Towards Optimality and Robustness Guarantees for Data-Driven Learning and Decision Making

by

Brendon G. Anderson

B.S., University of California, Los Angeles, 2018

A report submitted in partial satisfaction of the

requirements for the degree of

Master of Science, Plan II

in

Mechanical Engineering

at the

University of California, Berkeley

Committee in charge:

Professor Somayeh Sojoudi, Chair

Professor Javad Lavaei

Spring 2020

Abstract

Towards Optimality and Robustness Guarantees for Data-Driven

Learning and Decision Making

by

Brendon G. Anderson

Master of Science in Mechanical Engineering

University of California, Berkeley

Professor Somayeh Sojoudi, Chair

In nearly all fields of modern science and engineering, data is used to learn models of complex systems, and to generate decisions and control actions for their operation. Many of these systems, including autonomous vehicles and the power grid, are safety-critical, and therefore data-driven methods require performance guarantees in order to safely apply them. This report develops two novel computation techniques with associated theoretical guarantees. First, we consider the problem of unsupervised video object segmentation. We pose the nonsemantic extraction of a video's moving objects as a nonnegative variant of nonconvex robust principal component analysis. The resulting formulation is more computationally tractable than its commonly employed convex relaxation, although not generally solvable to global optimality. In spite of this limitation, we derive conditions on the video data under which the uniqueness and global optimality of the object segmentation are guaranteed using local search methods. The second part of this report considers certifying the robustness of neural networks to perturbed and adversarial input data. Certification techniques using convex optimization have been proposed, but they often suffer from relaxation errors that void the certificate. The work in this report 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 a tractable partitioning scheme that minimizes the worst-case relaxation error is derived. Consequently, the main developments of this report provide steps towards building a repertoire of optimality and robustness guarantees for data-driven learning and decision making.

Contents

List of Figures

List of Symbols

$\mathbb{R}$	Set of real numbers.
$\mathbb{Z}$	Set of integers.
$\mathbb{X}^n$	Set of n-vectors with elements in the set $\mathbb{X}$.
$\mathbb{X}^{m \times n}$	Set of $m \times n$ matrices with elements in the set $\mathbb{X}$.
$\mathbf{1}_n$	n-vector of all ones, i.e., $\mathbf{1}_n = (1, 1, \dots, 1) \in \mathbb{R}^n$.
0	Array (number, vector, or matrix, depending on context) of all zeros.
$X \leq Y$	The array X is element-wise less than or equal to the array Y.
$\mathbb{X}_+$	Element-wise nonnegative members of the set $\mathbb{X}$, i.e., $\mathbb{X}_+ = \{x \in \mathbb{X} : x \geq 0\}$.
$\mathbb{X}_{++}$	Element-wise positive members of the set $\mathbb{X}$, i.e., $\mathbb{X}_{++} = \{x \in \mathbb{X} : x > 0\}$.
$X \odot Y$	Element-wise (Hadamard) product between arrays X and Y.
$X \oslash Y$	Element-wise (Hadamard) division of array X by array Y.
$\lVert X \rVert_p$	Element-wise p-norm of the array X.
$\lvert X \rvert$	Element-wise absolute value if X is an array. Cardinality if X is a set.
$\lceil x \rceil$	Ceiling of $x \in \mathbb{R}$.
$\mathrm{vec}(X)$	Column-wise vectorization of the matrix X.
$\mathrm{ReLU}(X)$	Rectifier, i.e., $\mathrm{ReLU}(X) = \max\{0, X\}$, with element-wise maximum.

Chapter 1

Introduction

1.1 Motivations

This work focuses on two problems in the areas of data-driven learning and decision making. In the first part of the report, we consider learning the foreground and background of a video in an unsupervised manner. The second part considers the analysis of neural network decision-making algorithms under the influence of perturbed and adversarial input data. Although these applications are seemingly disjoint, the contributions of this report focus on a common underlying research direction: the development of data-driven learning and decision-making algorithms with theoretically guaranteed optimality and robustness properties. Such algorithms are of paramount importance for modeling and controlling safety-critical systems with complex and sensitive behavior. For instance, when employing a video segmentation algorithm on an autonomous vehicle, it is imperative to guarantee that the algorithm does not converge to a suboptimal result in the presence of noisy video data, e.g., when a shadow occludes a pedestrian. In the neural network setting, one needs to ensure that the algorithm's decisions and control actions are unaffected by adversarial input data, e.g., when an attacker attempts to crash the operation of an electricity network.

1.2 Contributions

As the title suggests, this work takes steps towards building a toolbox of efficient algorithms with theoretical performance guarantees. The main contributions are summarized as follows. In Chapter 2, we propose the use of a nonconvex model for performing computationally efficient unsupervised video object segmentation. Despite the model's nonconvexity, we derive intuitive and interpretable conditions on video data under which the uniqueness and global optimality of the resulting segmentation are guaranteed. These results initially appeared in [1]. In Chapter 3, we propose a simple and efficient method to certify the robustness of feedforward ReLU neural networks. In particular, we use partitioning to develop a linear programming certification procedure with provable bounds on its relaxation error, and use these bounds to derive an optimal partitioning scheme. This work initially appeared in [2]. In both Chapters 2 and 3, numerical experiments are provided to illustrate the efficacy of the results. Finally, concluding remarks are provided in Chapter 4.

Chapter 2

Optimality Guarantees for Unsupervised Video Segmentation

2.1 Introduction

One of the most fundamental problems in computer vision and machine learning is that of video object segmentation. In this domain, the general goal is to distinguish and extract objects of interest from the rest of the video's content. Visual segmentation algorithms take on a variety of different tasks and forms. For instance, semantic segmentation tackles the problem of assigning each extracted object to a certain cluster or predefined class, and supervised (or semi-supervised) methods are endowed with one or more ground truth extractions or annotations [3]. This wide range of methodologies makes video segmentation suitable for many applications, such as surveillance systems, traffic monitoring, and gesture recognition, and therefore video object segmentation remains an active and challenging area of research [4, 5].

2.1.1 Unsupervised Video Segmentation

This work is concerned with nonsemantic and unsupervised video segmentation via background subtraction; the task of extracting moving objects from a video's background using static cameras. Traditional techniques for moving object segmentation typically use Gaussian mixture models (GMM), which offer simple models but lack robustness [6, 7]. Neural networks have also found popularity due to the balance they strike between performance and computational efficiency [8]. However, due to their nonconvex nature, neural networks do not generally possess guarantees on the global optimality of their resulting segmentation [9].

Recently, much attention has been placed on approaches based on robust principal component analysis (RPCA), which model the video as the sum of low-rank and sparse matrices. Perhaps the most notable of these methods is Principal Component Pursuit (PCP) introduced in the seminal paper by Candès et al. [10]. Although the convexified approach in PCP provides conditions under which exact recovery of the sparse components is guaranteed, its use of lifted variables results in scalability and computational hindrances [9].

In order to tackle large-scale segmentation problems, lower-dimensional nonconvex formulations such as nonnegative robust principal component analysis (NRPCA) and robust nonnegative matrix factorization (RNMF) have been proposed [11, 12, 13, 14]. These nonconvex approaches often permit parallelization, lending themselves to lowered computational cost and scalability to larger problems [15]. Furthermore, the nonnegative nature of grayscale pixel values is explicitly embedded in modern methods like NRPCA and RNMF, unlike many of the more traditional techniques. Although these nonconvex formulations have been empirically shown to have performance on par with the popular PCP method, previous works have focused on local optimality of the resulting video segmentations, often solved for by alternating over the subproblems that are convex in the variables separately [12, 13].

2.1.2 Contributions

In this work, we aim to supplement the strong empirical and computational properties of video segmentation via nonconvex NRPCA by providing intuitive and interpretable global optimality guarantees. These guarantees target two key aspects of moving object segmentation. First, they promise global solutions when using local search algorithms, such as stochastic gradient descent or its variants. The computational efficiency of these simple algorithms is paramount in large-scale machine learning problems [16, 17, 18], e.g., those with high-resolution video data. Second, safety-critical video segmentation applications, such as autonomous driving [19] and medical imaging [20], demand global optimality guarantees to promise consistent performance and safety margins. With the recent influx of studies on spurious local minima of nonconvex optimization problems [21, 22, 23, 24], we approach this problem by exploiting new results on the benign landscape of rank-1 NRPCA [11]. Under this framework, we propose criteria under which the video segmentation is guaranteed to be unique and globally optimal.

2.1.3 Organization

The remainder of this chapter is structured as follows. In Section 2.2, we describe the problem and introduce our terminology and notations. In Section 2.3, we show that the problem can be simplified to one in which the moving objects consist of elementary shapes. Then, in Sections 2.4 and 2.5, we derive conditions on video data under which global optimality guarantees can be made. Finally, we perform numerical experiments in Section 2.6.

2.2 Problem Statement

2.2.1 Video Description

Consider a video sequence of d_f frames, each being d_m pixels tall and d_n pixels wide, where $d_f, d_m, d_n \in \mathbb{Z}_{++}$. We denote the video frames by the matrices $X^{(k)} \in \mathbb{R}^{d_m \times d_n}$, where $k \in K := \{1, 2, \dots, d_f\}$. By defining the *pixel set* as $\Pi = \{1, 2, \dots, d_m\} \times \{1, 2, \dots, d_n\}$, the pixels of a grayscale video are given by

$$X_{ij}^{(k)} \in \mathcal{X} \subseteq \mathbb{R}, \quad (i, j) \in \Pi, \; k \in K,$$

where conventionally, $\mathcal{X} = \{0, 1, \dots, 255\}$ or $\mathcal{X} = [0, 1]$. In this work, we scale the pixel values to an interval $\mathcal{X} = [X_{\text{black}}, X_{\text{white}}] \subseteq \mathbb{R}_+$, for technical reasons explained later. Vectorizing each frame of the video, we form the data matrix

$$X = \begin{bmatrix} \operatorname{vec}(X^{(1)}) & \operatorname{vec}(X^{(2)}) & \cdots & \operatorname{vec}(X^{(d_f)}) \end{bmatrix} \in \mathbb{R}^{m \times n},$$

where $m = d_m d_n$ and $n = d_f$. Note that $\operatorname{vec}\colon \mathbb{R}^{d_m \times d_n} \to \mathbb{R}^{d_m d_n}$ converts each frame into an equivalent extended vector, so that the single matrix X captures all of the video's information. We also define the *measurement set* as $\Omega = \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$.

2.2.2 Foreground-Background Segmentation

We choose to model the video data matrix as the sum of two components. The first component is chosen to be a nonnegative rank-1 matrix, used to capture the relatively static behavior of the video's background. The second component is a sparse matrix, taken to represent the dynamic foreground (i.e., the moving objects). Under this model, we seek the decomposition

$$X \approx uv^\top + S, \tag{2.1}$$

where $u \in \mathbb{R}^m_+$ and $v \in \mathbb{R}^n_+$, and $S \in \mathbb{R}^{m \times n}$ is sparse. This can be solved for through the following nonconvex, nonnegative l_1-minimization problem, termed in the literature as *nonnegative robust principal component analysis* (NRPCA) [11]:

$$\begin{aligned} &\text{minimize} && \|X - uv^\top\|_1 + \lambda \left| u^\top u - v^\top v \right| \\ &\text{subject to} && u \in \mathbb{R}^m_+, \ v \in \mathbb{R}^n_+. \end{aligned} \tag{2.2}$$

This is a nonconvex problem that may generally have spurious local minima, i.e., those points a local search algorithm may find which do not correspond to the globally optimal solution. Note that we enforce nonnegativity of the optimization variables u and v, yielding natural interpretations as the video's nominal background pattern and its associated scalings in each frame, respectively. Furthermore, we have added a regularization term with tuning parameter $\lambda \in \mathbb{R}_{++}$ to our formulation, since the unregularized objective is invariant to scaling. In other words, if (u^*, v^*) minimizes the unregularized problem, then so will $(\alpha u^*, \frac{1}{\alpha} v^*)$ for every $\alpha \in \mathbb{R}_{++}$. Therefore, under regularization, the unique solution should be the pair (u^*, v^*) for which $\|u^*\|_2 = \|v^*\|_2$.

Under the decomposition (2.1), we define the video's *background set* and *foreground set* as $B = \{(h, k) \in \Omega : S_{hk} = 0\}$ and $F = \Omega \setminus B$, respectively. Accordingly, two bipartite graphs can be introduced, the *background graph* $\mathcal{G}_{m,n}(B)$ having edge set B, and the *foreground graph* $\mathcal{G}_{m,n}(F)$ having edge set F. The first vertex set of each graph corresponds to pixel numbers: $V_u = \{1, 2, \dots, m\}$. The second vertex set associates with frame numbers: $V_v = \{m + 1, m + 2, \dots, m + n\}$. A toy example of these graphs follows.

Example 2.1. Suppose that a video has frames given by $X^{(1)} = \left[\begin{smallmatrix} 256 & 1 \\ 256 & 1 \end{smallmatrix}\right]$ and $X^{(2)} = \left[\begin{smallmatrix} 1 & 256 \\ 256 & 256 \end{smallmatrix}\right]$, where elements of 256 represent background. Then, the data matrix is

$$X = \begin{bmatrix} 256 & 1 \\ 256 & 256 \\ 1 & 256 \\ 1 & 256 \end{bmatrix},$$

and the foreground and background sets are, respectively, $F = \{(3,1),(4,1),(1,2)\}$ and $B = \{1,2,3,4\} \times \{1,2\} \setminus F$. The corresponding graphs are shown in Fig. 2.1.

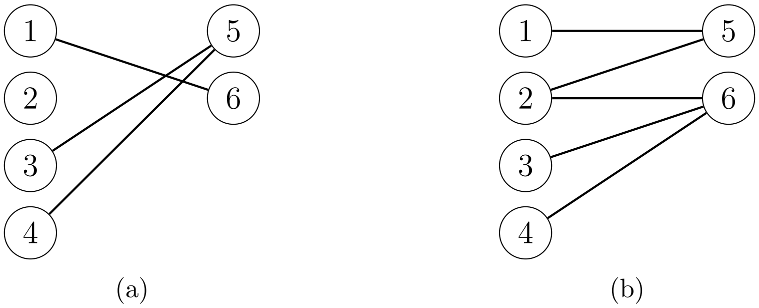


Figure 2.1: Example graphs $\mathcal{G}_{m,n}(F)$ (a), and $\mathcal{G}_{m,n}(B)$ (b).

2.2.3 Optimality Conditions

A remarkable property of the nonconvex and nonsmooth problem (2.2) is that, under certain conditions on the problem data, the optimization landscape is *benign*, i.e., there are no spurious local minima, and the global minimum is unique [11]. This permits the use of simple local search algorithms to solve (2.2) to global optimality. The nonconservative sufficient conditions for benign landscape follow:

$$\text{Connectivity:} \quad \mathcal{G}_{m,n}(B) \text{ is connected.} \tag{2.3}$$

$$\text{Identifiability:} \quad \delta(\mathcal{G}_{m,n}(B)) > \frac{48}{c^2}\kappa(w^*)^4 \Delta(\mathcal{G}_{m,n}(F)). \tag{2.4}$$

In these expressions, we denote the globally optimal solution of (2.2) as $w^* = (u^*, v^*) \in \mathbb{R}^{m+n}$, the condition number (maximum element divided by minimum element) of a vector in the positive orthant as $\kappa(\cdot)$, maximum degree of a graph as $\Delta(\cdot)$, and minimum degree of a graph as $\delta(\cdot)$. The value c is a constant that depends on problem data, which will be discussed in more detail later.

The problem to be addressed is as follows: *When do videos satisfy the conditions* (2.3) *and* (2.4) *to guarantee a benign landscape for the optimization problem* (2.2)*?* In other words, the goal is to determine conditions on the size, shape, and speed of a moving object to provide theoretical guarantees for the unique and globally optimal foreground segmentation of a video. We begin by showing that the problem can be simplified to one with elementary foreground shapes through the notion of object embedding.

2.3 Object Embedding

In this section, we consider two videos with identical backgrounds, each having one moving object (though the results are naturally generalized to multi-object videos). We are interested in the case that the moving object of one video can be completely covered by the moving object of the other video in each frame. Here is the question of interest: *If the video with the* *larger moving object satisfies the conditions* (2.3) *and* (2.4) *for benign landscape of* (2.2)*, does the video with the smaller object also satisfy these conditions?* To answer this question precisely, let us start with the following definition.

Definition 2.1 (Embedding)**.** Consider two videos $\mathcal{O}$ and $\mathcal{R}$ having the same background $uv^\top$, i.e., $X_{\mathcal{O}} = uv^\top + S_{\mathcal{O}}$ and $X_{\mathcal{R}} = uv^\top + S_{\mathcal{R}}$. We say that object $F_{\mathcal{O}}$ is *embedded* in object $F_{\mathcal{R}}$ if the foreground of video $\mathcal{O}$ is a subset of that of video $\mathcal{R}$ in every frame; if

$$F_{\mathcal{O}}^{(k)} \subseteq F_{\mathcal{R}}^{(k)} \text{ for all } k \in K,$$

where $F_{\mathcal{O}}^{(k)} = \{(i,j) \in \Pi : (S_{\mathcal{O}})_{hk} \neq 0,\ h = (j-1)d_m + i\}$, and similarly for $F_{\mathcal{R}}^{(k)}$.

It is desirable to show that the answer to our earlier question is affirmative. We prove these implications in the following two propositions.

Proposition 2.1 (Embedded connectivity)**.** *If object $F_{\mathcal{O}}$ is embedded in object $F_{\mathcal{R}}$ and video $\mathcal{R}$ satisfies the connectivity condition* (2.3)*, then video $\mathcal{O}$ also satisfies the connectivity condition.*

Proof. Since $F_{\mathcal{O}}$ is embedded in $F_{\mathcal{R}}$, we have for all $k \in K$ that $F_{\mathcal{O}}^{(k)} \subseteq F_{\mathcal{R}}^{(k)}$, which implies $\Pi \setminus F_{\mathcal{R}}^{(k)} \subseteq \Pi \setminus F_{\mathcal{O}}^{(k)}$. This shows that the background of video $\mathcal{R}$ is a subset of that of video $\mathcal{O}$, i.e., $B_{\mathcal{R}}^{(k)} \subseteq B_{\mathcal{O}}^{(k)}$ for all $k \in K$, which gives $B_{\mathcal{R}} \subseteq B_{\mathcal{O}}$. Therefore, we have that $\mathcal{G}_{m,n}(B_{\mathcal{R}})$ is a spanning subgraph of $\mathcal{G}_{m,n}(B_{\mathcal{O}})$. Since $\mathcal{G}_{m,n}(B_{\mathcal{R}})$ is connected by our assumption, so must be $\mathcal{G}_{m,n}(B_{\mathcal{O}})$, as desired. □

Proposition 2.2 (Embedded identifiability)**.** *If object $F_{\mathcal{O}}$ is embedded in object $F_{\mathcal{R}}$ and video $\mathcal{R}$ satisfies the identifiability condition* (2.4)*, then video $\mathcal{O}$ also satisfies the identifiability condition.*

Proof. Since $F_{\mathcal{O}}$ is embedded in $F_{\mathcal{R}}$, we have for all $k \in K$ that $F_{\mathcal{O}}^{(k)} \subseteq F_{\mathcal{R}}^{(k)}$, which implies $F_{\mathcal{O}} \subseteq F_{\mathcal{R}}$. Hence, the maximum degrees of the foreground graphs satisfy $\Delta(\mathcal{G}_{m,n}(F_{\mathcal{O}})) \leq \Delta(\mathcal{G}_{m,n}(F_{\mathcal{R}}))$. Similarly, we have that $B_{\mathcal{R}} \subseteq B_{\mathcal{O}}$, and therefore the minimum degrees of the background graphs satisfy $\delta(\mathcal{G}_{m,n}(B_{\mathcal{R}})) \leq \delta(\mathcal{G}_{m,n}(B_{\mathcal{O}}))$. Combining these inequalities with the identifiability inequality for video $\mathcal{R}$ yields

$$\begin{aligned}\delta(\mathcal{G}_{m,n}(B_{\mathcal{O}})) &\geq \delta(\mathcal{G}_{m,n}(B_{\mathcal{R}})) \\ &> \frac{48}{c^2}\kappa(w^*)^4\Delta(\mathcal{G}_{m,n}(F_{\mathcal{R}})) \\ &\geq \frac{48}{c^2}\kappa(w^*)^4\Delta(\mathcal{G}_{m,n}(F_{\mathcal{O}})),\end{aligned}$$

showing video $\mathcal{O}$ also satisfies the identifiability condition. □

It is clear that Propositions 2.1 and 2.2 are independent of the size, shape, and speed of a moving object. This allows us to restrict the rest of our analysis to videos with moving objects of elementary shapes, since a more complicated object may always be embedded into a larger object which covers it. In the case that the larger, simpler object is found to satisfy the conditions (2.3) and (2.4), the results of this section show the embedded object can be extracted to unique global optimality. Therefore, we will focus on rectangular moving objects for the remainder of this work, for convenience.

2.4 Conditions for Connectivity

In this section, we aim to derive necessary and sufficient criteria for a video to satisfy the connectivity condition (2.3). We will start by defining the notion of connected backgrounds, which will assist with streamlining the proofs in Sections 2.4.1 and 2.4.2, in addition to granting intuitive interpretations to the conditions that follow.

Definition 2.2 (Background connectivity)**.** Given a video with frames $k \in K$ having associated background pixel sets

$$B^{(k)} = \{(i,j) \in \Pi : S_{hk} = 0,\ \ h = (j-1)d_m + i\},$$

the video is said to have a *connected background* if the following two conditions are satisfied:

1. $\cup_{k \in K} B^{(k)} = \Pi$.
2. $B_1 \cap B_2 \neq \emptyset$ for all $B_1 = \cup_{k \in K_1} B^{(k)}$ and $B_2 = \cup_{k \in K_2} B^{(k)}$ such that $K_1 \cup K_2 = K$.

We now show that having a connected background is equivalent to the video's background graph $\mathcal{G}_{m,n}(B)$ being connected; videos with connected backgrounds satisfy the connectivity condition (2.3). This is useful, since we will use Definition 2.2 to derive simple and intuitive necessary conditions a video must satisfy in order to have a connected background (and therefore to satisfy the connectivity condition). Afterwards, we prove a sufficient condition for background connectivity, which we claim is likely satisfied for nearly any video in practice.

Proposition 2.3 (Connectivity equivalence)**.** *A video's background graph, $\mathcal{G}_{m,n}(B)$, is connected if and only if the video has a connected background.*

Proof. The proof will proceed via contrapositive argument. We will first prove necessity.

Necessity: Suppose that a video does not have a connected background. Then, one of the two following cases must hold:

1. $\cup_{k \in K} B^{(k)} \neq \Pi$.
2. There exist $B_1 = \cup_{k \in K_1} B^{(k)}$ and $B_2 = \cup_{k \in K_2} B^{(k)}$, where $K_1 \cup K_2 = K$, such that $B_1 \cap B_2 = \emptyset$.

Assume that the first case holds. Then, there exists a pixel $(i_0, j_0) \in \Pi$ such that $(i_0, j_0) \notin B^{(k)}$ for all $k \in K$. Therefore, we have $S_{h_0 k} \neq 0$ where $h_0 = (j_0 - 1)d_m + i_0$, which implies

$$(h_0, k) \notin B \text{ for all } k \in K.$$

This shows that vertex $h_0 \in V_u$ has no incident edges in $\mathcal{G}_{m,n}(B)$, and therefore the graph is disconnected.

Now, assume that the second case holds. We first note that $K_1 \cap K_2 = \emptyset$, since otherwise B_1 and B_2 cannot be disjoint. Now, $B_1 \cap B_2 = \emptyset$ implies that for all pixels $(i,j) \in \Pi$, either $(i,j) \in B_1$ and $(i,j) \notin B_2$, or $(i,j) \in B_2$ and $(i,j) \notin B_1$, or $(i,j) \notin B_1$ and $(i,j) \notin B_2$. In the trivial case that some pixel (i_0, j_0) is neither an element of B_1 nor an element of B_2, then $\cup_{k\in K} B^{(k)} = B_1 \cup B_2 \neq \Pi$, and the first case above shows that the graph $\mathcal{G}_{m,n}(B)$ is disconnected. For pixels $(i,j) \in B_1$, we have $(i,j) \notin B^{(k)}$ for all $k \in K_2$, and therefore $S_{hk} \neq 0$ where $h = (j-1)d_m + i$, which implies

$$(h,k) \notin B \text{ for all } k \in K_2.$$

This shows that vertex $h \in V_u$ is not adjacent to vertex $m+k \in V_v$ for all $k \in K_2$. Similarly, one can show that for each $(i,j) \in B_2$, the corresponding vertex $h \in V_u$ is not adjacent to vertex $m+k \in V_v$ for all $k \in K_1$. Since $B_1 \cap B_2 = \emptyset$ and $K_1 \cap K_2 = \emptyset$, the bipartite graph $\mathcal{G}_{m,n}(B)$ contains at least two connected components, defined by the disjoint edge sets $\mathcal{E}_1 \subseteq \{(h, m+k) : h = (j-1)d_m + i,\ (i,j) \in B_1,\ k \in K_1\}$ and $\mathcal{E}_2 \subseteq \{(h, m+k) : h = (j-1)d_m + i,\ (i,j) \in B_2,\ k \in K_2\}$. Therefore, the graph is disconnected.

Sufficiency: Suppose that a video's associated background graph, $\mathcal{G}_{m,n}(B)$, is disconnected. Then, one of the two following cases must hold:

1. There exists a vertex with no incident edges.

2. Every vertex has at least one incident edge.

Assume that the first case holds. Then, either the isolated vertex corresponds to a pixel number h_0 or to a frame number k_0. If vertex $h_0 \in V_u$ is isolated, then $(h_0, k) \notin B$ for all $k \in K$. This implies $S_{h_0 k} \neq 0$ and therefore $(i_0, j_0) \notin B^{(k)}$ for all $k \in K$, where

$$(i_0, j_0) = \left(h_0 - \left(\left\lceil \frac{h_0}{d_m} \right\rceil - 1 \right) d_m, \left\lceil \frac{h_0}{d_m} \right\rceil \right).$$

(Here, $\lceil \cdot \rceil$ represents the ceiling operator. This formula comes from the one-to-one correspondence between a pixel (i,j) and its pixel number h through the vectorization of a given video frame.) Thus, $(i_0, j_0) \notin \cup_{k\in K} B^{(k)}$, which implies $\cup_{k\in K} B^{(k)} \neq \Pi$. Hence, the video does not have a connected background. On the other hand, if vertex $k_0 \in K$ is isolated, then $(h, k_0) \notin B$ for all $h \in V_u$. This implies $S_{hk_0} \neq 0$ and therefore $(i,j) \notin B^{(k_0)}$ for all $(i,j) \in \Pi$. Thus, $B^{(k_0)} = \emptyset$. Define $B_1 = B^{(k_0)}$ and $B_2 = \cup_{k\in K\setminus\{k_0\}} B^{(k)}$. Then $B_1 \cap B_2 = \emptyset$, so again the video does not have a connected background.

Now, assume the second case holds. Then, the graph contains at least two nontrivial connected components. Therefore, the set B, which defines the edge set of the graph, can be partitioned as $B = Q_1 \cup Q_2$, where $Q_1 = \{(h,k) \in \Omega : S_{hk} = 0,\ h \in H_1,\ k \in K_1\}$ and $Q_2 = \{(h,k) \in \Omega : S_{hk} = 0,\ h \in H_2,\ k \in K_2\}$ are nonempty, such that $H_1 = V_u \setminus H_2$ and $K_1 = K \setminus K_2$. Now, define $B_1 = \cup_{k\in K_1} B^{(k)}$. This gives

$$\begin{aligned} B_1 &= \cup_{k\in K_1} \{(i,j) \in \Pi : S_{hk} = 0,\ h = (j-1)d_m + i\} \\ &= \{(i,j) \in \Pi : S_{hk} = 0,\ h = (j-1)d_m + i,\ k \in K_1\}. \end{aligned}$$

Now, from the partitions Q_1 and Q_2 we see that a frame $k \in K_1$ has $S_{hk} = 0$ only for pixel numbers $h \in H_1$. Thus, B_1 can be written equivalently as

$$B_1 = \{(i,j) \in \Pi : S_{hk} = 0,\ h = (j-1)d_m + i,\ k \in K_1,\ h \in H_1\}.$$

Similarly, it can be shown that by defining $B_2 = \cup_{k\in K_2} B^{(k)}$, we obtain

$$B_2 = \{(i,j) \in \Pi : S_{hk} = 0,\ h = (j-1)d_m + i,\ k \in K_2,\ h \in H_2\}.$$

Since $H_1 \cap H_2 = \emptyset$ and $K_1 \cap K_2 = \emptyset$, we immediately see that $B_1 \cap B_2 = \emptyset$. Therefore, the video does not have a connected background. □

Proposition 2.3 shows that the connectivity of the graph $\mathcal{G}_{m,n}(B)$ is entirely dictated by whether or not a video has a connected background. Therefore, we can use the notion of background connectivity to derive intuitive and meaningful criteria a video should satisfy in order to meet the connectivity condition (2.3).

2.4.1 Necessary Conditions for Connectivity

From Definition 2.2, we develop three necessary conditions for background connectivity of a video, which are intuitively interpretable in terms of properties of the video (i.e., properties of pixels and frames). These necessary conditions give simple methods for showing when a video does not have a connected background, in which case no guarantees on the global optimality of the minimization (2.2) can be made.

Proposition 2.4 (Object size)**.** *If a video has a connected background, then there are at most $d_m d_n d_f - (d_m d_n + d_f - 1)$ foreground pixels in the data matrix X.*

Proof. Since the background graph $\mathcal{G}_{m,n}(B)$ has $m+n = d_m d_n + d_f$ vertices and is connected, the number of edges $|B|$ is at least $d_m d_n + d_f - 1$. Therefore, $|F| = mn - |B| \leq d_m d_n d_f - (d_m d_n + d_f - 1)$. □

For an instance in which the upper bound given by Proposition 2.4 is tight, yet background connectivity is still achieved, see Example 2.1. Perhaps the most interesting implication of this result comes from the following corollary.

Corollary 2.1. *As the video resolution and number of frames increase, the maximum relative size of recognizable objects increases.*

Proof. Since there can be at most $d_m d_n d_f - (d_m d_n + d_f - 1)$ foreground pixels across all frames of the video, the maximum relative size of an object can be expressed as

$$p_{\max} := \frac{d_m d_n d_f - (d_m d_n + d_f - 1)}{d_m d_n + d_f - 1}.$$

As the resolution of the video increases, $d_m d_n \to \infty$, and therefore

$$\lim_{d_m d_n \to \infty} p_{\max} = d_f - 1.$$

Furthermore, as the length of the video increases, $d_f \to \infty$, and therefore

$$\lim_{\substack{d_m d_n \to \infty \\ d_f \to \infty}} p_{\max} = \infty.$$

Thus, we see that the maximum permissible ratio of foreground pixels to background pixels increases with the video's resolution and number of frames, as desired. □

Interestingly, the maximum relative object size $p_{\max}$ also shows us that with $d_f = 1$ frame (i.e., a single picture), the largest recognizable object size decreases to $p_{\max} = 0$. On the other hand, with $d_m d_n = 1$ (i.e., a single pixel resolution), the largest recognizable object again decreases to $p_{\max} = 0$. In other words, we cannot recognize moving objects with only one frame, even with infinite resolution, and we also cannot recognize objects with only one pixel, even with infinitely many frames. Both of these observations align with the restrictions on video properties one would expect.

Proposition 2.5 (Frame connectivity)**.** *If a video has a connected background, then each frame contains at least one background pixel.*

Proof. Suppose that there exists a frame $k_0 \in K$ that contains no background pixels, i.e., $B^{(k_0)} = \emptyset$. Then, the video's background pixel sets can be partitioned as $B_1 = B^{(k_0)} = \emptyset$ and $B_2 = \cup_{k \in K \setminus \{k_0\}} B^{(k)}$. Thus, $B_1 \cap B_2 = \emptyset$, and therefore the video does not have a connected background. □

Proposition 2.5 can be interpreted as the requirement that an object can at no point cover the entirety of the frame. This matches intuition, since a moving object surely cannot be uniquely segmented from its background in these types of frames. A similar necessary condition on the obscurement of pixels, rather than frames, is given in Proposition 2.6 that follows.

Proposition 2.6 (Pixel connectivity)**.** *If a video has a connected background, then each pixel is a background pixel in at least one frame.*

Proof. Suppose that there exists a pixel $(i_0, j_0) \in \Pi$ that is a foreground pixel for all frames $k \in K$. Then, $(i_0, j_0) \notin B^{(k)}$ for all $k \in K$. Thus, $(i_0, j_0) \notin \cup_{k \in K} B^{(k)}$, which implies $\cup_{k \in K} B^{(k)} \neq \Pi$, and therefore the video does not have a connected background. □

Proposition 2.6 shows that if any single pixel remains as part of the foreground throughout the video's duration, we cannot guarantee benign landscape of (2.2). This makes sense intuitively: if part of the background remains obscured throughout the video's entirety, it appears implausible to guarantee unique and globally optimal recovery of that part of the background.

2.4.2 Sufficient Conditions for Connectivity

The necessary conditions derived in Section 2.4.1 are most useful in determining when the global optimality guarantees for (2.2) *fail* to hold. In this section, we reverse the implications to derive a simple and relatively relaxed sufficient condition for ensuring the graph $\mathcal{G}_{m,n}(B)$ is connected. This leads to our first main result.

Theorem 2.1 (Common background pixel)**.** *Suppose that each pixel of a video is a background pixel in at least one frame. If any single pixel is a background pixel in all frames of the video, then the video has a connected background.*

Proof. Since each pixel in the video is assumed to be a background pixel in at least one frame, we have that for every $(i,j) \in \Pi$, there exists $k \in K$ such that $(i,j) \in B^{(k)}$. This implies $\cup_{k\in K} B^{(k)} = \Pi$, so the video satisfies the first condition for background connectivity.

Now, suppose that there exists a pixel $(i_0, j_0) \in \Pi$ such that (i_0, j_0) is a background pixel in all frames of the video. Furthermore, assume that the background pixels are partitioned as $B_1 = \cup_{k\in K_1} B^{(k)}$ and $B_2 = \cup_{k\in K_2} B^{(k)}$, where K_1 and K_2 are any two arbitrary subsets of K such that $K_1 \cup K_2 = K$. Since $(i_0, j_0) \in B^{(k)}$ for all $k \in K$, it must be that $(i_0, j_0) \in B_1$ and $(i_0, j_0) \in B_2$, and therefore $B_1 \cap B_2 \neq \emptyset$. Since B_1 and B_2 are arbitrary partitions, the video satisfies the second condition for background connectivity. Thus, the video has a connected background. □

The sufficient condition given in Theorem 2.1 is relaxed in the sense that many videos satisfy the property of having at least one common background pixel among all frames. These common background pixels are often found in the corners of a video, away from the "action" of the moving objects. Therefore, with the prior knowledge that a single pixel remains unobscured by the moving objects throughout the duration of the video, the connectedness of the video's background (and therefore the connectedness of $\mathcal{G}_{m,n}(B)$) comes at only the price of ensuring that no single pixel is obscured by foreground throughout the video's entirety. This property is instantiated later in the example of Section 2.6. We now focus our attention on the identifiability inequality (2.4).

2.5 Conditions for Identifiability

Recall the identifiability condition (2.4). The goal of this section is to determine what properties a video and its moving objects must possess in order to satisfy this condition. We make the following assumptions.

Assumption 2.1. As supported by Propositions 2.1 and 2.2, we assume that the foreground F is a $p_m \times p_n$ rectangle, and that at least one frame contains the entire object. Furthermore, we define $p_f \in \mathbb{Z}_{++}$ to be the maximum number of frames any pixel is obscured by the object. (Note that this can be directly computed for a variety of simple trajectories; see Remark 2.2.) We assume that there exists a black pixel in at least one frame, i.e., $X_{hk} = X_{\text{black}}$ for some $(h,k) \in \Omega$, and that $[X_{\text{black}}, X_{\text{white}}] \subseteq \mathbb{R}_{++}$. We also assume $\|u^*\|_2 = \|v^*\|_2$, as motivated in Section 2.2. Additionally, we take $v^* = v_0^* \mathbf{1}_n$ for some $v_0^* \in \mathbb{R}_{++}$, which holds when the background remains constant through the video's duration, and approximately holds when the illumination variance is small enough. Finally, we set $d_f = d_m d_n$, which turns out to be a key assumption for deriving bounds on $\kappa(w^*)$.

Remark 2.1. (Data preprocessing) Various preprocessing techniques can be used to ensure the assumptions on the problem data. For instance, shifting each pixel value by $\Delta X \in \mathbb{R}_{++}$ ensures that $[X_{\text{black}}, X_{\text{white}}] = [\Delta X, 255 + \Delta X] \subseteq \mathbb{R}_{++}$. Furthermore, in a high-resolution video, we will typically find that $d_f < d_m d_n$. In this case, the equality $d_f = d_m d_n$ can be achieved by either repeating the video to increase the overall length d_f, or by compressing the video to lower the resolution $d_m d_n$. The first approach is beneficial in the case that full-resolution video is needed, whereas the second approach lowers the problem dimension and speeds up computation. To appropriately rescale the resolution, one may set the new frame dimensions to $d'_m = \beta d_m$ and $d'_n = \beta d_n$, where $\beta = \sqrt{\frac{d_f}{d_m d_n}}$. This is the approach we take in the experiments in Section 2.6.

We now prove the main result of this section.

Theorem 2.2 (Rectangle identifiability)**.** *Suppose that a video satisfies Assumption 2.1. Then the video satisfies the identifiability condition* (2.4) *if and only if*

$$\begin{aligned} p_f &< \frac{1}{c_0} d_f, \\ p_m p_n &< \frac{1}{c_0} d_m d_n, \end{aligned} \tag{2.5}$$

where $c_0 = 49$*.*

Proof. As seen in the identifiability condition (2.4), there are four values to analyze: the condition number $\kappa(w^*)$, the parameter c, the maximum degree $\Delta(\mathcal{G}_{m,n}(F))$, and the minimum degree $\delta(\mathcal{G}_{m,n}(B))$. We will first divide the proof into four separate computations, each dedicated to one of these values, then combine the final results at the end.

1) Condition number: Since $\|u^*\|_2 = \|v^*\|_2$ and $v^* = v_0^* \mathbf{1}_n$, we have

$$d_m d_n u_{\min}^{*2} \leq \|u^*\|_2^2 = \sum_{k=1}^{n} v_k^{*2} = d_f v_0^{*2} \leq d_m d_n u_{\max}^{*2},$$

and so

$$u_{\min}^* \leq v_0^* \sqrt{\frac{d_f}{d_m d_n}} \leq u_{\max}^*.$$

Since $d_f = d_m d_n$ by Assumption 2.1, we find $u_{\min}^* \leq v_0^* \leq u_{\max}^*$. This implies $w_{\min}^* = u_{\min}^*$ and $w_{\max}^* = u_{\max}^*$, and therefore $\kappa(w^*) = \frac{u_{\max}^*}{u_{\min}^*}$. Hence,

$$1 \leq \frac{v_0^*}{u_{\min}^*} \leq \kappa(w^*).$$

Furthermore, since a background pixel must satisfy $u_h v_k \in [X_{\text{black}}, X_{\text{white}}]$ for all $(h, k) \in \Omega$, we have $u_{\min}^* v_0^* \geq X_{\text{black}}$ and $u_{\max}^* v_0^* \leq X_{\text{white}}$, and therefore

$$\kappa(w^*) \leq \frac{X_{\text{white}}}{X_{\text{black}}}. \tag{2.6}$$

Taking $[X_{\text{black}}, X_{\text{white}}] = [\Delta X, 255 + \Delta X]$ with $\Delta X \in \mathbb{R}_{++}$, as in Remark 2.1, we obtain $1 \leq \kappa(w^*) \leq 1 + \frac{255}{\Delta X}$. Therefore, taking ΔX large enough leads to

$$\kappa(w^*) \downarrow 1. \tag{2.7}$$

2) Parameter c: Notice that the identifiability condition (2.4) depends on a parameter c. This parameter is defined in [11] to be a value in the interval $(0, 1]$ such that the following holds:

$$\bar{S}_{\bar{h}\bar{k}} + w^*_{\bar{h}} w^*_{\bar{k}} > c w^{*2}_{\min}, \quad \bar{h}, \bar{k} \in \{1, 2, \ldots, m+n\}, \tag{2.8}$$

where $\bar{X} = \bar{S} + ww^\top$ and

$$\bar{S} = \begin{bmatrix} 0_{m\times m} & S \\ S^\top & 0_{n\times n} \end{bmatrix} \in \mathbb{R}^{(m+n)\times(m+n)}.$$

The elements of $\bar{X}$ therefore take on four forms:

1. $\bar{h}, \bar{k} \leq m$: We have $\bar{X}_{\bar{h}\bar{k}} = u^*_{\bar{h}} u^*_{\bar{k}} \geq u^{*2}_{\min}$.
2. $\bar{h}, \bar{k} > m$: We have $\bar{X}_{\bar{h}\bar{k}} = v^*_{\bar{h}-m} v^*_{\bar{k}-m} = v^{*2}_0 \geq u^{*2}_{\min}$.
3. $\bar{h} \leq m < \bar{k}$: We have $\bar{X}_{\bar{h}\bar{k}} = (S + u^* v^{*\top})_{\bar{h},\bar{k}-m} = X_{\bar{h},\bar{k}-m} \geq X_{\text{black}} = u^*_{\min} v^*_0 \geq u^{*2}_{\min}$, where $u^*_{\min} v^*_0 = X_{\text{black}}$ by Assumption 2.1.
4. $\bar{k} \leq m < \bar{h}$: Analogous to the case above, we again find $\bar{X}_{\bar{h}\bar{k}} \geq u^{*2}_{\min}$.

Since $\bar{S}_{\bar{h}\bar{k}} + w^*_{\bar{h}} w^*_{\bar{k}} = \bar{X}_{\bar{h}\bar{k}} \geq u^{*2}_{\min} = w^{*2}_{\min}$ for all $(\bar{h}, \bar{k})$, we find that (2.8) is satisfied for $c < 1$. Therefore, we can choose

$$c \uparrow 1. \tag{2.9}$$

3) Foreground graph: Consider the graph $\mathcal{G}_{m,n}(F)$ and let us denote the degree of a vertex in $\mathcal{G}_{m,n}(F)$ as $\deg(\cdot, F)$. Note that $\deg(h, F)$, $h \in V_u = \{1, 2, \ldots, m\}$, exactly equals the number of frames in which pixel h appears as foreground. Since, by Assumption 2.1, the maximum number of frames in which any single pixel appears as foreground is p_f frames, we have

$$\max\{\deg(h, F) : h \in V_u\} = p_f.$$

Next, we note that $\deg(m+k, F)$, $m+k \in V_v = \{m+1, m+2, \ldots, m+n\}$, exactly equals the number of foreground pixels in frame k. By Assumption 2.1, at least one frame contains the entire object, and therefore the maximum number of foreground pixels in any given frame is

$$\max\{\deg(m+k, F) : m+k \in V_v\} = p_m p_n.$$

Therefore, we find that the maximum degree of the foreground graph becomes

$$\Delta(\mathcal{G}_{m,n}(F)) = \max\{p_f, p_m p_n\}. \tag{2.10}$$

4) Background graph: Consider the graph $\mathcal{G}_{m,n}(B)$ and let us denote the degree of a vertex in $\mathcal{G}_{m,n}(B)$ as $\deg(\cdot, B)$. Since F and G are complements with respect to Ω, we have that $\mathcal{G}_{m,n}(F)$ and $\mathcal{G}_{m,n}(B)$ are bipartite complements of one another. Hence, it must be that

$$\begin{aligned} |V_u| =&\ \deg(m+k, F) + \deg(m+k, B), \\ |V_v| =&\ \deg(h, F) + \deg(h, B), \end{aligned}$$

for all $h \in V_u$ and $m + k \in V_v$. This, together with the analysis of the foreground graph above, yields

$$\min\{\deg(h, B) : h \in V_u\} = d_f - p_f,$$
$$\min\{\deg(m + k, B) : m + k \in V_v\} = d_m d_n - p_m p_n.$$

Therefore, we find that the minimum degree of the background graph becomes

$$\delta(\mathcal{G}_{m,n}(B)) = \min\{d_f - p_f, d_m d_n - p_m p_n\}. \tag{2.11}$$

Combining the results of the four computations above by substituting (2.7), (2.9), (2.10), and (2.11) into (2.4), we find that the identifiability condition is equivalent to

$$\min\{d_f - p_f, d_m d_n - p_m p_n\} > 48 \max\{p_f, p_m p_n\}. \tag{2.12}$$

Since $d_f = d_m d_n =: d$, we find $\min\{d_f - p_f, d_m d_n - p_m p_n\} = d - \max\{p_f, p_m p_n\}$, so this gives

$$d > 49 \max\{p_f, p_m p_n\}.$$

This is equivalent to the proposed set of inequalities (2.5). Hence, the conditions we provide relating the video length to the size and speed of the object are seen to be necessary and sufficient, as desired. □

Remark 2.2 (Constant trajectory). Take the special case of an object moving horizontally at a constant speed of $\dot{x} \in \mathbb{R}_{++}$ pixels per frame and vertically at a constant speed of $\dot{y} \in \mathbb{R}_{++}$ pixels per frame. Then, the number of frames in which any single pixel can be considered as foreground is no more than $\lceil \frac{p_n}{\dot{x}} \rceil$, and is also no more than $\lceil \frac{p_m}{\dot{y}} \rceil$. Assuming the object moves a sufficient distance so as to not obscure any part of the background for the entirety of the video, we have $d_f > \min\{\lceil \frac{p_n}{\dot{x}} \rceil, \lceil \frac{p_m}{\dot{y}} \rceil\}$, so one of the two proposed bounds is active. Hence, the maximum number of frames in which a single pixel appears as foreground becomes

$$p_f = \min\left\{\left\lceil \frac{p_n}{\dot{x}} \right\rceil, \left\lceil \frac{p_m}{\dot{y}} \right\rceil\right\}, \tag{2.13}$$

giving bounds directly in terms of the object's size and speed.

Theorem 2.2 provides us necessary and sufficient conditions to guarantee the satisfaction of the identifiability condition (2.4) in terms of a rectangular object's size and trajectory in relation to the resolution and length of the video. As one's intuition may predict, smaller rectangles and longer videos relax these conditions, indicating that videos with small moving objects and many frames are inherently easier to achieve globally optimal video segmentation. Together with Theorem 2.1, we can provide deterministic guarantees that the optimization problem (2.2) used to decompose a video has benign landscape, and that the resulting decomposition is unique and globally optimal. These concepts are showcased in the following video segmentation example.

2.6 Simulation Results

In this section, we perform moving object segmentation via NRPCA on an example video in an effort to corroborate the two main results given in Theorems 2.1 and 2.2. For this experiment, we recorded five minutes of surveillance video on the UC Berkeley campus, instructing volunteer human subjects to walk up and down a set of stairs, acting as the moving object to segment. The original video is $d_f = 19623$ frames long with a resolution of $d_m = 1920$ pixels by $d_n = 1080$ pixels. We preprocess the video data so that $d_m d_n = d_f$, as described in Assumption 2.1 and Remark 2.1. Therefore, the NRPCA problem takes on approximately $2d_f$ variables, whereas the popular convex PCP segmentation approach would optimize over an astronomical d_f^2 variables after lifting the problem to higher dimensions. We also shift the pixel values in X from $\{0, 1, \dots, 255\}$ to the interval $\mathcal{X} = [5000, 5255]$ in order to guarantee $\kappa(w^*)$ is sufficiently close to unity (in this case, $\kappa(w^*) \in [1, 1.05]$ by (2.6)).

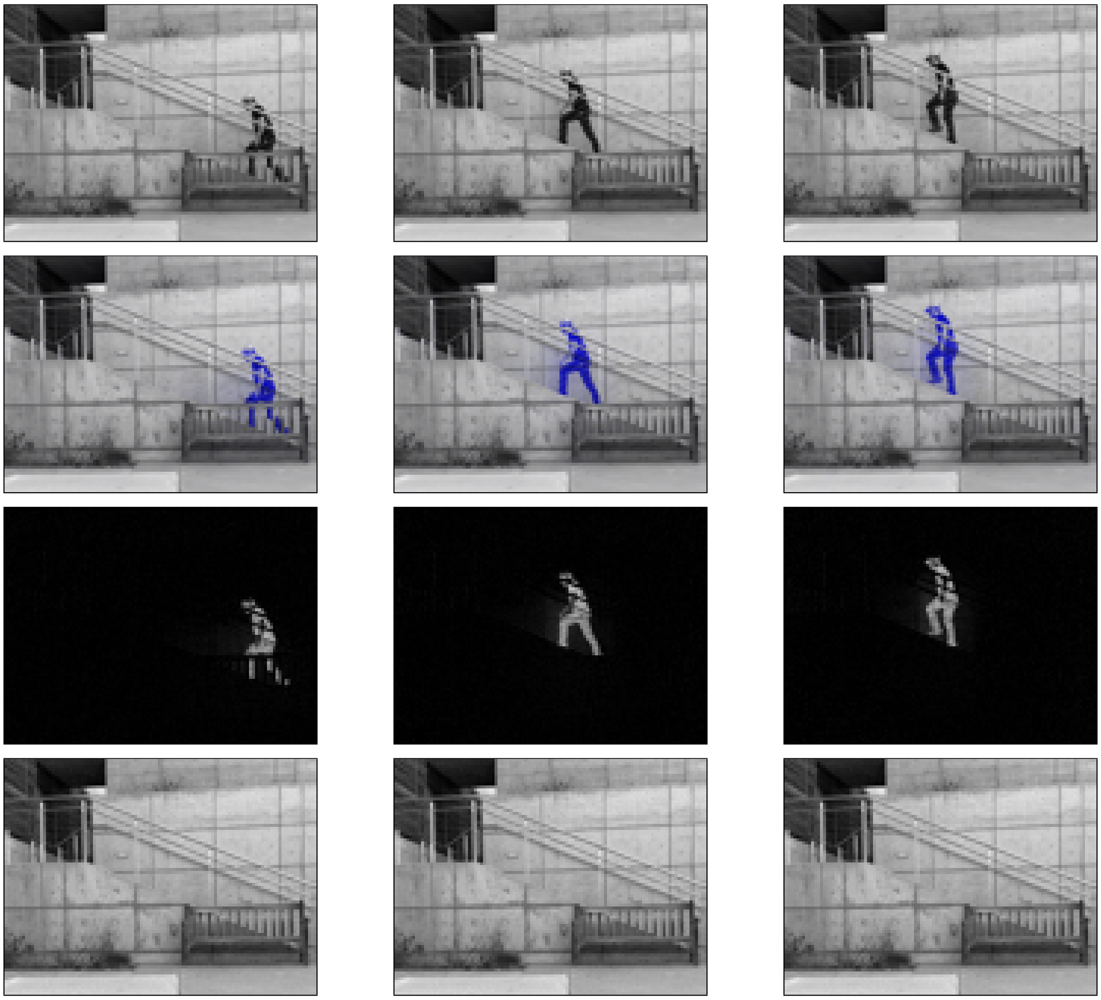
In order to solve the NRPCA problem, we set $\lambda = 1$ and initialize a point $w_0 = (u_0, \mathbf{1}_n)$, with each element of $u_0 \in \mathbb{R}^m_{++}$ drawn uniformly at random from the half-normal distribution. We then iterate using stochastic gradient descent with a learning rate of $\alpha = 10^{-4}$ and momentum coefficient of $\beta = 0.9$, with a projection onto the nonnegative orthant at each step of the algorithm. We find that 5000 iterations of this algorithm takes under 4 seconds to solve the problem on a standard laptop, and that convergence to within a small neighborhood is consistently achieved. We now demonstrate the use of our main results in this experiment.

Three frames from the video are shown in the columns of Fig. 2.2. These frames have been cropped vertically (but not horizontally) after performing the video segmentation, in order to enlarge the moving object and to save space. An *a priori* estimate of the relative object size, for example in the uncropped version of frame (c) in Fig. 2.2, shows that a rectangle of $p_m \approx \frac{1}{7} d_m$ and $p_n \approx \frac{1}{8} d_n$ should embed the object. Furthermore, the speed at which the subjects walk up the stairs is approximated to be $\dot{x} \approx \frac{1}{500} d_n$ pixels per frame, and therefore (2.13) can be used to approximate p_f for a single trajectory up the stairs. For the full video (with multiple trajectories up and down), we estimate this value to be $p_f \approx \frac{1}{70} d_f$. Hence, Theorem 2.2 is satisfied by our approximations. Furthermore, it is clear that Theorem 2.1 applies, and therefore we expect both conditions (2.3) and (2.4) to be satisfied, yielding global optimality guarantees for our video segmentation. After applying the preprocessing method described in Remark 2.1, we solve for w^*. An *a posteriori* computation gives the true values of $p_m p_n = \frac{1}{100} d_m d_n$ and $p_f = \frac{1}{72} d_f$, satisfying (2.5) as expected. The original identifiability condition (2.4) is also found to be satisfied with $\delta(\mathcal{G}_{m,n}(B)) = 19352$, $\Delta(\mathcal{G}_{m,n}(F)) = 271$, and $\kappa(w^*) = 1.05$. The resulting segmentation defined by w^* is shown in Fig. 2.2.

To empirically validate the theoretical absence of spurious local solutions in this segmentation problem, we ran the optimization $N = 1000$ times, obtaining the solution set $\mathcal{W}^* = \{w^{*(1)}, w^{*(2)}, \dots, w^{*(N)}\}$. Each run of the stochastic gradient descent algorithm used a random initial condition $w_0 = (u_0, \mathbf{1}_n)$ with the elements of u_0 again drawn uniformly from the half-normal distribution. We found the maximum relative distance between the resulting solutions to be

$$\max \left\{ \frac{\|w^* - w\|_2}{\|w^*\|_2} : w \in \mathcal{W}^* \right\} = 0.0023,$$

indicating a 100% success rate at converging to the same minimum w^*, as expected.



(a) (b) (c)

Figure 2.2: Three frames, (a), (b), and (c), of the stair walking video. The first row shows the original frames with the second row showing a color overlay of their segmentations. The third and fourth rows show each frame's foreground mask and its extracted background, respectively. The performance is visually pleasing, and the algorithm accurately learned the "difficult" parts of the background, e.g., the bench and handrail, which create thin lines obscuring parts of the subject and can be difficult to distinguish from the moving object even by eye.

Chapter 3

Robustness Guarantees for Neural Networks

3.1 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 [25, 26], data-driven system identification [27, 28, 29], and adaptive and self-learning control [30, 31]. 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 [32, 33] and the power grid [34, 35, 36]. 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 [37, 38, 39, 40, 41, 42, 43, 44]. 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 [37, 43]. From this perspective, certification amounts to proving that the image of an input uncertainty set is contained within a prescribed safe set. However, even when the input uncertainty set is convex, its output set may be nonconvex, which renders the certification an NP-complete nonconvex optimization problem [45, 40]. To overcome this issue, researchers have proposed various relaxations to over-approximate the output set 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 (e.g., [37]).

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 [38, 42].

3.1.1 Partition-Based Certification

In this work, we focus on feedforward ReLU networks, which are popular due to their simplicity, fast training speeds, and non-vanishing gradient property [39]. 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 [46] and deep learning [47], 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.

Previous works that apply partitioning to network certification include [39], 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 resulting outer approximations. The authors of [43] 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.

3.1.2 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 work 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.

3.1.3 Organization

This chapter is organized as follows. Section 3.2 introduces the robustness certification problem and linear program relaxation. In Section 3.3, we introduce the notion of partitioning and analyze its properties when applied to the robustness certification of ReLU networks. In Section 3.4, we further develop the theory to study the optimality of partitions and propose an optimal partitioning strategy. Illustrative examples are provided in Section 3.5.

3.2 Problem Statement

3.2.1 Network Description

Consider a K-layer ReLU neural network defined by

$$\begin{aligned} x^{[0]} &= x, \\ \hat{z}^{[k]} &= W^{[k-1]}x^{[k-1]} + b^{[k-1]}, \\ x^{[k]} &= \operatorname{ReLU}(\hat{z}^{[k]}), \\ z &= x^{[K]}, \end{aligned} \tag{3.1}$$

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 kth 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 kth 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 (3.1).

3.2.2 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 $\epsilon > 0$ [37, 38].

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., $\hat{z}^{[k]} = l^{[k]}$ for some $x \in \mathcal{X}$ and similarly for the upper bound $u^{[k]}$. From these bounds, we relax the kth ReLU constraint in (3.1) to its convex envelope, which leads to a relaxed ReLU constraint set associated with the kth layer:

$$\begin{aligned} \mathcal{N}^{[k]} = \{&(x^{[k-1]}, x^{[k]}) \in \mathbb{R}^{n_{k-1}} \times \mathbb{R}^{n_k} : \\ &x^{[k]} \leq u^{[k]} \odot (\hat{z}^{[k]} - l^{[k]}) \oslash (u^{[k]} - l^{[k]}), \\ &x^{[k]} \geq 0,\ x^{[k]} \geq \hat{z}^{[k]},\ \hat{z}^{[k]} = W^{[k-1]}x^{[k-1]}\}. \end{aligned} \tag{3.2}$$

Define the *relaxed network constraint set* as

$$\mathcal{N} = \{(x, z) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_z} : (x, x^{[1]}) \in \mathcal{N}^{[1]},\ (x^{[1]}, x^{[2]}) \in \mathcal{N}^{[2]}, \dots, (x^{[K-1]}, z) \in \mathcal{N}^{[K]}\}. \quad (3.3)$$

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 3.1. 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. 3.1 for this case.

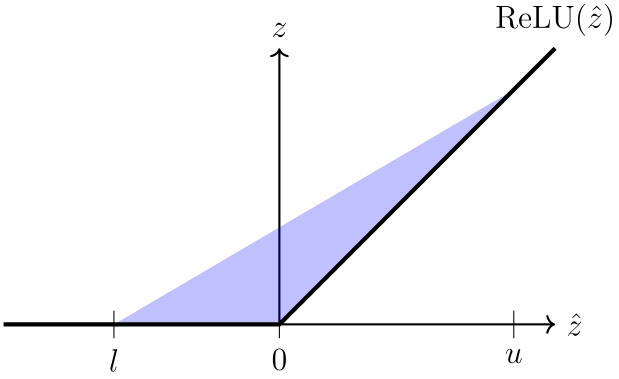


Figure 3.1: Relaxed ReLU constraint set at a single neuron for a one-layer network. The convex envelope $\mathcal{N}$ is shaded.

Remark 3.2. Consider a one-layer ReLU constraint relaxed according to (3.2). 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 $C \in \mathbb{R}^{n_S \times n_z}$ and $d \in \mathbb{R}^{n_S}$ are given. An output $z \in \mathcal{S}$ is said to be *safe*.

3.2.3 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

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

where $c_i^\top$ is the ith 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. 3.2.

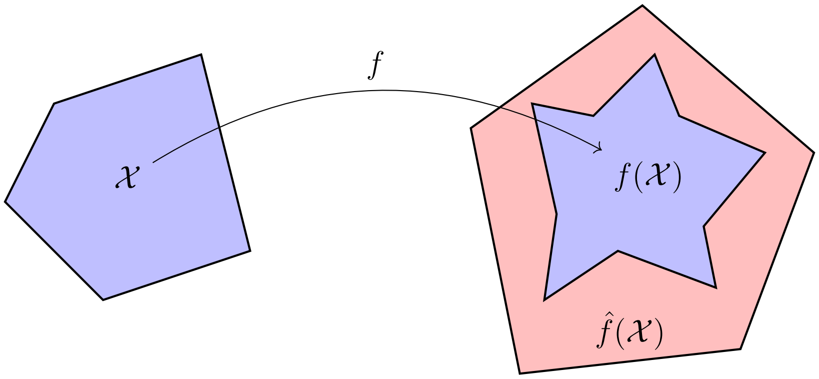


Figure 3.2: The convex outer approximation of the nonconvex set $f(\mathcal{X})$ is $\hat{f}(\mathcal{X})$. If the outer approximation is safe, i.e., $\hat{f}(\mathcal{X}) \subseteq \mathcal{S}$, then so is $f(\mathcal{X})$.

The robustness certification problem can be written as

$$f^*(\mathcal{X}) = \sup\{c^\top z : z = f(x),\ x \in \mathcal{X}\}. \tag{3.4}$$

The nonconvexity of (3.4) 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 [37]:

$$\hat{f}^*(\mathcal{X}) = \sup\{c^\top z : (x, z) \in \mathcal{N},\ x \in \mathcal{X}\}. \tag{3.5}$$

The suprema in (3.4) and (3.5) are assumed to be attained.

Due to the relaxation introduced in (3.5), it holds that

$$f^*(\mathcal{X}) \leq \hat{f}^*(\mathcal{X}). \tag{3.6}$$

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 work is dedicated to optimally tightening the bound (3.6) while maintaining the advantageous convexity and computational properties of the LP relaxation.

3.3 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 (3.4) can be obtained. In particular, the partitioning method yields a valid relaxation of (3.4).

3.3.1 Validation of Partitioned Relaxations

Definition 3.1 (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 jth *input part*.

Proposition 3.1 (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*

$$f^*(\mathcal{X}) \leq \max_{j \in \{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)}). \tag{3.7}$$

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

$$f^*(\mathcal{X}) > \hat{f}^*(\mathcal{X}^{(j)}) \text{ for all } j \in \{1, 2, \dots, p\}. \tag{3.8}$$

Let (x^*, z^*) denote an optimal solution to the unrelaxed problem (3.4), i.e., $x^* \in \mathcal{X}$, $z^* = f(x^*)$, and

$$c^\top z^* = f^*(\mathcal{X}). \tag{3.9}$$

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^*th subproblem and the final inequality is due to (3.8). This contradicts the optimality of (x^*, z^*) given in (3.9). Hence, (3.7) must hold. □

3.3.2 Tightening of the LP Relaxation

The objective is to show that by partitioning the input uncertainty set, the linear program relaxation bound in (3.6) is improved. The result will be presented for one-layer networks for simplicity, but the conclusion naturally generalizes to multi-layer networks.

Proposition 3.2 (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 jth 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*

$$\max_{j\in\{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)}) \leq \hat{f}^*(\mathcal{X}). \tag{3.10}$$

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 $i \in \{1,2,\dots,n_z\}$:

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

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

$$\begin{aligned} g_i^* &\geq a_i l_i' + b_i \\ &= \left(\frac{u_i}{u_i - l_i} - \frac{u_i'}{u_i' - l_i'} \right) l_i' + \left(\frac{u_i' l_i'}{u_i' - l_i'} - \frac{u_i l_i}{u_i - l_i} \right) \\ &= \frac{u_i}{u_i - l_i} (l_i' - l_i) \\ &\geq 0, \end{aligned}$$

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

$$\begin{aligned} g_i^* &\geq a_i u_i' + b_i \\ &= \left(\frac{u_i}{u_i - l_i} - \frac{u_i'}{u_i' - l_i'} \right) u_i' + \left(\frac{u_i' l_i'}{u_i' - l_i'} - \frac{u_i l_i}{u_i - l_i} \right) \\ &= \frac{u_i}{u_i - l_i}(u_i' - l_i) - u_i' \\ &= \frac{u_i' - u_i}{u_i - l_i} l_i \\ &\geq 0, \end{aligned}$$

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 (3.10). □

3.3.3 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 3.2 (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 3.3. 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 3.1 (Lipschitz LP relaxations)**.** *All K-layer neural networks defined by* (3.1) *have L-Lipschitz continuous LP relaxations.*

Proof. Let $x_1, x_2 \in \mathcal{X}$. By Theorem 2.4 in [48], 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 3.1 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 3.3 (Diameter)**.** Given a bounded 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 3.3 (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 $\{\mathcal{X}^{(j)} \subseteq \mathcal{X} : j \in \{1, 2, \ldots, p\}\}$ be a partition of $\mathcal{X}$. Denote the largest diameter among the input parts by d^*, i.e., $d^* = \max\{d(\mathcal{X}^{(j)}) : j \in \{1, 2, \ldots, p\}\}$. Then, there exists a finite constant $L \in \mathbb{R}$ such that*

$$\left| f^*(\mathcal{X}) - \max_{j \in \{1,2,\ldots,p\}} \hat{f}^*(\mathcal{X}^{(j)}) \right| \leq L d^*. \tag{3.11}$$

Proof. First, let j^* be the index corresponding to the partition subproblem with the highest objective value: $\hat{f}^*(\mathcal{X}^{(j^*)}) = \max_{j \in \{1,2,\ldots,p\}} \hat{f}^*(\mathcal{X}^{(j)})$. By Lemma 3.1, 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

$$\begin{aligned} d^* &= \max\{d(\mathcal{X}^{(j)}) : j \in \{1, 2, \ldots, p\}\} \\ &\geq d(\mathcal{X}^{(j^*)}) \\ &\geq \|x_1 - x_2\|_2 \text{ for all } x_1, x_2 \in \mathcal{X}^{(j^*)}. \end{aligned}$$

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 (3.5) over the j^*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

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

Now, since $(\bar{x}, \bar{z})$ is feasible for the unrelaxed problem (3.4) and the relaxation $\hat{f}^*(\mathcal{X}^{(j^*)})$ provides an upper bound on the unrelaxed problem by Proposition 3.1, 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,

$$\begin{aligned} Ld^* &\geq |f^*(\mathcal{X}) - \hat{f}^*(\mathcal{X}^{(j^*)})| \\ &= \left| f^*(\mathcal{X}) - \max_{j \in \{1,2,\dots,p\}} \hat{f}^*(\mathcal{X}^{(j)}) \right|, \end{aligned}$$

as desired. □

In the case that the network has a Lipschitz relaxation that is uniform over all possible input parts, Proposition 3.3 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.

3.4 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.

3.4.1 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 3.4 (Motivating partition)**.** *Consider a one-layer feedforward neural network and denote the ith 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*

$$\begin{aligned} \mathcal{X}^{(j)} = \{x \in \mathcal{X} : w_i^\top x \geq 0 \text{ for all } i \text{ such that } j_i = 1, \\ w_i^\top x < 0 \text{ for all } i \text{ such that } j_i = 0\}. \end{aligned} \tag{3.12}$$

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

$$f^*(\mathcal{X}) = \max_{j \in \mathcal{J}} \hat{f}^*(\mathcal{X}^{(j)}). \tag{3.13}$$

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 (3.12), 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 (3.13). 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 (3.3) for the jth 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 $z = \text{ReLU}(Wx)$. Hence, the LP over the jth input part yields that

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

Since j was chosen arbitrarily, it holds that

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

Since $f^*(\mathcal{X}) \leq \max_{j\in\mathcal{J}} \hat{f}^*(\mathcal{X}^{(j)})$ by the relaxation bound (3.7), the equality (3.13) holds, as desired. □

The partition introduced in Proposition 3.4 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 work, the robustness certification problem can be solved exactly via a finite number of linear program subproblems. Second, Proposition 3.4 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. 3.3). Motivated by this structure, we develop an optimal two-part partitioning scheme in the next section.

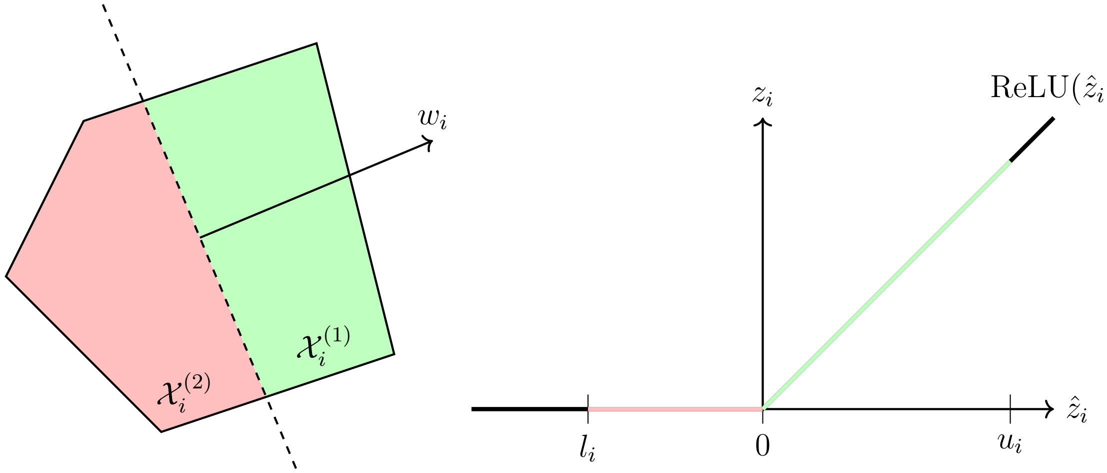


Figure 3.3: Partitioning based on row $w_i^\top$ of the weight matrix. This partition results in an exact ReLU constraint in coordinate i over the two resulting input parts $\mathcal{X}_i^{(1)} = \{x \in \mathcal{X} : w_i^\top x \geq 0\}$ and $\mathcal{X}_i^{(2)} = \mathcal{X} \setminus \mathcal{X}_i^{(1)}$.

3.4.2 Optimal Partitioning Scheme

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

Theorem 3.1 (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 $\Delta f^*(\mathcal{X}) := \hat{f}^*(\mathcal{X}) - f^*(\mathcal{X})$. Let $(\tilde{x}^*, \tilde{z}^*)$ and (x^*, z^*) be optimal solutions for the relaxation $\hat{f}^*(\mathcal{X})$ and the unrelaxed problem $f^*(\mathcal{X})$, respectively. Also, let $\|\cdot\|$ be a norm on $\mathbb{R}^{n_x}$. Then, there exists $\epsilon \in \mathbb{R}$ such that $\|\tilde{x}^* - x^*\| \leq \epsilon$ and*

$$\Delta f^*(\mathcal{X}) \leq \sum_{i=1}^{n_z} \left(\mathrm{ReLU}(c_i) \frac{u_i}{u_i - l_i} (\min\{\epsilon \|w_i\|_*, u_i\} - l_i) + \mathrm{ReLU}(-c_i) \min\{\epsilon \|w_i\|_*, u_i\} \right), \tag{3.14}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

Proof. First, since $\mathcal{X}$ is bounded, there exists $\epsilon \geq 0$ such that $\|\tilde{x}^* - x^*\| \leq \epsilon$. The definitions of $(\tilde{x}^*, \tilde{z}^*)$ and (x^*, z^*) give that

$$\Delta f^*(\mathcal{X}) = \sum_{i=1}^{n_z} c_i(\tilde{z}_i^* - z_i^*) \leq \sum_{i=1}^{n_z} \Delta f_i^*, \tag{3.15}$$

where

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

for all $i \in \{1, 2, \dots, n_z\}$. Note that

$$\Delta f_i^* = \sup\Bigg\{c_i(\tilde{z}_i - z_i) : z_i = \mathrm{ReLU}(\hat{z}_i),\ \tilde{z}_i \geq 0,\ \tilde{z}_i \geq \hat{\tilde{z}}_i,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(\hat{\tilde{z}}_i - l_i), \\ \|\tilde{x}^* - x^*\| \leq \epsilon,\ \hat{z} = Wx,\ \hat{\tilde{z}} = W\tilde{x},\ x, \tilde{x} \in \mathcal{X}\Bigg\}.$$

If $x, \tilde{x} \in \mathcal{X}$ and $\hat{z}, \hat{\tilde{z}}$ satisfy $\hat{z} = Wx$, $\hat{\tilde{z}} = W\tilde{x}$, and $\|\tilde{x} - x\| \leq \epsilon$, then they satisfy $l \leq \hat{z}, \hat{\tilde{z}} \leq u$ and $|\hat{\tilde{z}}_i - \hat{z}_i| = |w_i^\top(\tilde{x} - x)| \leq \|w_i\|_* \|\tilde{x} - x\| \leq \epsilon\|w_i\|_*$ for all $i \in \{1, 2, \dots, n_z\}$ by the Cauchy-Schwarz inequality for dual norms. Therefore,

$$\begin{aligned}\Delta f_i^* \leq\ & \sup\Bigg\{c_i(\tilde{z}_i - z_i) : z_i = \mathrm{ReLU}(\hat{z}_i),\ \tilde{z}_i \geq 0,\ \tilde{z}_i \geq \hat{\tilde{z}}_i,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(\hat{\tilde{z}}_i - l_i), \\ & l \leq \hat{z}, \hat{\tilde{z}} \leq u,\ |\hat{\tilde{z}}_k - \hat{z}_k| \leq \epsilon\|w_k\|_* \text{ for all } k \in \{1, 2, \dots, n_z\},\ \hat{z}, \hat{\tilde{z}} \in \mathbb{R}^{n_z}\Bigg\} \\ =\ & \sup\Bigg\{c_i(\tilde{z}_i - z_i) : z_i = \mathrm{ReLU}(\hat{z}_i),\ \tilde{z}_i \geq 0,\ \tilde{z}_i \geq \hat{\tilde{z}}_i,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(\hat{\tilde{z}}_i - l_i), \\ & l_i \leq \hat{z}_i, \hat{\tilde{z}}_i \leq u_i,\ |\hat{\tilde{z}}_i - \hat{z}_i| \leq \epsilon\|w_i\|_*,\ \hat{z}_i, \hat{\tilde{z}}_i \in \mathbb{R}\Bigg\}.\end{aligned}$$

For $c_i \geq 0$, the above inequality yields that

$$\Delta f_i^* \leq c_i \sup\Bigg\{\tilde{z}_i - z_i : z_i = \mathrm{ReLU}(\hat{z}_i),\ \tilde{z}_i \geq 0,\ \tilde{z}_i \geq \hat{\tilde{z}}_i,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(\hat{\tilde{z}}_i - l_i), \\ l_i \leq \hat{z}_i, \hat{\tilde{z}}_i \leq u_i,\ |\hat{\tilde{z}}_i - \hat{z}_i| \leq \epsilon\|w_i\|_*,\ \hat{z}_i, \hat{\tilde{z}}_i \in \mathbb{R}\Bigg\}.$$

The optimal solution to the above supremum is readily found by comparing the line $\tilde{z}_i = \frac{u_i}{u_i - l_i}(\hat{\tilde{z}}_i - l_i)$ to the function $z_i = \mathrm{ReLU}(\hat{z}_i)$ over $\hat{\tilde{z}}_i, \hat{z}_i \in [l_i, u_i]$. In particular, the maximum distance between $\tilde{z}_i$ and z_i on the above feasible set occurs when $z_i = \hat{z}_i = 0$, $\hat{\tilde{z}}_i = \epsilon\|w_i\|_*$, and $\tilde{z}_i = \frac{u_i}{u_i - l_i}(\epsilon\|w_i\|_* - l_i)$. Therefore, we find that

$$\Delta f_i^* \leq c_i \frac{u_i}{u_i - l_i}(\epsilon\|w_i\|_* - l_i), \tag{3.16}$$

for all $i \in \{1, 2, \ldots, n_z\}$ such that $c_i \geq 0$. We also note the trivial bound that $\tilde{z}_i - z_i \leq u_i$ on the feasible set of the above supremum, so that

$$\Delta f_i^* \leq c_i u_i = c_i \frac{u_i}{u_i - l_i}(u_i - l_i). \tag{3.17}$$

The inequalities (3.16) and (3.17) together imply that

$$\Delta f_i^* \leq c_i \frac{u_i}{u_i - l_i}(\min\{\epsilon \|w_i\|_*, u_i\} - l_i) \tag{3.18}$$

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

On the other hand, for all $i \in \{1, 2, \ldots, n_z\}$ such that $c_i < 0$, we have that

$$\Delta f_i^* \leq c_i \inf \Bigg\{ \tilde{z}_i - z_i : z_i = \mathrm{ReLU}(\hat{z}_i),\ \tilde{z}_i \geq 0,\ \tilde{z}_i \geq \hat{\tilde{z}}_i,\ \tilde{z}_i \leq \frac{u_i}{u_i - l_i}(\hat{\tilde{z}}_i - l_i), \\ l_i \leq \hat{z}_i, \hat{\tilde{z}}_i \leq u_i,\ |\hat{\tilde{z}}_i - \hat{z}_i| \leq \epsilon \|w_i\|_*,\ \hat{z}_i, \hat{\tilde{z}}_i \in \mathbb{R} \Bigg\}.$$

The optimal solution to the above infimum is readily found by comparing the line $\tilde{z}_i = 0$ to the function $z_i = \mathrm{ReLU}(\hat{z}_i)$ over $\hat{\tilde{z}}_i, \hat{z}_i \in [l_i, u_i]$. In particular, the minimum value of $\tilde{z}_i - z_i$ on the above feasible set occurs when $\tilde{z}_i = \hat{\tilde{z}}_i = 0$ and $z_i = \hat{z}_i = \epsilon \|w_i\|_*$. Therefore, we find that

$$\Delta f_i^* \leq -c_i \epsilon \|w_i\|_*, \tag{3.19}$$

for all $i \in \{1, 2, \ldots, n_z\}$ such that $c_i < 0$. We also note the trivial bound that $\tilde{z}_i - z_i \geq -u_i$ on the feasible set of the above infimum, so that

$$\Delta f_i^* \leq -c_i u_i. \tag{3.20}$$

The inequalities (3.19) and (3.20) together imply that

$$\Delta f_i^* \leq -c_i \min\{\epsilon \|w_i\|_*, u_i\} \tag{3.21}$$

for all $i \in \{1, 2, \ldots, n_z\}$ such that $c_i < 0$. Substituting (3.18) and (3.21) into (3.15) gives the desired bound (3.14). □

The value Δf_i^* can be interpreted as the worst-case relaxation error in coordinate i. From this perspective, Theorem 3.1 provides an upper bound on the overall worst-case relaxation error. In the case that $\tilde{x}^* \neq x^*$ and $\epsilon \|w_i\|_* \geq u_i$ for all $\epsilon \in \mathbb{R}$ such that $\|\tilde{x}^* - x^*\| \leq \epsilon$, for all $i \in \{1, 2, \ldots, n_z\}$, the bound (3.14) becomes

$$\Delta f^*(\mathcal{X}) \leq \sum_{i=1}^{n_z} |c_i| u_i.$$

This is the loosest the bound can ever be. On the contrary, if $\tilde{x}^* = x^*$, i.e., the relaxation and the true certification problem share an optimal input, then we conclude Theorem 3.1 holds for $\epsilon = 0$. Substituting this into (3.14) gives

$$\Delta f^*(\mathcal{X}) \leq -\sum_{i=1}^{n_z} \mathrm{ReLU}(c_i) \frac{u_i l_i}{u_i - l_i}. \tag{3.22}$$

Note that in practice we expect the condition $\epsilon = 0$ to hold, since a worst-case input to a neural network is likely to also be a worst-case input to the relaxed network. Therefore, for the remainder of this report, we take the worst-case relaxation bound to be that given by (3.22) to simplify the analysis. As shown in Corollary 3.1 that follows, the error bound (3.22) scales linearly as the input uncertainty set is made smaller.

Corollary 3.1 (Linear scaling of relaxation error)**.** *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 α, i.e.,*

$$\hat{f}^*(\tilde{\mathcal{X}}) - f^*(\mathcal{X}) \leq -\alpha \sum_{i=1}^{n_z} \mathrm{ReLU}(c_i) \frac{u_i l_i}{u_i - l_i}. \tag{3.23}$$

Proof. Since $\tilde{\mathcal{X}} \subseteq \mathcal{X}$, it holds that $f^*(\tilde{\mathcal{X}}) \leq f^*(\mathcal{X})$. Therefore, by (3.22) on $\tilde{\mathcal{X}}$ we have that

$$\begin{aligned} \hat{f}^*(\tilde{\mathcal{X}}) - f^*(\mathcal{X}) &\leq \hat{f}^*(\tilde{\mathcal{X}}) - f^*(\tilde{\mathcal{X}}) \\ &\leq -\sum_{i=1}^{n_z} \mathrm{ReLU}(c_i) \frac{\tilde{u}_i \tilde{l}_i}{\tilde{u}_i - \tilde{l}_i} \\ &= -\sum_{i=1}^{n_z} \mathrm{ReLU}(c_i) \frac{\alpha^2 u_i l_i}{\alpha u_i - \alpha l_i}. \end{aligned}$$

This completes the proof. □

We now focus on developing an optimal two-part partitioning scheme based on the worst-case relaxation bound (3.22). We start with the following lemma.

Lemma 3.2 (Two-part bound)**.** *Let $i \in \{1, 2, \dots, n_z\}$ 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 $\Delta f^*(\{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}\}) \coloneqq \max_{j \in \{1,2\}} \hat{f}^*(\mathcal{X}_i^{(j)}) - f^*(\mathcal{X})$. It holds that*

$$\Delta f^*(\{\mathcal{X}_i^{(1)}, \mathcal{X}_i^{(2)}\}) \leq -\sum_{\substack{k=1 \\ k \neq i}}^{n_z} \mathrm{ReLU}(c_k) \frac{u_k l_k}{u_k - l_k}. \tag{3.24}$$

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 (3.22) 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 3.2 established, we now present the optimal two-part partition.

Theorem 3.2 (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 bound in* (3.24) *is given by*

$$i^* \in \arg\min_{i\in\mathcal{I}} \mathrm{ReLU}(c_i)\frac{u_i l_i}{u_i - l_i}. \tag{3.25}$$

Proof. Minimizing the bound in (3.24) of Lemma 3.2 over the partition i gives rise to

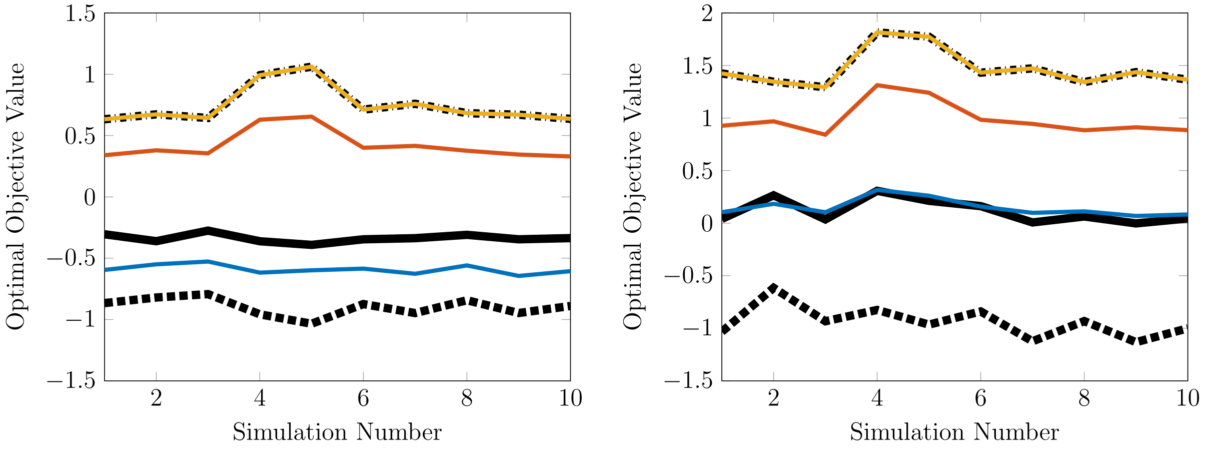
$$\min_{i\in\mathcal{I}}\left(-\sum_{\substack{k=1\\k\neq i}}^{n_z} \mathrm{ReLU}(c_k)\frac{u_k l_k}{u_k - l_k}\right) = -\sum_{k=1}^{n_z} \mathrm{ReLU}(c_k)\frac{u_k l_k}{u_k - l_k} + \min_{i\in\mathcal{I}} \mathrm{ReLU}(c_i)\frac{u_i l_i}{u_i - l_i},$$

as desired. □

Theorem 3.2 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 3.2 can be immediately extended to recursively choose the first $n_p > 1$ optimal partitions if it is preferable to perform more than just a two-part partition.

3.5 Simulation Results

Consider a one-layer classification network with four inputs and three outputs, trained on the celebrated Iris data set [49] 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. We solve the certification problem in its nonconvex form (via a multistart search), using an unpartitioned LP relaxation, and using two-part partitioned LP relaxations corresponding to each row of the weight matrix W. The results are shown in Fig. 3.4a. It can be observed that the partitioned LP corresponding to the optimal partition developed in Theorem 3.2 yields the best convex approximation of the true problem. Furthermore, ordering the rows by their suboptimality in (3.25) corresponds to the order of relaxation tightening. For instance, in Fig. 3.4, 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 relaxations do not issue a certificate of robustness, as the objective values are positive for each nominal input to be certified. 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. The same experiment using a two-layer network shows that the developed optimal partitioning scheme maintains the best convex approximation, albeit without guaranteed relaxation error bounds (see Fig. 3.4b).



(a) One-layer network.

(b) Two-layer network.

Figure 3.4: Objective values of the robustness certification problems at optimum for Iris classification. For the two-layer network, the optimal partitioning scheme is applied to only the ReLU constraints of the final layer. The lines correspond to: nonconvex problem via multistart (■■■), unpartitioned LP (▪╻▪╻), upper bound (3.24) for optimally partitioned LP (━━), optimally partitioned LP (──), suboptimally partitioned LP 1 (──), suboptimally partitioned LP 2 (──).

Chapter 4

Conclusions

In this report, we consider two applications of data-driven learning and decision making. First, we study the unsupervised extraction of a video's moving objects from its background via nonnegative robust principal component analysis. Although the optimization problem of interest is nonconvex, it exhibits benign landscape under certain criteria. We exploit this fact to develop conditions under which the video segmentation is unique and globally optimal. We derive these global optimality guarantees in terms of intuitive and meaningful parameters, such as the size and speed of the moving objects, as well as the length of the video. Furthermore, real video examples are given to illustrate the use of these criteria, and in what scenarios the problem's benign landscape holds.

In the second part of 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. The results demonstrate, both theoretically and experimentally, that partition-based certification procedures are capable of tightening relaxation errors with remarkable simplicity.

The contributions of this report show that high-performing and efficient algorithms need not be disjoint from theoretically understood and guaranteed results. In particular, the video segmentation process developed in the first part of this report showcases that realistic nonconvex learning methods can be provably solved to unique global optimality, allowing for the safe exploitation of their more efficient formulations. The second part demonstrates that the theoretical understanding and simplicity of linear programming can be utilized to efficiently and accurately certify the robustness of neural network decision-making algorithms, despite the nonconvexity of the underlying network structure. These contributions offer new building blocks to an emerging body of research, namely, the development of data-driven learning and decision-making algorithms with theoretically guaranteed optimality and robustness properties.

Bibliography

[1] B. G. Anderson and S. Sojoudi, "Global optimality guarantees for nonconvex unsupervised video segmentation," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 965–972, IEEE, 2019.

[2] B. G. Anderson, Z. Ma, J. Li, and S. Sojoudi, "Tightened convex relaxations for neural network robustness certification," *arXiv preprint arXiv:2004.00570*, 2020.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[4] T. Bouwmans and B. Garcia-Garcia, "Background subtraction in real applications: Challenges, current models and future directions," *arXiv preprint arXiv:1901.03577*, 2019.

[5] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 724–732, 2016.

[6] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, vol. 2, pp. 246–252, IEEE, 1999.

[7] K. Goyal and J. Singhai, "Review of background subtraction methods using gaussian mixture model for video surveillance systems," *Artificial Intelligence Review*, vol. 50, no. 2, pp. 241–259, 2018.

[8] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "A neural network approach to bayesian background modeling for video object segmentation.," in *VISAPP (1)*, pp. 474–479, 2006.

[9] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer science review*, vol. 11, pp. 31–66, 2014.

[10] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, 2011.

[11] S. Fattahi and S. Sojoudi, “Exact guarantees on the absence of spurious local minima for non-negative robust principal component analysis,” *arXiv preprint arXiv:1812.11466*, 2018.

[12] L. Zhang, Z. Chen, M. Zheng, and X. He, “Robust non-negative matrix factorization,” *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 2, pp. 192–200, 2011.

[13] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor, “Mahnmf: Manhattan non-negative matrix factorization,” *arXiv preprint arXiv:1207.3438*, 2012.

[14] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, “Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset,” *Computer Science Review*, vol. 23, pp. 1–71, 2017.

[15] Y. Chi, Y. M. Lu, and Y. Chen, “Nonconvex optimization meets low-rank matrix factorization: An overview,” *arXiv preprint arXiv:1809.09573*, 2018.

[16] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*, pp. 177–186, Springer, 2010.

[17] L. Bottou, “Stochastic gradient descent tricks,” in *Neural networks: Tricks of the trade*, pp. 421–436, Springer, 2012.

[18] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Advances in neural information processing systems*, pp. 315–323, 2013.

[19] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, “Failing to learn: autonomously identifying perception failures for self-driving cars,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3860–3867, 2018.

[20] X. Zou, X. Zhao, and Z. Chi, “An efficacious medical video segmentation approach with a moving camera,” in *2012 International Conference on Computerized Healthcare (ICCH)*, pp. 13–16, IEEE, 2012.

[21] S. Bhojanapalli, B. Neyshabur, and N. Srebro, “Global optimality of local search for low rank matrix recovery,” in *Advances in Neural Information Processing Systems*, pp. 3873–3881, 2016.

[22] R. Ge, J. D. Lee, and T. Ma, “Matrix completion has no spurious local minimum,” in *Advances in Neural Information Processing Systems*, pp. 2973–2981, 2016.

[23] C. Josz, Y. Ouyang, R. Zhang, J. Lavaei, and S. Sojoudi, “A theory on the absence of spurious solutions for nonconvex and nonsmooth optimization,” in *Advances in neural information processing systems*, pp. 2441–2449, 2018.

[24] R. Zhang, C. Josz, S. Sojoudi, and J. Lavaei, "How much restricted isometry is needed in nonconvex matrix recovery?," in *Advances in neural information processing systems*, pp. 5586–5597, 2018.

[25] 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.

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

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

[28] 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)*, pp. 4653–4660, IEEE, 2016.

[29] 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)*, pp. 4832–4839, IEEE, 2019.

[30] 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.

[31] 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, pp. 3527–3532, IEEE, 2001.

[32] 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.

[33] 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*, pp. 129–137, 2017.

[34] 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.

[35] 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.

[36] 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)*, pp. 1–6, IEEE, 2019.

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

[38] A. Raghunathan, J. Steinhardt, and P. S. Liang, "Semidefinite relaxations for certifying robustness to adversarial examples," in *Advances in Neural Information Processing Systems*, pp. 10877–10887, 2018.

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

[40] 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.

[41] 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*, pp. 4939–4948, 2018.

[42] 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.

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

[44] 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.

[45] 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*, pp. 97–117, Springer, 2017.

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

[47] 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*, pp. 2924–2932, 2014.

[48] 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.

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