2020 59th IEEE Conference on Decision and Control (CDC)
Jeju Island, Republic of Korea, December 14-18, 2020

# Tightened Convex Relaxations for Neural Network Robustness Certification

Brendon G. Anderson, Ziye Ma, Jingqi Li, and Somayeh Sojoudi

*Abstract*— **In this paper, we consider the problem of certifying the robustness of neural networks to perturbed and adversarial input data. Such certification is imperative for the application of neural networks in safety-critical decision-making and control systems. Certification techniques using convex optimization have been proposed, but they often suffer from relaxation errors that void the certificate. Our work exploits the structure of ReLU networks to improve relaxation errors through a novel partition-based certification procedure. The proposed method is proven to tighten existing linear programming relaxations, and asymptotically achieves zero relaxation error as the partition is made finer. We develop a finite partition that attains zero relaxation error and use the result to derive a tractable partitioning scheme that minimizes the worst-case relaxation error. Experiments using real data show that the partitioning procedure is able to issue robustness certificates in cases where prior methods fail. Consequently, partition-based certification procedures are found to provide an intuitive, effective, and theoretically justified method for tightening existing convex relaxation techniques.**

## I. INTRODUCTION

Recent successes of neural networks can be found in nearly all forms of data-driven decision making problems. In particular, both classical and modern problems within control theory have been addressed using neural networks, e.g., control of nonlinear systems [1], [2], data-driven system identification [3]–[5], and adaptive and self-learning control [6], [7]. With their increasing prevalence, neural networks have begun to find applications in highly sensitive data-driven decision-making problems involving the control of safety-critical systems, such as autonomous vehicles [8], [9] and the power grid [10]–[12]. The common underlying principle among these systems is that decisions and control actions must be robust against fluctuations in the measurements or inputs to the decision making algorithm. As a result, much effort has been placed on developing methods to certify the robustness of neural networks to perturbations in their input data [13]–[20]. Due to the vast range of network architectures, their inherent nonconvexity, and computational burdens arising with large-scale networks, the development of efficient and reliable certification methods remains an ongoing effort.

A common deterministic certification procedure is to verify that all possible unknown inputs are mapped to outputs that the network operator classifies as safe [13], [19]. From this perspective, certification amounts to proving that the image of an input uncertainty set is contained within a prescribed safe set. Such a worst-case analysis is naturally formulated as a robust optimization (RO) problem [21], [22], however, approaching the problem from a general RO framework neglects the informative structure of the network architecture. Furthermore, even when the input uncertainty set is convex, its output set may be nonconvex, which renders the certification an NP-complete nonconvex optimization problem [16], [23]. To overcome these issues, researchers have proposed various relaxations to over-approximate the output set of the network by a convex one and perform the certification on the easier-to-analyze convex set. One of the simplest and most popular approximation classes is based on linear program (LP) relaxations [13].

In the case that a convex outer approximation of the original nonconvex output set is contained in the safe set, a certificate of robustness for the true network can be obtained. An immediate problem arises with these convex relaxations: if the convex outer approximation of the output set is too loose, the relaxation may not issue a certificate even in the case the true network is robust. To tighten the outer approximation, more sophisticated and computationally demanding convex relaxations have been proposed in the literature, such as the semidefinite programming and quadratically-constrained semidefinite programming techniques [14], [18].

### A. *Partition-Based Certification*

In this paper, we focus on feedforward ReLU networks, which are popular due to their simplicity, fast training speeds, and non-vanishing gradient property [15]. Our approach to certifying these networks is based on partitioning the input uncertainty set and solving simple linear programs over each input part. Partitioning heuristics have been applied in areas such as robust optimization [24] and deep learning [25], and are often found to tighten bounds on optimization objectives. Furthermore, partitioning naturally allows for parallelization of the optimization, resulting in computational advantages over centralized methods. Our proposed method is closely related to solving mixed integer programming (MIP) problems, as ReLU robustness certification can be expressed as an MIP. However, mixed integer reformulations of ReLU constraints introduce new integer variables for each neuron in the network, making MIP approaches, such as the outer approximation algorithm in [26], unnecessarily large in dimension. Instead, our partition-based approach directly encodes the integral nature of the ReLU constraints without adding extra variables.

Previous works that apply partitioning to network certification include [15], where the authors perform a reachability analysis for the safety verification of neural network controllers. However, that method is restricted to hyperrectangular partitions of both the input uncertainty set and the re-

The authors are with the University of California, Berkeley. Somayeh Sojoudi is also with the Tsinghua-Berkeley Shenzhen Institute. Emails: {bganderson, ziyema, jingqili, sojoudi}@berkeley.edu.

This work was supported by grants from AFOSR, ONR, and NSF.

978-1-7281-7447-1/20/$31.00 ©2020 IEEE

sulting outer approximations. The authors of [19] use duality arguments to propose a novel partitioning scheme; however, the designed algorithm only considers splitting box-shaped uncertainty sets in half along coordinate axes. Not only are the current partition-based methods too restrictive in their partition structure and accordingly produce unnecessarily loose outer approximations, but they also lack mathematical support for the effectiveness of the partitioning in tightening the relaxations.

### B. Contributions

In an effort to improve relaxation errors, we exploit the nature of ReLU networks to achieve the following goals:

1) Prove that partitioning tightens existing linear program relaxations, and define the notion of Lipschitz relaxations to show that relaxation error converges to zero as partitions become finer;
2) Show that an intelligently designed finite partition attains zero relaxation error, and use this insight to derive a computationally tractable partitioning scheme that minimizes worst-case relaxation error;
3) Demonstrate on real data that the optimal partitioning scheme sufficiently reduces relaxation error to certify robustness where prior methods fail.

The contributions of this paper culminate into a theoretically justified and empirically validated robustness certification procedure that combines simple and efficient linear program models, computational parallelizability, and optimal relaxation tightening.

### C. Organization

In Section II, we define some mathematical notations. Section III introduces the robustness certification problem and linear program relaxation. In Section IV, we introduce the notion of partitioning and analyze its properties when applied to the robustness certification of ReLU networks. In Section V, we further develop the theory to study the optimality of partitions and propose an optimal partitioning strategy. We provide illustrative examples in Section VI and conclude in Section VII.

## II. Notations

We write the sets of $n$-vectors and $m \times n$ matrices with real-valued elements as $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$, respectively. For $X, Y \in \mathbb{R}^{m\times n}$, we write $X \leq Y$ to mean $X_{ij} \leq Y_{ij}$ for all $i \in \{1,2,\dots,m\}$ and all $j \in \{1,2,\dots,n\}$. We write the Hadamard (element-wise) product between $X$ and $Y$ as $X \odot Y$ and the Hadamard division of $X$ by $Y$ as $X \oslash Y$. Furthermore, for $f\colon \mathbb{R} \to \mathbb{R}$, we define $f(X)$ to be an $m \times n$ matrix whose $(i,j)$ element is equal to $f(X_{ij})$ for all $i \in \{1,2,\dots,m\}$ and all $j \in \{1,2,\dots,n\}$. In particular, let the ReLU function be denoted as $\mathrm{ReLU}(\cdot) = \max\{0,\cdot\}$.

## III. Problem Statement

### A. Network Description

Consider a $K$-layer ReLU neural network defined by

$$\begin{aligned} x^{[0]} &= x, \\ \hat{z}^{[k]} &= W^{[k-1]}x^{[k-1]} + b^{[k-1]}, \\ x^{[k]} &= \mathrm{ReLU}(\hat{z}^{[k]}), \\ z &= x^{[K]}, \end{aligned} \tag{1}$$

for all $k \in \{1,2,\dots,K\}$, where $x \in \mathbb{R}^{n_x}$ is the input to the neural network, $z \in \mathbb{R}^{n_z}$ is the output, and $\hat{z}^{[k]} \in \mathbb{R}^{n_k}$ is the preactivation of the $k^{\text{th}}$ layer. The parameters $W^{[k]} \in \mathbb{R}^{n_{k+1}\times n_k}$ and $b^{[k]} \in \mathbb{R}^{n_{k+1}}$ are the weight matrix and bias vector applied to the $k^{\text{th}}$ layer's activation $x^{[k]} \in \mathbb{R}^{n_k}$, respectively. Without loss of generality, assume that the bias terms are accounted for in the activations $x^{[k]}$, thereby setting $b^{[k]} = 0$ for all layers $k$. Let the function $f\colon \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ denote the map $x \mapsto z$ defined by (1).

### B. Input Uncertainty, Relaxed Network Constraint, and Safe Sets

We consider the scenario in which the network inputs are unknown but contained in a compact set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$. We call $\mathcal{X}$ the *input uncertainty set*, which is assumed to be a convex polytope. In the literature of neural network robustness certification, the input uncertainty set is commonly modeled as $\mathcal{X} = \{x \in \mathbb{R}^{n_x} : \|x - \bar{x}\|_\infty \leq \epsilon\}$, where $\bar{x} \in \mathbb{R}^{n_x}$ is a nominal input to the network and $\epsilon > 0$ [13], [14].

The bounds on the input, as defined by $\mathcal{X}$, implicitly define bounds on the preactivation at each layer. That is, $x \in \mathcal{X}$ implies that there exist bounds $l^{[k]}, u^{[k]} \in \mathbb{R}^{n_k}$ such that $l^{[k]} \leq \hat{z}^{[k]} \leq u^{[k]}$ for all $k \in \{1,2,\dots,K\}$. Although one can create an outer approximation of these bounds, we consider the true bounds $l^{[k]}$ and $u^{[k]}$ as tight, i.e., $z^{[k]} = l^{[k]}$ for some $x \in \mathcal{X}$ and similarly for the upper bound $u^{[k]}$. From these bounds, we relax the $k^{\text{th}}$ ReLU constraint in (1) to its convex envelope, which leads to a relaxed ReLU constraint set associated with the $k^{\text{th}}$ layer:

$$\begin{aligned} \mathcal{N}^{[k]} = \{&(x^{[k-1]}, x^{[k]}) \in \mathbb{R}^{n_{k-1}} \times \mathbb{R}^{n_k} : \\ & x^{[k]} \leq u^{[k]} \odot (\hat{z}^{[k]} - l^{[k]}) \oslash (u^{[k]} - l^{[k]}), \\ & x^{[k]} \geq 0,\ x^{[k]} \geq \hat{z}^{[k]},\ \hat{z}^{[k]} = W^{[k-1]}x^{[k-1]}\}. \end{aligned} \tag{2}$$

Define the *relaxed network constraint set* as

$$\begin{aligned} \mathcal{N} = \{&(x,z) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} : (x, x^{[1]}) \in \mathcal{N}^{[1]}, \\ &(x^{[1]}, x^{[2]}) \in \mathcal{N}^{[2]}, \dots, (x^{[K-1]}, z) \in \mathcal{N}^{[K]}\}. \end{aligned} \tag{3}$$

In essence, $\mathcal{N}$ is the set of all feasible input-output pairs of the network satisfying the relaxed ReLU constraint at each layer. Since the bounds $l^{[k]}$ and $u^{[k]}$ are determined by the input uncertainty set $\mathcal{X}$, the set $\mathcal{N}^{[k]}$ is also determined by $\mathcal{X}$ for all layers $k$.

*Remark* 1. In the context of one-layer networks (i.e., $K = 1$), the single relaxed ReLU constraint set coincides with the relaxed network constraint set: $\mathcal{N}^{[1]} = \mathcal{N}$. Therefore, for $K = 1$ we drop the $k$-notation from $z$, $\hat{z}$, $x$, $W$, $l$, $u$, and $\mathcal{N}$. A visualization of $\mathcal{N}$ is given in Fig. 1 for this case.

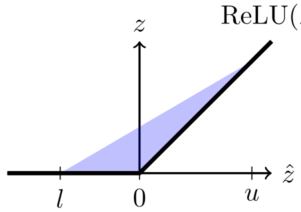



Fig. 1. Relaxed ReLU constraint set at a single neuron for a one-layer network. The convex envelope $\mathcal{N}$ is shaded.

*Remark* 2. Consider a one-layer ReLU constraint relaxed according to (2). Suppose that $l < u < 0$. A simple calculation shows that $\mathcal{N} = \{(x,0) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} : Wx = l\}$ due to the inequalities $l \leq Wx \leq u$. That is, the set of input-output pairs that are feasible for the relaxed network constraints exclude many possible inputs that are feasible for the input uncertainty constraints. The same problem occurs when $0 < l < u$. To overcome this issue, we impose the conditions that $l \leq 0 \leq u$ and $l < u$ so that the certification procedure considers all possible inputs in $\mathcal{X}$.

Now, consider a set $\mathcal{S} \subseteq \mathbb{R}^{n_z}$, termed the *safe set*. As is common in the adversarial machine learning literature, we consider (possibly unbounded) polyhedral safe sets defined as the intersection of a finite number of half-spaces: $\mathcal{S} = \{z \in \mathbb{R}^{n_z} : Cz \leq d\}$, where $C \in \mathbb{R}^{n_{\mathcal{S}} \times n_z}$ and $d \in \mathbb{R}^{n_{\mathcal{S}}}$ are given. An output $z \in \mathcal{S}$ is said to be *safe*.

### C. Robustness Certification

The goal is to certify that all inputs in $\mathcal{X}$ map to safe outputs in $\mathcal{S}$. If this is successfully accomplished, the network is said to be *certifiably robust*. Formally, this certificate is written as $f(\mathcal{X}) \subseteq \mathcal{S}$, or equivalently

$$\sup_{x \in \mathcal{X}} c_i^\top f(x) \leq d_i \text{ for all } i \in \{1, 2, \dots, n_{\mathcal{S}}\},$$

where $c_i^\top$ is the $i^{\text{th}}$ row of $C$. Thus, the certification procedure amounts to solving an optimization problem corresponding to each $c_i$. In the sequel, we focus on a single optimization problem, namely $\sup_{x \in \mathcal{X}} c^\top f(x)$, since the generalization to the case $n_{\mathcal{S}} > 1$ is straightforward. With no loss of generality, assume that $d = 0$ (if $d \neq 0$, one can first solve the optimization for $d = 0$ and then shift the corresponding result). Note that the proposed mathematical framework encapsulates the popular certification that a classification network will not misclassify any adversarial inputs within a bounded uncertainty set.

In general, the optimization $\sup_{x \in \mathcal{X}} c^\top f(x)$ is a nonconvex problem and $f(\mathcal{X})$ is a nonconvex set, and therefore computing a robustness certificate is intractable. To circumvent this issue, one can instead certify that a convex outer approximation of $f(\mathcal{X})$ is safe, as this inherently certifies the safety of the true nonconvex set $f(\mathcal{X})$, and hence certifies the robustness of the network. This process is illustrated in Fig. 2.

The robustness certification problem can be written as

$$f^*(\mathcal{X}) = \sup\{c^\top z : z = f(x),\ x \in \mathcal{X}\}. \tag{4}$$

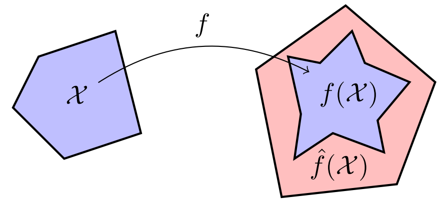



Fig. 2. The convex outer approximation of the nonconvex set $f(\mathcal{X})$ is $\hat{f}(\mathcal{X})$. If the outer approximation is safe, i.e., $\hat{f}(\mathcal{X}) \subseteq \mathcal{S}$, then so is $f(\mathcal{X})$.

The nonconvexity of (4) comes from the nonlinear equality constraint $z = f(x)$. Note that for all $x \in \mathcal{X}$, the equality $z = f(x)$ implies that $(x, z) \in \mathcal{N}$. Therefore, to avoid the nonconvex equality constraint, one can use the relaxed network constraint set to solve the following surrogate LP relaxation [13]:

$$\hat{f}^*(\mathcal{X}) = \sup\{c^\top z : (x,z) \in \mathcal{N},\ x \in \mathcal{X}\}. \tag{5}$$

The suprema in (4) and (5) are assumed to be attained.

Due to the relaxation introduced in (5), it holds that

$$f^*(\mathcal{X}) \leq \hat{f}^*(\mathcal{X}). \tag{6}$$

Therefore, a sufficient condition for the network to be certifiably robust is that $\hat{f}^*(\mathcal{X}) \leq 0$. In the case $\hat{f}^*(\mathcal{X}) > 0$, the relaxation cannot certify whether or not the true network is robust, since it may still hold that $f^*(\mathcal{X}) \leq 0$. The remainder of this paper is dedicated to optimally tightening the bound (6) while maintaining the advantageous convexity and computational properties of the LP relaxation.

## IV. Properties of Partitioned Relaxations

In this section, we investigate the notion of input partitioning and rigorously derive guarantees on the effectiveness of partitioned relaxations for the robustness certification problem. We will first show that by partitioning the input uncertainty set and solving separate LP relaxations over each part, a useful upper bound for the unrelaxed problem (4) can be obtained. In particular, the partitioning method yields a valid relaxation of (4).

### A. Validation of Partitioned Relaxations

**Definition 1** (Partition)**.** The collection $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \{1, 2, \dots, p\}\}$ is said to be a *partition* of the input uncertainty set $\mathcal{X}$ if $\mathcal{X} = \cup_{j=1}^p \mathcal{X}^{(j)}$ and $\mathcal{X}^{(j)} \cap \mathcal{X}^{(k)} = \emptyset$ for all $j \neq k$. The set $\mathcal{X}^{(j)}$ is called the $j^{\text{th}}$ *input part*.

**Proposition 1** (Partitioned relaxation bound)**.** *Let $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \{1, 2, \dots, p\}\}$ be a partition of $\mathcal{X}$. Then, it holds that*

$$f^*(\mathcal{X}) \leq \max_{j \in \{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)}). \tag{7}$$

*Proof.* Assume that $f^*(\mathcal{X}) > \max_{j \in \{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)})$. Then,

$$f^*(\mathcal{X}) > \hat{f}^*(\mathcal{X}^{(j)}) \text{ for all } j \in \{1, 2, \dots, p\}. \tag{8}$$

Let $(x^*, z^*)$ denote an optimal solution to the unrelaxed problem (4), i.e., $x^* \in \mathcal{X}$, $z^* = f(x^*)$, and

$$c^\top z^* = f^*(\mathcal{X}). \tag{9}$$

Since $\cup_{j=1}^{p}\mathcal{X}^{(j)} = \mathcal{X}$, there exists $j^* \in \{1,2,\dots,p\}$ such that $x^* \in \mathcal{X}^{(j^*)}$. Since $x^* \in \mathcal{X}^{(j^*)}$ and $z^* = f(x^*)$, it holds that $(x^*, z^*) \in \mathcal{N}^{(j^*)}$, where $\mathcal{N}^{(j^*)}$ is the relaxed network constraint set defined by $\mathcal{X}^{(j^*)}$. Therefore,

$$\begin{aligned}c^\top z^* \leq \sup\{c^\top z : x \in \mathcal{X}^{(j^*)},\ (x,z) \in \mathcal{N}^{(j^*)}\}\\ = \hat{f}^*(\mathcal{X}^{(j^*)}) < f^*(\mathcal{X}),\end{aligned}$$

where the first inequality comes from the feasibility of $(x^*, z^*)$ over the $j^{*\text{th}}$ subproblem and the final inequality is due to (8). This contradicts the optimality of $(x^*, z^*)$ given in (9). Hence, (7) must hold. □

### B. Tightening of the LP Relaxation

The objective is to show that by partitioning the input uncertainty set, the linear program relaxation bound in (6) is improved. The result will be presented for one-layer networks for simplicity, but the conclusion naturally generalizes to multi-layer networks.

**Proposition 2** (Improving the relaxation bound)**.** *Consider a one-layer feedforward neural network. Let $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \{1,2,\dots,p\}\}$ be a partition of $\mathcal{X}$. For the $j^{th}$ input part $\mathcal{X}^{(j)}$, denote the corresponding preactivation bounds by $l^{(j)}$ and $u^{(j)}$, where $l \leq l^{(j)} \leq Wx \leq u^{(j)} \leq u$ for all $x \in \mathcal{X}^{(j)}$. Then, it holds that*

$$\max_{j\in\{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)}) \leq \hat{f}^*(\mathcal{X}). \tag{10}$$

*Proof.* Let $j \in \{1,2,\dots,p\}$. It will be shown that $\mathcal{N}^{(j)} \subseteq \mathcal{N}$. Let $(x,z) \in \mathcal{N}^{(j)}$. Define $u' = u^{(j)}$, $l' = l^{(j)}$, and

$$\begin{aligned}g(x) &= u \odot (Wx - l) \oslash (u - l),\\ g'(x) &= u' \odot (Wx - l') \oslash (u' - l').\end{aligned}$$

Then, by letting $\Delta g(x) = g(x) - g'(x) = a \odot (Wx) + b$, where

$$\begin{aligned}a &= u \oslash (u - l) - u' \oslash (u' - l'),\\ b &= u' \odot l' \oslash (u' - l') - u \odot l \oslash (u - l),\end{aligned}$$

the following relations are derived for all $i \in \{1,2,\dots,n_z\}$:

$$\begin{aligned}g_i^* &:= \inf_{\{x: l' \leq Wx \leq u'\}} (\Delta g(x))_i \geq \inf_{\{\hat{z}: l' \leq \hat{z} \leq u'\}} (a \odot \hat{z} + b)_i\\ &= \inf_{\{\hat{z}_i: l'_i \leq \hat{z}_i \leq u'_i\}} (a_i \hat{z}_i + b_i) = \begin{cases} a_i l'_i + b_i & \text{if } a_i \geq 0,\\ a_i u'_i + b_i & \text{if } a_i < 0.\end{cases}\end{aligned}$$

In the case that $a_i \geq 0$, we have that

$$g_i^* \geq a_i l'_i + b_i = \frac{u_i}{u_i - l_i}(l'_i - l_i) \geq 0,$$

where the final inequality comes from the fact that $u \geq 0$, $l' \geq l$, and $u > l$. On the other hand, if $a_i < 0$, it holds that

$$g_i^* \geq a_i u'_i + b_i = \frac{u'_i - u_i}{u_i - l_i} l_i \geq 0,$$

where the final inequality comes from the fact that $u' \leq u$, $l \leq 0$, and $u > l$. Therefore, $g^* = (g_1^*, g_2^*, \dots, g_{n_z}^*) \geq 0$, which implies that $\Delta g(x) = g(x) - g'(x) \geq 0$ for all $x$ such that $l^{(j)} = l' \leq Wx \leq u' = u^{(j)}$. Hence, since $(x,z) \in \mathcal{N}^{(j)}$, it holds that $z \geq 0$, $z \geq Wx$, and

$$z \leq g'(x) \leq g(x) = u \odot (Wx - l) \oslash (u - l).$$

Therefore, we have that $(x,z) \in \mathcal{N}$.

Since $\mathcal{X}^{(j)} \subseteq \mathcal{X}$ (by definition) and $\mathcal{N}^{(j)} \subseteq \mathcal{N}$, it holds that the solution to the problem over the smaller feasible set gives a lower bound to the original solution: $\hat{f}^*(\mathcal{X}^{(j)}) \leq \hat{f}^*(\mathcal{X})$. Finally, since $j$ was chosen arbitrarily, this implies the desired inequality (10). □

### C. Asymptotic Exactness of Partitioned Relaxations

In this section, we define the notion of Lipschitz continuity of a relaxation. We use this property to show that, under appropriate conditions, LP relaxations asymptotically approach the true problem as the partition becomes finer.

**Definition 2** ($L$-Lipschitz relaxation)**.** A neural network $f$ is said to have an *$L$-Lipschitz continuous relaxation* (with respect to $\mathcal{N}$ on $\mathcal{X}$) if there exists a finite constant $L \in \mathbb{R}$ such that

$$|c^\top z_1^* - c^\top z_2^*| \leq L\|x_1 - x_2\|_2 \text{ for all } x_1, x_2 \in \mathcal{X},$$

where $c^\top z_i^* = \sup\{c^\top z : (x_i, z) \in \mathcal{N}\}$ for all $i \in \{1,2\}$.

*Remark* 3. In the case the relaxed network constraint set is exact (i.e., $(x^{[k-1]}, x^{[k]}) \in \mathcal{N}^{[k]}$ if and only if $x^{[k]} = \mathrm{ReLU}(W^{[k-1]}x^{[k-1]})$ for all layers $k \in \{1,2,\dots,K\}$), the relation $x_i \in \mathcal{X}$ implies that $c^\top z_i^* = c^\top f(x_i)$, and so the $L$-Lipschitz continuity of the relaxation reduces to the classical $L$-Lipschitz continuity of the function $c^\top f$ over the set $\mathcal{X}$.

**Lemma 1** (Lipschitz LP relaxations)**.** *All $K$-layer neural networks defined by (1) have $L$-Lipschitz continuous LP relaxations.*

*Proof.* Let $x_1, x_2 \in \mathcal{X}$. By Theorem 2.4 in [27], there exists a finite constant $\beta \in \mathbb{R}$ such that for all $z_1^* \in \arg\max\{c^\top z : (x_1, z) \in \mathcal{N}\}$ there exists $z_2^* \in \arg\max\{c^\top z : (x_2, z) \in \mathcal{N}\}$ satisfying

$$\|z_1^* - z_2^*\|_\infty \leq \beta\|x_1 - x_2\|_2.$$

By the Cauchy-Schwarz inequality, this yields that

$$|c^\top(z_1^* - z_2^*)| \leq \|c\|_1\|z_1^* - z_2^*\|_\infty \leq \beta\|c\|_1\|x_1 - x_2\|_2.$$

Defining $L = \beta\|c\|_1$ completes the proof. □

Lemma 1 shows that a partitioned LP relaxation is a Lipschitz relaxation over any chosen input part. This property will be used to derive a bound on the difference between the partitioned LP relaxation and the unrelaxed problem for one-layer networks based on the diameters of the input parts.

**Definition 3** (Diameter)**.** For a set $\mathcal{X} \subseteq \mathbb{R}^n$, the *diameter* of $\mathcal{X}$ is defined as $d(\mathcal{X}) = \sup\{\|x - y\|_2 : x, y \in \mathcal{X}\}$.

**Proposition 3** (Diameter bound)**.** *Consider a one-layer feedforward neural network over the input uncertainty set $\mathcal{X}$ and the relaxed network constraint set $\mathcal{N}$. Let $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \{1,2,\dots,p\}\}$ be a partition of $\mathcal{X}$. Denote the largest diameter among the input parts by $d^*$, i.e., $d^* =$*

$\max\{d(\mathcal{X}^{(j)}) : j \in \{1,2,\dots,p\}\}$. *Then, there exists a finite constant* $L \in \mathbb{R}$ *such that*

$$\left| f^*(\mathcal{X}) - \max_{j\in\{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)}) \right| \leq Ld^*. \tag{11}$$

*Proof.* First, let $j^*$ be the index corresponding to the partition subproblem with the highest objective value: $\hat{f}^*(\mathcal{X}^{(j^*)}) = \max_{j\in\{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)})$. By Lemma 1, the network has an $L$-Lipschitz relaxation with respect to $\mathcal{N}^{(j^*)}$ on $\mathcal{X}^{(j^*)}$. Thus, there exists a finite constant $L \in \mathbb{R}$ such that

$$L \geq \frac{|c^\top z_1^* - c^\top z_2^*|}{\|x_1 - x_2\|_2}$$

for all $x_1, x_2 \in \mathcal{X}^{(j^*)}$, where $c^\top z_1^* = \sup\{c^\top z : (x_1, z) \in \mathcal{N}^{(j^*)}\}$ and $c^\top z_2^* = \sup\{c^\top z : (x_2, z) \in \mathcal{N}^{(j^*)}\}$. Furthermore, by the definition of $d^*$ and $j^*$, we have that

$$d^* \geq d(\mathcal{X}^{(j^*)}) \geq \|x_1 - x_2\|_2 \text{ for all } x_1, x_2 \in \mathcal{X}^{(j^*)}.$$

Let $\bar{x} \in \mathcal{X}^{(j^*)}$ be such that $W\bar{x} = l^{(j^*)}$ and $\bar{z} = \mathrm{ReLU}(W\bar{x})$, so that $c^\top \bar{z} = \sup\{c^\top z : (\bar{x}, z) \in \mathcal{N}^{(j^*)}\}$. Furthermore, let $(\tilde{x}^*, \tilde{z}^*)$ denote a solution to the relaxed problem (5) over the $j^{*\text{th}}$ input part, i.e., corresponding to $\hat{f}^*(\mathcal{X}^{(j^*)})$. Then, since $\bar{x}, \tilde{x}^* \in \mathcal{X}^{(j^*)}$ and $c^\top \tilde{z}^* = \sup\{c^\top z : (\tilde{x}^*, z) \in \mathcal{N}^{(j^*)}\}$, it holds that

$$Ld^* \geq \frac{|c^\top \bar{z} - c^\top \tilde{z}^*|}{\|\bar{x} - \tilde{x}^*\|_2} \|\bar{x} - \tilde{x}^*\|_2 = |c^\top \bar{z} - \hat{f}^*(\mathcal{X}^{(j^*)})|.$$

Now, since $(\bar{x}, \bar{z})$ is feasible for the unrelaxed problem (4) and the relaxation $\hat{f}^*(\mathcal{X}^{(j^*)})$ provides an upper bound on the unrelaxed problem by Proposition 1, it holds that

$$c^\top \bar{z} \leq f^*(\mathcal{X}) \leq \hat{f}^*(\mathcal{X}^{(j^*)}).$$

This implies that $|f^*(\mathcal{X}) - \hat{f}^*(\mathcal{X}^{(j^*)})| \leq |c^\top \bar{z} - \hat{f}^*(\mathcal{X}^{(j^*)})|$. Therefore, $Ld^* \geq |f^*(\mathcal{X}) - \hat{f}^*(\mathcal{X}^{(j^*)})|$, proving (11), as desired. $\square$

In the case that the network has a Lipschitz relaxation that is uniform over all possible input parts, Proposition 3 shows that as the partition becomes finer, namely $d^* \to 0$, the gap between the partitioned relaxation and the true solution converges to zero. As a result, partitioned relaxations are asymptotically exact.

## V. Optimal Partitioning

In this section, we construct a partition with a finite number of input parts under which LP relaxations exactly recover the nonconvex robustness certification problem. Motivated by this optimal partition, we develop a simple and computationally tractable partitioning procedure that minimizes the worst-case relaxation error.

### A. Exact Partitioned Relaxation

The goal is to show that by meticulously selecting the partition of the input uncertainty set based on the rows of the weight matrix $W$, the relaxation introduced by the resulting linear programs becomes exact.

**Proposition 4** (Motivating partition)**.** *Consider a one-layer feedforward neural network and denote the* $i^{th}$ *row of* $W$ *by* $w_i^\top \in \mathbb{R}^{1\times n_x}$ *for all* $i \in \{1,2,\dots,n_z\}$. *Define* $\mathcal{J} = \{0,1\}^{n_z}$ *and take the partition of* $\mathcal{X}$ *to be indexed by* $\mathcal{J}$. *That is,* $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \mathcal{J}\}$, *where for a given* $j \in \mathcal{J}$ *we define*

$$\begin{aligned}\mathcal{X}^{(j)} = \{x \in \mathcal{X} : w_i^\top x \geq 0 \text{ for all } i \text{ such that } j_i = 1,\\ w_i^\top x < 0 \text{ for all } i \text{ such that } j_i = 0\}.\end{aligned} \tag{12}$$

*Then, the partitioned relaxation is exact, i.e.,*

$$f^*(\mathcal{X}) = \max_{j\in\mathcal{J}} \hat{f}^*(\mathcal{X}^{(j)}). \tag{13}$$

*Proof.* We first show that $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \mathcal{J}\}$ is a valid partition. Since $\mathcal{X}^{(j)} \subseteq \mathcal{X}$ for all $j \in \mathcal{J}$, the relation $\cup_{j\in\mathcal{J}} \mathcal{X}^{(j)} \subseteq \mathcal{X}$ is satisfied. Now, suppose that $x \in \mathcal{X}$. Then, for all $i \in \{1,2,\dots,n_z\}$, it holds that either $w_i^\top x \geq 0$ or $w_i^\top x < 0$. Define $j \in \{0,1\}^{n_z}$ as follows:

$$j_i = \begin{cases} 1 & \text{if } w_i^\top x \geq 0, \\ 0 & \text{if } w_i^\top x < 0, \end{cases}$$

for all $i \in \{1,2,\dots,n_z\}$. Then, by the definition of $\mathcal{X}^{(j)}$ in (12), it holds that $x \in \mathcal{X}^{(j)}$. Therefore, the relation $x \in \mathcal{X}$ implies that $x \in \mathcal{X}^{(j)}$ for some $j \in \{0,1\}^{n_z} = \mathcal{J}$. Hence, $\mathcal{X} \subseteq \cup_{j\in\mathcal{J}} \mathcal{X}^{(j)}$, and therefore $\cup_{j\in\mathcal{J}} \mathcal{X}^{(j)} = \mathcal{X}$.

We now show that $\mathcal{X}^{(j)} \cap \mathcal{X}^{(k)} = \emptyset$ for all $j \neq k$. Let $j, k \in \mathcal{J}$ with the property that $j \neq k$. Then there exists $i \in \{1,2,\dots,n_z\}$ such that $j_i \neq k_i$. Let $x \in \mathcal{X}^{(j)}$. In the case that $w_i^\top x \geq 0$, it holds that $j_i = 1$ and therefore $k_i = 0$. Hence, for all $y \in \mathcal{X}^{(k)}$, it holds that $w_i^\top y < 0$, and therefore $x \notin \mathcal{X}^{(k)}$. An analogous reasoning shows that $x \notin \mathcal{X}^{(k)}$ when $w_i^\top x < 0$. Therefore, one concludes that $x \in \mathcal{X}^{(j)}$ and $j \neq k$ implies that $x \notin \mathcal{X}^{(k)}$, i.e., that $\mathcal{X}^{(j)} \cap \mathcal{X}^{(k)} = \emptyset$. Hence, $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \mathcal{J}\}$ is a valid partition.

We now prove (13). Let $j \in \mathcal{J}$. Since $w_i^\top x \geq 0$ for all $i$ such that $j_i = 1$, the preactivation lower bound becomes $l_i^{(j)} = 0$ for all such $i$. On the other hand, since $w_i^\top x < 0$ for all $i$ such that $j_i = 0$, the preactivation upper bound becomes $u_i^{(j)} = 0$ for all such $i$. Therefore, the relaxed network constraint set (3) for the $j^{\text{th}}$ input part reduces to

$$\begin{aligned}\mathcal{N}^{(j)} = \{(x,z) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} :&\\ z_i = 0 \text{ for all } i \text{ such that } j_i = 0,&\\ z_i = w_i^\top x = (Wx)_i \text{ for all } i \text{ such that } j_i = 1\}.&\end{aligned}$$

That is, the relaxed ReLU constraint envelope collapses to the exact ReLU constraint through the *a priori* knowledge of each preactivation coordinate's sign. Therefore, we find that for all $x \in \mathcal{X}^{(j)}$ it holds that $(x,z) \in \mathcal{N}^{(j)}$ if and only if $z = \mathrm{ReLU}(Wx)$. Hence, the LP over the $j^{\text{th}}$ input part yields that

$$\begin{aligned}\hat{f}^*(\mathcal{X}^{(j)}) &= \sup\{c^\top z : (x,z) \in \mathcal{N}^{(j)},\ x \in \mathcal{X}^{(j)}\}\\ &= \sup\{c^\top z : z = \mathrm{ReLU}(Wx),\ x \in \mathcal{X}^{(j)}\}\\ &\leq \sup\{c^\top z : z = \mathrm{ReLU}(Wx),\ x \in \mathcal{X}\}\\ &= f^*(\mathcal{X}).\end{aligned}$$

Therefore, $\max_{j\in\mathcal{J}} \hat{f}^*(\mathcal{X}^{(j)}) \leq f^*(\mathcal{X})$, since $j$ was chosen arbitrarily. Since $f^*(\mathcal{X}) \leq \max_{j\in\mathcal{J}} \hat{f}^*(\mathcal{X}^{(j)})$ by the relaxation bound (7), the equality (13) holds, as desired. □

The partition introduced in Proposition 4 requires solving $2^{n_z}$ linear programs, which may quickly become computationally intractable in practice. Despite this limitation, the result provides two major theoretical implications. First, it shows that, using the input partitioning methodology presented in this paper, the robustness certification problem can be solved exactly via a finite number of linear program subproblems. Second, Proposition 4 provides a starting point to answer the following question: If the input uncertainty set is to be partitioned into only two parts, what is the optimal partition to choose? The proposition gives insight into the structure of an optimal partition, namely that it is defined by intersections of the half-spaces generated by the rows of $W$ (see Fig. 3). Motivated by this structure, we develop an optimal two-part partitioning scheme in the next section.

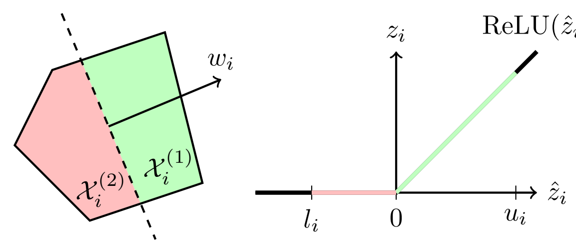



Fig. 3. Partitioning based on row $w_i^\top$ of the weight matrix. This partition results in an exact ReLU constraint in coordinate $i$ over the two resulting input parts $\mathcal{X}_i^{(1)} = \{x \in \mathcal{X} : w_i^\top x \geq 0\}$ and $\mathcal{X}_i^{(2)} = \mathcal{X} \setminus \mathcal{X}_i^{(1)}$.

### B. Optimal Partitioning Scheme

To derive an optimal partitioning scheme, we first bound the relaxation error in the worst-case sense.

**Theorem 1** (Worst-case relaxation bound)**.** *Consider a one-layer feedforward neural network with the input uncertainty set $\mathcal{X}$ and preactivation bounds $l, u \in \mathbb{R}^{n_z}$. Consider also the relaxation error $\Delta f^*(\mathcal{X}) := \hat{f}^*(\mathcal{X}) - f^*(\mathcal{X})$. Assume that there exists $x^* \in \mathcal{X}$ such that $(x^*, \tilde{z}^*)$ and $(x^*, z^*)$ are optimal solutions for the relaxation $\hat{f}^*(\mathcal{X})$ and the unrelaxed problem $f^*(\mathcal{X})$, respectively. Then, it holds that*

$$\Delta f^*(\mathcal{X}) \leq -\sum_{i=1}^{n_z} \mathrm{ReLU}(c_i)\frac{u_i l_i}{u_i - l_i}. \tag{14}$$

*Proof.* The definitions of $(x^*, \tilde{z}^*)$ and $(x^*, z^*)$ give that

$$\Delta f^*(\mathcal{X}) = \sum_{i=1}^{n_z} c_i(\tilde{z}_i^* - z_i^*) \leq \sum_{i=1}^{n_z} \Delta f_i^*, \tag{15}$$

where

$$\begin{aligned}\Delta f_i^* = \sup\{&c_i(\tilde{z}_i - z_i) : z_i = \mathrm{ReLU}(w_i^\top x),\ \tilde{z}_i \geq 0,\\ &\tilde{z}_i \geq w_i^\top x,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(w_i^\top x - l_i),\ x \in \mathcal{X}\}\end{aligned}$$

for all $i \in \{1, 2, \ldots, n_z\}$. For $c_i \geq 0$, it holds that

$$\begin{aligned}\Delta f_i^* &= c_i \sup\{\tilde{z}_i - z_i : z_i = \mathrm{ReLU}(w_i^\top x),\ \tilde{z}_i \geq 0,\\ &\qquad \tilde{z}_i \geq w_i^\top x,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(w_i^\top x - l_i),\ x \in \mathcal{X}\}\\ &= -c_i\frac{u_i l_i}{u_i - l_i},\end{aligned}$$

where the final equality is readily shown by computing the maximum difference between the line $\tilde{z}_i = \frac{u_i}{u_i - l_i}(\hat{z}_i - l_i)$ and the function $z_i = \mathrm{ReLU}(\hat{z}_i)$ over $\hat{z}_i \in [l_i, u_i]$. On the other hand, for $c_i < 0$, we have that

$$\begin{aligned}\Delta f_i^* &= c_i \inf\{\tilde{z}_i - z_i : z_i = \mathrm{ReLU}(w_i^\top x),\ \tilde{z}_i \geq 0,\\ &\qquad \tilde{z}_i \geq w_i^\top x,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(w_i^\top x - l_i),\ x \in \mathcal{X}\}\\ &= 0,\end{aligned}$$

where the final equality is due to the fact that $\tilde{z}_i \geq z_i$ on the above feasible set and $\tilde{z}_i = z_i = 0$ is feasible. Substituting these results into (15) gives the desired bound (14). □

The value $\Delta f_i^*$ can be interpreted as the worst-case relaxation error in coordinate $i$. From this perspective, Theorem 1 provides an upper bound on the overall worst-case relaxation error. The error bound scales linearly as the input uncertainty set is made smaller, as shown in Corollary 1 that follows.

**Corollary 1** (Linear scaling of relaxation error)**.** *Under the settings of Theorem 1, consider an input uncertainty subset $\tilde{\mathcal{X}} \subseteq \mathcal{X}$ such that its associated preactivation bounds are scaled inward by a factor of $\alpha \in (0, 1)$, namely $\tilde{u} = \alpha u$ and $\tilde{l} = \alpha l$. Then, the worst-case relaxation bound over $\tilde{\mathcal{X}}$ also scales by $\alpha$, i.e.,*

$$\hat{f}^*(\tilde{\mathcal{X}}) - f^*(\mathcal{X}) \leq -\alpha \sum_{i=1}^{n_z} \mathrm{ReLU}(c_i)\frac{u_i l_i}{u_i - l_i}. \tag{16}$$

*Proof.* Since $\tilde{\mathcal{X}} \subseteq \mathcal{X}$, it holds that $f^*(\tilde{\mathcal{X}}) \leq f^*(\mathcal{X})$. Therefore, by Theorem 1 on $\tilde{\mathcal{X}}$ we have that

$$\begin{aligned}\hat{f}^*(\tilde{\mathcal{X}}) - f^*(\mathcal{X}) &\leq \hat{f}^*(\tilde{\mathcal{X}}) - f^*(\tilde{\mathcal{X}})\\ &\leq -\sum_{i=1}^{n_z} \mathrm{ReLU}(c_i)\frac{\tilde{u}_i \tilde{l}_i}{\tilde{u}_i - \tilde{l}_i}.\end{aligned}$$

Substituting $\tilde{u}_i = \alpha u_i$ and $\tilde{l}_i = \alpha l_i$ completes the proof. □

We now focus on developing an optimal two-part partitioning scheme based on the worst-case relaxation bound of Theorem 1. We start with the following lemma.

**Lemma 2** (Two-part partition bound)**.** *Under the settings of Theorem 1, let $i \in \{1, 2, \ldots, n_z\}$ and consider a two-part partition of $\mathcal{X}$ given by $\{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}\}$, where $\mathcal{X}_i^{(1)} = \{x \in \mathcal{X} : w_i^\top x \geq 0\}$ and $\mathcal{X}_i^{(2)} = \mathcal{X} \setminus \mathcal{X}_i^{(1)}$. Consider also the partitioned relaxation error $\Delta f^*(\{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}\}) := \max_{j\in\{1,2\}} \hat{f}^*(\mathcal{X}_i^{(j)}) - f^*(\mathcal{X})$. It holds that*

$$\Delta f^*(\{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}\}) \leq -\sum_{\substack{k=1\\k\neq i}}^{n_z} \mathrm{ReLU}(c_k)\frac{u_k l_k}{u_k - l_k}. \tag{17}$$

*Proof.* Consider the relaxation solved over the first input part, $\mathcal{X}_i^{(1)}$, and denote by $l^{(1)}, u^{(1)} \in \mathbb{R}^{n_z}$ the corresponding preactivation bounds. Since $w_i^\top x \geq 0$ on this input part, the preactivation bounds for the first subproblem $\hat{f}^*(\mathcal{X}_i^{(1)})$ can be taken as $l^{(1)} = (l_1, l_2, \dots, l_{i-1}, 0, l_{i+1}, \dots, l_{n_z})$ and $u^{(1)} = u$. It follows from Theorem 1 that

$$\begin{aligned}\hat{f}^*(\mathcal{X}_i^{(1)}) - f^*(\mathcal{X}_i^{(1)}) &\leq -\sum_{k=1}^{n_z} \mathrm{ReLU}(c_k) \frac{u_k^{(1)} l_k^{(1)}}{u_k^{(1)} - l_k^{(1)}} \\ &= -\sum_{\substack{k=1 \\ k \neq i}}^{n_z} \mathrm{ReLU}(c_k) \frac{u_k l_k}{u_k - l_k}.\end{aligned}$$

Similarly, over the second input part, $\mathcal{X}_i^{(2)}$, we have that $w_i^\top x < 0$, and so the preactivation bounds for the second subproblem $\hat{f}^*(\mathcal{X}_i^{(2)})$ can be taken as $l^{(2)} = l$ and $u^{(2)} = (u_1, u_2, \dots, u_{i-1}, 0, u_{i+1}, \dots, u_{n_z})$, resulting in the same bound: $\hat{f}^*(\mathcal{X}_i^{(2)}) - f^*(\mathcal{X}_i^{(2)}) \leq -\sum_{\substack{k=1 \\ k\neq i}}^{n_z} \mathrm{ReLU}(c_k) \frac{u_k l_k}{u_k - l_k}$. Putting these two bounds together and using the fact that $f^*(\mathcal{X}_i^{(j)}) \leq f^*(\mathcal{X})$ for all $j \in \{1, 2\}$, we find that

$$\begin{aligned}\Delta f^*(\{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}\}) &= \max_{j \in \{1,2\}} \left(\hat{f}^*(\mathcal{X}_i^{(j)}) - f^*(\mathcal{X})\right) \\ &\leq \max_{j \in \{1,2\}} \left(\hat{f}^*(\mathcal{X}_i^{(j)}) - f^*(\mathcal{X}_i^{(j)})\right) \\ &\leq -\sum_{\substack{k=1 \\ k \neq i}}^{n_z} \mathrm{ReLU}(c_k) \frac{u_k l_k}{u_k - l_k},\end{aligned}$$

as desired. □

With the partitioned relaxation error bound of Lemma 2 established, we now present the optimal two-part partition.

**Theorem 2** (Optimal partition)**.** *Consider the two-part partitions defined by the rows of $W$: $\{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}\}$, where $\mathcal{X}_i^{(1)} = \{x \in \mathcal{X} : w_i^\top x \geq 0\}$ and $\mathcal{X}_i^{(2)} = \mathcal{X} \setminus \mathcal{X}_i^{(1)}$ for all $i \in \{1, 2, \dots, n_z\} =: \mathcal{I}$. The optimal partition that minimizes the worst-case relaxation error in (17) is given by*

$$i^* \in \arg\min_{i \in \mathcal{I}} \mathrm{ReLU}(c_i) \frac{u_i l_i}{u_i - l_i}. \tag{18}$$

*Proof.* Minimizing the bound in (17) of Lemma 2 over the partition $i$ gives rise to

$$\begin{aligned}&\min_{i \in \mathcal{I}} \left(-\sum_{\substack{k=1 \\ k \neq i}}^{n_z} \mathrm{ReLU}(c_k) \frac{u_k l_k}{u_k - l_k}\right) \\ &= -\sum_{k=1}^{n_z} \mathrm{ReLU}(c_k) \frac{u_k l_k}{u_k - l_k} + \min_{i \in \mathcal{I}} \mathrm{ReLU}(c_i) \frac{u_i l_i}{u_i - l_i},\end{aligned}$$

as desired. □

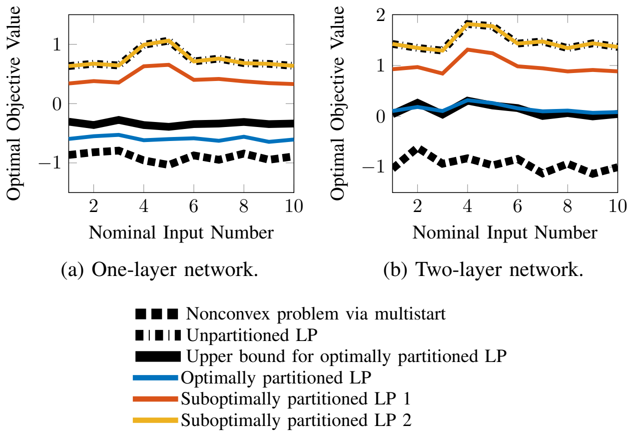
Theorem 2 provides a methodical way of selecting the optimal two-part partition based on the rows of $W$ in a worst-case sense. To understand the efficiency of this result, notice that the optimization over $i$ scales linearly with the dimension $n_z$, and the resulting LP subproblems require the addition of only one extra linear constraint. Finally, we note that Theorem 2 can be immediately extended in two ways. First, by ordering the values $\mathrm{ReLU}(c_i) \frac{u_i l_i}{u_i - l_i}$ in (18), we can choose the best $n_p > 1$ rows to partition along in order to perform a $2^{n_p}$-part partition. The other application of Theorem 2 is the following recursion: solve the two-part partitioned LP using Theorem 2 to partition $\mathcal{X}$. If $\mathcal{X}_{i^*}^{(j^*)}$ is the input part containing the solution $x^*$, then partition $\mathcal{X}_{i^*}^{(j^*)}$ into two smaller sub-parts again according to Theorem 2 and solve the refined partitioned LP over the sub-parts. Continuing this nested procedure results in tightened localization of a true worst-case input (i.e., a solution to $f^*(\mathcal{X})$) and further reduces conservatism of the partitioned LP relaxation.

## VI. Simulation Results

Consider a one-layer classification network with four inputs and three outputs, trained on the celebrated Iris data set [28] with a test accuracy of 97%. A negative optimal objective value in the robustness certification problem indicates that any perturbation in $\mathcal{X} = \{x \in \mathbb{R}^{n_x} : \|x - \bar{x}\|_\infty \leq \epsilon\}$ of the nominal input $\bar{x}$ will not change the input's classification. For this experiment, we solve the certification problem for 10 different nominal inputs using MATLAB and CVX on a Windows 7 laptop with a 2.9 GHz quad-core i7 processor.

We first solve the problem in its nonconvex form (using multistart and MATLAB's `fmincon` function). We then solve the problem using an unpartitioned LP relaxation, and then using partitioned LP relaxations, one per row of $W$. We restrict our experiments to two-part partitions to explore the effect of changing the row of $W$ along which we partition. The average time taken to solve the nonconvex, unpartitioned LP, and partitioned LPs are $0.21$, $0.26$, and $0.48$ seconds, respectively. As expected, the computational burden of the two-part partitioned LP is twice that of the unpartitioned LP, both of which are very fast for this network.

The optimal objective values for each nominal input are shown in Fig. 4a. As seen, the optimally partitioned LP developed in Theorem 2 yields the best convex upper bound on the true problem. Furthermore, ordering the rows $w_i^\top$ by their suboptimality in (18) corresponds to the order of relaxation tightening. For instance, in Fig. 4, we compare the LP partitioned via $i_1 \in \arg\min_{i \in \mathcal{I} \setminus \{i^*\}} \mathrm{ReLU}(c_i) \frac{u_i l_i}{u_i - l_i}$ (suboptimally partitioned LP 1) and that partitioned via $i_2 \in \arg\min_{i \in \mathcal{I} \setminus \{i^*, i_1\}} \mathrm{ReLU}(c_i) \frac{u_i l_i}{u_i - l_i}$ (suboptimally partitioned LP 2). In this example, the suboptimally partitioned LP 2 (partitioned along worst row $w_{i_2}^\top$) coincides with the unpartitioned LP, suggesting that none of the relaxation error is attributed to the $i_2^{\text{th}}$ coordinate of the ReLU layer. The suboptimally partitioned LPs do not certify robustness, as the objective values are positive for each nominal input tested. On the other hand, the developed optimal partitioning scheme tightens the relaxation enough to provide a certificate of robustness for the one-layer network corresponding to every nominal input tested here. For the same experiment on a two-layer network (with an added five-neuron ReLU layer), we find that the optimal partitioning scheme maintains the best convex upper bound, albeit without guaranteed relaxation error bounds (see Fig. 4b). The average computation times for the nonconvex, unpartitioned LP, and partitioned LPs rise to $0.94$, $0.68$, and $1.48$ seconds, respectively. For general network sizes, the two-part partitioned LP maintains the polynomial-time complexity of linear programming with respect to the number of neurons, since it requires solving two instances of the same LP structure [29].




(a) One-layer network. (b) Two-layer network.

Fig. 4. Optimal values of robustness certification for ReLU Iris classifier. For the two-layer network, the optimal partitioning scheme is applied to only the ReLU constraints of the final layer.

## VII. Conclusions

In this work, we develop a partition-based method for ReLU neural network robustness certification that systematically reduces relaxation error while maintaining the efficiency of linear programming. We theoretically justify the effectiveness of partitioning and derive an optimal partitioning scheme. A case study on real data shows that the proposed method is able to certify the robustness of a network while the existing methods fail. Our results demonstrate, both theoretically and experimentally, that partition-based certification procedures are capable of tightening relaxation errors with remarkable simplicity.

## References

[1] S. N. Kumpati, P. Kannan *et al.*, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 4–27, 1990.

[2] F. Lewis, S. Jagannathan, and A. Yesildirak, *Neural network control of robot manipulators and non-linear systems*. CRC Press, 1998.

[3] G. P. Liu, *Nonlinear identification and control: a neural network approach*. Springer Science & Business Media, 2012.

[4] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 4653–4660.

[5] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for koopman operators of nonlinear dynamical systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4832–4839.

[6] D. H. Nguyen and B. Widrow, "Neural networks for self-learning control systems," *IEEE Control systems magazine*, vol. 10, no. 3, pp. 18–23, 1990.

[7] E. N. Johnson and A. J. Calise, "Neural network adaptive control of systems with input saturation," in *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, vol. 5. IEEE, 2001, pp. 3527–3532.

[8] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[9] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 129–137.

[10] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, 2017.

[11] K. Muralitharan, R. Sakthivel, and R. Vishnuvarthan, "Neural network based optimization approach for energy demand prediction in smart grid," *Neurocomputing*, vol. 273, pp. 199–208, 2018.

[12] X. Pan, T. Zhao, and M. Chen, "Deepopf: Deep neural network for dc optimal power flow," in *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2019, pp. 1–6.

[13] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," *arXiv preprint arXiv:1711.00851*, 2017.

[14] A. Raghunathan, J. Steinhardt, and P. S. Liang, "Semidefinite relaxations for certifying robustness to adversarial examples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 877–10 887.

[15] W. Xiang and T. T. Johnson, "Reachability analysis and safety verification for neural network control systems," *arXiv preprint arXiv:1805.09944*, 2018.

[16] T.-W. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, D. Boning, I. S. Dhillon, and L. Daniel, "Towards fast computation of certified robustness for relu networks," *arXiv preprint arXiv:1804.09699*, 2018.

[17] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in neural information processing systems*, 2018, pp. 4939–4948.

[18] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *arXiv preprint arXiv:1903.01287*, 2019.

[19] V. R. Royo, R. Calandra, D. M. Stipanovic, and C. Tomlin, "Fast neural network verification via shadow prices," *arXiv preprint arXiv:1902.07247*, 2019.

[20] M. Jin, H. Chang, W. Zhu, and S. Sojoudi, "Power up! robust graph convolutional network against evasion attacks based on graph powering," *arXiv preprint arXiv:1905.10029*, 2019.

[21] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton University Press, 2009, vol. 28.

[22] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.

[23] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 97–117.

[24] D. Bertsimas and I. Dunning, "Multistage robust mixed-integer optimization with adaptive partitions," *Operations Research*, vol. 64, no. 4, pp. 980–998, 2016.

[25] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in neural information processing systems*, 2014, pp. 2924–2932.

[26] M. A. Duran and I. E. Grossmann, "An outer-approximation algorithm for a class of mixed-integer nonlinear programs," *Mathematical programming*, vol. 36, no. 3, pp. 307–339, 1986.

[27] O. L. Mangasarian and T.-H. Shiau, "Lipschitz continuity of solutions of linear inequalities, programs and complementarity problems," *SIAM Journal on Control and Optimization*, vol. 25, no. 3, pp. 583–595, 1987.

[28] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[29] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 1984, pp. 302–311.