ON THE DECISION BOUNDARY OF DEEP NEURAL NETWORKS: A TROPICAL GEOMETRY APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

We use tropical geometry, a new development in the area of algebraic geometry, to provide a characterization of the decision boundary of a simple piece-wise linear binary neural network in the form (Affine, ReLU, Affine). Concretely, we show that the decision boundary is a subset of a tropical hypersurface, which is intimately related to a polytope formed by the convex-hull of two zonotopes. The generators of the zonotopes are precisely functions of the neural network parameters. Then, we utilize the characterization to propose a new tropical view of the lottery ticket hypothesis (Frankle & Carbin, 2019). We leverage the aforementioned characterization as a new set of tropical regularizes which tackle directly the decision boundary and further investigate its application in neural network pruning and adversarial attacks. ...

Gomaa: We should add here some important insights from the experiments in the last paragraph of the abstract.

Bibi: Things to be careful about to stay consistent.
(i) Use "non-linear" not nonlinear.
(ii) Use "piece-wise" not piece wise.
(iii) Use "element-wise" not element wise.
(iv) Use *i.e.*and *e.g.et al.*not i.e.,e.g. or et al.
(v) Make sure you use cite and citep appropriately.
(vi) Use EQref when referring to equations and ref when referring to figures/tables/plots/sections etc.
(vii) Lets stick to following notation. small f for the neural network. for Theorem 1 and the rest "f = H - Q" We will leave the G for the generator set

Gomaa: Should we abbreviate tropical geometry with TG in the whole text?

1 INTRODUCTION

Deep Neural Networks (DNNs) have recently demonstrated an outstanding performance across several research domains, including speech recognition (Hinton et al., 2012), computer vision (Krizhevsky et al., 2012), natural language processing (Bahdanau et al., 2015; Devlin et al., 2018), quantum chemistry (Schütt et al., 2017) and healthcare (Ardila et al., 2019; Zhou et al., 2019), among others (LeCun et al., 2015). Nonetheless, a rigorous interpretation of their success stories remains big challenge in the research community (Shalev-Shwartz & Ben-David, 2014; Montufar et al., 2014; Arora et al., 2018).

Tropical geometry is a relatively recent area at the interface between algebraic geometry and combinatorics (Maclagan & Sturmfels, 2015). It has experienced a rapid developing in recent years with applications in several fields, such as dynamic programming (Joswig & Schröter, 2019), linear programming (Allamigeon et al., 2015), multi-objective discrete optimization (Joswig & Loho, 2019), enumerative geometry (Mikhalkin, 2004), economics (Akian et al., 2009; Mai Tran & Yu, 2015) and DNNs (Zhang et al., 2018). In the ideal case, for every construction in algebraic geometry, there should be a combinatorial counterpart in tropical geometry. Thus, One may try to obtain results in an algebraic geometry scenario by investigating the, hopefully easier, tropical analogue first and then hope to prove that this implies the original statement in algebraic geometry. Furthermore, a tropical analogue to linear algebra and convex geometry is provided in Butkovič (2010) and Gaubert & Katz (2006), respectively.

In DNNs, Zhang et al. (2018) showed an equivalency between the family of feedforward neural networks with piece-wise linear activation, integer weight matrices and real-valued biases and the

family of tropical rational maps, ratio between two multi-variate polynomials in tropical algebra. Moreover, they showed that the a zonotope, a Minkowski sum of a set of line segments, is the building blocks for DNNs. Also, they related the decision boundary of DNN to major objects in tropical geometry called tropical hypersurfaces, in addition to proving that DNNs are exponentially expressive[1] than shallow networks.

Thus in this paper, we took the previous results of Zhang et al. (2018) some steps further. Specifically, for a simple neural network in the form (Affine, ReLU, Affine) with two output classes, we give a concrete formulation of a super-set for the decision boundary of DNNs as a convex hull of two zonotopes, which is general a polytope. Then, we leverage this formulation of the polytope and its geometry to analyze DNNs and try to unravel some of the interesting phenomena in DNNs. Then, with the help of this formulation, we could demonstrate a support and geometric explanation of the recently surprising lottery ticket hypothesis (Frankle & Carbin, 2019). The authors surmise the existence of *winning lottery tickets*, 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. They propose a very simple algorithm to find such lottery tickets within larger networks by deleting small weights and retrain. Furthermore, we use the formulation of the polytope of the decision boundary to propose a new geometric prospective for two interesting application in DNNs, namely network pruning and adversarial attacks design. Pruning is a method to reduce the size of DNNs through compression. In our tropical network pruning, we reduce redundant information by removing the network parameters that do not contribute much to the geometry of the decision boundary object. Moreover, in our proposed white-box targeted tropical adversarial attack, we design perturbation to the input space in order to change the geometry of the decision boundary polytope drastically to classify a certain input vector to a certain class of our choice. Lastly, we show how to naturally extend theses results to deeper networks with more than two output classes.

The main contributions of this paper can be summarized into three folds. **(i)** We provide a new prospective of the decision boundary of the DNNs for a simple piece-wise linear function (Affine, ReLU, Affine). Then, we show how to extend this simple structure to deeper networks. **(ii)** We demonstrate a confirmation for the lottery ticket hypothesis (Frankle & Carbin, 2019) using a geometric prospective. **(iii)** We show some interesting applications of tropical geometry—with the help of the geometrical view of the decision boundary—in network pruning and adversarial attacks.

2 Preliminaries to Tropical Geometry

We provide here some preliminaries for analyzing DNNs using tropical geometry. For a thorough detailed review for tropical geometry, we refer the reader to Itenberg et al. (2009); Maclagan & Sturmfels (2015).

Definition 1. *(Tropical Semiring) The tropical semiring*[2] $\mathbb{T}$ *is the triplet* $\{\mathbb{R} \cup \{-\infty\}, \oplus, \odot\}$*, where* $\oplus$ *and* $\odot$ *are the tropical addition and tropical multiplication, respectively which are defined as* [3]*:*

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

It can be readily shown that $-\infty$ *is the additive identity and* 0 *is the multiplicative identity.*

Given the previous definition, one can equip this semiring with a tropical power.

Definition 2. *(Tropical Powers) For* $x \in \mathbb{T}$*,* $a \in \mathbb{N}$*, a tropical power is defined as follows:* $x^{\odot a} = x \odot x \cdots \odot x = a.x$ *where* $a.x$ *is standard multiplication. For ease of notation, we write* $x^{\odot a}$ *as* x^a*.*

Now we are in a position to define tropical monomials, tropical polynomials, their solution sets and tropical rational functions. Constructing a tropical monomial is analogues to standard monomials.

[1]The measure used in (Zhang et al., 2018) is the complexity of DNNs, which are defined by the number of linear regions that the function represented by the neural network can divide the input space into. The more number of linear regions, the more complexity of DNNs, the more expressiveness of DNNs .

[2]A semiring is a ring that lacks an additive inverse.

[3]The choice of $\max$ is not canonical and $\min$ can be used instead. This is because the equality $\max\{x, y\} = -\min\{-x, -y\}$ establishes an isomorphism of the two semirings $(\mathbb{R}, \max, +)$ and $(\mathbb{R}, \min, +)$. In this case, the additive identity will be ∞, with the semiring triplet $\{\mathbb{R} \cup \{\infty\}, \oplus, \odot\}$.

Definition 3. *(Tropical Monomials) A tropical monomial in d-variables is the tropical product of d-variables each raised tropically to a_i which can be expressed in the form $c \odot x_1^{a_1} \odot x_2^{a_2} \cdots \odot x_d^{a_d}$ where $c \in \mathbb{R} \cup \{-\infty\}$ and $a_i \in \mathbb{N}$. For ease, we use the more compact vector notation $\mathbf{x}^{\mathbf{a}} = c \odot x_1^{a_1} \odot x_2^{a_2} \cdots \odot x_d^{a_d}$ where $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{a} \in \mathbb{R}^d$.*

Definition 4. *(Tropical Polynomials) A tropical polynomial is a tropical sum of tropical monomials. Thus, for $\mathbf{x}_i \in \mathbb{T}^d$, $c_i \in \mathbb{R}$ and $\mathbf{a}_i \in \mathbb{N}^d$, a d-variable tropical polynomial with n monomials[4] f can be expressed as*

$$f(\mathbf{x}) = (c_1 \odot \mathbf{x}^{\mathbf{a}_1}) \oplus (c_2 \odot \mathbf{x}^{\mathbf{a}_2}) \oplus \cdots \oplus (c_n \odot \mathbf{x}^{\mathbf{a}_n}).$$

From now onwards, we will denote the tropical product $c_i \odot \mathbf{x}^{\mathbf{a}_i}$ as $c_i \mathbf{x}^{\mathbf{a}_i}$ for ease.

Definition 5. *(Tropical Rational Functions) A tropical rational function is a standard difference or equivalently, a tropical quotient of two tropical polynomials as $f(\mathbf{x}) - g(\mathbf{x}) = f(\mathbf{x}) \oslash g(\mathbf{x})$.*

It is now clear that tropical polynomials define a convex function since in standard operations they are the equivalent to taking a pointwise maximum over a set of affine functions. Thereafter, tropical rational functions are difference of convex functions (DC) Hartman et al. (1959); Maclagan & Sturmfels (2015). In classical algebraic geometry, the solution set to polynomials define hyperspaces or an algebraic curve for polynomials of two variables. A natural extension to this for tropical polynomials is tropical hypersurfaces.

Definition 6. *(Tropical Hypersurfaces) A tropical hypersurface of a tropical polynomial $f(\mathbf{x}) = c_1 \mathbf{x}^{\mathbf{a}_1} \oplus \cdots \oplus c_n \mathbf{x}^{\mathbf{a}_n}$ is the set of points $\mathbf{x}$ where f is attained by two or more monomials in f, i.e.*

$$\mathcal{T}(f) := \{\mathbf{x} \in \mathbb{R}^d : c_i \mathbf{x}^{\mathbf{a}_i} = c_j \mathbf{x}^{\mathbf{a}_j} \text{ for some } \mathbf{a}_i \neq \mathbf{a}_j\}.$$

It is easy to see that the tropical hypersurface divides the domain of f into convex regions where f is linear in each region. Analogous to classical polynomials, every tropical polynomial can be associated with a Newton polytope.

Definition 7. *(Newton Polytopes) The Newton polytope of a tropical polynomial $f(\mathbf{x}) = c_1 \mathbf{x}^{\mathbf{a}_1} \oplus \cdots \oplus c_n \mathbf{x}^{\mathbf{a}_n}$ is the convex hull of the exponents $\alpha_i \in \mathbb{N}^d$ regarded as points in $\mathbb{R}^d$, i.e.*

$$\Delta(f) := \textit{ConvHull}\{\mathbf{a}_i \in \mathbb{R}^d, c_i \neq -\infty, i = 1, 2, \ldots, n\}.$$

A tropical polynomial determines a dual subdivision which can thus be constructed by projecting the collection of upper faces in $\mathcal{P}(f) := \text{ConvHull}\{(\mathbf{a}_i, c_i) \in \mathbb{R}^d \times \mathbb{R} : i = 1, \ldots, n\}$ to $\mathbb{R}^d$. That is to say that the dual subdivision determined by f is given as $\delta(f) := \{\pi(p) \subset \mathbb{R}^d : p \in \text{UF}(\mathcal{P}(f))\}$ where $\pi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$. It has been shown by Maclagan & Sturmfels (2015) that the tropical hypersurface $\mathcal{T}(f)$ is the (d-1)-skeleton of the dual to $\delta(f)$. This means that each vertex of $\delta(f)$ corresponds to one region in $\mathbb{R}^d$ where f is linear. Therefore, the number of vertices in $\mathcal{P}(f)$ provides an upper bound to the number of linear regions of f.

The work of Zhang et al. (2018) showed an intimate relation between a feedforward neural network with piece-wise linear activation function $v : \mathbb{R}^d \to \mathbb{R}^d$, under the assumption of integer-valued network weights, and tropical rational maps. They presented the following Theorem.

Theorem 1. *(Tropical characterization of neural networks, Zhang et al. (2018)). A feedforward neural network with integer weights and real biases with piece-wise linear activation functions is a function , $f : \mathbb{R}^d \to \mathbb{R}^p$ whose coordinates are tropical rational functions of the input, i.e.,*

$$f(\mathbf{x}) = H(\mathbf{x}) \oslash Q(\mathbf{x}) = H(\mathbf{x}) - Q(\mathbf{x}),$$

where H and Q are tropical polynomials.

While this result is new in the context of tropical geometry, it is not utterly surprising. This is since it is known that any piece-wise linear function can be presented as the difference of a set of two max functions over a set of hyperplanes. More formally,

Theorem 2. *(Melzer (1986)) A piece-wise linear function $f(\mathbf{x})$ is a function of the following form :*

$$f(\mathbf{x}) = \max_{i \in [m]} \{\mathbf{a}_i^\top \mathbf{x}\} - \max_{j \in [n]} \{\mathbf{b}_j^\top \mathbf{x}\}, \text{ where } [m] = \{1, \ldots, m\} \text{ and } [n] = \{1, \ldots, n\}.$$

It is now easy to see that each of the two maxima above is a tropical polynomial recovering the result of Zhang et al. (2018) Theorem 1.

[4]Here, we assume that $\mathbf{a}_i \neq \mathbf{a}_j$ for any $i \neq j$.

A plot on explaining relation between the decision boundary/hypersurfaces and the polytope.

Figure 1: Very Sad Caption ..

3 DECISION BOUNDARIES OF DEEP NEURAL NETWORKS AS POLYTOPES

In this section, we analyze the decision boundaries of binary networks in the form (Affine, ReLU, Affine) using tropical geometry. For ease, we use ReLUs as non-linear activations but any other peice-wise linear function can be used too. The functional form of this binary network is: $f(\mathbf{x}) = \mathbf{B}\max\left(\mathbf{A}\mathbf{x} + \mathbf{c}_1\right) + \mathbf{c}_2$ where $\max(.)$ is an element-wise operator. The output of the network f[5] can be the output of the logits of d classes. The affine mappings can be of any size, and throughout this section, we assume[6] that $\mathbf{A} \in \mathbb{Z}^{p\times n}$, $\mathbf{B} \in \mathbb{Z}^{2\times p}$, $\mathbf{c}_1 \in \mathbb{R}^p$ and $\mathbf{c}_2 \in \mathbb{R}^2$. Now since f is a piece-wise linear function; each output class can be expressed as tropical rational. Let f_1 and f_2 refer to the first and second output classes, respectively, thus one can write $f_1 = H_1(\mathbf{x}) \oslash Q_1(\mathbf{x})$ and $f_2 = H_2(\mathbf{x}) \oslash Q_2(\mathbf{x})$ where $\mathbf{H}_1, \mathbf{H}_2, \mathbf{Q}_1$ and $\mathbf{Q}_2$ are tropical polynomials. Before presenting our main result, we start first by defining zonotopes and their properties as they will play an essential role in later analysis.

Definition 8. *Let $\mathbf{b}^1, \dots, \mathbf{b}^n \in \mathbb{R}^d$. The zonotope formed by $\mathbf{b}^1, \dots, \mathbf{b}^n \in \mathbb{R}^d$ is defined as $\mathcal{Z}(\mathbf{b}^1, \dots, \mathbf{b}^n) := \{\sum_{i=1}^n x_i \mathbf{b}^i : 0 \leq x_i \leq 1, i = 1, \dots, n\}$ or equivalently with respect to the generator matrix $\mathbf{B}$ where $\mathbf{b}^i = \mathbf{B}(:, i)$ as $\mathcal{Z}_{\mathbf{B}} := \{\mathbf{B}\mathbf{x} : \forall \mathbf{x} \in [0, 1]^n\}$.*

A common used definition for zonotopes is the Minkowski sum of the line segments that start from origin $\mathbf{b}^1, \dots, \mathbf{b}^n \in \mathbb{R}^d$. It is also well known that the number of vertices of a zonotope is polynomial in the number of line segments. That is to say, $|\text{vert}\left(\mathcal{Z}\left(\mathbf{B}\right)\right)| \leq 2\sum_{i=0}^{d-1}\binom{n-1}{i} = \mathcal{O}\left(n^{d-1}\right)$ (Gritzmann & Sturmfels, 1993). In what follows, we present our main results where the network $\mathbf{g}(\mathbf{x})$ has no biases, *i.e.*$\mathbf{c}_1 = \mathbf{0}$ and $\mathbf{c}_2 = \mathbf{0}$ for ease of presentation, but the results generalize without any loss of generality for arbitrary biases. Now we are ready to present our main results.

Proposition 1. *For a bias-free neural network in the form of $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^2$ where $\mathbf{A}$ and $\mathbf{B}$ have integer weights, and let $R(x) = H_1(\mathbf{x}) \odot Q_2(\mathbf{x}) \oplus H_2(\mathbf{x}) \odot Q_1(\mathbf{x})$ be a tropical polynomial, then:*

- *if the decision boundaries of f is given by the set $\mathcal{B} = \{x \in \mathbb{R}^n : f_1(\mathbf{x}) = f_2(\mathbf{x})\}$, then we have $\mathcal{B} \subseteq \mathcal{T}\left(R(x)\right)$.*
- *$\delta\left(R(\mathbf{x})\right) = \textit{ConvexHull}\left(\mathcal{Z}_{G_1}, \mathcal{Z}_{G_2}\right)$ where $\mathcal{Z}_{G_1}$ is a zonotope in $\mathbb{R}^n$ with line segments $\mathbb{R}^n$ $\{(\mathbf{B}(1,j)^+ + \mathbf{B}(2,j)^-)[\mathbf{A}(j,:)^+, \mathbf{A}(j,:)^-]_{j=1}^p\}$ with shift $(\mathbf{B}(1,j)^- + \mathbf{B}(2,j)^+)\mathbf{A}^-$ while $\mathcal{Z}_{G_2}$ is a zonotope in $\mathbb{R}^n$ with line segments $\{(\mathbf{B}(1,j)^- + \mathbf{B}(2,j)^+)[\mathbf{A}(j,:)^+, \mathbf{A}(j,:)^-]_{j=1}^p\}$ with shift $(\mathbf{B}(1,j)^+ + \mathbf{B}(2,j)^-)\mathbf{A}^-$.*

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

The proof is left for the appendix. The previous result bridges the gap between the behaviour of the decision boundaries $\mathcal{B}$, through the superset $\mathcal{T}\left(R(x)\right)$, and the well-structured geometric object

[5] f includes convolutional layers since ($\mathbf{A}$,$\mathbf{B}$) can have a circular/toeplitz structure.

[6] There is no loss of generality with the assumption of integer weights instead of real weights, *i.e.*$\mathbf{A}$ and $\mathbf{B}$. This is justified as the set of rational numbers is dense in the set of real numbers, *i.e.*real weights can be approximated arbitrary closely by rationals and then we clear denominators in the rational weights by multiplying by the least common multiple of the denominators to obtain integer weights (Zhang et al., 2018).

The intuition behind the compression approach we proposed ..

Figure 2: Very Sad Caption ..

$\delta\left(R(x)\right)$ which is the convexhull of two zonotopes. Figure 2, details the intimate relation between the tropical hypersurface $\mathcal{T}\left(R(x)\right)$ and the corresponding polytope $\delta\left(R(x)\right)$. Moreover, Figure 2 shows how one can recover the superset to the decision boundaries $\mathcal{T}\left(R(x)\right)$ from the given polytope $\delta\left(R(x)\right)$. Details of the relation between the Tropical hypersurface and the dual subdivision can be found in the work of (Brugallé & Shaw, 2014; Maclagan & Sturmfels, 2015). This result now allows for a new set of geometric regularizers that directly tackle the decision boundaries.

Bibi: The paragraph below this is not finalized as we may want to move this to the appendix.

While there exists fast algorithms that enumerate the vertices of a zonotope given a set of line segments Stinson et al. (2016), the algorithms always assume the line segments has zero as one end point. However, Proposition 1 assumes the line segments can have arbitrary end points. To this end, we present the next result that allows for a transformation of arbitrary end points of line segments to a generator matrix that has all linse segments with zero as a startign end point. Such a result is essential towards efficient computation of the zonotopes in Proposition 1.

Proposition 2. *Consider n line segments in $\mathbb{R}^d$ with arbitrary two end points as follows $\{[\mathbf{u}_1^i, \mathbf{u}_2^i]\}_{i=1}^n$. The Minkowski sum between this set of points is equivalent to the Minkowski sum to the line segments $\{[\mathbf{u}_1^i - \mathbf{u}_2^i, \mathbf{0}]\}_{i=1}^n$ with a shift of $\sum_{i=1}^n \mathbf{u}_2^i$.*

To this end, by the help of the geometry of the polytope of the decision boundary formulation in Proposition 1, we show the validation of the lottery ticket hypothesis and propose a new prospective in network pruning and design of adversarial attacks.

4 Tropical View to the Lottery Ticket Hypothesis

In this section, we demonstrate a confirmation of the lottery ticket hypothesis (Frankle & Carbin, 2019) by the help of the formulation of the decision boundary related polytope from Proposition 1. The main idea of the lottery ticket is summarized as follows: prune the network, then initialize the pruned network with the same initialization that was used in the original training settings, train with the same number of epochs and the accuracy will be at least as same as the original model. This idea gave insight to the importance of initialization when training the models. A good question to ask here is what is special about the initializing the pruned model with the same initialization?. To answer this question lets have a look to the decision boundary's polytope under different initialization. We trained a single hidden layer classifier, and prune it by removing the least important x percentage of rows. The pruned network is retrained using different initializations, which are the following: (i) same initialization as the original network was initialized. (ii) Xavier initialization. (iii) Sampled from Standard Gaussian. (iv) Sampled from zero-mean Gaussian with variance of 0.1. (v) Glorot-Bengio initialization. We can see that the only initialization that preserves the same shape of the decision boundary's polytope is the same initialization as the originally trained network, otherwise the shape of the decision boundary will change and hence the the decision boundary will change and hence you might not achieve the same accuracy of the original model. This gives insight that same initialization revolutionize the zonotopes of the network to achieve the same decision boundary's polytope and hence reaching to the same accuracy.

5 TROPICAL NETWORK PRUNING

Network pruning has been identified as an effective approach towards reducing the computational cost and memory efficiency during network inference time. While pruning dates back to the work of LeCun et al. (1990) and Hassibi & Stork (1993) which prunes weights based on Hessian of the loss function, it has recently gained more attention. This is due to the fact that most neural networks are over-paramertizing the commonly used datasets. In network pruning, the task is to find a smaller subset of the network parameters such that the resultant smaller network has a similar decision boundary to the original network and thus achieving a similar accuracy to the over-parameterized original network.

In this section, we show how a new geometric approach towards network pruning. In particular, as indicated by Proposition 1, preserving the polytope $\delta(R(\mathbf{x})$ preserves the super set to the decision boundary and thereafter the decision boundaries. To this end, we propose in this work geometric approach that acts on the polytope directly towards pruning the network while maintaining the decision boundaries.

5.1 MOTIVATIONAL INSIGHT

Since the decision boundary's polytope is coming from a convex hull of two zonotopes, and each is coming from minkowski sum of line segments. A possible scenario that can happen is that some of the vertices of these zonotopes lies inside the zonotope. Hence, removing the corresponding generators will not affect their shape and size, and the decision boundary's polytope as well. Figure

Motasem: Refer to the pipe line figure

shows how can we remove a part from the network without affecting the decision boundary's polytope. After giving this motivation, the rest of this section will formulate the problem of how can we prune the network based on geometrical intuition and will show some experimental results as well.

5.2 PROBLEM FORMULATION

Since the decision boundary of a neural network is a subset of the perpendicular lines, in case of 2 dimensional dual subdivision, of the dual subdivision $\delta(R(\mathbf{x}))$; thereafter, several natural questions may arise. For instance, given an over-parameterized neural network $g(\mathbf{x}) = \mathbf{B}\max(\mathbf{A}\mathbf{x}, \mathbf{0})$, may one construct a new neural network, parametrized by the sparser $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$, such that the resultant network gives rise to an approximate dual subdivision $\tilde{\delta}(R(\mathbf{x}))$ to $\delta(R(\mathbf{x}))$ that will consequently preserve the decision boundary? To address this question, we propose the following general optimization problem.

$$\min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} d\Big(\tilde{\delta}(R(\mathbf{x})), \delta(R(\mathbf{x}))\Big) = \min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} d\Big(\text{ConvexHull}\left(\mathcal{Z}_{\tilde{\mathbf{G}}_1}, \mathcal{Z}_{\tilde{\mathbf{G}}_2}\right), \text{ConvexHull}\left(\mathcal{Z}_{\mathbf{G}_1}, \mathcal{Z}_{\mathbf{G}_2}\right)\Big). \quad (1)$$

The function d defines a distance between two geometric structures. Note that as per Proposition 1, the generators $\tilde{\mathbf{G}}_1$ and $\tilde{\mathbf{G}}_2$ are functions of $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$. However, the aforementioned optimization problem can be challenging, since its optimizing a convex hull of zonotopes that are generated by the network parameters. To this end, we observe that if $\tilde{\mathbf{G}}_1 \approx \mathbf{G}_1$ and that $\tilde{\mathbf{G}}_2 \approx \mathbf{G}_2$, we then have that $\delta(\tilde{R}(\mathbf{x})) \approx \delta(R(\mathbf{x}))$. However, for pruning purposes, one can observe from Proposition 1 that if the generators $\tilde{\mathbf{G}}_1$ and $\tilde{\mathbf{G}}_2$ had fewer number of line segments, *i.e.*rows i think this should be rows but we can confirm later, this corresponds to fewer number of rows in the weight matrix $\tilde{\mathbf{A}}$. Thereafter, we propose the following optimization problem as a surrogate to Problem 1.

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

The matrix mixed norm for $\mathbf{C} \in \mathbb{R}^{n\times k}$ is defined as $\|\mathbf{C}\|_{2,1} = \sum_{i=1}^{n} \|\mathbf{C}(i,:)\|_2$ which incites complete rows of $\mathbf{C}$ to be zero where $\tilde{\mathbf{G}}_1 = \text{Diag}\Big[\text{ReLU}(\tilde{\mathbf{B}}(1,:)) + \text{ReLU}(-\tilde{\mathbf{B}}(2,:))\Big]\tilde{\mathbf{A}}$ and that $\tilde{\mathbf{G}}_2 = \text{Diag}\Big[\text{ReLU}(\tilde{\mathbf{B}}(2,:)) + \text{ReLU}(-\tilde{\mathbf{B}}(1,:))\Big]\tilde{\mathbf{A}}$. Note that $\text{Diag}(\mathbf{v})$ rearranges the elements of the

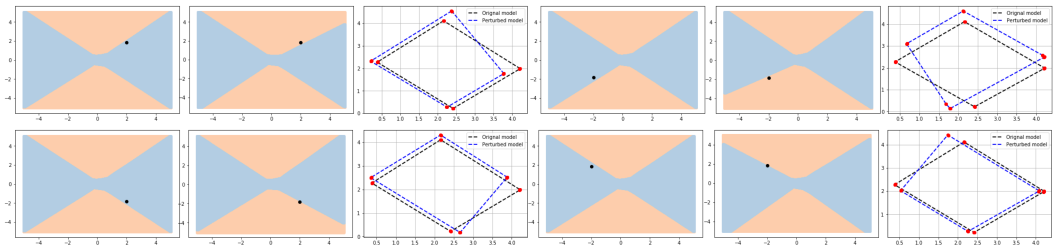

Figure 3: An intuitive view of the effect of tropical adversarial attack on a synthetic dataset with two classes in four different scenarios for the black input point. From left to right: Decision boundary of Original model, perturbed model and decision boundary polytopes.

Bibi: The font size of the axis and legend has to increase. The white space in the plot should be taken off. The caption will be changed later.

vector $\mathbf{v}$ on a diagonal matrix. The problem can be solved with coordinate descent where proximal gradient descent can be applied per co-ordinate. The details of the derivation of this objective and the solver is left for the **supplementary material**.

Extension to Deeper Networks. The extension of the above formulation to deeper networks follows naturally from the fact that deeper networks are composed of blocks of (Affine-ReLU-Affine). Therefore, Each block can be compressed separately and hence compressing the deeper network can be achieved.

5.3 Experiments of Tropical Pruning:

In this section we demonstrate the strength of the geometrical pruning on different networks and different datasets. We compare our method to other pruning methods like Class Blind, Class Uniform and Class Distribution Han et al. (2015); See et al. (2016), We focus on networks that has massive fully connected layers like AlexNet, and VGG16, each trained on SVHN, CIFAR10, and CIFAR 100 datasets. We observe that we can prune more than 90% of the classifier of both networks without affecting the accuracy. Moreover, we can boost the percentage of compression using our method without affecting the accuracy by simply retraining the biases only of the network.

Setup. We adapt VGG16 on the mentioned datasets by changing the sizes of the fully connected layers from (25088-4096-1000) to (512-512-10) for CIFAR10 and SVHN, and (512-512-100) for CIFAR100, and also removing the adaptive average pooling layer. In AlexNet, we modify the fully connected layers to (256-512-10) for CIFAR10 and SVHN, and to (256-512-100) for CIFAR100.

Motasem: We should insert and cite the needed figures for compression and retraining the biases

Our experiments are divided into four parts, pruning the network without fine tunning, pruning and then fine tunning the biases of the classifier, fine tunning the biases of the network, and finally fine tunning the feature block along with the biases of the classifier. The results show that we can prune more than 90% of the classifier of both AlexNet and VGG16 trained on SVHN and CIFAR10 without loosing more than 1% of accuracy. Not only, that, we can still boost the pruning percentage by few more percents by retraining the biases only. Since we are trying to preserve the shape of the decision boundary, the biases are the responsible for the offset caused to the decision boundary. Retraining the biases will be responsible of the offset required to the decision boundary of the pruned network. It can also be noticed that the pruned percentage of the network trained on CIFAR100 will be less since it is a harder dataset. Although we simplified the main objective in the formulation and relaxed it, we are yet achieving as good, and sometimes better, as the other pruning methods. We also want to emphasize that our contribution does not lie in beating state of the art methods of pruning, but resides in giving a geometrical based methods for pruning.

6 TROPICAL ADVERSARIAL ATTACKS

While deep neural networks have shown impressive performance on the recognition task, they are still susceptible to adversarial attacks. That is to say, adding small imperceptible noise, referred to as adversarial attacks, at the input of deep neural networks can hinder their performance. It has been shown that such attacks are rather easy to generate, in some cases, as easy as performing a gradient ascent step on the loss function for a single iteration (Goodfellow et al., 2014).

In this work, we provide a tropical view to this nuisance. In particular, we show how Proposition 1 can be utilized to construct a geometric based targeted white-box adversarial attacks from a new fresh dual perspective.

6.1 MOTIVATIONAL INSIGHT

Proposition 1 gives a characterization of the decision boundary polytope. In our tropical adversarial attack, we propose to perturb the first affine layer of the network and equivalently, the input image in order to change the geometry of the polytope and accordingly move the decision boundary causing the misclassification of the input image. This intuition is presented in Figure 5.3 which shows the affect of the perturbation of the model on the polytope of the decision boundary and how this change of the polytope causes the misclassification of the black input point in different orientations. In the following, we formulate the problem, present our novel dual view of the attack and provide some experimentation results.

6.2 DUAL VIEW TO ADVERSARIAL ATTACKS

For a classifier $f : \mathbb{R}^n \to \mathbb{R}^d$ and input $\mathbf{x}_0$ that is classified as k, a standard formulation for targeted adversarial attacks that flip the classifier prediction to a particular class t, is given as follows

$$\min_{\eta} \mathcal{D}(\eta) \;\; \text{subject to} \;\; \arg\max_{i} f_i(\mathbf{x}_0 + \eta) = t \neq k.$$

The former objective aims at designing input noise η, with low energy captured by $\mathcal{D}$, such that the the new sample $\mathbf{x}_0 + \eta$ crosses the decision boundary of f to a new classification region. In what follows we present a dual view to adversarial attacks.

Instead of designing a sample noise η such that $\eta + \mathbf{x}_0$ belongs to a new decision region, one can instead fix $\mathbf{x}_0$ and perturb the network parameters to move the decision boundary in a way that $\mathbf{x}_0$ appears in a new classification region. In particular, let $\mathbf{A}_1$ be the first linear layer of f, such that $f(\mathbf{x}_0) = g(\mathbf{A}_1\mathbf{x}_0)$. One can now perturb $\mathbf{A}_1$ to alter the decision boundaries as follows:

$$g((\mathbf{A}_1 + \xi_{\mathbf{A}_1})\mathbf{x}_0) = g\left(\mathbf{A}_1\mathbf{x}_0 + \xi_{\mathbf{A}_1}\mathbf{x}_0\right). \tag{3}$$

One can now observe that traditional adversarial attacks, *i.e.* $f(\mathbf{x}_0 + \eta) = g(\mathbf{A}_1\mathbf{x}_0 + \mathbf{A}_1\eta)$, is intimately related to perturbing the parameters of the first linear layer through the liner system

$$\mathbf{A}_1\eta = \xi_{\mathbf{A}_1}\mathbf{x}_0 \tag{4}$$

To this end, Proposition 1 provides an excellent approach towards geometrically perturbing decision regions as a mean towards constructing adversarial attacks. In particular, since the perpendicular lines to the dual subdivision of the newton polytope of a given neural network is a super-set to the decision boundaries, thus one can hope to design $\xi_{\mathbf{A}_1}$ that makes minimal perturbation to the dual subdivision and change the class of the point to the targeted class. Based on the previous observation, we formulate the following optimization problem:

$$\begin{aligned}
\min_{\eta, \xi_{\mathbf{A}_1}} \quad & \mathcal{D}_1(\eta) + \mathcal{D}_2(\xi_{\mathbf{A}_1}) \\
\text{s.t.} \quad & \mathcal{L}(x + \eta) \leq 0, \quad \bar{\mathcal{L}}(\mathbf{A}_1) \leq 0, \\
& (\mathbf{x}_0 + \eta) \in [0, 1]^n, \quad \|\eta\|_\infty \leq \epsilon_1, \quad \|\xi_{\mathbf{A}_1}\|_{\infty,\infty} \leq \epsilon_2, \quad \mathbf{A}_1\eta - \xi_{\mathbf{A}_1}\mathbf{x}_0 = 0.
\end{aligned}$$

The first and second constraints ensures that the class changed to the target class (Carlini & Wagner, 2016) by perturbing either the input x by a perturbation η, or perturbing the first linear layer $\mathbf{A}_1$ by $\xi_{\mathbf{A}_1}$. Carlini & Wagner (2016) proposed one valid choice for $\mathcal{L}$ and $\bar{\mathcal{L}}$ as follows:

$$\mathcal{L}(x+\eta) = -loss_{F,t}(x+\eta) + 1,$$
$$\bar{\mathcal{L}}(\mathbf{A}_1) = -loss_{\bar{F},t}(\mathbf{A}_1; x) + 1,$$

where $loss$ is nothing but a cross entropy loss, and F is the output of the network before the softmax layer when the input is perturbed (*i.e.* at $x+\eta$). $\bar{F}$ is the output of the network after perturbing the first linear layer A_1 evaluated on the clean input x. Also, the second constraint ensures that the change in the first linear layer is in the direction of fooling the classifier. Moreover, The third and fourth constraints guarantee that the perturbation is bounded and the perturbed input is feasible. Furthermore, The fifth constraint is to limit the perturbation on the decision boundary's polytope, and the last constraint followed from equation 4.

It can be noticed that the first and second constraint can be turned into a soft constraint embedded in the objective function. Hence, we can rewrite the above problem in the following relaxed form:

$$\min_{\eta, \xi_{\mathbf{A}_1}} \quad \mathcal{D}_1(\eta) + \mathcal{D}_2(\xi_{\mathbf{A}_1}) + \mathcal{L}(x+\eta) + h_1(\eta) + h_2(\xi_{\mathbf{A}_1}) + \lambda \|A\eta - \xi_{\mathbf{A}_1} x\|_2^2 + \bar{\mathcal{L}}(\mathbf{A}_1), \tag{5}$$

where:

$$h_1(\eta) = \begin{cases} 0, & \text{if } (x+\eta) \in [0,1]^n \ \ \& \ \ \|\eta\|_\infty \leq \epsilon_1 \\ \infty, & else \end{cases} \qquad h_2(\xi_{\mathbf{A}_1}) = \begin{cases} 0, & \text{if } \|\xi_{\mathbf{A}_1}\|_{\infty,\infty} \leq \epsilon_2 \\ \infty, & else \end{cases}.$$

The above problem can be solved efficiently using the alternating direction method of multipliers (ADMM), and it is left for the **supplementary material**.

Motasem: Dont forget to add Why did we choose D(ΔA) like this in the appendix

6.3 Experiments of Tropical Adversarial Attacks:

We show the results of the attacks on two sets of data. First we construct a binary classification problem to see the effect of our attack on the decision boundary and the decision boundary's polytope. Second, we evaluate our attack on a neural network trained on MNIST dataset. Since the attack is trying to make a minimal perturbation to the decision boundary, therefore, we expect only few edges of the decision boundary's polytope to change its orientation, and the rest will stay the same.

Setup. For the binary classification problem, we trained a single hidden layer neural network with 50 hidden nodes. We sampled one data point "Green dot" in the figure and attacked it. For the MNIST dataset, we trained five fully connected layers neural network up to 97% validation accuracy. We design the attack to both the input x and the first linear layer A.

Results. For the binary classification problem, both of the input perturbation and the layer perturbation fooled the network. Surprisingly, only one of the edges of the decision boundary has moved as discussed in the motivation subsection. The effect of that on the decision boundary's polytope is just a rotation on one of the edges, as expected. We show different scenarios of the aforementioned experiment. Moreover, we design perturbation to miss-classify MNIST images. We successfully fooled the model as shown in the figure with $\epsilon = 0.1$. Moreover, we can increase the sparsity of the attack by suppressing the small values of the perturbation (i.e. $\delta^* \in \mathcal{S}_\Delta = \{\delta_i = 0 \quad \forall \delta_i \leq \sigma\}$). More results of attacked images are in the appendix.

Motasem: Mention and implement the ASR and put some examples on MNIST

7 Conclusion and Future Extension

Tropical geometry can be considered as a combinatorical shadow of algebraic geometry, which is based on tropical algebra on the tropical semiring $\mathbb{R} \cup \{-\infty\}$, where the tropical sum of two numbers is their maximum and the tropical product is their sum. In this paper, we use tropical

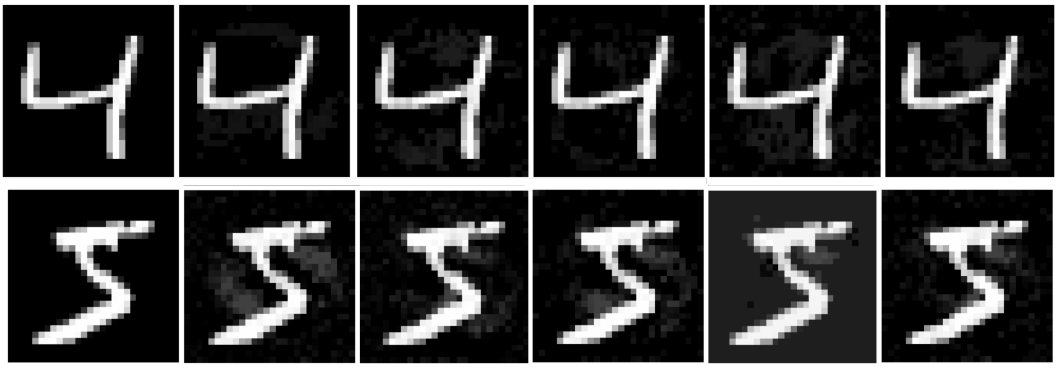

Figure 4: Effect of Tropical Adversarial Attack on MNIST dataset. First row from the left: Clean image, perturbed images classified as $\{2,3,7,8,9\}$ respectively with $\epsilon_1 = 0.1$. Second row from left: Clean image, perturbed images to be classified as $\{0,1,2,3,7\}$ respectively with $\epsilon_1 = 0.2$. Shall we write that the attack is basically trying to draw a small sample from the target class?

geometry to give a full characterization for a super set of the decision boundary of a simple two-layer piece-wise linear network in the form (Affine, ReLU, Affine) with two output classes. This characterization relate the decision boundary with well-studied geometric objects such as tropica hypersurfaces, dual subdivisions, zonotoes and polytopes. Then, a tropical prospective to support the lottery ticket hypothesis have been proposed by investigating the geometry of the decision boundary polytope. Finally, we showed that our ideas can help pruning the neural network while preserving the generalization performance by preserving the decision boundary polytope. Additionally, by deliberately changing the geometry of the decision boundary polytope, we could attack a pre-trained model to classify an input to a targeted class other than the ground truth.

One natural extension for the work in this paper is the derivation of the characterization of the decision boundary with convolutional layers. One possible treatment is to express both affine and convolutional layers as special cases of high-order tensor product.

Motasem: Extension to CNN is a must as well as GCNs

REFERENCES

Marianne Akian, Stphane Gaubert, and Alexander Guterman. Tropical polyhedra are equivalent to mean payoff games. *International Journal of Algebra and Computation*, 22, 12 2009.

Xavier Allamigeon, Pascal Benchimol, Stphane Gaubert, and Michael Joswig. Tropicalizing the simplex algorithm. *SIAM J. Discrete Math. 29:2*, pp. 751–795, 2015.

Diego Ardila, Atilla P. Kiraly, Sujeeth Bharadwaj, Bokyung Choi, Joshua J. Reicher, Lily Peng, Daniel Tse, Mozziyar Etemadi, Wenxing Ye, Greg Corrado, David P. Naidich, and Shravya Shetty. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25:1, 06 2019.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015.

Leonard Berrada, Andrew Zisserman, and M Pawan Kumar. Trusting svm for piecewise linear cnns. *arXiv preprint arXiv:1611.02185*, 2016.

Erwan Brugallé and Kristin Shaw. A bit of tropical geometry. *The American Mathematical Monthly*, 121(7):563–589, 2014.

Peter Butkovič. *Max-linear systems: theory and algorithms*. Springer Science & Business Media, 2010.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644, 2016. URL `http://arxiv.org/abs/1608.04644`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*. OpenReview.net, 2019.

Stéphane Gaubert and Ricardo Katz. Max-plus convex geometry. In *International Conference on Relational Methods in Computer Science*, pp. 192–206. Springer, 2006.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Peter Gritzmann and Bernd Sturmfels. Minkowski addition of polytopes: computational complexity and applications to gröbner bases. *SIAM Journal on Discrete Mathematics*, 6(2):246–269, 1993.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. *CoRR*, abs/1506.02626, 2015. URL `http://arxiv.org/abs/1506.02626`.

Philip Hartman et al. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):707–713, 1959.

Babak Hassibi and David G Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in neural information processing systems*, pp. 164–171, 1993.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012.

Ilia Itenberg, Grigory Mikhalkin, and Eugenii I Shustin. *Tropical algebraic geometry*, volume 35. Springer Science & Business Media, 2009.

Michael Joswig and Georg Loho. Monomial tropical cones for multicriteria optimization. *AIP Conference Proceedings*, 2070:020–025, 02 2019.

Michael Joswig and Benjamin Schröter. The tropical geometry of shortest paths. *arXiv preprint arXiv:1904.01082*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pp. 598–605, 1990.

Yann LeCun, Y Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

Qinghua Liu, Xinyue Shen, and Yuantao Gu. Linearized admm for nonconvex nonsmooth optimization with convergence analysis. *IEEE Access*, 2019.

D. Maclagan and B. Sturmfels. *Introduction to Tropical Geometry*. Graduate Studies in Mathematics. American Mathematical Society, 2015.

Ngoc Mai Tran and Josephine Yu. Product-mix auctions and tropical geometry. *Mathematics of Operations Research*, 05 2015.

D Melzer. On the expressibility of piecewise-linear continuous functions as the difference of two piecewise-linear convex functions. In *Quasidifferential Calculus*, pp. 118–134. Springer, 1986.

Grigory Mikhalkin. Enumerative tropical algebraic geometry in r2. *Journal of the American Mathematical Society*, 18, 01 2004.

Guido Montufar, Razvan Pascanu, Kyunghyun Cho, and Y Bengio. On the number of linear regions of deep neural networks. *NIPS 2014*, 02 2014.

Kristof Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8, 01 2017.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. Compression of neural machine translation models via pruning. *CoRR*, abs/1606.09274, 2016. URL `http://arxiv.org/abs/1606.09274`.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Kerrek Stinson, David F Gleich, and Paul G Constantine. A randomized algorithm for enumerating zonotope vertices. *arXiv preprint arXiv:1602.06620*, 2016.

Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. *arXiv preprint arXiv:1808.01664*, 2018.

Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In *International Conference on Machine Learning, ICML 2018*, 2018.

Jian Zhou, Christopher Y. Park, Chandra L. Theesfeld, Aaron Wong, Yuan Yuan, Claudia Scheckel, John Fak, Julien Funk, Kevin Yao, Yoko Tajima, Alan Packer, Robert Darnell, and Olga G. Troyanskaya. Whole-genome deep-learning analysis identifies contribution of noncoding mutations to autism risk. *Nature Genetics*, 51:1, 05 2019.

A FACTS AND PRELIMINARIES.

Fact 1. *$P\tilde{+}Q = \{p+q, \forall p \in P$ and $q \in Q\}$ is the Minkowski sum between two sets P and Q.*

Fact 2. *Let f be a tropical polynomial and let $a \in \mathbb{N}$. Then*

$$\delta(f^a) = a\delta(f).$$

Let both f and g be tropical polynomials, Then

Fact 3.

$$\delta(f \odot g) = \delta(f)\tilde{+}\delta(g). \tag{6}$$

Fact 4.

$$\delta(f \oplus g) = \textit{ConvexHull}\Big(\mathcal{V}\left(\delta(g)\right) \cup \mathcal{V}\left(\delta(g)\right)\Big). \tag{7}$$

Note that $\mathcal{V}(\delta(f))$ is the set of vertices of the polytope $\delta(f)$

B PROOF OF PROPOSITION 1

Proposition 1. *For a bias-free neural network in the form of $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^2$ where $\mathbf{A}$ and $\mathbf{B}$ have integer weights, and let $R(x) = H_1(\mathbf{x}) \odot Q_2(\mathbf{x}) \oplus H_2(\mathbf{x}) \odot Q_1(\mathbf{x})$ be a tropical polynomial, then:*

- *if the decision boundaries of f is given by the set $\mathcal{B} = \{x \in \mathbb{R}^n : f_1(\mathbf{x}) = f_2(\mathbf{x})\}$, then we have $\mathcal{B} \subseteq \mathcal{T}(R(x))$.*
- *$\delta(R(\mathbf{x})) = \textit{ConvexHull}(\mathcal{Z}_{G_1}, \mathcal{Z}_{G_2})$ where $\mathcal{Z}_{G_1}$ is a zonotope in $\mathbb{R}^n$ with line segments $\mathbb{R}^n$ $\{(\mathbf{B}(1,j)^+ + \mathbf{B}(2,j)^-)[\mathbf{A}(j,:)^+, \mathbf{A}(j,:)^-]_{j=1}^p\}$ with shift $(\mathbf{B}(1,j)^- + \mathbf{B}(2,j)^+)\mathbf{A}^-$ while $\mathcal{Z}_{G_2}$ is a zonotope in $\mathbb{R}^n$ with line segments $\{(\mathbf{B}(1,j)^- + \mathbf{B}(2,j)^+)[\mathbf{A}(j,:)^+, \mathbf{A}(j,:)^-]_{j=1}^p\}$ with shift $(\mathbf{B}(1,j)^+ + \mathbf{B}(2,j)^-)\mathbf{A}^-$.*

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

Proof. For the first part, recall from Theorem1 that both f_1 and f_2 are tropical rationals and hence,

$$f_1(\mathbf{x}) = H_1(\mathbf{x}) - Q_1(\mathbf{x}) \qquad f_2(\mathbf{x}) = H_2(\mathbf{x}) - Q_2(\mathbf{x})$$

Thus:

$$\begin{aligned}\mathcal{B} &= \{x \in \mathbb{R}^n : f_1(\mathbf{x}) = f_2(\mathbf{x})\} = \{x \in \mathbb{R}^n : H_1(\mathbf{x}) - Q_1(\mathbf{x}) = H_2(\mathbf{x}) - Q_2(\mathbf{x})\} \\ &= \{x \in \mathbb{R}^n : H_1(\mathbf{x}) + Q_2(\mathbf{x}) = H_2(\mathbf{x}) + Q_1(\mathbf{x})\} \\ &= \{x \in \mathbb{R}^n : H_1(\mathbf{x}) \odot Q_2(\mathbf{x}) = H_2(\mathbf{x}) \odot Q_1(\mathbf{x})\}\end{aligned}$$

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

$$\mathcal{T}(R(\mathbf{x})) = \mathcal{T}(H_1(\mathbf{x}) \odot Q_2(\mathbf{x})) \cup \mathcal{T}(H_2(\mathbf{x}) \odot Q_1(\mathbf{x})) \cup \mathcal{B}$$

Therefore:

$$\mathcal{B} \subseteq \mathcal{T}(R(x))$$

For the second part of the proof, we first use the decomposition proposed by Zhang et al. (2018); Berrada et al. (2016) to show that for a network $\mathbf{g}(\mathbf{x}) = \mathbf{B}\max(\mathbf{A}\mathbf{x}, \mathbf{0})$, it can be decomposed as tropical rational as follows:

$$\begin{aligned}\mathbf{g}(\mathbf{x}) &= \left(\mathbf{B}^+ - \mathbf{B}^-\right)\Big(\max(\mathbf{A}^+\mathbf{x}, \mathbf{A}^-\mathbf{x}) - \mathbf{A}^-\mathbf{x}\Big) \\ &= \Big[\mathbf{B}^+\max(\mathbf{A}^+\mathbf{x}, \mathbf{A}^-\mathbf{x}) + \mathbf{B}^-\mathbf{A}^-\mathbf{x}\Big] - \Big[\mathbf{B}^-\max(\mathbf{A}^+\mathbf{x}, \mathbf{A}^-\mathbf{x}) + \mathbf{B}^+\mathbf{A}^-\mathbf{x}\Big].\end{aligned}$$

Therefore, we have that

$$\begin{aligned}F_1(\mathbf{x}) + Z_2(\mathbf{x}) &= \Big(\mathbf{B}^+(1,:) + \mathbf{B}^-(2,:)\Big)\max(\mathbf{A}^+\mathbf{x}, \mathbf{A}^-\mathbf{x}) + \Big(\mathbf{B}^-(1,:) + \mathbf{B}^+(2,:)\Big)\mathbf{A}^-\mathbf{x}, \\ F_2(\mathbf{x}) + Z_1(\mathbf{x}) &= \Big(\mathbf{B}^-(1,:) + \mathbf{B}^+(2,:)\Big)\max(\mathbf{A}^+\mathbf{x}, \mathbf{A}^-\mathbf{x}) + \Big(\mathbf{B}^+(1,:) + \mathbf{B}^-(2,:)\Big)\mathbf{A}^-\mathbf{x}.\end{aligned}$$

Therefore note that:

$$\begin{aligned}\delta(R(\mathbf{x})) &= \delta\Bigg(\Big(F_1(\mathbf{x}) \odot Z_2(\mathbf{x})\Big) \oplus \Big(F_2(\mathbf{x}) \odot Z_1(\mathbf{x})\Big)\Bigg) \\ &\overset{(7)}{=} \text{ConvexHull}\Bigg(\delta\Big(F_1(\mathbf{x}) \odot Z_2(\mathbf{x})\Big), \delta\Big(F_2(\mathbf{x}) \odot Z_1(\mathbf{x})\Big)\Bigg) \\ &\overset{(6)}{=} \text{ConvexHull}\Bigg(\delta\Big(F_1(\mathbf{x})\Big) \tilde{+} \delta\Big(Z_2(\mathbf{x})\Big), \delta\Big(F_2(\mathbf{x})\Big) \tilde{+} \delta\Big(Z_1(\mathbf{x})\Big)\Bigg).\end{aligned}$$

Now observe that $F_1(\mathbf{x}) = \sum_{j=1}^{p} \Big(\mathbf{B}^+(1,j) + \mathbf{B}^-(2,j)\Big) \max\Big(\mathbf{A}^+(j,:), \mathbf{A}^-(j,:)\mathbf{x}\Big)$ tropically is given as follows $F_1(\mathbf{x}) = \odot_{j=1}^{p} \Big[\mathbf{x}^{\mathbf{A}^+(j,:)} \oplus \mathbf{x}^{\mathbf{A}^-(j,:)}\Big]^{\mathbf{B}^+(1,j)\odot\mathbf{B}^-(2,j)}$, thus we have that

$$\begin{aligned}
\delta(F_1(\mathbf{x})) &= \Big(\mathbf{B}^+(1,1) + \mathbf{B}^-(2,1)\Big)\delta\Big(\mathbf{x}^{\mathbf{A}^+(1,:)} \oplus \mathbf{x}^{\mathbf{A}^-(1,:)}\Big) \tilde{+} \dots \\
&\quad \tilde{+}\Big(\mathbf{B}^+(1,p) + \mathbf{B}^-(2,p)\Big)\Big(\delta(\mathbf{x}^{\mathbf{A}^+(p,:)} \oplus \mathbf{x}^{\mathbf{A}^-(p,:)})\Big) \\
&= \Big(\mathbf{B}^+(1,1) + \mathbf{B}^-(2,1)\Big)\text{ConvexHull}\Big(\mathbf{A}^+(1,:), \mathbf{A}^-(1,:)\Big) \tilde{+} \dots \\
&\quad \tilde{+}\Big(\mathbf{B}^+(1,p) + \mathbf{B}^-(2,p)\Big)\text{ConvexHull}\Big(\mathbf{A}^+(p,:), \mathbf{A}^-(p,:)\Big).
\end{aligned}$$

The operator $\tilde{+}$ indicates a Minkowski sum between sets. Note that $\text{ConvexHull}\Big(\mathbf{A}^+(i,:), \mathbf{A}^-(i,:)\Big)$ is the convexhull between two points which is a line segment in $\mathbb{Z}^n$ with end points that are $\{\mathbf{A}^+(i,:), \mathbf{A}^+(i,:)\}$ scaled with $\mathbf{B}^+(1,i) + \mathbf{B}^-(2,i)$. Observe that $\delta(F_1(\mathbf{x}))$ is a Minkowski sum of line segments which is is a zonotope. Moreover, note that $\mathbf{Z}_2(\mathbf{x}) = (\mathbf{B}^-(1,:) + \mathbf{B}^+(2,:))\mathbf{A}^-\mathbf{x}$ tropically is given as follows $Z_2(\mathbf{x}) = \odot_{j=1}^{p} \mathbf{x}^{\mathbf{A}^-(j,:)^{(\mathbf{B}^+(1,j)\odot\mathbf{B}^-(2,j))}}$. Thus it is easy to see that $\delta(Z_2(\mathbf{x}))$ is the Minkowski sum of the points $\{(\mathbf{B}^-(1,j) - \mathbf{B}^+(2,j))\mathbf{A}^-(j,:)\}\forall j$ in $\mathbb{R}^n$ (which is a standard sum) resulting in a point. Lastly, it is easy to see that $\delta(F_1(\mathbf{x}))\tilde{+}\delta(Z_2(\mathbf{x}))$ 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 $\delta(F_2(\mathbf{x}))\tilde{+}\delta(Z_1(\mathbf{x}))$. □

C Proof of Proposition 2

Gomaa: I changed the wording of this proposition. please compare it to the original in the paper and tell me if i need to change something.

Proposition 2. *Consider n line segments in $\mathbb{R}^d$ with two arbitrary end points as follows $\{[\mathbf{u}_1^i, \mathbf{u}_2^i]\}_{i=1}^n$. The Zonotope formed by these line segments is equivalent to the zonotope formed be the line segments $\{[\mathbf{u}_1^i - \mathbf{u}_2^i, \mathbf{0}]\}_{i=1}^n$ with a shift of $\sum_{i=1}^n \mathbf{u}_2^i$.*

Proof. Let $\mathbf{U}_j$ be a matrix with $\mathbf{U}_j(:,i) = \mathbf{u}_j^i, i = 1,\dots,n$, $\mathbf{w}$ be a column-vector with $\mathbf{w}(i) = w_i, i = 1,\dots,n$ and $\mathbf{1}$ is a column-vector of ones with length n. Then, the zonotope $\mathcal{Z}$ formed by the given line segments with the two arbitrary end points can be defined as the following Minkowski sum

$$\begin{aligned}
\mathcal{Z} &= \Big\{\sum_{i=1}^{n} w_i\mathbf{u}_1^i + (1-w_i)\mathbf{u}_2^i; w_i \in [0,1], \text{ for all } i\Big\} \\
&= \Big\{\mathbf{U}_1\mathbf{w} - \mathbf{U}_2\mathbf{w} + \mathbf{U}_2\mathbf{1}, \ \ \mathbf{w} \in [0,1]^n\Big\} \\
&= \Big\{\left(\mathbf{U}_1 - \mathbf{U}_2\right)\mathbf{w} + \mathbf{U}_2\mathbf{1}, \ \ \mathbf{w} \in [0,1]^n\Big\} \\
&= \Big\{\left(\mathbf{U}_1 - \mathbf{U}_2\right)\mathbf{w}, \ \ \mathbf{w} \in [0,1]^n\Big\}\tilde{+}\Big\{\mathbf{U}_2\mathbf{1}\Big\}.
\end{aligned}$$

Hence the result follows directly from Definition 8. □

C.1 Optimization of Objective equation 2 of the Binary Classifier

$$\min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} \frac{1}{2}\left\|\tilde{\mathbf{G}}_1-\mathbf{G}_1\right\|_F^2+\left\|\frac{1}{2}\tilde{\mathbf{G}}_2-\mathbf{G}_2\right\|_F^2+\lambda_1\left\|\tilde{\mathbf{G}}_1\right\|_{2,1}+\lambda_2\left\|\tilde{\mathbf{G}}_2\right\|_{2,1}. \tag{8}$$

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

Update $\tilde{\mathbf{A}}$.

$$\tilde{\mathbf{A}} \leftarrow \arg\min_{\tilde{\mathbf{A}}} \frac{1}{2}\left\|\text{Diag}\left(\mathbf{c}_1\right)\tilde{\mathbf{A}}-\mathbf{G}_1\right\|_F^2+\frac{1}{2}\left\|\text{Diag}(\mathbf{c}_2)\tilde{\mathbf{A}}-\mathbf{G}_2\right\|_F^2+\lambda_1\left\|\text{Diag}(\mathbf{c}_1)\tilde{\mathbf{A}}\right\|_{2,1}+\lambda_2\left\|\text{Diag}(\mathbf{c}_2)\tilde{\mathbf{A}}\right\|_{2,1},$$

where $\mathbf{c}_1 = \text{ReLU}(\mathbf{B}(1,:)) + \text{ReLU}(-\mathbf{B}(2,:))$ and $\mathbf{c}_2 = \text{ReLU}(\mathbf{B}(2,:)) + \text{ReLU}(-\mathbf{B}(1,:))$. Note that the problem is separable per-row of $\tilde{\mathbf{A}}$. Therefore, the problem reduces to updating rows of $\tilde{\mathbf{A}}$ independently and the problem exhibits a closed form solution.

$$\begin{aligned}
\tilde{\mathbf{A}}(i,:) &= \arg\min_{\tilde{\mathbf{A}}(i,:)} \frac{1}{2}\left\|\mathbf{c}_1^i\tilde{\mathbf{A}}(i,:)-\mathbf{G}_1(i,:)\right\|_2^2+\frac{1}{2}\left\|\mathbf{c}_2^i\tilde{\mathbf{A}}(i,:)-\mathbf{G}_2(i,:)\right\|_2^2+(\lambda_1\sqrt{\mathbf{c}_1^i}+\lambda_2\sqrt{\mathbf{c}_2^i})\left\|\tilde{\mathbf{A}}(i,:)\right\|_2 \\
&= \arg\min_{\tilde{\mathbf{A}}(i,:)} \frac{1}{2}\left\|\tilde{\mathbf{A}}(i,:)-\frac{\mathbf{c}_1^i\mathbf{G}_1(i,:)+\mathbf{c}_2^i\mathbf{G}_2(i,:)}{\frac{1}{2}(\mathbf{c}_1^i+\mathbf{c}_2^i)}\right\|_2^2+\frac{1}{2}\frac{\lambda_1\sqrt{\mathbf{c}_1^i}+\lambda_2\sqrt{\mathbf{c}_2^i}}{\frac{1}{2}(\mathbf{c}_1^i+\mathbf{c}_2^i)}\left\|\tilde{\mathbf{A}}(i,:)\right\|_2 \\
&= \left(1-\frac{1}{2}\frac{\lambda_1\sqrt{\mathbf{c}_1^i}+\lambda_2\sqrt{\mathbf{c}_2^i}}{\frac{1}{2}(\mathbf{c}_1^i+\mathbf{c}_2^i)}\frac{1}{\left\|\frac{\mathbf{c}_1^i\mathbf{G}_1(i,:)+\mathbf{c}_2^i\mathbf{G}_2(i,:)}{\frac{1}{2}(\mathbf{c}_1^i+\mathbf{c}_2^i)}\right\|_2}\right)\left(\frac{\mathbf{c}_1^i\mathbf{G}_1(i,:)+\mathbf{c}_2^i\mathbf{G}_2(i,:)}{\frac{1}{2}(\mathbf{c}_1^i+\mathbf{c}_2^i)}\right).
\end{aligned}$$

Update $\tilde{\mathbf{B}}^+(1,:)$.

$$\tilde{\mathbf{B}}^+(1,:) = \arg\min_{\tilde{\mathbf{B}}^+(1,:)} \frac{1}{2}\left\|\text{Diag}\left(\tilde{\mathbf{B}}^+(1,:)\right)\tilde{\mathbf{A}}-\mathbf{C}_1\right\|_F^2+\lambda_1\left\|\text{Diag}\left(\tilde{\mathbf{B}}^+(1,:)\right)\tilde{\mathbf{A}}+\mathbf{C}_2\right\|_{2,1}, \quad \text{s.t. } \tilde{\mathbf{B}}^+(1,:)\geq\mathbf{0}.$$

Note that $\mathbf{C}_1 = \mathbf{G}_1 - \text{Diag}\left(\tilde{\mathbf{B}}^-(2,:)\right)\tilde{\mathbf{A}}$ and where $\text{Diag}\left(\tilde{\mathbf{B}}^-(2,:)\right)\tilde{\mathbf{A}}$. Note the problem is separable in the coordinates of $\tilde{\mathbf{B}}^+(1,:)$ and a projected gradient descent can be used to solve the problem in such a way as:

$$\tilde{\mathbf{B}}^+(1,j) = \arg\min_{\tilde{\mathbf{B}}^+(1,j)} \frac{1}{2}\left\|\tilde{\mathbf{B}}^+(1,j)\tilde{\mathbf{A}}(j,:)-\mathbf{C}_1(j,:)\right\|_2^2+\lambda_1\left\|\tilde{\mathbf{B}}^+(1,j)\tilde{\mathbf{A}}(j,:)+\mathbf{C}_2(j,:)\right\|_2, \quad \text{s.t. } \tilde{\mathbf{B}}^+(1,j)\geq 0.$$

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

C.2 ADAPTING OPTIMIZATION EQUATION 2 FOR MULTI-CLASS CLASSIFIER

Note that Proposition 1 describes a superset to the decision boundaries of a binary classifier through the dual subdivision $R(\mathbf{x})$, *i.e.*$\delta(R(\mathbf{x}))$. For a neural network f with k classes, a natural extension for it is to analyze the pair-wise decision boundary of of all k-classes. Thus, let $\mathcal{T}(R_{ij}(\mathbf{x}))$ be the superset to the decision boundary separating classes i and j. Therefore, a natural extension to the geometric loss in equation 1 is to preserve the polytopes among all pairwise follows

$$\min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} \sum_{\forall[i,j]\in S} d\Big(\text{ConvexHull}\left(\mathcal{Z}_{\tilde{\mathbf{G}}_{(i^+,j^-)}}, \mathcal{Z}_{\tilde{\mathbf{G}}_{(j^+,i^-)}}\right), \text{ConvexHull}\left(\mathcal{Z}_{\mathbf{G}_{(i^+,j^-)}}, \mathcal{Z}_{\mathbf{G}_{(j^+,i^-)}}\right)\Big). \quad (9)$$

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 $\mathcal{Z}(\tilde{G}_{(i,j)})$ is the zonotope with the generator matrix $\tilde{\mathbf{G}}_{(i^+,j^-)} = \text{Diag}\left[\text{ReLU}(\tilde{\mathbf{B}}(i,:)) + \text{ReLU}(-\tilde{\mathbf{B}}(j,:))\right]\tilde{\mathbf{A}}$. However, such an approach is generally computationally expensive, particularly, when k is very large. To this end, we make the following observation that $\tilde{\mathbf{G}}_(i^+, j^-)$ can be equivalently written as a Minkowski sum between two sets zonotopes with the generators $\mathbf{G}_{(i^+)} = \text{Diag}\left[\text{ReLU}(\tilde{\mathbf{B}}(i,:)\right]\tilde{\mathbf{A}}$ and $\mathbf{G}_{j^-} = \text{Diag}\left[\text{ReLU}(\tilde{\mathbf{B}}_{j^-})\right]\tilde{\mathbf{A}}$. That is to say, $\mathcal{Z}_{\tilde{\mathbf{G}}_{(i^+,j^-)}} = \mathcal{Z}_{\tilde{\mathbf{G}}_{i^+}} \tilde{+} \mathcal{Z}_{\tilde{\mathbf{G}}_{j^-}}$. This follows from the associative property of Minkowski sums given as follows:

Fact 5. *Let $\{S_i\}_{i=1}^n$ be the set of n line segments. Then we have that*

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

where the sets $P = \tilde{+}_{j\in C_1} S_j$ and $V = \tilde{+}_{j\in C_2} S_j$ where c_1 and c_2 are any complementary partitions of the set $\{S_i\}_{i=1}^n$.

Thus, the objective in 9 can be expanded as follows:

$$\begin{aligned}
&\min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} \sum_{\forall[i,j]\in S} d\Big(\text{ConvexHull}\left(\mathcal{Z}_{\tilde{\mathbf{G}}_{(i^+,j^-)}}, \mathcal{Z}_{\tilde{\mathbf{G}}_{(j^+,i^-)}}\right), \text{ConvexHull}\left(\mathcal{Z}_{\mathbf{G}_{(i^+,j^-)}}, \mathcal{Z}_{\mathbf{G}_{(j^+,i^-)}}\right)\Big) \\
&= \min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} \sum_{\forall[i,j]\in S} d\Big(\text{ConvexHull}\left(\mathcal{Z}_{\tilde{\mathbf{G}}_{i^+}} \tilde{+} \mathcal{Z}_{\tilde{\mathbf{G}}_{j^-}}, \mathcal{Z}_{\tilde{\mathbf{G}}_j^+} \tilde{+} \mathcal{Z}_{\tilde{\mathbf{G}}_{i^-}}\right), \text{ConvexHull}\left(\mathcal{Z}_{\mathbf{G}_{i^+}} \tilde{+} \mathcal{Z}_{\mathbf{G}_{j^-}}, \mathcal{Z}_{\mathbf{G}_j^+} \tilde{+} \mathcal{Z}_{\mathbf{G}_{i^-}}\right)\Big) \\
&\approx \min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} \sum_{\forall[i,j]\in S} \frac{1}{2}\left\|\tilde{\mathbf{G}}_{i^+} - \mathbf{G}_{i^+}\right\|_F^2 + \frac{1}{2}\left\|\tilde{\mathbf{G}}_{i^-} - \mathbf{G}_{i^-}\right\|_F^2 + \frac{1}{2}\left\|\tilde{\mathbf{G}}_{j^+} - \mathbf{G}_{j^+}\right\|_F^2 + \frac{1}{2}\left\|\tilde{\mathbf{G}}_{j^-} - \mathbf{G}_{j^-}\right\|_F^2 \\
&= \min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} \sum_{i=1}^{k} \frac{1}{2}(k-1)\left(\left\|\tilde{\mathbf{G}}_{i^+} - \mathbf{G}_{i^+}\right\|_F^2 + \left\|\tilde{\mathbf{G}}_{i^-} - \mathbf{G}_{i^-}\right\|_F^2 + \left\|\tilde{\mathbf{G}}_{j^+} - \mathbf{G}_{j^+}\right\|_F^2 + \left\|\tilde{\mathbf{G}}_{j^-} - \mathbf{G}_{j^-}\right\|_F^2\right).
\end{aligned}$$

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 following form:

$$\begin{aligned}
\min_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}} \sum_{i=1}^{k} \frac{1}{2}&\left(\left\|\tilde{\mathbf{G}}_{i^+} - \mathbf{G}_{i^+}\right\|_F^2 + \left\|\tilde{\mathbf{G}}_{i^-} - \mathbf{G}_{i^-}\right\|_F^2 + \left\|\tilde{\mathbf{G}}_{j^+} - \mathbf{G}_{j^+}\right\|_F^2 + \left\|\tilde{\mathbf{G}}_{j^-} - \mathbf{G}_{j^-}\right\|_F^2\right) \\
&+ \lambda\left(\left\|\tilde{\mathbf{G}}_{i^+}\right\|_{2,1} + \left\|\tilde{\mathbf{G}}_{i^-}\right\|_{2,1} + \left\|\tilde{\mathbf{G}}_{j^+}\right\|_{2,1} + \left\|\tilde{\mathbf{G}}_{j^-}\right\|_{2,1}\right).
\end{aligned}$$

For completion, we derive the updates for $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$.

Update $\tilde{\mathbf{A}}$.

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

Similar to the binary classification, the problem is seprable in the rows of $\tilde{\mathbf{A}}$. and a closed form solution in terms of the proximal operator of ℓ_2 norm follows naturally for each $\tilde{\mathbf{A}}(i,:)$.

Update $\tilde{\mathbf{B}}^{+}(i,:)$.

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

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

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

A similar argument can be used to update $\tilde{\mathbf{B}}^{-}(i,:)\ \forall i$.

D ALGORITHM FOR SOLVING 5.

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

$$\min_{\eta,\xi_{\mathbf{A}_1}} \quad \mathcal{D}_1(\eta) + \mathcal{D}_2(\xi_{\mathbf{A}_1}) \tag{10}$$

$$\begin{aligned} \text{s.t.} \quad & \mathcal{L}(\mathbf{x}+\eta) \leq 0, \quad \bar{\mathcal{L}}(\mathbf{A}_1) \leq 0, \\ & (\mathbf{x}+\eta) \in [0,1]^n, \quad \|\eta\|_\infty \leq \epsilon_1, \quad \|\xi_{\mathbf{A}_1}\|_{\infty,\infty} \leq \epsilon_2, \quad \mathbf{A}_1\eta - \xi_{\mathbf{A}_1}\mathbf{x} = 0. \end{aligned}$$

However, the choice for $\mathcal{D}_2(\xi_{\mathbf{A}_1})$ should be carefully chosen to cause minimal affect to the decision boundary. For multi-class case, we can follow the observation in **??** to deduce an expression for $\mathcal{D}_2(\xi_{\mathbf{A}_1})$. Recall that we want to make minimal effect on the decision boundary, and the classifier will be fooled. For ease of notation, we refer to $\text{ReLU}(\mathbf{B}(j,:))$ and $\text{ReLU}(-\mathbf{B}(j,:))$ as $\mathbf{B}^+(j,:)$ and $\mathbf{B}^-(j,:)$, respectively. Therefore:

$$\min_{\tilde{\mathbf{A}}} \quad \sum_j \|\tilde{\mathbf{G}}_{j,1} - \mathbf{G}_{j,1}\|_F^2 + \|\tilde{\mathbf{G}}_{j,2} - \mathbf{G}_{j,2}\|_F^2.$$

But notice that $\tilde{\mathbf{G}}_{j,1}$ can be expressed of the attacked model's parameters for class j. Therefore,

$$\begin{aligned} & \sum_j \|\tilde{\mathbf{G}}_{j,1} - \mathbf{G}_{j,1}\|_F^2 + \|\tilde{\mathbf{G}}_{j,2} - \mathbf{G}_{j,2}\|_F^2 = \\ & \sum_j \|\text{Diag}(\mathbf{B}^+(j,:))(\mathbf{A}_1 + \xi_{\mathbf{A}_1}) - \mathbf{G}_{j,1}\|_F^2 + \|\text{Diag}(\mathbf{B}^-(j,:))(\mathbf{A}_1 + \xi_{\mathbf{A}_1}) - \mathbf{G}_{j,2}\|_F^2 = \\ & \sum_j \|\text{Diag}(\mathbf{B}^+(j,:))\xi_{\mathbf{A}_1}\|_F^2 + \|\text{Diag}(\mathbf{B}^-(j,:))\xi_{\mathbf{A}_1}\|_F^2 = \mathcal{D}_2(\xi_{\mathbf{A}_1}). \end{aligned}$$

Therefore, we can write 10 as follows:

$$\begin{aligned} \min_{\eta,\xi_{\mathbf{A}_1}} \quad & \mathcal{D}_1(\eta) + \sum_{j=1}^{d} \|\text{Diag}\Big(\text{ReLU}(\mathbf{B}(j,:)\Big)\xi_{\mathbf{A}_1}\|_F^2 + \|\text{Diag}\Big(\text{ReLU}(-\mathbf{B}(j,:))\Big)\xi_{\mathbf{A}_1}\|_F^2. \\ \text{s.t.} \quad & \mathcal{L}(\mathbf{x}+\eta) \leq 0, \quad \bar{\mathcal{L}}(A) \leq 0, \\ & (\mathbf{x}+\eta) \in [0,1]^n, \quad \|\eta\|_\infty \leq \epsilon_1, \quad \|\xi_{\mathbf{A}_1}\|_{\infty,\infty} \leq \epsilon_2, \quad \mathbf{A}_1\eta - \xi_{\mathbf{A}_1}\mathbf{x}_0 = 0. \end{aligned}$$

To enforce the linear equality constraints $\mathbf{A}_1\eta - \xi_{\mathbf{A}_1}\mathbf{x}_0 = 0$, 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 λ such that $\lambda \to \infty$ Also, we follow Xu et al. (2018) in taking $\mathcal{D}_1(\eta) = \frac{1}{2}\|\eta\|_2^2$.

$$\begin{aligned} \min_{\eta,z,w,\xi_{\mathbf{A}_1}} \quad & \|\eta\|_2^2 + \sum_{j=1}^{d} \|\text{Diag}\Big(\text{ReLU}(\mathbf{B}(j,:)\Big)\xi_{\mathbf{A}_1}\|_F^2 + \|\text{Diag}\Big(\text{ReLU}(-\mathbf{B}(j,:))\Big)\xi_{\mathbf{A}_1}\|_F^2 \\ & + \mathcal{L}(\mathbf{x}+\mathbf{z}) + h_1(\mathbf{w}) + h_2(\xi_{\mathbf{A}_1}) + \lambda\|\mathbf{A}\eta - \xi_{\mathbf{A}_1}\mathbf{x}\|_2^2 + \bar{\mathcal{L}}(\mathbf{A}). \\ \text{s.t.} \quad & \eta = \mathbf{z} \quad \mathbf{z} = \mathbf{w}. \end{aligned}$$

where:

$$h_1(\eta) = \begin{cases} 0, & \text{if } (\mathbf{x}+\eta) \in [0,1]^n \text{and} \|\eta\|_\infty \leq \epsilon_1 \\ \infty, & else \end{cases} \qquad h_2(\xi_{\mathbf{A}_1}) = \begin{cases} 0, & \text{if } \|\xi_{\mathbf{A}_1}\|_{\infty,\infty} \leq \epsilon_2 \\ \infty, & \text{else} \end{cases}.$$

Minimizing the augmented Lagrangian:

$$\begin{aligned} \mathcal{L}(\eta,\mathbf{w},\mathbf{z},\xi_{\mathbf{A}_1},\mathbf{u},\mathbf{v}) = \quad & \|\eta\|_2^2 + \mathcal{L}(\mathbf{x}+\mathbf{z}) + h_1(\mathbf{w}) + \sum_j \|\text{Diag}(\mathbf{B}^+(j,:))\xi_{\mathbf{A}_1}\|_F^2 + \|\text{Diag}(\mathbf{B}^-(j,:))\xi_{\mathbf{A}_1}\|_F^2 + \bar{\mathcal{L}}(\mathbf{A}) + \\ & h_2(\xi_{\mathbf{A}_1}) + \lambda\|\mathbf{A}\eta - \xi_{\mathbf{A}_1}\mathbf{x}\|_2^2 + \mathbf{u}^T(\eta - \mathbf{z}) + \mathbf{v}^T(\mathbf{w} - \mathbf{z}) + \frac{\rho}{2}(\|\eta - \mathbf{z}\|_2^2 + \|\mathbf{w} - \mathbf{z}\|_2^2). \end{aligned}$$

ADMM will split the variables in the above Lagrangian into two blocks of variables and optimize them:

$$\{\eta^{k+1}, \mathbf{w}^{k+1}\} = \arg\min_{\eta,\mathbf{w}} \mathcal{L}(\eta, \mathbf{w}, \mathbf{z}^k, \xi^k_{\mathbf{A}_1}, \mathbf{u}^k, \mathbf{v}^k).$$
$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \mathcal{L}(\eta^{k+1}, \mathbf{w}^{k+1}, \mathbf{z}, \xi^k_{\mathbf{A}_1}, \mathbf{u}^k, \mathbf{v}^k).$$
$$\xi^{k+1}_{\mathbf{A}_1} = \arg\min_{\xi_{\mathbf{A}_1}} \mathcal{L}(\eta^{k+1}, \mathbf{w}^{k+1}, \mathbf{z}^{k+1}, \xi_{\mathbf{A}_1}, \mathbf{u}^k, \mathbf{v}^k).$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \rho(\eta^{k+1} - \mathbf{z}^{k+1}).$$
$$\mathbf{v}^{k+1} = \mathbf{v}^k + \rho(\mathbf{w}^{k+1} - \mathbf{z}^{k+1}).$$

Starting with η:

$$\eta^{k+1} = \arg\min_{\eta} \|\eta\|_2^2 + \lambda\|\mathbf{A}\eta - \xi_{\mathbf{A}_1}\mathbf{x}\|_2^2 + \mathbf{u}^T\eta + \frac{\rho}{2}\|\eta - \mathbf{z}\|_2^2$$
$$= (2\lambda\mathbf{A}^T\mathbf{A} + (2+\rho)\mathbf{I})^{-1}(2\lambda\mathbf{A}^T\xi^k_{\mathbf{A}_1}\mathbf{x} + \rho\mathbf{z}^k - \mathbf{u}^k).$$

Next, for $\mathbf{w}$:

$$\mathbf{w}^{k+1} = \arg\min_{\mathbf{w}} \mathbf{v}^{k^T}\mathbf{w} + h_1(\mathbf{w}) + \frac{\rho}{2}\|\mathbf{w} - \mathbf{z}^k\|_2^2$$
$$= \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{a}\|_2^2 + \frac{1}{\rho}h_1(\mathbf{w}) = prox_{\frac{1}{\rho}h_1(.)}(\mathbf{a}).$$

Where $\mathbf{a} = \mathbf{z}^k - \frac{\mathbf{v}^k}{\rho}$. It is easy to show that the update step for $\mathbf{w}$ can be computed per coordinate in closed form as follows:

$$w_i^{k+1} = \begin{cases} \min(1 - x_i, \epsilon_1) & : a_i > \min(1 - x_i, \epsilon_i) \\ \max(-x_i, -\epsilon_1) & : a_i < \max(-x_i, -\epsilon_1) \\ a_i & : \textit{else} \end{cases}$$

Then, lets move to update $\mathbf{z}$:

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \mathcal{L}(\mathbf{x} + \mathbf{z}) - \mathbf{u}^T\mathbf{z} - \mathbf{v}^T\mathbf{z} + \frac{\rho}{2}(\|\eta - \mathbf{z}\|_2^2 + \|\mathbf{w} - \mathbf{z}\|_2^2).$$

It was shown in Liu et al. (2019) that the linearized ADMM converges for some non-convex problems. Therefore, by linearizing $\mathcal{L}$ and adding Bergman divergence term $(\eta_k/2)\|\mathbf{z} - \mathbf{z}^k\|_2^2$, we can then update z by the following:

$$\mathbf{z}^{k+1} = \frac{1}{\eta_k + 2\rho}(\eta_k\mathbf{z}^k + \rho(\mathbf{b} + \mathbf{c}) - \nabla\mathcal{L}(\mathbf{z}^k + \mathbf{x})),$$

where $\mathbf{b} = \eta^{k+1} + \frac{\mathbf{u}^k}{\rho}$, $\mathbf{c} = \mathbf{w}^{k+1} + \frac{\mathbf{v}^k}{\rho}$. It is worthy to mention that the analysis until this step is inspired by Xu et al. (2018) with modifications to adapt our new formulation. Finally, we need to update $\xi_{\mathbf{A}}$:

$$\xi^{k+1}_{\mathbf{A}} = \arg\min_{\xi_{\mathbf{A}}} h_2(\xi_{\mathbf{A}}) + \|\xi_{\mathbf{A}}\|_F^2 + \lambda\|\xi_{\mathbf{A}}\mathbf{x} - \mathbf{A}\eta\|_2^2 + \bar{\mathcal{L}}(\mathbf{A}),$$

which can be updated using proximal gradient descent, where the proximal operator for $h_2(.)$ is the projection on the l_1 ball. A step of proximal gradient descent can be written as follows:

$$\xi^{k+1}_{\mathbf{A}} = prox_{h_2(.)}(\xi_{\mathbf{A}} - \kappa(\xi^k_{\mathbf{A}}(2\lambda\mathbf{x}\mathbf{x}^T) + 2\lambda\mathbf{A}\eta^k\mathbf{x}^T + 2\sum_j \text{Diag}(\mathbf{B}^2(j,:))\xi_{\mathbf{A}_1} + \nabla\bar{\mathcal{L}}(\mathbf{A}))).$$

Note that κ is step size for gradient descent step, and $\mathbf{B}^2(j,:)$ is squaring each element in the vector $\mathbf{B}(j,:)$.