A Tropical Approach to Neural Networks with Piecewise Linear Activations

Vasileios Charisopoulos* Petros Maragos†

May 22, 2018; Revised January 31, 2019

Abstract

We present a new, unifying approach following some recent developments on the complexity of neural networks with piecewise linear activations. We treat neural network layers with piecewise linear activations as *tropical polynomials*, which generalize polynomials in the so-called $(\max, +)$ or *tropical* algebra, with possibly real-valued exponents. Motivated by the discussion in [23], this approach enables us to refine their upper bounds on linear regions of layers with ReLU or leaky ReLU activations to $\min\left\{2^m, \sum_{j=0}^{n} \binom{m}{j}\right\}$, where n, m are the number of inputs and outputs, respectively. Additionally, we recover their upper bounds on maxout layers. Our work follows a novel path, exclusively under the lens of tropical geometry, which is independent of the improvements reported in [1, 30]. Finally, we present a geometric approach for effective counting of linear regions using random sampling in order to avoid the computational overhead of exact counting approaches.

1 Introduction

In the past decade, multilayered architectures of neural networks have enjoyed an unprecedented growth in popularity, with the introduction of the paradigm of *deep learning* [4, 13, 18]. Deep neural networks consist of the composition of many layers of neurons, which are typically fed through nonlinear activation functions. Two of the most widely used such activations are rectifier linear units (ReLUs) and *maxout* units, which are both piecewise-linear. ReLUs have been shown to outperform traditional choices of activation functions in empirical studies [12, 19], while maxout networks [14] were also quickly adopted after their introduction (see e.g. [34]), as they were empirically validated to integrate well with an averaging technique called *dropout* [31]. The output of a neural network employing either of the above activations is a piecewise-linear function; [23, 27] argued that the number of *linear regions* (i.e. regions of the input space where the output is locally linear) designated by a neural network is a good indicator of its expressive power, and consequently sought to derive upper bounds.

We briefly sketch the outline of this paper:

1. We show that families of piecewise-linear activation functions employed in (deep) neural networks naturally correspond to so-called *max-polynomials* or *tropical polynomials* with real exponents. We obtain bounds on the number of linear regions of

* School of Operations Research and Information Engineering, Cornell University, Ithaca, NY 14850, USA. vc333@cornell.edu

† School of Electrical & Computer Engineering, National Technical University of Athens, Zografou Campus, 15773 Athens, Greece. maragos@cs.ntua.gr

piecewise-linear neural network layers employing a certain duality between tropical polynomials and their corresponding Newton Polytopes.

2. We identify an efficient way for counting linear regions of neural network layers in practice, which adapts a randomized algorithm for counting extreme points of convex polytopes to the Minkowski sum setting.

1.1 Notation and terminology

For the reader's convenience, it is necessary to explain the notation and terminology used in subsequent chapters, as well as a few conventions that we will follow. We denote by $\mathbb{R}$ the line of real numbers and use $\mathbb{R}_{\max}$ for the extended real numbers $\mathbb{R}_{\max} := \mathbb{R} \cup \{-\infty\}$. We denote scalars by regular lowercase font, such as $x \in \mathbb{R}$; vectors by bold lowercase, such as $\boldsymbol{x} \in \mathbb{R}^n$; and matrices by bold uppercase, such as $\boldsymbol{X} \in \mathbb{R}^{m \times n}$. We follow the convention of column vectors, unless explicitly stated otherwise. We denote the set of indices $[n] := \{1, \ldots, n\}$, and write $\|\cdot\|$ for the ℓ_2 norm, $\|\boldsymbol{x}\| := \left(\sum_{i=1}^n |x_i|^2\right)^{1/2}$.

We also follow the lattice-theoretic notation of the mathematical morphology community with regard to the idempotent operators $\max, \min$, in the spirit of [22]. Specifically, given $v_i \in \mathbb{R}$:

$$\bigvee_{i=1}^{n} v_i := \max\left(v_1, \ldots v_n\right), \quad \bigwedge_{i=1}^{n} v_i := \min\left(v_1, \ldots v_n\right) \tag{1}$$

Finally, we write $\mathsf{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ for the multivariate centered normal vector with unit covariance matrix.

1.2 Related Work

The use of tropical geometry to bound the representation power and complexity of learning models has been pioneered by [24] in their seminal paper, which used tropical geometry to assess the effect of graphical model parameters on the solutions of the corresponding inference problems. This line of work was later extended in more general settings, ranging from applications on computational biology [25] to the identifiability of the Restricted Boltzmann Machine [8].

Bounds on the inference regions of neural networks were, to the best of our knowledge, first given in [21], who considered a 2-layer neural network with 0-1 activations. More than two decades later, in [23, 27], the authors rederived essentially the same bounds for layers of neural networks with convex piecewise linear activations, which are more common in contemporary architectures. These bounds were also employed in [28], where the authors are concerned with identifying varying measures of expressivity of deep neural networks. Other authors [1, 30] have since refined these types bounds and proposed practical ways of counting linear regions of neural networks [29, 30]. Concurrently to the publication of the first edition of this paper, [33] established a similar correspondence between inference regions of neural networks and tropical geometry. However, to the best of our knowledge, such a connection had already been encountered in [7], where it was observed that maxout and ReLU activations are essentially represented by their corresponding Newton polytopes. Finally, in [6] the authors design universal approximators of certain classes of data using an argument related to the *Maslov dequantization*, an important transform in tropical algebra.

Table 1: Correspondences between linear and $(\max, +)$ arithmetic

Linear arithmetic	$(\max, +)$ arithmetic
$+$	$\max$
$\times$	$+$
0	$-\infty$
1	0
$x^{-1} = 1/x$	$x^{-1} = -x$

2 Background

2.1 The tropical semiring

The term "tropical semiring" refers to one of the $(\max, +)$ or $(\min, +)$ semirings, which are the algebraic structures defined as $(\mathbb{R}_{\max}, \max, +)$ and $(\mathbb{R}_{\min}, \min, +)$, respectively. In short, ordinary "addition" is replaced by the maximum or minimum, and "multiplication" is replaced by ordinary addition. We use the symbols $\vee, \boxplus$ to refer to matrix/vector addition and multiplication in the case of the $(\max, +)$ semiring; a notable exception is when the operands are scalars, where we may use just max/min and $+$ for simplicity. Table 1 summarizes some important correspondences between linear and $(\max, +)$ algebra. Vector operations generalize in the obvious way: for example, the dot product is as follows:

$$\boldsymbol{c}^\top \boxplus \boldsymbol{d} := \bigvee_{i=1}^{k} c_i + d_i \tag{2}$$

Similar definitions hold for the $(\min, +)$ semiring.

2.2 Elements of Discrete & Tropical Geometry

Subsequent sections make extensive use of results & definitions from discrete geometry, which we briefly present here; we mainly follow [35]. First, we need the notion of a convex hull:

Definition 1. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ be a collection of points in $\mathbb{R}^n$. Their **convex hull** is defined as*

$$\operatorname{conv}\{\boldsymbol{v}_i : i \in [m]\} := \sum_{i=1}^{m} \lambda_i \boldsymbol{v}_i, \quad \lambda_i \geq 0, \ \sum_{i=1}^{m} \lambda_i = 1. \tag{3}$$

A (convex) *polytope* $P \subseteq \mathbb{R}^n$ is a set which can be written as the convex hull of a finite set of points; if these points are known, we say that P admits a $\mathcal{V}$-representation:

$$P = \operatorname{conv}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\} \tag{4}$$

Additionally, we write

$$\operatorname{vert}(P) := \{\boldsymbol{v} \mid \boldsymbol{v} \text{ is a vertex of } P\}. \tag{5}$$

We define the **upper hull** $P^{\max}$ of a polytope P as

$$P^{\max} := \{(\lambda, \boldsymbol{x}) \in P : (t, \boldsymbol{x}) \in P \Rightarrow t \leq \lambda\}. \tag{6}$$

The **lower hull**, $P^{\min}$, is defined in an analogous fashion. We also deal with *Minkowski sums* of convex polytopes, which are defined as follows:

Definition 2. *Let $P, Q \in \mathbb{R}^n$ be convex polytopes. Their* ***Minkowski sum*** *is*

$$P \oplus Q := \{\boldsymbol{p} + \boldsymbol{q} \in \mathbb{R}^n : \boldsymbol{p} \in P,\ \boldsymbol{q} \in Q\} \tag{7}$$

$$= \operatorname{conv}\{\boldsymbol{p} + \boldsymbol{q} \mid \boldsymbol{p} \in \operatorname{vert}(P),\ \boldsymbol{q} \in \operatorname{vert}(Q)\}, \tag{8}$$

where we can write the latter if their $\mathcal{V}$-representations are given. Obviously, the Minkowski sum of two or more convex polytopes is also a convex polytope. Another fundamental object we employ is the **normal cone** to a point of a polytope:

Definition 3. *The* ***normal cone*** *to a polytope P at $\boldsymbol{x}$ is*

$$N_P(\boldsymbol{x}) := \left\{\boldsymbol{c} \in \mathbb{R}^n \;\middle|\; \boldsymbol{c}^\top(\boldsymbol{z} - \boldsymbol{x}) \leq 0,\ \forall \boldsymbol{z} \in P\right\}. \tag{9}$$

Lemma 1 tells us that the normal cones of a polytope cover the whole underlying space:

Lemma 1. *Let $P \subset \mathbb{R}^n$ be a polytope, and denote* $\operatorname{vert}(P)$ *for its collection of vertices. Then* $\bigcup_{\boldsymbol{v} \in \operatorname{vert}(P)} N_P(\boldsymbol{v}) = \mathbb{R}^n$.

Proof. Consider an **arbitrary** vector $\boldsymbol{c} \in \mathbb{R}^n$ and its associated linear functional $\boldsymbol{x} \mapsto \boldsymbol{c}^\top \boldsymbol{x}$, which attains a maximizer on P. By the fundamental theorem of linear programming [32], all linear functionals attain their maxima / minima on one of the vertices of P, which means that $\exists \boldsymbol{v} \in \operatorname{vert}(P)$ such that

$$\boldsymbol{c}^\top \boldsymbol{v} \geq \boldsymbol{c}^\top \boldsymbol{x},\ \forall \boldsymbol{x} \in P \Rightarrow \boldsymbol{c} \in N_P(\boldsymbol{v}).$$

□

Given a cone, its **solid angle** is as follows:

Definition 4. *Consider a convex cone $K \subseteq \mathbb{R}^n$. The* ***solid angle*** *of K, $\omega(K)$, is defined as*

$$\omega(K) := \int_K \exp\left(-\pi \|\boldsymbol{x}\|^2\right) \mathrm{d}\boldsymbol{x}$$

$$= \frac{1}{(2\pi)^{n/2}} \int_K \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2}\right) \mathrm{d}\boldsymbol{x}$$

Note that the latter expression in Definition 4 is equal to $\mathbb{P}(\boldsymbol{g} \in K),\ \boldsymbol{g} \sim \mathsf{N}(\boldsymbol{0}, \boldsymbol{I}_n)$, implying the following:

Corollary 1. *Given a convex polytope P, the solid angles of the normal cones to its vertices form a probability distribution, i.e.* $\sum_{\boldsymbol{v} \in \operatorname{vert}(P)} \omega(N_P(\boldsymbol{v})) = 1$.

Proof. Obviously, $\omega(N_P(\boldsymbol{v})) \geq 0,\ \forall \boldsymbol{v}$. Using Definition 4, we may write

$$\sum_{\boldsymbol{v} \in \operatorname{vert}(P)} \omega(N_P(\boldsymbol{v})) = \sum_{\boldsymbol{v} \in \operatorname{vert}(P)} \mathbb{P}(g \in N_P(\boldsymbol{v}))$$

$$= \mathbb{P}\left(\bigcup_{\boldsymbol{v} \in \operatorname{vert}(P)} \{g \in N_P(\boldsymbol{v})\}\right) = 1,$$

where we made use of the fact that $\omega(N_P(\boldsymbol{v}_i) \cap N_P(\boldsymbol{v}_j)) = 0$ and Lemma 1. □

Finally, we call a set of m points in $\mathbb{R}^n$ to be in **general position** if no $n+1$ of them lie on a common hyperplane.

2.2.1 Tropical Polynomials

We briefly introduce tropical polynomials, on which we heavily rely in our approach. A polynomial in n variables with coefficients from a field $\mathbb{K}$, $p \in \mathbb{K}[x_1, x_2, \dots x_n]$, is defined as

$$p(\boldsymbol{x}) = \sum_i a_i \cdot \boldsymbol{x}^{\boldsymbol{u}^i}, \quad \boldsymbol{u}^i \in \mathbb{N}^n$$

so that the exponent $\boldsymbol{u}^i$ results in $\boldsymbol{x}^{\boldsymbol{u}^i} = x_1^{u_1^i} x_2^{u_2^i} \cdots x_n^{u_n^i}$. If one relaxes the assumption on the exponent $\boldsymbol{u}^i$ being an integer vector, and allowing $\boldsymbol{u}^i \in \mathbb{R}^n$ instead, we then call the resulting expression a **signomial** [10]. Signomials and their positive-coefficient special cases, called *posynomials*, appear in the context of geometric programming. In tropical geometry, polynomials exhibit fundamental differences due to the underlying binary operations. The multi-exponent $\boldsymbol{u}^i$ is replaced by a vector of coefficients $\boldsymbol{c}_i$, and exponentiation becomes the dot product. A tropical polynomial can be viewed as the "tropicalization" of an ordinary polynomial over a non-Archimedean field. For further details, we refer the reader to [20]. However, given that we wish to model activations of neural networks which can have real coefficients, we adopt the corresponding terminology and talk about **tropical signomials** (also referred to as *maxpolynomials* in [5]), where $\boldsymbol{c}_i \in \mathbb{R}^n$ as shown below:

$$h(\boldsymbol{x}) = \bigvee_{i=1}^{m} b_i + \boldsymbol{c}_i^\top \boldsymbol{x}, \quad \boldsymbol{c}_i \in \mathbb{R}^n \tag{10}$$

In the sequel, we will use the terms "polynomials" and "signomials" interchangeably, i.e. tropical polynomials will always allow real exponents. We say a polynomial is of *rank* k if it is the maximum of k terms.

A *hypersurface* associated with a "classical" polynomial p is defined as its zero set, $V(p) = \{\boldsymbol{x} \in \mathbb{R}^n : p(\boldsymbol{x}) = 0\}$. On the contrary, the "zero locus" of a tropical polynomial p is simply the set of points where the maximum is attained by more than one of its terms:

$$V(p) = \{\boldsymbol{x} \in \mathbb{R}^n_{\max} : p(\boldsymbol{x}) \text{ is singular } \} \tag{11}$$

An example of a tropical curve in $\mathbb{R}^2_{\max}$ is depicted in Fig. 1. Every half-ray corresponds to a different pair of maximizing terms: the diagonal corresponds to $\{(x, y) : \ x = y > 0\}$, the vertical half-ray to $\{(x, y) : \ x = 0 > y\}$, and the horizontal to $\{(x, y) : \ y = 0 > x\}$. More elaborate examples can be found in [20]. Informally, one can think of this duality

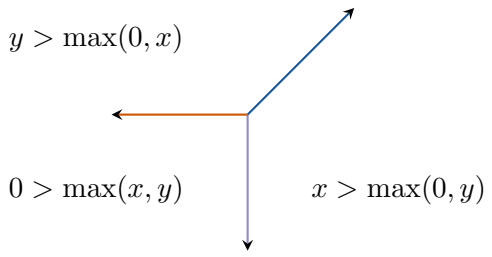

Figure 1: Tropical curve of $p(x, y) = \max(x, y, 0)$

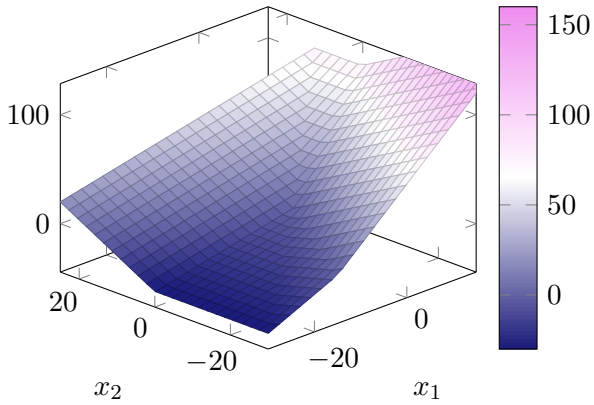

Figure 2: $g(x, y) = \max(x+y, 2x, x+2y) + \max(0, -y, 3x - 2y)$

as a one-to-one correspondence between the vectors $\begin{pmatrix} b_i \\ \boldsymbol{c}_i \end{pmatrix}$ that define the maximizing terms on each open sector, and open sectors of $V(p)$. We will elaborate on this duality in Section 3.

3 Connections to Tropical Geometry

With the definition of a tropical polynomial at hand, we can already draw some connections between popular neural network models and tropical geometry. We are concerned with the following cases:

- ReLU activations: given input $v = \boldsymbol{w}^\top \boldsymbol{x} + b$ with $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^n$, a Rectifier Linear Unit computes
$$\text{ReLU}(\boldsymbol{x}) = \max(0, \boldsymbol{w}^\top \boldsymbol{x} + b) \tag{12}$$
- Maxout units: given $\boldsymbol{W} \in \mathbb{R}^{n \times k}$ and $\boldsymbol{b} \in \mathbb{R}^k, \boldsymbol{x} \in \mathbb{R}^n$:
$$\text{maxout}(\boldsymbol{x}) = \max_{j \in [k]} \left(\boldsymbol{W}_j^\top \boldsymbol{x} + b_j \right), \tag{13}$$
where we denote $\boldsymbol{W}_j$ for the j-th row of $\boldsymbol{W}$.

A variation of ReLU for which this paper's results are also applicable is the Leaky ReLU [19], which replaces the standard activation function with
$$\text{LReLU}_\alpha(v) = \max(v, \alpha v), \quad 0 < \alpha < 1. \tag{14}$$
Notice that maxout units and ReLUs are tropical polynomials of rank k and 2, respectively.

3.1 Newton Polytopes of Tropical Polynomials

Our investigation leverages a fundamental geometric object: the (extended) **Newton Polytope** of a tropical polynomial. Given a polynomial as in Eq. (10), its corresponding Newton Polytope is defined as in Eq. (15).
$$\mathcal{N}(p) := \text{conv}\left\{ \begin{pmatrix} b_i \\ \boldsymbol{c}_i \end{pmatrix} : i \in [m] \right\} \tag{15}$$
Tropical addition and multiplication can also be interpreted as operations on polytopes; [25] elaborate on applications of this interpretation.

Proposition 1. *Let $h_1, \ldots, h_m : \mathbb{R}^n_{\max} \to \mathbb{R}_{\max}$ be a collection of tropical polynomials. It holds that:*
$$V\left(\sum_{i=1}^m h_i \right) = \bigcup_{i=1}^m V(h_i) \tag{16}$$
$$\mathcal{N}\left(\sum_{i=1}^m h_i \right) = \mathcal{N}(h_1) \oplus \cdots \oplus \mathcal{N}(h_m) \tag{17}$$

Proof. The first identity can be found as Proposition 1.16 in [17] for two polynomials and extended via induction. Importantly, its proof does not require the exponents to be integer-valed. For the second identity, consider

$$h_1(\boldsymbol{x}) := \bigvee_{i=1}^{k_1} \alpha_i + \boldsymbol{\beta}_i^\top \boldsymbol{x},\ h_2(\boldsymbol{x}) := \bigvee_{i=1}^{k_2} \gamma_i + \boldsymbol{\delta}_i^\top \boldsymbol{x} \tag{18}$$

$$(h_1 + h_2)(\boldsymbol{x}) = \bigvee_{i\in[k_1], j\in[k_2]} \alpha_i + \gamma_j + (\boldsymbol{\beta}_i + \boldsymbol{\delta}_j)^\top \boldsymbol{x}, \tag{19}$$

where Eq. (19) follows from the identity $(a+b)\vee(c+d) = (a+c)\vee(b+c)\vee(a+d)\vee(b+d)$. However, the terms inside the maximum are precisely sums of individual terms of the two polynomials, so the claim follows. The proof can again be extended via induction. □

We present a few results about faces of polytopes that will be needed in Sec. 3.2. First, recall the definition for a special kind of polytope, called a *zonotope*:

Definition 5. *A **zonotope** $Z \in \mathbb{R}^n$ is a polytope in $\mathbb{R}^n$ which can be written as the Minkowski sum of a set of line segments (edges).*

The *edgotope* is the *smallest zonotope covering P*:

Definition 6. *The edgotope $Z(P)$ of a polytope P is the Minkowski sum of all the edges of P:*

$$Z(P) := \bigoplus_{\boldsymbol{e}\in\text{edges}(P)} \boldsymbol{e} \tag{20}$$

Proposition 2 is a remarkable inequality between faces of polytopes and their edgotopes. Theorem 1 leverages it to upper bound the faces of an arbitrary Minkowski sum. Both appear in [15, Section 2].

Proposition 2. *Let $f_i(P)$ denote the number of i-dimensional faces of a polytope P. Given polytopes $P_1, P_2, \ldots, P_k \in \mathbb{R}^n$, we have:*

$$f_i(P_1 \oplus P_2 \cdots \oplus P_k) \leq f_i\left(Z(P_1) \oplus Z(P_2) \cdots \oplus Z(P_k)\right)$$

Theorem 1. *Let $P_1, P_2, \ldots P_k$ be polytopes in $\mathbb{R}^n$, m denote the number of nonparallel edges of $P_1, P_2, \ldots P_k$, and $i \in \{0, \ldots, n-1\}$. Then*

$$f_i(P_1 \oplus P_2 \cdots \oplus P_k) \leq 2\binom{m}{i} \sum_{j=0}^{n-1-i} \binom{m-1-i}{j} \tag{21}$$

Moreover, for $f_0(P_1 \oplus \cdots \oplus P_k)$, which denotes the number of vertices of the Minkowski sum, the bound of (21) is tight when $2k > n$.

In Eq. (21), the right hand side is the number of i-faces of a zonotope generated by m line segments.

3.2 On the number of linear regions of ReLU/Maxout layers

Pioneering work on DNNs with piecewise-linear activation units focuses on extracting bounds for the number of linear regions they designate [23, 27]. In our treatment, we extract asymptotically similar upper bounds for maxout units and a tight upper bound for layers of rectifier networks, leveraging the corresponding Newton polytopes. In [23], the authors argue that the number of linear regions of a maxout unit is upper bounded by its rank. In fact, that number is in bijection with the number of vertices of the *upper hull* of the corresponding Newton polytope. The following appears in [7] without proof:

Proposition 3. *Let $h(\boldsymbol{x})$, as in* (10)*, describe the activation of a maxout unit. Then there is a bijection between h's linear regions and the vertices lying on the* ***upper hull*** *$\mathcal{N}^{\max}(h)$ of $\mathcal{N}(h)$.*

Proof. Consider

$$\boldsymbol{c}' = \begin{pmatrix} b \\ \boldsymbol{c} \end{pmatrix}, \quad \boldsymbol{x}' = \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}. \tag{22}$$

We can thus rewrite the maxpolynomial's response as a linear program:

$$\begin{aligned} &\text{Maximize } (\boldsymbol{x}')^\top \boldsymbol{c}' \\ &\text{s.t. } \boldsymbol{c}' \in \mathcal{N}(h) \end{aligned} \tag{23}$$

From the fundamental theorem of linear programming [32], we know that optimal solutions to (23) will lie at one of the vertices of $\mathcal{N}(h)$. However, the restriction of the first element of $\boldsymbol{x}'$ hints that some vertices might be redundant. Indeed, pick any vertex $\boldsymbol{c}'_j \notin \mathcal{N}^{\max}(h)$, which implies that $\exists \boldsymbol{c}'_i \in \mathcal{N}^{\max}(h)$, not necessarily a vertex, satisfying:

$$(\boldsymbol{c}'_j)_1 = b_j \leq (\boldsymbol{c}'_i)_1 = b_i, \quad \boldsymbol{c}_j = \boldsymbol{c}_i \tag{24}$$

$$\Rightarrow \boldsymbol{x}'^\top \boldsymbol{c}'_j = b_j + \boldsymbol{x}^\top \boldsymbol{c}_j \overset{(24)}{\leq} b_i + \boldsymbol{x}^\top \boldsymbol{c}_i = \boldsymbol{x}'^\top \boldsymbol{c}'_i. \tag{25}$$

Inequality (25) means that, if we let $\boldsymbol{c}'$ run over all of the Newton polytope, all points not in the upper hull are redundant. Every point in the upper hull that maximizes a linear functional either is a vertex, or can be substituted by a vertex in the upper hull that maximizes the same linear form, from which the claim follows. □

In Fig. 3 we illustrate the canonical projections of the Newton polytopes of the individual summands of $g(x, y)$, which is depicted in Fig. 2. It appears to designate a total of 4 linear regions, as Proposition 3 suggests.

3.2.1 Upper bounds for Relu layers

[23] argue that a linear region in a ReLU layer corresponds to a configuration of active units. Letting $\mathcal{N}_m^n$ denote the number of linear regions of a ReLU layer with n inputs and m outputs, this observation immediately gives $\mathcal{N}_m^n \leq 2^m$. Using the notion of the Newton polytope, we can derive tighter bounds:

Proposition 4. *Let $h_i(\boldsymbol{w}_i, b_i) = \max(0, \boldsymbol{w}_i^\top \boldsymbol{x} + b_i)$, $i = 1, \dots m$ be an arbitrary collection of rectifier units. Then, the Minkowski sum $h_1 \oplus \cdots \oplus h_m$ has at most k nonparallel edges.*

Proof. By definition, $\mathcal{N}(h_i)$ is a zonotope since h_i is a rank-2 polynomial. Zonotopes are line segments, so the Minkowski sum of k such zonotopes has at most k nonparallel edges. □

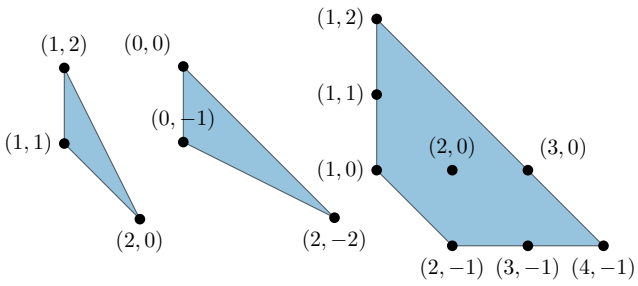

Figure 3: Projected Newton polytopes for the polynomial in Fig. 2. Left and center: polytopes of the summands. Right: polytope of the sum.

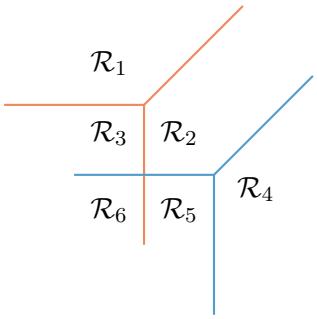

Figure 4: $V(p_1) \cup V(p_2)$ and corresponding linear regions

Notice that Proposition 4 still holds for leaky ReLUs, in which case

$$\mathcal{N}(h_i) = \operatorname{conv}\left\{ \begin{pmatrix} \alpha b \\ \alpha \boldsymbol{w} \end{pmatrix}, \begin{pmatrix} b \\ \boldsymbol{w} \end{pmatrix} \right\}.$$

Assume we are given a collection of ReLUs (i.e. a layer). Each of these ReLUs is a polynomial $p_i : \mathbb{R}^n \to \mathbb{R}$, therefore the total number of linear regions is dual to the hypersurface of that collection of polynomials, which is $V(p_1) \cup \dots V(p_m)$ (see Fig. 4). By Eq. (16), this is the same as $V\left(\sum_{i=1}^m p_i\right)$, which by Eq. (17) is dual to $\mathcal{N}(p_1) \oplus \cdots \oplus \mathcal{N}(p_m)$. The latter is itself a Newton polytope of a polynomial, hence only vertices on its upper hull correspond to linear regions of the collection $\{p_i\}_{i=1}^m$. Proposition 3 specializes that fact to a single polynomial.

Theorem 1 together with Prop. 4 then suggest that:

$$f_i(\mathcal{N}(h_1) \oplus \cdots \oplus \mathcal{N}(h_k)) = 2\binom{k}{i} \sum_{j=0}^{n-i} \binom{k-1-i}{j} \tag{26}$$

Moreover, it is known that zonotopes are centrally symmetric (see e.g. [3]), which implies that their upper and lower hulls have the same number of vertices. Consequently:

Proposition 5. *The number of linear regions of a ReLU/LReLU layer with n inputs and m outputs is upper bounded as*

$$\mathcal{N}_m^n \leq \min\left(2^m, \sum_{j=0}^{n} \binom{m}{j}\right) \tag{27}$$

Moreover, this bound is tight when the zonotopes corresponding to the ReLU activations, as well as the canonical projection to the last n coordinates of its vertices, are in general position.

Proof. By the preceding discussion, it is clear than a ReLU layer with m outputs defines a union of m hypersurfaces, $\bigcup_{i=1}^m V(h_i)$. By Prop. 1, this is equal to $V(\sum_{i=1}^m h_i)$. Therefore, it suffices to upper bound the number of vertices on the upper hull of

$$\mathcal{N}\left(\sum_{i=1}^m h_i\right) = \mathcal{N}(h_1) \oplus \cdots \oplus \mathcal{N}(h_m). \qquad \text{(By (17))}$$

From then, the proof is an application of Theorem 1, Prop. 4 and Prop. 2, in which the inequality is tight since $Z(P_i) = P_i$ for any zonotope P_i. Notice that a zonotope P

being centrally symmetric means that its lower and upper hulls have the same number of vertices, say $n_\ell = n_u = n$. However, its total number of vertices $|\text{vert}(P)| \neq 2n$ in general, since it's possible to have vertices in both the lower and upper hulls at the same time, as Fig. 5 shows. Another example of such a zonotope is the ℓ_1-ball in $d \geq 2$ dimensions.

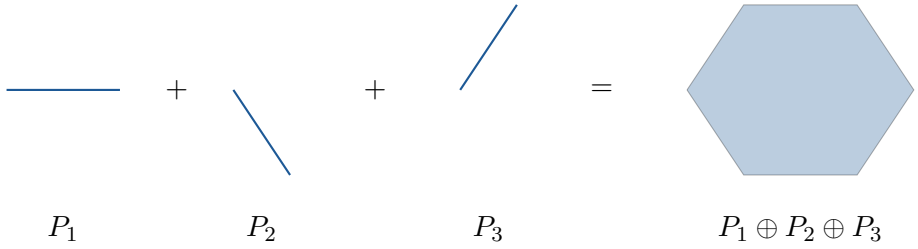

Figure 5: Zonotope with vertices in both envelopes.

Denote $P^{\max}, P^{\min}$ for the upper and lower hulls respectively. A vertex $\boldsymbol{v} \in P^{\max} \cap P^{\min}$ if it is also a vertex for the canonical projection of $P \in \mathbb{R}^n$ to the last $n-1$ coordinates, denoted by P'. Therefore:

$$|\text{vert}(P)| = |\text{vert}(P^{\max})| + |\text{vert}(P^{\min})| - |\text{vert}(P')| \tag{28}$$

$$= 2n - |\text{vert}(P')| \Rightarrow n = \frac{|\text{vert}(P)| + |\text{vert}(P')|}{2}. \tag{29}$$

Theorem 1 applied for P and P' tells us that the right hand side in Eq. (29) is bounded above by

$$\sum_{j=0}^{n} \binom{m-1}{j} + \sum_{j=0}^{n-1} \binom{m-1}{j} = 1 + \sum_{j=1}^{n} \binom{m-1}{j} + \binom{m-1}{j-1} \tag{30}$$

$$= 1 + \sum_{j=1}^{n} \binom{m}{j} = \sum_{j=0}^{n} \binom{m}{j}, \tag{31}$$

where we've made use of the identity $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$. This gives one part of the claimed bound. The other part of the claimed bound follows from the argument in [23], i.e. the number of possible ReLU patterns is bounded above by 2^m. The claim follows. □

The result above assumes a fully-connected neural network layer. It is straightforward to obtain a similar bound for convolutional layers. For a convolutional layer, one may write $\boldsymbol{y} = \boldsymbol{W}\text{vec}(\boldsymbol{X})$, where $\text{vec}(\cdot)$ "reshapes" its argument into a single vector, and deduce the following:

Corollary 2. *The number of linear regions of a single-channel ReLU/LReLU convolutional layer with filter size k and padding p, applied on square images of size d^2, is upper bounded by*

$$\min\left(2^{(d-k+2p+1)^2}, \sum_{j=0}^{d^2} \binom{(d-k+2p+1)^2}{j}\right).$$

Proof. A convolutional layer applies a 2D convolution to the set of input images

$$\{\boldsymbol{X}_i\}_{i=1}^{n}, \ \boldsymbol{X}_i \in \mathbb{R}^{d_w \times d_h},$$

where d_w, d_h are the width and height of the images (assume single-channel). Equivalently, m filters of size $k \times k$ are applied to $\boldsymbol{X}_i$ on (possibly) overlapping regions. We now assume that those regions are separated by a stride of size 1, but our analysis extends in a straightforward way to the case where we have larger strides. In practice, images are also zero-padded by p pixels.

When the conv-layer's activations are ReLUs or leaky ReLUs, our previous arguments apply in a straightforward fashion. The dimension of the output is $d_{\text{out}} = (d_w + 2p - k + 1) \times (d_h + 2p - k + 1)$. The convolution operation is an affine mapping $\boldsymbol{X} \mapsto \boldsymbol{W}\text{vec}(\boldsymbol{X}) + \boldsymbol{b}$, where $\text{vec}(\boldsymbol{X})$ denotes the vectorization of $\boldsymbol{X}$. The weight matrix has at least 1 and at most k^2 elements on every row. By our previous arguments, this will result in a collection of d_{out} tropical signomials. The case of interest is square images with $d_w = d_h = d$, which results in $d_{\text{in}} = d^2,\ d_{\text{out}} = (d - k + 2p + 1)^2$. Then, an application of Prop. 5 gives the result. □

3.2.2 Upper bounds for Maxout layers

By a similar argument, we can recover bounds for maxout units. Let $h(\boldsymbol{x})$ be a maxout activation of rank k, which defined at most k linear regions; by our observation its Newton polytope will have at most k vertices. Therefore, the maximal number of edges it will contain is $\binom{k}{2} = \frac{k(k-1)}{2}$. If we also assume that all the edges of all m polytopes are in general position, we immediately arrive at

Corollary 3. *The linear regions of a maxout layer of n inputs and m outputs, using units of rank k, are upper bounded by*

$$\min\left(k^m, 2 \cdot \sum_{j=0}^{n} \binom{m \cdot \frac{k(k-1)}{2}}{j}\right) \tag{32}$$

The same bound holds for the linear regions of

$$g_+(\boldsymbol{x}) = \sum_{i=1}^{m} w_i \cdot h_i(\boldsymbol{x}), \quad \boldsymbol{w} \geq 0,$$

when $\{h_i\}_{i=1}^m$ are rank-k tropical polynomials, since $\mathcal{N}(g_+)$ is the Minkowski sum of scaled Newton polytopes of h_i. Notice that we cannot refine the binomial sum in Corollary 3, as the resulting Newton polytope is not necessarily centrally symmetric.

4 Counting linear regions in practice

In this section, we provide a computational method to measure the expressive power of a neural network layer, by enumerating its linear regions. In contrast to approaches relying on mixed-integer programming (MIP) such as [29, 30], which usually assume that the input data are bounded in some range, we make no such assumption here.

Suppose we are given m piecewise-linear activation functions $\{h_i\}_{i=1}^m$ such that $h_i = \bigvee_{j=1}^{k_i} \boldsymbol{W}_{i,j}^\top \boldsymbol{x} + b_{i,j}$. Knowing h_i immediately gives us a (not necessarily minimal) $\mathcal{V}$-representation of the corresponding polytope $P_i = \text{conv}\left\{\begin{pmatrix}\boldsymbol{W}_{i,1}\\ b_{i,1}\end{pmatrix}, \ldots, \begin{pmatrix}\boldsymbol{W}_{i,k_i}\\ b_{i,k_i}\end{pmatrix}\right\}$. It thus suffices to compute the number of vertices in the upper hull of the Minkowski sum $P_1 \oplus \cdots \oplus P_m$.

Exact counting for a single layer. It is widely known that the extreme points of Minkowski sums of polytopes are sums of extreme points of the individual polytopes. Additionally, there exist algorithms for enumerating vertices of Minkowski sums of polytopes $P_1, \dots P_m$ when the $\mathcal{V}$-representation of the P_i's is available: this has become widely known as the *reverse search* method [2, 11].

Theorem 3.3 in [11] proves the existence of a polynomial algorithm for enumerating the vertices of $P := P_1 \oplus \cdots \oplus P_m$ in time $\mathcal{O}(\sum_i \delta_i \mathrm{LP}(n, \delta) \, |\mathrm{vert}(P)|)$, where is δ_i the maximum degree of the vertex adjacency graph of P_i and $\mathrm{LP}(n, \delta)$ denotes the time required to solve a linear program (LP) in n variables and δ inequalities. Combined with our estimates, that implies straightforward bounds for exact counting of the linear regions of ReLU/LReLU/Maxout layers. In our case, $\delta = 2m$ for ReLU/LReLU layers and $\delta = \sum_i k_i$ in the case of general convex PWL functions.

Let us briefly address the issue of having a non-minimal $\mathcal{V}$ representation for some of the polytopes P_i. In the case of a ReLU/LReLU network, all polytopes P_i will be edgotopes, which will admit a minimal $\mathcal{V}$ representation unless $\boldsymbol{W}_i = 0$. In the case of a Maxout network, we can eliminate redundant terms by solving k_i LPs (see [26] for more details).

Unfortunately, counting the vertices using reverse search requires solving a prohibitive number of LPs, rendering the approach outlined above impractical. Recent approaches count linear regions using mixed-integer formulations that effectively identify the activation patterns of rectifier networks (e.g. [29]). We attack this problem from a different angle, by considering the "dual" problem of counting vertices of convex polytopes by sampling.

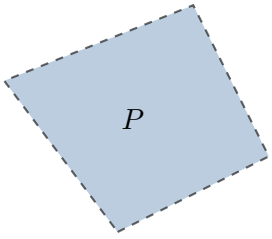

Figure 6: Regular solid angles

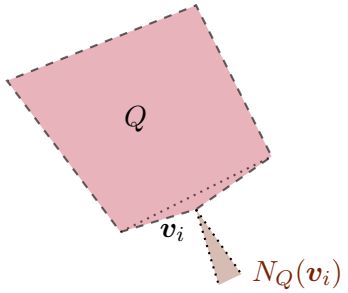

Figure 7: $\omega(N_Q(\boldsymbol{v}_i)) \ll 1$

4.1 A sampling method for polytopes

We briefly present a randomized heuristic for "sampling" the extreme points of the upper hull of a polytope $P = P_1 \oplus \cdots \oplus P_m$. We generate K standard normal vectors, i.e. $\boldsymbol{g}^k \sim_{\text{i.i.d}} \mathsf{N}(\boldsymbol{0}, \boldsymbol{I})$ and compute $\langle \boldsymbol{g}^k, \boldsymbol{v}_i \rangle$, $\forall$ extreme point $\boldsymbol{v}_i$. We record the minimizers/maximizers for each polytope P_j and repeat the trial. This gives us a lower bound for the total number of vertices in the Minkowski sum, since it is well-known that extreme points of a polytope are maximizers of linear functionals over it, and extreme points of Minkowski sums maximize the same linear functional over all individual summands. Let

$$\boldsymbol{V}_i = \begin{pmatrix} \boldsymbol{v}_1^i & \dots & \boldsymbol{v}_{k_i}^i \end{pmatrix}^\top \in \mathbb{R}^{k_i \times n}, \ \forall i \in [m],$$

each row of which is a vertex of P_i. By convention, the first coordinate of each row contains the bias term. Our proposed method, Algorithm 1, leverages the techniques in [9]. We stress that this method and its specialization to upper hulls, Algorithm 2, work

for *general* polytopes, while the mixed-integer-program based methods in the literature are only presented for rectifier networks.

Algorithm 1 Sampling points in the convex hull

Input: polytopes $P_1, \ldots, P_m$ in $\mathcal{V}$-representation
$I_{\text{ext}} := \emptyset$.
for $j = 1, \ldots, K$ **do**
  Sample $\boldsymbol{g}_j \sim \mathsf{N}(\boldsymbol{0}, \boldsymbol{I}_n)$
  Compute $\boldsymbol{z}^i := \boldsymbol{V}_i \boldsymbol{g}_j$, $\forall i \in [m]$.
  Collect $\boldsymbol{z}_{\max} := (\operatorname{argmax} \boldsymbol{z}^1, \ldots, \operatorname{argmax} \boldsymbol{z}^m)$, $\boldsymbol{z}_{\min} := (\operatorname{argmin} \boldsymbol{z}^1, \ldots, \operatorname{argmin} \boldsymbol{z}^m)$.
  $I_{\text{ext}} := I_{\text{ext}} \cup \{\boldsymbol{z}_{\max}, \boldsymbol{z}_{\min}\}$
end for

Algorithm 1 provides a nontrivial lower bound to the number of extreme points of the resulting Minkowski sum with high probability, as Proposition 6 shows.

Proposition 6. *Let* $N = |\operatorname{vert}(P_1 \oplus \cdots \oplus P_m)|$ *and denote*

$$\tilde{N} = \log\left(\frac{1}{\max_k (1 - 2\omega(N_P(\boldsymbol{v}_k)))}\right) \geq \frac{N}{2}.$$

Then, for $K \geq \tilde{N} \log(N/\delta)$ *in Algorithm 1, the algorithm counts all the vertices with probability at least* $1 - \delta$.

Proof. An extreme point of a Minkowski sum is necessarily a sum of extreme points of individual summands. Each time we draw a random sample $\boldsymbol{g}_j$ and record the minimizers of $\{\boldsymbol{V}_i \boldsymbol{g}_j\}_{i \in [m]}$, we are recording one possible extreme point of $P_1 \oplus \cdots \oplus P_m$. Consequently, missing a "configuration" of minimizers across our trials is equivalent to missing an extreme point $\boldsymbol{v}$ of the Minkowski Sum.

Enumerate the individual vertices as $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$. Then,

$$\mathbb{P}(\text{fail}) = \mathbb{P}\left(\bigcup_{k=1}^{N} \text{miss } \boldsymbol{v}_k\right) \overset{\text{(union bound)}}{\leq} \sum_{k=1}^{N} \mathbb{P}(\text{miss } \boldsymbol{v}_k) \tag{33}$$

"Missing" $\boldsymbol{v}_k$ means that it was not a minimizer for any functional $\langle \boldsymbol{g}_j, \cdot \rangle$; equivalently (by independence across samples):

$$\mathbb{P}(\text{miss } \boldsymbol{v}_k) = \mathbb{P}\left(\bigcap_{j=1}^{K} \{\pm \boldsymbol{g}_j \notin N_P(\boldsymbol{v}_k)\}\right)$$

$$= \prod_{j=1}^{K} [1 - \mathbb{P}(\pm \boldsymbol{g}_j \in N_P(\boldsymbol{v}_k))] \leq (1 - 2\omega(N_P(\boldsymbol{v}_k)))^K \tag{34}$$

$$\Rightarrow \mathbb{P}(\text{miss a vertex}) \leq N \max_k (1 - 2\omega(N_P(\boldsymbol{v}_k)))^K \tag{35}$$

If we require the above to be less than δ, we obtain $\delta \geq N \max_k (1 - 2\omega(N_P(\boldsymbol{v}_k)))^K$, which gives the result. □

Our guarantee heavily depends on the cones $N_P(\boldsymbol{v}_k)$. If there are vertices that only slightly "extend" out of the polytope, our required sample size will be a large multiple of N. Figures 6 and 7 illustrate (non-zonotopal) examples in $\mathbb{R}^2$; Q has a vertex where

the solid angle of the normal cone is close to 0, in contrast to P which is more "regular". If one can "get away" with computing a lower bound on the actual number of linear regions, a similar guarantee is available; instead of the exact number of linear regions we may consider a threshold $\frac{1}{2} > \eta > 0$ and the set $\mathcal{V}_\eta := \{\boldsymbol{v}_i \in \mathrm{vert}(P) \mid \omega(N_P(\boldsymbol{v}_i)) \geq \eta\}$; informally, $\mathcal{V}_\eta$ is the set of vertices whose normal cones' angles are not "too small'.

Corollary 4. *Let η be such that $|\mathcal{V}_\eta| \geq cN$, for some $c \in [0,1]$. Then Algorithm 1 counts at least cN vertices with probability at least $1-\delta$, for $K \geq \frac{1}{2\eta}\log\frac{N}{\delta}$.*

Proof. We follow the proof of Prop. 6, making use of the inequality $1-x \leq e^{-x}$ to simplify the expression:

$$\begin{aligned}
\mathbb{P}\left(\text{miss from } \mathcal{V}_\eta\right) &= \mathbb{P}\left(\bigcup_{\boldsymbol{v}\in\mathcal{V}_\eta} \{\text{miss } \boldsymbol{v}\}\right) \\
&\leq \sum_{\boldsymbol{v}\in\mathcal{V}_\eta} \mathbb{P}\left(\text{miss } \boldsymbol{v}\right) \leq |\mathcal{V}_\eta| \max_{\boldsymbol{v}\in\mathcal{V}_\eta} \left(1-2\omega(N_P(\boldsymbol{v}))\right)^K \\
&\leq N\exp\left(-K\min_{\boldsymbol{v}\in\mathcal{V}_\eta} 2\omega(N_P(\boldsymbol{v}))\right) \leq N\exp(-2K\eta)
\end{aligned}$$

Setting $N\exp(-2K\eta) \leq \delta$ gives us $K \geq \frac{1}{2\eta}\log\frac{N}{\delta}$. □

Unfortunately, the correct parameter η in Corollary 4 is not known a priori. Bounding the (expected) number of vertices of the Minkowski sum when the generating distribution of vertices of the summands is known (e.g. using some empirical initialization rule, such as in [16]), is deferred to future work.

What about the upper hull? The analysis of Algorithm 1 assumed that we are counting *all* vertices of P; however, in our setting, we are only interested in the upper hull. It is known that $\boldsymbol{v} \in P^{\min}$ implies that $c \in N_P(\boldsymbol{v}) \Rightarrow c_1 \leq 0$, so it suffices to consider only samples $\boldsymbol{g}_j$ with $(\boldsymbol{g}_j)_1 > 0$. We thus obtain a similar guarantee, stated in Corollary 5.

Algorithm 2 Sampling points in the upper hull

Input: polytopes $P_1, \ldots, P_m$ in $\mathcal{V}$-representation
$I_{\text{ext}} := \emptyset$.
for $j = 1, \ldots, K$ **do**
  Sample $\boldsymbol{g}_j \sim \mathsf{N}(\boldsymbol{0}, \boldsymbol{I}_n)$
  if $(\boldsymbol{g}_j)_1 < 0$ **then**
    $\boldsymbol{g}_j := -\boldsymbol{g}_j$
  end if
  Compute $\boldsymbol{z}^i := \boldsymbol{V}_i\boldsymbol{g}_j$, $\forall i \in [m]$.
  $\boldsymbol{z}_{\max} := (\operatorname{argmax}\boldsymbol{z}^1, \ldots, \operatorname{argmax}\boldsymbol{z}^m)$
  $I_{\text{ext}} := I_{\text{ext}} \cup \{\boldsymbol{z}_{\max}\}$
end for

Corollary 5. *Let N denote the number of vertices on the upper hull of $P := P_1 \oplus \cdots \oplus P_m$, $\{\boldsymbol{v}_k\}_k$ be an enumeration of the vertices in $P^{\max}$, and $N'_P(\boldsymbol{v}) := \{\boldsymbol{c} \in N_P(\boldsymbol{v}) \mid c_1 \geq 0\}$. Set $\tilde{N} = \log\left(\frac{1}{\max_k(1-\omega(N'_P(\boldsymbol{v}_k)))}\right)$. Then, for $K \geq \tilde{N}\log(N/\delta)$, Algorithm 2 counts all the vertices in $P^{\max}$ with probability at least $1-\delta$.*

Proof. We follow the proof of Proposition 6, with the slight alteration that the number of extreme points calculated at each step is just one. Enumerate the individual vertices as $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$. Again, the union bound gives us

$$\mathbb{P}\left(\text{fail}\right) \leq \sum_{k=1}^{N} \mathbb{P}\left(\text{miss } \boldsymbol{v}_k\right) \tag{36}$$

Now, consider a functional $\langle \boldsymbol{g}_j, \cdot \rangle$. Let us define

$$\boldsymbol{q}_j := \begin{cases} \boldsymbol{g}_j, & \text{if } (\boldsymbol{g}_j)_1 < 0 \\ -\boldsymbol{g}_j, & \text{otherwise.} \end{cases} \tag{37}$$

Notice that setting $\boldsymbol{q}_j := -\boldsymbol{g}_j$ does not change the underlying distribution $\mathsf{N}(\boldsymbol{0}, \boldsymbol{I}_n)$, since centered normal random variables are symmetric. Again, "missing" $\boldsymbol{v}_k$ and its interpretation in terms of the truncated normal cones N'_P means

$$\mathbb{P}\left(\text{miss } \boldsymbol{v}_k\right) = \mathbb{P}\left(\bigcap_{j=1}^{K} \{\boldsymbol{g}_j \notin N'_P(\boldsymbol{v}_k)\}\right)$$

$$= \prod_{j=1}^{K} \left[1 - \mathbb{P}\left(\boldsymbol{g}_j \in N'_P(\boldsymbol{v}_k)\right)\right] \leq \left(1 - \omega(N'_P(\boldsymbol{v}_k))\right)^K \tag{38}$$

$$\Rightarrow \mathbb{P}\left(\text{fail}\right) \leq N \max_k \left(1 - \omega(N'_P(\boldsymbol{v}_k))\right)^K \tag{39}$$

Notice that since we are only considering vertices in the upper hull of P, it must hold that $N'_P(\boldsymbol{v}_k) > 0$, so the bound above is indeed not vacuous. Requiring $\mathbb{P}\left(\text{fail}\right) < \delta$ gives us the claimed lower bound for K. □

5 Conclusion

We presented a unifying approach to bounding the number of linear regions of neural networks using maxout/ReLU activations by treating the latter as polynomials in tropical algebra. We showed that linear regions are in bijection with vertices of the Newton polytopes of corresponding tropical polynomials, which we leveraged to recover upper bounds. Finally, we introduced a sampling algorithm for approximately counting the number of linear regions of a single piecewise-linear layer. Our algorithm does not impose any assumptions over the range of the input, avoids the computational overhead of LP/MIP-based approaches, and extends beyond rectifier networks. We hope that this contribution serves as a further step towards underlining the importance of algebraic geometric methods in understanding the complexity of learning models such as deep neural networks.

References

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

[2] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3):21–46, 1996.

[3] Matthias Beck and Sinai Robins. *Computing the continuous discretely*. Undergraduate Texts in Mathematics. Springer, 2015.

[4] Yoshua Bengio et al. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

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

[6] Giuseppe C Calafiore, Stephane Gaubert, and Corrado Possieri. Log-sum-exp neural networks and posynomial models for convex and log-log-convex data. *arXiv preprint arXiv:1806.07850*, 2018.

[7] Vasileios Charisopoulos and Petros Maragos. Morphological perceptrons: Geometry and training algorithms. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, volume 10225 of *Lecture Notes in Computer Science*, pages 3–15. Springer, Cham, 2017.

[8] María Angélica Cueto, Jason Morton, and Bernd Sturmfels. Geometry of the restricted boltzmann machine. *Algebraic Methods in Statistics and Probability II*, 516:135–153, 2010.

[9] Anil Damle and Yuekai Sun. A geometric approach to archetypal analysis and nonnegative matrix factorization. *Technometrics*, 59(3):361–370, 2017.

[10] R. J. Duffin, E. L. Peterson, and C. Zener. *Geometric Programming*. John Wiley, New York, 1967.

[11] Komei Fukuda. From the zonotope construction to the minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272, 2004.

[12] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *AISTATS (14)*, pages 315–323, 2011.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[14] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1319–1327, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

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

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[17] Kerstin Hept. *Projections of Tropical Varieties and an Application to Small Tropical Bases*. PhD thesis, 2009.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS (25)*, pages 1097–1105, 2012.

[19] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30, 2013.

[20] Diane Maclagan and Bernd Sturmfels. *Introduction to Tropical Geometry*, volume 161 of *Graduate Studies in Mathematics*. American Mathematical Soc., 2015.

[21] John Makhoul, Richard Schwartz, and Amro El-Jaroudi. Classification capabilities of two-layer neural nets. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 635–638. IEEE, 1989.

[22] Petros Maragos. Dynamical systems on weighted lattices: general theory. *Mathematics of Control, Signals, and Systems*, 29(4):21, Dec 2017.

[23] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *NIPS (27)*, pages 2924–2932, 2014.

[24] Lior Pachter and Bernd Sturmfels. Tropical geometry of statistical models. *Proceedings of the National Academy of Sciences*, 101(46):16132–16137, 2004.

[25] Lior Pachter and Bernd Sturmfels. *Algebraic Statistics for Computational Biology*. Cambridge University Press, 2005.

[26] PM Pardalos, Y Li, and WW Hager. Linear programming approaches to the convex hull problem in rm. *Computers & Mathematics with Applications*, 29(7):23–29, 1995.

[27] Razvan Pascanu, Guido Montufar, and Yoshua Bengio. On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv preprint arXiv:1312.6098*, 2013.

[28] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl Dickstein. On the expressive power of deep neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2847–2854. JMLR. org, PMLR, 2017.

[29] Thiago Serra and Srikumar Ramalingam. Empirical bounds on linear regions of deep rectifier networks. *arXiv preprint abs:1810.03370*, 2018.

[30] Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. Bounding and counting linear regions of deep neural networks. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4558–4566, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[32] Robert J Vanderbei. *Linear Programming*, volume 196 of *International Series in Operations Research & Management Science*. Springer, 4th edition, 2014.

[33] Liwen Zhang, Gregory Naitzat, and Lek-Heng Lim. Tropical geometry of deep neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5824–5832. PMLR, 2018.

[34] Xiaohui Zhang, Jan Trmal, Daniel Povey, and Sanjeev Khudanpur. Improving deep neural network acoustic models using generalized maxout networks. In *ICASSP*, pages 215–219. IEEE, 2014.

[35] Günter M Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Studies in Mathematics*. Springer Science & Business Media, 1995.