2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Local Lipschitz Constant Computation of ReLU-FNNs: Upper Bound Computation with Exactness Verification

Yoshio Ebihara, Xin Dai, Tsuyoshi Yuno, Victor Magron, Dimitri Peaucelle, and Sophie Tarbouriech

***Abstract*— This paper is concerned with the computation of the local Lipschitz constant of feedforward neural networks (FNNs) with activation functions being rectified linear units (ReLUs). The local Lipschitz constant of an FNN for a target input is a reasonable measure for its quantitative evaluation of the reliability. By following a standard procedure using multipliers that capture the behavior of ReLUs, we first reduce the upper bound computation problem of the local Lipschitz constant into a semidefinite programming problem (SDP). Here we newly introduce copositive multipliers to capture the ReLU behavior accurately. Then, by considering the dual of the SDP for the upper bound computation, we second derive a viable test to conclude the exactness of the computed upper bound. However, these SDPs are intractable for practical FNNs with hundreds of ReLUs. To address this issue, we further propose a method to construct a reduced order model whose input-output property is identical to the original FNN over a neighborhood of the target input. We finally illustrate the effectiveness of the model reduction and exactness verification methods with numerical examples of practical FNNs.**

Keywords: feedforward neural networks (FNNs), rectified linear units (ReLUs), local Lipschitz constant, upper bound, copositive multiplier, exactness verification, model reduction.

I. Introduction

Control theoretic approaches for the reliability certification of deep neural networks have gained considerable attention recently. These studies are roughly classified into two categories: the treatment of static feedforward neural networks (FNNs) [1], [2], [3], [4], [5] and the treatment of dynamical networks such as recurrent neural networks (RNNs) [6], [7], [8], [9]. In the present paper, we focus on FNNs, and investigate the computation problem of the local Lipschitz constant for a target input. The effectiveness of FNNs in image recognition and pattern classification, etc., is widely recognized. However, it is known that some FNNs exhibit unreliable behavior such that small perturbation on an input leads to a totally different output [10]. The existence of such inputs, known as adversarial inputs (perturbations), shows the unreliability of the FNN. By computing the local Lipschitz constant, we can ensure the absence of such adversarial inputs, and thereby certify the reliability of the FNN.

In this paper, we first reduce the upper bound computation problem of the local Lipschitz constant into a semidefinite programming problem (SDP), by following the standard procedure using multipliers that capture the input-output behavior of ReLUs in quadratic constraints [4]. Here we newly introduce a set of multipliers constructed from copositive matrices [11], in order to accurately capture the nonnegative behavior of ReLUs. In addition, we prove that the copositive multipliers encompass existing multipliers such as O'Shea-Zames-Falb multipliers [12], [13], [14], [15] and the multipliers introduced in [4]. We next focus on the dual of the SDP for the upper bound computation. Then we derive a rank condition on the dual optimal solution under which we can conclude the exactness of the obtained upper bound. In particular, this also enables us to extract the worst case input that deviates the corresponding output most far away from the original one. However, for practical FNNs with hundreds of ReLUs, these SDPs are inherently intractable, since the size of the multiplier grows linearly with the number of ReLUs and thus the computational complexity for solving these SDPs increases quite rapidly. To address this issue, we further propose a method to construct a reduced order model whose input-output property is identical to the original FNN over a neighborhood of the target input. We finally illustrate the effectiveness of the model reduction and exactness verification methods by numerical examples on a practical FNN in [2] designed as an MNIST classifier.

In this paper, all the proofs for technical results are omitted due to limited space.

Notation: The set of $n \times m$ real matrices is denoted by $\mathbb{R}^{n\times m}$, and the set of $n \times m$ entrywise nonnegative matrices is denoted by $\mathbb{R}_+^{n\times m}$. For a matrix A, we also write $A \geq 0$ to denote that A is entrywise nonnegative. We denote the set of $n \times n$ real symmetric matrices by $\mathbb{S}^n$. For $A \in \mathbb{S}^n$, we write $A \succ 0$ ($A \prec 0$) to denote that A is positive (negative) definite. For $A \in \mathbb{R}^{n\times n}$, we define $\mathrm{He}\{A\} := A + A^T$. For $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, $(*)^T AB$ is a shorthand notation of $B^T AB$. We denote by $\mathbb{D}^n \subset \mathbb{R}^{n\times n}$ the set of diagonal matrices. A matrix $M \in \mathbb{R}^{n\times n}$ is said to be Z-matrix if $M_{ij} \leq 0$ for all $i \neq j$. Moreover, M is said to be doubly hyperdominant if it is a Z-matrix and $M\mathbf{1}_n \geq 0$, $\mathbf{1}_n^T M \geq 0$, where $\mathbf{1}_n \in \mathbb{R}^n$ stands for the all-ones-vector. We denote by $\mathbb{DHD}^n \subset \mathbb{R}^{n\times n}$ the set of doubly hyperdominant matrices. For $v \in \mathbb{R}^n$, we define $|v|_2 := \sqrt{\sum_{j=1}^n v_j^2}$. For $v_0 \in \mathbb{R}^n$

Y. Ebihara, T. Yuno, and X. Dai are with the Graduate School of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan, Y. Ebihara was also with LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, in 2011. V. Magron, D. Peaucelle, and S. Tarbouriech are LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France. This work was supported by JSPS KAKENHI Grant Number JP20H04145 and JP21H01354. This work was also supported by the AI Interdisciplinary Institute ANITI funding, through the French "Investingfor the Future PIA3" program under the Grant agreement n° ANR-19-PI3A-0004 as well as by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

and $\varepsilon > 0$, we define $\mathcal{B}_\varepsilon(v_0) := \{v \in \mathbb{R}^n : \ |v - v_0|_2 \le \varepsilon\}$. Finally, for the i-th unit vector $e_i \in \mathbb{R}^n$ and the index set $\mathcal{N} \subset \{1, \cdots, n\}$, we denote by $\bigoplus_{i\in\mathcal{N}} e_i \in \mathbb{R}^{|\mathcal{N}|\times n}$ the matrix that arrays e_i^T vertically for $i \in \mathcal{N}$. For instance, if $n = 4$ and $\mathcal{N} = \{1, 2\}$ then we have $\bigoplus_{i\in\mathcal{N}} e_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

II. Local Lipschitz Constant Computation Problem of ReLU-FNNs and Basic Results

A. Local Lipschitz Constant Computation Problem

Let us consider the single-layer FNN described by

$$G: \ z = W_{\text{out}}\Phi(W_{\text{in}}w + b_{\text{in}}) + b_{\text{out}}. \tag{1}$$

Here $w \in \mathbb{R}^m$ and $z \in \mathbb{R}^l$ are the input and the output of the FNN, respectively. On the other hand, $W_{\text{out}} \in \mathbb{R}^{l\times n}$ and $W_{\text{in}} \in \mathbb{R}^{n\times m}$ are constant matrices constructed from the weightings of the edges in the FNN. The constant vectors $b_{\text{in}} \in \mathbb{R}^n$ and $b_{\text{out}} \in \mathbb{R}^l$ are biases at input and output, respectively. Finally, $\Phi : \ \mathbb{R}^n \to \mathbb{R}^n$ is the static activation function typically being nonlinear. In the following, we denote by $z = G(w)$ the input-output map of the FNN.

In this paper, we consider the typical case where the activation function is the (entrywise) rectified linear unit (ReLU) whose input-output property is given by

$$\begin{aligned} &\Phi(q) = [\ \phi(q_1) \ \cdots \ \phi(q_n)\ ]^T, \\ &\phi : \mathbb{R} \to \mathbb{R}, \quad \phi(\eta) = \begin{cases} \eta & (\eta \ge 0), \\ 0 & (\eta < 0). \end{cases} \end{aligned} \tag{2}$$

For $p, q \in \mathbb{R}^n$, it is shown in [1], [16] that $p = \Phi(q)$ holds if and only if

$$p - q \ge 0, \ p \ge 0, \ (p - q) \otimes p = 0 \tag{3}$$

where $\otimes$ stands for the Hadamard product.

The local Lipschitz constant for the FNN given by (1) at a target input is defined as follows.

Definition 1: (Local Lipschitz Constant at Target Input) For given target input $w_0 \in \mathbb{R}^m$ and error bound $\varepsilon > 0$, the local Lipschitz constant $L_{w_0,\varepsilon}$ for the FNN G given by (1) and (2) is defined by

$$L_{w_0,\varepsilon} := \min\{L : \ |G(w) - G(w_0)|_2 \le L \ \forall w \in \mathcal{B}_\varepsilon(w_0)\}. \tag{4}$$

The target input $w_0 \in \mathbb{R}^m$ and the error bound $\varepsilon > 0$ are determined by the one who evaluates the reliability of the FNN. The goal of this paper is to compute the local Lipschitz constant as accurately as possible. From Definition 1, we see that the bias b_{out} at output is irrelevant to the local Lipschitz constant. Therefore we let $b_{\text{out}} = 0$ in (1) in the following.

The computation of the local Lipschitz constant is strongly motivated by the demand of robustness analysis of FNNs against adversarial inputs (perturbations). To be more concrete, let us follow the convincing discussion in [17] and consider the case where the FNN G given by (1) is used as a classifier. Namely, the FNN G receives m features as input and returns l scores as output. We define the classification rule $C : \mathbb{R}^m \to \{1, \cdots, l\}$ by

$$C(w) := \operatorname*{argmax}_{1\le i\le l} G_i(w). \tag{5}$$

Now suppose w_0 is an input that is classified correctly by the classifier. Then, the classifier is locally robust at the target input w_0 against all the perturbed inputs $w \in \mathcal{B}_\varepsilon(w_0)$ if

$$C(w) = C(w_0) \ \forall w \in \mathcal{B}_\varepsilon(w_0). \tag{6}$$

Regarding this condition for ensuring the robustness of the FNN, it has been shown in [17] that the next result holds.

Proposition 1: [17] For the classifier $C : \mathbb{R}^m \to \{1, \cdots, l\}$ given by (5) and (1) and given $w_0 \in \mathbb{R}^m$ and $\varepsilon > 0$, let us define $i^\star := C(w_0)$. Then the condition (6) holds if

$$L_{w_0,\varepsilon} \le \frac{1}{\sqrt{2}} \min_{1\le j\le l, j\ne i^\star} G_{i^\star}(w_0) - G_j(w_0). \tag{7}$$

From this proposition, we see that we can examine the absence of adversarial inputs and hence ensure the reliability of the FNN by computing the local Lipschitz constant.

B. Basic Results

To grasp the properties of the ReLU given by (2), we borrow the idea of integral quadratic constraint (IQC) theory [18]. Namely, we introduce the set of multipliers $\mathbf{\Pi}^\star \subset \mathbb{S}^{2n+1}$ such that

$$\mathbf{\Pi}^\star := \left\{ \Pi \in \mathbb{S}^{2n+1} : \ \begin{bmatrix} 1 \\ q \\ p \end{bmatrix}^T \Pi \begin{bmatrix} 1 \\ q \\ p \end{bmatrix} \ge 0 \ \forall q, p \in \mathbb{R}^n \text{ s.t. } p = \Phi(q) \right\}. \tag{8}$$

This type of multiplier is already introduced in [3], [4]. Then, we can readily obtain the next results.

Theorem 1: For given input $w_0 \in \mathbb{R}^m$ and $\varepsilon > 0$, let us define $z_0 := G(w_0)$ where G is the FNN given by (1) and (2). Then, $L_{w_0,\varepsilon} \le \sqrt{L_{\text{sq}}}$ holds for the FNN G if there exist $\tau \ge 0$ and $\Pi \in \mathbf{\Pi}^\star$ such that

$$\begin{aligned} (*)^T \begin{bmatrix} -L_{\text{sq}} + \tau\varepsilon^2 & 0 & 0 \\ * & -\tau I_m & 0 \\ * & * & I_l \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -w_0 & I_m & 0 \\ -z_0 & 0 & W_{\text{out}} \end{bmatrix} \\ +(*)^T \Pi \begin{bmatrix} 1 & 0 & 0 \\ b_{\text{in}} & W_{\text{in}} & 0 \\ 0 & 0 & I_n \end{bmatrix} \preceq 0 \end{aligned} \tag{9}$$

where $*$ stands for the matrix determined by symmetry.

From Theorem 1, we see that the upper bound computation problem of the Lipschitz constant for the FNN given by (1) can be formulated as

$$\inf_{L_{\text{sq}}, \tau, \Pi\in\mathbf{\Pi}^\star} L_{\text{sq}} \quad \text{subject to (9)}. \tag{10}$$

We note that Theorem 1 is a special case result of [4] where more general class of reliability verification problems are considered. Our novel contributions in this paper include: (i) providing novel copositive multipliers capturing accurately the input-output behavior of ReLUs; (ii) deriving an exactness verification test of the computed upper bounds by taking the dual of the problem (10); (iii) constructing reduced order models enabling us to deal with practical FNNs with hundreds of ReLUs.

III. Novel Multipliers for ReLUs and Comparison with Existing Ones

In this section, we consider the concrete set of multipliers that satisfy the condition in (8). To this end, we define the positive semidefinite cone $\mathcal{PSD}^n \subset \mathbb{S}^n$, the copositive cone $\mathcal{COP}^n \subset \mathbb{S}^n$, and the nonnegative cone $\mathcal{NN}^n \subset \mathbb{S}^n$ as follows:

$$\mathcal{PSD}^n := \{P \in \mathbb{S}^n : \ x^T P x \geq 0 \ \forall x \in \mathbb{R}^n\},$$
$$\mathcal{COP}^n := \{P \in \mathbb{S}^n : \ x^T P x \geq 0 \ \forall x \in \mathbb{R}_+^n\},$$
$$\mathcal{NN}^n := \{P \in \mathbb{S}^n : \ P \geq 0\}.$$

We can readily see that $\mathcal{PSD}^n \subset \mathcal{PSD}^n + \mathcal{NN}^n \subset \mathcal{COP}^n$ where "+" here stands for the Minkowski sum. The copositive programming problem (COP) is a convex optimization problem in which we minimize a linear objective function over the linear matrix inequality (LMI) constraints on the copositive cone [11]. As mentioned in [11], the problem to determine whether a given symmetric matrix is copositive or not is a co-NP complete problem. Therefore, it is hard to solve COP numerically in general. However, since the problem to determine whether a given matrix is in $\mathcal{PSD} + \mathcal{NN}$ can readily be reduced to an semidefinite programming problem (SDP), we can numerically solve the convex optimization problems on $\mathcal{PSD} + \mathcal{NN}$.

A. Novel Copositive Multipliers for ReLU

By focusing on (3), we can state the first main result of this paper.

Theorem 2: Let us define $\mathbf{\Pi}_{\mathcal{COP}}, \mathbf{\Pi}_{\mathcal{NN}} \subset \mathbb{S}^{2n+1}$ by

$$\begin{aligned} \mathbf{\Pi}_{\mathcal{COP}} &:= \big\{\Pi \in \mathbb{S} : \ \Pi = (*)^T \left(Q + \mathcal{J}(J)\right) E, \\ &\qquad J \in \mathbb{D}^n, \ Q \in \mathcal{COP}^{2n+1}\big\}, \\ \mathbf{\Pi}_{\mathcal{NN}} &:= \big\{\Pi \in \mathbb{S} : \ \Pi = (*)^T \left(Q + \mathcal{J}(J)\right) E, \\ &\qquad J \in \mathbb{D}^n, \ Q \in \mathcal{NN}^{2n+1}\big\}, \\ E &:= \begin{bmatrix} 1 & 0_{1,n} & 0_{1,n} \\ 0_{n,1} & -I_n & I_n \\ 0_{n,1} & 0_n & I_n \end{bmatrix}, \ \mathcal{J}(J) := \begin{bmatrix} 0 & 0_{1,n} & 0_{1,n} \\ * & 0_{n,n} & J \\ * & * & 0_{n,n} \end{bmatrix}. \end{aligned} \tag{11}$$

Then we have $\mathbf{\Pi}_{\mathcal{NN}} \subset \mathbf{\Pi}_{\mathcal{COP}} \subset \mathbf{\Pi}^\star$.

In the following, we review existing multipliers capturing the behavior of ReLUs. Then, we clarify the inclusion relationship among those existing ones and the set of copositive multipliers $\mathbf{\Pi}_{\mathcal{COP}}$ and its inner approximation $\mathbf{\Pi}_{\mathcal{NN}}$.

B. O'Shea-Zames-Falb Multipliers

We now review the arguments in [15] on O'Shea-Zames-Falb multipliers [12], [13]. The overview of O'Shea-Zames-Falb multipliers can be found at [14], where the contributions of O'Shea to the development of the multipliers are emphasized. Due to this reason, we call the multipliers as O'Shea-Zames-Falb multipliers as opposed to the well celebrated moniker Zames-Falb multipliers. This modified moniker has already been employed, e.g., in [19].

We first introduce the following definition.

Definition 2: Let $\mu \leq 0 \leq \nu$. Then the nonlinearity $\phi : \mathbb{R} \to \mathbb{R}$ is slope-restricted, in short $\phi \in \mathrm{slope}(\mu, \nu)$, if $\phi(0) = 0$ and

$$\mu \leq \frac{\phi(x) - \phi(y)}{x - y} \leq \sup_{x \neq y} \frac{\phi(x) - \phi(y)}{x - y} \leq \nu$$

for all $x, y \in \mathbb{R}$, $x \neq y$.

The main result of [15] on the static O'Shea-Zames-Falb multipliers for slope-restricted nonlinearities can be summarized by the next lemma.

Lemma 1: [15] For a given nonlinearity $\phi \in \mathrm{slope}(\mu, \nu)$ with $\mu \leq 0 \leq \nu$, let us define $\Phi : \ \mathbb{R}^m \to \mathbb{R}^m$ by $\Phi := \mathrm{diag}(\phi, \cdots, \phi)$. Assume $M \in \mathbb{DHD}^m$. Then we have

$$(*)^T \begin{bmatrix} 0 & M^T \\ M & 0 \end{bmatrix} \left(\begin{bmatrix} \nu I_m & -I_m \\ -\mu I_m & I_m \end{bmatrix} \begin{bmatrix} x \\ \Phi(x) \end{bmatrix} \right) \geq 0 \ \forall x \in \mathbb{R}^m.$$

From this key lemma and the fact that the ReLU $\phi : \mathbb{R} \to \mathbb{R}$ satisfies $\phi \in \mathrm{slope}(0, 1)$, we can obtain the next result.

Proposition 2: Let us define

$$\begin{aligned} \widehat{\mathbf{\Pi}}_{\mathrm{OZF}} &:= \left\{\widehat{\Pi} \in \mathbb{S} : \ \widehat{\Pi} = (*)^T \begin{bmatrix} 0 & M \\ M^T & 0 \end{bmatrix} \begin{bmatrix} I_n & -I_n \\ 0 & I_n \end{bmatrix}, \ M \in \mathbb{DHD}^n \right\}, \\ \mathbf{\Pi}_{\mathrm{OZF}} &:= \left\{\Pi \in \mathbb{S} : \ \Pi = \mathrm{diag}(0, \widehat{\Pi}), \ \widehat{\Pi} \in \widehat{\mathbf{\Pi}}_{\mathrm{OZF}}\right\}. \end{aligned} \tag{12}$$

Then we have $\mathbf{\Pi}_{\mathrm{OZF}} \subset \mathbf{\Pi}^\star$.

C. Multipliers by Fazlyab et al. [4]

We next review the multipliers introduced by Fazlyab et al. in [4]. To this end, let us define

$$\mathbb{T}^n := \left\{ T \in \mathbb{S}^n : \ T = \sum_{1 \leq i < j \leq n} \lambda_{ij} (e_i - e_j)(e_i - e_j)^T, \ \lambda_{ij} \geq 0 \right\}.$$

Then, the result in Lemma 3 of [4] can be summarized as follows.

Proposition 3: [4] Let us define $\mathbf{\Pi}_{\mathrm{Faz}} \in \mathbb{S}^{2n+1}$ by

$$\mathbf{\Pi}_{\mathrm{Faz}} := \left\{ \Pi \in \mathbb{S} : \ \Pi = \begin{bmatrix} 0 & -\nu & \nu + \eta \\ * & 0 & \Lambda + T \\ * & * & -2(\Lambda + T) \end{bmatrix}, \ \nu^T, \eta^T \in \mathbb{R}_+^n, \ \Lambda \in \mathbb{D}^n, \ T \in \mathbb{T}^n \right\}. \tag{13}$$

Then we have $\mathbf{\Pi}_{\mathrm{Faz}} \subset \mathbf{\Pi}^\star$.

D. Comparison among Multipliers

We are now ready to clarify the relationship among the set of multipliers $\mathbf{\Pi}_{\mathcal{COP}}$ and $\mathbf{\Pi}_{\mathcal{NN}}$ given by (11), $\mathbf{\Pi}_{\mathrm{OZF}}$ given by (12), and $\mathbf{\Pi}_{\mathrm{Faz}}$ given by (13). The next theorem is one of the main results of the present paper.

Theorem 3: For the sets of multipliers $\mathbf{\Pi}_{\mathcal{COP}}$ and $\mathbf{\Pi}_{\mathcal{NN}}$ given in (11), $\mathbf{\Pi}_{\mathrm{OZF}}$ in (12), and $\mathbf{\Pi}_{\mathrm{Faz}}$ in (13), we have $\mathbf{\Pi}_{\mathrm{OZF}} \subset \mathbf{\Pi}_{\mathcal{NN}} \subset \mathbf{\Pi}_{\mathcal{COP}}$ and $\mathbf{\Pi}_{\mathrm{Faz}} \subset \mathbf{\Pi}_{\mathcal{NN}} \subset \mathbf{\Pi}_{\mathcal{COP}}$.

On the basis of this result, we now describe the SDP for the upper bound computation of the local Lipschitz constant of the FNN given by (1) and (2). Since the inclusion relationships in Theorem 3 hold, and since $\mathbf{\Pi}_{\mathcal{COP}}$ is numerically intractable, we focus on the following SDP:

$$\gamma_{\mathrm{primal}}^2 := \inf_{L_{\mathrm{sq}}, \tau, \Pi \in \mathbf{\Pi}_{\mathcal{NN}}} \ L_{\mathrm{sq}} \text{ subject to } (9). \tag{14}$$

If this SDP is feasible, we can conclude that $L_{w_0,\varepsilon} \leq \gamma_{\mathrm{primal}}$.

Remark 1: The usefulness of the copositive multipliers or its inner approximation has been already observed in the stability analysis of recurrent neural networks (RNNs) with activation functions being ReLUs, see [9], [20].

IV. Dual SDP and Exactness Verification

By solving the (primal) SDP (14), we can obtain an upper bound of the local Lipschitz constant. However, if we merely rely on such upper bound computation, we cannot draw any definite conclusion on how far the upper bound is close to the exact one. To address this issue, it is well known in the field of robust control that considering the dual problem is useful [21], [22], [23]. By following the Lagrange duality theory for SDPs [24], the dual SDP of (14) can be obtained as follows:

$$
\begin{aligned}
&\gamma_{\rm dual}^2 := \sup_{H\in\mathbb{S}_+^{1+m+n}} \mathrm{trace}\left((*)^T H \begin{bmatrix} z_0^T \\ 0_{m,l} \\ -W_{\rm out}{}^T \end{bmatrix}\right) \ \text{subject to} \\
&H_{11} = 1, \\
&\mathrm{trace}\left(\begin{bmatrix} \varepsilon^2 - w_0^T w_0 & w_0^T \\ * & -I_m \end{bmatrix}\begin{bmatrix} H_{11} & H_{12} \\ * & H_{22}\end{bmatrix}\right) \geq 0, \\
&\mathrm{diag}(-b_{\rm in} H_{13} - W_{\rm in} H_{23} + H_{33}) = 0, \\
&(*)^T H \begin{bmatrix} 1 & 0_{1,m} & 0_{1,n} \\ -b_{\rm in} & -W_{\rm in} & I_n \\ 0_{1,n} & 0_{n,m} & I_n \end{bmatrix}^T \geq 0, \\
&H =: \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ * & H_{22} & H_{23} \\ * & * & H_{33} \end{bmatrix},\ H_{22}\in\mathbb{S}_+^m,\ H_{33}\in\mathbb{S}_+^n
\end{aligned}
\tag{15}
$$

We can prove that the primal SDP (14) is strongly feasible. Therefore, the dual SDP (15) has an optimal solution, and there is no duality gap between (14) and (15) and thus $\gamma_{\rm primal} = \gamma_{\rm dual}$ holds [25]. By focusing on the dual SDP (15), we can obtain the next results on the exactness verification of the computed upper bounds.

Theorem 4: Suppose $\mathrm{rank}(H) = 1$ holds for the optimal solution $H \in \mathbb{S}_+^{1+m+n}$ of the dual SDP (15). Then we have $L_{w_0,\varepsilon} = \gamma_{\rm dual}$. Moreover, the full-rank factorization of the optimal solution $H \in \mathbb{S}_+^{1+m+n}$ is given by

$$
H = \begin{bmatrix} 1 \\ h_2 \\ h_3 \end{bmatrix}\begin{bmatrix} 1 \\ h_2 \\ h_3 \end{bmatrix}^T,\ h_2\in\mathbb{R}^m,\ h_3\in\mathbb{R}^n. \tag{16}
$$

In addition, if we define $w^\star := h_2$, then $w^\star \in \mathcal{B}_\varepsilon(w_0)$ holds, and $w^\star \in \mathcal{B}_\varepsilon(w_0)$ is one of the worst-case inputs satisfying $L_{w_0,\varepsilon} = |G(w^\star) - G(w_0)|_2$.

V. Exact Model Reduction around Target Input

When solving the SDP (14), we have to deal with the multiplier variable $\Pi \in \mathbf{\Pi}^\star \subset \mathbb{S}^{2n+1}$. Here, n stands for the number of ReLUs and this would be at least a few hundred in practical FNNs. Therefore, in practical problem settings the SDP (14) becomes intractable. To address this issue, in this section, we propose a method for the reduction of the number of ReLUs while maintaining the input-output behavior of the original FNN, by making full use of the property of ReLUs and the local Lipschitz constant computation problem.

To be precise, let us consider the behavior of

$$
z = W_{\rm out}\Phi(W_{\rm in} w + b_{\rm in}),\ W_{\rm in}\in\mathbb{R}^{n\times m},\ W_{\rm out}\in\mathbb{R}^{l\times n}
$$

for $w \in \mathcal{B}_\varepsilon(w_0)$. To this end, let us define $q = W_{\rm in} w + b_{\rm in}$ and $q_0 = W_{\rm in} w_0 + b_{\rm in}$. Then we see

$$
\min_{w\in\mathcal{B}_\varepsilon(w_0)} q_i = q_{0,i} - \varepsilon|W_{{\rm in},i}|_2,\ \max_{w\in\mathcal{B}_\varepsilon(w_0)} q_i = q_{0,i} + \varepsilon|W_{{\rm in},i}|_2
$$

where $W_{{\rm in},i} \in \mathbb{R}^{1\times m}$ stands for the i-th row of $W_{\rm in} \in \mathbb{R}^{n\times m}$. It follows that

$$
\begin{cases} q_{0,i} \geq \varepsilon|W_{{\rm in},i}|_2 \ \Rightarrow\ q_i \geq 0\ \forall w\in\mathcal{B}_\varepsilon(w_0), \\ q_{0,i} \leq -\varepsilon|W_{{\rm in},i}|_2 \ \Rightarrow\ q_i \leq 0\ \forall w\in\mathcal{B}_\varepsilon(w_0). \end{cases} \tag{17}
$$

With this fact in mind, let us define

$$
\begin{aligned}
&\mathcal{Z}_n := \{1, 2, \cdots, n\}, \\
&\mathcal{N}_+ := \{i \in \mathcal{Z}_n :\ q_{0,i} \geq \varepsilon|W_{{\rm in},i}|_2\}, \\
&\mathcal{N}_0 := \{i \in \mathcal{Z}_n :\ q_{0,i} \leq -\varepsilon|W_{{\rm in},i}|_2\}, \\
&\mathcal{N}_r := \mathcal{Z}_n \setminus \{\mathcal{N}_+ \cup \mathcal{N}_0\}.
\end{aligned}
\tag{18}
$$

Then, q_i with $i \in \mathcal{N}_+$ is never rectified for all $w \in \mathcal{B}_\varepsilon(w_0)$, q_i with $i \in \mathcal{N}_0$ is rectified for all $w \in \mathcal{B}_\varepsilon(w_0)$, and for q_i with $i \in \mathcal{N}_r$ we cannot say anything definitely. This motivates us to define

$$
\begin{aligned}
&E_+ := \bigoplus_{i\in\mathcal{N}_+} e_i,\ E_r := \bigoplus_{i\in\mathcal{N}_r} e_i, \\
&\widetilde{W}_{\rm in} := E_+ W_{\rm in},\ \widetilde{b}_{\rm in} := E_+ b_{\rm in}, \widetilde{W}_{\rm out} := W_{\rm out} E_+^T, \\
&\widehat{W}_{\rm in} := E_r W_{\rm in},\ \widehat{b}_{\rm in} := E_r b_{\rm in}, \widehat{W}_{\rm out} := W_{\rm out} E_r^T.
\end{aligned}
\tag{19}
$$

Then, we can obtain the next reduced order model with the number of ReLUs being $n_r := |\mathcal{N}_r|$.

$$
G_r : z = \widetilde{W}_{\rm out}(\widetilde{W}_{\rm in} w + \widetilde{b}_{\rm in}) + \widehat{W}_{\rm out}\Phi(\widehat{W}_{\rm in} w + \widehat{b}_{\rm in}). \tag{20}
$$

By the construction procedure of this reduced order model, the next results readily hold.

Theorem 5: For the FNN G given by (1) and (2), let us consider the reduced order model G_r constructed by (20), (18), and (19). Then, we have

$$
G(w) = G_r(w)\ \forall w \in \mathcal{B}_\varepsilon(w_0). \tag{21}
$$

Remark 2: From the observation (17), we have constructed explicitly an exact reduced order model G_r whose input-output behavior is identical to the original FNN G for inputs $w \in \mathcal{B}_\varepsilon(w_0)$. Similar observations to (17) are used in [4] for the purpose of tightening multiplier relaxations.

By using the reduced order model G_r, we can derive the primal and dual SDPs for the upper bound computation of the local Lipschitz constant $L_{w_0,\varepsilon}$ of the original FNN G. These are given as follows:

Primal SDP:

$$
\begin{aligned}
&\gamma_{\rm red,primal}^2 := \inf_{L_{\rm sq},\tau,\Pi\in\mathbf{\Pi}_{\mathcal{NN}}} L_{\rm sq} \ \text{subject to} \\
&(*)^T \begin{bmatrix} -L_{\rm sq} + \tau\varepsilon^2 & 0 & 0 \\ * & -\tau I_m & 0 \\ * & * & I_l \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ -w_0 & I_m & 0 \\ \widetilde{W}_{\rm out}\widetilde{b}_{\rm in} - z_0 & \widetilde{W}_{\rm out}\widetilde{W}_{\rm in} & \widehat{W}_{\rm out} \end{bmatrix} \\
&\qquad + (*)^T \Pi \begin{bmatrix} 1 & 0 & 0 \\ \widehat{b}_{\rm in} & \widehat{W}_{\rm in} & 0 \\ 0 & 0 & I_{n_r} \end{bmatrix} \preceq 0.
\end{aligned}
\tag{22}
$$

Dual SDP:

$$\gamma_{\rm red,dual}^2 := \sup_{H\in\mathbb{S}_+^{1+m+n_r}} \mathrm{trace}\left((*)^T H \begin{bmatrix} (\widetilde{W}_{\rm out}\widetilde{b}_{\rm in} - z_0)^T \\ \widetilde{W}_{\rm in}^T\widetilde{W}_{\rm out}^T \\ \widehat{W}_{\rm out}^T \end{bmatrix}\right) \text{ subject to}$$

$$\begin{aligned}
&H_{11} = 1,\\
&\mathrm{trace}\left(\begin{bmatrix}\varepsilon^2 - w_0^T w_0 & w_0^T \\ * & -I_m\end{bmatrix}\begin{bmatrix}H_{11} & H_{12}\\ * & H_{22}\end{bmatrix}\right) \geq 0,\\
&\mathrm{diag}(-\widehat{b}_{\rm in}H_{13} - \widehat{W}_{\rm in}H_{23} + H_{33}) = 0,\\
&(*)^T H \begin{bmatrix} 1 & 0_{1,m} & 0_{1,n_r} \\ -\widehat{b}_{\rm in} & -\widehat{W}_{\rm in} & I_{n_r} \\ 0_{1,n_r} & 0_{n_r,m} & I_{n_r}\end{bmatrix}^T \geq 0,\\
&H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ * & H_{22} & H_{23} \\ * & * & H_{33}\end{bmatrix},\ H_{22}\in\mathbb{S}_+^m,\ H_{33}\in\mathbb{S}_+^{n_r}.
\end{aligned} \tag{23}$$

In the primal SDP (22), we emphasize that the size of the multiplier variable Π has been reduced to $2n_r+1$. Similarly, in the dual SDP (23), the size of the dual variable H has been reduced to $1+m+n_r$.

Again, we can prove that the primal SDP (22) is strongly feasible. Therefore, the dual SDP (23) has an optimal solution, and there is no duality gap between (22) and (23). Namely, $\gamma_{\rm red,primal} = \gamma_{\rm red,dual} \geq L_{w_0,\varepsilon}$ holds. In addition, as for the exactness verification, the next results follow.

Theorem 6: Suppose $\mathrm{rank}(H) = 1$ holds for the optimal solution $H \in \mathbb{S}_+^{1+m+n_r}$ of the dual SDP (23). Then we have $L_{w_0,\varepsilon} = \gamma_{\rm red,dual}$. Moreover, the full-rank factorization of the optimal solution $H \in \mathbb{S}_+^{1+m+n_r}$ is given by

$$H = \begin{bmatrix}1\\h_2\\h_3\end{bmatrix}\begin{bmatrix}1\\h_2\\h_3\end{bmatrix}^T,\ h_2\in\mathbb{R}^m,\ h_3\in\mathbb{R}^{n_r}. \tag{24}$$

In addition, if we define $w^\star := h_2$, then $w^\star \in \mathcal{B}_\varepsilon(w_0)$ holds, and $w^\star \in \mathcal{B}_\varepsilon(w_0)$ is one of the worst-case inputs satisfying $L_{w_0,\varepsilon} = |G(w^\star) - G(w_0)|_2$.

VI. Numerical Examples

In this section, we illustrate the usefulness of the model reduction and exactness verification methods by numerical examples. When solving SDPs, we used MATLAB 2023a and the SDP solver MOSEK [26] together with the parser YALMIP [27] on a computer with CPU 12th Gen Intel(R) Core(TM) i9-12900 2.40 GHz.

We use the MNIST classifier (SDP-NN) described in [2], which is a single-layer ReLU-FNN. The inputs are 28×28 pixel data of handwritten digits from 0 to 9. The magnitude of each pixel is normalized to [0 1]. The number of ReLUs used in this FNN is 500. It follows that $m = 784$, $l = 10$, and $n = 500$ in (1). The classification rule is given by $C(w) := \operatorname{argmax}_{1\leq i\leq 10} G_i(w) - 1$.

As for the target input $w_0 \in \mathbb{R}^m$, we chose the one whose image is shown in Fig. 1 where $|w_0|_2 = 9.9652$. The corresponding output $z_0 = G(w_0)$ is also shown in Fig. 1. We can confirm that the input w_0 is correctly classified as 2.

If we directly work on this FNN and solve the primal SDP (14), we have to deal with the multiplier variable Π of size $2n+1 = 1001$. However, this is numerically intractable. We therefore applied the exact model reduction method described in Section V. The results are shown in Fig. 2. As expected, we have achieved considerable reduction of the numbers of ReLUs especially when ε is small. In particular, when $\varepsilon = 0.1$, the number of ReLUs in the reduced order model G_r given by (20) was $n_r = 26$.

By letting $\varepsilon = 0.1$, we solved the primal SDP (22) relying on the reduced order model and obtained $\gamma_{\rm red,primal} = 0.1416$. The CPU time was 1476 [sec]. In addition, by the command `dual`, we extracted the dual optimal solution H, and its rank was numerically verified to be one. From Theorem 6, we then constructed $w^\star \in \mathbb{R}^{784}$ whose image is shown in Fig. 3. The corresponding output $z^\star = G(w^\star)$ is also shown in Fig. 3. It turned out that

$$|w^\star - w_0|_2 = 0.1000,\ |G(w^\star) - G(w_0)|_2 = 0.1416.$$

Namely, the computed upper bound is exact and hence $L_{w_0,\varepsilon} = 0.1416$ holds. In addition, $w^\star \in \mathbb{R}^{784}$ is one of the worst-case inputs achieving $L_{w_0,\varepsilon} = |G(w^\star) - G(w_0)|_2$. We emphasize that the w_0 is a genuine worst-case input for the original FNN G, even though it is constructed from the reduced order model G_r. Here, if we replace the set of multiplier $\mathbf{\Pi}_{\mathcal{NN}}$ by $\mathbf{\Pi}_{\rm OZF}$ in the primal SDP (22), the resulting upper bound turned out to be 0.1599. The CPU time was 188 [sec]. Even though the computation time is much shorter, the obtained upper bound is far from the exact one obtained above. This clearly shows the usefulness of the newly proposed set of multiplier $\mathbf{\Pi}_{\mathcal{NN}}$.

We see from Fig. 3 that even the worst-case input $w^\star$ is correctly classified as 2. This assertion can be strengthened by the robustness test (7) in Proposition 1. Namely, regarding $z_0 = G(w_0)$ shown in Fig. 1, we see that

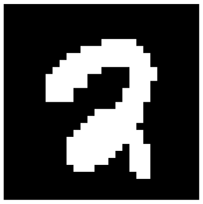
$$L_{w_0,\varepsilon} = 0.1416 < 0.1485 = \frac{1}{\sqrt{2}}(0.7448 - 0.5347).$$

Therefore we are led to the definite conclusion that there is no adversarial input that leads to false-classification within $w \in \mathcal{B}_{w_0,\varepsilon}$ for $\varepsilon = 0.1$.

VII. Conclusion and Future Works

In this paper, we considered the computation problem of the local Lipschitz constants for single-layer FNNs. By following standard procedure using multipliers, we reduced the upper bound computation problem into an SDP. Our novel contributions in this paper include: (i) providing novel copositive multipliers capturing accurately the input-output behavior of ReLUs; (ii) deriving an exactness verification test of the computed upper bounds by taking the dual of the SDP; (iii) constructing reduced order models enabling us to deal with practical FNNs with hundreds of ReLUs. We finally illustrated the usefulness of the model reduction and exactness verification methods by numerical examples on practical FNNs.

In this paper, we evaluated the magnitude of signals with respect to the L_2 norm. However, when dealing with robustness certification problems in deep learning, it can be preferable to employ the L_∞-norm. We intend to extend the present methods to this setting and compare the resulting performance with those in [1], [5].



$$\Rightarrow \quad \text{FNN} \Rightarrow z_0 = \begin{bmatrix} 0.1506 \\ -0.1238 \\ \mathbf{0.7448} \\ -0.1835 \\ -0.1983 \\ -0.2167 \\ 0.0474 \\ 0.3736 \\ 0.5347 \\ 0.3528 \end{bmatrix}$$

Fig. 1. Image of the input w_0 and corresponding output $z_0 = G(w_0)$.

$$\Rightarrow \quad \text{FNN} \Rightarrow z^\star = \begin{bmatrix} 0.1017 \\ -0.1703 \\ \mathbf{0.6991} \\ -0.2244 \\ -0.2384 \\ -0.2613 \\ 0.0026 \\ 0.3275 \\ 0.4931 \\ 0.3051 \end{bmatrix}$$

Fig. 3. Image of the input $w^\star$ and corresponding output $z^\star = G(w^\star)$.

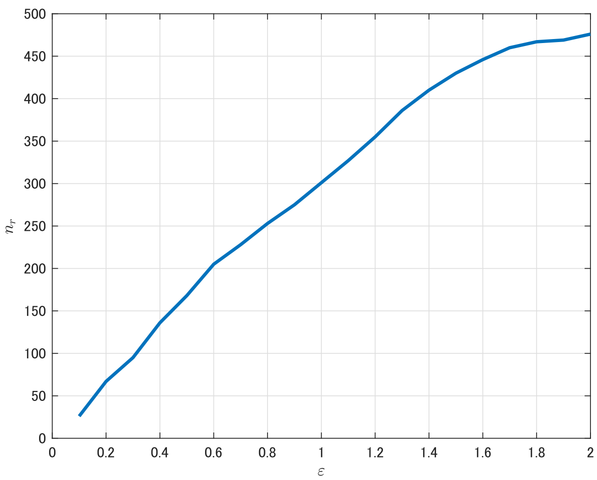


Fig. 2. Model reduction results.

REFERENCES

[1] A. Raghunathan, J. Steinhardt, and P. Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems*, pp. 10900–10910, 2018.

[2] A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. In *Proc. the International Conference on Learning Representations*, 2018.

[3] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas. Efficient and accurate estimation of Lipschitz constants for deep neural networks. In *Proc. the 33rd International Conference on Neural Information Processing Systems*, pp. 11427–11438, 2019.

[4] M. Fazlyab, M. Morari, and G. J. Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Transactions on Automatic Control*, Vol. 67, No. 1, pp. 1–15, 2022.

[5] T. Chen, J. B. Lasserre, V. Magron, and E. Pauwels. Semialgebraic optimization for Lipschitz constants of ReLU networks. *Advances in Neural Information Processing Systems*, Vol. 33, pp. 19189–19200, 2020.

[6] H. Yin, P. Seiler, and M. Arcak. Stability analysis using quadratic constraints for systems with neural network controllers. *IEEE Transactions on Automatic Control*, Vol. 67, No. 4, pp. 1980–1987, 2022.

[7] C. W. Scherer. Dissipativity and integral quadratic constraints: Tailored computational robustness tests for complex interconnections. *IEEE Control Systems Magazine*, Vol. 42, No. 3, pp. 115–139, 2022.

[8] M. Revay, R. Wang, and I. R. Manchester. A convex parameterization of robust recurrent neural networks. *IEEE Control Systems Letters*, Vol. 5, No. 4, pp. 1363–1368, 2021.

[9] Y. Ebihara, H. Waki, V. Magron, N. H. A. Mai, D. Peaucelle, and S. Tarbouriech. l_2 induced norm analysis of discrete-time LTI systems for nonnegative input signals and its application to stability analysis of recurrent neural networks. *The 2021 ECC Special Issue of the European Journal of Control*, Vol. 62, pp. 99–104, 2021.

[10] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust physical-world attacks on deep learning models. In *arxiv:1707.08945 [cs.CR]*.

[11] M. Dür. Copositive programming - a survey. In M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels, editors, *Recent Advances in Optimization and Its Applications in Engineering*, pp. 3–20. Springer, 2010.

[12] R. O'Shea. An improved frequency time domain stability criterion for autonomous continuous systems. *IEEE Transactions on Automatic Control*, Vol. 12, No. 6, pp. 725–731, 1967.

[13] G. Zames and P. Falb. Stability conditions for systems with monotone and slope-restricted nonlinearities. *SIAM Journal on Control*, Vol. 6, No. 1, pp. 89–108, 1968.

[14] J. Carrasco, M.C. Turner, and W.P. Heath. Zames–Falb multipliers for absolute stability: From O'shea's contribution to convex searches. *European Journal of Control*, Vol. 28, pp. 1–19, 2016.

[15] M. Fetzer and C. W. Scherer. Absolute stability analysis of discrete time feedback interconnections. *IFAC PapersOnline*, Vol. 50, No. 1, pp. 8447–8453, 2017.

[16] L. B. Groff, G. Valmorbida, and J. M. Gomes da Silva. Stability analysis of piecewise affine discrete-time systems. In *Proc. Conference on Decision and Control*, pp. 8172–8177, 2019.

[17] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. J. Pappas. Efficient and accurate estimation of Lipschitz constants for deep neural networks. In *arxiv:1906.04893v2 [cs.LG]*.

[18] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, Vol. 42, No. 6, pp. 819–830, 1997.

[19] C. W. Scherer. Dissipativity, convexity and tight O'shea-Zames-Falb multipliers for safety guarantees. In *arxiv: 2207.01363 [math.OC]*, 2022.

[20] Y. Ebihara, H. Waki, V. Magron, N. H. A. Mai, D. Peaucelle, and S. Tarbouriech. Stability analysis of recurrent neural networks by IQC with copositive mutipliers. In *Proc. of the 60th Conference on Decision and Control*, 2021.

[21] C. W. Scherer. Relaxations for robust linear matrix inequality problems with verifications for exactness. *SIAM Journal on Matrix Analysis and Applications*, Vol. 27, No. 2, pp. 365–395, 2005.

[22] I. Masubuchi and C. W. Scherer. A recursive algorithm of exactness verification of relaxations for robust SDPs. *Systems and Control Letters*, Vol. 58, No. 8, pp. 592–601, 2009.

[23] Y. Ebihara, Y. Onishi, and T. Hagiwara. Robust performance analysis of uncertain LTI systems: Dual LMI approach and verifications for exactness. *IEEE Transactions on Automatic Control*, Vol. 54, No. 5, pp. 938–951, 2009.

[24] C. W. Scherer. LMI relaxations in robust control. *European Journal of Control*, Vol. 12, No. 1, pp. 3–29, 2006.

[25] E. Klerk. *Aspects of Semidefinite Programming*. Kluwer Academic Publishers, 2002.

[26] https://www.mosek.com/.

[27] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proc. IEEE Computer Aided Control System Design*, pp. 284–289, 2004.