_i \in (-1,1)$ . As such, to parameterize  $x_i$ , the optimization parameter for the network becomes optimizing

Parameter	Inequality	Constraint	Indexing
$S_l$	=	$L_l + m_l$	$\forall l\{1,\cdots,n\}$
$P_l$	=	$L_l m_l$	$\forall l\{1,\cdots,n\}$
$\lambda_{n,i}$	≥	$\frac{b_i^2 \! + \! a_i b_i}{2 \! - \!  S_n  \sum_{j=1}^{d_n-1}  c_{n,ij} }$	$\forall i\{1,\cdots,d_n\}$
$\ C_{n,i}\ _1$	<	$\frac{2}{ S_l }$	$\forall i\{1,\cdots,d_l\} \forall l\{1,\cdots,n\}$
$\lambda_{l,i}$	2	$\frac{\sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} \left(  S_{l+1}   c_{l+1,ji}  - 2P_{l+1} c_{l+1,ji}^2 \right) + 2 P_{l+1}  \sum_{z=1,z\neq i}^{d_{l}} \left  \sum_{j=1}^{d_{l+1}} \lambda_{l+1,j} c_{l+1,ji} c_{l+1,jz} \right }{2 -  S_{l}  \sum_{j=1}^{d_{l-1}}  c_{l,ij} }$	$\forall i\{1,\cdots,d_l\} \forall l\{1,\cdots,n-1\}$
$ c_{l,ij} $	≤	$rac{ S_l ^2 + 4 P_l }{2( P_l  + P_l) S_l }$	$\forall i \{1, \cdots, d_l\} \forall j \{1, \cdots, d_{l-1}\} \forall l \{2, \cdots, n\}$
$ c_{1,ij} $	≤	$\frac{\left(\mathcal{L}^2 - a_i^2 - a_i b_i\right)  S_1 }{d_1 \lambda_{1,i} \left( S_1 ^2 + 4 P_1 \right)}$	$\forall i\{1,\cdots,d_x\}$
$ a_i $	€	$(0,\mathcal{L})$	$orall i\{1, \cdots, d_x\}$
$ b_i $	€	$[0,rac{\mathcal{L}^2-a_i^2}{ a_i })$	$\forall i\{1,\cdots,d_x\}$

Table I: General LMI Condensed Constraints

Table II: General LMI Matrix Condensed Constraints

Parameter	Inequality	Constraint	Indexing
$S_l$	=	$L_l + m_l$	
$P_l$	=	$L_l m_l$	
$D_l$	=	$\begin{bmatrix} \left\  C_{l,1} \right\ _1 & \cdots & \left\  C_{l,m_l} \right\ _1 \end{bmatrix}^{T}$	$\forall l\{1,\cdots,n\}$
$\Lambda_n$	≥	$\frac{B^2 +  A  B }{2 -  S_n D_n}$	
$D_l$	<	$\frac{2}{ S_l }$	$\forall l\{1,\cdots,n\}$
$Q_l$	=	$C_l^{ op} \mathrm{diag}(\Lambda_l) C_l$	
$\Lambda_l$	≥	$\frac{\Lambda_{l+1}^{T}( S_{l+1}  C_{l+1} -2P_{l+1}C_{l+1}^{\circ 2})+2 P_{l+1} 1^{T} Q_{l+1}-\mathrm{diag}(\mathrm{diag}(Q_{l+1})) }{2- S_{l} D_{l}}$	$\forall l\{1,\cdots,n-1\}$
$ C_l $	≤	$\frac{ S_l ^2 + 4 P_l }{2( P_l  + P_l) S_l }$	$\forall l\{2,\cdots,n\}$
$ C_1 $	≤	$\frac{ s_1  \left(\mathcal{L}^2 - A^2 -  A  B \right)}{d_1 \left( s_1 ^2 + 4 P_1 \right)} \Lambda_1^{-1}$	
A	€	$(0,\mathcal{L})$	
B	€	$[0,\frac{\mathcal{L}^2}{ A }- A )$	

 $p_i$ , where  $p_i = \tanh(w_i)$ , where  $w_i$  was an unconstrained optimization parameter. As demonstrated by the normalization factor  $\frac{\partial x_i}{\partial p_i} \propto O(\frac{1}{nv_i})$ , which implied that the gradients of  $x_i$  became proportionally smaller as the dimension of the vector became smaller. Small or vanishing gradients could cause problems for large and deeper networks.

# E. Elementwise vs. row constraint bound switching

1)  $C_l$  constraints: For the matrices  $C_l$  with  $l \in \{2, \dots, n\}$ , two simultaneous constraints were imposed on the system: the row-wise and element-wise constraints. In this context, the objective was to derive the upper bounds for the values of  $C_l$  that the optimization would be based on.

The following constraint was derived from the norm constraints established in the unparameterized optimization formulation given by Equation (35), where the upper bound was set as  $a = \frac{2}{|S_i|}$  and  $v_i = 1$ . The goal, therefore, was to identify the conditions under which the row-wise element constraint would dominate over the overall element-wise constraint.

$$\frac{2}{|S_{l}|d_{l-1}} \le \frac{|S_{l}|^{2} + 4|P_{l}|}{2(|P_{l}| + P_{l})|S_{l}|},$$

$$d_{l-1} \ge \frac{4(|P_{l}| + P_{l})}{|S_{l}|^{2} + 4|P_{l}|}.$$
(36)

This thus informed us that when  $P_l \le 0$  ( $\forall l, d_l \ge 0$ ), the element-wise constraint would always be greater than the element-wise, and if  $P_l > 0$  and,

$$d_{l-1} \ge \frac{8P_l}{|S_l|^2 + 4P_l}. (37)$$

By examining the maximum value of the bound, it was found that, due to the equation's symmetry concerning  $L_l$  and  $m_l$ , solving for either the optimal value of  $m_l$  or  $L_l$  led to the optimal solution. This symmetry implied that both parameters contributed equivalently to the system, and thus, optimizing one in isolation was sufficient to determine the overall optimal configuration.

$$\frac{\partial}{\partial m_l} \frac{8L_l m_l}{(L_l + m_l)^2 + 4L_l m_l} = \frac{8L_l (L_l - m_l)(L_l + m_l)}{(L_l^2 + 6L_l m_l + m_l^2)^2}, \quad (38)$$

solving for 0 the optimal value was obtained when  $m_l = \{-L_l, L_l\}$ , where only the  $m_l = L_l$  solution was kept due to the  $P_l > 0$  constraint. Which gave the solution that (when  $m_l = L_l$ ,  $S_l = 2L$ ,  $P_l = L_l^2$ ):

$$\frac{2}{|S_l|d_{l-1}} \ge \frac{|S_l|^2 + 4|P_l|}{2(|P_l| + P_l)|S_l|},$$

$$\frac{1}{|L_l|d_{l-1}} \ge \frac{4L_l^2 + 4L_l^2}{8L_l^2|L_l|},$$

$$\frac{1}{|L_l|d_{l-1}} \ge \frac{1}{|L_l|}. (39)$$

This demonstrated that even in the specific condition when  $m_l = L_l$  and  $d_l \le 1$ , the element and row-wise constraints would be equivalent to each other. This implied that the row-wise constraint would always be smaller than the element-wise constraint and should thus be the only one considered when constraining  $C_l$  for  $\forall \{2, \dots, n\}$ .

2)  $C_1$  constraints: Upon analyzing the constraints derived for  $C_1$ , it was observed that a mutual dependence existed between  $C_1$  and  $\Lambda_1$ . Specifically, the definition of  $C_1$  necessitated the prior specification of  $\Lambda_1$ , and conversely, the determination of  $\Lambda_1$  was contingent upon the specification of  $C_1$ . This interdependence introduced significant complexity in deriving an appropriate parameterization for  $C_1$ . As a result, an additional constraint was imposed on  $C_1$  to address this issue, such that:

$$|S_l| \sum_{j=1}^{d_x} |c_{1,ij}| \le 1, \tag{40}$$

Which thus enforced the constraint that,

$$G_{l} = \Lambda_{l}^{\mathsf{T}}(|S_{l}||C_{l}| - 2P_{l}C_{l}^{\circ 2}) + 2|P_{l}|\mathbf{1}^{\mathsf{T}}|Q_{l} - \operatorname{diag}(\operatorname{diag}(Q_{l}))|, \tag{41}$$

$$\lambda_{1,i} \ge \frac{G_2}{2 - |S_l| \sum_{j=1}^{d_x} |c_{1,ij}|} \ge G_2,$$
 (42)

where,  $G_l$  represented the numerator of the  $\Lambda_l$  parameterization. Enforcing this additional constraint on the row norm of  $C_1$  thus imposed an upper bound of  $\Lambda_1$ , which no longer contained a dependence on  $C_1$ , breaking the cyclical parameterization.

## F. LMI parameterization

The eigenvalue distribution of the LMI was displayed below in Figure 1 (The eigenvalue range was truncated to 10 times the quartile range; however, some of the eigenvalues have reached a magnitude of  $-10^{11}$ ). To generate this distribution, all the parameters, the weights  $C_l$  (parameterized by  $p_l$ ), and the biases  $b_l$  were initialized with a uniform distribution. The biases and weights were initialized using the standard Kaiming initialization scheme, where the weights used tanh gains (i.e., scaling of 1 for tanh [22]) given that the variables  $p_l$  were constrained by tanh.

The constraints above, when implemented, thus generated the following example of Gershgorin circles for the LMI illustrated in Figure 2. The Figure demonstrates that the Gershgorin circles were all constrained on the negative real plane.

It was also interesting to observe that due to the recursive nature of the  $\Lambda_l$  parameterization, the Gershgorin circles ended up encapsulating each other most of the time (this is not a general statement).

For the sake of completeness, the L and m constants of the activation functions defined in PyTorch (assuming default values if not specified) were derived and defined in Table III. It should be noted that the Hardshrink and RReLU could not

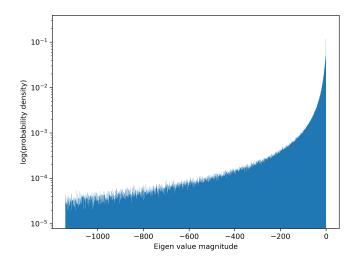


Figure 1: Eigenvalue distribution

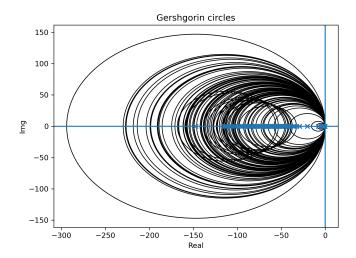


Figure 2: LMI Gershgorin Circles

be used due to their infinite L,m constants; Hardshink has infinite L,m due to its noncontinuous piece-wise definition, and PReLU due to its stochastic definition, which no longer made it's L,m computable. Where,

$$\operatorname{erfc}(z) = 1 - \operatorname{erf}(z),$$
 (43)

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt.$$
 (44)

#### IV. EXPERIMENT

Based on the designed network constraints, a network was thus generated. To run such a network due to the codependence of the  $\Lambda_l, L_l$  and  $m_l$  from the next layer and the first layer, the evaluation of the network needed to be run in two passes, a backward pass which computed the  $\Lambda_l$  and  $C_l$  parameters, as illustrated in Figure 3, and then, the forward pass performed the inferences using the computed parameters as a standard residual network as illustrated in 4. It should be noted that it was not possible to make use of techniques

Activation Function	L	m	S	P
ELU ( $\alpha = 1$ ) [23]	$\max(1, \alpha)$	0	$\max(1, \alpha)$	0
Hardshrink [24]	∞	0	∞	∞
Hardsigmoid [25]	$\frac{1}{6}$	0	$\frac{1}{6}$	0
Hardtanh [26]	Ĭ	0	Ĭ	0
Hardswish [27]	1.5	-0.5	1	-0.75
LeakyReLU ( $\alpha = 1e^{-2}$ ) [28]	1	α	$1 + \alpha$	α
LogSigmoid	1	0	1	0
PReLU $(\alpha = \frac{1}{4})$ [29]	1	$\alpha$	$1 + \alpha$	$\alpha$
ReLU [30]	1	0	1	0
ReLU6 [31]	1	0	1	0
RReLU [32]	∞	-∞	∞	∞
SELU [33]	$\alpha \times \text{scale} \approx 1.758099341$	0	$\alpha \times \text{scale} \approx 1.758099341$	0.0
CELU [34]	1	0	1	0
GELU [35]	$\frac{\operatorname{erfc}(1)}{2} - \frac{1}{e\sqrt{\pi}}$	$\frac{1}{2}(\text{erf}(1)+1)+\frac{1}{e\sqrt{\pi}}$	1	$\frac{\left(e\sqrt{\pi}(\operatorname{erf}(1)+1)+2\right)\left(e\sqrt{\pi}\operatorname{erfc}(1)-2\right)}{4e^2\pi}$
	≈ 1.128904145	≈ -0.1289041452		≈ -0.145520424
Sigmoid [36]	1	0	1	0
SiLU [37]	1.099839320	-0.09983932013	1	-0.1098072100
Softplus [38]	1	0	1	0
Mish $(\alpha \ge \frac{1}{2})$ [39]	1.199678640	-0.2157287822	0.8060623125	-0.2204297485
Softshrink [24]	1	0	1	0
Softsign [40]	1	0	1	0
Tanh [36]	1	0	1	0
Tanhshrink	1	0	1	0
Threshold	1	0	1	0

Table III: Convexity constants of the element-wise activation functions in PyTorch

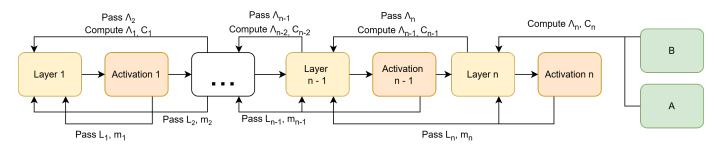


Figure 3: Backwards pass

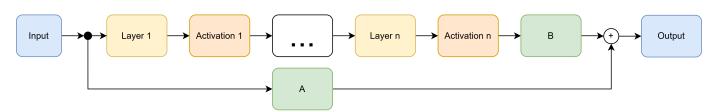


Figure 4: Forward pass

such as a batch normalization [17], [41], [42], which is a common practice in more modern ResNet architectures. This was because normalization was not a constrained  $\mathcal{L}$ -Lipschitz formulation as the normalized features were computed as [13], [43]:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}]}},$$
(45)

Which could be represented as a linear layer where,

$$C_b = \operatorname{diag}\left(\sqrt{\operatorname{Var}[x^{(1)}]}, \cdots, \sqrt{\operatorname{Var}[x^{(d)}]}\right)^{-1}, \qquad (46)$$

$$b_b = -\left[\frac{E[x^{(1)}]}{\sqrt{Var[x^{(1)}]}} \quad \cdots \quad \frac{E[x^{(d)}]}{\sqrt{Var[x^{(d)}]}}\right]^{\top}.$$
 (47)

Where it would only be in particular conditions that the batch normalization would follow the constraints posed by Table II; this is due to the variance scaling term being very hard to control and is defined by the dataset that is inputted into the system.

To test the network's capabilities, it was initially tested on a straightforward dataset to fit  $y=\frac{1}{2}\sin(x)$  on  $x\in(-2\pi,2\pi)$ , which is a  $\frac{1}{2}$  Lipchitz bounded function as  $arg\max_x\frac{\partial}{\partial x}\frac{1}{2}\sin(x)=\frac{1}{2}arg\max_x\cos(x)=\frac{1}{2}$ , which should thus make it possible to train the network on. However, it was noticed that no matter what optimizer, activation function, size or number of hidden layers, learning rate, or other hyperparameters used, the system would be unable to fit the function to any degree of accuracy using the MSE loss function. This

is illustrated from the output results in Figures 5 and 6.

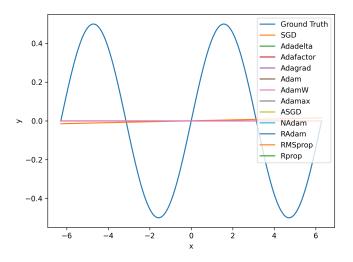


Figure 5: Trained L-Lipschitz network output over multiple optimizers

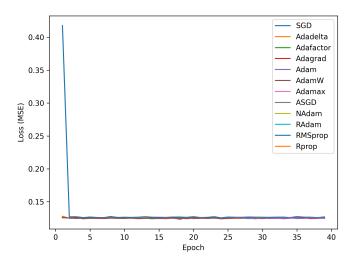


Figure 6: MSE training loss over multiple optimizers

The network's output looked like a single regression line, and no expressiveness of the network could be observed. For a minimum error to fit with this line, it would have to be a line defined as:

$$\mathcal{L}_{l} = \frac{1}{4\pi} \int_{-2\pi}^{2\pi} (ax + b - \sin(x))^{2} dx,$$

$$= \frac{1}{6} \left( 4a \left( 2\pi^{2} a + 3 \right) + 6b^{2} + 3 \right). \tag{48}$$

Whose minimum would be defined when  $a=-\frac{3}{4\pi^2}$  and b=0, with a total error of  $\mathcal{L}_l=\frac{1}{2}-\frac{3}{4\pi^2}$ . After further inspection of the network, the main culprit in the decay issue was  $C_1$ 's magnitude as the  $\Lambda_1$  magnitude in the network became very large, which caused an over-constraining of the  $C_1$  matrix parameter. Given that the Gershgorin circle theorem is only an approximation of the eigenvalue locations, this caused the overall network's compounding approximations to overconstrain the network and thus disable the non-linear portion

of the system as such, the network comes the simple  $y \approx Ax$  formula, where A is the parameterized diagonal matrix. It is thus sadly noted that this type of network with the current type of parameterization for the weights and biases of the system does not function as a universal function approximator. As such this paper is only able to elaborate on the current methodology for solving the LMI using the Gershgorin circle for more complicated general LMI structures; however, it should be noted that if the LMI follows a more standard matrix structure such as a tri-diagonal form [10] common in a standard Feedforward Neural Network (FNN) or the likes it is possible to derive more exact eigenvalue constraints on the system.

#### V. Conclusion

This study rigorously derived constraints for the pseudo-tridiagonal matrix LMI representing a residual network. Given the absence of explicit eigenvalue computations for the tridiagonal matrix with off-diagonal elements, the Gershgorin circle theorem was employed to approximate the eigenvalue locations of this complex recursive system. The system was decomposed into three distinct blocks, and weight parameterizations were systematically derived and summarized in Table II.

A two-step process was detailed once the constraints were established and the network was constructed. Due to the residual network's recursive nature, the normalization parameters needed to be computed and propagated in advance to enable the creation of layer weight parameterizations. This stage was defined as the backward pass. The forward pass involved performing inference on the network.

Upon evaluating the implemented network, it was observed that the Gershgorin circle approximations caused the normalization factors of the inner layers to deactivate the network's non-linear components. Consequently, based on the Gershgorin formulation, the final implementation proved ineffective and unsuitable as a functional approximation. This study establishes a foundation for future research into alternative eigenvalue approximations and refined parameterization strategies, advancing robust deep learning architectures' theoretical and practical development.

## REFERENCES

- [1] M. Inkawhich, Y. Chen, and H. Li, "Snooping attacks on deep reinforcement learning," Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS, vol. 2020-May, pp. 557–565, 5 2019. [Online]. Available: https://arxiv.org/abs/1905.11832v2
- [2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 12 2014. [Online]. Available: https://arxiv.org/abs/1412.6572v3
- [3] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks," *CoRR*, vol. abs/1802.04034, 2018. [Online]. Available: http://arxiv.org/abs/1802.04034
- [4] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2 2018. [Online]. Available: https://arxiv.org/abs/ 1802.05957v1

- [5] P. L. Bartlett, D. J. Foster, and M. Telgarsky, "Spectrally-normalized margin bounds for neural networks," *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 6241–6250, 6 2017. [Online]. Available: https://arxiv.org/abs/1706.08498v2
- [6] B. Prach and C. H. Lampert, "Almost-orthogonal layers for efficient general-purpose lipschitz networks," 8 2022. [Online]. Available: https://arxiv.org/abs/2208.03160v2
- [7] L. Meunier, B. J. Delattre, A. Araujo, and A. Allauzen, "A dynamical system perspective for lipschitz neural networks," pp. 15 484–15 500, 6 2022. [Online]. Available: https://proceedings.mlr. press/v162/meunier22a.html
- [8] A. Araujo, A. Havens, B. Delattre, A. Allauzen, and B. Hu, "A unified algebraic perspective on lipschitz neural networks," 3 2023. [Online]. Available: http://arxiv.org/abs/2303.03169
- [9] L. Meunier, B. Delattre, A. Araujo, and A. Allauzen, "A dynamical system perspective for lipschitz neural networks," *Proceedings of Machine Learning Research*, vol. 162, pp. 15484–15500, 10 2021. [Online]. Available: https://arxiv.org/abs/2110.12690v2
- [10] Y. Xu and S. Sivaranjani, "Eclipse: Efficient compositional lipschitz constant estimation for deep neural networks," 4 2024. [Online]. Available: https://arxiv.org/abs/2404.04375v2
- [11] A. Sandryhaila and J. M. F. Moura, "Eigendecomposition of block tridiagonal matrices," 6 2013. [Online]. Available: https://arxiv.org/abs/1306.0217v1
- [12] E. Agarwal, S. Sivaranjani, V. Gupta, and P. Antsaklis, "Sequential synthesis of distributed controllers for cascade interconnected systems," *Proceedings of the American Control Conference*, vol. 2019-July, pp. 5816–5821, 7 2019.
- [13] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing lipschitz continuity," *Machine Learning*, vol. 110, pp. 393–416, 2 2021. [Online]. Available: http://link.springer.com/10.1007/s10994-020-05929-w
- [14] S. Aziznejad, H. Gupta, J. Campos, and M. Unser, "Deep neural networks with trainable activations and controlled lipschitz constant," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4688– 4699, 1 2020. [Online]. Available: http://arxiv.org/abs/2001.06263http: //dx.doi.org/10.1109/TSP.2020.3014611
- [15] J. Bear, A. Prügel-Bennett, and J. Hare, "Rethinking deep thinking: Stable learning of algorithms using lipschitz constraints," 10 2024. [Online]. Available: https://arxiv.org/abs/2410.23451v1
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, 12 2015. [Online]. Available: https://arxiv.org/abs/1512.03385v1
- [17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 31st AAAI Conference on Artificial Intelligence, AAAI 2017, pp. 4278–4284, 2 2016. [Online]. Available: https://arxiv.org/abs/1602.07261v2
- 4284, 2 2016. [Online]. Available: https://arxiv.org/abs/1602.07261v2 [18] S. Zagoruyko and N. Komodakis, "Wide residual networks," *British Machine Vision Conference 2016, BMVC 2016*, vol. 2016-September, pp. 87.1–87.12, 5 2016. [Online]. Available: https://arxiv.org/abs/1605. 07146v4
- [19] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2011–2023, 9 2017. [Online]. Available: https://arxiv.org/abs/1709.01507v4
- [20] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR* 2017, vol. 2017-January, pp. 5987–5995, 11 2016. [Online]. Available: https://arxiv.org/abs/1611.05431v2
- [21] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas, Efficient and accurate estimation of lipschitz constants for deep neural networks. Red Hook, NY, USA: Curran Associates Inc., 2019.
- [22] S. K. Kumar, "On weight initialization in deep neural networks," 4 2017. [Online]. Available: https://arxiv.org/abs/1704.08863v2
- [23] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, 11 2015. [Online]. Available: http://arxiv.org/abs/1511.07289
- [24] H. F. Cancino-De-Greiff, R. Ramos-Garcia, and J. V. Lorenzo-Ginori, "Signal de-noising in magnetic resonance spectroscopy using wavelet transforms," *Concepts in Magnetic Resonance*, vol. 14, pp. 388–401, 1 2002. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/ cmr.10043

- [25] M. Courbariaux, Y. Bengio, and J. P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," *Advances in Neural Information Processing Systems*, vol. 2015-January, pp. 3123–3131, 11 2015. [Online]. Available: https://arxiv.org/abs/1511. 00363v3
- [26] R. Collobert, "Large scale machine learning," Ph.D. dissertation, Université de Paris VI, 2004.
- [27] A. Howard, M. Sandler, B. Chen, W. Wang, L. C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, Q. Le, and H. Adam, "Searching for mobilenetv3," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 1314–1324, 5 2019. [Online]. Available: https://arxiv.org/abs/1905.02244v5
- [28] A. L. Maas, Y. H. Awni, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *Proceedings of the 30th International Conference on Machine Learning*, vol. 28, 6 2013.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2 2015. [Online]. Available: http://arxiv.org/abs/1502.01852
- [30] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 12 1943. [Online]. Available: https://link.springer.com/article/10.1007/BF02478259
- [31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 4 2017. [Online]. Available: http://arxiv.org/abs/1704.04861
- [32] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 5 2015. [Online]. Available: http://arxiv.org/abs/1505.00853
- [33] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," Advances in Neural Information Processing Systems, vol. 30, 2017.
- [34] J. T. Barron, "Continuously differentiable exponential linear units," 4 2017. [Online]. Available: http://arxiv.org/abs/1704.07483
- [35] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 6 2016. [Online]. Available: https://arxiv.org/abs/1606.08415v5
- [36] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2 2014. [Online]. Available: https://arxiv.org/abs/1402. 1128v1
- [37] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2 2017. [Online]. Available: https://arxiv.org/abs/1702.03118v3
- [38] M. Zhou, "Softplus regressions and convex polytopes," 8 2016. [Online]. Available: http://arxiv.org/abs/1608.06383
- [39] D. Misra, "Mish: A self regularized non-monotonic activation function," 31st British Machine Vision Conference, BMVC 2020, 8 2019. [Online]. Available: http://arxiv.org/abs/1908.08681
- [40] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," 6th International Conference on Learning Representations, ICLR 2018 Conference Track Proceedings, 10 2017. [Online]. Available: http://arxiv.org/abs/1710.07654
- [41] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 6 2017. [Online]. Available: https://arxiv.org/abs/1706.05587
- [42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-December, pp. 770–778, 12 2015. [Online]. Available: https://arxiv.org/abs/1512.03385v1
- [43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 32nd International Conference on Machine Learning, ICML 2015, vol. 1, pp. 448–456, 2 2015. [Online]. Available: https://arxiv.org/abs/1502.03167v3