Tropical Geometry and Machine Learning

Petros Maragos, *Fellow, IEEE*, Vasileios Charisopoulos, and Emmanouil Theodosis, *Member, IEEE*

***Abstract*—Tropical geometry is a relatively recent field in mathematics and computer science combining elements of algebraic geometry and polyhedral geometry. The scalar arithmetic of its analytic part pre-existed in the form of max-plus and min-plus semiring arithmetic used in finite automata, nonlinear image processing, convex analysis, nonlinear control, optimization, and idempotent mathematics. Tropical geometry recently emerged in the analysis and extension of several classes of problems and systems in both classical machine learning and deep learning. Three such areas include (1) deep neural networks with piecewise-linear (PWL) activation functions, (2) probabilistic graphical models, and (3) nonlinear regression with PWL functions. In this article, we first summarize introductory ideas and objects of tropical geometry, providing a theoretical framework for both the max-plus algebra that underlies tropical geometry as well as its extensions to general max algebras. This unifies scalar and vector/signal operations over a class of nonlinear spaces, called weighted lattices, and allows us to provide optimal solutions for algebraic equations used in tropical geometry and generalize tropical geometrical objects. Then, we survey the state-of-the-art and recent progress in the aforementioned areas: first, we illustrate a purely geometric approach for studying the representation power of neural networks with PWL activations. Then, we review the tropical geometric analysis of parametric statistical models, such as HMMs; later, we focus on the Viterbi algorithm and related methods for Weighted Finite State Transducers and provide compact and elegant representations via their formal tropical modeling. Finally, we provide optimal solutions and an efficient algorithm for the convex regression problem, using concepts and tools from tropical geometry and max-plus algebra.**

Throughout this article we also outline problems and future directions in machine learning that can benefit from the tropical-geometric point of view.

***Index Terms*—tropical geometry, max-plus algebra, lattices, neural networks, regression, graphs**

I. Introduction

TROPICAL geometry is a relatively recent field in mathematics and computer science that combines elements from algebraic geometry and polyhedral geometry. The scalar arithmetic of its analytic part pre-existed in the form of max-plus and min-plus semiring arithmetic used in finite automata, nonlinear image processing, convex analysis, nonlinear control, optimization, and idempotent mathematics. In max-plus arithmetic the real number addition and multiplication are replaced by the max and sum operations, respectively. The name 'tropical semiring' initially referred to the min-plus semiring and was used in finite automata [55], [95], speech recognition using graphical models [78], and tropical geometry [65], [76]. However, nowadays tropical semiring may refer to both the max-plus and its dual min-plus arithmetic, whose combinations with corresponding nonlinear matrix algebra and nonlinear signal convolutions have been used in operations research and scheduling [25]; discrete event systems, max-plus control and optimization [1], [2], [6], [15], [22], [35], [37], [46], [74], [105]; convex analysis [62], [81], [90]; morphological image analysis [47], [69], [75], [91], [92]; nonlinear difference equations for distance transforms [11], [67]; nonlinear PDEs of the Hamilton-Jacobi type for vision scale-spaces [14], [48]; speech recognition and natural language processing [54], [78]; neural networks [18], [19], [32], [38], [79], [85], [89], [98], [109], [110]; idempotent mathematics (nonlinear functional analysis) [60], [61].

The goal of this paper is threefold: (i) provide a brief background from tropical geometry and its underlying max-plus algebra, (ii) summarize its applications in three areas of machine learning (neural networks, graphical models, and nonlinear regression), and (iii) provide recent progress and some extensions using a generalized max algebra. Parts (i) and (ii) provide tutorial information and survey state-of-the-art results. Some recent progress from the authors is included in parts (ii) and (iii).

We begin in Section II with elementary ideas and objects of tropical geometry. Section III provides the required theoretical background on max-plus algebra, its underlying nonlinear vector spaces called weighted lattices, and monotone operators in the form of lattice duality pairs called adjunctions (a.k.a. residuation pairs). This section also provides some tools from a generalized max-$\star$ algebra to extend tropical geometrical objects. Further, in Section IV we show that adjunction pairs lead to optimal solutions of max-plus and general max-$\star$ equations, as nonlinear projections on weighted lattices. Then, the concepts and tools of the previous sections are applied to analyzing and/or providing solutions for problems in the following three broad areas of machine learning.

1) *Neural networks with piecewise-linear (PWL) activations* (Section V): Tropical geometry recently emerged in the study of deep neural networks (DNNs) and variations of the perceptron operating in the max-plus semiring. Standard activation functions employed in DNNs, including the ReLU activation and its "leaky" variants, induce neural network layers which are PWL convex functions of their inputs and create a partition of space well-described by concepts from tropical geometry. Following [18], [19], we illustrate a purely geometric approach for studying the representation power of DNNs – measured via the concept of a network's "linear regions" – under the lens of tropical geometry.

2) *Probabilistic graphical models and algorithms* (Section VI): As we review in Sec. VI-A, a novel application of tropical geometry is its usage in [82] for analyzing parametric statistical models, including hidden Markov models

P. Maragos is with the National Technical University of Athens, Greece (email: maragos@cs.ntua.gr.)

V. Charisopoulos is with Cornell University, USA (email: vc333@cornell.edu.)

E. Theodosis is with Harvard University, USA (email: etheodosis@g.harvard.edu.)

and restricted Boltzmann machines. Further, among the max-sum and max-product algorithms used in graphical models, a prime representative is the Viterbi algorithm. This can also be viewed in the general setting of Weighted Finite State Transducers [54], [78] which have found extensive use in speech recognition and other decoding schemes. Practical reasons led researchers to adopt a tropical version of these algorithms in order to resolve numerical issues that arose from using sum-product algebras. However, as we explain in Sec. VI-B, tropicalization is not restricted merely as a numerical tool; further tropical modeling of the algorithms as in [101], [102] leads to a compact and elegant representation, while highlighting geometric properties.

3) *Piecewise-linear (PWL) regression* (Section VII): Fitting PWL functions to data is a fundamental regression problem in multidimensional signal modeling and machine learning, since approximations with PWL functions have proven analytically and computationally very useful in many fields of science and engineering. We focus on functions that admit a convex representation as the maximum of affine functions (e.g. lines, planes), represented with max-plus tropical polynomials. This allows us to use concepts and tools from tropical geometry and max-plus algebra to optimally approximate the shape of curves and surfaces by fitting tropical polynomials to data, possibly in the presence of noise; this yields polygonal or polyhedral shape approximations. For this convex PWL regression problem we provide optimal solutions w.r.t. ℓ_p error norms, derived using monotone operator adjunctions that are projections on weighted lattices, and an efficient algorithm based on preliminary work in [72].

Finally in Section VIII, extending preliminary work in [71], we generalize tropical geometry using the max-$\star$ algebra and weighted lattices framework of [70], as summarized in Sec III-B, with an arbitrary binary operation $\star$ that distributes over max, and apply it to optimal convex piecewise-linear regression for fitting max-$\star$ tropical curves and surfaces to arbitrary data.

II. Elements of Tropical Geometry

After some notation and definitions from tropical and related semirings, we first present some simple examples of tropical[1] curves and surfaces which result from tropicalizing the polynomials that analytically describe their Euclidean counterparts. Then, we explain this tropicalization as a dequantization of real algebraic geometry. Finally, Newton polytopes and tropical halfspaces are defined with examples.

Notation: For maximum (or supremum) and minimum (or infimum) operations we use the well-established lattice-theoretic[2] symbols of $\vee$ and $\wedge$. We use roman letters for functions, signals and their arguments, and greek letters mainly for operators. Also, boldface roman letters for vectors (lowcase) and matrices (capital). If $\boldsymbol{M} = [m_{ij}]$ is a matrix, its (i,j)-th element is denoted as m_{ij} or $[\boldsymbol{M}]_{ij}$. Similarly, $\boldsymbol{x} = [x_i]$ denotes a column vector, whose i-th element is denoted as $[\boldsymbol{x}]_i$ or simply x_i. We also use the set notation $[n] := \{1, \dots, n\}$.

A. Tropical Semirings

Compared with the classical real number ring $(\mathbb{R}, +, \times)$, the *max-plus semiring* $(\mathbb{R}_{\max}, \vee, +)$ consists of the set $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$ equipped with an idempotent 'addition' which is the maximum operation and a generalized 'multiplication' which is the extended real addition. Similarly, we consider the dual *min-plus semiring* $(\mathbb{R}_{\min}, \wedge, +)$ where $\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$. Both tropical semirings are special cases of *dioids* [37]. From a different viewpoint that we follow in this paper, if we combine both the maximum and minimum operations we obtain the complete lattice $(\overline{\mathbb{R}}, \vee, \wedge)$ of extended real numbers $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. Further, as done more generally in Sec. III-B, we can combine the max-plus and min-plus scalar arithmetic into an algebraic structure called complete lattice-ordered double monoid (*clodum*) which consists of the extended reals $\overline{\mathbb{R}}$ equipped with the maximum $(\vee)$, minimum $(\wedge)$, addition $(+)$ and dual addition $(+')$ operations. The operations $+$ and $+'$ are respectively the 'lower addition' and 'upper addition' used in convex analysis [81]. They are identical for finite reals and differ only when combining $-\infty$ with $+\infty$; in all cases, they are commutative:

$$\begin{array}{l} a + b = a +' b, \;\; \forall a \in \overline{\mathbb{R}}, \; \forall b \in \mathbb{R} \\ a + (-\infty) = -\infty, \; a +' (+\infty) = +\infty, \;\; \forall a \in \overline{\mathbb{R}} \end{array} \tag{1}$$

In idempotent mathematics [61], convex optimization [13], and the theory of dioids [37], the following *Log-Sum-Exp* approximation is often used for the max and min operations:

$$\begin{array}{rcl} a \curlyvee_\theta b & := & \theta \cdot \log(e^{a/\theta} + e^{b/\theta}) = \phi_\theta^{-1}[\phi_\theta(a) + \phi_\theta(b)] \\ a \curlywedge_\theta b & := & (-\theta) \log(e^{-a/\theta} + e^{-b/\theta}) \end{array} \tag{2}$$

where $\phi_\theta(a) := \exp(a/\theta)$, and $\theta > 0$ is usually called a 'temperature' parameter. In the limit as $\theta \to 0$ we obtain the max and min operations:

$$\begin{array}{l} \lim_{\theta \downarrow 0} \;\; a \curlyvee_\theta b = \max(a, b) \\ \lim_{\theta \downarrow 0} \;\; a \curlywedge_\theta b = \min(a, b) \end{array} \tag{3}$$

This approximation and limit is the *Maslov Dequantization* [73] of real numbers, and generates a whole family of semirings $S_\theta = (\mathbb{R}_{\max}, \curlyvee_\theta, +)$, $\theta > 0$, whose operations are the generalized 'addition' $\curlyvee_\theta$ and 'multiplication' $+$. This makes S_θ isomorphic to the semiring of nonnegative real numbers $\mathbb{R}_{\geq 0}$ equipped with standard addition and multiplication. This isomorphism is enabled via the logarithmic mapping $\phi_\theta^{-1} = \theta \log(a) : \mathbb{R}_{\geq 0} \to S_\theta$. In the limit $\theta \downarrow 0$ we get S_0 which is the max-plus semiring.

[1] The adjective 'tropical' was coined by French mathematicians, including Dominique Perrin and Jean-Eric Pin, to honor their Brazilian colleague Imre Simon who was one of the pioneers of min-plus algebra as applied to automata. However, we give it an alternative and substantial meaning in connection with its Greek origin word 'τροπικός', which comes from the Greek word 'τροπή' which means "turn" or "changing the way/direction", to literally express the fact that tropical curves and surfaces bend and turn.

[2] We do *not* use the notation $(\oplus, \otimes)$ for $(\max, +)$ or $(\min, +)$ which is frequently used in max-plus algebra, because in functional analysis and image procesing i) the symbol $\oplus$ is extensively used for Minkowski set addition and max-plus signal convolution, and ii) $\otimes$ is unnecessarily confusing compared to the classic symbol $+$ of addition.

B. Examples of Tropical Polynomial Curves and Surfaces

Tropical Polynomial Curves: Consider the analytic expressions for a Euclidean line and parabola:

$$p_1(x) = ax + b,\ p_2(x) = ax^2 + bx + c \tag{4}$$

'Tropicalization', i.e. replacing sum with max and multiplication with addition, yields the corresponding max-plus tropical polynomials:

$$\begin{array}{l} p_1^{\max}(x) = \max(a + x, b) \\ p_2^{\max}(x) = \max(a + 2x, b + x, c) \end{array} \tag{5}$$

The equations for the min-plus case are identical as in (5) by replacing max with min. The graphs of all the above can be seen in Fig. 1.

Tropical Polynomial Surfaces: Consider the equations of the following tropical planes represented as 2D max-plus and min-plus polynomial of degree 1:

$$f(x, y) = \max(x, 2{+}y, 7),\ g(x, y) = \min(5{+}x, 7{+}y, 9) \tag{6}$$

whose graphs can be seen as surfaces in Fig. 2(a),(b).

Next, to the general Euclidean conic polynomial

$$p_{\text{e-conic}}(x, y) = ax^2 + bxy + cy^2 + dx + ey + f \tag{7}$$

there corresponds the following two-variable max-plus tropical polynomial of degree 2:

$$p_{\text{t-conic}}(x, y) = \max(a{+}2x, b{+}x{+}y, c{+}2y, d{+}x, e{+}y, f) \tag{8}$$

Its min-plus version is shown in Fig. 2(c).

C. Tropicalization via Dequantization of Algebraic Geometry

The algebraic side of tropical geometry [65] results from a transformation of analytic Euclidean geometry where the traditional arithmetic of the real field $(\mathbb{R}, +, \times)$ involved in the analytic expressions of geometric objects is replaced by the arithmetic of the max-plus or min-plus semiring. A geometric explanation and visualization of this transformation is obtained from Viro's graphing of polynomial curves on log-log paper [106]. Consider the monomial curve $v = cu^a$, $c > 0$, on the positive quadrant of the (u, v) plane and consider the log-log transformation of both coordinates composed with a uniform scaling by $\theta > 0$: $x = \theta \log u$, $y = \theta \log v$. Then, on the (x, y) plane the curve becomes the line $y = b/\theta + ax$, where $b = \log c$. If we have a K-term polynomial curve $v = P(u) = \sum_{k=1}^{K} c_k u^{a_k}$ with $c_k = \exp(b_k) > 0$ and $a_k \in \mathbb{R}$ (i.e. a posynomial [12]) then we convert it to

$$P_\theta(x) = \theta \log \left(\sum_{k=1}^{K} \exp(b_k/\theta) \exp(a_k x/\theta) \right) \tag{9}$$

As $\theta \downarrow 0$ this yields via Maslov dequantization a K-term **1D max-plus tropical polynomial**

$$\lim_{\theta \downarrow 0} P_\theta(x) = p(x) = \max_{k=1}^{K} \left\{ b_k + a_k x \right\} \tag{10}$$

While each $P_\theta(x)$ is a smooth function, their limit $p(x)$ is a max-affine function and represents a PWL convex function. If we perform dequantization with negative exponents we obtain a min-plus polynomial which is a PWL concave function.

The above procedure extends to multiple dimensions or higher degrees and shows us the way to tropicalize any classical d-variable polynomial (linear combination of power monomials) $\sum_k c_k u_1^{a_{k1}} \cdots u_d^{a_{kd}}$ defined over $\mathbb{R}^d_{>0}$ where $c_k > 0$ and $\boldsymbol{a}_k = (a_{k1}, ..., a_{kd})^T$ is traditionally some nonnegative integer[3] vector but herein we allow $\boldsymbol{a}_k \in \mathbb{R}^d$: replace the sum with max and log the individual monomials. Thus, a general d-variable max-plus polynomial $p : \mathbb{R}^d \to \mathbb{R}$ has the expression:

$$p(\boldsymbol{x}) = \bigvee_{k=1}^{K} \boldsymbol{a}_k^T \boldsymbol{x} + b_k, \quad \boldsymbol{x} = (x_1, \ldots, x_d)^T \tag{11}$$

where $K = \operatorname{rank}(p)$ is the number of terms of p. Its graph is a max of K hyperplanes with intercepts $b_k = \log c_k \in \mathbb{R}$ and real slope vectors $\boldsymbol{a}_k \in \mathbb{R}^d$. The degree of p is $|\boldsymbol{a}| = \max_k \|\boldsymbol{a}_k\|_1$ where $\|\boldsymbol{a}_k\|_1 = |a_{k1}| + \cdots + |a_{kd}|$. Thus, the curves or surfaces of real algebraic geometry become via dequantization the graphs of *convex* PWL functions represented by tropical polynomials.

D. Tropical Curves and Newton Polytopes

To the zero set of a classical polynomial there corresponds the *tropical curve or hypersurface* of a max-plus tropical polynomial $p : \mathbb{R}^d \to \mathbb{R}$

$$\mathcal{V}(p) := \{\boldsymbol{x} \in \mathbb{R}^d : \text{more than one terms of } p(\boldsymbol{x}) \text{ attain the max}\} \tag{12}$$

The above also defines the tropical curve of min-plus polynomials by replacing max with min. Thus, $\mathcal{V}(p)$ consists of the singularity points (of non-differentiability) of $p(\boldsymbol{x})$. Examples are shown in Fig. 3 for degree-1 tropical polynomials and in Fig. 2(c) for a degree-2 polynomial.

Another interesting geometric object related to a max-plus polynomial p is its *Newton polytope* which is the convex hull (denoted by $\operatorname{conv}(\cdot)$) of the set of points represented by its slope coefficient vectors:

$$\operatorname{Newt}(p) := \operatorname{conv}\{\boldsymbol{a}_k : k = 1, \ldots, \operatorname{rank}(p)\} \tag{13}$$

This satisfies several important properties [18]:

$$\operatorname{Newt}(p_1 \vee p_2) = \operatorname{conv}(\operatorname{Newt}(p_1) \cup \operatorname{Newt}(p_2)) \tag{14}$$

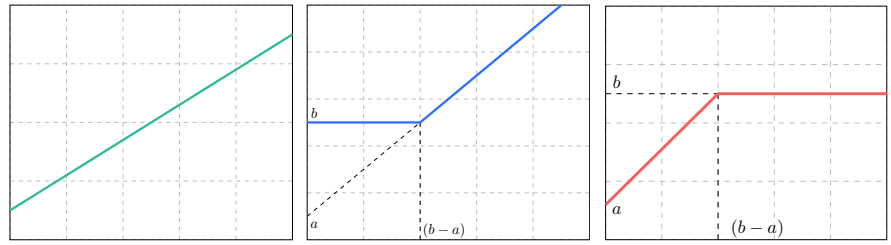
$$\operatorname{Newt}(p_1 + p_2) = \operatorname{Newt}(p_1) \oplus \operatorname{Newt}(p_2) \tag{15}$$

where $\oplus$ denotes Minkowski set addition, defined in (21). Examples are shown in Fig. 4. Thus, the Newton polytope of the sum (resp. max) of two tropical polynomials is the Minkowski sum (resp. the convex hull of the union) of their individual polytopes.

E. Tropical Halfspaces and Polytopes

In pattern analysis problems on Euclidean spaces $\mathbb{R}^d$ we often use halfspaces $\mathcal{H}(\boldsymbol{a}, b) := \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{a}^T \boldsymbol{x} \leq b\}$, polyhedra (finite intersections of halfspaces), and polytopes (compact polyhedra formed as the convex hull of a finite set

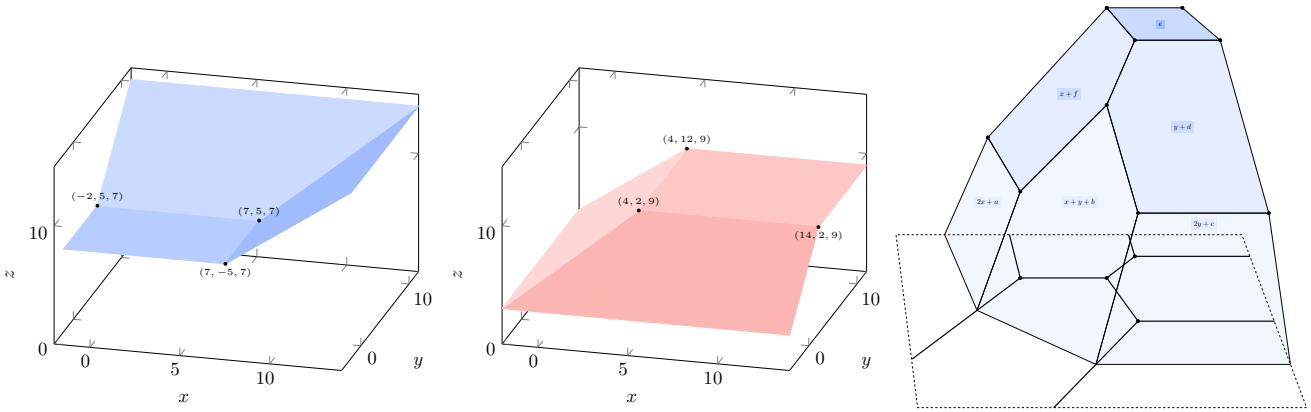
[3]Traditionally, 'tropical polynomials' assume that the parameters a_{ki} are nonnegative integers. If we also allow negative integers, we get 'Laurent tropical polynomials'. As in [15], we allow any real coefficients; this may be called 'tropical posynomials' [16].



(a) Euclidean line (b) Max-plus line (c) Min-plus line

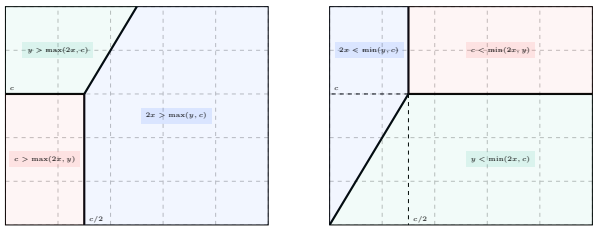
(d) Euclidean parabola (e) Max-plus parabola (f) Min-plus parabola

Figure 1. Euclidean and tropical 1D polynomial curves of 1st and 2nd degree.



(a) Max-plus plane f (b) Min-plus plane g (c) Tropic Conic p

Figure 2. (a),(b) Surfaces (graphs) of the two tropical planes defined in (6). (c) Surface (graph) of the 2D min-plus tropical polynomial function $p(x, y) = \min(a + 2x, b + x + y, c + 2y, d + x, e + y, f)$ and its tropical quadratic curve. ((c) is inspired by Fig. 1.3.2 of [65].)



(a) Max-plus curve (b) Min-plus curve

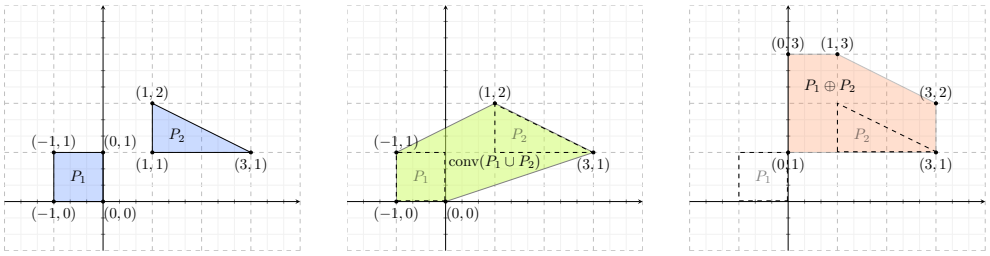
Figure 3. Tropical curve of the max-polynomial $p(x, y) = \max(2x, y, c)$ left and its dual min-polynomial $p'(x, y) = \min(2x, y, c)$ right.

of points). Replacing linear inner products $\boldsymbol{a}^T\boldsymbol{x}$ with max-plus versions yields *tropical halfspaces* [34] with parameters $\boldsymbol{a} = [a_i], \boldsymbol{b} = [b_i] \in \mathbb{R}^{d+1}_{\max}$:

$$\mathcal{T}(\boldsymbol{a}, \boldsymbol{b}) = \left\{ \boldsymbol{x} \in \mathbb{R}^d : \begin{array}{l} \max\{a_1 + x_1, \ldots, a_d + x_d, a_{d+1}\} \leq \\ \max\{b_1 + x_1, \ldots, b_d + x_d, b_{d+1}\} \end{array} \right\} \tag{16}$$

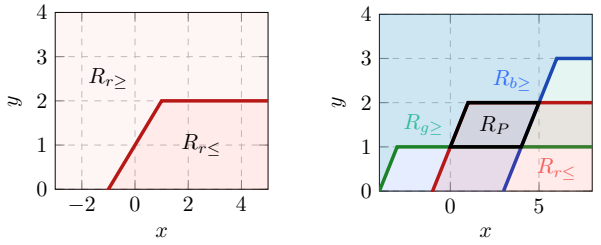
where $\min(a_i, b_i) = -\infty \ \forall i$. Thus, for each i, only one coefficient is needed either in the left or in the right side of inequality (16). Replacing max with min yields tropical halfspaces that are min-plus hyperplanes. Examples of polytopes in the plane are shown in Fig. 5. Obviously, their separating boundaries are tropical lines. Such regions in multiple dimensions were used in [18], [19], [108] as morphological perceptrons.

As an example in 3D space, in Fig. 6 we can see the intersection of the tropical halfspaces corresponding to the two tropical polynomials in (6). This polytope is the polyhedral region formed by intersecting the halfspace above the surface of the 2D max-plus polynomial f with the halfspace below the surface of the min-plus polynomial g.



(a) Polytopes (b) Newton (max) (c) Newton (sum)

Figure 4. Newton polytopes of (a) two max-polynomials $p_1(x, y) = \max(x + y, 3x + y, x + 2y)$ and $p_2(x, y) = \max(0, -x, y, y - x)$, (b) their max $p_1 \vee p_2$, and (c) their sum $p_1 + p_2$.



(a) Single region (b) Multiple regions

Figure 5. Regions $R_{c\geq}$ and $R_{c\leq}$ formed by min-plus tropical halfspaces in $\mathbb{R}^2$, where c denotes the color of the tropical boundary and ≥ 0 (resp. ≤ 0) the set of points above (resp. below) the boundary. (a) The red boundary is the min-plus tropical line $y = \min(1 + x, 2)$. (b) The green and blue boundaries are respectively the tropical lines $y = \min(4 + x, 1)$ and $y = \min(x - 3, 3)$. R_P is the polytope formed by the intersection of three tropical halfplanes.

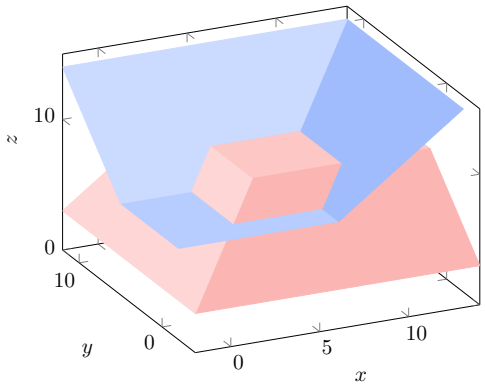


Figure 6. Intersection of halfspaces of the 2D max-plus and min-plus tropical polynomials in (6).

We note from Fig. 5 and Fig. 6 that the number of tropical boundaries required to form polytopes, which could serve as decision regions in pattern classification problems, is smaller than the number of linear boundaries. See, for instance, the polytope R_P in Fig. 5(b). This observation remains valid in higher dimensions too; namely, decision regions can be formed with fewer tropical lines or hyper-planes than their Euclidean counterparts. Intuitively, the nonlinearity of a tropical halfspace lets us form more complex decision regions with possibly fewer parameters.

III. Elements of Max-plus Algebra, Weighted Lattices, and Monotone Operators

A. Lattices and Monotone Operators

Signals and vectors can be viewed as elements of *complete lattices* $(\mathcal{L}, \vee, \wedge)$, where $\mathcal{L}$ is the set of lattice elements equipped with two binary operations, $\vee$ and $\wedge$, which denote the lattice supremum and infimum respectively. Each of these operations induces a partial ordering $\leq$; e.g. for any $X, Y \in \mathcal{L}$, $X \leq Y \iff Y = X \vee Y$. The lattice operations satisfy many properties, including associativity, commutativity, idempotence, and compatibility with the partial ordering. Completeness means that the supremum and infimum of any (even infinite) subset of $\mathcal{L}$ exists and belongs to $\mathcal{L}$. Examples of complete lattices used in image processing include (i) the lattice of Euclidean shapes, i.e. subsets of $\mathbb{R}^d$, equipped with set union and intersection, and (ii) the lattice of functions $f : E \to \overline{\mathbb{R}}$ with (arbitrary) domain E and values in $\overline{\mathbb{R}}$, equipped with the pointwise supremum and pointwise infimum of extended real numbers.

Monotone Operators: For data processing, we also consider operators $\psi : \mathcal{L} \to \mathcal{M}$ between two complete lattices. A lattice operator ψ is called *increasing* if it is order preserving; i.e. if, for any $X, Y \in \mathcal{L}$, $X \leq Y \implies \psi(X) \leq \psi(Y)$. Examples of increasing operators are the lattice homomorphisms which preserve suprema and infima. If a lattice homomorphism is also a bijection, then it becomes an automorphism. Four fundamental types of increasing operators are: *dilations* δ and *erosions* ε that satisfy respectively $\delta(\bigvee_i X_i) = \bigvee_i \delta(X_i)$ and $\varepsilon(\bigwedge_i X_i) = \bigwedge_i \varepsilon(X_i)$ over arbitrary (possibly infinite) collections; *openings* α that are increasing, idempotent ($\alpha^2 = \alpha$), and antiextensive ($\alpha \leq \mathbf{id}$), where $\mathbf{id}$ denotes the identity operator; *closings* β that are increasing, idempotent, and extensive ($\beta \geq \mathbf{id}$).

A lattice operator ψ is called *decreasing* if it is order-inverting, i.e. $X \leq Y \implies \psi(X) \geq \psi(Y)$. Dual homomorphisms interchange suprema with infima and hence are decreasing operators. For example, *anti-dilations* δ^a satisfy $\delta^a(\bigvee_i X_i) = \bigwedge_i \delta^a(X_i)$. A lattice dual automorphism is a bijection that interchanges suprema with infima. For example, a *negation* ν is a dual automorphism that is also involutive, i.e. $\nu^2 = \mathbf{id}$.

Residuation and Adjunctions: An increasing operator $\psi : \mathcal{L} \to \mathcal{M}$ between two complete lattices is called *residuated*

[8], [9] if there exists an increasing operator $\psi^\sharp : \mathcal{M} \to \mathcal{L}$ such that

$$\psi\psi^\sharp \leq \mathbf{id} \leq \psi^\sharp\psi \tag{17}$$

Here, $\psi^\sharp$ is called the **residual** of ψ, is unique, and is the closest to being an inverse of ψ. Specifically, the residuation pair $(\psi, \psi^\sharp)$ can solve inverse problems of the type $\psi(X) = Y$ either exactly since $\hat{X} = \psi^\sharp(Y)$ is the greatest solution of $\psi(X) = Y$ if a solution exists, or approximately since $\hat{X}$ is the *greatest subsolution* in the sense that

$$\hat{X} = \psi^\sharp(Y) = \bigvee\{X : \psi(X) \leq Y\} \tag{18}$$

On complete lattices an increasing operator ψ is residuated (resp. a residual $\psi^\sharp$) if and only if it is a dilation (resp. erosion). The residuation theory has been used for solving inverse problems (mainly in matrix algebra) over the extended max-plus semiring $(\overline{\mathbb{R}}, \vee, +)$ or other idempotent semirings which as lattices are made complete [6], [23], [25], [26].

A pair (δ, ε) of two operators $\delta : \mathcal{L} \to \mathcal{M}$ and $\varepsilon : \mathcal{M} \to \mathcal{L}$ between two complete lattices is called **adjunction** if

$$\delta(X) \leq Y \iff X \leq \varepsilon(Y) \quad \forall X \in \mathcal{L}, Y \in \mathcal{M} \tag{19}$$

In any adjunction, δ is a dilation and ε is an erosion. The double inequality (19) is equivalent to the inequality (17) satisfied by a residuation pair of increasing operators if we identify the residuated map ψ with δ and its residual $\psi^\sharp$ with ε. Further, from (19) or (17) it follows that any adjunction (δ, ε) automatically yields an opening $\alpha = \delta\varepsilon$ and a closing $\beta = \varepsilon\delta$, where the composition of two operators is written as an operator product. To view (δ, ε) as an adjunction instead of a residuation pair has the advantage of the additional geometrical intuition and visualization afforded by the dilation and erosion operators in image and shape analysis.

Given a dilation δ, there is a unique erosion

$$\varepsilon(Y) = \delta^\sharp(Y) = \bigvee\{X \in \mathcal{L} : \delta(X) \leq Y\} \tag{20}$$

such that (δ, ε) is an adjunction, and conversely. Thus, dilations and erosions on complete lattices always come in pairs. In any adjunction (δ, ε), ε is called the *adjoint erosion* of δ, whereas δ is the *adjoint dilation* of ε.

Example 1. (a) A morphological set adjunction is the pair of Minkowski set addition $\oplus$ and subtraction $\ominus$: for $X, B \subseteq \mathbb{R}^d$

$$\begin{array}{rcl} \delta_B(X) = X \oplus B & := & \{\boldsymbol{x} + \boldsymbol{b} \in \mathbb{R}^d : \boldsymbol{x} \in X,\, \boldsymbol{b} \in B\} \\ \varepsilon_B(X) = X \ominus B & := & \{\boldsymbol{x} - \boldsymbol{b} \in \mathbb{R}^d : \boldsymbol{x} \in X,\, \boldsymbol{b} \in B\} \end{array} \tag{21}$$

(b) A signal adjunction is the supremal (max-plus) convolution $f \oplus g$ of $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ by g and the infimal convolution $f \ominus g$ of $f(\boldsymbol{x})$ by $-g(-\boldsymbol{x})$ used in morphological image processing:

$$\begin{array}{rcl} \delta_g(f)(\boldsymbol{x}) = f \oplus g(\boldsymbol{x}) & := & \sup_{\boldsymbol{y}}\{f(\boldsymbol{y} - \boldsymbol{x}) + g(\boldsymbol{y})\} \\ \varepsilon_g(f)(\boldsymbol{x}) = f \ominus g(\boldsymbol{x}) & := & \inf_{\boldsymbol{y}}\{f(\boldsymbol{x} - \boldsymbol{y}) - g(\boldsymbol{y})\} \end{array} \tag{22}$$

B. Max-$\star$ Algebra and Weighted Lattices

1) Clodum – Extending Tropical Scalar Arithmetic: A lattice $\mathcal{M}$ is often endowed with an additional binary operation, called symbolically the 'multiplication' $\star$, under which $(\mathcal{M}, \star)$ is a semigroup or a monoid or a group. Such ordered monoids have been studied in detail in [7], [37], [111] and form the algebraic basis of max-plus algebra.

Consider now an algebra $(\mathcal{K}, \vee, \wedge, \star, \star')$ with four binary operations satisfying the following:
(C1) $(\mathcal{K}, \vee, \wedge)$ is a complete distributive lattice. Thus, it contains its least $\bot := \bigwedge \mathcal{K}$ and greatest element $\top := \bigvee \mathcal{K}$. The supremum $\vee$ (resp. infimum $\wedge$) plays the role of a generalized *'addition'* (resp. *'dual addition'*).
(C2) $(\mathcal{K}, \star)$ is a monoid whose operation $\star$ plays the role of a generalized *'multiplication'* with identity ('unit') element e and is a dilation (i.e. distributes over $\vee$).
(C3) $(\mathcal{K}, \star')$ is a monoid with identity e' whose operation $\star'$ plays the role of a generalized *'dual multiplication'* and is an erosion (i.e. distributes over $\wedge$).
The least (greatest) element $\bot$ ($\top$) of $\mathcal{K}$ is both the 'zero' element for the 'addition' $\vee$ ($\wedge$) and an absorbing null for the 'multiplication' $\star$ ($\star'$).
To the above definitions we add the word *complete* if $\mathcal{K}$ is a complete lattice and the distributivities involved are infinite. We call the resulting algebra a *complete lattice-ordered double monoid*, in short **clodum** [68], [70]. Previous works on minimax or max-plus algebra have used alternative names for structures similar to the above definitions which emphasize semigroups and semirings instead of lattices [6], [25], [37]; see [70] for similarities and differences.

A clodum $\mathcal{K}$ is called *self-conjugate* if it has a lattice negation $a \mapsto a^*$ such that

$$\big(\bigvee_i a_i\big)^* = \bigwedge_i {a_i}^*, \;\; \big(\bigwedge_i b_i\big)^* = \bigvee_i {b_i}^*, \;\; \big(a \star b\big)^* = a^* \star' b^* \tag{23}$$

The suprema and infima in (23) may be over any collections.

Examples of clodums are summarized in Table I. The max-plus and max-times clodums have a richer structure. Specifically, if $\star = \star'$ over $G = \mathcal{K} \setminus \{\bot, \top\}$ where $(G, \star)$ is a group and $(G, \vee, \wedge)$ is a conditionally complete lattice, then the clodum $\mathcal{K}$ becomes a *complete lattice-ordered group*, in short **clog**. Then, for each $a \in G$ there exists its 'multiplicative inverse' a^{-1} such that $a \star a^{-1} = e$. Further, the 'multiplication' $\star$ and its self-dual $\star'$ can be extended over the whole $\mathcal{K}$ by involving the nulls, and the clodum becomes self-conjugate by setting $a^* = a^{-1}$ if $\bot < a < \top$, $\top^* = \bot$, and $\bot^* = \top$. Thus, in a clog $\mathcal{K}$ the $\star$ and $\star'$ coincide in all cases with only one exception: the combination of the least and greatest elements.

All clodum examples of Table I have commutative 'multiplications'. An example with *non-commutative* 'multiplications' is the *matrix* max-$\star$ clodum $(\mathcal{K}^{n\times n}, \vee, \wedge, \boxtimes, \boxtimes')$ where $\mathcal{K}^{n\times n}$ is the set of $n \times n$ matrices with entries from a clodum $\mathcal{K}$, $\vee/\wedge$ denote here elementwise matrix sup/inf, and $\boxtimes$, $\boxtimes'$ denote max-$\star$ and min-$\star'$ matrix 'multiplications':

$$[\boldsymbol{A} \boxtimes \boldsymbol{B}]_{ij} = \bigvee_{k=1}^{n} a_{ik} \star b_{kj} \;,\; [\boldsymbol{A} \boxtimes' \boldsymbol{B}]_{ij} = \bigwedge_{k=1}^{n} a_{ik} \star' b_{kj} \tag{24}$$

For the max-plus clog $(\overline{\mathbb{R}}, \vee, \wedge, +, +')$, these matrix 'multiplications' are denoted by $\boxplus$ and $\boxplus'$, defined as

$$[\boldsymbol{A} \boxplus \boldsymbol{B}]_{ij} = \bigvee_{k=1}^{n} a_{ik} + b_{kj} \;,\; [\boldsymbol{A} \boxplus' \boldsymbol{B}]_{ij} = \bigwedge_{k=1}^{n} a_{ik} +' b_{kj} \tag{25}$$

Table I
EXAMPLES OF CLODUMS.

Clodum	Set $\mathcal{K}$	'Add'	'Zero' $\bot$	'Dual Add'	'Dual Zero' $\top$	'Mult' $\star$	'Id' e	'Dual Mult' $\star'$	'Dual Id' e'	Conjugate a^*
Max-plus	$\overline{\mathbb{R}}$	$\vee$	$-\infty$	$\wedge$	$+\infty$	$+$	0	$+'$	0	$-a$
Max-times	$[0,+\infty]$	$\vee$	0	$\wedge$	$+\infty$	$\times$	1	$\times'$	1	a^{-1}
Max-min	$[0,1]$	$\vee$	0	$\wedge$	1	$\wedge$	1	$\vee$	0	$1-a$
Max-softmin	$\mathbb{R}$	$\vee$	$-\infty$	$\wedge$	$+\infty$	$\curlywedge_\theta$	$+\infty$	$\curlyvee_\theta$	$-\infty$	$-a$

2) Complete Weighted Lattices – Nonlinear Spaces: Consider a nonempty collection $\mathcal{W}$ of mathematical objects, which will be our space; examples of such objects include the vectors in $\overline{\mathbb{R}}^d$ or signals in $\mathrm{Fun}(E,\overline{\mathbb{R}})$. Also, consider a clodum $(\mathcal{K},\vee,\wedge,\star,\star')$ of scalars with *commutative* operations $\star,\star'$ and $\mathcal{K}\subseteq\overline{\mathbb{R}}$. We define *two internal operations* among vectors/signals X,Y in $\mathcal{W}$: their supremum $X\vee Y:\mathcal{W}^2\to\mathcal{W}$ and their infimum $X\wedge Y:\mathcal{W}^2\to\mathcal{W}$, which we denote using the same supremum symbol ($\vee$) and infimum symbol ($\wedge$) as in the clodum, hoping that the differences will be clear to the reader from the context. Further, we define *two external operations* among any vector/signal X in $\mathcal{W}$ and any scalar c in $\mathcal{K}$: a 'scalar multiplication' $c\star X:(\mathcal{K},\mathcal{W})\to\mathcal{W}$ and a 'scalar dual multiplication' $c\star' X:(\mathcal{K},\mathcal{W})\to\mathcal{W}$, again by using the same symbols as in the clodum. Now, we define $\mathcal{W}$ to be a **weighted lattice** space over the clodum $\mathcal{K}$ if it satisfies a set of axioms postulated in [70] which (i) make $\mathcal{W}$ a distributive lattice w.r.t. its two internal vector operations $\vee,\wedge$, and (ii) endow the external operations $\star,\star'$ between scalars and vectors with associativity and distributivity properties. These axioms bear a striking similarity with those of a linear space. One difference is that the vector/signal addition ($+$) of linear spaces is now replaced by two dual superpositions, the lattice supremum ($\vee$) and infimum ($\wedge$); further, the scalar multiplication ($\times$) of linear spaces is now replaced by two operations $\star$ and $\star'$ which are dual to each other. Only one major property of linear spaces is missing from the weighted lattices: the existence of 'additive inverses'. We define the space $\mathcal{W}$ to be a **complete weighted lattice (CWL)** if (i) $\mathcal{W}$ is closed under any (possibly infinite) suprema and infima, and (ii) the distributivity laws between the scalar operations $\star$ ($\star'$) and the supremum (infimum) are of the infinite type.

3) Vector and Signal Operators on Weighted Lattices: We focus on CWLs whose underlying set is a *space* $\mathcal{W}$ of *functions* $f:E\to\mathcal{K}$ with values from a clodum $(\mathcal{K},\vee,\wedge,\star,\star')$ of scalars. Such functions include d-dimensional vectors if $E=\{1,2,...,d\}$ or d-dimensional signals of continuous ($E=\mathbb{R}^d$) or discrete domain ($E=\mathbb{Z}^d$). Then, we extend *pointwise* the supremum, infimum, and scalar multiplications of $\mathcal{K}$ to functions: e.g., for $F,G\in\mathcal{W}$, $a\in\mathcal{K}$ and $x\in E$, we define $(F\vee G)(x):=F(x)\vee G(x)$ and $(a\star F)(x):=a\star F(x)$. Further, the scalar operations $\star$ and $\star'$, extended pointwise to functions, distribute over any suprema and infima, respectively. If the clodum $\mathcal{K}$ is self-conjugate, then we can extend the conjugation $(\cdot)^*$ to functions F pointwise: $F^*(x):=(F(x))^*$.

Elementary increasing operators on $\mathcal{W}$ are those that act as **vertical translations** (in short V-translations) of functions. Specifically, pointwise $\star$ ($\star'$) 'multiplications' of functions in $\mathcal{W}$ by scalars in $\mathcal{K}$ yield the (dual) *V-translations*. A function operator ψ on $\mathcal{W}$ is called **V-translation invariant** if it commutes with any V-translation τ, i.e., $\psi\tau=\tau\psi$. Similarly for dual translations.

More complex increasing operators are combinations of (dual) V-translations and dilations (erosions), called **dilation V-translation invariant (DVI)** operators δ or **erosion V-translation invariant (EVI)** operators ε. Such operators obey a sup-$\star$ or an inf-$\star'$ superposition:

$$\delta\big(\bigvee_i c_i\star F_i\big)=\bigvee_i c_i\star\delta(F_i),\quad \varepsilon\big(\bigwedge_i c_i\star' F_i\big)=\bigwedge_i c_i\star'\varepsilon(F_i) \tag{26}$$

On signal spaces these properties create supremal and infimal *nonlinear convolutions*; details can be found in [70].

Next we focus on finite-dimensional CWLs that are nonlinear vector spaces $\mathcal{W}=\mathcal{K}^d$, equipped with the pointwise partial ordering $\boldsymbol{x}\le\boldsymbol{y}$, supremum $\boldsymbol{x}\vee\boldsymbol{y}=[x_i\vee y_i]$, and infimum $\boldsymbol{x}\wedge\boldsymbol{y}=[x_i\wedge y_i]$ between any vectors $\boldsymbol{x},\boldsymbol{y}\in\mathcal{W}$. Then, $(\mathcal{W},\vee,\wedge,\star,\star')$ is a complete weighted lattice. Elementary increasing operators are the *vector V-translations* $\tau_a(\boldsymbol{x})=a\star\boldsymbol{x}=[a\star x_i]$ and their duals $\tau'_a(\boldsymbol{x})=a\star'\boldsymbol{x}$, which 'multiply' a scalar a with a vector $\boldsymbol{x}$ elementwise. A vector transformation on $\mathcal{W}$ is called (dual) V-translation invariant if it commutes with any vector (dual) V-translation. Each vector $\boldsymbol{x}=[x_1,...,x_d]^T$ can be expressed as max of V-translated impulse vectors $\boldsymbol{q}_j=[q_j(i)]$, where $q_j(i)=e$ at $i=j$ and $\bot$ else, or as min of dual V-translated impulses $\boldsymbol{q}'_j=[q'_j(i)]$, where $q'_j(i)=e'$ at $i=j$ and $\top$ else. Based on these vector representations, the following theorem establishes that all V-translation invariant dilations and erosions of vectors are essentially max-$\star$ and min-$\star'$ matrix-vector 'products', respectively.

Theorem 1 ([70]). *(a) Any vector transformation between two finite-dimensional CWLs, i.e. from $\mathcal{K}^n$ to $\mathcal{K}^m$ is DVI iff it can be represented as a matrix-vector max-$\star$ product $\delta_{\boldsymbol{A}}(\boldsymbol{x}):=\boldsymbol{A}\boxtimes\boldsymbol{x}$ where $\boldsymbol{A}=[a_{ij}]\in\mathcal{K}^{m\times n}$ with $a_{ij}=[\delta(\boldsymbol{q}_j)]_i$, $i=1,\dots,m$, $j=1,\dots,n$.*
(b) Any vector transformation from $\mathcal{K}^n$ to $\mathcal{K}^m$ is EVI iff it can be represented as a matrix-vector min-$\star'$ product $\varepsilon_{\boldsymbol{A}}(\boldsymbol{x}):=\boldsymbol{A}\boxtimes'\boldsymbol{x}$ where $\boldsymbol{A}=[a_{ij}]$ with $a_{ij}=[\varepsilon(\boldsymbol{q}'_j)]_i$.

Given such a vector dilation $\delta(\boldsymbol{x})=\boldsymbol{A}\boxtimes\boldsymbol{x}:\mathcal{K}^n\to\mathcal{K}^m$, there corresponds a unique erosion $\varepsilon:\mathcal{K}^m\to\mathcal{K}^n$ (equal to the residual operator $\delta^\sharp$) so that (δ,ε) is a *vector adjunction*, i.e. $\delta(\boldsymbol{x})\le\boldsymbol{y}\iff\boldsymbol{x}\le\varepsilon(\boldsymbol{y})$. We can find the adjoint vector erosion by decomposing both vector operators based on *scalar operators* (η,ζ) that form a *scalar adjunction* on $\mathcal{K}$:

$$\eta(a,v)\le w\iff v\le\zeta(a,w) \tag{27}$$

If we use as scalar 'multiplication' a commutative binary operation $\eta(a,v) = a \star v$ that is a dilation on $\mathcal{K}$, its scalar adjoint erosion becomes

$$\zeta(a,w) = \sup\{v \in \mathcal{K} : a \star v \leq w\} \tag{28}$$

which is a (possibly non-commutative) binary operation on $\mathcal{K}$. Then, the original vector dilation $\delta(\boldsymbol{x}) = \boldsymbol{A} \boxtimes \boldsymbol{x}$ is decomposed as

$$[\delta(\boldsymbol{x})]_i = \bigvee_{j=1}^{n} \eta(a_{ij}, x_j) = \bigvee_{j=1}^{n} a_{ij} \star x_j, \quad i = 1, ..., m \tag{29}$$

whereas its adjoint vector erosion (i.e. the residual $\delta^\sharp$ of δ) is decomposed as

$$[\delta^\sharp(\boldsymbol{y})]_j = [\varepsilon(\boldsymbol{y})]_j = \bigwedge_{i=1}^{m} \zeta(a_{ij}, y_i), \quad j = 1, ..., n \tag{30}$$

Further, if $\mathcal{K} = (\vee, \wedge, \star, \star')$ is a *clog*, then $\zeta(a,w) = w \star' a^*$ and hence

$$\varepsilon(\boldsymbol{y}) = \boldsymbol{A}^* \boxtimes' \boldsymbol{y}, \quad [\varepsilon(\boldsymbol{y})]_j = \bigwedge_{i=1}^{m} y_i \star' {a_{ij}}^*, \quad j = 1, ..., n \tag{31}$$

where $\boldsymbol{A}^* = [{a_{ji}}^*]$ is the *adjoint matrix* (i.e. conjugate transpose) of $\boldsymbol{A} = [a_{ij}]$.

IV. Solving Max-$\star$ Equations and Optimization

A. ℓ_∞ Optimal Solutions of Max-plus Equations

Consider the max-plus clog $(\overline{\mathbb{R}}, \vee, \wedge, +, +')$, a matrix $\boldsymbol{A} \in \overline{\mathbb{R}}^{m \times n}$ and a vector $\boldsymbol{b} \in \overline{\mathbb{R}}^m$. The set of solutions of the max-plus equation

$$\boldsymbol{A} \boxplus \boldsymbol{x} = \boldsymbol{b} \tag{32}$$

over $\overline{\mathbb{R}}$ is either empty or forms an idempotent semigroup under vector $\vee$, because if $\boldsymbol{x}_1, \boldsymbol{x}_2$ are two solutions then $\boldsymbol{x}_1 \vee \boldsymbol{x}_2$ is also a solution. A related problem in applications of max-plus algebra to scheduling is when a vector $\boldsymbol{x}$ represents start times, a vector $\boldsymbol{b}$ represents finish times, and the matrix $\boldsymbol{A}$ represents processing delays. Then, if (32) does not have an exact solution, it is possible to find the optimum $\boldsymbol{x}$ such that we minimize a norm of the earliness subject to zero lateness:

$$\text{Minimize } \|\boldsymbol{A} \boxplus \boldsymbol{x} - \boldsymbol{b}\|_p \text{ s.t. } \boldsymbol{A} \boxplus \boldsymbol{x} \leq \boldsymbol{b} \tag{33}$$

where $\|\cdot\|_p$ denotes the ℓ_p norm. Both problem (32) and the constrained minimization problem (33) for $p = 1$ or $p = \infty$ have been solved by Cuninghame-Green [25].

Theorem 2 ([25]). *If Eq. (32) has a solution, then*[4]

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^* \boxplus' \boldsymbol{b} = \left[\bigwedge_{i=1}^{m} b_i - a_{ij} \right] \tag{34}$$

is its greatest solution and the optimum solution to problem (33).

[4]To cover all cases of combining finite and infinite scalar numbers in the max-plus clog $(\overline{\mathbb{R}}, \vee, \wedge, +, +')$, we should write the subtractions $b_i - a_{ij}$ in (34) as $b_i +' (-a_{ij})$ and use the rules (1).

The proof results since $\hat{\boldsymbol{x}}$ is the greatest solution of $\boldsymbol{A} \boxplus \boldsymbol{x} \leq \boldsymbol{b}$, as shown in [15], [25]. It can also be directly seen from the adjunction (δ, ε) where

$$\boldsymbol{A} \boxplus \boldsymbol{x} = \delta(\boldsymbol{x}) \leq \boldsymbol{b} \Longleftrightarrow \boldsymbol{x} \leq \varepsilon(\boldsymbol{b}) = \boldsymbol{A}^* \boxplus' \boldsymbol{b} \tag{35}$$

The solutions of (32) and of (33) for the ℓ_∞ case have been further analyzed in [15] both algebraically and combinatorially. It is also possible to search and find *sparse solutions* of either the exact equation (32) or the approximate problem (33), as done in [104], where sparsity here means a large number of $-\infty$ values in the solution vector.

Further, there is actually a stronger result that is not biased to be a subsolution of (32) but provides the *unconstrained optimal solution* of the following problem

$$\text{Minimize } \|\boldsymbol{A} \boxplus \boldsymbol{x} - \boldsymbol{b}\|_\infty \tag{36}$$

Theorem 3 ([15], [25]). *If $2\mu = \|\boldsymbol{A} \boxplus \hat{\boldsymbol{x}} - \boldsymbol{b}\|_\infty = \|\boldsymbol{A} \boxplus (\boldsymbol{A}^* \boxplus' \boldsymbol{b}) - \boldsymbol{b}\|_\infty$ is the ℓ_∞ error corresponding to the greatest subsolution of $\boldsymbol{A} \boxplus \boldsymbol{x} = \boldsymbol{b}$, then the unique solution of (36) is*

$$\tilde{\boldsymbol{x}} = \mu + \hat{\boldsymbol{x}} = \mu + \boldsymbol{A}^* \boxplus' \boldsymbol{b} \tag{37}$$

The computational complexity to find both optimal solutions $\hat{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}$ is $O(mn)$.

B. Projections on Weighted Lattices

The optimal subsolution of (33) can be viewed as a nonlinear 'projection' of $\boldsymbol{b}$ onto the column-space of $\boldsymbol{A}$ [26]. To understand this, note first that any adjunction (δ, ε) automatically yields two lattice projections, an opening $\alpha = \delta\varepsilon$ and a closing $\beta = \varepsilon\delta$, such that

$$\alpha^2 = \alpha \leq \mathbf{id} \leq \beta = \beta^2$$

We call them 'projections' because, in analogy to projection operators on linear spaces, they preserve the structure of the lattice space w.r.t. the partial ordering and they are idempotent.

Projections on idempotent semimodules[5] have been studied in [23] for the general case and with more details in [2], [26] for the max-plus case to which we focus herein. Let $\mathcal{X} = \overline{\mathbb{R}}^d$ be viewed as complete idempotent semimodule over the complete max-plus semiring $\mathbb{R}_{\max} \cup \{\infty\} = \overline{\mathbb{R}}$, and let $\mathcal{S}$ be a subsemimodule of $\mathcal{X}$. Then a *canonical projector* on $\mathcal{S}$ is defined as the nonlinear map [23]

$$P_{\mathcal{S}} : \mathcal{X} \to \mathcal{X}, \quad P_{\mathcal{S}}(\boldsymbol{x}) := \bigvee \{\boldsymbol{v} \in \mathcal{S} : \boldsymbol{v} \leq \boldsymbol{x}\} \tag{38}$$

Its definition implies that $P_{\mathcal{S}}$ is a lattice opening, i.e. increasing, antiextensive, and idempotent. Further, there is a concept of 'distance' on such semimodules which allows to use a nonlinear projection theorem for best approximations. Specifically, let us consider the *range semimetric* [25]

$$d_H(\boldsymbol{x}, \boldsymbol{y}) = \max_i (x_i - y_i) - \min_i (x_i - y_i), \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \tag{39}$$

[5]Idempotent semimodules are like vector spaces with vector 'addition' $\vee$ whose vector and scalar arithmetic are defined over idempotent semirings. If in our definition of a weighted lattice, one focuses only on one vector 'addition', say the supremum, and its corresponding scalar 'multiplication', then the weaker algebraic structure becomes an idempotent semimodule over an idempotent semiring $(\mathcal{K}, \vee, \star)$. This has been studied in [23], [37], [61].

also known, in a more general form, as the *Hilbert projective metric* [23]. Then for any vector $\boldsymbol{x} \in \overline{\mathbb{R}}^d$, $P_{\mathcal{S}}(\boldsymbol{x})$ is the best approximation (but not necessarily unique) of $\boldsymbol{x}$ by elements of $\mathcal{S}$. Specifically [2], [23], the projection $P_{\mathcal{S}}(\boldsymbol{x})$ of $\boldsymbol{x}$ onto $\mathcal{S}$ is any element of $\mathcal{S}$ attaining the shortest distance from $\boldsymbol{x}$; i.e.,

$$d_H(\boldsymbol{x}, P_{\mathcal{S}}(\boldsymbol{x})) = d_H(\boldsymbol{x}, \mathcal{S}) \tag{40}$$

where the distance between a vector $\boldsymbol{x}$ and the subspace $\mathcal{S}$ is defined by $d_H(\boldsymbol{x}, \mathcal{S}) := \inf\{d_H(\boldsymbol{x}, \boldsymbol{v}) : \boldsymbol{v} \in \mathcal{S}\}$. Note the analogy with Euclidean spaces $\mathbb{R}^d$ where the linear projection of a point $\boldsymbol{x} \in \mathbb{R}^d$ to a linear subspace $\mathcal{S}$ is given by the unique point $\boldsymbol{y} \in \mathcal{S}$ such that $\boldsymbol{x} - \boldsymbol{y}$ is orthogonal to $\mathcal{S}$.

Now, if we consider the optimization problem (33) and define the subsemimodule $\mathcal{S}$ in (38) as the max-plus span of the columns of matrix $\boldsymbol{A}$, then the canonical projection of $\boldsymbol{b}$ onto it equals

$$P_{\mathcal{S}}(\boldsymbol{b}) = \boldsymbol{A} \boxplus \hat{\boldsymbol{x}} = \boldsymbol{A} \boxplus (\boldsymbol{A}^* \boxplus' \boldsymbol{b}) \leq \boldsymbol{b} \tag{41}$$

which is a lattice opening $\delta(\varepsilon(\boldsymbol{b})) \leq \boldsymbol{b}$ from (35).

C. ℓ_p Optimal Solutions of Max-$\star$ Equations

Herein we generalize the results of Sec. IV-A from max-plus to max-$\star$ algebra. Consider a scalar commutative clodum $(\mathcal{K}, \vee, \wedge, \star, \star')$, a matrix $\boldsymbol{A} \in \mathcal{K}^{m \times n}$ and a vector $\boldsymbol{b} \in \mathcal{K}^m$. We consider the set of solutions of both the exact max-$\star$ equation

$$\boldsymbol{A} \boxtimes \boldsymbol{x} = \boldsymbol{b} \tag{42}$$

as well as its approximate solutions that are optimal solutions of the following constrained minimization problem:

$$\text{Minimize } \|\boldsymbol{A} \boxtimes \boldsymbol{x} - \boldsymbol{b}\|_p \text{ s.t. } \boldsymbol{A} \boxtimes \boldsymbol{x} \leq \boldsymbol{b} \tag{43}$$

where $\|\cdot\|_p$ is any ℓ_p norm with $p = 1, 2, \ldots, \infty$. By using adjunctions, we provide next a more general result (than Theorem 2) for the general case when $\mathcal{K}$ is a general clog or just a clodum (which has no inverses for its 'multiplication' operations).

Theorem 4 ([70])**.** *Consider the vector dilation $\delta(\boldsymbol{x}) = \boldsymbol{A} \boxtimes \boldsymbol{x} : \mathcal{K}^n \to \mathcal{K}^m$ and let ε be its adjoint vector erosion.*
(a) If Eq. (42) has a solution, then

$$\hat{\boldsymbol{x}} = \varepsilon(\boldsymbol{b}) = \left[\bigwedge_{i=1}^{m} \zeta(a_{ij}, b_i)\right] \tag{44}$$

is its greatest solution, where ζ is the scalar adjoint erosion of $\star$ as in (28).
(b) If $\mathcal{K}$ is a clog, the solution (44) becomes

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^* \boxtimes' \boldsymbol{b} = \left[\bigwedge_{i=1}^{m} b_i \star' {a_{ij}}^*\right] \tag{45}$$

(c) The solution to the optimization problem (43) for any ℓ_p norm $\|\cdot\|_p$ is generally (44), or (45) in the case of a clog.

A main idea for solving (43) is to consider vectors $\boldsymbol{x}$ that are *subsolutions* in the sense that $\delta(\boldsymbol{x}) = A \boxtimes \boldsymbol{x} \leq \boldsymbol{b}$ and find the greatest such subsolution $\hat{\boldsymbol{x}} = \varepsilon(\boldsymbol{b})$, which yields either the greatest exact solution of (42) or an optimum subsolution in the sense of (43). To prove the latter note that, since $\boldsymbol{y} = \delta(\varepsilon(\boldsymbol{b}))$ is the greatest lower estimate of $\boldsymbol{b}$, $b_i - y_i$ is nonnegative and minimum for all i, and hence the norm $\|\boldsymbol{b} - \boldsymbol{y}\|_p$ is minimum for any $p = 1, 2, \ldots, \infty$.

Unfortunately, the type of unconstrained ℓ_∞ optimal solution offered by Theorem 3 in the max-plus case does not generally carry over to a general clodum, as shown for the max-min clodum in [27].

V. Tropical Geometry of Neural Networks with PWL Activations

In this section, we present some applications of concepts and techniques from tropical geometry in studying neural networks with piecewise linear activations. Early connections between tropical geometry and neural networks were sketched in [18] and later developed in greater detail in [19], [109]. Tools from tropical geometry (in particular, the *Maslov dequantization*) have also been used to design neural networks that approximate convex and log-log convex data [16] and general continuous functions over convex sets [17]. For the remainder of the section, we primarily develop the tropical-geometric characterization of neural network layers following [19], [109], and describe other applications near its end.

A central motivation for the use of tropical geometry in the study of neural networks is characterizing their *expressive power*. Tools used for this purpose range from the *Vapnik-Chervonenkis* (VC) dimension to the *activation pattern* of a neural network. The seminal work of [80], [84] proposed studying the expressive power of networks whose output is a piecewise-linear function via the number of its *inference regions* (also interchangeably called **linear regions**) – defined to be the *maximally connected partitions of the input space, on which the output of the network is a linear function*. Intuitively, networks with many linear regions can represent more complicated functions compared to networks with only a few regions of linearity. Upper and lower bounds on the number of linear regions of ReLU networks have been derived in e.g. [4], [80], [84], [94], using arguments from combinatorics and/or polyhedral geometry. As tropical geometry is centered around the study of piecewise linear curves, it emerges as a natural tool for tackling this problem.

A. A geometric characterization of NN layers

Motivated by the approach of [82], we seek a similar characterization of the geometry of a neural network in terms of the vertices of the appropriate Newton polytopes. With such a characterization at hand, we will then proceed to derive a simple geometric algorithm for enumerating the number of these vertices given a fixed network, to serve as a proxy for its expressive power. Our initial observation is that all piecewise-linear activation functions used in practice are tropical polynomials:

Example 2 (ReLU / Leaky ReLU)**.** Given input $v = \boldsymbol{w}^T \boldsymbol{x} + b$ with $\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^d$, a Rectifier Linear Unit computes

$$\text{ReLU}(v) = \max(0, v). \tag{46}$$

A commonly used variation is the Leaky ReLU [64], which computes (for some $\alpha \in (0,1)$):

$$\mathrm{LReLU}_\alpha(v) = \max(v, \alpha v). \tag{47}$$

Thus ReLU/LReLU units, viewed as functions of the input $\boldsymbol{x}$, are simply tropical polynomials of rank 2.

Example 3 (Maxout)**.** Given $\boldsymbol{W} \in \mathbb{R}^{d\times k}$ and $\boldsymbol{b} \in \mathbb{R}^k, \boldsymbol{x} \in \mathbb{R}^d$:

$$\mathrm{maxout}(\boldsymbol{x}) = \max_{j\in[k]} \left(\boldsymbol{W}_j^T \boldsymbol{x} + b_j\right), \tag{48}$$

where we denote $\boldsymbol{W}_j$ for the j-th row of $\boldsymbol{W}$. Thus a maxout unit is a tropical polynomial of rank k.

These connections were observed in [18]. In particular, the authors showed that for a single maxout unit, the number of linear regions it determines is equal to the number of vertices in the *upper hull* of its **extended Newton polytope**, defined as

$$\mathrm{ENewt}(p) := \mathrm{conv}\{(b_j, \boldsymbol{a}_j) : j \in [K]\}, \tag{49}$$

where $p(\boldsymbol{x})$ is given in the form of (11). For a polytope P, its **upper hull** is defined as

$$P^{\max} := \{(\lambda, \boldsymbol{x}) : \lambda = \sup\{t \in \mathbb{R} : (t, \boldsymbol{x}) \in P\}\} \tag{50}$$

An simple example is shown in Fig. 7.

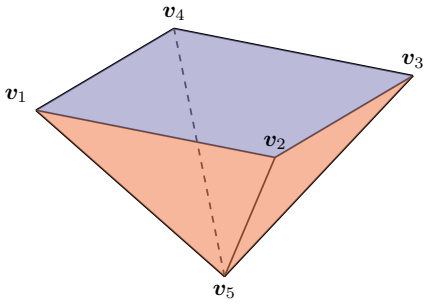


Figure 7. $P := \mathrm{conv}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_5\}$. The upper hull, $P^{\max}$, is depicted in light blue color.

Proposition 1 ([18])**.** *The linear regions defined by a piecewise-linear function p of the form* (11) *are in bijection with the number of vertices in* $\mathrm{ENewt}^{\max}(p)$.

Proof sketch. The function computed by the tropical polynomial at each $\boldsymbol{x}$ is the value of the following linear program:

$$p(\boldsymbol{x}) := \max\left\{b + \boldsymbol{a}^T\boldsymbol{x} : \begin{pmatrix} b & \boldsymbol{a}^T \end{pmatrix}^T \in \mathrm{ENewt}(p)\right\} \tag{51}$$

It is straightforward to show that minimizers to (51) cannot exist outside $\mathrm{ENewt}(p)$; an appeal to the fundamental theorem of linear programming completes the proof. □

Proposition 1 is limited as it only characterizes a single PWL unit. However, it forms the basis for a geometric characterization of an entire NN layer. In particular, we can view each layer as a collection of tropical polynomials. Recall that the tropical hypersurface of a tropical polynomial p, $\mathcal{V}(p)$, is the set of points $\boldsymbol{x}$ on which $p(\boldsymbol{x})$ is nondifferentiable. Given a collection $p_1, \ldots, p_m$, the union $\bigcup_i \mathcal{V}(p_i)$ contains all the points $\boldsymbol{x}$ for which *at least* one of the polynomials is non-differentiable. Thus each region of linearity of a neural network layer corresponds to an open cell in $\bigcup_i \mathcal{V}(p_i)$.

We may now appeal to a fundamental duality result from tropical geometry, restated in the language necessary for our application. For a proof, see [19, Proposition 1], as well as the discussion following [109, Definition 3.2].

Proposition 2. *Let $p_1, \ldots, p_m : \mathbb{R}^d \to \mathbb{R}$ denote a collection of tropical polynomials. Moreover, let $\mathcal{V}(p)$ denote the tropical hypersurface of a polynomial p. Then the number of open cells induced by $\bigcup_{i=1}^m \mathcal{V}(p_i)$ is equal to the number of vertices in* $\mathrm{Newt}(p_1) \oplus \cdots \oplus \mathrm{Newt}(p_m)$.

An illustration appears in Fig. 8. By Proposition 2 and the preceding discussion, we have reduced the problem of counting linear regions to that of counting the number of vertices of Minkowski sums of Newton polytopes. However, as the tropical polynomials involved also have constant terms, we need to apply a "lifting" argument to treat the p_i's as functions on $\mathbb{R}^{d+1}$, and apply Proposition 2. The resulting Minkowski sum is precisely

$$\mathrm{ENewt}(p_1) \oplus \cdots \oplus \mathrm{ENewt}(p_m) = \mathrm{ENewt}\left(\sum_{i=1}^m p_i\right). \tag{52}$$

Thus it suffices to count the number of vertices in the upper hull of the Minkowski sum of Eq. (52). Based on this observation, one may appeal to standard results on the number of vertices of Minkowski sums.

Theorem 5 ([39])**.** *Let $P_1, \ldots P_k$ be polytopes in $\mathbb{R}^d$ and let m denote the number of their nonparallel edges. Then the number of vertices of $P_1 \oplus \cdots \oplus P_k$ is bounded above by*

$$2\sum_{j=0}^{d-1} \binom{m-1}{j} \tag{53}$$

Moreover, the bound of (53) *is tight when $2k > d$.*

When $P_1, \ldots, P_m$ are the extended Newton polytopes of ReLU units, the number of non-parallel edges is at most m, since each polytope is a line segment. When $P_1, \ldots, P_m$ are generic maxout units of rank k, each polytope has at most k vertices, hence the number of non-parallel edges is at most $m \cdot \binom{k}{2} = m \cdot \frac{k(k-1)}{2}$. Since Theorem 5 gives upper bounds for the *total* number of vertices, it is not clear a-priori how loose these bounds are for the number of vertices in the *upper hull*; when each P_i is the Minkowski sum of line segments, a symmetry argument can be invoked to yield bounds for the upper hull. The results are summarized below.

Corollary 1. *The number of linear regions of a neural network layer with d inputs and m outputs is upper bounded as follows:*

1) in the case of ReLU/LReLU activations, we have

$$\mathcal{N}_m^d \leq \min\left\{2^m, \sum_{j=0}^d \binom{m}{j}\right\}. \tag{54}$$

The bound in (54) *is tight if all the line segments generating the Newton polytopes, as well as their projections to the last d coordinates, are in general position.*

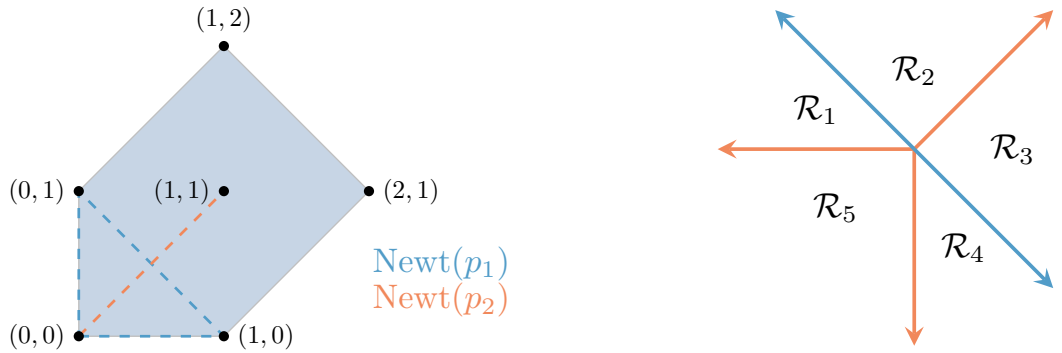


Figure 8. Visualization of Proposition 2 for $p_1(x, y) = \max(x, y, 0)$, $p_2(x, y) = \max(x + y, 0)$. On the left, $\mathrm{Newt}(p_1 \oplus p_2)$ has 5 vertices, equal to the number of open cells formed by $\mathcal{V}(p_1) \cup \mathcal{V}(p_2)$ (corresponding to linear regions of the polynomial $p \equiv p_1 + p_2$), shown on the right.

2) in the case of Maxout activations of rank k, we have

$$\mathcal{N}_m^d \leq \min\left\{k^m, 2\sum_{j=0}^{d}\binom{m\frac{k(k-1)}{2}}{j}\right\}. \tag{55}$$

Similar upper bounds are straightforward to derive for convolutional layers [19] as well as multilayer networks, which are most common in practice. In particular, one can show the following:

Proposition 3 (Theorem 6.3 in [109])**.** *Consider a ReLU network with L layers of size $n_1, \ldots, n_L$ and inputs of dimension d. If $n_\ell \geq d$, $\ell = 1, \ldots, L-1$, the number of linear regions of the network is upper bounded by*

$$\prod_{\ell=1}^{L-1}\sum_{j=0}^{d}\binom{n_\ell}{j}. \tag{56}$$

Example 4. Suppose we are given inputs of dimension $d = 5$. Consider the two following cases:

- *1 hidden layer with $n_1 = 20$*: applying the formula from (54), the number of linear regions generated by this network is at most $\sum_{j=0}^{d}\binom{20}{j} = 21,700$.
- *2 hidden layers with $n_1 = n_2 = 10$*: using Eq. (56), the number of linear regions generated by this network is at most

$$\left(\sum_{j=0}^{d}\binom{10}{j}\right)^2 = 407,044.$$

We see that distributing $m = 20$ hidden units over 2 layers instead of forming a "wide" layer increases the expressiveness of the resulting network by at least one order of magnitude.

Note that naively chaining $L-1$ applications of Corollary 1 gives a strictly worse bound than that of Proposition 3, as the size of the intermediate inputs for each layer can be arbitrarily larger than d. However, since the dimension of the input to the neural network is d, the "effective" dimension of each intermediate output can be at most d as well.

We conclude this section with a discussion of other research directions intimately related to piecewise-linear neural networks and their implications.

a) Lower bounds: Earlier works have provided almost-matching lower bounds for the number of linear regions of a multilayer network. In particular, [80] shows – via a constructive proof – that the number of linear regions of a deep neural network is on the order of $\Omega((d/W)^{(L-1)W} \cdot d^W)$, when each layer consists of W units.

The lower bound is rather existential in nature; it merely exhibits a function with a large number of linear regions representable by a DNN, instead of providing sufficient conditions (as a function of network parameters) for networks to attain this lower bound. Nevertheless, it is another argument in favor of choosing deep vs. shallow architectures for learning, piling on a wealth of existing theoretical and/or empirical evidence; for example, [100] follows a different approach, constructing a "hard" family of functions that are representable by networks of constant width and polynomial depth but cannot be approximated by shallow networks of subexponential width. Subsequent work [86] employs a different measure of expressive power called "trajectory length", which measures changes in the output of a network as its input is varied on a one-dimensional path, to arrive at a similar conclusion.

b) Generative priors in signal recovery: In addition to the depth vs. width discourse, the number of linear regions of neural networks plays an important role in signal recovery with *generative priors*; a motivating example is that of compressed sensing, where one observes a set of measurements

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{\eta},$$

where $\boldsymbol{x}^\star \in \mathbb{R}^d$ is an unknown signal to be recovered, $\boldsymbol{\eta}$ is observation noise, and $\boldsymbol{A}$ is a known *design matrix*, typically consisting of standard Gaussian elements, with far fewer rows than columns. The resulting problem is underdetermined and calls for further assumptions to be placed on $\boldsymbol{x}^\star$ (for a comprehensive review of compressed sensing, see [30]).

The most common assumption in the literature is that $\boldsymbol{x}^\star$ is *sparse*, in which case the information-theoretic requirement for recovery is $k \ll d$ measurements, where k is the number of nonzero entries of $\boldsymbol{x}^\star$. However, known tractable algorithms exhibit a *computational-statistical gap* for certain problems (such as sparse phase retrieval or low-rank matrix recovery), in the sense that their sample complexity scales *quadratically*, instead of *linearly*, in k. To overcome this, researchers have proposed replacing sparsity with a less restrictive assumption;

in particular, that $\boldsymbol{x}^\star$ **lies in the range of a ReLU network** $G : \mathbb{R}^k \to \mathbb{R}^d$; in other words, $\boldsymbol{x}^\star = G(\boldsymbol{z}^\star)$ for some latent vector $\boldsymbol{z}^\star$. This assumption places a so-called *generative prior* [10], [42] on $\boldsymbol{x}^\star$.

Generative priors are known to "close" the statistical-computational gap in several applications of interest. Developing the theory behind this crucially relies on the fact that *the output of a ReLU network lies in the union of linear subspaces*, the number of which is sufficiently bounded for reasonable architectures. Tight upper bounds on the number of linear regions of ReLU networks enable precise statements about the sample complexity of recovery algorithms under a generative prior.

B. Counting linear regions in practice

In this section, we provide a geometric algorithm for approximating the number of linear regions of a neural network layer, after a brief overview of existing approaches.

Mixed-integer formulations: A number of works have used Mixed Integer Programming (MIP) formulations to obtain empirical bounds on the number of linear regions; in [94], the authors showed that deep rectifier networks are mixed-integer representable when the input is restricted to a polytope. Their proof is constructive, and crucially depends on a mixed-integer formulation, summarized below.

Fix i and ℓ to index a neuron i within a layer ℓ, we denote as $\boldsymbol{h}^\ell$ the vector containing the output of the ℓ-th layer and let $\boldsymbol{h}_0 = \boldsymbol{x}$ be the input to the neural network. The MIP from [94] enforces the following constraints $\forall i,\ \ell$:

$$\begin{cases} \boldsymbol{W}_i^\ell \boldsymbol{h}^{\ell-1} + b_i^\ell = h_i^\ell - \bar{h}_i^\ell \\ h_i^\ell \leq M z_i^\ell \\ \bar{h}_i^\ell \leq M(1 - z_i^\ell) \\ \boldsymbol{h}^\ell, \bar{\boldsymbol{h}}^\ell \geq 0 \\ z_i^\ell \in \{0,1\} \end{cases} \tag{57}$$

Let us parse the constraints in (57). First, z_i^ℓ is an indicator that reveals whether neuron i in layer ℓ is active or not. M is an unspecified, sufficiently large constant that enforces h_i^ℓ to be 0 when $z_i = 0$. If h_i^ℓ denotes the output of the neuron, $\bar{h}_i^\ell$ is a complementary "output" which satisfies $\bar{h}_i^\ell = \max(0, -\boldsymbol{W}_i^\ell \boldsymbol{h}^{\ell-1} - b_i^\ell)$. In [94, Theorem 11], it is shown that for a fixed $\boldsymbol{x}$ and as long as $\left|\boldsymbol{W}_i^\ell \boldsymbol{h}^{\ell-1} + b_i^\ell\right| \leq M$, enforcing the constraints in (57) for every neuron returns a feasible solution yielding the output of the rectifier network. Given that result, we can allow $\boldsymbol{x}$ to vary over the input domain $\mathcal{X}$ and enumerate the integer solutions $\boldsymbol{z}$ of the following MIP:

$$\begin{aligned} \text{Maximize } & f \\ \text{s.t. } & (57) \text{ holds },\ \forall i, \ell \\ & f \leq h_i^\ell + (1 - z_i^\ell) M,\ \forall i, \ell \\ & \boldsymbol{x} \in \mathcal{X} \end{aligned} \tag{$\mathcal{P}$}$$

However, enumerating solutions of a mixed-integer program can be computationally intractable. To address this issue, a probabilistic algorithm was proposed in [93] to produce lower bounds to the number of possible solutions.

Enumeration via reverse search: The MIP-based approach above makes the simplifying assumption that the input domain is bounded, which helps determine a lower bound for M so that ($\mathcal{P}$) is a valid formulation. Even though [94] discusses the issue of unbounded input domains, it tends to complicate algorithm design. In contrast, treating the enumeration problem from the scope of Newton Polytope vertices applies to general domains. It is known that extreme points of Minkowski sums of polytopes are sums of extreme points of the individual polytopes; moreover, enumerating vertices of Minkowski sums of polytopes in vertex representation is possible via the so-called *reverse search* method [5], [33].

The resulting algorithm for vertex enumeration has runtime

$$\mathcal{O}\left(\delta \cdot \mathrm{LP}(d, \delta) \cdot N\right), \quad \delta := \sum_{i=1}^{m} \delta_i,$$

where N is the number of vertices, $P_i \subset \mathbb{R}^d$, δ_i denotes the maximal degree of the vertex adjacency graph of P_i and $\mathrm{LP}(d, \delta)$ denotes the complexity of solving an LP in d variables and δ constraints. The above implies straightforward bounds for exact counting of linear regions of ReLU/maxout layers. For ReLUs, $\delta_i = 2,\ \forall i$, so $\delta = 2m$. In the latter case, denoting k_i for the rank of the i-th unit, $\delta = \sum_i k_i$.

Unfortunately, reverse search requires solving a prohibitive number of LPs, rendering the above approach impractical. We attack this problem from a different angle, by considering the "dual" problem of counting vertices of convex polytopes by sampling.

C. A geometric algorithm

We present a randomized method 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. $\boldsymbol{g}_k \sim_{\text{i.i.d}} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ and compute $(\boldsymbol{g}_k)^T \boldsymbol{v}_i,\ \forall$ extreme points $\boldsymbol{v}_i$. We record the minimizers/maximizers for each polytope P_j and repeat the trial. Denoting by $\boldsymbol{V}_i \in \mathbb{R}^{k_i \times d}$ the matrix whose rows contain the coordinates of each vertex of P_i, the above procedure essentially counts the number of unique tuples giving the row indices of the extrema of $\boldsymbol{V}_i \boldsymbol{g}_k,\ \forall i$.

From our discussion motivating the use of the reverse search method, it is clear that the resulting number is a lower bound on the number of vertices in the Minkowski sum. The resulting Algorithm 1 leverages the techniques in [28]. This method and its specialization to upper hulls work for *general* polytopes, while the MIP-based methods in the literature are only presented for rectifier networks. Adapting Alg. 1 for counting vertices in upper hulls is described in [19, Section 4.1].

Algorithm 1 provides a nontrivial lower bound to the number of extreme points of the resulting Minkowski sum with high probability, as Proposition 4 shows.

Proposition 4. *Let N denote the number of vertices of $P = P_1 \oplus \cdots \oplus P_m$, a failure probability δ, and define*

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

Algorithm 1 Sampling points in the convex hull

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

where $\omega(N_P(\boldsymbol{v}_k))$ is the solid angle of the normal cone of the k-th vertex, $N_P(\boldsymbol{v}_k)$. Then, for $K \geq \tilde{N} \log(N/\delta)$ in Algorithm 1, the algorithm records all the vertices with probability at least $1 - \delta$.

Proof sketch. The key idea in the proof is the following: extreme points of Minkowski sums are also extreme points of individual summands. Consequently, missing a "configuration" of minimizers across our trials is equivalent to missing an extreme point $\boldsymbol{v}$ of the Minkowski sum.

Moreover, it is not hard to see (see e.g. [19, Corollary 1]) that the solid angles of the normal cones of the vertices of a polytope P form a probability distribution, with

$$\omega(N_P(\boldsymbol{v}_k)) = \mathbb{P}_{\boldsymbol{g} \sim \mathcal{N}(0, \boldsymbol{I}_d)} \left(\boldsymbol{g} \in N_P(\boldsymbol{v}_k) \right), \tag{58}$$

the probability that $\boldsymbol{g}$ is in the normal cone at $\boldsymbol{v}_k$ and, consequently, $\boldsymbol{v}_k$ being the minimizer of the linear function $\boldsymbol{x} \mapsto \boldsymbol{x}^T \boldsymbol{g}$. The rest follows from a coupon collector-style argument; a detailed proof is available in [19, Section 4]. □

Example 5. Suppose that P has all-equal solid angles, i.e. $\omega(N_P(\boldsymbol{v}_k)) = \frac{1}{N}$, $\forall k$, in which case $\tilde{N} = \log\left(\frac{N}{N-2}\right)^{-1}$. Rewriting $\log \frac{N}{N-2} = \log\left(1 + \frac{2}{N-2}\right)$ and combining with the inequality $\log(1+x) \leq x$, we see that

$$\tilde{N} \geq \frac{N-2}{2} \Rightarrow K \geq \left(\frac{N}{2} - 1 \right) \log(N/\delta)$$

is necessary to achieve probability failure at most δ. Note that this shows that Algorithm 1 will require at least this many samples for **any** polytope P; indeed, it is easy to see that $\min_k \omega(N_P(\boldsymbol{v}_k)) \leq \frac{1}{N}$ for any polytope with N vertices.

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. Figure 9 illustrates (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". Nevertheless, the proposed algorithm can be easily parallelized, only relies on computing inner products, and crucially utilizes the geometric insights from the Newton polytope characterization of neural network layers.

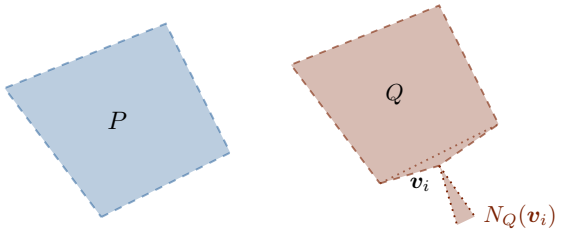


Figure 9. Polytopes P, Q and their solid angles. All the solid angles of P (**left**) are bounded away from zero. On the other hand, for Q (**right**), we have $\omega(N_Q(\boldsymbol{v}_i)) \ll 1$.

D. Other connections between tropical geometry and neural networks

a) Tropical polynomial division and network simplification: Another problem where tropical geometry can be of use is neural network minimization; as neural networks increase in complexity, so do their needs in computing time and memory, limiting their use in time-sensitive applications. Therefore, we seek to reduce the size of a neural network while maintaining its accuracy. Several methods have attempted to solve this problem, by removing either connections between neurons [41] or neurons themselves [45], [63] from the network. The former is referred to as *weight* or *unstructured pruning* and the latter as *channel/neuron* or *structured pruning*. These studies show that minimal drops in accuracy (roughly 1% on the VGG-16 architecture) are possible, despite significant decrease in network complexity.

Tropical geometry can also provide novel methods for neural network simplification, such as the approaches described below:

- given a fixed ReLU network, we can attempt to construct a smaller neural network whose Newton polytopes closely approximate the polytopes of the original network. The resulting algorithm is *constructive* and relies on the concept of *tropical polynomial division* [96], which approximates the divident using the Newton polytopes of the divisor and quotient. Since this method constructs a network from scratch, it can be much faster than pruning methods in practice. It was originally applied to minimize the second-to-last layer of networks with a single output neuron in context of binary classification problems, with less than $.5\%$ loss in accuracy even when only 1% of the hidden units are retained. Extensions to multiclass problems are considered in [97].
- a complementary approach appeared in [3]. The authors first obtain a tropical geometric characterization of the decision boundaries of neural networks using their Newton polytopes; following that, they present a regularization method that balances a sparsity-inducing penalty with an objective that attempts to preserve the decision boundaries of the neural network. In contrast to the previous two approaches, this is a pruning method.

b) Morphological neural networks: Though feedforward networks with piecewise-linear activations have become the de-facto standard in neural computation, the paradigm of so-called *morphological computation* is also closely related to tropical geometry. In morphological computation, linear operations are replaced with their "tropical" versions; thus the building blocks of a morphological neural network are replaced by *dilations* and *erosions* instead of linear operations. In its most elementary version, a morphological $(\max, +)$-

perceptron computes the function

$$\boldsymbol{x} \mapsto \boldsymbol{w}^T \boxplus \boldsymbol{x} = \max_i \{w_i + x_i\}, \tag{59}$$

where $\boldsymbol{w} \in \overline{\mathbb{R}}^d$ is a set of trainable weights. In binary classification, the decision regions induced by a $(\max, +)$ perceptron are collections of so-called *tropical halfspaces*. A $(\max, +)$-perceptron can separate two classes if and only if a certain *tropical polyhedron* is nonempty, a condition that can be checked efficiently for this particular case.

Proposition 5 (Proposition 1 in [18])**.** *Consider N_1 points from class $\mathcal{C}_1$ and N_2 points from class $\mathcal{C}_2$, forming the matrices $\boldsymbol{X}_1 \in \mathbb{R}^{N_1 \times d}$, $\boldsymbol{X}_2 \in \mathbb{R}^{N_2 \times d}$. Then these points can be separated by a morphological perceptron of the form* (59) *if and only if*

$$\left\{\boldsymbol{w} \in \overline{\mathbb{R}}^d : \boldsymbol{X}_1 \boxplus \boldsymbol{w} \geq \boldsymbol{0}_{N_1},\ \boldsymbol{X}_2 \boxplus \boldsymbol{w} \leq \boldsymbol{0}_{N_2}\right\} \neq \emptyset \\ \Leftrightarrow \boldsymbol{X}_1 \boxplus (\boldsymbol{X}_2{}^* \boxplus' \boldsymbol{0}) \geq \boldsymbol{0}. \tag{60}$$

An example of a tropically separable configuration of points is shown in Fig. 10. Even though the morphological paradigm

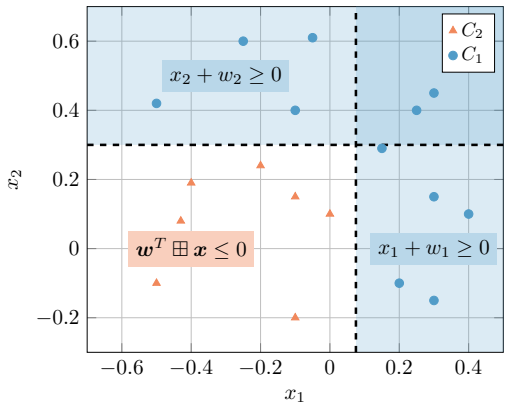


Figure 10. Example of tropically separable patterns in $\mathbb{R}^2$, using the weight vector $\boldsymbol{w} = \begin{pmatrix}0.075 & 0.3\end{pmatrix}$.

dates back almost 3 decades [87], [88], [98], [108], a recent resurgence of interest has led to new developments; for example, it was recently shown [79], [110] that a morphological neural network with a hidden layer consisting of dilations and erosions followed by a linear layer is a universal approximator. In a more recent publication [32] the authors focus on deep learning for image processing, treating all nonlinear operations (e.g. max-pooling) as trainable morphological operators to complement trainable convolutional operations, and achieve competitive results in tasks such as boundary detection using considerably fewer parameters than other architectures.

VI. Tropical Geometry and Graphical Models

A. Hidden Markov Models

The use of tropical geometry within the framework of parametric statistics was pioneered by Pachter & Sturmfels [82]. Specifically, they consider graphical models, which are formally represented by directed acyclic graphs with two sets of vertices, the *hidden variables* $\boldsymbol{X} = (X_1, \ldots, X_m)$ and the *observed variables* $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$. Moreover, we denote $s_1, \ldots, s_d$ for the *model parameters*. Given an observation $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)$ the observation probabilities are polynomials of degree E in the model parameters, where E is the number of edges of the aforementioned graph. We denote $f_{\boldsymbol{\sigma}}(s_1, \ldots, s_d) = \mathbb{P}_{s_1,\ldots,s_d}(\boldsymbol{Y} = \boldsymbol{\sigma})$ for the observation probability. The authors in [82] ask a fundamental question about this family of models:

How do the solutions to inference problems depend on the model parameters?

The authors fix the numbers d, n of model parameters and observations, and furthermore assume that each of the observed variables can take ℓ different values. Mathematically, this model is a polynomial map $f : \mathbb{R}^d \to \mathbb{R}^{\ell^n}$, each of the coordinates being one of the aforementioned polynomials of degree E. Let $u_i := -\log(s_i)$ determine the associated logarithmic parameter space. Moreover, define

$$g_{\boldsymbol{\sigma}}(u_1, \ldots, u_d) := -\max_{\boldsymbol{h}} \log\left(\mathbb{P}_{s_1,\ldots,s_d}(\boldsymbol{X} = \boldsymbol{h} \mid \boldsymbol{Y} = \boldsymbol{\sigma})\right). \tag{61}$$

The authors in [82] show that $g_{\boldsymbol{\sigma}}$ is piecewise linear and concave on the logarithmic parameter space, with the normal cones of $\mathrm{Newt}(f_{\boldsymbol{\sigma}})$ identifying its domains of linearity. As the parameters $u_1, \ldots, u_d$ vary, they define inference functions $\boldsymbol{\sigma} \mapsto \hat{\boldsymbol{h}}$, where $\hat{\boldsymbol{h}}$ is the most likely tuple of hidden variables given an observation $\boldsymbol{\sigma}$. This leads to the following:

Proposition 6 (Proposition 6 in [82])**.** *The inference functions $\boldsymbol{\sigma} \mapsto \hat{\boldsymbol{h}}$ of a graphical model f are in bijection with the vertices of the Newton polytope of the map f. The explanations $\hat{\boldsymbol{h}}$ for a fixed observation $\boldsymbol{\sigma}$ in a graphical model are in bijection with the vertices of the Newton polytope of the polynomial $f_{\boldsymbol{\sigma}}$.*

This is the main ingredient in [82], which the authors employ to deduce upper bounds on the number of inference functions and explanations of graphical models, by leveraging known bounds on the number of vertices of Newton polytopes. Finally, they motivate theoretically the use of the so-called *polytope propagation* algorithm to enumerate the vertices of the aforementioned polytopes, including an application to inference for biological sequence analysis [83].

The authors of a later publication [24] study the *Restricted Boltzmann Machine (RBM)*, a graphical model that is the building block of deep belief networks [50], using techniques from algebraic and tropical geometry. Formally, RBMs are represented by a bipartite graph on hidden variables $\boldsymbol{h} \in \{0,1\}^k$ and observed variables $\boldsymbol{v} \in \{0,1\}^n$, with "activation"

$$\psi(\boldsymbol{v}, \boldsymbol{h}) := \exp\left(\boldsymbol{h}^T \boldsymbol{W} \boldsymbol{v} + \boldsymbol{b}^T \boldsymbol{v} + \boldsymbol{c}^T \boldsymbol{h}\right), \tag{62}$$

that determines a probability distribution

$$p(\boldsymbol{v}) := \frac{1}{Z} \sum_{\boldsymbol{h} \in \{0,1\}^k} \psi(\boldsymbol{v}, \boldsymbol{h}), \quad Z := \sum_{\boldsymbol{v}, \boldsymbol{h}} \psi(\boldsymbol{v}, \boldsymbol{h}). \tag{63}$$

Here, Z is the induced *log-partition function*. The authors then define the **tropical RBM** model by applying the Maslov

dequantization principle to $\log p(\boldsymbol{v})$, leading to the piecewise-linear convex model in (64):

$$q(\boldsymbol{v}) := \max\left\{\boldsymbol{h}^T\boldsymbol{W}\boldsymbol{v} + \boldsymbol{b}^T\boldsymbol{v} + \boldsymbol{c}^T\boldsymbol{h} : \boldsymbol{h} \in \{0,1\}^k\right\}. \quad (64)$$

Similarly to [82], varying the parameters $(\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{W})$ determines a collection of inference functions. The authors in [24] then obtain the following characterization of an RBM's inference functions (recall that a *linear threshold function* is a function of the form $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{\alpha}^T\boldsymbol{x} + \beta)$):

Proposition 7 (Proposition 5.1 in [24])**.** *The inference functions for the RBM in k hidden and n observed variables are precisely those Boolean functions $\{0,1\}^n \to \{0,1\}^k$ for which each of the coordinate functions is a linear threshold function.*

B. Tropical Algorithms on WFSTs

1) Introduction: Weighted Finite State Transducers (WFSTs) introduce a computational framework that extends traditional automata, with applications in automatic speech recognition, natural language rocessing, computational biology, and more. The workhorse of the framework is the Viterbi algorithm; a decoding procedure that performs inference over graphs. The framework also includes a variety of algorithms aiming to reduce the computational footprint, which can be split into two categories; (i) algorithms that respect the initial topology of the network, refactoring the weights or removing extraneous transitions, and (ii) algorithms that fundamentally alter the structure of the network, via network minimization or composition. In any case, WFSTs, complete with their suit of diverse algorithms, present a formal mathematical framework whose properties have been analyzed for decades. A simple WSFT is shown in Fig. 11.

WFST algorithms historically employed tropical arithmetic [77], [78] for practical reasons. However, their formal modeling using tropical matrix algebra was only recently explored. A recent work [101] tropicalized the Viterbi algorithm[6] and its pruning variant, both seminal communications algorithms, by expressing the symbol observation probabilities as a tropical diagonal matrix. A following work [102] extended the tropicalization to other instrumental WFST algorithms, namely *epsilon removal* and *weight pushing*, via the strong and weak transitive closures of the network.

In addition, a tropical analogue to spectral graph theory can be found, which studies the existence and characterization of solutions to the tropical (sub)eigenvalue problem. While mathematically analyzing WFSTs, certain elements from tropical spectral theory arise, which are introduced in the next section.

2) Background: A WFST is mainly characterized by the *transition matrix* of a network, which we denote $\boldsymbol{A} \in \mathbb{R}_{\min}^{d\times d}$, and where each entry a_{ij} corresponds to the cost of transitioning from state i to state j. The initial states are denoted by $\boldsymbol{\pi} \in \mathbb{R}_{\min}^{d}$, where each initial state has a finite cost, and ∞ otherwise. Similarly, emitting (or final) states are denoted by $\boldsymbol{\rho} \in \mathbb{R}_{\min}^{d}$ and have also finite costs (and ∞ if they are not final states). We define the *weak transitive closure* of $\boldsymbol{A}$ as

$$\Gamma(\boldsymbol{A}) \;:=\; \boldsymbol{A} \wedge \boldsymbol{A}^2 \wedge \ldots \wedge \boldsymbol{A}^d \wedge \ldots, \quad (65)$$

and the *strong transitive closure* as

$$\Delta(\boldsymbol{A}) \;:=\; \boldsymbol{I} \wedge \boldsymbol{A} \wedge \boldsymbol{A}^2 \wedge \ldots \wedge \boldsymbol{A}^d \wedge \ldots, \quad (66)$$

where $\boldsymbol{A}^k = \overbrace{\boldsymbol{A} \boxplus' \ldots \boxplus' \boldsymbol{A}}^{k \text{ times}}$. The *minimum cycle mean* of $\boldsymbol{A}$ is defined as

$$\lambda(\boldsymbol{A}) = \min_{c\in C(\boldsymbol{A})} \frac{\text{weight}(c)}{\text{length}(c)},$$

where $C(\boldsymbol{A})$ is the set of cycles of the network, and $\text{weight}(\cdot)$ and $\text{length}(\cdot)$ denote the weight (sum of the costs along the cycle) and length of a cycle, respectively.

In *tropical spectral analysis*, the $\min$-plus eigenproblem of $\boldsymbol{A}$ consists of finding the *eigenvalues* λ and *eigenvectors* $\boldsymbol{u}$ such that

$$\boldsymbol{A} \boxplus' \boldsymbol{u} = \boldsymbol{u} + \lambda. \quad (67)$$

The minimum cycle mean $\lambda(\boldsymbol{A})$ plays a fundamental role in the $\min$-plus eigenproblem; indeed, it is the *smallest* eigenvalue, and the only one whose eigenvectors may be *finite* [70]. For the spectral analysis component, we will heavily rely on the following theorem, which characterizes the *subeigenvectors* of $\boldsymbol{A}$.

Theorem 6 (Theorem 1.6.18 in [15])**.** *Suppose $\boldsymbol{A}$ has at least one finite entry. If $\lambda \le \lambda(\boldsymbol{A})$ and $\lambda < \infty$, then*

(a) $\boldsymbol{A} \boxplus' \boldsymbol{x} \ge \lambda + \boldsymbol{x}$ has a finite solution.
(b) The solution set is

$$V^*(\boldsymbol{A}, \lambda) = \left\{\Delta(\boldsymbol{A} - \lambda) \boxplus' \boldsymbol{u} : \boldsymbol{u} \in \mathbb{R}_{\min}^{d}\right\}. \quad (68)$$

(c) $\boldsymbol{A} \boxplus' \boldsymbol{x} \ge \lambda + \boldsymbol{x}$ only holds if $\boldsymbol{x} = \Delta(\boldsymbol{A} - \lambda) \boxplus' \boldsymbol{u}$.

3) Tropicalization of WFST algorithms: The *Viterbi algorithm* was the first WFST algorithm that was formally tropicalized [101]. In short, the algorithm, stemming from the field of communications, attempts to decode the most probable series of latent states from a data sequence. Given an observation σ_t, observation probabilities $\boldsymbol{b}(\sigma_t)$, a transition

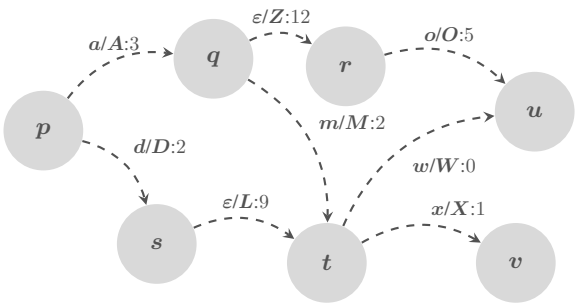


Figure 11. A toy WFST. Transitions are of the form $i/o{:}c$, where i is the input symbol, o is the output symbol, and c is the cost of the transition. For example, the input sequence `dx` would be decoded to `DLX` with a total cost of 12.

[6]The framework of weighted lattices allows to analyze the max-product form of the Viterbi algorithm as a nonlinear dynamical system in state-space and extend it to more general forms that accept control inputs [70].

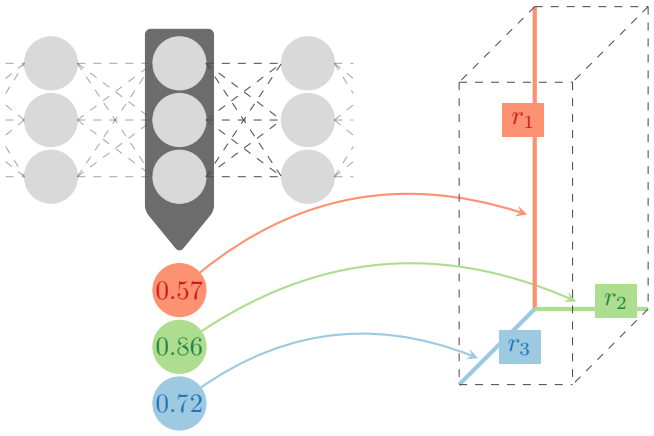


Figure 12. At each decoding step $\boldsymbol{x}(t)$ and $\boldsymbol{\eta}$ of (73) define a polytope. The vector $\boldsymbol{r} = \boldsymbol{\eta} - \boldsymbol{x}(t)$ denotes the *range* of each dimension (negative ranges indicate that the index is pruned).

matrix $\boldsymbol{W}$, and the previous state $\boldsymbol{q}(t-1)$, the maximum probability for each state is given by

$$q_i(t) = \max_j \; b_i(\sigma_t) W_{ji} q_j(t-1), \tag{69}$$

for each state i and for every observation in the sequence $\{t\}_{t=1}^{T}$. We can formally tropicalize (69) and provide a closed form solution for the state vector $\boldsymbol{x}(t)$:

$$\boldsymbol{x}(t) = \boldsymbol{P}(\sigma_t) \boxplus' \boldsymbol{A}^T \boxplus' \boldsymbol{x}(t-1) \tag{70}$$

where $\boldsymbol{x}(t) = -\log \boldsymbol{q}(t)$, $\boldsymbol{A} = -\log \boldsymbol{W}$, and $\boldsymbol{P}(\sigma_t) = \operatorname{diag}(-\log \boldsymbol{b}(\sigma_t))$ with $\operatorname{diag}(\cdot)$ denotes a matrix with the argument in the diagonal and ∞ elsewhere.

Viterbi pruning is a practical technique that is frequently used in order to reduce the computational burden of the decoding. In essence, the *optimal* path is computed at each step, and only the paths whose cost is worse up to a certain threshold are further expanded. An intuitive example is given in Fig. 13. Viterbi pruning can be thought as the problem

$$\boldsymbol{X}(t) \boxplus' \boldsymbol{y} \geq \boldsymbol{\eta}, \tag{71}$$

where $\boldsymbol{X}(t) = \operatorname{diag}(\boldsymbol{x}(t))$ and $\boldsymbol{\eta}$ is a vector with $\eta_i = \frac{1}{2}\left(\boldsymbol{x}(t)^T \boxplus' \boldsymbol{x}(t)\right) + \theta$, where θ is the pruning parameter. We can then interpret pruning as finding the smallest solution $\overline{\boldsymbol{y}} \in \mathbb{R}_{\min}^d$ satisfying the min-plus inequality (71), which can be done using the dual of Theorem 2:

$$\overline{\boldsymbol{y}} = \boldsymbol{X}^*(t) \boxplus \boldsymbol{\eta}, \tag{72}$$

where $\boldsymbol{X}^*(t) = -\boldsymbol{X}^T(t)$ and the negative entries of $\overline{\boldsymbol{y}}$ indicate the indices to be pruned. A geometrical interpretation can be given to the Viterbi pruning; in particular, the set of feasible solutions at each step is a *tropical polytope* (see also Fig. 12)

$$T(\boldsymbol{G}, \boldsymbol{H}) = \{\boldsymbol{z} \in \mathbb{R}_{\min}^d : \boldsymbol{z} \geq \boldsymbol{x}(t), \boldsymbol{z} \leq \boldsymbol{\eta}\}. \tag{73}$$

Example 6. Let the state vector be

$$\boldsymbol{x}(t) = \begin{bmatrix} 1 & 7 & 4 \end{bmatrix}^T,$$

and suppose that the pruning parameter is $\theta = 5$. Then, $\eta_i = \frac{1}{2}\left(\boldsymbol{x}(t)^T \boxplus' \boldsymbol{x}(t)\right) + \theta = 6$. The optimal solution then is given by (72)

$$\overline{\boldsymbol{y}} = \begin{bmatrix} -1 & -\infty & -\infty \\ -\infty & -7 & -\infty \\ -\infty & -\infty & -4 \end{bmatrix} \boxplus \begin{bmatrix} 6 \\ 6 \\ 6 \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ 2 \end{bmatrix}.$$

As $\overline{y}_2$ is negative it gets pruned and the resulting vector is

$$\boldsymbol{x}_{\mathrm{p}}(t) = \begin{bmatrix} 1 & \infty & 4 \end{bmatrix}^T.$$

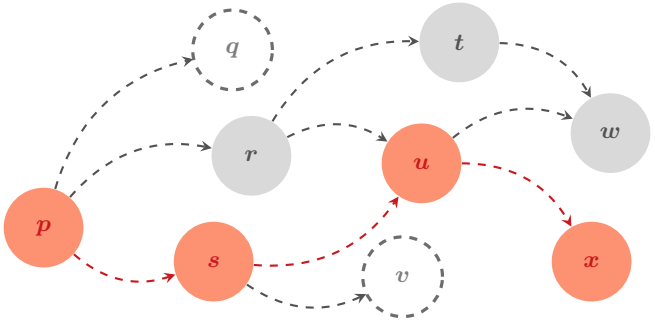


Figure 13. An illustration of the *Viterbi pruning*. The path of optimal states is denoted by red. States colored gray were examined by the algorithm for optimality, whereas the *dashed* states had high costs and were pruned.

The *weight pushing algorithm* is an essential component of the WFST framework [78]. The algorithm improves the effectivenes of the Viterbi pruning by *pushing* weights towards earlier transitions and states, without alterning the overall path statistics (i.e. the decoded sequences and their probabilities). After weight pushing, low-probability sequences can be identified and pruned early during decoding, increasing efficiency.

Integral to the weight pushing algorithm is the computation of a *potential* for each state of the graph. In short, the potential value is the weight amount that can be "pushed" to earlier states, and can be computed via an iterative evaluation. A single iteration of the potential vector can be expressed as [102]

$$\boldsymbol{v}_{i+1} = \boldsymbol{v}_i \wedge \boldsymbol{A} \boxplus' \boldsymbol{v}_i, \tag{74}$$

with $\boldsymbol{v}_0 = \boldsymbol{\rho}$ being the emission vector. Recursively iterating (74), we arrive at the final potential vector

$$\boldsymbol{v}_\infty = \boldsymbol{\rho} \wedge \boldsymbol{A} \boxplus' \boldsymbol{\rho} \wedge \boldsymbol{A}^2 \boxplus' \boldsymbol{\rho} \wedge \ldots \wedge \boldsymbol{A}^n \boxplus' \boldsymbol{\rho} \wedge \ldots = \Delta(\boldsymbol{A}) \boxplus' \boldsymbol{\rho}, \tag{75}$$

where the computation of $\Delta(\boldsymbol{A})$ is finite under very mild assumptions; namely that the graph does not contain cycles of negative weight, and thus $\lambda(\boldsymbol{A}) \geq 0$ (a standard assumption for WFSTs). Henceforth, we assume that these conditions hold. After the potential computation, the network parameters can be updated via the rules

$$\boldsymbol{\pi}' = \boldsymbol{\pi} + \boldsymbol{v}_\infty, \quad \boldsymbol{\rho}' = \boldsymbol{\rho} - \boldsymbol{v}_\infty, \quad \boldsymbol{A}' = \boldsymbol{V}^- \boxplus' \boldsymbol{A} \boxplus' \boldsymbol{V}^+, \tag{76}$$

where $\boldsymbol{V}^+ = \operatorname{diag}(\boldsymbol{v}_\infty)$ and $\boldsymbol{V}^- = \operatorname{diag}(-\boldsymbol{v}_\infty)$.

Another instrumental algorithm to the WFST framework is *epsilon removal* [78]. Similar to weight pushing, this algorithm facilitates decoding by decreasing the size of the network by removing extraneous transitions and states. Examples of these

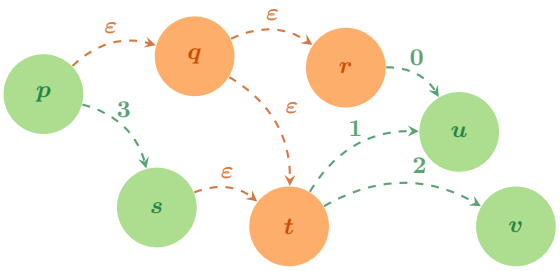


Figure 14. An illustration of *epsilon transtitions* and *states*. orange denotes states and transitions that will be removed by the epsilon removal algorithm, whereas surviving ones are denoted by green.

states and transitions can be seen in Fig. 14. The removal of extraneous transitions and states is achieved through the computation of the *epsilon closure* of every state, which encapsulates the states that are reachable using only epsilon transitions. To that end, the network matrix $\boldsymbol{A}$ can be decomposed [102] into two components

$$\boldsymbol{A} = \boldsymbol{A}_{\varepsilon} \wedge \boldsymbol{A}_{\varepsilon^{\perp}}, \tag{77}$$

where $\boldsymbol{A}_{\varepsilon}$ contains only the epsilon transitions and $\boldsymbol{A}_{\varepsilon^{\perp}}$ contains the non-epsilon transitions. The epsilon closure is then computed as the shortest distances of the network matrix $\boldsymbol{A}_{\varepsilon}$, which, via the definition of the weak transitive closure, is given by $\Gamma(\boldsymbol{A}_{\varepsilon})$, where the computation is finite and equal to $\Gamma(\boldsymbol{A}_{\varepsilon}) = \boldsymbol{A}_{\varepsilon} \wedge \ldots \wedge \boldsymbol{A}_{\varepsilon}^{d}$. Having computed the epsilon closure, the updated network parameters take the form

$$\begin{aligned} \boldsymbol{A}' &= \boldsymbol{A}_{\varepsilon^{\perp}} \wedge (\Gamma(\boldsymbol{A}_{\varepsilon}) \boxplus' \boldsymbol{A}_{\varepsilon^{\perp}}) = \Delta(\boldsymbol{A}_{\varepsilon}) \boxplus' \boldsymbol{A}_{\varepsilon^{\perp}}, \\ \boldsymbol{\rho}' &= \boldsymbol{\rho} \wedge (\Gamma(\boldsymbol{A}_{\varepsilon}) \boxplus' \boldsymbol{\rho}) = \Delta(\boldsymbol{A}_{\varepsilon}) \boxplus' \boldsymbol{\rho}. \end{aligned} \tag{78}$$

4) Spectral analysis of tropical WFST algorithms: The representation we developed in the previous sections offers a unified computational framework that enables a holistic analysis of the WFSTs; in certain cases, it also enables a geometrical characterization of the algorithms via elements of algebraic geometry, such as polytopes. Herein, the computational framework of tropical algebra further enables the *spectral characterization* of the graph algorithms; i.e. we are able to characterize the introduced algorithms via their *subeigenvalues*. This characterization introduces a new dimension to these algorithms, as we are now able to examine their properties for different subeigenvalues.

We established that a mild (and realistic) assumption for the class of networks is that the cycles of the network have nonnegative weights, and therefore $\lambda(\boldsymbol{A}) \geq 0$. Having made this remark, we can view (75) in the scope of Theorem 6

$$\boldsymbol{v}_{\infty} = \Delta(\boldsymbol{A}) \boxplus' \boldsymbol{\rho} = \Delta(\boldsymbol{A} - \lambda) \boxplus' \boldsymbol{u}, \tag{79}$$

with $\boldsymbol{u} = \boldsymbol{\rho}$ and $\lambda = 0$. Thus, $\boldsymbol{v}_{\infty}$ is a tropical eigenvector of $\boldsymbol{A}$ for the tropical eigenvalue 0. Similarly, we can revisit (76), and express the updated network parameters as

$$\begin{aligned} \boldsymbol{\rho}' &= \Delta(\boldsymbol{A}_{\varepsilon}) \boxplus' \boldsymbol{\rho} = \Delta(\boldsymbol{A}_{\varepsilon} - \lambda) \boxplus' \boldsymbol{u}, \\ \boldsymbol{A}' &= \Delta(\boldsymbol{A}_{\varepsilon}) \boxplus' \boldsymbol{A}_{\varepsilon^{\perp}} = \Delta(\boldsymbol{A}_{\varepsilon} - \lambda) \boxplus' \boldsymbol{U}, \end{aligned} \tag{80}$$

where, again, $\lambda = 0$ and $\boldsymbol{u} = \boldsymbol{\rho}$ and $\boldsymbol{U} = \boldsymbol{A}_{\varepsilon^{\perp}}$, respectively. Note that $\boldsymbol{\rho}' = \Delta(\boldsymbol{A}_{\varepsilon} - \lambda) \boxplus' \boldsymbol{u}$ is similar to (79), it simply refers to the subeigenproblem of $\boldsymbol{A}_{\varepsilon}$. The second equation of (80) consists of a *collection* of tropical subeigenvectors of $\boldsymbol{A}_{\varepsilon}$.

From this analysis, we make two remarks; first, $\boldsymbol{A}'$ of (76) is, by definition, *visualized* [15], meaning that it has a simpler structure than $\boldsymbol{A}$, while still maintaining the same spectral properties. As a second remark, we note that tropical subeigenvalue problems have *infinite* solutions. Indeed, it is a well known fact in tropical algebra [15] that subeigenvectors exist for each subeigenvalue $\lambda \in [0, \lambda(\boldsymbol{A})]$. Therefore, this creates a whole family of WFSTs that all solve the same subeigenvalue problem, for all λ in the aforementioned range.

VII. Tropical Regression

Herein we expand on our previous work [72] and apply tropical geometry and max-plus algebra to a fundamental regression problem of approximating the shape of curves and surfaces by fitting *piecewise-linear (PWL)* functions, represented by tropical polynomials (11), to data possibly sampled from a functional form and in the presence of noise. We begin with a brief sampling of PWL models.

A. PWL Function Representation and Data Fitting

PWL functions $f : \mathbb{R}^d \to \mathbb{R}$ are defined as follows: (i) their domain is divided into a finite number of polyhedral regions separated by linear $(n-1)$-dimensional boundaries that are hyperplanes or subsets of hyperplanes; (ii) they are affine over each region and continuous on each boundary. In using them for regression, two major problems are *representation*, i.e. finding better analytical expressions to represent them, and their *parameter estimation* for modeling a nonlinear system or fitting some data. Further, while these problems are well-explored in the 1D case, they remain relatively underdeveloped for multi-dimensional data.

The so-called *canonical representation* for continuous PWL functions, consisting of an affine function plus a weighted sum of absolute-value affine functions, has been extensively studied and applied to nonlinear circuit analysis and modeling [21], [40], [57], [59]. However, it is complete only for 1D PWL functions. In higher dimensions it needs multi-level nestings of the absolute-value functions [56], [57]. The *lattice representation*, developed in [99], is a constructive way to generate min-max combinations of affine functions that provide a complete representation of continuous PWL function in arbitrary dimensions. Combining the canonical with the lattice representations in [107] involved producing an equivalent representation as a difference of two convex max-affine functions.

A more recent approach is to focus on the class of *convex* PWL functions represented by a *maximum of affine functions*, which are essentially max-plus topical polynomials as in (11), and use them for data fitting. Starting from early least-squares solutions [49], [52], some representative recent approaches to solve this *convex regression* problem include [43], [44], [51], [58], [66]. In all these approaches, there is an iteration that alternates between partitioning the data domain and locally fitting affine functions (using least-squares or some linear optimization procedure) to update the local coefficients. For

a known partition the convex PWL function is formed as the max of the local affine fits. Then, a PWL function generates a new partition which can be used to refit the affine functions and improve the estimate. As explained in [66], this iteration can be viewed as a Gauss-Newton algorithm to solve the above nonlinear least-squares problem. The rank K of the model can be increased until some error threshold is reached. Interesting and promising generalizations of the above max-affine representation for convex functions include works that use softmax instead of max, via the *log-sum-exp* models for convex and log-log convex data [16], [17], [51]. Other iterative approaches for convex PWL data fitting include [103]. Closer to our work is [53] which however solves max-plus equations using least squares and assumes that the slope parameters $\boldsymbol{a}_k$ in (11) are known. Reaching a local minimum of the ℓ_2 error norm for approximately solving max-plus equations was approached in [53] both via steepest descent (which was found computationally infeasible for large problems) and via Newton's method with undershooting (which could not guarantee convergence to a local minimum). Very recently, [36] showed that, under certain assumptions, a carefully initialized alternating minimization algorithm converges linearly for max-affine regression. Finally, [20] demonstrates how to efficiently solve large scale convex regression – albeit with an unconstrained number of affine pieces. For additional references, we refer the reader to the bibliography in the above works.

Next, we focus on convex PWL regression via the max-affine model, which has a tropical interpretation, and propose a direct *non-iterative* and *low-complexity* approach to estimate its parameters by using the optimal solutions of max-plus equations of Sec. IV-A.

B. Optimal Fitting Max-plus Tropical Lines and Planes

Given data $(x_i, y_i) \in \mathbb{R}^2$, $i = 1, ..., N$, if we wish to fit a Euclidean line $y = ax + b$ by minimizing the ℓ_2 error norm $\|\boldsymbol{y} - a\boldsymbol{x} - b\|_2$ where $\boldsymbol{y} = [y_i]$ and $\boldsymbol{x} = [x_i]$, the optimal solution (*least squares estimate - LSE*) for the parameters a, b is

$$\begin{aligned} \hat{a}_{\text{LS}} &= \frac{N\sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{N\sum_i (x_i)^2 - (\sum_i x_i)^2}, \\ \hat{b}_{\text{LS}} &= \frac{\sum_i (y_i - \hat{a}_{\text{LS}} x_i)}{N} \end{aligned} \tag{81}$$

Suppose now we wish to fit a max-plus tropical line $p(x) = \max(a + x, b)$ by minimizing some ℓ_p error norm. The equations to solve for finding the optimal parameter vector $\boldsymbol{w} = [a, b]^T$ become:

$$\underbrace{\begin{bmatrix} x_1 & 0 \\ \vdots & \vdots \\ x_N & 0 \end{bmatrix}}_{\boldsymbol{X}} \boxplus \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\boldsymbol{w}} = \underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}}_{\boldsymbol{y}} \tag{82}$$

By Theorem 2, the optimal (min ℓ_p error) subsolution is

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \boldsymbol{X}^* \boxplus' \boldsymbol{y} = \begin{bmatrix} \bigwedge_i y_i - x_i \\ \bigwedge_i y_i \end{bmatrix} \tag{83}$$

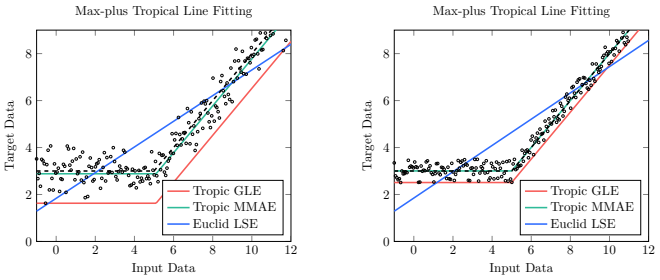
where $\boldsymbol{X}^* = -\boldsymbol{X}^T$ and $\bigwedge_i = \bigwedge_{i=1}^N$. This vector $\hat{\boldsymbol{w}}$ yields (after max-plus 'multiplication' with $\boldsymbol{X}$) the *greatest lower estimate (GLE)* of the data $\boldsymbol{y}$. Thus, the above approach allows to optimally fit (w.r.t. any ℓ_p error norm) max-plus tropical lines to arbitrary data from below. In addition, we can obtain the best (unconstrained) approximation with a tropical line that yields the smallest ℓ_∞ error. This *minimum max absolute error (MMAE)* solution is, by Theorem 3,

$$\tilde{\boldsymbol{w}} = \hat{\boldsymbol{w}} + \mu, \quad \mu = \frac{1}{2}\|\boldsymbol{X} \boxplus \hat{\boldsymbol{w}} - \boldsymbol{y}\|_\infty \tag{84}$$

Example 7. Suppose we have $N = 200$ data observations (x_i, y_i) from the tropical line $p(x) = \max(x - 2, 3)$, where the 200 abscissae x_i were uniformly spaced in $[-1, 12]$ and their corresponding values $y_i = p(x_i) + \epsilon_i$ are contaminated with two different types of zero-mean noise i.i.d. random variables ϵ_i, Gaussian noise $\sim \mathcal{N}(0, 0.25)$ and uniform noise $\sim$ Unif$[-0.5, 0.5]$. Figure 15 shows the two optimal solutions (83) and (84) for fitting a max-plus tropical line, superimposed with the least-squares Euclidean line fit. The parameter estimates and errors are shown in Table II.

Table II
ERRORS AND PARAMETER ESTIMATES FOR OPTIMAL FITTING OF A MAX-PLUS TROPICAL LINE TO DATA CORRUPTED BY UNIFORM NOISE.

Line fit Method	$\|\text{error}\|_{\text{RMS}}$	$\|\text{error}\|_\infty$	$\hat{a}$	$\hat{b}$
Tropical GLE	0.598	0.988	-2.492	2.509
Tropical MMAE	0.288	0.494	-1.998	3.003
Euclidean LSE	0.968	2.135	0.560	1.849



(a) T-line with Gaussian Noise (b) T-line with Uniform Noise

Figure 15. (a) Optimal fitting via (83) or (84) of a max-plus tropical line $y = \max(x - 2, 3)$ (shown in black dashed curve) to data from the line corrupted by additive i.i.d. Gaussian noise $\sim \mathcal{N}(0, 0.25)$. Blue line: Euclidean line fitting via least squares. Red line: best subsolution (GLE). Green line: best unconstrained (MMAE) solution. (b) Same experiment as in (a) but with uniform noise $\sim$ Unif$[-0.5, 0.5]$. Best viewed in color.

The above approach and tropical solution can also be extended to fitting planes. Specifically, we wish to fit a general max-plus tropical plane $p(x, y)$

$$p(x, y) = \max(a + x, b + y, c) \tag{85}$$

to given data $(x_i, y_i, z_i) \in \mathbb{R}^3$, $i = 1, ..., N$, where $z_i = p(x_i, y_i) +$ error, by minimizing some ℓ_p error norm. The

equations to solve for finding the parameters $\boldsymbol{w} = [a, b, c]^T$ become:

$$\underbrace{\begin{bmatrix} x_1 & y_1 & 0 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 0 \end{bmatrix}}_{\boldsymbol{X}} \boxplus \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\boldsymbol{w}} = \underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}}_{\boldsymbol{z}} \tag{86}$$

By Theorem 2 the optimal subsolution, which yields approximations of $\boldsymbol{z} = [z_i]$ from below, is $\hat{\boldsymbol{w}} = \boldsymbol{X}^* \boxplus' \boldsymbol{z}$. Hence,

$$\underbrace{\begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix}}_{\hat{\boldsymbol{w}}} = \underbrace{\begin{bmatrix} -x_1 & \cdots & -x_N \\ -y_1 & \cdots & -y_N \\ 0 & \cdots & 0 \end{bmatrix}}_{\boldsymbol{X}^*} \boxplus' \underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}}_{\boldsymbol{z}} = \begin{bmatrix} \bigwedge_i z_i - x_i \\ \bigwedge_i z_i - y_i \\ \bigwedge_i z_i \end{bmatrix} \tag{87}$$

Furthermore, the minimum max absolute error (MMAE) solution is given by $\tilde{\boldsymbol{w}} = \hat{\boldsymbol{w}} + \mu$ where $\mu = \frac{1}{2}\|\boldsymbol{X} \boxplus \hat{\boldsymbol{w}} - \boldsymbol{z}\|_\infty$.

C. Optimal Fitting Tropical Polynomial Curves and Surfaces

The above approach and solution can also be generalized to polynomial curves of higher degree and to multi-dimensional data $\boldsymbol{x} \in \mathbb{R}^d$. We wish to fit a max-plus tropical polynomial

$$p(\boldsymbol{x}) = \max(\boldsymbol{a}_1^T\boldsymbol{x}+b_1, \ldots, \boldsymbol{a}_K^T\boldsymbol{x}+b_K) = \bigvee_{k=1}^{K} \boldsymbol{a}_k^T\boldsymbol{x}+b_k \tag{88}$$

to given data $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^{d+1}$, $i = 1, ..., N$, where $y_i = p(\boldsymbol{x}_i) + \text{error}$, by minimizing some ℓ_p error norm. The exact equations are

$$\underbrace{\begin{bmatrix} \boldsymbol{a}_1^T\boldsymbol{x}_1 & \boldsymbol{a}_2^T\boldsymbol{x}_1 & \cdots & \boldsymbol{a}_K^T\boldsymbol{x}_1 \\ \boldsymbol{a}_1^T\boldsymbol{x}_2 & \boldsymbol{a}_2^T\boldsymbol{x}_2 & \cdots & \boldsymbol{a}_K^T\boldsymbol{x}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{a}_1^T\boldsymbol{x}_N & \boldsymbol{a}_2^T\boldsymbol{x}_N & \cdots & \boldsymbol{a}_K^T\boldsymbol{x}_N \end{bmatrix}}_{\boldsymbol{X}} \boxplus \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix}}_{\boldsymbol{w}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\boldsymbol{y}} \tag{89}$$

We assume that the slope vectors $\boldsymbol{a}_k$ are given and we optimize for the parameters $\{b_k\}$. By Theorem 2, the optimal subsolution for minimum ℓ_p error is

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \hat{b}_1 \\ \vdots \\ \hat{b}_K \end{bmatrix} = \boldsymbol{X}^* \boxplus' \boldsymbol{y} = \begin{bmatrix} \bigwedge_{i=1}^N y_i - \boldsymbol{a}_1^T\boldsymbol{x}_i \\ \vdots \\ \bigwedge_{i=1}^N y_i - \boldsymbol{a}_K^T\boldsymbol{x}_i \end{bmatrix} \tag{90}$$

Note that $\boldsymbol{X} \boxplus \hat{\boldsymbol{w}} \leq \boldsymbol{y}$. Further, by Theorem 3, the unconstrained solution that yields the minimum ℓ_∞ error is

$$\tilde{\boldsymbol{w}} = \mu + \hat{\boldsymbol{w}}, \quad \mu = \frac{1}{2}\|\boldsymbol{X} \boxplus \hat{\boldsymbol{w}} - \boldsymbol{y}\|_\infty \tag{91}$$

There are two major categories of problems to which the above general tropical regression model can be applied: First, if the slopes $\boldsymbol{a}_k$ are known for all K terms, then the above optimal solutions estimate the rest of the tropical model parameters (i.e. the intercepts b_k) with a linear complexity $O(dNK)$. The assumption for known slope vectors $\boldsymbol{a}_k$ does not pose a significant constraint in many cases where the degree $r = |\boldsymbol{a}| = \max_k \|\boldsymbol{a}_k\|_1$ of the required tropical polynomial for a good fit is relatively small or comparable to the rank K. For instance, in this case we can assume all integer slopes up to r or integer multiples of a slope step. Second, in case of a-priori unknown slopes, we can cluster the data gradients using K-means, use the centroids of the K clusters as estimates of the slope vectors, and then optimally solve for the intercepts; this approach was proposed in [72]. In both approaches, setting $b_k = -\infty$ for some k, removes the corresponding line or hyperplane from the max-affine combination. Next we illustrate both approaches via an example.

Example 8. Suppose we are given $N = 500$ data observations (x_i, y_i, z_i) as in Fig. 16 from the noisy paraboloid surface [43]

$$z = x^2 + y^2 + \epsilon \tag{92}$$

where $\epsilon \sim \mathcal{N}(0, 0.25^2)$ is zero-mean noise and the planar locations x_i, y_i of the data points were drawn as i.i.d. random variables $\sim \text{Unif}[-1, 1]$. First, as a tropical regression example with known slopes, let us fit to the above data a symmetric (with all positive and negative integer slopes in $[-2, 2]$) max-plus tropical conic polynomial

$$p(x, y) = \bigvee_{0 \leq |k+\ell| \leq 2,\ k \cdot \ell \geq 0} b_{k\ell} + kx + \ell y \tag{93}$$

where $z_i = p(x_i, y_i) + \text{error}$, by minimizing some ℓ_p error norm. The equations to solve for finding the 11 parameters $\boldsymbol{w} = [b_{0,-2}, \ldots, b_{0,0}, b_{1,0}, b_{0,1}, b_{1,1}, b_{2,0}, b_{0,2}]^T$ become:

$$\underbrace{\begin{bmatrix} -2y_1 & \cdots & 0 & x_1 & \cdots & 2x_1 & 2y_1 \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ -2y_N & \cdots & 0 & x_N & \cdots & 2x_N & 2y_N \end{bmatrix}}_{\boldsymbol{X}} \boxplus \boldsymbol{w} = \boldsymbol{z} \tag{94}$$

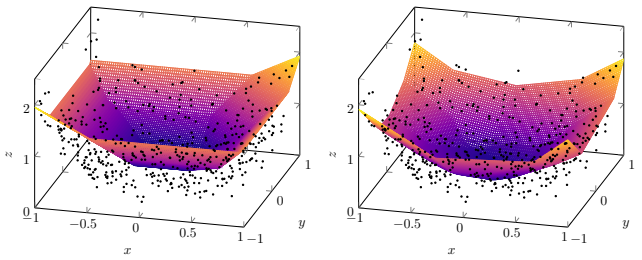
By Theorem 3, the optimal unconstrained solution for MMAE is $\tilde{\boldsymbol{w}} = \mu + \hat{\boldsymbol{w}}$ where $\hat{\boldsymbol{w}} = \boldsymbol{X}^* \boxplus' \boldsymbol{z}$ and μ is half the ℓ_∞ error incurred by $\hat{\boldsymbol{w}}$. The resulting MMAE conic surface is shown in Fig. 16.

As a second approach, let us fit a tropical model of rank K:

$$p_K(x, y) = \max(a_1 x + b_1 y + c_1, ..., a_K x + b_K y + c_K), \tag{95}$$

This consists of K planes of unknown slopes estimated by using K-means on the numerical gradients of the 2D data, whereas the intercepts c_k are computed using the tropical fitting algorithm as in (91). By varying K, we find that even a small number of planes with adaptive slopes (e.g. see the case $K = 25$ shown in Fig. 16) can yield better approximations than the fixed slope case but of course at a higher computational cost, as discussed next. Further, the errors given in Table III for various K indicate that even a small number of adaptive planes can yield good approximations.

Computational Complexity: Recent methods for convex PWL data fitting are commonly variations of iterative nonlinear least-squares algorithms. The standard least-squares estimator [52] scales cubically in d and N, becoming intractable in the high-dimensional and/or large sample setting. The nonlinear least-squares problems in [51], [66] are solved



(a) 2D conic (K=11) (b) K=25

Figure 16. 2D Tropical fitting using the optimal unconstrained (MMAE) approach to data from (92). (a) Tropic conic with known integer slopes. (b) Slopes found via K-means on gradients.

Table III
ERRORS FOR OPTIMAL TROPICAL FITTING OF THE FUNCTION (92) USING 2D MAX-PLUS POLYNOMIALS.

	GLE		MMAE	
K	$\text{error}_{\text{RMS}}$	$\|\text{error}\|_\infty$	$\text{error}_{\text{RMS}}$	$\|\text{error}\|_\infty$
11 (conic)	0.6307	1.7049	0.4167	0.8524
10	0.6659	1.6022	0.3641	0.8011
25	0.5674	1.2779	0.3016	0.6389
50	0.5489	1.3068	0.3159	0.6534
100	0.5364	1.2828	0.3135	0.6414

iteratively via a partitioning algorithm, with each iteration taking time $O((d+1)^2 N)$; however, these algorithms may not converge or fit the data poorly. This obstacle, largely due to nonconvexity, is empirically overcome by running multiple instances of the algorithm from random initializations. The convex adaptive partitioning (CAP) algorithm proposed in [43] solves a linear regression problem for each partition, leading to time complexity $O(d(d+1)^2 N \log(N) \log\log(N))$.

In contrast, the complexity of our algorithm for the case of unknown slopes has a complexity of $O(dNKi_K)$, where i_K is the number of K-means iterations. After computing the K centroids $\boldsymbol{a}_k$, it performs a single pass over the data to form (89) and solve for b_k, with total complexity $O(dNK)$. If the true slopes have some clustering structure, K-means will converge quickly and the cost of our algorithm will be practically linear. As such, in non-pathological cases, we can assume that $Ki_K \ll dN$ and may be treated as a constant, thus improving on both the CAP algorithm and on the traditional LSE. Finally, note that for case with known slopes our algorithm has a very small complexity $O(dN)$.

VIII. TROPICAL ALGEBRA AND GEOMETRY ON WEIGHTED LATTICES

A. Generalized Tropical Lines and Planes

In the same way that weighted lattices generalize max-plus algebra and extend it to other types of clodum arithmetic, we can extend the basic objects of max-plus tropical geometry (i.e. tropical lines and planes) to other max-$\star$ geometric objects. For example, over a clodum $(\mathcal{K}, \vee, \wedge, \star, \star')$, we can generalize max-plus tropical lines $y = \max(a+x, b)$ as $y = \max(a \star x, b)$. Figure 17(a)-(d) shows some generalized tropical lines where the $\star$ operation is sum $(+)$, product $(\times)$, min $(\wedge)$, and softmin $(\curlywedge_\theta)$. In the first three cases the generalized tropical lines are PWL functions. However, in Fig. 17(d) a portion of the line is curving. To further illustrate this curving and create a symmetry between the max and min operations, we show in Fig. 17(e) a smooth function

$$\begin{aligned} s(x) &= (a \curlywedge_\theta x) \curlyvee_\theta b \\ &= \theta \log[\exp(-\log(e^{-a/\theta} + e^{-x/\theta}) + e^{b/\theta}] \end{aligned} \tag{96}$$

that goes beyond the max-$\star$ framework and is actually a softmax-softmin.

Similarly, we can generalize max-plus tropical planes $z = \max(a+x, b+y, c)$ to max-$\star$ as $z = \max(a \star x, b \star y, c)$. Figure 18 shows a max-min plane where $\star = \min$. This is an interesting geometrical polyhedral object that consists of portions of planes, either sloped or horizontal, at several levels.

Further, we can generalize max-plus halfspaces (16) to max-$\star$ tropical halfspaces:

$$\mathcal{T}(\boldsymbol{a}, \boldsymbol{b}) := \left\{ \boldsymbol{x} \in \mathcal{K}^d : \boldsymbol{a}^T \boxtimes \begin{bmatrix} \boldsymbol{x} \\ e \end{bmatrix} \leq \boldsymbol{b}^T \boxtimes \begin{bmatrix} \boldsymbol{x} \\ e \end{bmatrix} \right\} \tag{97}$$

Examples of max-plus tropical halfspaces are shown in Fig. 5 and Fig. 6. The slopes of their bounding line segments or faces are either zero or equal to 1. Max-product halfspaces can give boundaries that are piecewise-linear but have arbitrary slopes. Max-min halfspaces have piecewise-linear boundaries with more corner points or edges; see examples in Fig. 17(c) and Fig. 18. Finally, a totally different generalization results if we replace the 'multiplication' $\star$ in a generalized tropical line with the (log-sum-exp) softmin operation of (2), as shown in Fig. 17(d)-(e), in which case the line segments of a tropical line will become partially or totally smooth exponential curves.

B. Generalized Tropical Regression

Suppose we wish to fit a general max-$\star$ tropical plane

$$p(x, y) = \max(a \star x, b \star y, c) \tag{98}$$

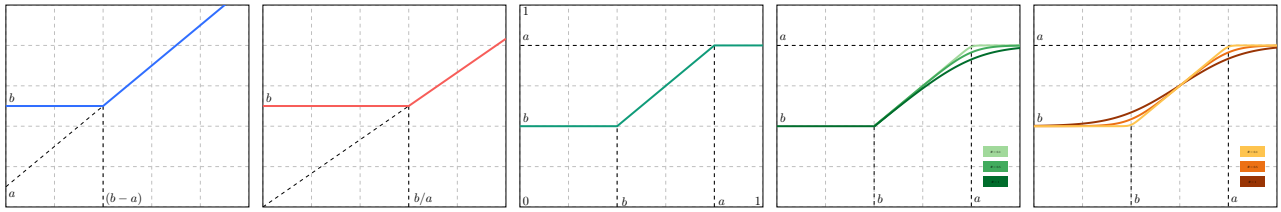
to given data $(x_i, y_i, z_i) \in \mathbb{R}^3$, $i = 1, \ldots, N$, where $z_i = p(x_i, y_i) + \text{error}$, by minimizing some ℓ_p error norm. The equations to solve for finding the optimal parameters $\boldsymbol{w} = [a, b, c]^T$ become:

$$\underbrace{\begin{bmatrix} x_1 & y_1 & e \\ \vdots & \vdots & \vdots \\ x_N & y_N & e \end{bmatrix}}_{\boldsymbol{X}} \boxtimes \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\boldsymbol{w}} = \underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix}}_{\boldsymbol{z}} \tag{99}$$

If we can accept subsolutions, which yield approximations of the given data from below, then by Theorem 4 the optimal subsolution for any clodum arithmetic is

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix} = \begin{bmatrix} \bigwedge_i \zeta(x_i, z_i) \\ \bigwedge_i \zeta(y_i, z_i) \\ \bigwedge_i \zeta(e, z_i) \end{bmatrix} \tag{100}$$

where ζ is the scalar adjoint erosion (28) of $\star$. This vector $\hat{\boldsymbol{w}}$ yields (after max-$\star$ 'multiplication' with $\boldsymbol{X}$) the greatest lower estimate of $\boldsymbol{z}$. Next we write in detail the solution for the three



(a) Max-plus line (b) Max-product line (c) Max-min line (d) Max-softmin line (e) Softmax-softmin line

Figure 17. (a)-(d) Max-$\star$ tropical lines $y = \max(a \star x, b)$: (a) Max-plus: $y = \max(a + x, b)$, (b) Max-times: $y = \max(a \cdot x, b)$, (c) Max-min: $y = \max(a \wedge x, b)$, (d) Max-softmin: $y = \max(a \curlywedge_\theta x, b)$. (e) Softmax-softmin line: $s(x) = (a \curlywedge_\theta x) \curlyvee_\theta b$. In Figs. (d) and (e) the parameter θ varies.

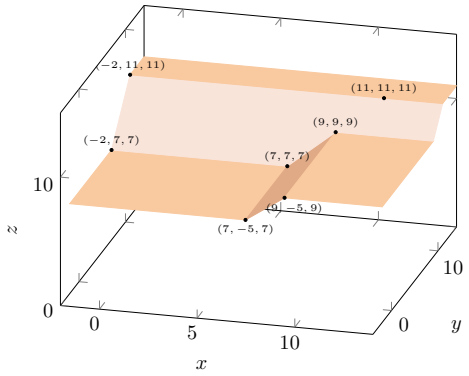


Figure 18. Max-min plane $z = \max(9 \wedge x, 11 \wedge y, 7)$.

special cases where the scalar arithmetic is based either on the max-plus, or the max-times, or the max-min clodum:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix}^T = \begin{cases} (\bigwedge_i z_i - x_i,\ \bigwedge_i z_i - y_i,\ \bigwedge_i z_i) \\ (\bigwedge_i z_i / x_i,\ \bigwedge_i z_i / y_i,\ \bigwedge_i z_i) \\ (\bigwedge_i (z_i \vee \mathbf{1}_{[z_i \geq x_i]}),\ \bigwedge_i (z_i \vee \mathbf{1}_{[z_i \geq y_i]}),\ \bigwedge_i z_i) \end{cases} \tag{101}$$

This approach allows to optimally fit (w.r.t. any ℓ_p error norm) general max-$\star$ tropical planes to arbitrary data from below.

IX. Conclusions and Future Directions

Tropical geometry and max-plus algebra offer a rich collection of ideas and tools to model and solve problems in machine learning. In this work we have surveyed the state-of-the-art and some recent progress in three areas: (1) deep neural networks with PWL activation functions, (2) probabilistic graphical models and algorithms for Weighted Finite State Transducers, and (3) nonlinear regression with PWL functions. Further we have introduced extensions to general max algebras that allowed us to (i) express the optimal solutions of several of the above problems as projections onto nonlinear vector spaces called weighted lattices and (ii) generalize tropical geometrical objects. We conclude by outlining below some future research directions.

A) This work developed a Newton polytope representation of neural network layers, which was explored in the context of single-layer networks; owing to the stability of Newton polytopes under addition and multiplication, one could try to derive similar representations for compositions of multiple layers. On one hand, this may allow for refined empirical estimates on their complexity measured in terms of linear regions; on the other hand, further developing the aforementioned representation can pave the way for better network minimization methods, possibly combined with ideas from sparse regression.

B) It was briefly mentioned in Section II-E that tropical polytopes are more "economical" in their number of required parameters. This can introduce a whole new field of study, where there is an explicit characterization of Euclidean polytopes that can be exactly represented by a, more efficient, tropical polytope, while also providing quantifiable metrics for the relative gain.

C) The results of Section VI-B4 can be extended to more concrete benefits. While mainly algebraic, there is an extensive theory on the *reachability* and *robustness* [15] of the tropical matrices via their (sub)eigenvalue characterizations. Adapting these results to the WFST setting is nontrivial and a possible avenue for future study.

D) Regarding our work on tropical regression, we note that the max-affine representation is not limited to PWL functions only, because we can represent any convex function as a supremum of a (possibly infinite) number of affine functions via the Fenchel-Legendre transform [31], [62], [90]. Closely related ideas are based on morphological slope transforms [29], [48], [67] that offer generalizations of this result to non-convex functions and approximate representations via adjunctions.

Acknowledgment

The authors wish to thank G. Smyrnis for insightful discussions on tropical geometry and neural networks.

References

[1] M. Akian, S. Gaubert, and A. Guterman, "Tropical Polyhedra Are Equivalent To Mean Payoff Games," *International Journal of Algebra and Computation*, vol. 22, no. 1, 2012.

[2] M. Akian, S. Gaubert, V. Nitica, and I. Singer, "Best approximation in max-plus semimodules," *Linear Alegbra and its Applications*, vol. 435, p. 3261–3296, 2011.

[3] M. Alfarra, A. Bibi, H. Hammoud, M. Gaafar, and B. Ghanem, "On the Decision Boundaries of Deep Neural Networks: A Tropical Geometry Perspective," *arXiv*, 2020.

[4] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding Deep Neural Networks with Rectified Linear Units," in *International Conference on Learning Representations*, 2018.

[5] D. Avis and K. Fukuda, "Reverse search for enumeration," *Discrete Applied Mathematics*, vol. 65, no. 1-3, pp. 21–46, 1996.

[6] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems*. J. Wiley & Sons, 2001.

[7] G. Birkhoff, *Lattice Theory*. American Mathematical Soeciety, 1967.

[8] T. S. Blyth, *Lattices and Ordered Algebraic Structures*. Springer-Verlag, 2005.

[9] T. S. Blyth and M. F. Janowitz, *Residuation Theory*. Pergamon Press, 1972.

[10] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed Sensing using Generative Models," in *International Conference on Machine Learning*, 2017.

[11] G. Borgefors, "Distance Transformations in Arbitrary Dimensions," *Computer Vision, Graphics, and Image Processing*, vol. 27, pp. 321–345, 1984.

[12] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, "A tutorial on geometric programming," *Optimization and Engineering*, vol. 8, p. 67–127, 2007.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[14] R. W. Brockett and P. Maragos, "Evolution Equations for Continuous-Scale Morphological Filtering," *IEEE Transactions on Signal Processing*, vol. 42, no. 12, pp. 3377–3386, 1994.

[15] P. Butkovič, *Max-linear Systems: Theory and Algorithms*. Springer, 2010.

[16] G. C. Calafiore, S. Gaubert, and C. Possieri, "Log-Sum-Exp Neural Networks and Posynomial Models for Convex and Log-Log-Convex Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1–12, 2019.

[17] G. C. Calafiore, S. Gaubert, and C. Possieri, "A Universal Approximation Result for Difference of Log-Sum-Exp Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–10, 2020.

[18] V. Charisopoulos and P. Maragos, "Morphological Perceptrons: Geometry and Training Algorithms," in *International Symposium on Mathematical Morphology*, 2017.

[19] ——, "A Tropical Approach to Neural Networks with Piecewise Linear Activations," *arXiv*, 2018.

[20] W. Chen and R. Mazumder, "Multivariate Convex Regression at Scale," *arXiv*, 2020.

[21] L. O. Chua and A.-C. Deng, "Canonical Piecewise-Linear Representation," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 1, p. 101–111, 1988.

[22] G. Cohen, D. Dubois, J. Quadrat, and M. Viot, "A Linear System Theoretic View of Discrete Event Processes and Its Use for Performance Evaluation in Manufacturing," *IEEE Transactions on Automatic Control*, vol. 30, pp. 210–220, 1985.

[23] G. Cohen, S. Gaubert, and J. Quadrat, "Duality and separation theorems in idempotent semimodules," *Linear Alegbra and its Applications*, vol. 379, p. 395–422, 2004.

[24] M. A. Cueto, J. Morton, and B. Sturmfels, "Geometry of the restricted Boltzmann machine," *Algebraic Methods in Statistics and Probability II*, vol. 516, pp. 135–153, 2010.

[25] R. Cuninghame-Green, *Minimax Algebra*. Springer-Verlag, 1979.

[26] R. A. Cuninghame-Green, "Projections in Minimax Algebra," *Mathematical Programming*, vol. 10, pp. 111–123, 1976.

[27] R. A. Cuninghame-Green and K. Cechlarova, "Residuation in fuzzy algebra and some applications," *Fuzzy Sets and Systems*, vol. 71, pp. 227–239, 1995.

[28] A. Damle and Y. Sun, "A Geometric Approach to Archetypal Analysis and Nonnegative Matrix Factorization," *Technometrics*, vol. 59, no. 3, pp. 361–370, 2017.

[29] L. Dorst and R. van den Boomgaard, "Morphological Signal Processing and the Slope Transform," *Signal Processing*, vol. 38, no. 1, pp. 79–98, 1994.

[30] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[31] W. Fenchel, "On Conjugate Convex Functions," *Canadian Journal of Mathematics*, vol. 1, pp. 73–77, 1949.

[32] G. Franchi, A. Fehri, and A. Yao, "Deep Morphological Networks," *Pattern Recognition*, vol. 102, 2020.

[33] K. Fukuda, "From the zonotope construction to the Minkowski addition of convex polytopes," *Journal of Symbolic Computation*, vol. 38, no. 4, pp. 1261–1272, 2004.

[34] S. Gaubert and R. D. Katz, "Minimal half-spaces and external representation of tropical polyhedra," *Journal of Algebraic Combinatorics*, vol. 33, pp. 325–348, 2011.

[35] S. Gaubert and M. Plus, "Methods and Applications of (max,+) Linear Algebra," in *Symposium on Theoretical Aspects of Computer Science*, 1997.

[36] A. Ghosh, A. Pananjady, A. Guntuboyina, and K. Ramchandran, "Max-affine regression: Provable, tractable, and near-optimal statistical estimation," *arXiv*, 2019.

[37] M. Gondran and M. Minoux, *Graphs, Dioids and Semirings: New Models and Algorithms*. Springer, 2008.

[38] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *International Conference on Machine Learning*, 2013.

[39] P. Gritzmann and B. Sturmfels, "Minkowski addition of polytopes: Computational complexity and applications to Gröbner bases," *SIAM Journal of Discrete Mathematics*, vol. 6, no. 2, pp. 246–269, 1993.

[40] C. Güzelis and I. C. Göknar, "A Canonical Representation for Piecewise-Affine Maps and Its Applications to Circuit Analysis," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 11, pp. 1342–1354, 1991.

[41] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both Weights and Connections for Efficient Neural Network," in *Neural Information Processing Systems*, 2015.

[42] P. Hand and V. Voroninski, "Global Guarantees for Enforcing Deep Generative Priors by Empirical Risk," in *Conference On Learning Theory*, 2018.

[43] L. A. Hannah and D. B. Dunson, "Multivariate convex regression with adaptive partitioning," *arXiv*, 2011.

[44] ——, "Ensemble Methods for Convex Regression with Applications to Geometric Programming Based Circuit Design," in *International Conference on Machine Learning*, 2012.

[45] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "AMC: AutoML for Model Compression and Acceleration on Mobile Devices," in *European Conference on Computer Vision*, 2018.

[46] B. Heidergott, G. J. Olsder, and J. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems: a Course on Max-Plus Algebra and Its Applications*. Princeton University Press, 2006.

[47] H. Heijmans, *Morphological Image Operators*. Academic Press, 1994.

[48] H. Heijmans and P. Maragos, "Lattice Calculus of the Morphological Slope Transform," *Signal Processing*, vol. 59, no. 1, pp. 17–42, 1997.

[49] C. Hildreth, "Point estimates of ordinates of concave functions," *Journal of the American Statistical Association*, vol. 49, no. 267, pp. 598–619, 1954.

[50] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[51] W. Hoburg, P. Kirschen, and P. Abbeel, "Data fitting with geometric-programming-compatible softmax functions," *Optimization and Engineering*, vol. 17, p. 897–918, 2016.

[52] C. A. Holloway, "On the estimation of convex functions," *Operations Research*, vol. 27, no. 2, pp. 401–407, 1979.

[53] J. Hook, "Linear regression over the max-plus semiring: algorithms and applications," *arXiv*, 2017.

[54] T. Hori and A. Nakamura, *Speech Recognition Algorithms Using Weighted Finite-State Transducers*. Morgan & Claypool, 2013.

[55] J.-E. Perin, "Tropical Semirings," in *Idempotency*, J. Gunawardena, Ed. Cambridge University Press, 1998, pp. 50–69.

[56] P. Julian, "The Complete Canonical Piecewise-Linear Representation: Functional Form Minimal Degenerate Intersections," *IEEE Transactions on Circuits and Systems-I: Fundamental Theories and Applications*, vol. 50, no. 3, pp. 387–396, 2003.

[57] C. Kahlert and L. O. Chua, "The Complete Canonical Piecewise-Linear Representation - Part I: The Geometry of the Domain Space," *IEEE Transactions on Circuits and Systems-I: Fundamental Theories and Applications*, vol. 39, no. 3, pp. 222–236, 1992.

[58] J. Kim, L. Vandenberghe, and C. Yang, "Convex Piecewise-Linear Modeling Method for Circuit Optimization via Geometric Programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 11, pp. 1823–1827, 2010.

[59] J.-N. Lin, H.-Q. Xu, and R. Unbehauen, "A Generalization of Canonical Piecewise-Linear Functions," *IEEE Transactions on Circuits and Systems-I: Fundamental Theories and Applications*, vol. 41, no. 4, pp. 345–347, 1994.

[60] G. L. Litvinov, "Maslov Dequantization, Idempotent and Tropical Mathematics: A Brief Introduction," *Journal of Mathematical Sciences*, vol. 140, no. 3, 2007.

[61] G. L. Litvinov, V. P. Maslov, and G. B. Shpiz, "Idempotent functional analysis: An algebraic approach," *Mathematical Notes*, vol. 69, no. 5, p. 696–729, 2001.

[62] Y. Lucet, "What Shape is Your Conjugate? A Survey of Computational Convex Analysis and Its Applications," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 216–250, 2009.

[63] J.-H. Luo, J. Wu, and W. Lin, "ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression," in *International Conference on Computer Vision*, 2017.

[64] A. Maas, A. Hannun, and A. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *International Conference on Machine Learning*, 2013.

[65] D. Maclagan and B. Sturmfels, *Introduction to Tropical Geometry*. American Mathematical Society, 2015.

[66] A. Magnani and S. P. Boyd, "Convex piecewise-linear fitting," *Optimization and Engineering*, vol. 10, pp. 1–17, 2009.

[67] P. Maragos, "Morphological Systems: Slope Transforms and Max-Min Difference and Differential Equations," *Signal Processing*, vol. 38, no. 1, pp. 57–77, 1994.

[68] ——, "Lattice image processing: A unification of morphological and fuzzy algebraic systems," *Journal of Mathematical Imaging and Vision*, vol. 22, pp. 333–353, 2005.

[69] ——, "Morphological Filtering for Image Enhancement and Feature Detection," in *Image and Video Processing Handbook*, 2nd ed., A. Bovik, Ed. Elsevier Academic Press, 2005, pp. 135–156.

[70] ——, "Dynamical systems on weighted lattices: General theory," *Mathematics of Control, Signals, and Systems*, vol. 29, no. 21, 2017.

[71] ——, "Tropical Geometry, Mathematical Morphology and Weighted Lattices," in *International Symposium on Mathematical Morphology*, 2019.

[72] P. Maragos and E. Theodosis, "Multivariate Tropical Regression and Piecewise-Linear Surface Fitting," in *International Conference on Acoustics, Speech, and Signal Processing*, 2020.

[73] V. P. Maslov, "On a new superposition principle for optimization problems," *Russian Mathematical Surveys*, vol. 42, no. 3, p. 39–48, 1987.

[74] W. M. McEneaney, *Max-Plus Methods for Nonlinear Control and Estimation*. Birkhauser, 2006.

[75] F. Meyer, *Topographic Tools for Filtering and Segmentation 1 & 2*. Wiley, 2019.

[76] G. Mikhalkin, "Enumerative Tropical Algebraic Geometry in $\mathbb{R}^2$," *Journal of the American Mathematical Society*, vol. 18, no. 2, pp. 313–377, 2005.

[77] M. Mohri, "Weighted Automata Algorithms," in *Handbook of Weighted Automata*. Springer, 2009, pp. 213–254.

[78] M. Mohri, F. Pereira, and M. Ripley, "Weighted Finite-State Transducers in Speech Recognition," *Computer Speech and Language*, vol. 16, pp. 69–88, 2002.

[79] R. Mondal, S. Santra, and B. Chanda, "Dense Morphological Network: An Universal Function Approximator," *arXiv*, 2019.

[80] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Neural Information Processing Systems*, 2014.

[81] J.-J. Moreau, "Inf-convolution, Sous-additivité, Convexité des Fonctions Numériques," *Journal de Mathématiques Pures et Appliquées*, vol. 49, pp. 109–154, 1970.

[82] L. Pachter and B. Sturmfels, "Tropical geometry of statistical models," *Proceedings of the National Academy of Sciences*, vol. 101, no. 46, pp. 16 132–16 137, 2004.

[83] ——, "Parametric inference for biological sequence analysis," *Proceedings of the National Academy of Sciences*, vol. 101, no. 46, pp. 16 138–16 143, 2004.

[84] R. Pascanu, G. Montufar, and Y. Bengio, "On the number of response regions of deep feed forward networks with piece-wise linear activations," *arXiv*, 2013.

[85] L. F. Pessoa and P. Maragos, "Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition," *Pattern Recognition*, vol. 33, pp. 945–960, 2000.

[86] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the Expressive Power of Deep Neural Networks," in *International Conference on Machine Learning*, 2017.

[87] G. X. Ritter and P. Sussner, "An introduction to morphological neural networks," in *International Conference on Pattern Recognition*, 1996.

[88] G. X. Ritter, P. Sussner, and J. D. de Leon, "Morphological Associative Memories," *IEEE Transactions on Neural Networks*, vol. 9, no. 2, pp. 281–293, 1998.

[89] G. X. Ritter and G. Urcid, "Lattice Algebra Approach to Single-Neuron Computation," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 282–295, 2003.

[90] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[91] J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, 1982.

[92] J. Serra, Ed., *Image Analysis and Mathematical Morphology*. Academic Press, 1988, vol. 2: Theoretical Advances.

[93] T. Serra and S. Ramalingam, "Empirical Bounds on Linear Regions of Deep Rectifier Networks," in *AAAI Conference on Artificial Intelligence*, 2020.

[94] T. Serra, C. Tjandraatmadja, and S. Ramalingam, "Bounding and Counting Linear Regions of Deep Neural Networks," in *International Conference on Machine Learning*, 2018.

[95] I. Simon, "On Semigroups of Matrices Over the Tropical Semiring," *Theoretical Informatics and Applications*, vol. 28, no. 3-4, pp. 277–294, 1994.

[96] G. Smyrnis, P. Maragos, and G. Retsinas, "Maxpolynomial Division with Application To Neural Network Simplification," in *International Conference on Acoustics, Speech, and Signal Processing*, 2020.

[97] G. Smyrnis and P. Maragos, "Multiclass Neural Network Minimization via Tropical Newton Polytope Approximation," in *International Conference on Machine Learning*, 2020.

[98] P. Sussner and E. L. Esmi, "Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm," *Information Sciences*, vol. 181, p. 1929–1950, 2011.

[99] J. M. Tarela and M. V. Martinez, "Region Configurations for Realizability of Lattice Piecewise-linear Models," *Computational Mathematics and Modelling*, vol. 30, pp. 17–27, 1999.

[100] M. Telgarsky, "Benefits of depth in neural networks," in *Conference on Learning Theory*, 2016.

[101] E. Theodosis and P. Maragos, "Analysis of the Viterbi Algorithm Using Tropical Algebra and Geometry," in *International Workshop on Signal Processing Advances in Wireless Communications*, 2018.

[102] ——, "Tropical Modeling of Weighted Transducer Algorithms on Graphs," in *International Conference on Acoustics, Speech, and Signal Processing*, 2019.

[103] A. Toriello and J. P. Vielma, "Fitting piecewise linear continuous functions," *European Journal on Operational Research*, vol. 219, pp. 86–95, 2012.

[104] A. Tsiamis and P. Maragos, "Sparsity in Max-plus Algebra," *Discrete Events Dynamic Systems*, vol. 29, p. 163–189, 2019.

[105] T. van den Boom and B. D. Schutter, "Modeling and control of switching max-plus-linear systems with random and deterministic switching," *Discrete Event Dynamic Systems*, vol. 22, pp. 293—33, 2012.

[106] O. Viro, "Dequantization of Real Algebraic Geometry on Logarithmic Paper," in *European Congress of Mathematics*, 2001.

[107] S. Wang, "General Constructive Representations for Continuous Piecewise-Linear Functions," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, vol. 51, no. 9, pp. 1889–1896, 2004.

[108] P.-F. Yang and P. Maragos, "Min-Max Classifiers: Learnability, Design And Application," *Pattern Recognition*, vol. 28, no. 6, pp. 879–899, 1995.

[109] L. Zhang, G. Naitzat, and L.-H. Lim, "Tropical Geometry of Deep Neural Networks," in *International Conference on Machine Learning*, 2018.

[110] Y. Zhang, S. Blusseau, S. Velasco-Forero, I. Bloch, and J. Angulo, "Max-Plus Operators Applied to Filter Selection and Model Pruning in Neural Networks," in *International Symposium on Mathematical Morphology*, 2019.

[111] U. Zimmermann, *Linear and Combinatorial Optimization in Ordered Algebraic Structures*. North-Holland Publishing, 1981.