Applications of Tropical Geometry in Deep Neural Networks

Thesis by

Motasem H. A. Alfarra

In Partial Fulfillment of the Requirements

For the Degree of

Masters of Science

in Electrical Engineering

King Abdullah University of Science and Technology
Thuwal, Kingdom of Saudi Arabia

April, 2020
The thesis of Motasem H. A. Alfarra is approved by the examination committee

Committee Chairperson: Bernard S. Ghanem
Committee Members: Bernard S. Ghanem, Wolfgang Heidrich, Xiangliang Zhang
Applications of Tropical Geometry in Deep Neural Networks
Motasem H. A. Alfarra

This thesis tackles the problem of understanding deep neural network with piecewise linear activation functions. We leverage tropical geometry, a relatively new field in algebraic geometry to characterize the decision boundaries of a single hidden layer neural network. This characterization is leveraged to understand, and reformulate three interesting applications related to deep neural network. First, we give a geometrical demonstration of the behaviour of the lottery ticket hypothesis. Moreover, we deploy the geometrical characterization of the decision boundaries to reformulate the network pruning problem. This new formulation aims to prune network parameters that are not contributing to the geometrical representation of the decision boundaries. In addition, we propose a dual view of adversarial attack that tackles both designing perturbations to the input image, and the equivalent perturbation to the decision boundaries.
ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my thesis advisor Prof. Bernard Ghanem who supported me through this journey. Prof. Ghanem convincingly guided and encouraged me to do the right thing even when the road got tough. Second, I would like to recognize the invaluable assistance to my dear colleague Adel A. Bibi who continuously helped me through this journey. Without their limitless support, passionate and guidance, I would not have finished this work.

In addition, I would like to thank all of my colleagues in the Image and Video Understanding Lab in King Abdullah University of Science and Technology with whom I had this great experience. Also, I would like to pay my special regards to Modar M. Alfadly for his assistance through this journey.

Finally, I am indebted to my mother, father, the rest of my extended family, and my friends who gave me the warmest emotional support. I would not have accomplished this without their precious presence.
TABLE OF CONTENTS

- Examination Committee Page 2
- Copyright 3
- Abstract 4
- Acknowledgements 5
- List of Figures 8
- 1 Introduction 10
- 2 Overview on Tropical Geometry 13
- 3 Functional and Geometrical Representations of Deep Neural Networks in Tropical Geometry 17
  - 3.1 Deep Neural Networks in Tropical Geometry 17
  - 3.2 Decision Boundaries of DNNs as Polytopes 19
  - 3.3 Disclaimer 22
  - 3.4 Handling Biases 22
  - 3.5 Potential Applications 24
- 4 Topical Perspective to the Lottery Ticket Hypothesis 26
- 5 Tropical Network Prunning 29
- 6 Tropical Adversarial Attacks 35
- 7 Concluding Remarks 39
- References 40
  - A.0.1 Tropical View to the Lottery Ticket Hypothesis 44
  - A.0.2 Tropical Pruning 46
Appendices

B.1 Proof Of Theorem 2 .................................................. 50
B.2 Proof of Proposition 3 .................................................. 53
B.3 Optimization of Objective (5.2) of the Binary Classifier ............ 54
B.4 Optimization of Objective 5.4 of the Multi Class Classifier ............ 55
B.5 Algorithm for Solving 6.3 ............................................. 57
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td><strong>Duality between tropical hypersurface and dual subdivision.</strong> For each function $f$, we show both its tropical hypersurface $T(f)$, and the dual subdivision for its Newton polygon $\delta(f)$. Note that the tropical hypersurface is perpendicular to the edges of the dual subdivision.</td>
</tr>
<tr>
<td>3.1</td>
<td><strong>Decision Boundaries as Geometric Structures.</strong> The decision boundaries $B$ (in red) comprise two linear pieces separating classes $C_1$ and $C_2$. As per Theorem 2, the dual subdivision of this single hidden neural network is the convex hull between the zonotopes $Z_{G_1}$ and $Z_{G_2}$. The normals to the dual subdivision $\delta(R(x))$ are in one-to-one correspondence to the tropical hypersurface $T(R(x))$, which is a superset to the decision boundaries $B$. Note that some of the normals to $\delta(R(x))$ (in red) are parallel to the decision boundaries.</td>
</tr>
<tr>
<td>3.2</td>
<td><strong>Applications of Tropical Geometry in DNNs.</strong> Transforming the decision boundaries into its geometrical representation yields to three different applications in lottery ticket hypothesis, network pruning and adversarial attacks.</td>
</tr>
<tr>
<td>4.1</td>
<td><strong>Effect of Different Initializations on the Decision Boundaries Polytope.</strong> From left to right: training dataset, decision boundaries polytope of original network (before pruning), followed by the decision boundaries polytope for networks pruned at different pruning percentages across different initializations. Note that in the original polytope there are many more vertices than just 4 but they are very close to each other forming many small edges that are not visible in the figure.</td>
</tr>
<tr>
<td>5.1</td>
<td><strong>Tropical Pruning Pipeline.</strong> Pruning the $4^{th}$ node, or equivalently removing the two yellow vertices of zonotope $Z_{G_2}$ does not affect the decision boundaries polytope, which will lead to no change in accuracy.</td>
</tr>
<tr>
<td>5.2</td>
<td><strong>Results of Tropical Pruning.</strong> Pruning-accuracy plots for AlexNet (top) and VGG16 (bottom) trained on SVHN, CIFAR10, and CIFAR100, pruned with our tropical method and three other pruning methods.</td>
</tr>
</tbody>
</table>
6.1 **Dual View of Tropical Adversarial Attacks.** We show the effects of tropical adversarial attacks on a synthetic binary dataset at two different input points (in black). From left to right: the decision regions of the original and perturbed models, and decision boundaries polytopes (green for original and blue for perturbed).

6.2 **Effect of Tropical Adversarial Attacks on MNIST Dataset.** We show qualitative examples of adversarial attacks, produced by solving Problem (6.3), on two digits (8,9) from MNIST. From left to right, images are classified as [8,7,5,4] and [9,7,5,4] respectively.

A.1 **Effect of Different Initializations on the Decision Boundaries Polytope.** From left to right: training dataset, decision boundaries polytope of original network followed by the decision boundaries polytope during several iterations of pruning with different initializations.

A.2 **Comparison between the decision boundaries polytope and the polytopes representing the functional representation of the network.** First column: decision boundaries polytope $\delta(R(x))$ while the remainder of the columns are the zonotopes $\delta(H_1(x)), \delta(Q_1(x)), \delta(H_2(x))$ and $\delta(Q_2(x))$ respectively. Under varying pruning rate across the rows, it is to be observed that the changes that affected the dual subdivisions of the functional representations are far smaller compared to the decision boundaries polytope.

A.3 **Results of Tropical Pruning with Fine Tuning the Biases of the Classifier.** Tropical pruning applied on AlexNet and VGG16 trained on SVHN, CIFAR10, CIFAR100 against different pruning methods with fine tuning the biases of the classifier only.

A.4 **Results of Tropical Pruning with Fine Tuning the Biases of the Network.** Tropical pruning applied on AlexNet and VGG16 trained on SVHN, CIFAR10, CIFAR100 against different pruning methods with fine tuning the biases of the network.
Chapter 1

Introduction

Deep Neural Networks (DNNs) have defined the state of the art across a variety of research domains, such as image classification [1, 2], semantic segmentation [3], speech recognition [4], natural language processing [5, 6], and healthcare [7, 8]. Despite this success, a solid understanding of their boom is still ambiguous [9]. For instance, in an attempt to uncover the expressive power of DNNs, the authors in [10, 11] studied the complexity of functions computable by DNNs that have piecewise linear activations and derived a lower bound on the maximum number of linear regions. Moreover, a lot of work has been done to understand the behaviour of the decision boundaries of DNNs. To this regard, the authors in [12] showed that the smoothness of these decision boundaries and their curvature can play a vital role in network robustness. In addition, decision boundaries around perturbed examples were studied in [13], where it is shown that these boundaries do not resemble the boundaries around benign examples. Besides, the authors in [14] have shown that the decision boundaries of DNNs with width smaller than input the input dimension is unbounded. Also, under certain assumptions, the decision boundaries of the last fully connected layer converge to a linear SVM [15]. Recently, piece wise linear activation functions (e.g. ReLU) gained a lot of popularity since they helped in solving the gradient vanishing/exploding problem. As a result, a lot of work have been developed to analyze this class of DNNs as piecewise linear functions [16, 17]. These analysis were accomplished by looking to DNNs via the lens of different mathematical tools such as tropical geometry.

Tropical geometry is a relatively new field in algebraic geometry that analyzes
piecewise linear functions. This new branch of algebraic geometry demonstrated its potential, by transforming algebraic problems from piecewise linear nature to a combinatoric problem on general polytopes, on different applications such as linear programming [18], dynamic programming [19], multi-objective discrete optimization [20], enumerative geometry [21], and economics [22] [23] and more recently, analyzing DNNs [24]. For instance, the later showed an equivalency between the family of DNNs with piecewise linear activation functions and tropical rational functions, i.e. ratio between two multi-variate polynomials in tropical geometry. This equivalency allowed characterizing the complexity of DNNs by counting the number of linear regions, into which the function represented by the DNN can divide the input space. This was accomplished, with the means of tropical geometry, by counting the number of vertices of some polytope representation of the functional form of a DNN. It is worthwhile to mention that this study recovered the results of [10] with simpler analysis. However, it is still unclear how can this analogy affect practical computer vision tasks. In this thesis we extend the aforementioned results of [24] theoretically to shed lights into few practical tasks. In particular, we investigate the decision boundaries of a family of DNNs through the lens of tropical geometry. It turns out that one can represent a super set that contains the decision boundaries as a tropical hypersurface of a specific tropical polynomial. In addition, we derive a geometric representation for this super set for a network that is the composition of a linear layer followed a ReLU non-linearity followed by another linear layer.

This new characterization is leveraged to shed light on three interesting applications. First, we provide a novel geometric reaffirmation to the lottery ticket hypothesis phenomena [25]. To do so, we visualize the evolution of the geometrical representation of the decision boundaries of a DNN while pruning its parameters under different initializations which is now possible without the need for any forward passes through the network. Second, we deploy this characterization in network pruning. To this
regard, we design a geometrically inspired optimization problem to prune the parameters of a given network while preserving the geometrical representation of the decision boundaries. As a result, the pruned network will enjoy a minimally deviated decision boundaries from the decision boundaries of the original overparameterized network, and hence will have similar performance. We validate the proposed method by conducting extensive experiments with AlexNet \cite{alexnet} and VGG16 \cite{vgg} on SVHN \cite{svhn}, CIFAR10, and CIFAR 100 \cite{cifar100} datasets, in which 90% pruning rate can be achieved with a marginal drop in testing accuracy. Third, we deploy this fresh characterization of the decision boundaries to construct a new type of adversarial attack, namely \textit{tropical adversarial attacks}. The latter aims to generate input perturbation that is equivalent to perturbing the decision boundaries of the network by perturbing its parameters. We conduct experiments on both synthetic data and MNIST dataset \cite{mnist} to validate the proposed method.
Chapter 2

Overview on Tropical Geometry

We state the following brief introduction to tropical geometry for completeness and clarity without claiming novelty. For a detailed review, we refer interested readers to the work of [30, 31].

Definition 1. (Tropical Semiring) The tropical semiring $\mathbb{T}$ is the triplet $\{\mathbb{R}\cup\{-\infty\}, \oplus, \odot\}$, where $\oplus$ and $\odot$ define tropical addition and tropical multiplication, respectively. They are denoted as:

$$x \oplus y = \max\{x, y\}, \quad x \odot y = x + y, \quad \forall x, y \in \mathbb{T}.$$

Moreover, the tropical quotient of $x$ over $y$ can be defined as: $x \odot y = x - y$ where $x - y$ is the standard subtraction.

Given the previous definition, it can be shown that $-\infty$ is the additive identity and $0$ is the multiplicative identity. In addition, the tropical power can be formulated as $x^{\odot a} = x \odot x \cdots \odot x = a.x$, for $x \in \mathbb{T}$, $a \in \mathbb{N}$, where $a.x$ is standard multiplication. For ease of notation, we write $x^{\odot a}$ as $x^a$. Next, we move on to defining tropical polynomials and their solution set.

Definition 2. (Tropical Polynomials) For $x \in \mathbb{T}^d$, $c_i \in \mathbb{R}$ and $a_i \in \mathbb{N}^d$, a $d$-variable tropical polynomial with $n$ monomials $f : \mathbb{T}^d \to \mathbb{T}^d$ can be expressed as:

$$f(x) = (c_1 \odot x^{a_1}) \oplus (c_2 \odot x^{a_2}) \oplus \cdots \oplus (c_n \odot x^{a_n}), \quad \forall \ a_i \neq a_j \text{ when } i \neq j.$$

\footnote{A semiring is a ring that lacks an additive inverse.}
Figure 2.1: **Duality between tropical hypersurface and dual subdivision.** For each function $f$, we show both its tropical hypersurface $\mathcal{T}(f)$, and the dual subdivision for its newton polygon $\delta(f)$. Note that the tropical hypersurface is perpendicular to the edges of the dual subdivision.

We use the more compact vector notation $x^a = x_1^{a_1} \circ x_2^{a_2} \cdots \circ x_d^{a_d}$. Moreover and for ease of notation, we will denote $c_i \circ x^{a_i}$ as $c_i x^{a_i}$ throughout the paper.

Note that a tropical polynomial is a finite tropical sum of tropical monomials ($c_i \circ x^{a_i}$). Moreover, a $d-$variate tropical polynomial forms a convex function since it is a maximum across linear hyperplanes [32].

**Definition 3.** *(Tropical Rational Functions)* A tropical rational is a standard difference or a tropical quotient of two tropical polynomials: $f(x) \odot g(x) = f(x) - g(x)$.

Tropical geometry extends many notations in classical algebraic geometry, among which hypersurfaces. A tropical hypersurface of a tropical polynomial $f$, which is the solution set of $f$, is the set of points where $f$ is non-linear in $x$.

**Definition 4.** *(Tropical Hypersurfaces)* A tropical hypersurface of a tropical polynomial $f(x) = c_1 x^{a_1} + \cdots + c_n x^{a_n}$ is the set of points $x$ where $f$ is attained by two or more monomials in $f$, i.e.

$$\mathcal{T}(f) := \{ x \in \mathbb{R}^d : c_i x^{a_i} = c_j x^{a_j} = f(x), \ \forall \ a_i \neq a_j \}.$$  

Tropical hypersurfaces divide the domain of $f$ into convex regions, where $f$ is linear in
each region (e.g. Figure 2.1). Next, we define the geometric objects that are associated with tropical polynomial, namely Newton polytope and dual subdivision.

**Definition 5. (Newton Polytopes)** The Newton polytope of a tropical polynomial $f(x) = c_1x^{a_1} \oplus \cdots \oplus c_nx^{a_n}$ is the convex hull of the exponents $a_i \in \mathbb{N}^d$ regarded as points in $\mathbb{R}^d$, i.e.

$$\Delta(f) := \text{ConvHull}\{a_i \in \mathbb{R}^d : i = 1, \ldots, n \text{ and } c_i \neq -\infty\}.$$ 

To determine the dual subdivision of a tropical polynomial $f$, first we construct the polytope $\mathcal{P}(f) := \text{ConvHull}\{(a_i, c_i) \in \mathbb{R}^d \times \mathbb{R} : i = 1, \ldots, n\}$ which appends the coefficients of the tropical polynomial as an extra dimension. Then, we get the dual subdivision determined by $f; \delta(f)$, by projecting the collection of the upper faces of $\mathcal{P}(f)$ with the projection $\pi$ which drops the last coordinate $\delta(f) := \{\pi(p) \subset \mathbb{R}^d : p \in \text{UF}(\mathcal{P}(f))\}$, where $\pi : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$. It is worthwhile to mention that this projection will be essential in extending the results of the following chapter into more general case. Tropical hypersurfaces are dual to $\delta(f)$ as shown by [31]. This duality implies that each vertex in $\delta(f)$ corresponds to a region where $f$ is linear or equivalently one of the convex cells that the tropical hypersurface of $f$ divides the space into. This is clearly shown in Figure 2.1 with three different tropical polynomials, where the number of regions where $f$ is linear and the number of vertices of $\delta(f)$ for the three examples are 3, 6 and 10, respectively. More interestingly, this duality entails that the tropical hypersurfaces $T(f)$ are parallel to the normals of the edges of the dual subdivision $\delta(f)$. The former duality is essential in the rest of the thesis. For example, in Chapter 3, we show that one can prune the parameters of a neural network by removing those which do not contribute to the shape of the dual subdivision of the decision boundaries. Finally, the analysis of neural networks will require knowing the effect of tropical power, topical product, and tropical sum on the polytope $\mathcal{P}(f)$.
and it is accomplished through the following proposition [24].

**Proposition 1.** [24] Let both $f$ and $g$ be tropical polynomials and let $a \in \mathbb{N}$, then

\[
\mathcal{P}(f^a) = a\mathcal{P}(f),
\]

\[
\mathcal{P}(f \odot g) = \mathcal{P}(f) \tilde{+} \mathcal{P}(g),
\]

\[
\mathcal{P}(f \oplus g) = \text{ConvexHull}\left(\mathcal{V}(\mathcal{P}(g)) \cup \mathcal{V}(\mathcal{P}(g))\right),
\]

where $\mathcal{V}(\mathcal{P}(f))$ is the set of vertices of the polytope $\mathcal{P}(f)$, and $\tilde{+}$ is the minkowski sum that is defined between the sets $P$ and $Q$ as $P \tilde{+} Q = \{p + q, \forall p \in P \text{ and } q \in Q\}$.

Further details and standard results are summarized by [33].
Chapter 3

Functional and Geometrical Representations of Deep Neural Networks in Tropical Geometry

In this chapter, we investigate both the functional representation of DNNs in tropical geometry, and the tropical geometrical characterization of the decision boundaries of DNNs. In the functional part, we show that a certain family of DNNs can be represented as a tropical rational map, and connect this result to an earlier result of classical algebra. In the geometrical part, we develop a geometrical characterization to the decision boundaries of a neural network in the form (Linear - ReLU - Linear).

3.1 Deep Neural Networks in Tropical Geometry

While the architectures of DNNs can vary significantly depending on the application, they can be viewed as a repetitive composition of a linear layer followed by a non-linear activation function. In this work, we are interested in studying DNNs in the form

\[ f(x) = \sigma^{(L)} \circ A^{(L)} \circ \sigma^{(L-1)} \circ A^{(L-1)} \cdots \circ \sigma^{(1)} \circ A^{(1)} x, \]

where \( \sigma \) is the non-linear activation function, \( L \) is the depth of the network and \( \circ \) is the composition operation. In this context, Zhang et al. \[24\] showed an equivalency between tropical rational maps and any neural network \( f : \mathbb{R}^n \rightarrow \mathbb{R}^k \) with piecewise linear activation function with real biases and integer weights through the following theorem.

**Theorem 1.** (Tropical Characterization of Neural Networks, \[24\]). A feedforward
neural network with integer weights and real biases with piecewise linear activation functions is a function \( f : \mathbb{R}^n \to \mathbb{R}^k \), whose coordinates are tropical rational functions of the input, i.e.,

\[
f(x) = H(x) \odot Q(x) = H(x) - Q(x),
\]

where \( H \) and \( Q \) are tropical polynomials.

Note that constraining the weights to be integers is due to the fact that the exponents in tropical powers must be positive integers. This issue can be addressed since real numbers can be represented as fractions up to a certain precision, and multiplying by the common denominator will result in integer weight.

While this result is new in the context of tropical geometry, it is not surprising. A prior result showed that any piecewise linear function can be represented as a difference of two convex piecewise linear functions \([34]\). The former can be formulated by the following proposition.

**Proposition 2.** Every piecewise linear continuous function \( f(x) \) can be expressed as

\[
f(x) = \max_{i \in [m]} \{ a_i^\top x \} - \max_{j \in [n]} \{ b_j^\top x \}, \quad \text{where } [m] = \{1, \ldots, m\} \text{ and } [n] = \{1, \ldots, n\}.
\]

Note that for integer elements of both \( a_i^\top \) and \( b_j^\top \), each of the two maxima above turns into a tropical polynomial and \( f(x) \) becomes a tropical rational function recovering Theorem \([1]\). This equivalency opens the door towards understanding deep neural networks by analyzing their corresponding tropical polynomials which can be accomplished by studying their solution set (tropical hypersurfaces), or their geometrical characterizations.
3.2 Decision Boundaries of DNNs as Polytopes

After studying the functional representation of DNNs in tropical geometry, we deploy it to exploit the decision boundaries of a family of DNNs. In particular, we show an equivalency between a super set to the decision boundaries of a family of DNNs and a tropical hypersurface of a tropical polynomial. To do so, we analyze the decision boundaries of a network in the form (Linear - ReLU - Linear) with integer weights in the linear layers using tropical geometry. Note that the proposed analysis covers any piecewise linear activation function while ReLU was chosen due to its popularity.

The functional form of this network is: \( f(x) = B \max (Ax + c_1, 0) + c_2 \), where \( \max(.) \) is an element-wise operator. Throughout this chapter, we assume that \( A \in \mathbb{Z}^{p \times n} \), \( B \in \mathbb{Z}^{2 \times p} \), \( c_1 \in \mathbb{R}^p \) and \( c_2 \in \mathbb{R}^2 \). For ease of notation, we only consider networks with two outputs, i.e. \( B^{2 \times p} \), where the extension to a multi-class output follows naturally and is discussed in the appendix. By Theorem 1 and since the output of the network is piecewise linear function, then each output can be expressed as tropical rational function. If \( f_1 \) and \( f_2 \) refer to the first and second outputs respectively, we have \( f_1(x) = H_1(x) \odot Q_1(x) \) and \( f_2(x) = H_2(x) \odot Q_2(x) \), where \( H_1, H_2, Q_1 \) and \( Q_2 \) are tropical polynomials. For simplicity, we present the main result with a bias-free network, i.e. \( c_1 = 0 \) and \( c_2 = 0 \), where the general case where the network has non-zero biases is discussed later in this chapter.

**Theorem 2.** For a bias-free neural network in the form \( f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^2 \), where \( A \in \mathbb{Z}^{p \times n} \) and \( B \in \mathbb{Z}^{2 \times p} \), let \( R(x) = H_1(x) \odot Q_2(x) \oplus H_2(x) \odot Q_1(x) \) be a tropical polynomial. Then:

- Let \( \mathcal{B} = \{x \in \mathbb{R}^n : f_1(x) = f_2(x)\} \) define the decision boundaries of \( f \), then \( \mathcal{B} \subseteq \mathcal{T}(R(x)) \).

- \( \delta(R(x)) = \text{ConvHull}(\mathcal{Z}_{G_1}, \mathcal{Z}_{G_2}) \). \( \mathcal{Z}_{G_1} \) is a zonotope in \( \mathbb{R}^n \) with line segments \( \{(B^+(1,j) + B^-(2,j))(A^+(j,:)\), \( A^-(j,:)\}\}_{j=1}^p \) and shift \( (B^-(-1,:) + B^+(2,:))A^- \). \( \mathcal{Z}_{G_2} \)
Figure 3.1: **Decision Boundaries as Geometric Structures.** The decision boundaries $\mathcal{B}$ (in red) comprise two linear pieces separating classes $C_1$ and $C_2$. As per Theorem 2, the dual subdivision of this single hidden neural network is the convex hull between the zonotopes $Z_{G_1}$ and $Z_{G_2}$. The normals to the dual subdivision $\delta(R(\mathbf{x}))$ are in one-to-one correspondence to the tropical hypersurface $\mathcal{T}(R(\mathbf{x}))$, which is a superset to the decision boundaries $\mathcal{B}$. Note that some of the normals to $\delta(R(\mathbf{x}))$ (in red) are parallel to the decision boundaries.

is a zonotope in $\mathbb{R}^n$ with line segments $\{(B^{-}(1, j) + B^{+}(2, j))[A^{+}(j, :), A^{-}(j, :)\}]_{j=1}^p$ and shift $(B^{+}(1, :) + B^{-}(2, :))A^-$. Note that $A^+ = \max(A, 0)$ and $A^- = \max(-A, 0)$. The line segment $(B^{+}(1, j) + B^{-}(2, j))[A^{+}(j, :)A^{-}(j, :)\]$ has end points $A^+(j, :)$ and $A^-(j, :)$ in $\mathbb{R}^n$ and scaled by $(B^{+}(1, j) + B^{-}(2, j))$.

**Discussion.** This theorem characterizes the decision boundaries of a bias-free neural network of the form (Linear - ReLU - Linear) through the lens of tropical geometry. The first result states that the set of points forming the decision boundaries $\mathcal{B}$ is a subset of the tropical hypersurface of the tropical polynomial $R(\mathbf{x})$, i.e. $\mathcal{T}(R(\mathbf{x}))$. Similar to what have been discussed earlier and shown in Figure 2.1 tropical hypersurfaces are associated with a dual subdivision polytope $\delta(R(\mathbf{x}))$. The second result of Theorem 2 expresses the dual subdivision as a convex hull of two zonotopes $Z_{G_1}$ and $Z_{G_2}$, where each of these zonotopes is only a function of the network parameters $A$ and $B$.

Theorem 2 bridges the gap between the behaviour of the decision boundaries $\mathcal{B}$, through the superset $\mathcal{T}(R(\mathbf{x}))$, and the polytope $\delta(R(\mathbf{x}))$, which is the convex hull of two zonotopes. The first result of the theorem suggests that one can understand the decision boundaries $\mathcal{B}$ through understanding the tropical hypersurface $\mathcal{T}(R(\mathbf{x}))$. However, studying the superset $\mathcal{T}(R(\mathbf{x}))$ can be equally difficult to study the decision boundaries $\mathcal{B}$ and thus the second result of Theorem 2 shines. In the bias free
case, the projection $\pi$ becomes an identity mapping, and thus $\delta(R(x)) = \Delta(R(x))$ becomes a well structured geometric object. Furthermore, based on the results of [31] (Proposition 3.1.6) and as discussed in Figure 2.1, the normals to the edges of the polytope $\delta(R(x))$ (convex hull of two zonotopes) are in one-to-one correspondence with the tropical hypersurface $T(R(x))$. This suggests that one can exploit the decision boundaries $\mathcal{B}$ through studying the orientation of $\delta(R(x))$. It is worthy to mention that a special case of the first part of Theorem 2 was discussed in [24] where they have a single output with a scoring function. Since the dual subdivision becomes a convex hull of two zonotopes, we recap the definition of zonotopes.

**Definition 6.** Let $u^1, \ldots, u^p \in \mathbb{R}^n$. The zonotope formed by $u^1, \ldots, u^p$ is defined as $\mathcal{Z}(u^1, \ldots, u^p) := \{ \sum_{i=1}^{p} x_i u^i : 0 \leq x_i \leq 1 \}$. Equivalently, $\mathcal{Z}$ can be expressed with respect to the generator matrix $U \in \mathbb{R}^{p \times n}$, where $U(i,:) = u^i$ as $\mathcal{Z}_U := \{ U^\top x : \forall x \in [0,1]^p \}$.

Another common definition for a zonotope is the Minkowski sum of a set of line segments that start from the origin in $\mathbb{R}^n$ (refer to appendix). It is well known that the number of vertices of a zonotope is polynomial in the number of line segments i.e. $|\text{vert} (\mathcal{Z}_U)| \leq 2 \sum_{i=0}^{n-1} (p-1)_i = O(p^{n-1})$ [35].

Although Theorem 2 ties the decision boundaries to a polytope that is a convex hull between two zonotopes, it is still ambiguous how to can we construct this polytope efficiently. While the number of vertices of a zonotope is polynomial in the number of generators, fast algorithms for enumerating these vertices requires having line segments that have the origin as an end point [36]. Since the zonotopes in Theorem 2 are generated by line segments that have arbitrary end points, we need to transform them into a set of generators that have the origin as an end point. to this end, we propose the following proposition to address the aforementioned issue, where its proof is left for the appendix.
Proposition 3. The zonotope formed by \( p \) line segments in \( \mathbb{R}^n \) with two arbitrary end points as follows \( \{[u_{i1}, u_{i2}]\}_{i=1}^p \) is equivalent to the zonotope formed by the line segments \( \{[u_i - u_2, 0]\}_{i=1}^p \) with a shift of \( \sum_{i=1}^p u_i^2 \).

As per Proposition 3, the generator matrices of zonotopes \( Z_{G_1}, Z_{G_2} \) in Theorem 2 can be defined as \( G_1 = \text{Diag}((B^+(1,:)) + (B^-(2,:)))A \) and \( G_2 = \text{Diag}((B^+(2,:))+(B^-(1,:)))A \), both with shift \( (B^-(1,:) + B^+(2,:) + B^+(1,:) + B^-(2,:))A^- \), where \( \text{Diag}(v) \) rearranges the elements of \( v \) in a diagonal matrix. By that, we can use efficient algorithms that enumerates zonotope vertices in Theorem 2 in polynomial time [36], and thus analyze the decision boundaries polytope \( \delta(R(x)) \) efficiently.

3.3 Disclaimer

This thesis is built on a submitted paper. Here, we present the full work but we emphasize on Adel’s strong involvement in deriving both Theorem 2 and Proposition 3. In what follows, we present the key contribution results of this thesis.

3.4 Handling Biases

Now, we show that the results of Theorem 2 are the unaltered in the presence of biases \((c_1, c_2)\). To do so, we will derive the dual subdivision of the output of the first Affine layer, the ReLU and the last Affine layer consecutively.

As for the output of the first affine layer, note that for an input \( x \) the output of the first affine layer \( z_1 = Ax + c_1 \) can be presented tropically per coordinate as:

\[
z_{1i} = A^+(i,:)x + c_1(i) - A^-(i,:)x = (c_1(i) \odot x^{A^+(i,:)}) \odot x^{A^-(i,:)} = H_{1i} \odot Q_{1i}. \tag{3.1}
\]

To construct the dual subdivision for each tropical polynomial \( H_{1i} \) and \( Q_{1i} \), one needs first to construct the tropical newton polytope in \( \mathbb{R}^{n+1} \) as defined in Definition...
Since both of \( H_{1i} \) and \( Q_{1i} \) are tropical polynomials with a single monomials thus \( \Delta(H_{1i}) \) and \( \Delta(Q_{1i}) \) are the points \((A^+(i,:), c_1(i))\) and \((A^-(i,:), 0)\) in \( \mathbb{R}^{n+1} \), respectively. To construct the dual subdivision of each tropical polynomial, one needs to project the newton polytope to \( \mathbb{R}^n \) through the operator \( \pi \) which will again result in an identical dual subdivision if biases were not introduced.

As for the output of the ReLU layer, note that for an input \( x \), it can be presented tropically per coordinate as follows

\[
\begin{align*}
z_{2i} &= \max(z_{1i}, 0) = \max(A^+(i,:)x + c_1(i), A^-(i,:)x) - A^-(i,:)x \\
&= (H_{1i} \oplus Q_{1i}) - Q_{1i} = H_{2i} \ominus Q_{2i}.
\end{align*}
\]

Following Equation equation 2.3, the newton polytope of \( H_{2i} \) is a line segment, \( i.e. \Delta(H_{2i}) \), with end points \([A^+(i,:), c_1(i)], (A^-(i,:), 0)\]. Constructing the dual subdivision by projecting both \( \Delta(H_{2i}) \) and \( \Delta(Q_{1i}) \) to \( \mathbb{R}^n \) recovers an identical dual subdivision for a bias free-network. Similarly for \( \Delta(Q_{2i}) \) which is a point in \( \mathbb{R}^{n+1} \) with coordinate \([A^-(i,:), 0]\). Applying the projection \( \pi \) recovers to construct the dual subdivision recovers the point \([A^-(i,:)]\) in \( \mathbb{R}^n \).

Lastly, the output of the second affine layer per coordinate can be expressed as:

\[
\begin{align*}
z_{3i} &= B(i,:)z_2 + c_2(i) = (B^+(i,:) - B^-(i,:))(H_{2i} - Q_{2i}) + c_2(i) \\
&= (B^+(i,:)H_{2i} + B^-(i,:)Q_{2i} + c_2(i)) - (B^-(i,:)H_{2i} + B^+(i,:)Q_{2i}) \\
&= (B^+(i,:)H_{2i} + B^-(i,:)Q_{2i} + c_2(i)) - (B^-(i,:)H_{2i} + B^+(i,:)Q_{2i}) \\
&= H_{3i} \ominus Q_{3i}.
\end{align*}
\]

Following Equations equation 2.1 and equation 2.2 we have that \( \Delta(H_{3i}) = \bigoplus_j \left( B(i,j) \Delta(H_{2j}) \right) + \Delta(\bigoplus_j B^-(i,j) \Delta(Q_{2j}) \bigg|_{\mathbb{R}^n}, c_2(i) \bigg) \) where \( \Delta(Q_{2j}) \bigg|_{\mathbb{R}^n} \) is the newton
polytope of $Q_j$ just as before but in $\mathbb{R}^n$. Note that the first term, i.e. $\tilde{\gamma}_j\left(B(i,j)\Delta(H_{2,j})\right)$ is a Minkowski sum of scaled line segments $\Delta(H_{2,j})$ that are in $\mathbb{R}^{n+1}$ resulting in a zonotope in $\mathbb{R}^{n+1}$. The second term is the polytope that results from the Minkowski sum, as per Equation 2.1 of the scaled polytopes $\Delta(Q_j)|_{\mathbb{R}^n}$ where each polytope is a point in $\mathbb{R}^n$ with the last coordinate be $c_2(i)$. Therefore, the polytope $\Delta(H_{3,i})$ is a Minkowski sum between a zonotope and a point resulting in a shifted zonotope in $\mathbb{R}^{n+1}$. Constructing the dual subdivision $\delta(H_{3,i})$ results in the same polytope projected in $\mathbb{R}^n$. This is an identical dual subdivision to a bias free network. Therefore, the shape of the geometric representation of the decision boundaries with non-zero biases will not be altered the projection $\pi$, and hence the presence of the biases will not alter the results of Theorem 2.

### 3.5 Potential Applications

A valid question that can be made is that how useful are the stated theoretical results can be in practice. To answer this question we shed light on three potential applications that will be investigated afterwards. First, by the means of both Theorem 2 and Proposition 3 one can extract useful information about the decision boundaries cheaply instead of making huge number of forward passes that is exponential in the dimension of the input space. Second, we revisit network pruning, where design a geometrically based objective function to output a network with sparse weight matrices $\tilde{A}$ and $\tilde{B}$, but preserves the decision boundaries polytope $\delta(R(x))$. Formally speaking, we want to design the following objective to compute $\tilde{A}$ and $\tilde{B}$:

$$\min_{A, B} d\left(\delta(\tilde{R}(x)), \delta(R(x))\right) = \min_{A, B} d\left(\text{ConvHull} (Z_{\tilde{G}_1}, Z_{\tilde{G}_2}), \text{ConvHull} (Z_{G_1}, Z_{G_2})\right),$$

where $d(.)$ is a distance measure between two geometric objects. The output of
Figure 3.2: **Applications of Tropical Geometry in DNNs.** Transforming the decision boundaries into its geometrical representation yields to three different applications in lottery ticket hypothesis, network pruning and adversarial attacks.

This optimization problem is a sparse network that has a similar decision boundaries polytope as the original network. The third potential application is to construct a new view to the adversarial attacks. It is known that designing adversarial attacks aims to constructing input perturbation that changes the network prediction of a given input. Here, we construct perturbations to the decision boundaries through perturbing the parameters of the network, and erect an equivalent perturbation to the input such that the non-perturbed network is fooled.
Chapter 4

Topical Perspective to the Lottery Ticket Hypothesis

The lottery ticket hypothesis was recently proposed by [25], in which the authors surmise the existence of sparse trainable sub-networks of dense, randomly-initialized, feed-forward networks that when trained in isolation perform as well as the original network in a similar number of iterations.

To find such sub-networks, [25] propose the following simple algorithm: perform standard network pruning, initialize the pruned network with the same initialization that was used in the original training setting, and train with the same number of epochs. They hypothesize that this should result in a smaller network with a similar accuracy to the larger dense network. In other words, a subnetwork can have similar decision boundaries to the original network. While in this section we do not provide a theoretical reason for why this proposed pruning algorithm performs favorably, we utilize the geometric structure that arises from Theorem 2 to reaffirm such behaviour.

In particular, we show that the orientation of the dual subdivision \( \delta(R(x)) \) (referred to as decision boundaries polytope form now onwards), where the normals to its edges are parallel to \( T(R(x)) \) that is a superset to the decision boundaries, is preserved after pruning with the proposed initialization algorithm of [25]. On the other hand, pruning routines with a different initialization at each pruning iteration will result in a severe variation in the orientation of the decision boundaries polytope. This leads to a large change in the orientation of the decision boundaries, which tends to hinder accuracy.

To this end, we train a neural network with 2 inputs \((n = 2)\), 2 outputs, and
a single hidden layer with 40 nodes \( (p = 40) \). We then prune the network by removing the smallest \( x\% \) of the weights. The pruned network is then trained using different initializations: (i) the same initialization as the original network [25], (ii) Xavier [37], (iii) standard Gaussian and (iv) zero mean Gaussian with variance of 0.1. Figure 4.1 shows the decision boundaries polytope, \( \delta(R(x)) \), as we perform more pruning (increasing the \( x\% \)) with different initializations. First, we show the decision boundaries by sampling and classifying points in a grid with the trained network (first subfigure). We then plot the decision boundaries polytope \( \delta(R(x)) \) as per the second part of Theorem 2 denoted as original polytope (second subfigure). While there are many overlapping vertices in the original polytope, the normals to some of the edges (the major visible edges) are parallel in direction to the decision boundaries shown in the first subfigure of Figure 4.1. We later show the decision boundaries polytope for the same network with varying levels of pruning. It is to be observed that the orientation of the polytopes \( \delta(R(x)) \) deviate from the decision boundaries polytope of the original network without any pruning much more for all different initialization schemes as compared to the lottery ticket initialization. This gives an indication that lottery ticket initialization indeed preserves the decision boundaries, since it preserves the orientation of the decision boundaries polytope throughout the evolution of pruning. Several other examples are left for the appendix. Another approach to investigate the lottery ticket could be by observing the polytopes repre-
resenting the functional form of the network directly, i.e. \( \delta(H_{1,2}(x)) \) and \( \delta(Q_{1,2}(x)) \), in lieu of the decision boundaries polytopes. However, this does not provide conclusive answers to the lottery ticket, since there can exist multiple functional forms, and correspondingly multiple polytopes \( \delta(H_{1,2}(x)) \) and \( \delta(Q_{1,2}(x)) \), for networks with the same decision boundaries. This is why we explicitly focus our analysis on \( \delta(R(x)) \), which is directly related to the decision boundaries of the network. Further discussions and experiments are left for the appendix. Note that extracting \( \delta(R(x)) \) cheaply is vital since it gives important information about the orientation of the decision boundaries without doing any forward passes. Next, we move on to deploy this piece of information to design a geometrical based optimization problem to prune network parameters.
Chapter 5

Tropical Network Prunning

Network pruning has been identified as an effective approach for reducing the computational cost and memory usage during network inference. While it dates back to the work of [38] and [39], network pruning has recently gained more attention. This is due to the fact that most neural networks over-parameterize commonly used datasets. In network pruning, the task is to find a smaller subset of the network parameters, such that the resulting smaller network has similar decision boundaries (and thus supposedly similar accuracy) to the original over-parameterized network. In this section, we show a new geometric approach towards network pruning. In particular, as indicated by Theorem 2, preserving the polytope $\delta(R(x))$ preserves a superset to the decision boundaries $T(R(x))$, and thus supposedly the decision boundaries themselves.

Motivational Insight. For a single hidden layer neural network, the dual subdivision to the decision boundaries is the polytope that is the convex hull of two zonotopes, where each is formed by taking the Minkowski sum of line segments (Theorem 2). Figure 5.1 shows an example, where pruning a neuron in the neural network has no effect on the dual subdivision polytope and equivalently no effect on the accuracy. This is since the orientation of the decision boundaries polytope did not change, thus, preserving the tropical hypersurface $T(R(x))$ and keeping the decision boundaries of both networks the same.

Problem Formulation. In light of the motivational insight, a natural question arises: Given an over-parameterized binary output neural network $f(x) = B \max (Ax, 0)$, can one construct a new neural network, parameterized by some sparser weight ma-
Figure 5.1: Tropical Pruning Pipeline. Pruning the 4th node, or equivalently removing the two yellow vertices of zonotope $Z_{G_2}$ does not affect the decision boundaries polytope, which will lead to no change in accuracy.

To address this question, we propose the following general optimization problem to compute $\tilde{A}$ and $\tilde{B}$:

$$
\min_{\tilde{A}, \tilde{B}} \| \delta(\tilde{R}(x)) - \delta(R(x)) \|_{\mathcal{D}(\tilde{A}, \tilde{B}), \mathcal{D}(A, B)} = \min_{\tilde{A}, \tilde{B}} \| \text{ConvHull}(Z_{\tilde{G}_1}, Z_{\tilde{G}_2}) - \text{ConvHull}(Z_{G_1}, Z_{G_2}) \|_{\mathcal{D}(\tilde{A}, \tilde{B}), \mathcal{D}(A, B)}.
$$

(5.1)

The function $d(.)$ defines a distance between two geometric objects. Since the generators $\tilde{G}_1$ and $\tilde{G}_2$ are functions of $\tilde{A}$ and $\tilde{B}$ (as per Theorem 2), this optimization problem can be challenging to solve. However, for pruning purposes, one can observe from Theorem 2 that if the generators $\tilde{G}_1$ and $\tilde{G}_2$ had fewer number of line segments (rows), this corresponds to a fewer number of rows in the weight matrix $\tilde{A}$ (sparser weights). So, we observe that if $\tilde{G}_1 \approx G_1$ and $\tilde{G}_2 \approx G_2$, then $\delta(\tilde{R}(x)) \approx \delta(R(x))$, and thus the decision boundaries tend to be preserved as a consequence. Therefore, we propose the following optimization problem as a surrogate to Problem (5.1):

$$
\min_{\tilde{A}, \tilde{B}} \frac{1}{2} \left( \| \tilde{G}_1 - G_1 \|_F^2 + \| \tilde{G}_2 - G_2 \|_F^2 \right) + \lambda_1 \| \tilde{G}_1 \|_{2,1} + \lambda_2 \| \tilde{G}_2 \|_{2,1}.
$$

(5.2)

The matrix mixed norm for $C \in \mathbb{R}^{n \times k}$ is defined as $\| C \|_{2,1} = \sum_{i=1}^{n} \| C(i,:) \|_2$, which encourages the matrix $C$ to be row sparse, i.e. complete rows of $C$ are zero. Note that $\tilde{G}_1 = \text{Diag}[\text{ReLU}(\tilde{B}(1,:)) + \text{ReLU}(-\tilde{B}(2,:))]\tilde{A}$ and $\tilde{G}_2 = \text{Diag}[\text{ReLU}(\tilde{B}(2,:))].$
We solve Problem (5.2) through alternating optimization over the variables $\tilde{A}$ and $\tilde{B}$, where each sub-problem can be solved in closed form.

**Extension to multi-class networks.** Note that Theorem 2 describes a superset to the decision boundaries of a binary classifier through the dual subdivision $R(x)$, i.e. $\delta(R(x))$. For a neural network $f$ with $k$ classes, a natural extension for it is to analyze the pair-wise decision boundaries of all $k$-classes. Thus, let $T(R_{ij}(x))$ be the superset to the decision boundaries separating classes $i$ and $j$. Therefore, a natural extension to the geometric loss in Equation (5.1) is to preserve the polytopes among all pairwise follows:

$$
\min_{\tilde{A},\tilde{B}} \sum_{\forall \{i,j\} \in S} d \left( \text{ConvexHull} \left( \mathcal{Z}_{\tilde{G}(i^+,j^-)}; \mathcal{Z}_{\tilde{G}(j^+,i^-)} \right), \text{ConvexHull} \left( \mathcal{Z}_{\tilde{G}(i^+,j^-)}; \mathcal{Z}_{\tilde{G}(j^+,i^-)} \right) \right).
$$

(5.3)

The set $S$ is all possible pairwise combinations of the $k$ classes such that $S = \{\{i,j\}, \forall i \neq j, i = 1, \ldots, k, j = 1, \ldots, k\}$. The generator $\mathcal{Z}(\tilde{G}_{ij})$ is the zonotope with the generator matrix $\tilde{G}_{ij} = \text{Diag} \left[ \text{ReLU}(-\tilde{B}(i,:)) + \text{ReLU}(\tilde{B}(j,:)) \right] \tilde{A}$. However, such an approach is generally computationally expensive, particularly, when $k$ is very large. To this end, we make the following observation that $\mathcal{Z}_{\tilde{G}(i^+,j^-)}$ can be equivalently written as a Minkowski sum between two zonotopes with the generators $G_{i^+} = \text{Diag} \left[ \text{ReLU}(\tilde{B}(i,:)) \right] \tilde{A}$ and $G_{j^-} = \text{Diag} \left[ \text{ReLU}(\tilde{B}(j,:)) \right] \tilde{A}$. That is to say, $\mathcal{Z}_{\tilde{G}(i^+,j^-)} = \mathcal{Z}_{G_{i^+}} + \mathcal{Z}_{G_{j^-}}$ where this follows from the associative property of Minkowski sums. Generally speaking, if $\{S_i\}_{i=1}^n$ is a set of $n$ line segments. Then we have that

$$
S = S_1 \dot{+} \ldots \dot{+} S_n = P \dot{+} V
$$

where the sets $P = \dot{+} \{j \in C_1, S_j\}$ and $V = \dot{+} \{j \in C_2, S_j\}$ and $C_1$ and $C_2$ are any complementary partitions of the set $\{S_i\}_{i=1}^n$. Hence, $\tilde{G}(i^+,j^-)$ can be seen a concatenation between
\[ \tilde{G}(i^+ \text{ and } \tilde{G}(j^-). \text{ Thus, the objective in 5.3 can be expanded as follows:} \]

\[
\min_{A,B} \sum_{\forall [i,j] \in S} d\left( \text{ConvexHull} \left( Z\tilde{G}(i^+,j^-), Z\tilde{G}(j^+,i^-) \right), \text{ConvexHull} \left( ZG(i^+,j^-), ZG(j^+,i^-) \right) \right) \\
= \min_{A,B} \sum_{\forall [i,j] \in S} d\left( \text{ConvexHull} \left( Z\tilde{G}(i^+,j^-), Z\tilde{G}(j^+,i^-) \right), \text{ConvexHull} \left( Z\tilde{G}(i^+,j^-), Z\tilde{G}(j^+,i^-) \right) \right) \\
\approx \min_{A,B} \sum_{\forall [i,j] \in S} \left( \| \tilde{G} - G \|_F^2 + \| \tilde{G} - G \|_F^2 + \| \tilde{G} - G \|_F^2 + \| \tilde{G} - G \|_F^2 \right). \\
= \min_{A,B} \sum_{i=1}^{k} \frac{1}{2} (k - 1) \left( \| \tilde{G} - G \|_F^2 + \| \tilde{G} - G \|_F^2 \right).
\]

The approximation follows in a similar argument to the binary classifier case where approximating the generators. The last equality follows from a counting argument. We solve the objective for all multi-class networks in the experiments with alternating optimization in a similar fashion to the binary classifier case. Similarly to the binary classification approach, we introduce the \( \| . \|_{2,1} \) to enforce sparsity constraints for pruning purposes. Therefore the overall objective has the form:

\[
\min_{A,B} \sum_{i=1}^{k} \frac{1}{2} \left( \| \tilde{G} - G \|_F^2 + \| \tilde{G} - G \|_F^2 \right) + \lambda \left( \| \tilde{G} \|_{2,1} + \| \tilde{G} \|_{2,1} \right). \tag{5.4}
\]

Details of the optimization and the derivation of the update steps of both the binary and the multi-class case are in the appendix.

**Extension to Deeper Networks.** For deeper networks, one can still apply the aforementioned optimization for consecutive blocks. In particular, we prune each consecutive block of the form (Affine,ReLU,Affine) starting from the input and ending at the output of the network.

**Experiments on Tropical Pruning.** Here, we evaluate the performance of the pro-
Figure 5.2: **Results of Tropical Pruning.** Pruning-accuracy plots for AlexNet (top) and VGG16 (bottom) trained on SVHN, CIFAR10, and CIFAR100, pruned with our tropical method and three other pruning methods.

Our proposed pruning approach as compared to several classical approaches on several architectures and datasets. In particular, we compare our tropical pruning approach against Class Blind (CB), Class Uniform (CU) and Class Distribution (CD) \[40, 41\]. In Class Blind, all the parameters across all nodes of a layer are sorted by magnitude where \(x\%\) with smallest magnitude are pruned. In contrast, Class Uniform prunes the parameters with smallest \(x\%\) magnitudes per node in a layer. Lastly, Class Distribution performs pruning of all parameters for each node in the layer, just as in Class Uniform, but the parameters are pruned based on the standard deviation \(\sigma_c\) of the magnitude of the parameters per node. Since fully connected layers in deep neural networks tend to have much higher memory complexity than convolutional layers, we restrict our focus to pruning fully connected layers. We train AlexNet and VGG16 on SVHN, CIFAR10, and CIFAR100 datasets. We observe that we can prune more than 90% of the classifier parameters for both networks without affecting the accuracy. Moreover, we demonstrate experimentally that our approach can outperform all other methods even when all parameters or when only the biases are fine tuned after pruning (these experiments in addition to many others are left for the appendix).

**Setup.** We adapt the architectures of AlexNet and VGG16, since they were originally trained on ImageNet \[42\], to account for the discrepancy in the input resolution. The fully connected layers of AlexNet and VGG16 have sizes of (256,512,10) and (512,512,10),
respectively on SVHN and CIFAR100 with the last layer replaced to 100 for CIFAR100. All networks were trained to baseline test accuracy of (92%, 74%, 43%) for AlexNet on SVHN, CIFAR10 and CIFAR100, respectively and (92%, 92%, 70%) for VGG16. To evaluate the performance of pruning and following previous work [40], we report the area under the curve (AUC) of the pruning-accuracy plot. The higher the AUC is, the better the trade-off is between pruning rate and accuracy. For efficiency purposes, we run the optimization in Problem (5.2) for a single alternating iteration to identify the rows in \( \tilde{A} \) and elements of \( \tilde{B} \) that will be pruned, since an exact pruning solution might not be necessary. The algorithm and the parameter setup to solving Problem (5.2) is left for the appendix.

**Results.** Figure 4 shows the comparison between our tropical approach and the three popular pruning schemes on both AlexNet and VGG16 over the different datasets. Our proposed approach can indeed prune out as much as 90% of the parameters of the classifier without sacrificing much of the accuracy. For AlexNet, we achieve much better performance in pruning as compared to other methods. In particular, we are better in AUC by 3%, 3%, and 2% over other pruning methods on SVHN, CIFAR10 and CIFAR100, respectively. This indicates that the decision boundaries can indeed be preserved by preserving the dual subdivision polytope. For VGG16, we perform similarly well on both SVHN and CIFAR10 and slightly worse on CIFAR100. While the performance achieved here is comparable to the other pruning schemes, if not better, we emphasize that our contribution does not lie in outperforming state-of-the-art pruning methods, but in giving a new geometry-based perspective to network pruning. We conduct more experiments where only the biases of the network or only the classifier are fine tuned after pruning. Retraining biases can be sufficient as they do not contribute to the orientation of the decision boundaries polytope (and effectively the decision boundaries) but only a translation. Discussion on biases and more results are left for appendix.
Chapter 6

Tropical Adversarial Attacks

DNNs are notorious for being susceptible to adversarial attacks. In fact, adding small imperceptible noise, referred to as adversarial attacks, to the input of these networks can hinder their performance. Several works investigated the decision boundaries of neural networks in the presence of adversarial attacks. For instance, [43] analyzed the high dimensional geometry of adversarial examples by means of manifold reconstruction. Also, [44] crafted adversarial attacks by estimating the distance to the decision boundaries using random search directions. In this work, we provide a tropical geometric view to this task, where we show how Theorem 2 can be leveraged to construct a tropical geometry-based targeted adversarial attack.

Dual View to Adversarial Attacks. For a classifier $f : \mathbb{R}^n \to \mathbb{R}^k$ and input $x_0$ classified as $c$, a standard formulation for targeted adversarial attacks flips the prediction to a particular class $t$ and is usually defined as

$$
\min_{\eta} D(\eta) \quad \text{s.t.} \quad \arg \max_i f_i(x_0 + \eta) = t \neq c
$$

(6.1)

This objective aims to compute the lowest energy input noise $\eta$ (measured by $D$) such that the new sample $(x_0 + \eta)$ crosses the decision boundaries of $f$ to a new classification region. Here, we present a dual view to adversarial attacks. Instead of designing a sample noise $\eta$ such that $(x_0 + \eta)$ belongs to a new decision region, one can instead fix $x_0$ and perturb the network parameters to move the decision boundaries in a way that $x_0$ appears in a new classification region. In particular, let $A_1$ be the first linear layer of $f$, such that $f(x_0) = g(A_1 x_0)$. One can now perturb $A_1$ to alter the decision boundaries and relate this
Algorithm 1: Solving Problem (6.3)

Input: $A_1 \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{k \times p}, x_0 \in \mathbb{R}^n, t, \lambda > 0, \gamma > 1, K > 0, \xi_{A_1} = 0_{p \times n}, \eta^1 = z^1 = w^1 = u^1 = w^1 = 0_n$.

Output: $\eta, \xi_{A_1}$

Initialize: $\rho = \rho_0$

while not converged do

for $k \leq K$ do

$\eta$ update: $\eta^{k+1} = (2\lambda A_1^T A_1 + (2 + \rho)I)^{-1}(2\lambda A_1^T \xi_{A_1} x_0 + \rho z^k - u^k)$

$w$ update: $w^{k+1} = \begin{cases} 
\text{min}(1 - x_0, -\epsilon_1) : z^k - 1/\rho v^k > \text{min}(1 - x_0, \epsilon_1) \\
\text{max}( -x_0, -\epsilon_1) : z^k - 1/\rho v^k < \text{max}( -x_0, -\epsilon_1) \\
\text{otherwise}
\end{cases}$

$z$ update: $z^{k+1} = \frac{1}{\eta^{k+1} + 2\rho} (\eta^{k+1} z^k + \rho (\eta^{k+1} + 1/\rho u^k + w^k + 1/\rho v^k) - \nabla L(z^k + x_0))$

$\xi_{A_1}$ update: $\xi_{A_1}^{k+1} = \arg\min_{\xi_{A_1}} \|\xi_{A_1}\|_2 + \lambda \|\xi_{A_1} x_0 - A_1 \eta^{k+1}\|^2 + \tilde{L}(A_1) \text{ s.t. } \|\xi_{A_1}\|_{\infty, \infty} \leq \epsilon_2$

$u$ update: $u^{k+1} = u^k + \rho (\eta^{k+1} - z^{k+1})$

$v$ update: $v^{k+1} = v^k + \rho (w^{k+1} - z^{k+1})$

$\rho \leftarrow \gamma \rho$

$\lambda \leftarrow \gamma \lambda$

$\rho \leftarrow \rho_0$

\]

parameter perturbation to the input perturbation as follows:

$$g((A_1 + \xi_{A_1}) x_0) = g(A_1 x_0 + \xi_{A_1} x_0) = g(A_1 x_0 + A_1 \eta) = f(x_0 + \eta). \quad (6.2)$$

From this dual view, we observe that traditional adversarial attacks are intimately related to perturbing the parameters of the first linear layer through the linear system: $A_1 \eta = \xi_{A_1} x_0$. The two views and formulations are identical under such condition. With this analysis, Theorem 2 provides explicit means to geometrically construct adversarial attacks by perturbing the decision boundaries. In particular, since the normals to the dual subdivision polytope $\delta(R(x))$ of a given neural network represent the tropical hypersurface $\mathcal{T}(R(x))$, which is a superset to the decision boundaries set $\mathcal{B}$, $\xi_{A_1}$ can be designed to result in a minimal perturbation to the dual subdivision that is sufficient to change the network prediction of $x_0$ to the targeted class $t$. Based on this observation, we formulate the problem
Figure 6.1: **Dual View of Tropical Adversarial Attacks.** We show the effects of tropical adversarial attacks on a synthetic binary dataset at two different input points (in black). From left to right: the decision regions of the original and perturbed models, and decision boundaries polytopes (green for original and blue for perturbed).

as follows:

\[
\min_{\eta, \xi A_1} \ D_1(\eta) + D_2(\xi A_1)
\]

\[
\text{s.t.} \quad - \text{loss}(g(A_1(x_0 + \eta)), t) \leq -1; \quad - \text{loss}(g(A_1 + \xi A_1)x_0, t) \leq -1;
\]

\[
(x_0 + \eta) \in [0, 1]^n, \quad \|\eta\|_\infty \leq \epsilon_1; \quad \|\xi A_1\|_{\infty, \infty} \leq \epsilon_2, \quad A_1 \eta = \xi A_1 x_0.
\]

(6.3)

The loss is the standard cross-entropy loss. The first row of constraints ensures that the network prediction is the desired target class \(t\) when the input \(x_0\) is perturbed by \(\eta\), and equivalently by perturbing the first linear layer \(A_1\) by \(\xi A_1\). This is identical to \(f_1\) as proposed by [45]. Moreover, the third and fourth constraints guarantee that the perturbed input is feasible and that the perturbation is bounded, respectively. The fifth constraint is to limit the maximum perturbation on the first linear layer, while the last constraint enforces the dual equivalence between input perturbation and parameter perturbation. The function \(D_2\) captures the perturbation of the dual subdivision polytope upon perturbing the first linear layer by \(\xi A_1\). For a single hidden layer neural network parameterized as \((A_1 + \xi A_1) \in \mathbb{R}^{p \times n}\) and \(B \in \mathbb{R}^{2 \times p}\) for the first and second layers respectively, \(D_2\) can capture the perturbations in each of the two zonotopes discussed in Theorem 2 and we define it as:

\[
D_2(\xi A_1) = \frac{1}{2} \sum_{j=1}^2 \|\text{Diag}(B^+(j,:))\xi A_1\|_F^2 + \|\text{Diag}(B^-(j,:))\xi A_1\|_F^2.
\]

(6.4)

The derivation, discussion, and extension of (6.4) to multi-class neural networks is left for the **appendix**. We solve Problem (6.3) with a penalty method on the linear equality
Figure 6.2: **Effect of Tropical Adversarial Attacks on MNIST Dataset.** We show qualitative examples of adversarial attacks, produced by solving Problem (6.3), on two digits (8,9) from MNIST. From left to right, images are classified as [8,7,5,4] and [9,7,5,4] respectively.

**Motivational Insight to the Dual View.** Here, we train a single hidden layer neural network, where the size of the input is 2 with 50 hidden nodes and 2 outputs on a simple dataset as shown in Figure 6.1. We then solve Problem (6.3) for a given $x_0$ shown in black. We show the decision boundaries for the network with and without the perturbation at the first linear layer $\xi_{A_1}$. Figure 6.1 shows that perturbing an edge of the dual subdivision polytope, by perturbing the first linear layer, indeed corresponds to perturbing the decision boundaries and results in the misclassification of $x_0$. Interestingly and as expected, perturbing different decision boundaries corresponds to perturbing different edges of the dual subdivision.

**MNIST Experiments.** Here, we design perturbations to misclassify MNIST images. Figure 6.2 shows several adversarial examples that change the network prediction for digits 8 and 9 to digits 7, 5, and 4, respectively. In some cases, the perturbation $\eta$ is as small as $\epsilon = 0.1$, where $x_0 \in [0,1]^n$. Several other adversarial results are reported in Figure 6.2. We again emphasize that our approach is not meant to be compared with (or beat) state of the art adversarial attacks but rather to provide a novel geometrically inspired perspective that can shed new light in this field. Several other experiments are left for the **appendix.**
Chapter 7

Concluding Remarks

We leverage tropical geometry to characterize the decision boundaries of neural networks in the form (Affine - ReLU - Affine) and relate it to geometric objects such as zonotopes. This characterization enabled analyzing the decision boundaries of neural networks through manipulating the dual subdivision. We deployed this characterization into three different applications. First, we then provide a tropical perspective to support the lottery ticket hypothesis where we visualized the effect of different initializations to the decision boundaries polytope. Second, we deploy the tropical characterization of the decision boundaries in Network pruning, where the proposed optimization problem zeros out the network parameters that do not affect the decision boundaries polytope. We experimented the proposed method on several networks and datasets to demonstrate its potential. Finally, we designed tropical adversarial attacks which constructs input perturbation that is equivalent to perturbing the decision boundaries of network. We also experimented the proposed method on both synthetic and real datasets where it was able on both cases to fool the network. A natural extension is a compact derivation for the characterization of the decision boundaries of convolutional neural networks and graphical convolutional networks as well.
REFERENCES

convolutional neural networks,” in Advances in Neural Information Processing Systems
(NeurIPS), 2012.

“Large scale learning of general visual representations for transfer,” 2019.

segmentation,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition
(CVPR), 2015, pp. 3431–3440.

modeling in speech recognition: The shared views of four research groups,” IEEE


rectional transformers for language understanding,” arXiv preprint arXiv:1810.04805,
2018.

M. Etemadi, W. Ye, G. Corrado, D. P. Naidich, and S. Shetty, “End-to-end lung
Cancer screening with three-dimensional deep learning on low-dose chest computed

deep-learning analysis identifies contribution of noncoding mutations to autism risk,”
Nature Genetics, 2019.

[9] S. Shalev-Shwartz and S. Ben-David, Understanding machine learning: From theory

depth neural networks,” Advances in Neural Information Processing Systems (NeurIPS),
2014.


APPENDICES

A Experimental Details and Supplemental Results

In this section, we describe the settings and values of the hyper parameters used in the experiments. Moreover, we will show some further supplemental results to the results in the main manuscript paper.

A.0.1 Tropical View to the Lottery Ticket Hypothesis.

We first conduct some further supplemental experiments to those conducted in Chapter 4. In particular, we conduct further experiments re-affirming the lottery ticket hypothesis on three more synthetic datasets in a similar experimental setup to the one shown in Figure 4.1. The new supplemental experiments are shown in Figure A.1. A similar conclusion is present where the lottery ticket initialization consistently better preserves the decision boundaries polytope compared to other initialization schemes over different percentages of pruning.

A natural question is whether it is necessary to visualize the dual subdivision polytope of the decision boundaries, i.e. $\delta(R(x))$, where $R(x) = H_1(x) \odot Q_2(x) \oplus H_2(x) \odot Q_1(x)$ as opposed to visualizing the tropical polynomials $\delta(H_{\{1,2\}}(x))$ and $\delta(Q_{\{1,2\}}(x))$ directly for the tropical re-affirmation of the lottery ticket hypothesis. That is similar to asking whether it is necessary to visualize and study the decision boundaries polytope $\delta(R(x))$ as compared to the dual subdivision polytope of the functional form of the network since for the 2-output neural network described in Theorem 2, we have that $f_1(x) = H_1(x) \odot Q_1(x)$ and $f_2(x) = H_2(x) \odot Q_2(x)$. We demonstrate this with an experiment that demonstrates the differences between these two views. For this purpose, we train a single hidden layer neural network on
Figure A.1: **Effect of Different Initializations on the Decision Boundaries Polytope.**
From left to right: training dataset, decision boundaries polytope of original network followed by the decision boundaries polytope during several iterations of pruning with different initializations.

the same dataset shown in Figure 4.1. We perform several iterations of pruning in a similar fashion to Chapter 5 and visualise at each iteration both the decision boundaries polytope and all the dual subdivisions of the aforementioned tropical polynomials representing the functional form of the network, i.e. \( \delta(H_{\{1,2\}}(x)) \) and \( \delta(Q_{\{1,2\}}(x)) \). It is to be observed from Figure A.2 that despite that the decision boundaries were barely affected with the lottery ticket pruning, the zonotopes representing the functional form of the network endure large variations. That is to say, investigating the dual subdivisions describing the functional form of the networks through the four zonotopes \( \delta(H_{\{1,2\}}(x)) \) and \( \delta(Q_{\{1,2\}}(x)) \) is not indicative enough to the behaviour of the decision boundaries.
Figure A.2: Comparison between the decision boundaries polytope and the polytopes representing the functional representation of the network. First column: decision boundaries polytope $\delta(R(x))$ while the remainder of the columns are the zonotopes $\delta(H_1(x))$, $\delta(Q_1(x))$, $\delta(H_2(x))$ and $\delta(Q_2(x))$ respectively. Under varying pruning rate across the rows, it is to be observed that the changes that affected the dual subdivisions of the functional representations are far smaller compared to the decision boundaries polytope.

A.0.2 Tropical Pruning

Experimental Setup. In all experiments of the tropical pruning chapter, all algorithms are run for only a single iteration where $\lambda$ increases linearly from 0.02 with a factor of 0.01. Increasing $\lambda$ corresponds to increasing weight sparsity and we keep doing until sparsification is 100%.

Supplemental Experiments. We conduct more experimental results on AlexNet and VGG16 on SVHN, CIFAR10 and CIFAR100 datasets. We examine the performance for when the networks have only the biases of the classifier fine tuned after tuning as shown in Figure A.3. Moreover, a similar experiments is reported for the same networks but for
when the biases for the complete networks are fine tuned as in Figure A.4.

Figure A.3: **Results of Tropical Pruning with Fine Tuning the Biases of the Classifier.** Tropical pruning applied on AlexNet and VGG16 trained on SVHN, CIFAR10, CIFAR100 against different pruning methods with fine tuning the biases of the classifier only.

Figure A.4: **Results of Tropical Pruning with Fine Tuning the Biases of the Network.** Tropical pruning applied on AlexNet and VGG16 trained on SVHN, CIFAR10, CIFAR100 against different pruning methods with fine tuning the biases of the network.
A.0.3 Tropical Adversarial Attacks

Experimental Setup. For the tropical adversarial attacks experiments, there are five different hyper parameters which are

\( \epsilon_1 \): The upper bound for the infinite norm of \( \delta \).

\( \epsilon_2 \): The upper bound for the \( \| \cdot \|_{\infty, \infty} \) of the perturbation on the first linear layer.

\( \lambda \): Regularizer to enforce the equality between input and first layer perturbation

\( \eta \): Bergman divergence constant.

\( \rho \): ADMM constant.

For all of the experiments, we set the values of \( \epsilon_2, \lambda, \eta \) and \( \rho \) to 1, \( 10^{-3} \), 2.5 and 1, respectively. As for \( \epsilon_1 \) it is set to 0.1 upon attacking MNIST images of digit 4 set to 0.2 for all other MNIST images.
Figure A.5: **Effect of Tropical Adversarial Attacks on MNIST Images.** First row from the left: Clean image, perturbed images classified as [7,3,2,1,0] respectively. Second row from left: Clean image, perturbed images classified as [9,8,7,3,2] respectively. Third row from left: Clean image, perturbed images classified as [9,8,7,5,3] respectively. Fourth row from left: Clean image, perturbed images classified as [9,4,3,2,1] respectively. Fifth row from left: Clean image, perturbed images classified as [8,4,3,2,1] respectively.
B  Proofs and Derivations

B.1  Proof Of Theorem 2

Theorem 2. For a bias-free neural network in the form of \( f(x) : \mathbb{R}^n \to \mathbb{R}^2 \) where \( A \in \mathbb{Z}^{p \times n} \) and \( B \in \mathbb{Z}^{2 \times p} \), let \( R(x) = H_1(x) \circ Q_2(x) \oplus H_2(x) \circ Q_1(x) \) be a tropical polynomial. Then:

- Let \( B = \{ x \in \mathbb{R}^n : f_1(x) = f_2(x) \} \) define the decision boundaries of \( f \), then \( B \subseteq \mathcal{T}(R(x)) \).

- \( \delta(R(x)) = \text{ConvHull}(Z_{G_1}, Z_{G_2}) \). \( Z_{G_1} \) is a zonotope in \( \mathbb{R}^n \) with line segments \( \{(B^+(1,j) + B^-(2,j))[A^+(j,:), A^-(j,:)]\}_{j=1}^{p} \) and shift \( (B^-(1,:) + B^+(2,:))A^- \). \( Z_{G_2} \) is a zonotope in \( \mathbb{R}^n \) with line segments \( \{(B^-(1,j) + B^+(2,j))[A^+(j,:), A^-(j,:)]\}_{j=1}^{p} \) and shift \( (B^+(1,:) + B^-(2,:))A^- \). Note that \( A^+ = \max(A, 0) \) and \( A^- = \max(-A, 0) \).

The line segment \( (B^+(1,j) + B^-(2,j))[A^+(j,:), A^-(j,:)] \) has end points \( A^+(j,:) \) and \( A^-(j,:) \) in \( \mathbb{R}^n \) and scaled by \( (B^+(1,j) + B^-(2,j)) \).

Note that \( A^+ = \max(A, 0) \) and \( A^- = \max(-A, 0) \) where the \( \max(,) \) is element-wise. The line segment \( (B(1,j)^+ + B(2,j)^-)[A(j,:), A(j,:)] \) is one that has the end points \( A(j,:)^+ \) and \( A(j,:)^- \) in \( \mathbb{R}^n \) and scaled by the constant \( B(1,j)^+ + B(2,j)^- \).

Proof. For the first part, recall from Theorem 1 that both \( f_1 \) and \( f_2 \) are tropical rationals and hence,

\[ f_1(x) = H_1(x) - Q_1(x) \quad f_2(x) = H_2(x) - Q_2(x) \]
Thus

\[ B = \{ x \in \mathbb{R}^n : f_1(x) = f_2(x) \} = \{ x \in \mathbb{R}^n : H_1(x) - Q_1(x) = H_2(x) - Q_2(x) \} \]
\[ = \{ x \in \mathbb{R}^n : H_1(x) + Q_2(x) = H_2(x) + Q_1(x) \} \]
\[ = \{ x \in \mathbb{R}^n : H_1(x) \odot Q_2(x) = H_2(x) \odot Q_1(x) \} \]

Recall that the tropical hypersurface is defined as the set of \( x \) where the maximum is attained by two or more monomials. Therefore, the tropical hypersurface of \( R(x) \) is the set of \( x \) where the maximum is attained by two or more monomials in \( (H_1(x) \odot Q_2(x)) \), or attained by two or more monomials in \( (H_2(x) \odot Q_1(x)) \), or attained by monomials in both of them in the same time, which is the decision boundaries. Hence, we can rewrite that as

\[ T(R(x)) = T(H_1(x) \odot Q_2(x)) \cup T(H_2(x) \odot Q_1(x)) \cup B. \]

Therefore \( B \subseteq T(R(x)) \). For the second part of the Theorem, we first use the decomposition proposed by [24, 17] to show that for a network \( f(x) = B \max(Ax, 0) \), it can be decomposed as tropical rational as follows

\[ f(x) = (B^+ - B^-) \left( \max(A^+x, A^-x) - A^-x \right) \]
\[ = \left[ B^+ \max(A^+x, A^-x) + B^- A^-x \right] \]
\[ - \left[ B^- \max(A^+x, A^-x) + B^+ A^-x \right]. \]

Therefore, we have that

\[ H_1(x) + Q_2(x) = \left( B^+(1,:) + B^-(2,:) \right) \max(A^+x, A^-x) \]
\[ + \left( B^-(1,:) + B^+(2,:) \right) A^-x \]
\[ H_2(x) + Q_1(x) = \left( B^- (1,:) + B^+ (2,:) \right) \max(A^+ x, A^- x) + \left( B^+ (1,:) + B^- (2,:) \right) A^- x. \]

Therefore, note that:

\[
\delta(R(x)) = \delta \left( \left( H_1(x) \otimes Q_2(x) \right) \cup \left( H_2(x) \otimes Q_1(x) \right) \right)
\]

\[
\overset{\circ}{\operatorname{ConvexHull}} \left( \delta \left( H_1(x) \otimes Q_2(x) \right), \delta \left( H_2(x) \otimes Q_1(x) \right) \right)
\]

\[
\overset{\circ}{\operatorname{ConvexHull}} \left( \delta \left( H_1(x) \right), \delta \left( H_2(x) \right), \delta \left( Q_1(x) \right) \right).
\]

Now observe that \( H_1(x) = \sum_{j=1}^{P} \left( B^+(1,j) + B^-(2,j) \right) \max \left( A^+(j,:), A^-(j,:) \right) x \) tropically is given as follows \( H_1(x) = \odot_{j=1}^{P} \left[ x^{A^+(j,:)} \oplus x^{A^-(j,:)} \right]^{B^+(1,j) \odot B^-(2,j)} \), thus we have that:

\[
\delta(H_1(x)) = \left( B^+(1,1) + B^-(2,1) \right) \delta \left( x^{A^+(1,:)} \oplus x^{A^-(1,:)} \right) \overset{\circ}{+} \ldots \]

\[
\overset{\circ}{+} \left( B^+(1,p) + B^-(2,p) \right) \left( \delta(x^{A^+(p,:)} \oplus x^{A^-(p,:)} \right)
\]

\[
= \left( B^+(1,1) + B^-(2,1) \right) \operatorname{ConvexHull} \left( A^+(1,:), A^-(1,:) \right) \overset{\circ}{+} \ldots \]

\[
\overset{\circ}{+} \left( B^+(1,p) + B^-(2,p) \right) \operatorname{ConvexHull} \left( A^+(p,:), A^-(p,:) \right).
\]

The operator \( \overset{\circ}{+} \) indicates a Minkowski sum between sets. Note that \( \operatorname{ConvexHull} \left( A^+(i,:), A^-(i,:) \right) \) is the convexhull between two points which is a line segment in \( \mathbb{Z}^n \) with end points that are \( \{ A^+(i,:), A^+(i,:) \} \) scaled with \( B^+(1,i) + B^-(2,i) \). Observe that \( \delta(F_1(x)) \) is a Minkowski sum of line segments which is a zonotope. Moreover, note that \( Q_2(x) = (B^-(1,:) + B^+(2,:))A^-x \) tropically is given as follows \( Q_2(x) = \odot_{j=1}^{P} \left[ x^{A^-(j,:)} \right]^{(B^+(1,j) \odot B^-(2,j))} \). One can see that \( \delta(Q_2(x)) \) is the Minkowski sum of the points \( \{ (B^-(1,j) - B^+(2,j))A^-(j,:),) \} \forall j \) in \( \mathbb{R}^n \) (which is a standard sum) resulting in a point. Lastly, \( \delta(H_1(x)) \overset{\circ}{+} \delta(Q_2(x)) \) is a Minkowski sum between a zonotope and a single point which corresponds to a shifted zonotope. A similar symmetric argument can be applied for the second part \( \delta(H_2(x)) \overset{\circ}{+} \delta(Q_1(x)) \).
It is also worthy to mention that the extension to network with multi class output is trivial. In that case all of the analysis can be exactly applied studying the decision boundary between any two classes \((i, j)\) where \(\mathcal{B} = \{x \in \mathbb{R}^n : f_i(x) = f_j(x)\}\) and the rest of the proof will be exactly the same.

### B.2 Proof of Proposition 3

**Proposition 3.** The zonotope formed by \(p\) line segments in \(\mathbb{R}^n\) with two arbitrary end points as follows \(\{[u^1_i, u^2_i]\}_{i=1}^p\) is equivalent to the zonotope formed by the line segments \(\{[u^1_i - u^2_i, 0]\}_{i=1}^p\) with a shift of \(\sum_{i=1}^p u^2_i\).

**Proof.** Let \(U_j\) be a matrix with \(U_j(:,i) = u^i_j, i = 1, \ldots, p\), \(w\) be a column-vector with \(w(i) = w_i, i = 1, \ldots, p\) and \(1_p\) is a column-vector of ones of length \(p\). Then, the zonotope \(Z\) formed by the Minkowski sum of line segments with arbitrary end points can be defined as:

\[
Z = \left\{ \sum_{i=1}^p w_i u^1_i + (1 - w_i) u^2_i ; w_i \in [0, 1], \forall i \right\}
\]

\[
= \left\{ U_1 w - U_2 w + U_2 1_p ; w \in [0, 1]^p \right\}
\]

\[
= \left\{ (U_1 - U_2) w + U_2 1_p ; w \in [0, 1]^p \right\}
\]

\[
= \left\{ (U_1 - U_2) w ; w \in [0, 1]^p \right\} \tilde{+} \left\{ U_2 1_p \right\}.
\]

Since the Minkowski sum of between a polytope and a point is a translation; thereafter, the proposition follows directly from Definition 6. \(\square\)
B.3 Optimization of Objective (5.2) of the Binary Classifier

\[
\min_{\tilde{A}, \tilde{B}} \frac{1}{2} \left\| \tilde{G}_1 - G_1 \right\|_F^2 + \frac{1}{2} \left\| \tilde{G}_2 - G_2 \right\|_F^2 + \lambda_1 \left\| \tilde{G}_1 \right\|_{2,1} + \lambda_2 \left\| \tilde{G}_2 \right\|_{2,1}.
\]  
\text{(B.1)}

Note that \( \tilde{G}_1 = \text{Diag} \left[ \text{ReLU}(\tilde{B}(1,:)) + \text{ReLU}(-\tilde{B}(2,:)) \right] \tilde{A} \), \( \tilde{G}_2 = \text{Diag} \left[ \text{ReLU}(\tilde{B}(2,:)) + \text{ReLU}(-\tilde{B}(1,:)) \right] \tilde{A} \). Note that \( G_1 = \text{Diag} \left[ \text{ReLU}(B(1,:)) + \text{ReLU}(-B(2,:)) \right] A \) and \( G_2 = \text{Diag} \left[ \text{ReLU}(B(2,:)) + \text{ReLU}(-B(1,:)) \right] A \). For ease of notation, we refer to \( \text{ReLU}(\tilde{B}(i,:)) \) and \( \text{ReLU}(-\tilde{B}(i,:)) \) as \( \tilde{B}^+(i,:) \) and \( \tilde{B}^-(i,:) \), respectively. We solve the problem with coordinate descent an alternate over variables.

**Update \( \tilde{A} \).**

\[
\tilde{A} \leftarrow \arg\min_{\tilde{A}} \frac{1}{2} \left\| \text{Diag}(c_1) \tilde{A} - G_1 \right\|_F^2 + \frac{1}{2} \left\| \text{Diag}(c_2) \tilde{A} - G_2 \right\|_F^2 + \lambda_1 \left\| \text{Diag}(c_1) \tilde{A} \right\|_{2,1} + \lambda_2 \left\| \text{Diag}(c_2) \tilde{A} \right\|_{2,1},
\]

where \( c_1 = \text{ReLU}(B(1,:)) + \text{ReLU}(-B(2,:)) \) and \( c_2 = \text{ReLU}(B(2,:)) + \text{ReLU}(-B(1,:)) \). Note that the problem is separable per-row of \( \tilde{A} \). Therefore, the problem reduces to updating rows of \( \tilde{A} \) independently and the problem exhibits a closed form solution.

\[
\tilde{A}(i,:) = \arg\min_{\tilde{A}(i,:)} \frac{1}{2} \left\| c_1^i \tilde{A}(i,:) - G_1(i,:) \right\|_2^2 + \frac{1}{2} \left\| c_2^i \tilde{A}(i,:) - G_2(i,:) \right\|_2^2 + \lambda_1 \frac{1}{2} \left\| \text{Diag}(c_1) \tilde{A}(i,:) \right\|_{2,1} + \lambda_2 \frac{1}{2} \left\| \text{Diag}(c_2) \tilde{A}(i,:) \right\|_{2,1},
\]

\[
= \left( 1 - \frac{1}{2} \frac{\lambda_1 \sqrt{c_1^i} + \lambda_2 \sqrt{c_2^i}}{\frac{1}{2} (c_1^i + c_2^i)} \right) \left( \frac{c_1^i G_1(i,:) + c_2^i G_2(i,:)}{\frac{1}{2} (c_1^i + c_2^i)} \right).
\]

**Update \( \tilde{B}^+(1,:) \).**
\[ \tilde{B}^+ (1,:) = \arg \min_{\tilde{B}^+ (1,:)} \frac{1}{2} \| \text{Diag} (\tilde{B}^+ (1,:)) \tilde{A} - C_1 \|_F^2 + \lambda_1 \| \text{Diag} (\tilde{B}^+ (1,:)) \tilde{A} + C_2 \|_{2,1}, \]

\[ \text{s.t. } \tilde{B}^+ (1,:) \geq 0. \]

Note that \( C_1 = G_1 - \text{Diag} (\tilde{B}^- (2,:)) \tilde{A} \) and where \( \text{Diag} (\tilde{B}^- (2,:)) \tilde{A} \). Note the problem is separable in the coordinates of \( \tilde{B}^+ (1,:) \) and a projected gradient descent can be used to solve the problem in such a way as:

\[ \tilde{B}^+ (1,j) = \arg \min_{\tilde{B}^+ (1,j)} \frac{1}{2} \| \tilde{B}^+ (1,j) \tilde{A}(j,:) - C_1(j,:) \|_2^2 + \lambda_1 \| \tilde{B}^+ (1,j) \tilde{A}(j,:) + C_2(j,:) \|_2, \]

\[ \text{s.t. } \tilde{B}^+ (1,j) \geq 0. \]

A similar symmetric argument can be used to update the variables \( \tilde{B}^+ (2,:) \), \( \tilde{B}^+ (1,:) \) and \( \tilde{B}^- (2,:) \).

**B.4 Optimization of Objective 5.4 of the Multi Class Classifier**

**Update \( \tilde{A} \).**

\[ \tilde{A} = \arg \min_{\tilde{A}} \sum_{i=1}^{k} \frac{1}{2} \left( \| \text{Diag} (\tilde{B}^+ (i,:)) \tilde{A} - G_{i^+} \|_F^2 + \| \text{Diag} (\tilde{B}^- (i,:)) \tilde{A} - G_{i^-} \|_F^2 \right) + \lambda \left( \| \text{Diag} (\tilde{B}^+ (i,:)) \tilde{A} \|_{2,1} + \| \text{Diag} (\tilde{B}^- (i,:)) \tilde{A} \|_{2,1} \right). \]

Similar to the binary classification, the problem is separable in the rows of \( \tilde{A} \) and a closed form solution in terms of the proximal operator of \( \ell_2 \) norm follows naturally for each \( \tilde{A}(i,:) \).
Update $\tilde{B}^+(i,:)$. 

$$
\tilde{B}^+(i,:) = \arg\min_{\tilde{B}^+(i,:)} \frac{1}{2} \left\| \text{Diag} \left( \tilde{B}^+(i,:) \right) \tilde{A} - \tilde{G}_{i+} \right\|_F^2 + \lambda \left\| \text{Diag} \left( \tilde{B}^+(i,:) \right) \tilde{A} \right\|_{2,1}, \\
\text{s.t. } \tilde{B}^+(i,:) \geq 0.
$$

Note that the problem is separable per coordinates of $\tilde{B}^+(i,:)$ and each subproblem is updated as:

$$
\tilde{B}^+(i,j) = \arg\min_{\tilde{B}^+(i,j)} \frac{1}{2} \left\| \tilde{B}^+(i,j) \tilde{A}(j,:) - \tilde{G}_{i+}(j,:) \right\|_2^2 + \lambda \left\| \tilde{B}^+(i,j) \tilde{A}(j,:) \right\|_2, \text{ s.t. } \tilde{B}^+(i,j) \geq 0 \\
= \arg\min_{\tilde{B}^+(i,j)} \frac{1}{2} \left\| \tilde{B}^+(i,j) \tilde{A}(j,:) - \tilde{G}_{i+}(j,:) \right\|_2^2 + \lambda \left\| \tilde{B}^+(i,j) \tilde{A}(j,:) \right\|_2, \text{ s.t. } \tilde{B}^+(i,j) \geq 0 \\
= \max \left( 0, \tilde{A}(j,:)^\top \tilde{G}_{i+}(j,:) - \lambda \left\| \tilde{A}(j,:) \right\|_2 \right).
$$

A similar argument can be used to update $\tilde{B}^-(i,:) \forall i$. Finally, the parameters of the pruned network will be constructed $A \leftarrow \tilde{A}$ and $B \leftarrow \tilde{B}^+ - \tilde{B}^-$. 
B.5 Algorithm for Solving 6.3.

In this section, we are going to derive an algorithm for solving the following problem:

\[
\min_{\eta, \xi_{A_1}} D_1(\eta) + D_2(\xi_{A_1})
\]

subject to

\[
-\text{loss}(g(A_1(x_0 + \eta), t) \leq -1, \quad -\text{loss}(g(A_1 + \xi_{A_1}x_0, t) \leq -1, \quad (x_0 + \eta) \in [0,1]^n, \quad \|\eta\|_{\infty} \leq \epsilon_1, \quad \|\xi_{A_1}\|_{\infty,\infty} \leq \epsilon_2, \quad A_1\eta - \xi_{A_1}x_0 = 0.
\]

(B.2)

The function \(D_2(\xi_{A_1})\) captures the perturbation in the dual subdivision polytope such that the dual subdivision of the network with the first linear layer \(A_1\) is similar to the dual subdivision of the network with the first linear layer \(A_1 + \xi_{A_1}\). This can be generally formulated as an approximation to the following distance function \(d(\text{ConvHull}(Z_{\tilde{G}_1}, Z_{\tilde{G}_2}), \text{ConvHull}(Z_{G_1}, Z_{G_2}))\), where \(\tilde{G}_1 = \text{Diag}[\text{ReLU}(\tilde{B}(1,:)) + \text{ReLU}(-\tilde{B}(2,:))](\tilde{A} + \xi_{A_1})\), \(\tilde{G}_2 = \text{Diag}[\text{ReLU}(\tilde{B}(2,:)) + \text{ReLU}(-\tilde{B}(1,:))](\tilde{A} + \xi_{A_1})\), \(G_1 = \text{Diag}[\text{ReLU}(\bar{B}(1,:)) + \text{ReLU}(-\bar{B}(1,:))](\bar{A} + \xi_{A_1})\) and \(G_2 = \text{Diag}[\text{ReLU}(\bar{B}(2,:)) + \text{ReLU}(-\bar{B}(1,:))](\bar{A} + \xi_{A_1})\). In particular, to approximate the function \(d\), one can use a similar argument as in used in network pruning such that \(D_2\) approximates the generators of the zonotopes directly as follows:

\[
\begin{align*}
D_2(\xi_{A_1}) &= \frac{1}{2} \|\tilde{G}_1 - G_1\|^2_F + \frac{1}{2} \|\tilde{G}_2 - G_2\|^2_F \\
&= \frac{1}{2} \|\text{Diag}(\text{ReLU}(\bar{B}(1,:))\xi_{A_1})\|^2_F + \frac{1}{2} \|\text{Diag}(\text{ReLU}(-\bar{B}(1,:))\xi_{A_1})\|^2_F \\
&+ \frac{1}{2} \|\text{Diag}(\text{ReLU}(\bar{B}(2,:))\xi_{A_1})\|^2_F + \frac{1}{2} \|\text{Diag}(\text{ReLU}(-\bar{B}(2,:))\xi_{A_1})\|^2_F.
\end{align*}
\]

This can thereafter be extended to multi-class network with \(k\) classes as follows \(D_2(\xi_{A_1}) = \frac{1}{2} \sum_{j=1}^k \|\text{Diag}(\text{ReLU}(\bar{B}(j,:))\xi_{A_1})\|^2_F\). Following \[48\], we take \(D_1(\eta) = \frac{1}{2} \|\eta\|^2_2\). Therefore, we can write B.2 as follows:
\[
\min_{\eta, \xi_{A}} \mathcal{D}_1(\eta) + \sum_{j=1}^{k} \left\| \text{Diag}(B^+(j,:))\xi_{A} \right\|_F^2 + \left\| \text{Diag}(B^-(j,:))\xi_{A} \right\|_F^2.
\]

s.t. \[\text{loss}(g(A_1(x_0 + \eta), t) \leq -1, \quad \text{loss}(g((A_1 + \xi_{A})x_0), t) \leq -1,
\]
\[
(x_0 + \eta) \in [0, 1]^n, \quad \|\eta\|_{\infty} \leq \epsilon_1, \quad \|\xi_{A}\|_{\infty, \infty} \leq \epsilon_2, \quad A_1\eta - \xi_{A}x_0 = 0.
\]

To enforce the linear equality constraints \(A_1\eta - \xi_{A}x_0 = 0\), we use a penalty method, where each iteration of the penalty method we solve the sub-problem with ADMM updates. That is, we solve the following optimization problem with ADMM with increasing \(\lambda\) such that \(\lambda \to \infty\). For ease of notation, let’s denote \(\tilde{L}(x_0 + \eta) = \text{loss}(g(A_1(x_0 + \eta)), t)\), and \(\tilde{L}(A_1) = \text{loss}(g((A_1 + \xi_{A})x_0), t)\).

\[
\min_{\eta, z, w, \xi_{A}} \|\eta\|_2^2 + \sum_{j=1}^{k} \left\| \text{Diag}(\text{ReLU}(B(j,:))\xi_{A}) \right\|_F^2 + \left\| \text{Diag}(\text{ReLU}(-B(j,:))\xi_{A}) \right\|_F^2
\]
\[
+ \tilde{L}(x_0 + z) + h_1(w) + h_2(\xi_{A}) + \lambda\|A_1\eta - \xi_{A}x_0\|_2^2 + \tilde{L}(A_1).
\]

s.t. \(\eta = z, \quad z = w\).

where

\[
h_1(\eta) = \begin{cases} 
0, & \text{if } (x_0 + \eta) \in [0, 1]^n, \|\eta\|_{\infty} \leq \epsilon_1 \\
\infty, & \text{else}
\end{cases}
\]

\[
h_2(\xi_{A}) = \begin{cases} 
0, & \text{if } \|\xi_{A}\|_{\infty, \infty} \leq \epsilon_2 \\
\infty, & \text{else}
\end{cases}
\]

The augmented Lagrangian is given as follows:

\[
\mathcal{L}(\eta, w, z, \xi_{A}, u, v) := \|\eta\|_2^2 + \mathcal{L}(x_0 + z) + h_1(w) + \sum_{j=1}^{k} \left\| \text{Diag}(B^+(j,:))\xi_{A} \right\|_F^2
\]
\[
+ \left\| \text{Diag}(B^-(j,:))\xi_{A} \right\|_F^2 + \tilde{L}(A_1) + h_2(\xi_{A}) + \lambda\|A_1\eta - \xi_{A}x_0\|_2^2
\]
\[
+ u^T(\eta - z) + v^T(w - z) + \frac{\rho}{2}(\|\eta - z\|_2^2 + \|w - z\|_2^2).
\]
Thereafter, ADMM updates are given as follows:

\[
\begin{align*}
\eta^{k+1}, w^{k+1} &= \arg \min_{\eta, w} L(\eta, w, z^k, \xi_{A_1}^k, u^k, v^k), \\
z^{k+1} &= \arg \min_z L(\eta^{k+1}, w^{k+1}, z, \xi_{A_1}^k, u^k, v^k), \\
\xi_{A_1}^{k+1} &= \arg \min_{\xi_{A_1}} L(\eta^{k+1}, w^{k+1}, z^{k+1}, \xi_{A_1}, u^k, v^k).
\end{align*}
\]

\[
\begin{align*}
u^{k+1} &= u^k + \rho(\eta^{k+1} - z^{k+1}), \\
v^{k+1} &= v^k + \rho(w^{k+1} - z^{k+1}).
\end{align*}
\]

Updating \( \eta \):

\[
\eta^{k+1} = \arg \min_{\eta} \|\eta\|^2 + \lambda \|A_1 \eta - \xi_{A_1} x_0\|^2 + u^\top \eta + \frac{\rho}{2} \|\eta - z\|^2
\]

\[
= \left(2\lambda A_1^\top A_1 + (2 + \rho)I\right)^{-1} \left(2\lambda A_1^\top \xi_{A_1} x_0 + \rho z^k - u^k\right).
\]

Updating \( w \):

\[
w^{k+1} = \arg \min_w v^\top w + h_1(w) + \frac{\rho}{2} \|w - z^k\|^2
\]

\[
= \arg \min_w \frac{1}{2} \left\|w - \left(z^k - \frac{v^k}{\rho}\right)\right\|^2 + \frac{1}{\rho} h_1(w).
\]

The update \( w \) is separable in coordinates as follows:

\[
w^{k+1} = \begin{cases} 
\min(1 - x_0, \epsilon_1) & : z^k - 1/\rho v^k > \min(1 - x_0, \epsilon_1) \\
\max(-x_0, -\epsilon_1) & : z^k - 1/\rho v^k < \max(-x_0, -\epsilon_1) \\
z^k - 1/\rho v^k & : otherwise
\end{cases}
\]
Updating $z$:

$$
z^{k+1} = \arg \min_z \mathcal{L}(x_0 + z) - u^T z - v^T z + \frac{\rho}{2} \left( \|\eta^{k+1} - z\|_2^2 + \|w^{k+1} - z\|_2^2 \right).
$$

[49] showed that the linearized ADMM converges for some non-convex problems. Therefore, by linearizing $\mathcal{L}$ and adding Bergman divergence term $\eta^k/2\|z - z_k\|_2^2$, we can then update $z$ as follows:

$$
z^{k+1} = \frac{1}{\eta^k + 2\rho} \left( \eta^k z^k + \rho \left( \eta^{k+1} + \frac{1}{\rho} u^k + w^{k+1} + \frac{1}{\rho} v^k \right) - \nabla \mathcal{L}(z^k + x_0) \right).
$$

It is worthy to mention that the analysis until this step is inspired by [48] with modifications to adapt our new formulation.

Updating $\xi_A$:

$$
\xi_{A}^{k+1} = \arg \min_{\xi_A} \|\xi_A\|_F^2 + \lambda \|\xi_A x_0 - A_1 \eta\|_2^2 + \bar{\mathcal{L}}(A_1) \quad \text{s.t.} \quad \|\xi_A\|_{\infty, \infty} \leq \epsilon_2.
$$

The previous problem can be solved with proximal gradient methods.
C Papers Submitted and Under Preparation


  I was fortunate to collaborate with many researchers on several beautiful projects that are not related to the thesis. I list them below
