Intelligent Robotics and Automation Laboratory
Computer Vision, Speech Communication & Signal Processing Group,
National Technical University of Athens, Greece (NTUA)
Institute of Robotics,
Athena Research and Innovation Center (Athena RC)

Tropical Geometry for Machine Learning

Petros Maragos

(ICASSP2024 Tutorial) slides: https://robotics.ntua.gr/icassp-2024-tutorial/

CaLISTA Workshop, Geometry-informed Machine Learning, Paris, 02 Sep. 2024

Talk Outline

- 1. Elements from Tropical Geometry and Max-Plus Algebra
- 2. Neural Networks with Piecewise-linear (PWL) Activations
- 3. Morphological (Max-plus) Neural Networks
- 4. Piecewise-linear (PWL) Regression

This research work was supported by the Hellenic Foundation for Research & Innovation (H.F.R.I.) under the "2nd Call for H.F.R.I. Research Projects to support Faculty Members & Researchers" (Project Number:2656, Acronym: TROGEMAL).

H.F.R.I. Hellenic Foundation for Research & Innovation

Collaborators

TG&ML: Petros Maragos, Vasilis Charisopoulos, Manos Theodosis



Neural Net Minimization:

Georgios Smyrnis, Panos Misiakos, George Retsinas, Nikos Dimitriadis, Konst. Fotopoulos




Tropical Approximation: Ioannis Kordonis

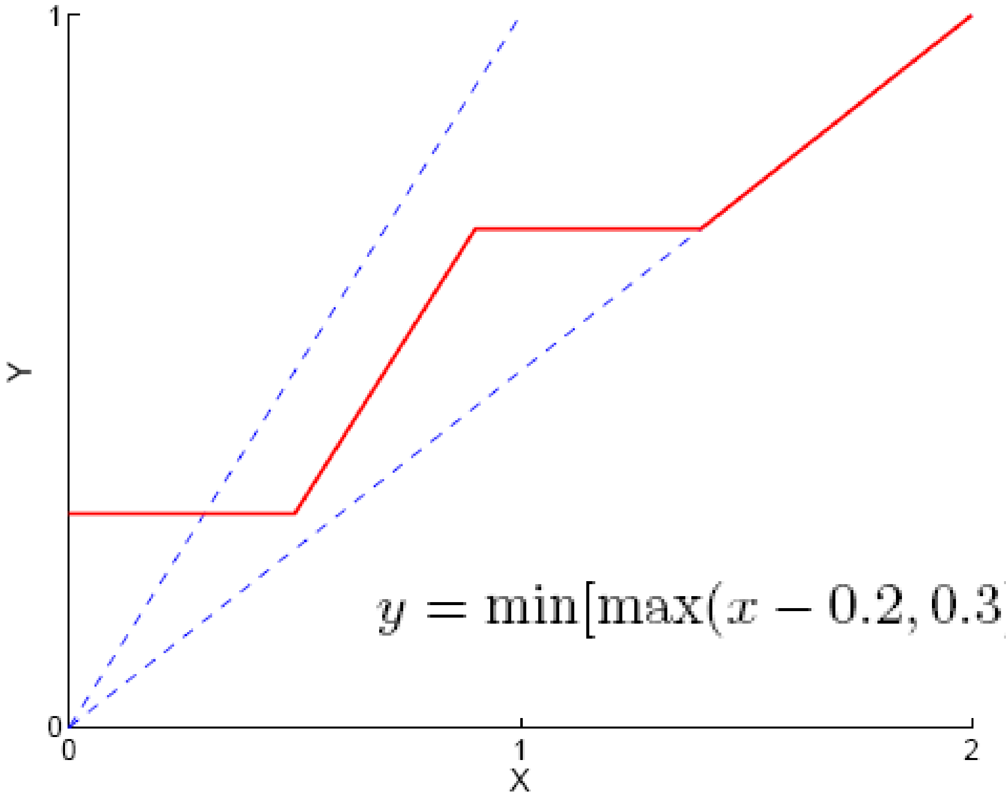
Tropical Sparsity: Anastasios Tsiamis, Nikos Tsilivis


What does TROPICAL mean?

- The adjective “tropical” was coined by French mathematicians Dominique Perrin and Jean-Eric Pin, to honor their Brazilian colleague Imre Simon, a pioneer of min-plus algebra as applied to finite automata in computer science.

- Tropical (**Τροπικός** in Greek) comes from the greek word «**Τροπή**» which means “turning” or “changing the way/direction”.

Polygonal lines



$$y = \min[\max(x - 0.2, 0.3), \max(x/2, 0.7)]$$

Elements of Tropical Geometry

“*TG is a marriage between algebraic geometry and polyhedral geometry. A piecewise-linear version of algebraic geometry.*” [Maclagan & Sturmfels 2015]

Our view: TG is a “dequantized” version of Euclidean geometry and analytic geometry.

References on TG and its Applications to Machine Learning & Optimization

Books & Math Articles on Tropical Geometry (TG):

- D. Maclagan & B. Sturmfels, *Introduction to Tropical Geometry*, AMS 2015.
- I. Itenberg, G. Mikhalkin, and E. I. Shustin, *Tropical Algebraic Geometry,* Springer 2009.
- M. Joswig, *Essentials of Tropical Combinatorics*, AMS 2021.
- *Max-plus Convex Sets/Cones*: [Cuninghame-Green 1979; Butkovic 2007], [Litvinov, Maslov & Sphiz 2001], [Cohen, Gaubert & Quadrat 2004; Gaubert & Katz 2007; Allamigeon et al 2010]
- *Tropical Convexity, Tropical Halfspaces/Polyhedra*: [Maslov 1987], [Develin & Sturmfels 2004], [Joswig 2005], [Gaubert & Katz 2011]. *TG and Mean Payoff Games*: [Akian et al 2012; Akian et al 2021]
- O. Viro, *Dequantization of Real Algebraic Geometry on Logarithmic Paper*, ArXiv 2000.

Some Applications of TG to Machine Learning:

- L. Pachter & B. Sturmfels, *Tropical geometry of statistical models*, PNAS 2004.
- V.Charisopoulos & P.M., *Tropical Approach to Neural Nets with Piecewise Linear Activations*, ISMM2017, ArXiv2018.
- L. Zhang, G. Naitzat, L.-H. Lim, *Tropical Geometry of Deep Neural Networks*, ICML 2018.
- P.M., V. Charisopoulos & E. Theodosis, *Tropical Geometry and Machine Learning*, Proc. IEEE 2021.
- NTUA Group: P.M., Charisopoulos, Dimitriadis, Kordonis, Misiakos, Retsinas, Smyrnis, Theodosis, Tsiamis, Tsilivis
- + Other References in this talk.

Tropical Semirings

Scalar Arithmetic Rings

Integer/Real Addition & Multiplication Ring: $(\mathbb{R}, +, \times)$, $(\mathbb{Z}, +, \times)$

Tropical Semirings

$\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$, $\mathbb{R}_{\min} = \mathbb{R} \cup \{+\infty\}$

$\vee = \max$, $\wedge = \min$

Max-plus semiring: $(\mathbb{R}_{\max}, \vee, +)$

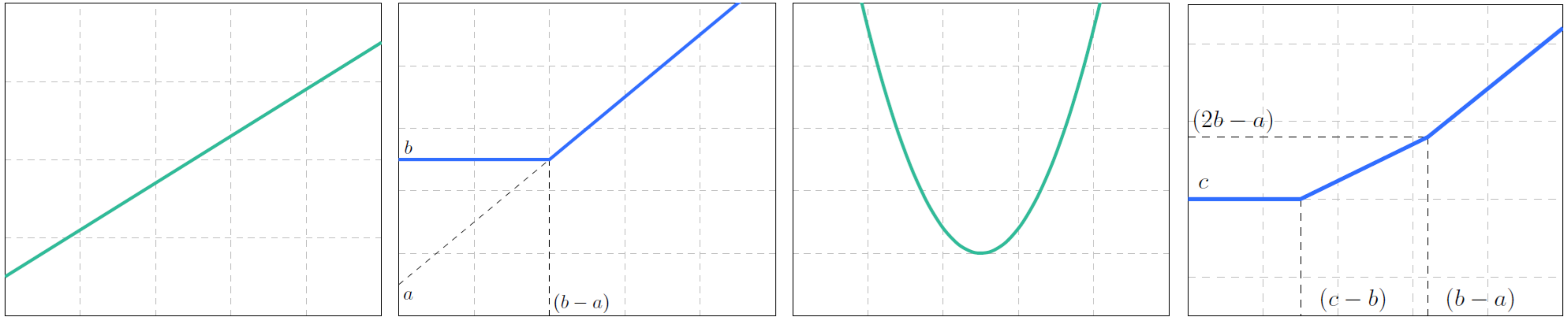
Min-plus semiring: $(\mathbb{R}_{\min}, \wedge, +)$

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

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

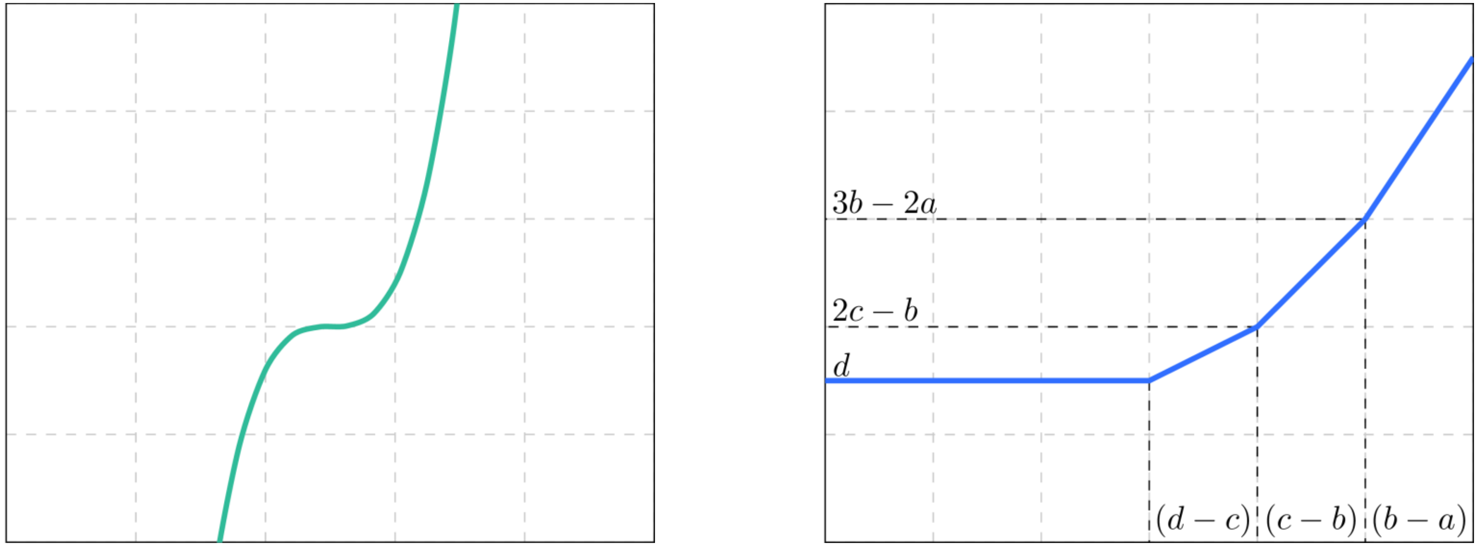
Graphs of Max-plus Tropical 1D Polynomials

$$y_{\text{t-line}} = \max(a + x, b), \quad y_{\text{t-parab}} = \max(a + 2x, b + x, c)$$

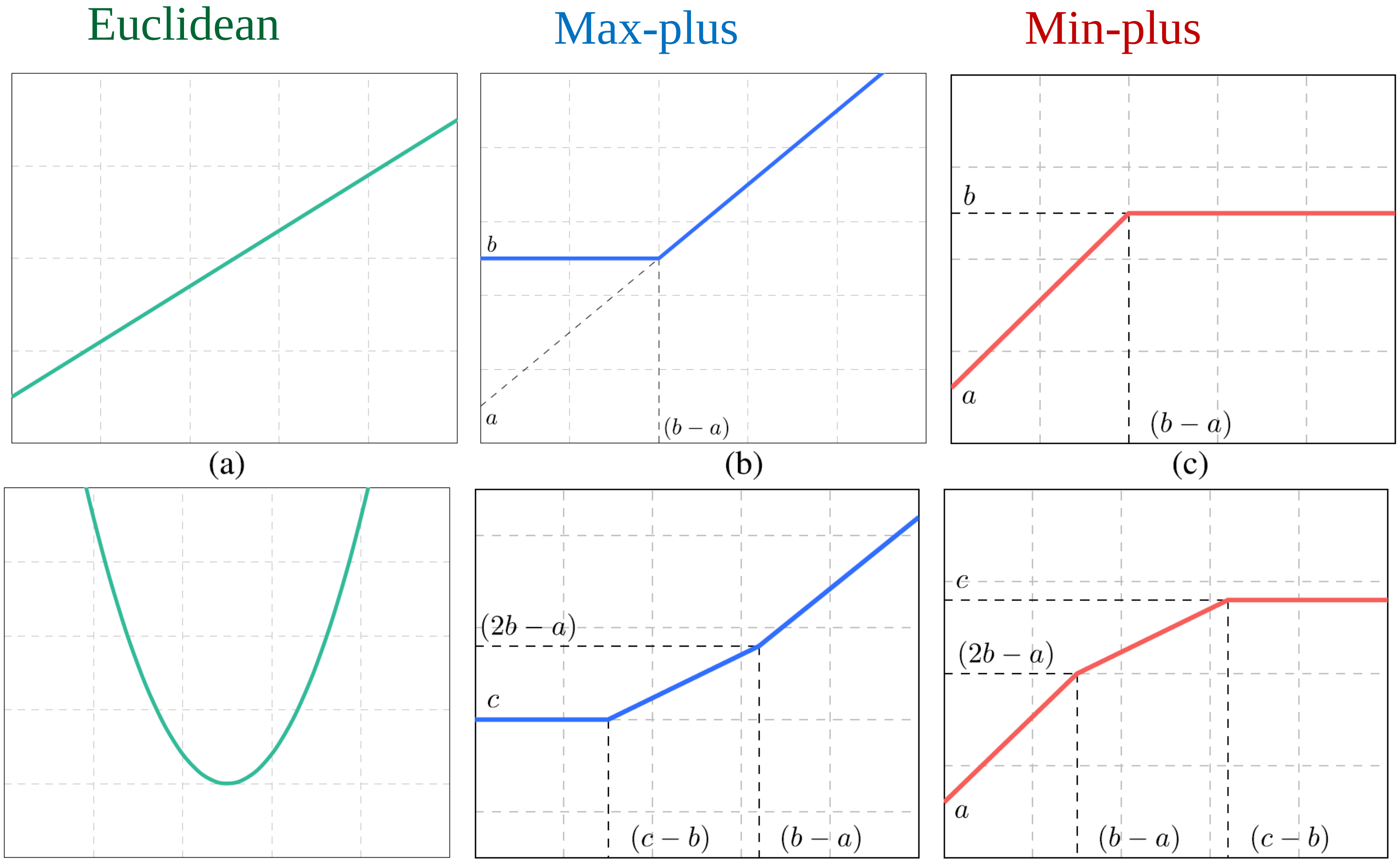


(a) Euclidean line (b) Tropical line (c) Euclid parabola (d) Tropic parabola

Cubic polynomial



Max-plus and Min-Plus Tropical 1D Polynomials

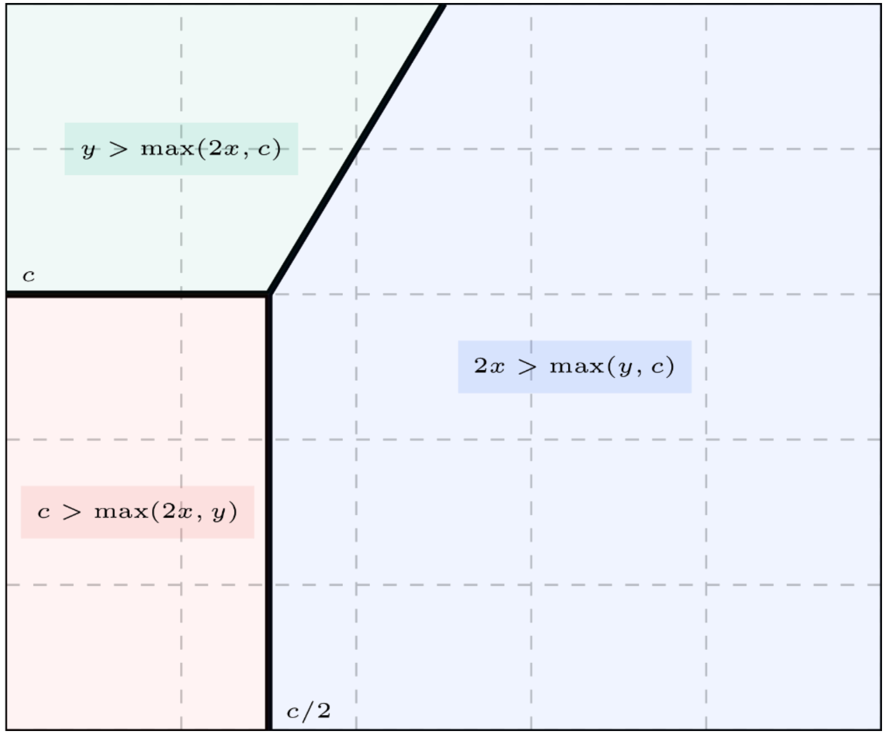


(a) (b) (c)
(d) (e) (f)

Tropical Curve of Max/Min-Polynomials

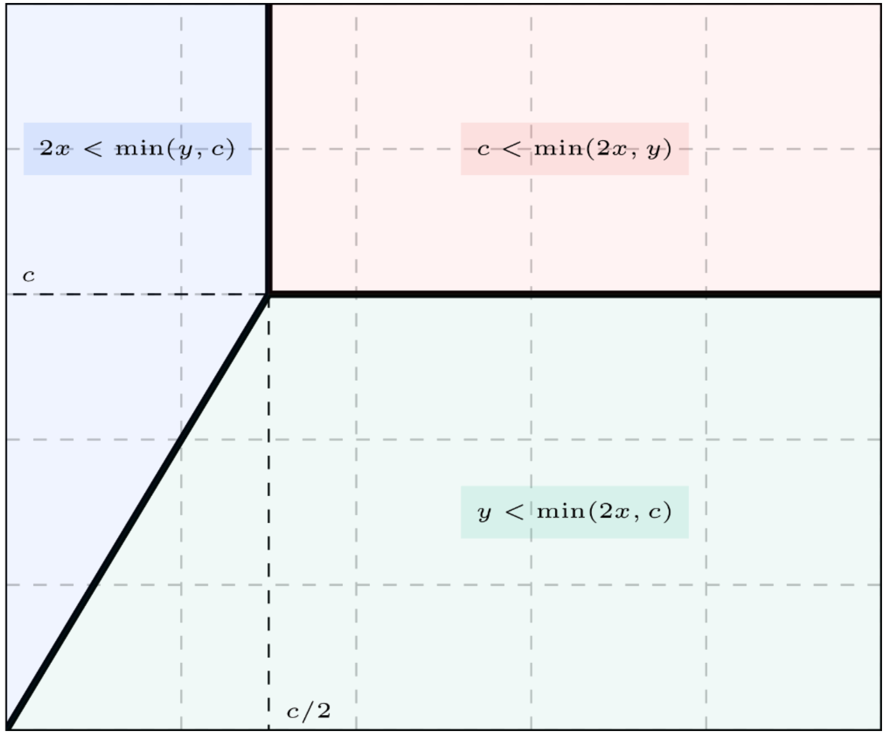
Tropical curve of $p(x,y)$ =

"Zero locus" of a max/min polynomial is the set of points where the max/min is attained by more than one of the "monomial" terms of the polynomial.



Tropical curve of the max-polynomial

$$p(x,y) = \max(2x,y,c)$$



Tropical curve of the min-polynomial

$$p'(x,y) = \min(2x,y,c)$$

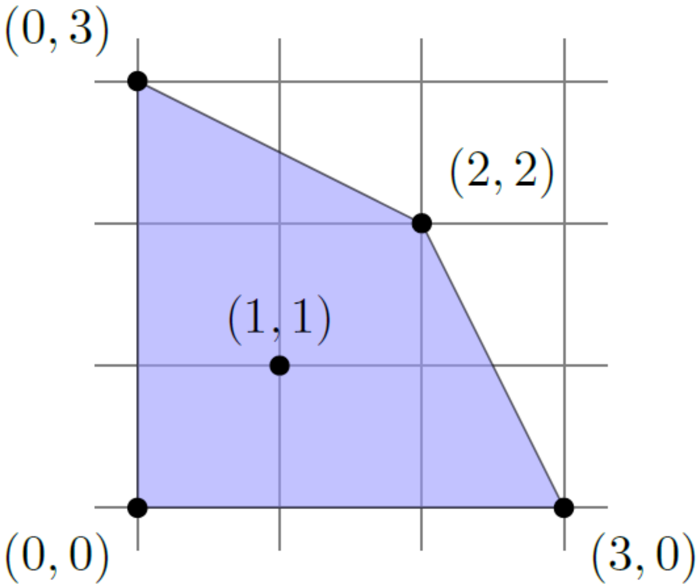
Newton Polytope of Tropical Polynomial

Max polynomial

$$p(\boldsymbol{x}) = \max_{i \in 1,2,\ldots,k} \{c_{i1}x_1 + c_{i2}x_2 + \cdots + c_{in}x_n\} = \bigvee_{i=1}^{k} \boldsymbol{c}_i^T \boldsymbol{x}$$

Newton polytope $N(p)$ of max polynomial p is the convex hull of its coefficients' vectors.

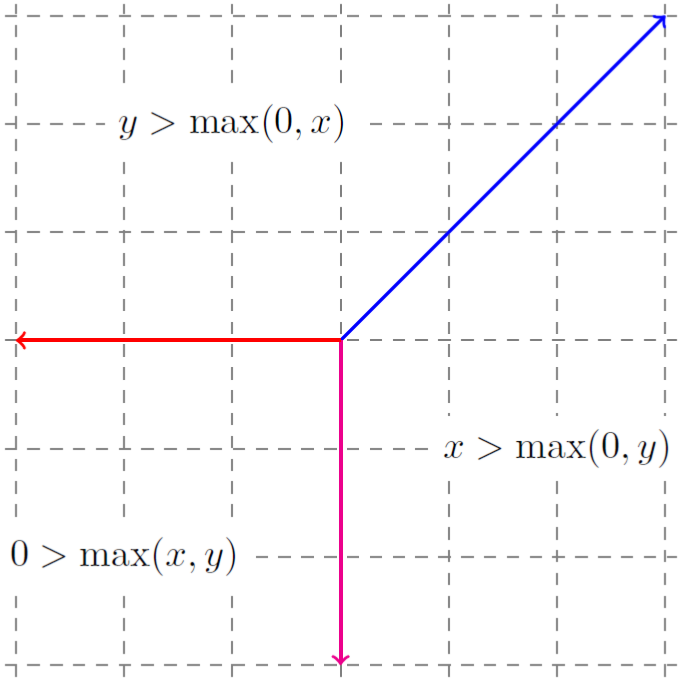
$$p(\boldsymbol{x}) = \max(0, x_1 + x_2, 2x_1 + 2x_2, 3x_1, 3x_2)$$



Tropical Curve vs Newton Polytope

Max polynomial: $p(x,y) = \max(x,y,0)$

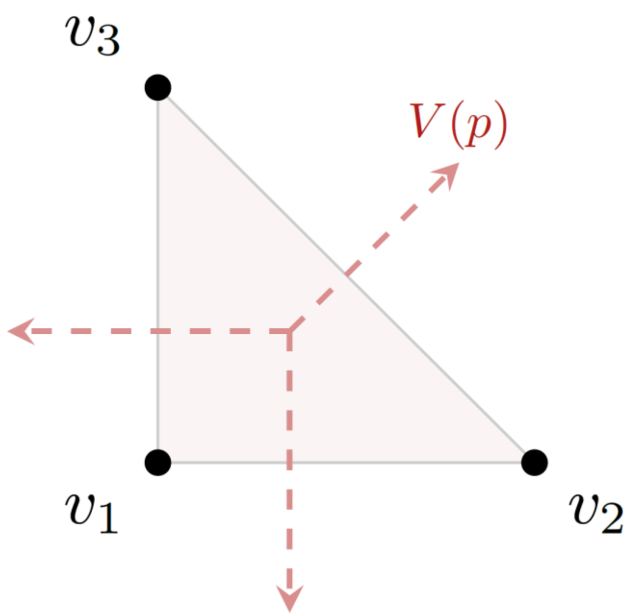
Tropical curve ("Zero locus") $V(p)$ of a max polynomial p is the set of points where the max is attained by more than two polynomial terms.



Tropical curve $V(p)$
of $p(x,y) = \max(x,y,0)$

Newton polytope $N(p)$ of max polynomial p is the convex hull of its coefficients' vectors.

$$\mathcal{N}(p) = \text{conv}\,\{v_1, v_2, v_3\}$$



Duality between Newton polytope $N(p)$
and tropical curve $V(p)$

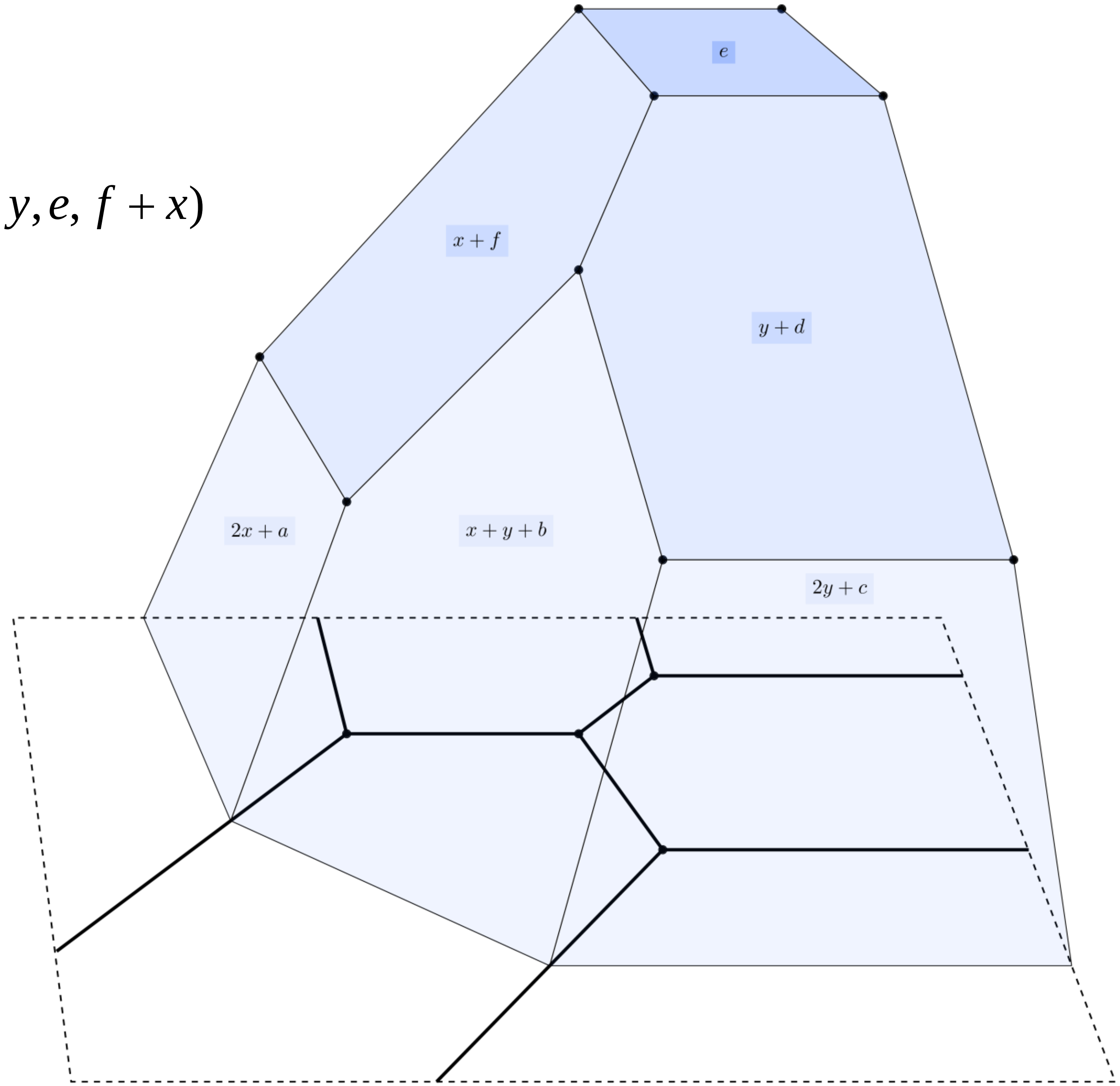
Graph and Tropical Curve of a tropical "Conic" polynomial

Tropical Polynomial of degree 2 in two variables

classical: $"ax^2 + bxy + cy^2 + dy + e + fx"$

tropical: $p(x, y) = \min(a + 2x, b + x + y, c + 2y, d + y, e, f + x)$

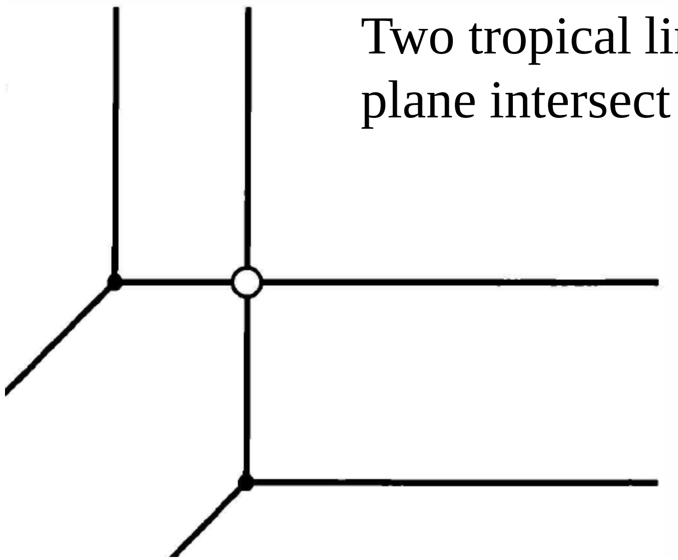
Graph ("tent") of $p(x,y)$
and
its **Tropical Curve** = set of (x,y) points where the min is attained by more than one terms.



Tropical Curves of Min-plus Polynomials on the plane

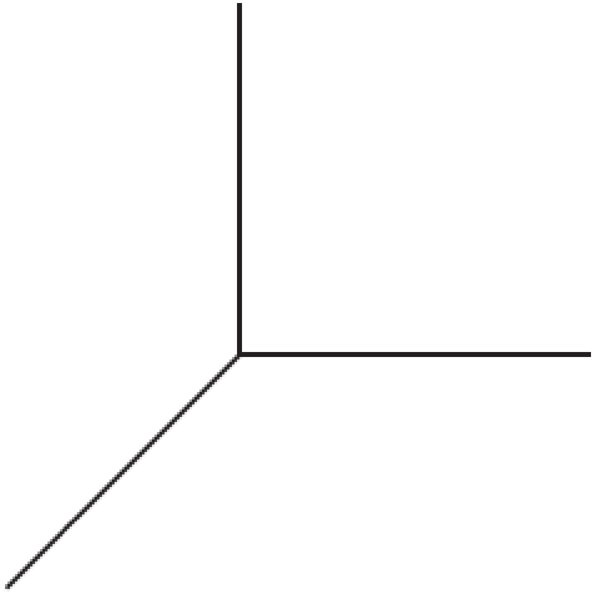
Min Polynomial of degree 1 in two variables

$$p(x, y) = \min(a + x, b + y, c)$$
$$= (a + x) \wedge (b + y) \wedge c$$

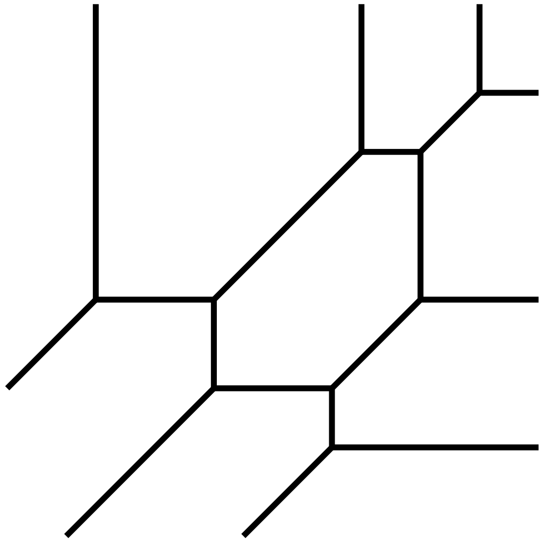


Two tropical lines on the plane intersect at one point

Tropical curve of $p(x,y)$



Cubic tropical curve



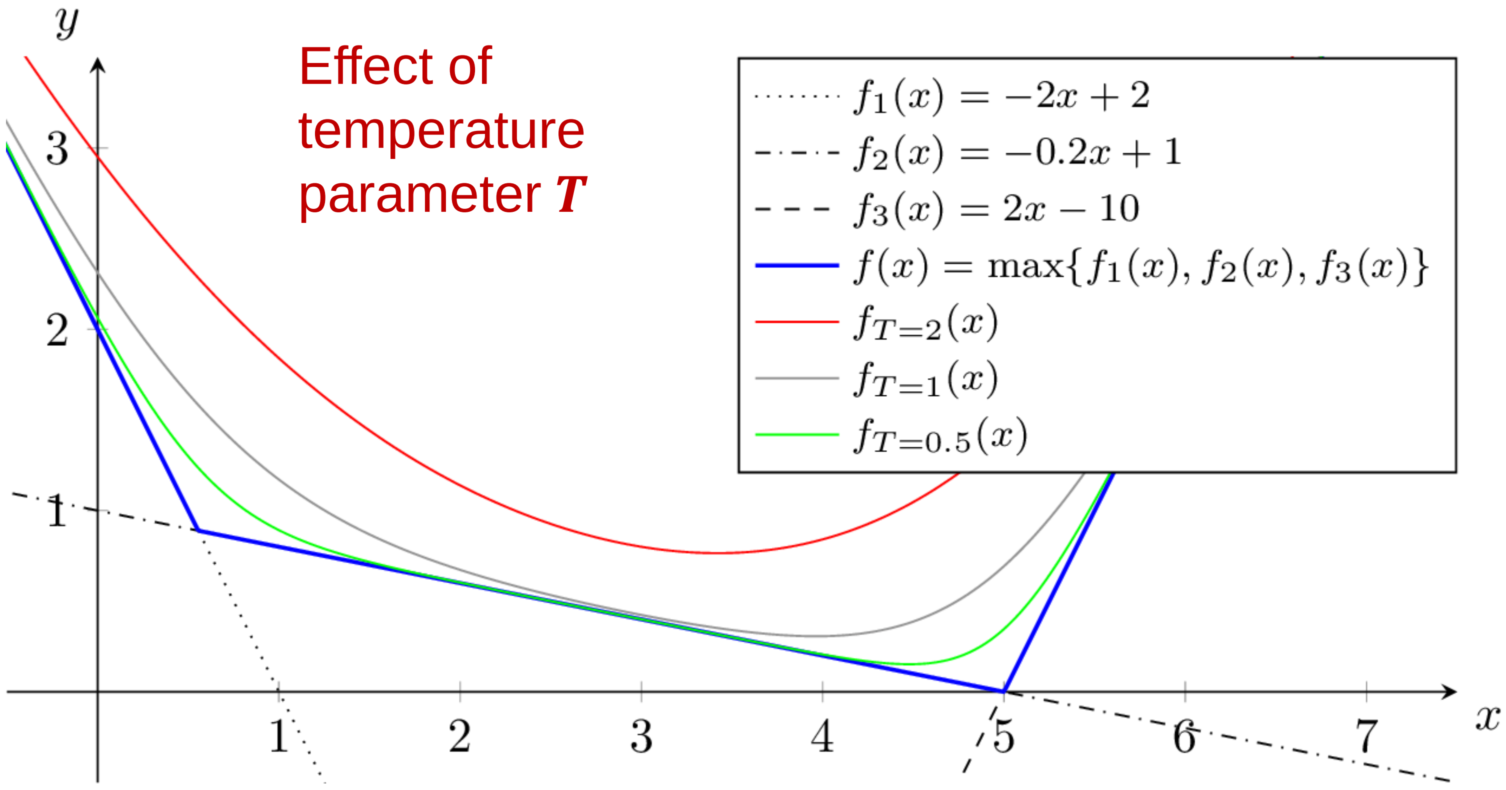
Log-Sum-Exp (LSE) approximation

(Maslov "Dequantization" in idempotent mathematics [Maslov 1987, Litvinov 2007])

$$\lim_{T \downarrow 0} T \cdot \log(e^{a/T} + e^{b/T}) = \max(a,b)$$

$$\lim_{T \downarrow 0} (-T) \log(e^{-a/T} + e^{-b/T}) = \min(a,b)$$

Effect of temperature parameter ***T***



Obtain Tropical Polynomials via Dequantization

Classic polynomial: $f(\mathbf{u}) = \sum_{k=1}^{K} c_k u_1^{a_{k1}} u_2^{a_{k2}} \cdots u_n^{a_{kn}}, \quad \mathbf{u} = (u_1, u_2, \ldots, u_n)$

Posynomial if $c_k > 0$, $\mathbf{a}_k = (a_{k1}, \ldots, a_{kn}) \in \mathbb{R}^n$, $\mathbf{u} > 0$;

Log-Sum-Exp (Viro's "logarithmic paper" [Viro 2001]):

$\mathbf{x} = \log(\mathbf{u}), \quad b_k = \log(c_k)$

$$\lim_{T \downarrow 0} T \cdot \log f(e^{\mathbf{x}/T}) = \lim_{T \downarrow 0} T \cdot \log \sum_{k=1}^{K} \exp(\langle \mathbf{a}_k, \mathbf{x}/T \rangle + b_k / T) \rightarrow$$

Tropical (max-plus) P**olynomial** = Piecewise-Linear Function

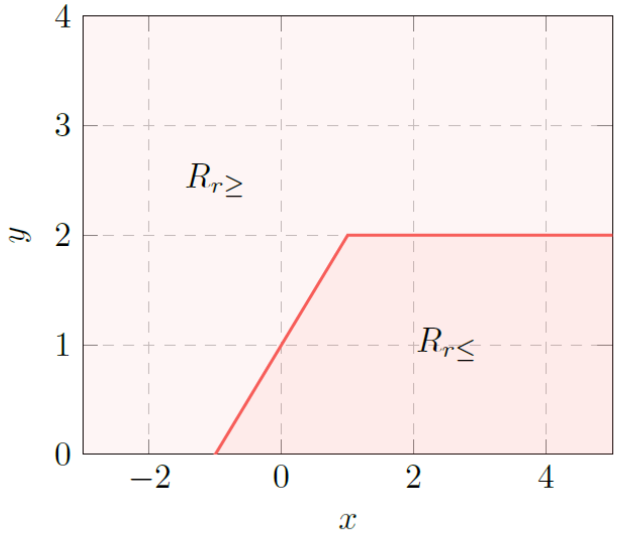
$$p(\mathbf{x}) = \underset{k=1}{\overset{K}{\mathrm{MAX}}} \{\langle \mathbf{a}_k, \mathbf{x} \rangle + b_k\} = \underset{k=1}{\overset{K}{\mathrm{MAX}}} \{a_{k1} x_1 + \cdots + a_{kn} x_n + b_k\}$$

Tropical Half-spaces and Polytopes in 2D

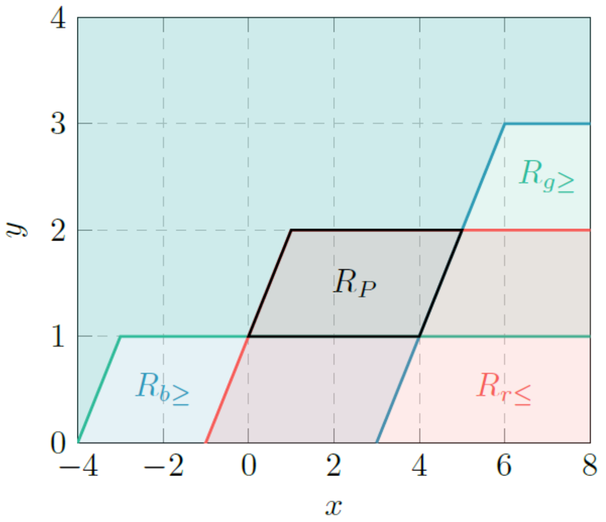
Tropical (affine) Half-space of $\mathbb{R}^n_{\max}$

[Gaubert & Katz 2011]

$$\mathscr{T}(\mathbf{a},\mathbf{b}) \triangleq \{\mathbf{x} \in \mathbb{R}^n_{\max} : \max(a_{n+1}, \bigvee_{i=1}^{n} a_i + x_i) \leq \max(b_{n+1}, \bigvee_{i=1}^{n} b_i + x_i)\}$$



(a) Single region



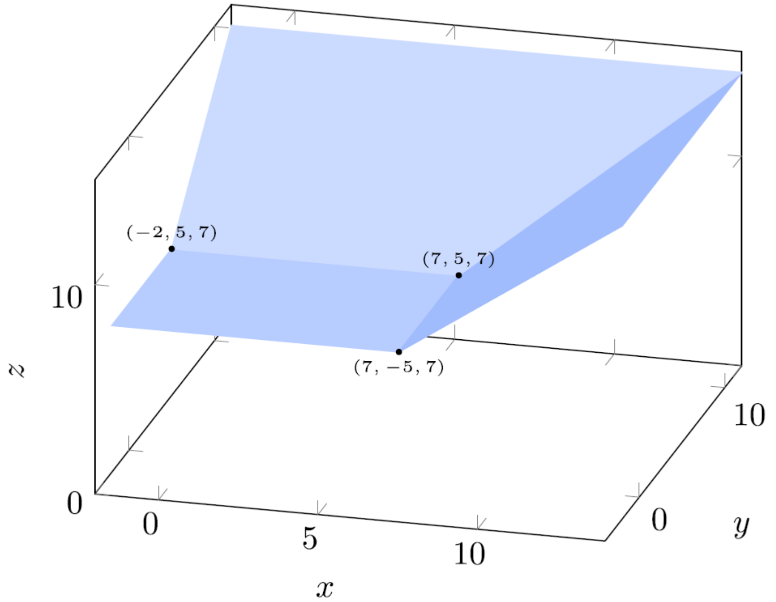
(b) Multiple regions

The region separating boundaries are tropical lines (or hyper-planes).

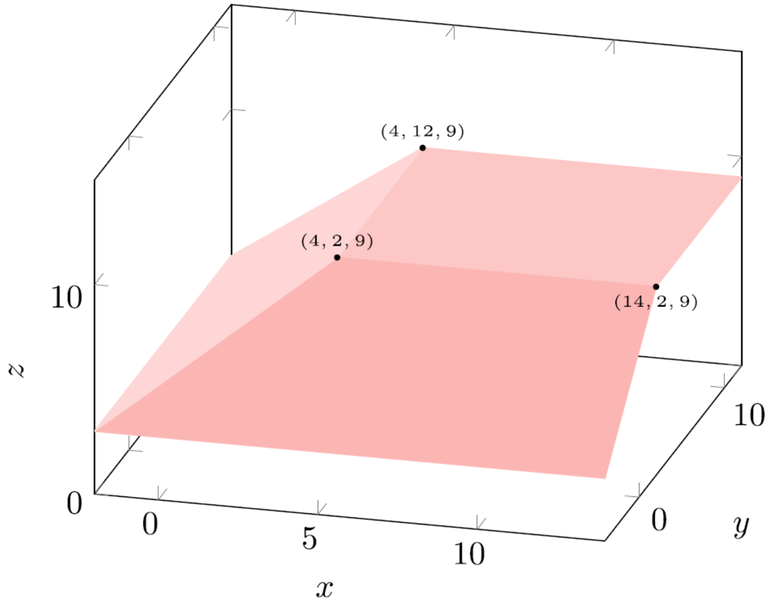
Tropical **Polyhedra** are formed from finite intersections of tropical half-spaces. **Polytopes** are compact polyhedra.

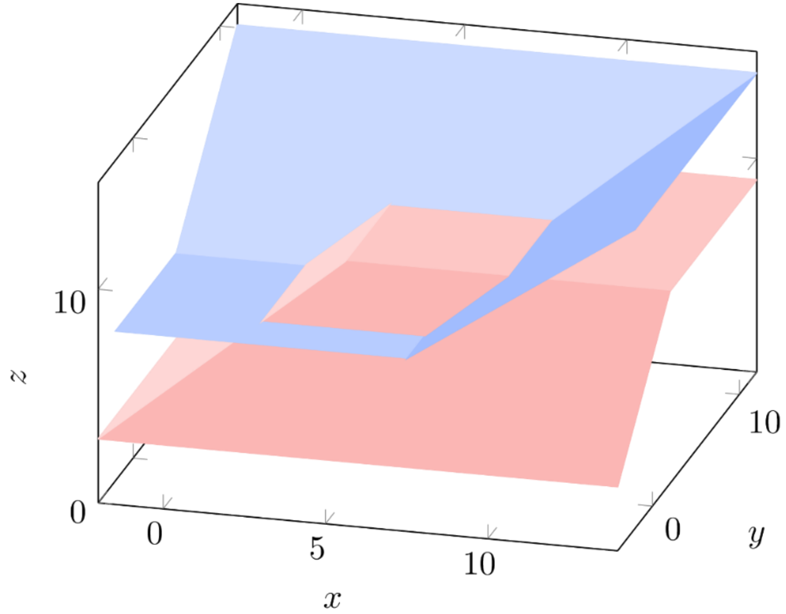
Tropical Halfspaces and Polyhedra in 3D

$$f(x,y) = \max(x, 2+y, 7)$$



$$g(x,y) = \min(5+x, 7+y, 9)$$





(Extended) Newton Polytope

Let $p(\boldsymbol{x}) = \max_{i=1,\dots,k} (\boldsymbol{a}_i^T \boldsymbol{x} + b_i)$ be a max-polynomial.

<u>Definition ((Extended) Newton Polytope):</u> We define as the (Extended) Newton Polytope of p the following:

$$\mathrm{Newt}(p) = \mathrm{conv}\{\boldsymbol{a}_i, i = 1, \dots, k\}$$

$$\mathrm{ENewt}(p) = \mathrm{conv}\{(\boldsymbol{a}_i, b_i), i = 1, \dots, k\}$$

where conv denotes the convex hull of the given set.

<u>Theorem</u> <u>[Charisopoulos & Maragos, 2018; Zhang et al., 2018]:</u>

Max-polynomials with the same vertices in the upper hull of their Extended Newton Polytope correspond to the same function.

Examples of (Ext) Newton Polytopes

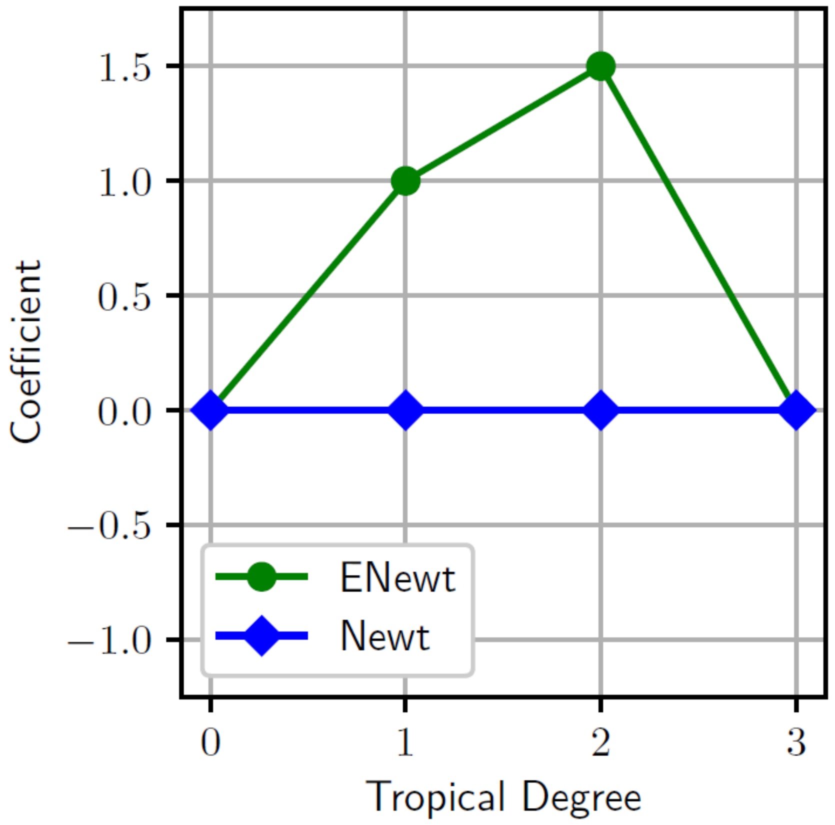


Figure: Polytopes of $\max(3x, 2x + 1.5, x + 1, 0)$.

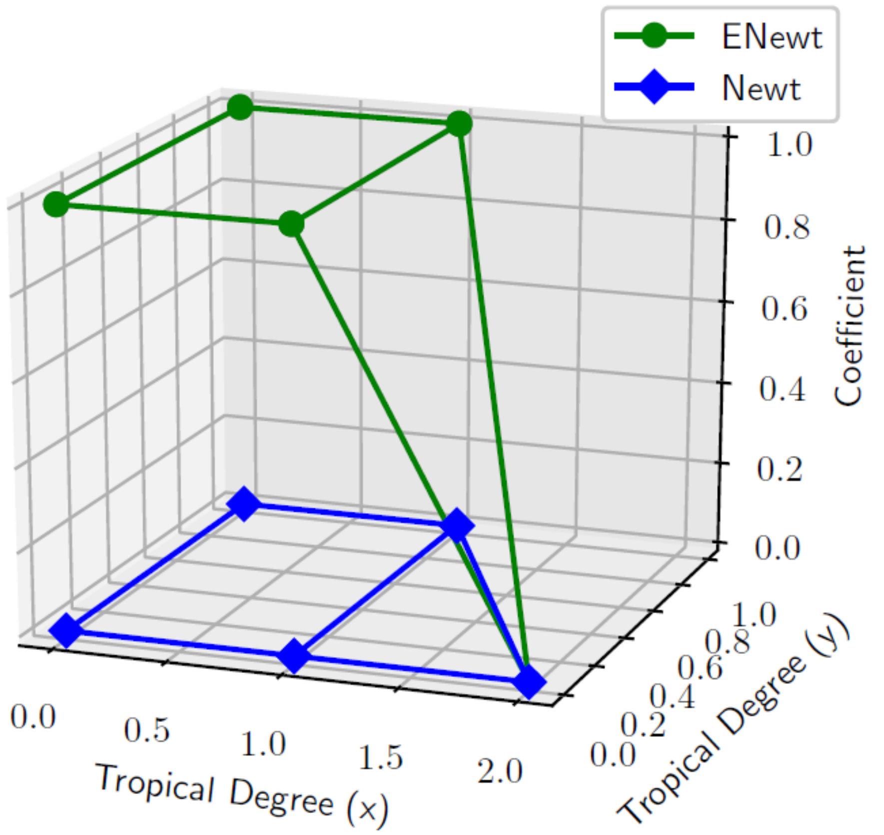


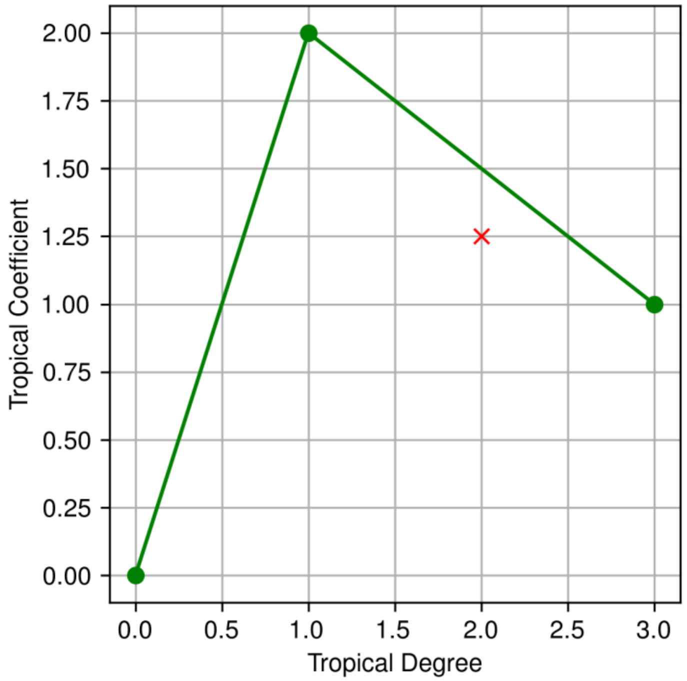
Figure: Polytopes of $\max(2x, x + y + 1, x + 1, y + 1, 1)$.

Newton Polytope and Max-polynomial Function

- "Upper" vertices of $\mathrm{ENewt}(p)$ define $p(x)$ as a function.
- Geometrically:

$$\max(3x + 1, 2x + 1.25, x + 2, 0)$$
$$= \max(3x + 1, x + 2, 0)$$

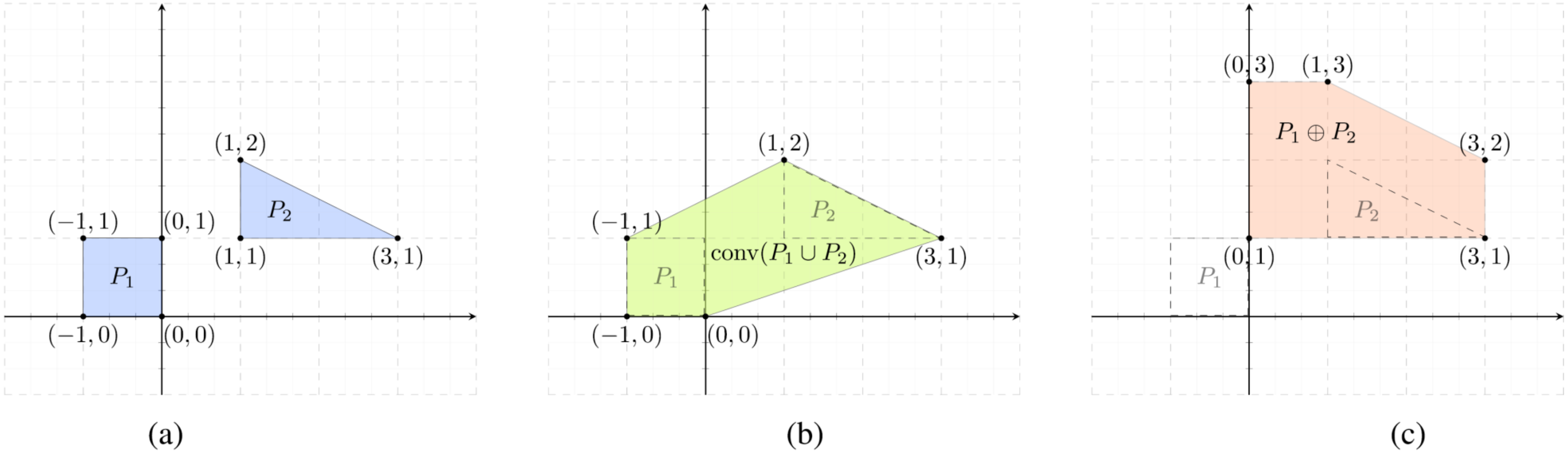
(extra point is not on the upper hull).



$\mathrm{ENewt}(p)$, $p(x) = \max(3x + 1, x + 2{,}0)$

Tropical Algebra of Max-plus Polynomials ←→ Tropical Geometry of their Newton Polytopes

$$\mathrm{Newt}\,(p_1 \vee p_2) = \mathrm{conv}\,(\mathrm{Newt}\,(p_1) \cup \mathrm{Newt}\,(p_2))$$
$$\mathrm{Newt}\,(p_1 + p_2) = \mathrm{Newt}\,(p_1) \oplus \mathrm{Newt}\,(p_2)$$



Newton polytopes of (a) two max-polynomials

$p_1(x,y) = max(0, -x, y, y-x)$ and $p_2(x,y) = max(x+y, 3x+y, x+2y)$,

(b) their $max(p_1 , p_2)$, and (c) their sum $p_1 + p_2$

Elements of Max-plus Algebra

(“*Linear algebra of Dynamic Programming & Combinatorics*”: [Butkovic 2010])

Some Earlier Special Cases and Applications

Research Areas using Max/Min(+) Algebra

- **Scheduling & Operations Research, Graphs**: **Minimax Algebra** [Cuninghame-Green 1979]: mainly Max-Plus.
- **Tropical Arithmetic: Min-plus/Max-plus Semirings** [I. Simon 1994; J.-E. Pin 1998]
- **Image & Vision, Nonlinear SP:** Image Algebra [Ritter et al, 1980s-90s], Math. Morphology [Serra 88; Heijmans & Ronse 1990s]. Morphological & Rank Filters, [Maragos & Schafer 1987]. Nonlinear Scale-Space PDEs [Brockett & Maragos 1992; Alvarez et al 1993]. Distance Transforms [Borgefors 1984; Felzenszwalb et al 2004].
- **Control**: Discrete-Event Dynamical Systems [Cohen et al 1985; Kamen 1993; Cassandras et al 2013; Heidergot et al 2006]. Dioid algebra [Cohen et al 1989; Baccelli et al 1992-2001; Gaubert & Max-plus Group 1997; Lahaye & Hardouin et al 2004; Gondran & Minoux 2008], Max-Linear Systems [Butkovic 2010, van den Boom & de Shutter 2012]. Optimization/Approximation on Semimodules [Cohen et al 2004, Akian et al 2011].
- **Speech & Language** Processing: Weighted Finite-StateAutomata/Transducers: Tropical Semiring Algorithms on Graphs [Mohri, Pereira et al, 1990s; Hori & Nakamura 2013].
- **Probabilistic Graphical Models**: Max-Sum and Max-Product algorithms in Belief Propagation [Pearl 1988; Bishop 2006; Felzenszwalb 2011].
- **Math-Physics**: Convex analysis & Optimization [Bellman & Karush 1960’s; Rockafellar 1970; Lucet 2010]. **Lattices** [Birkhoff 1967]. Residuation and Ordered Algebraic Structures [Blyth 2005].

  Idempotent Mathematics [Maslov 1987; Litvinov, Maslov et al 2000s].

Linear vs. Max-Plus Algebra: Scalar Operations

$$+ \longrightarrow \max$$

$$\times \longrightarrow +$$

Max-plus has properties similar to linear algebra:

- Commutativity: $a \vee b = b \vee a$
- Associativity: $a \vee (b \vee c) = (a \vee b) \vee c$
- Distributivity: $a + (b \vee c) = (a + b) \vee (a + c)$
- Idempotency: $3 \vee 3 = 3$
- Inverse?: $3 \vee x = 6 \Rightarrow x = 6$

  $3 \vee x = 3 \Rightarrow x = ?$

Max-plus Matrix Algebra

(Finite-dimensional Weighted Lattices)

- vector/matrix **'addition'** = pointwise max

$$\begin{aligned} \mathbf{x} \vee \mathbf{y} &= [x_1 \vee y_1, \ldots, x_n \vee y_n]^T \\ \mathbf{A} \vee \mathbf{B} &= [a_{ij} \vee b_{ij}] \end{aligned}$$

- vector/matrix **'dual addition'** = pointwise min

$$\begin{aligned} \mathbf{x} \wedge \mathbf{y} &= [x_1 \wedge y_1, \ldots, x_n \wedge y_n]^T \\ \mathbf{A} \wedge \mathbf{B} &= [a_{ij} \wedge b_{ij}] \end{aligned}$$

- vector/matrix **'multiplication by scalar'**

$$\begin{aligned} c + \mathbf{x} &= [c + x_1, \ldots, c + x_n]^T \\ c + \mathbf{A} &= [c + a_{ij}] \end{aligned}$$

- $(\max, +)$ **'matrix multiplication'**

$$[\mathbf{A} \boxplus \mathbf{B}]_{ij} = \bigvee_{k=1}^{n} a_{ik} + b_{kj}$$

- $(\min, +)$ **'matrix dual multiplication'**

$$[\mathbf{A} \boxplus' \mathbf{B}]_{ij} = \bigwedge_{k=1}^{n} a_{ik} + b_{kj}$$

Tropical Semirings versus Weighted Lattices

Weighted Lattice = Tropical Space

	Flat Lattice $(\mathbb{R} \cup \{-\infty, +\infty\}, \vee, \wedge)$	
Max-plus Semiring $(\mathbb{R} \cup \{-\infty\}, \vee, +)$	$(\mathbb{R} \cup \{-\infty\}, \max)$ is Idempotent Semigroup	$(\mathbb{R}, +)$ is Group. Addition $(+)$ distributes over $\vee$
Min-plus Semiring $(\mathbb{R} \cup \{+\infty\}, \wedge, +')$	$(\mathbb{R} \cup \{+\infty\}, \min)$ is Idempotent Semigroup	$(\mathbb{R}, +')$ is Group. Dual Addition $(+')$ distributes over $\wedge$
	Duality between $\vee$ and $\wedge$	

[P. Maragos, "*Dynamical Systems on Weighted Lattices: General Theory*", Math. Control, Signals and Systems, 2017.]

Linear and Nonlinear Spaces

Linear spaces (Vector Spaces):

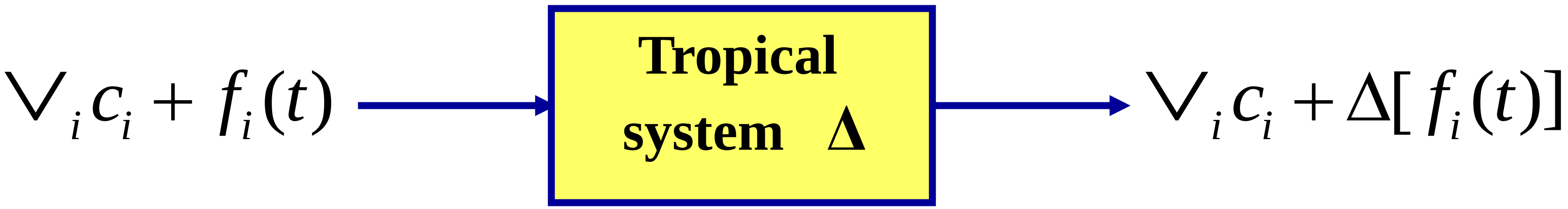
Signal Superposition (+): $f(t)+g(t)$

Scaling (x): $c \cdot f(t)$

Nonlinear spaces (Tropical spaces = Weighted Lattices):

Signal Superposition : **max:** $f(t) \vee g(t)$ **min:** $f(t) \wedge g(t)$

Scaling (+): $c + f(t)$



Morphological Operators on Lattices

($\leq$ = partial ordering, V = supremum, Λ = infimum)

- ψ is **increasing** iff $f \leq g \Rightarrow \psi(f) \leq \psi(g)$.
- δ is **dilation** iff $\delta(\vee_i f_i) = \vee_i \delta(f_i)$.
- ε is **erosion** iff $\varepsilon(\wedge_i f_i) = \wedge_i \varepsilon(f_i)$.
- α is **opening** iff increasing and antiextensive $(\alpha(f) \leq f)$,
  and idempotent $(\alpha = \alpha^2)$: lattice projection
- β is **closing** iff increasing and extensive $(\beta(f) \geq f)$,
  and idempotent $(\beta = \beta^2)$: lattice projection
- (δ, ε) is **adjunction** iff $\delta(f) \leq g \Leftrightarrow f \leq \varepsilon(g)$ (Galois connection)

Then: δ is dilation, ε is erosion,

$\delta\varepsilon$ is opening, $\varepsilon\delta$ is closing.

[Serra 1988; Heijmans & Ronse 1990]

Minkowski-Hadwiger Morphological Set Operators

Translation: $B_{+z} = \{b + z : b \in B\}$

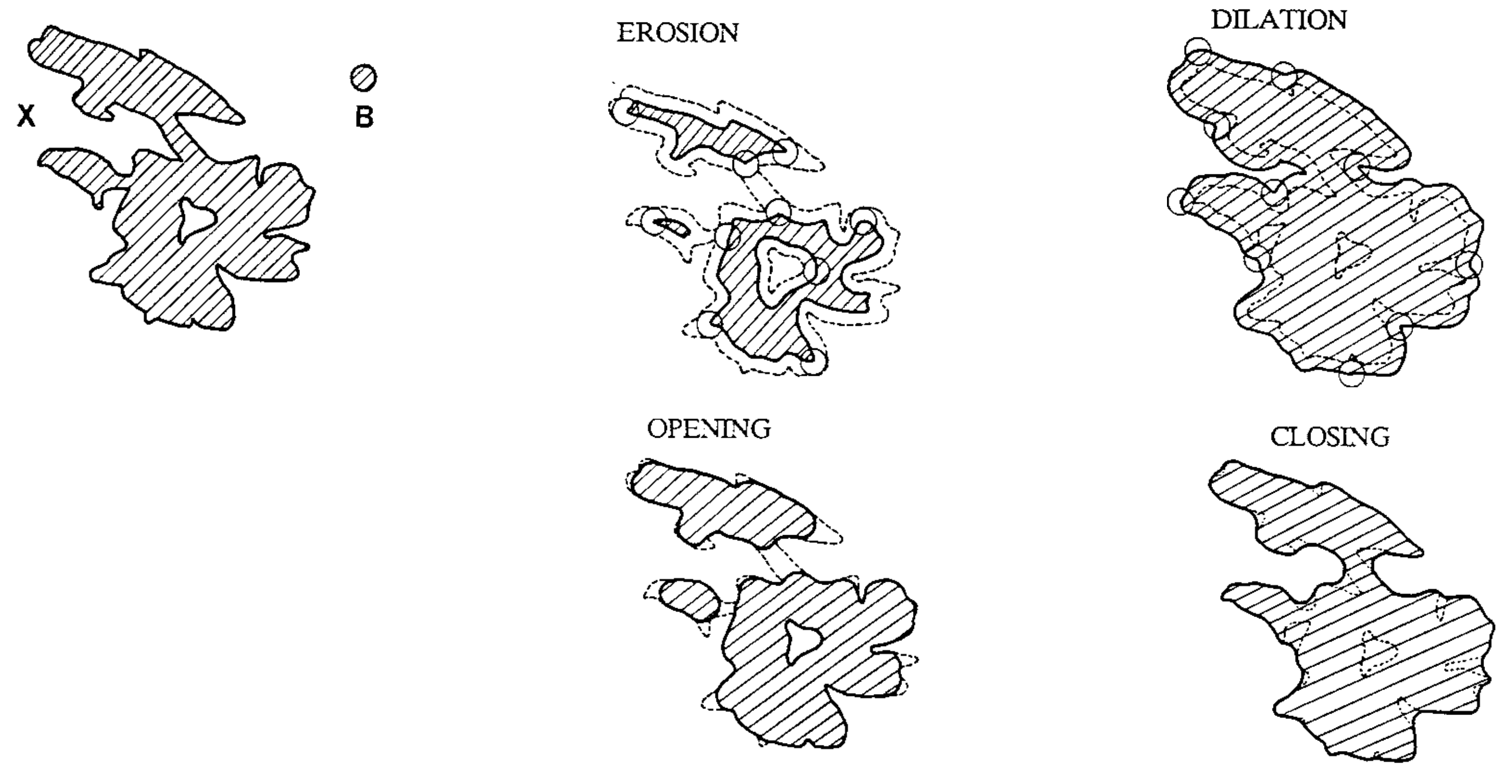
Symmetric: $B^s = \{-b : b \in B\}$

Dilation (Minkowski addition): $X \oplus B = \{z : (B^s)_{+z} \cap X \neq \varnothing\} = \bigcup_{b \in B} X_{+b}$

Erosion (Minkowski subtraction): $X \ominus B = \{z : B_z \subseteq X\} = \bigcap_{b \in B} X_{-b}$

Hadwiger Opening: $X \circ B = (X \ominus B) \oplus B$

Closing: $X \bullet B = (X \oplus B) \ominus B$



Max/Min-plus Convolutions and Filters-Projections

Max-plus Convolution (**Dilation**) by a square (flat g)
(= Max Pooling in CNNs)

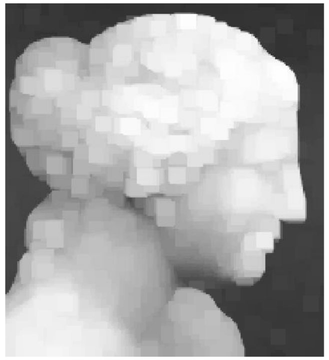
$$(f \oplus g)(x) = \bigvee_y f(y) + g(x-y)$$

Adjoint Min-plus Correlation (**Erosion**)

$$(f \ominus g)(x) = \bigwedge_y f(y) - g(y-x)$$

IMAGE

DILATION 9x9



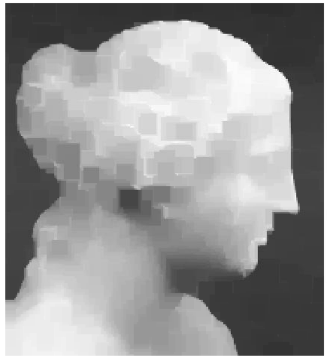
EROSION 9x9

Serial compositions of max-convolution and adjoint min-plus correlation: **Opening**, **Closing**

$$f \circ g \triangleq (f \ominus g) \oplus g \qquad f \bullet g \triangleq (f \oplus g) \ominus g$$

OPENING 9x9

CLOSING 9x9



Idempotent Operators = **Projections**
on Nonlinear Spaces (Weighted Lattices)

$$(f \circ g) \circ g = f \circ g$$
$$(f \bullet g) \bullet g = f \bullet g$$

Examples of Adjunctions

- **Set Operator Adjunction:** Minkowski set addition $\oplus$ and subtraction $\ominus$: for $X, B \subseteq \mathbb{R}^d$

$$\begin{aligned}\delta_B(X) = X \oplus B &:= \{\mathbf{x}+\mathbf{b} \in \mathbb{R}^d : \mathbf{x} \in X,\ \mathbf{b} \in B\} \\ \varepsilon_B(X) = X \ominus B &:= \{\mathbf{x}-\mathbf{b} \in \mathbb{R}^d : \mathbf{x} \in X,\ \mathbf{b} \in B\}\end{aligned}$$

- **Vector Operator Adjunction:** max-plus vector multiplication by matrix $\mathbf{A} \in \overline{\mathbb{R}}^{m\times n}$ and min-plus vector multiplication by matrix $\mathbf{A}^* = -\mathbf{A}^T$:

$$\begin{aligned}\delta_{\mathbf{A}}(\mathbf{x}) &= \mathbf{A} \boxplus \mathbf{x}, \quad [\delta_{\mathbf{A}}(\mathbf{x})]_i = \bigvee_{j=1}^{n} a_{ij} + x_j \\ \varepsilon_{\mathbf{A}}(\mathbf{y}) &= \mathbf{A}^* \boxplus' \mathbf{y}, \quad [\varepsilon_A(\mathbf{y})]_j = \bigwedge_{i=1}^{m} y_i - a_{ij}\end{aligned}$$

- **Signal Operator Adjunction:** max-plus convolution of $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ with k and min-plus convolution of $g(\mathbf{x})$ with $-k(-\mathbf{x})$:

$$\begin{aligned}\delta_k(f)(\mathbf{x}) = f \oplus k(\mathbf{x}) &:= \bigvee_{\mathbf{y}} \{f(\mathbf{y}-\mathbf{x}) + k(\mathbf{y})\} \\ \varepsilon_k(g)(\mathbf{x}) = g \ominus k(\mathbf{x}) &:= \bigwedge_{\mathbf{y}} \{g(\mathbf{x}+\mathbf{y}) - k(\mathbf{y})\}\end{aligned}$$

Operation	Meaning
$\bigvee$	Maximum/Supremum: applies for scalars, vectors and matrices
$\bigwedge$	Minimum/Infimum: applies for scalars, vectors and matrices
$\boxed{\star}$ ($\boxed{\star}'$)	General max-$\star$ (min-$\star'$) matrix multiplication
$\boxplus$ ($\boxplus'$)	Max-sum (min-sum) matrix multiplication
$\boxtimes$ ($\boxtimes'$)	Max-product (min-product) matrix multiplication
$\circledast$ ($\circledast'$)	General max-$\star$ (min-$\star'$) signal convolution
$\oplus$ ($\oplus'$)	Max-sum (min-sum) signal convolution
$\otimes$ ($\otimes'$)	Max-product (min-product) signal convolution

max-sum and min-sum

matrix multiplications

$$\boldsymbol{C} = \boldsymbol{A} \boxplus \boldsymbol{B} = [c_{ij}] \quad , \quad c_{ij} = \bigvee_{k=1}^{n} a_{ik} + b_{kj}$$

$$\boldsymbol{C} = \boldsymbol{A} \boxplus' \boldsymbol{B} = [c_{ij}] \quad , \quad c_{ij} = \bigwedge_{k=1}^{n} a_{ik} + b_{kj}$$

signal convolutions

$$(f \oplus h)(t) \quad = \quad \bigvee_{k=-\infty}^{+\infty} f(t-k) + h(k)$$

$$(f \oplus' h)(t) \quad = \quad \bigwedge_{k=-\infty}^{+\infty} f(t-k) + h(k)$$

Solve Max-plus Equations via Adjunctions

- **Problems**:

$$(1)\quad \text{Exact problem: Solve } \delta_{\mathbf{A}}(\mathbf{x}) = \overbrace{\mathbf{A} \boxplus \mathbf{x}}^{\text{max-plus}} = \mathbf{b}, \quad \mathbf{A} \in \overline{\mathbb{R}}^{m\times n}, \quad \mathbf{b} \in \overline{\mathbb{R}}^{m}$$

$$(2)\quad \text{Approximate Constrained: Min } \|\mathbf{A} \boxplus \mathbf{x} - \mathbf{b}\|_{p=1\ldots\infty} \text{ s.t. } \mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$$

- **Theorem**: The **greatest (sub)solution** of (1) and unique solution of (2) is

$$\hat{\mathbf{x}} = \varepsilon_{\mathbf{A}}(\mathbf{b}) = \mathbf{A}^* \boxplus' \mathbf{b}, \quad [\hat{\mathbf{x}}]_j = \bigwedge_{i=1}^{m} b_i - a_{ij}, \quad \mathbf{A}^* \triangleq -\mathbf{A}^T$$

and yields the **Greatest Lower Estimate (GLE)** of data $\mathbf{b}$:

$$\delta_{\mathbf{A}}(\varepsilon_{\mathbf{A}}(\mathbf{b})) = \underbrace{\mathbf{A} \boxplus \Big(\underbrace{\mathbf{A}^* \boxplus' \mathbf{b}}_{\text{min-plus}} \Big)}_{\text{max-plus matrix product}} \leq \mathbf{b}$$

- **Geometry**: Operators δ, ε are vector dilation and erosion, and the GLE $\mathbf{b} \mapsto \delta(\varepsilon(\mathbf{b}))$ is an opening (lattice projection).

- **Complexity**: $O(mn)$

Adjunction versus Residuation pairs

- An increasing operator $\psi : \mathcal{L} \to \mathcal{M}$ between complete lattices is called **residuated** if there exists an increasing operator $\psi^\sharp : \mathcal{M} \to \mathcal{L}$ such that

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

  $\psi^\sharp$ is called the **residual** of ψ, is unique, and closest to being an inverse of ψ.

- A residuation pair $(\psi, \psi^\sharp)$ can solve **inverse problems** $\psi(X) = Y$ 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**:

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

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

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

  δ is a **dilation** and ε is an **erosion**.
  Each dilation δ corresponds to a unique *adjoint erosion*

$$\varepsilon(Y) = \delta^\sharp(Y) = \bigvee\{X : \delta(X) \leq Y\}$$

- Adjunction $\iff$ Residuation *iff* $\psi = \delta$ and $\psi^\sharp = \varepsilon$.

- Viewing (δ, ε) as adjunction instead of residuation offers *geometric intuition*.

Some Earlier Special Cases
of Max-plus Algebra and Applications

Linear versus Max-Plus Systems

- **State space representation**: linear vs. max-plus

$$x(k) = Ax(k-1) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

$$x(k) = A \boxplus x(k-1) \vee B \boxplus u(k)$$
$$y(k) = C \boxplus x(k) \vee D \boxplus u(k)$$

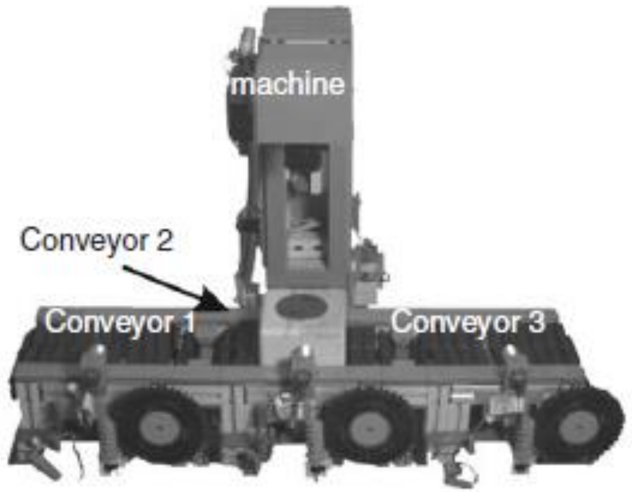
- **Matrix products**
  - Linear: $[AB]_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$
  - Max-plus: $[A \boxplus B]_{ij} = \bigvee_{k=1}^{n} a_{ik} + b_{kj}$
- **Example**

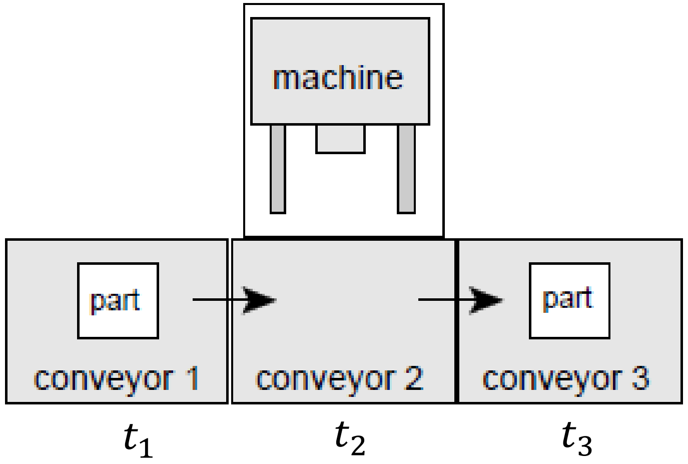
$$\begin{bmatrix} 4 & -1 \\ 2 & -\infty \end{bmatrix} \boxplus \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \left\{ \begin{array}{r} \max(x+4, y-1) = 3 \\ x+2 = 1 \end{array} \right\} \Longrightarrow \begin{array}{c} x = -1 \\ y \leq 4 \end{array}$$

- What can we model with max-plus systems?

Automated Manufacturing as Max-plus System

Discrete event systems (*)





$x_i(k)$**:** time product k enters conveyor i
$u(k)$: time we put product k in conveyor 1
t_i: conveyor i waiting time
Only one product in a conveyor during each cycle

$$x_1(k) = \max\left(x_1(k-1) + t_1, u(k)\right)$$

$$x_2(k) = \max\left(x_1(k) + t_1, x_2(k-1) + t_2\right)$$

$$x_3(k) = \max\left(x_2(k) + t_2, x_3(k-1) + t_3\right)$$

$$A = \begin{bmatrix} t_1 & -\infty & -\infty \\ 2t_1 & t_2 & -\infty \\ 2t_1 + t_2 & 2t_2 & t_3 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ t_1 \\ t_1 + t_2 \end{bmatrix}$$

$$x(k) = A \boxplus x(k-1) \vee B \boxplus u(k)$$

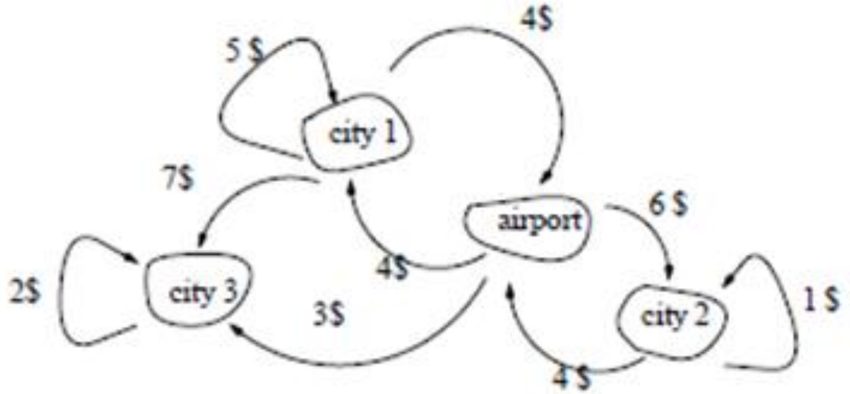
(*) Example from: [G. Schullerus, V. Krebs, B. De Schutter & T. van den Boom, "*Input signal design for identification of max-plus-linear systems*", Automatica 2006.]

Longest/Shortest Paths as Max/Min-plus Systems

Dynamic Programming

- **Taxi drivers (*)**

$$\boldsymbol{x}(k+1) = \boldsymbol{A}^T \boxplus \boldsymbol{x}(k)$$



$$A^T = \begin{bmatrix} 5 & 4 & -\infty & 7 \\ 4 & -\infty & 6 & 3 \\ -\infty & 4 & 1 & -\infty \\ -\infty & -\infty & -\infty & 2 \end{bmatrix}$$

$$money_i(k) = \max_j \left(money_j(k-1) + a_{ji} \right)$$

x_1, x_2, x_3, x_4 correspond to city 1, airport, city 2 and city 3

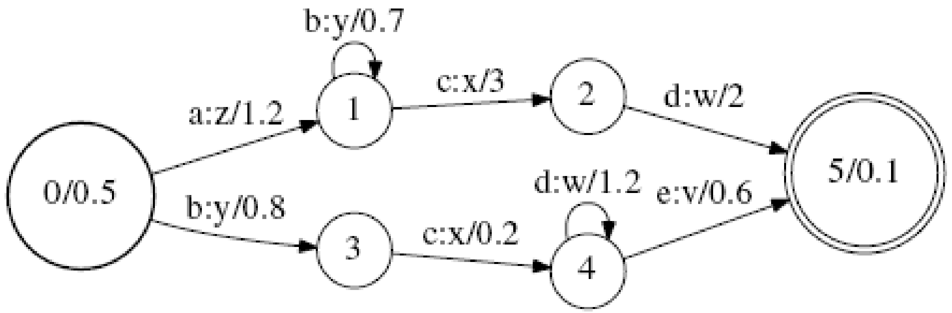
(*) Example from:
[S. Gaubert and Max-Plus group, "*Methods and applications of (max,+) linear algebra*", STACS 1997.]

WFSTs for Speech Recognition: Tropical (Min-Plus) Algebra

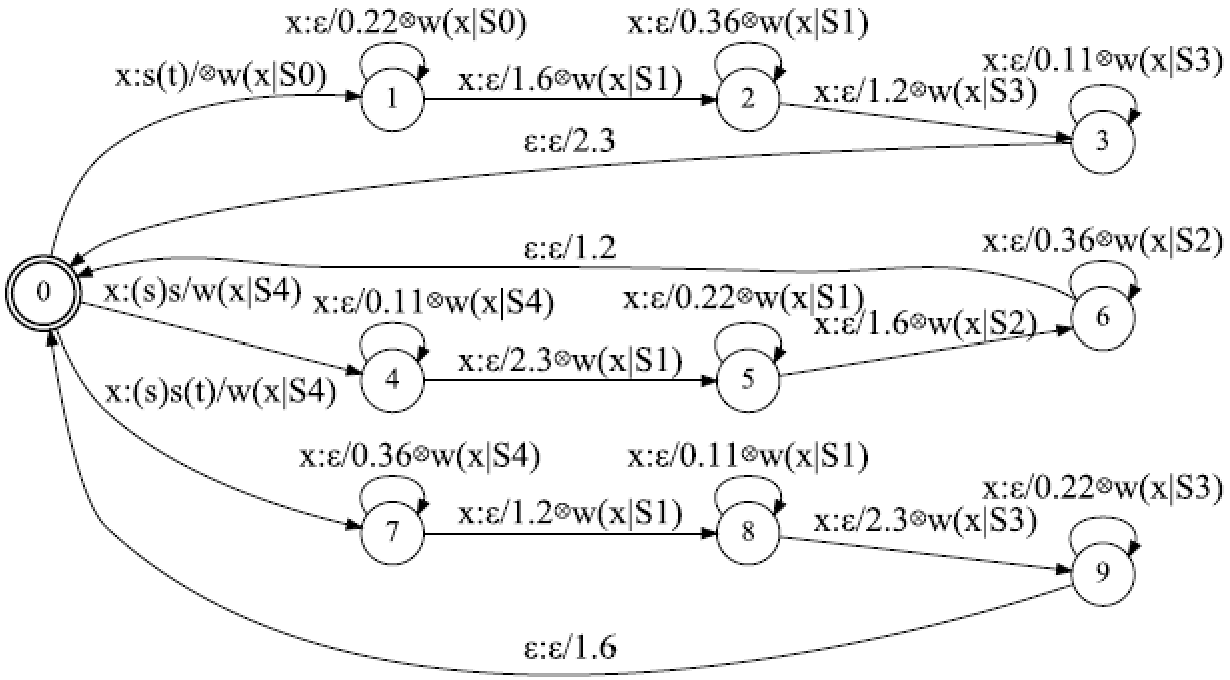
Weighted Finite State Transducer (WFST)

[Mohri, Pereira & Ripley, CSL 2002]

[Hori and Nakamura, 2013]



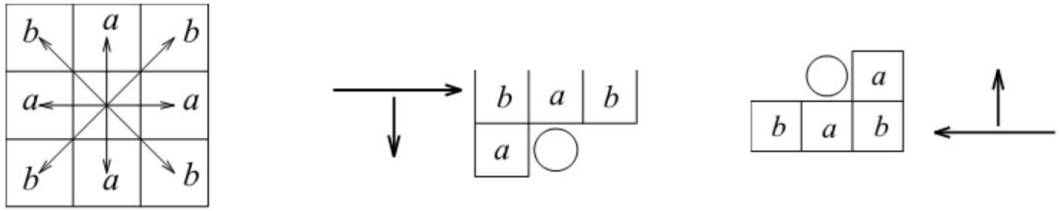
HMM Transducer: converts an input speech signal into a seq of context-dependent phone units

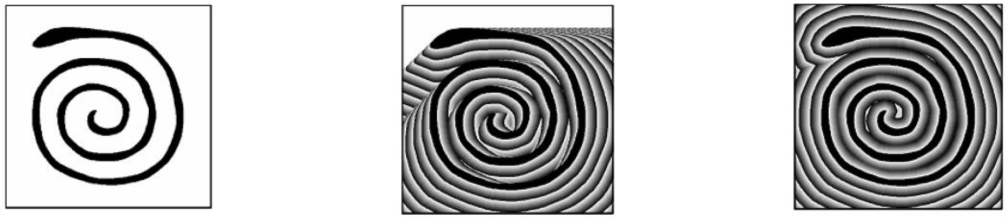


Distance Computation with Min-plus Difference Eqns

Two – Pass Algorithm

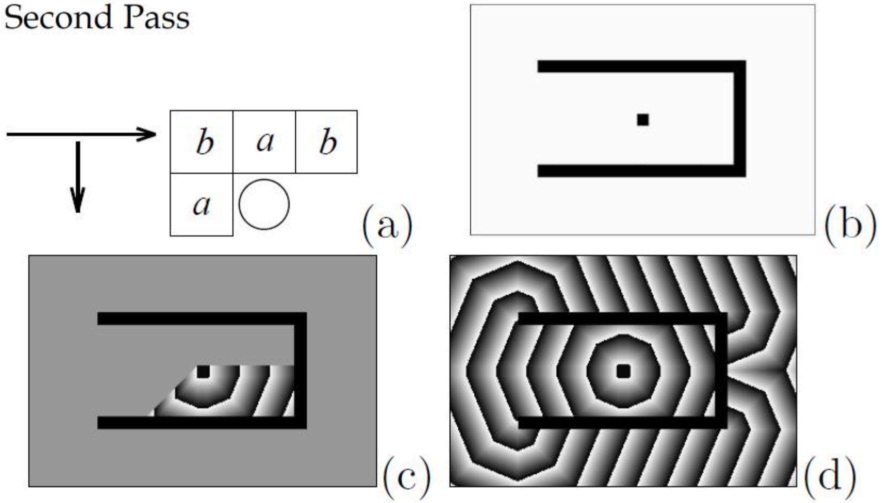
$$
\begin{aligned}
u_1[i,j] &= \min(u_1[i,j-1]+a, u_1[i-1,j]+a,\\
&\quad u_1[i-1,j-1]+b, u_1[i-1,j+1]+b, u_0[i,j])\\
u_2[i,j] &= \min(u_2[i,j+1]+a, u_2[i+1,j]+a,\\
&\quad u_2[i+1,j+1]+b, u_2[i+1,j-1]+b, u_1[i,j])
\end{aligned}
$$





Initial Image First Pass Second Pass

Sequential Distance Computation with Obstacles

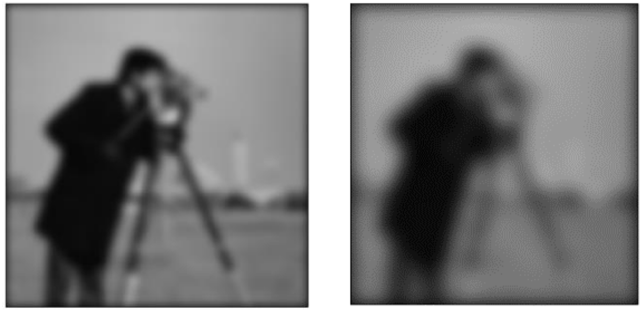


Gaussian Scale-Space → Maslov Dequantiz → Dilation/Erosion Scale-Space

Heat PDE: $$\frac{\partial u}{\partial t} = \frac{h}{2}\frac{\partial^2 u}{\partial x^2}$$

F

$\Rightarrow$



Multiscale Gaussian Blurring

Substitution (LSE - LogSumExpon): $u = e^{-W/h}$

Hopf's eqn: $$\frac{\partial W}{\partial t} + \frac{1}{2}\left(\frac{\partial W}{\partial x}\right)^2 - \frac{h}{2}\frac{\partial^2 W}{\partial x^2} = 0$$

Dequantization: $$\lim_{h \to 0} h \cdot \log(e^{-a/h} + e^{-b/h}) = \min(a, b)$$

Multiscale Max/Min Pooling

HJE: $$\frac{\partial S}{\partial t} + \frac{1}{2}\left(\frac{\partial S}{\partial x}\right)^2 = 0$$

$\Rightarrow$

- Erosion (-F)

$$\Rightarrow S(x,t) = \text{Multiscale Erosion by Parabola } (-x^2/2t)$$

Tropical Geometry of Neural Nets with Piecewise-Linear Activations

Main References:

1. Charisopoulos, V., & Maragos, P. (2017, May). *Morphological perceptrons: geometry and training algorithms*, ISMM '17.
2. Charisopoulos, V., & Maragos, P. (2018). A Tropical Approach to Neural Networks with Piecewise Linear Activations. arXiv:1805.08749.
3. Zhang, Liwen and Naitzat, Gregory and Lim, Lek-Heng. *Tropical geometry of deep neural networks*, Proc. ICML(35) 2018.

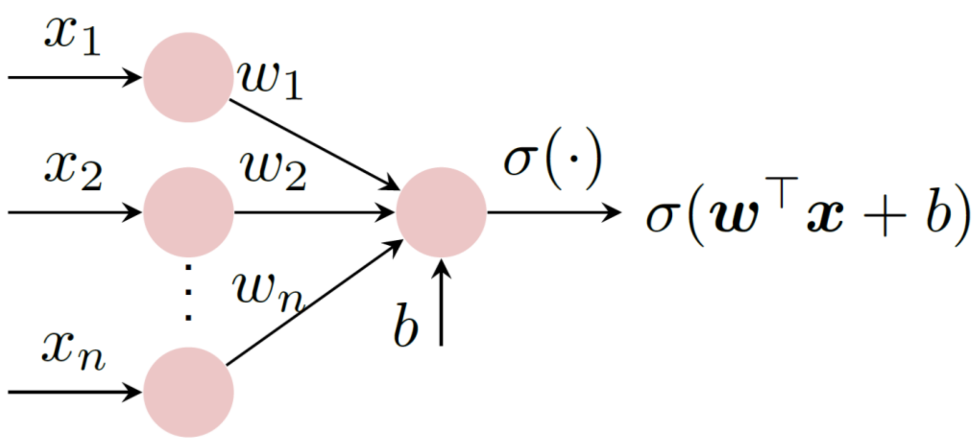
Related:

- M. Alfarra et al, *On the decision boundaries of neural networks: A tropical geometry perspective*, arXiv 2020.
- A. Humayun et al., *SplineCam: Visualization of Deep Network Geometry and Decision Boundaries*, CVPR 2023.

NNs with PWL functions

Piecewise-linear functions used as *activation* functions σ:

1. **ReLU**: $\max(0, v)$ or $\max(\alpha v, v)$, $\alpha \ll 1$ with $v := \boldsymbol{w}^\top \boldsymbol{x} + b$
2. **Maxout**: $\max_{k \in [K]} v_k$ with $v_k := \boldsymbol{W}_k^\top \boldsymbol{x} + b_k$



Linear regions: maximally connected regions of input space on which the NN's output is linear [Montufar et al., 2014].

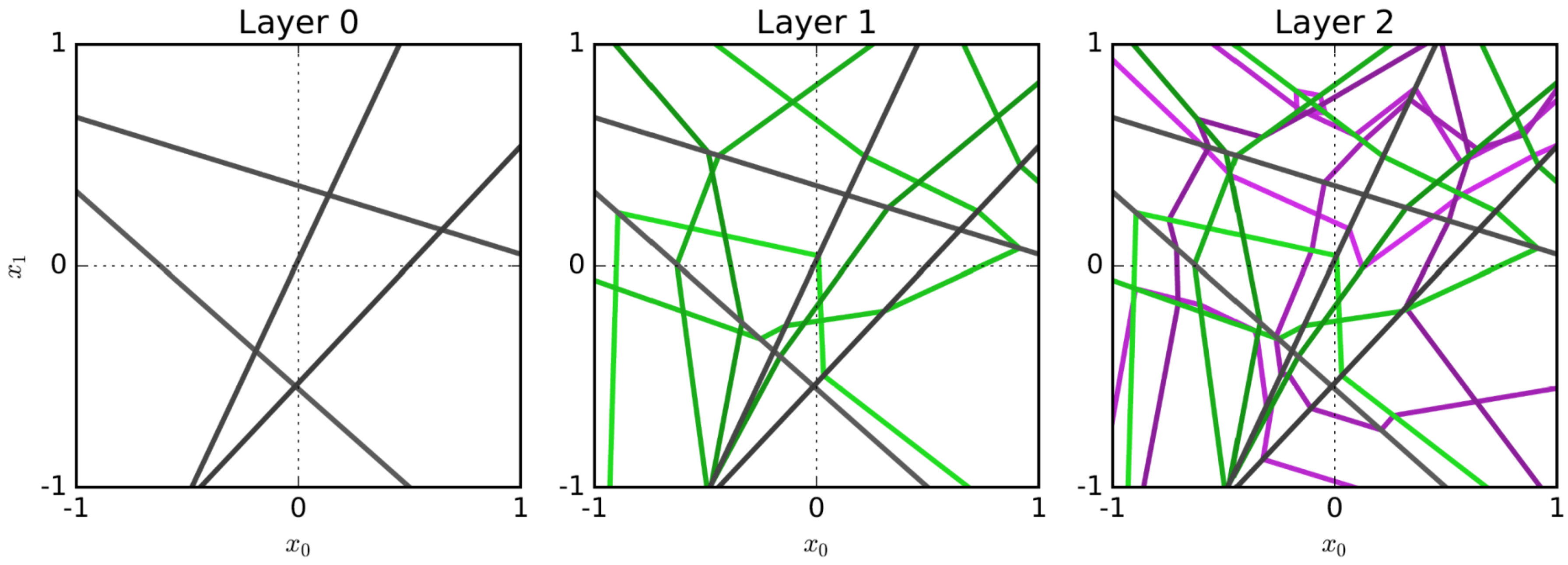


Figure: Input space is subdivided into convex polytopes, each of which is a "linear region" for the NN. Reproduced from [Raghu et al., 2016]

Claim: more linear regions ≡ more expressive power

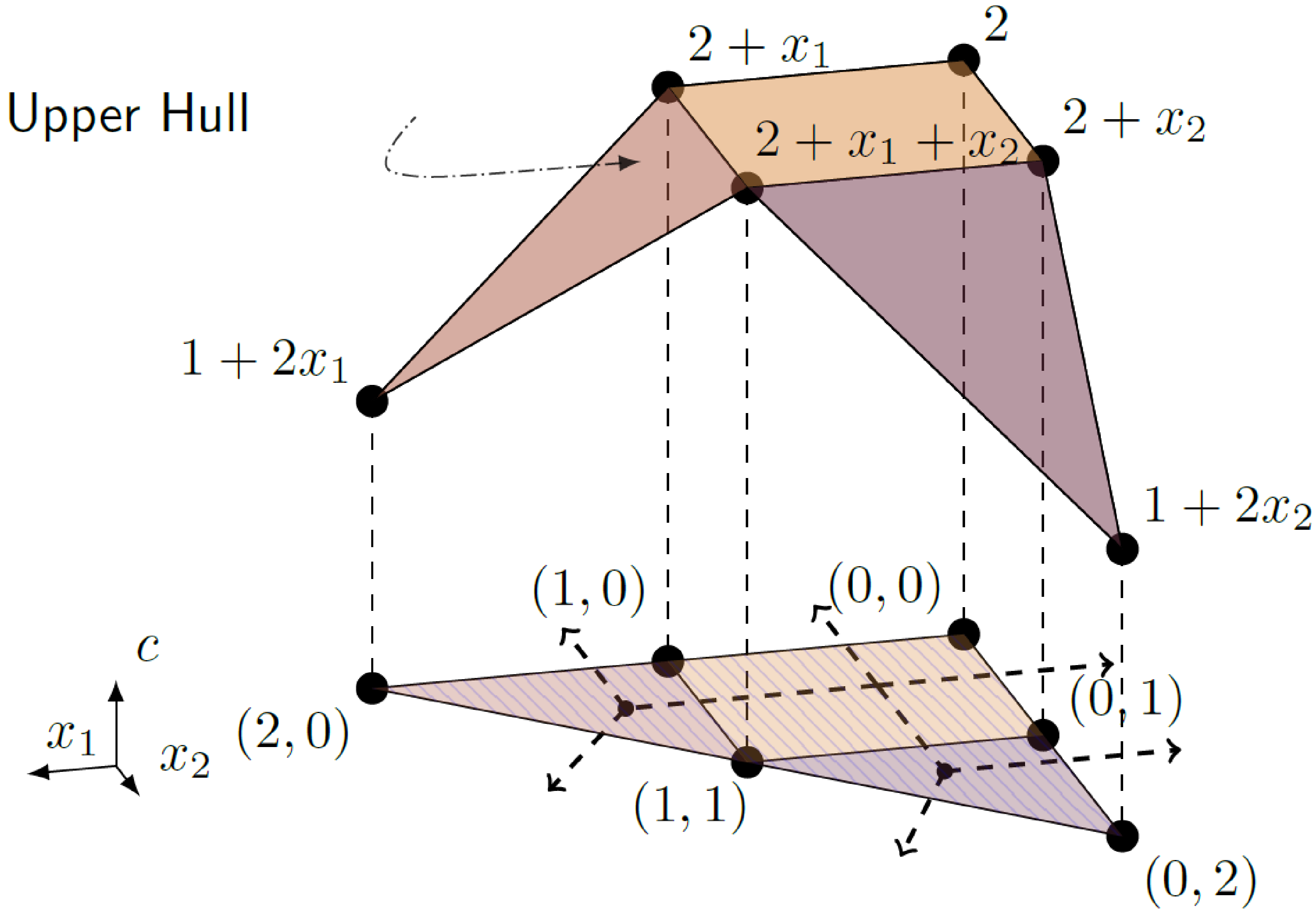
Single neuron result

An application of the fundamental theorem of LP yields:

Proposition [Charisopoulos & Maragos, 2017]

The number of linear regions for a single maxout unit $p(\boldsymbol{x}) = \max_{j\in[k]} \boldsymbol{w}_j^\top \boldsymbol{x} + b_i$ are equal to the number of vertices on the upper hull of $\mathcal{N}(p)$

- subsumes **relu**
- all terms corresponding to interior vertices can be *removed* without affecting $p(\boldsymbol{x})$ *as a function*.



$$p(x_1, x_2) = \max(1+2x_1, 2+x_1, 2, 2+x_2, 1+2x_2, 2+x_1+x_2)$$

For a collection of tropical polynomials, suffices to work with Minkowski sums:

Proposition [Charisopoulos & Maragos, 2018] [Zhang et al., 2018]

The number of linear regions of a layer with n inputs and m neurons is upper bounded by the number of vertices in the upper convex hull of

$$\mathcal{N}(p_1) \oplus \cdots \oplus \mathcal{N}(p_m),$$

where $\oplus$ denotes Minkowski sum.

Main Result

Immediate application of a bound from [Gritzmann and Sturmfels, 1993] on faces of Minkowski sums gives

Proposition [Charisopoulos & Maragos, 2018]

The number of linear regions of n input, m output layer consisting of convex PWL activations of rank k is bounded above by

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

In case of ReLU, use symmetry of zonotopes to refine to

$$\min\left\{2^m, \sum_{j=0}^{n}\binom{m}{j}\right\}$$

Counting in practice

Goal: given a network, count # of linear regions (exactly or approximately)

Exact counting using insight from Newton polytopes:

- ▷ vertex enumeration algorithm for Mink. sums [Fukuda, 2004] $\Rightarrow$ requires solving $\Omega(|\mathrm{vert}(P)|)$ LPs.
- ▷ impractical unless problem is small

MIP representability of NNs [Serra et al., 2018]:

- ▷ Assumes bounded range of input space
- ▷ Requires enumerating solutions of MILPs

Geometric Algorithm: Randomized method for Sampling the Extreme Points of the Upper Hull of a Polytope [Charisopoulos & Maragos 2019, arXiv:1805.08749v2], [Maragos, Charisopoulos & Theodosis, Proc. IEEE 2021]

Computational Geometry: [Karavelas & Tzanaki, ISCG 2015]: A Geometric Approach for the Upper Bound Theorem for Minkowski Sums of Convex Polytopes

Geometry & Algebra of NNs with PWL Activations

Theorem (Wang 2004): A continuous piecewise linear function is equal to the difference of two max-polynomials.

Theorem (Charisopoulos & Maragos 2018): The essential terms of a tropical polynomial are in bijection $1 - 1$ with the vertices on the upper hull of its extended Newton polytope.

Theorem (Zhang et al. 2018): A neural network with ReLU-type activations can be represented as the difference of two max-polynomials(*), i.e. with a tropical rational function.

[(*) Zhang et al. only call "max polynomials" those polynomials with integer slopes]

[Calafiore et al., 2019] use the Maslov dequantization to design universal approximators for convex (+loglog-convex) data

$$f \text{ convex} \Rightarrow f \simeq f_{\mathrm{PWL}} \Leftrightarrow f \simeq f_T,$$

where $f_{\mathrm{PWL}} \leq f_T \leq T \log K + f_{\mathrm{PWL}}$ and are given by

$$\begin{cases} f_{\mathrm{PWL}} := \max_{k \in [K]} \langle \boldsymbol{a}_k, \boldsymbol{x} \rangle + b_k, \\ f_T := T \log \left(\sum_{k=1}^{K} \exp \{ b_k + \langle \boldsymbol{a}_k, \boldsymbol{x} \rangle \}^{1/T} \right) \end{cases}$$

In particular, fixing $\varepsilon > 0$ and compact $\mathcal{C}$, a small enough T will satisfy

$$\sup_{\boldsymbol{x} \in \mathcal{C}} |f_T(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \varepsilon.$$

Morphological Networks:
Geometry, Training, and Pruning

References:

- V. Charisopoulos and P. Maragos, "Morphological Perceptrons: Geometry and Training Algorithms", Proc. ISMM 2017, LNCS 10225, Springer.

- N. Dimitriadis and P. Maragos, "Advances in Morphological Neural Networks: Training, Pruning and Enforcing Shape Constraints", Proc. ICASSP, 2021.

Motivation

- Explosion of ML research in the last decade (now models with near-human or even human performance)
- Recent advances indicate shift towards nonlinearity, but…
- ...the “multiply-accumulate” (= linear) operations of the perceptron are still ubiquitous

Our Questions:

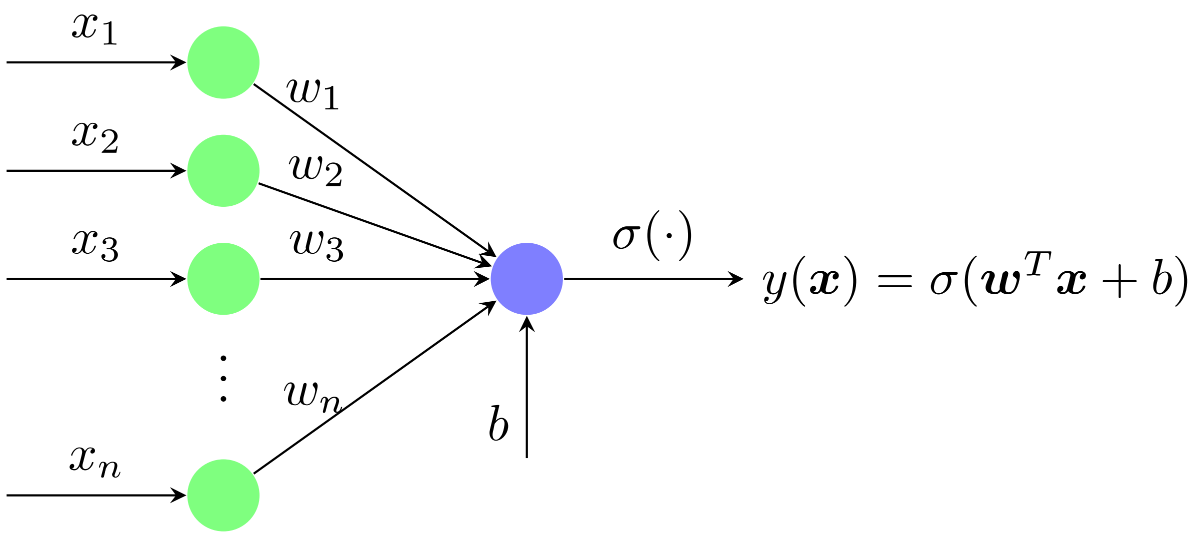
- Are dot products and convolutions the only biologically plausible models of neuronal computation?
- Can we use results and tools from “nonlinear” mathematics to reason about complexity and dimension of learning models in current literature?

Rosenblatt's perceptron

- Introduced in 1943, still prevalent neural model
- Activation: $\phi(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$
- Nonlinearity at the output (e.g logistic sigmoid, ReLU):

$$y(\boldsymbol{x}) = \sigma(\phi(\boldsymbol{x}))$$

- Multiply-accumulate architecture → archetypal building block of all architectures (e.g. fully-connected, convolutional etc.)



Morphological (Max-Plus) Perceptron

- Introduced in the 1990's. Instead of multiply-accumulate, computes a dilation (max-of-sums):

$$\tau(\boldsymbol{x}) = \boldsymbol{w}^T \boxplus \boldsymbol{x} \triangleq \bigvee_{i=1}^{n} w_i + x_i$$

or an erosion:

$$\tau'(\boldsymbol{x}) = \boldsymbol{w}^T \boxplus' \boldsymbol{x} \triangleq \bigwedge_{i=1}^{n} w_i + x_i$$

- Ritter & Urcid (2003): argued about biological plausibility and proved that every compact region in *n*-dim Euclidean space can be approximated by morphological perceptrons to arbitrary accuracy.

- Related to a Maxout unit.

Feasible Regions & Separability Condition for Max-plus Percepton

Let $\boldsymbol{X} \in \mathbb{R}_{\max}^{k \times n}$ be a matrix containing the patterns to be classified as its rows, let $\boldsymbol{x}^{(k)}$ denote the k-th pattern (row) and let $\mathcal{C}_1, \mathcal{C}_0$ be the two classes

Max-plus perceptron $\tau(\boldsymbol{x}) = \boldsymbol{w}^T \boxplus \boldsymbol{x}$

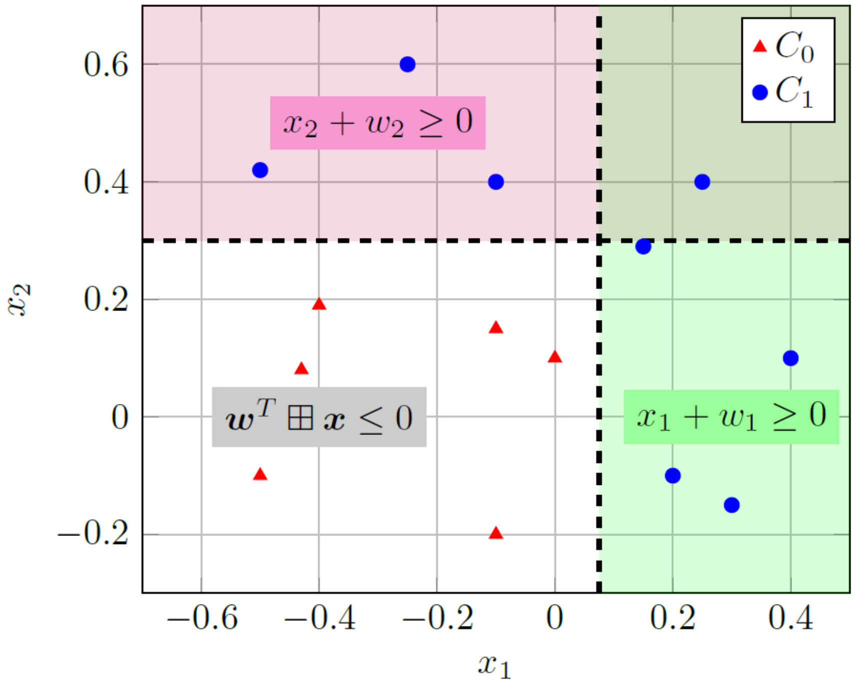
$$\tau(\boldsymbol{x}) = w_0 \vee (w_1 + x_1) \vee \cdots \vee (w_n + x_n) = w_0 \vee \left(\bigvee_{i=1}^{n} w_i + x_i \right)$$

Feasible Region = Tropical Polyhedron

$$\mathcal{T}(\boldsymbol{X}_{\text{pos}}, \boldsymbol{X}_{\text{neg}}) = \{\boldsymbol{w} \in \mathbb{R}_{\max}^{n} : \boldsymbol{X}_{\text{pos}} \boxplus \boldsymbol{w} \geq 0, \ \boldsymbol{X}_{\text{neg}} \boxplus \boldsymbol{w} \leq 0\}$$

Separability Condition, equivalent to Nonempty Trop. Polyhedron

$$\mathbf{X}_{\text{pos}} \boxplus (\mathbf{X}_{\text{neg}}^{*} \boxplus' \mathbf{0}) \geq \mathbf{0}$$



[Charisopoulos & Maragos, ISMM 2017]

Morphological Neural Nets (MNNs) and Training Approaches

- **Constructive Algorithms**

Dendrite Learning [Ritter & Urcid, 2003], Iterative Partitioning / Competitive Learning [Sussner & Esmi, 2011]: combine (max, +) and (min, +) classifiers, build “bounding boxes” around patterns

- "perfect" fit to data, no concept of outlier

- **Morphological Associative Memories**

Introduce a Hopfield-type network, computing (noniteratively) a morphological/fuzzy response (e.g. Sussner & Valle, 2006):

- **PAC Learning**

Min-max classifiers [Yang & Maragos, 1995]

- **Gradient Descent Variants**

MRL nodes [Pessoa & Maragos, 2000], Dilation-Erosion Linear Perceptron [Araujo et al. 2012].

- **Recent Approaches:**

Convex-Concave Programming (CCP) for Max-plus Perceptron and DEP (Binary Classification) [Charisopoulos & Maragos 2017]

Reduced Dilation-Erosion Perceptron (r-DEP) trained via CCP for Binary Classification [Valle 2020]

Dense Morphological Networks [Mondal et al. 2019]

Deep Morphological Networks [Franchi et al. 2020]

r-DEP for Multiclass Classification via CCP, L1 Pruning on Dense MNNs [Dimitriadis & Maragos 2021]

A CCP Approach for Training MP on Non-separable Data

Training a (max, +) perceptron can be stated as a difference-of-convex (DC) optimization problem. Solved iteratively (but global optimum not guaranteed) by the Convex-Concave Procedure (**CCP**) [Yuille & Rangarajan 2003], implemented via Disciplined CCP (DCCP - CvxPy) [Shen et al. 2016]

Given a sequence of training data $\{\boldsymbol{x}^k\}_{k=1}^K$:

$$\text{Minimize } J(\boldsymbol{X}, \boldsymbol{w}, \boldsymbol{\nu}) = \sum_{k=1}^{K} \nu_k \cdot \max(\xi_k, 0)$$

$$\text{s. t.} \begin{cases} \bigvee_{i=1}^{n} w_i + x_i^{(k)} \leq \xi_k & \text{if } \boldsymbol{x}^{(k)} \in \mathcal{C}_0 \\ \bigvee_{i=1}^{n} w_i + x_i^{(k)} \geq -\xi_k & \text{if } \boldsymbol{x}^{(k)} \in \mathcal{C}_1 \end{cases}$$

Weighted DCCP
[Charisopoulos & Maragos 2017]

Negative target

Positive target

ν_k Some measure of "being outlier“ (e.g. proportional to 1/distance of the k-th pattern from its class centroid)

ξ_k (slack variables) Positive only if misclassification occurs at *k*-th pattern

Gradient Descent vs. CCP for Training (max,+) Perceptron

Two Binary Classification Experiments with small datasets,

Ripley (GMM-2) and WBCD (~1k):

Gradient descent with fixed N = 100 epochs vs. CCP using the DCCP toolkit for CvxPy (default parameters).

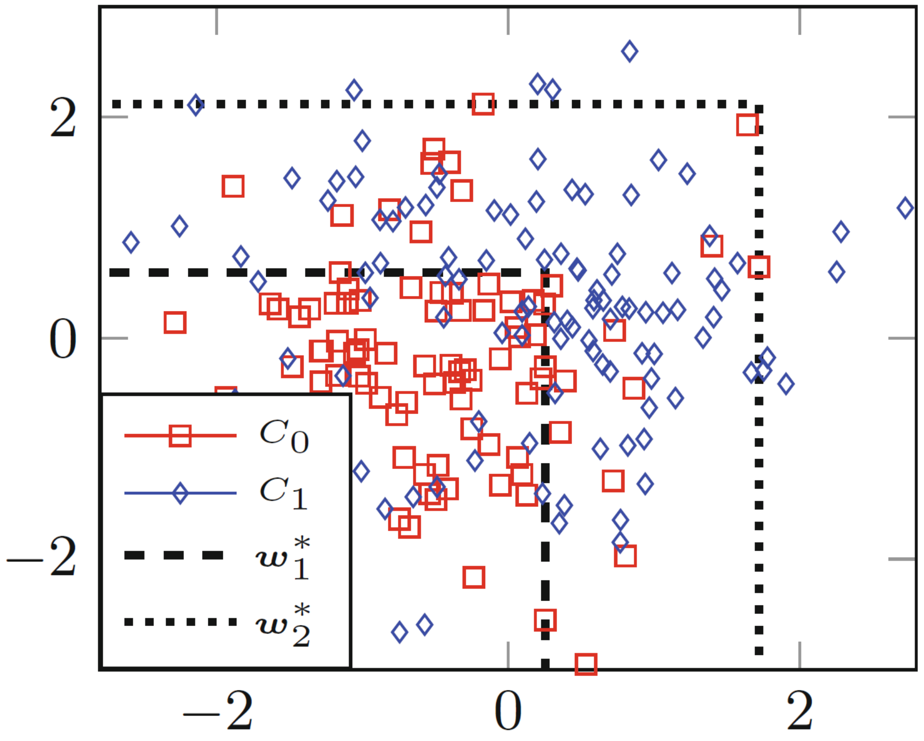
η	Ripleys		WDBC	
	SGD	WDCCP	SGD	WDCCP
0.01	0.838 ± 0.011		0.726 ± 0.002	
0.02	0.739 ± 0.012		0.763 ± 0.006	
0.03	0.827 ± 0.008		0.726 ± 0.004	
0.04	0.834 ± 0.008		0.751 ± 0.007	
0.05	0.800 ± 0.009	**0.902** ± 0.001	0.783 ± 0.012	**0.908** ± 0.001
0.06	0.785 ± 0.008		0.768 ± 0.01	
0.07	0.776 ± 0.009		0.729 ± 0.009	
0.08	0.769 ± 0.01		0.732 ± 0.01	
0.09	0.799 ± 0.009		0.730 ± 0.015	
0.1	0.749 ± 0.011		0.729 ± 0.009	

CCP: more robust results

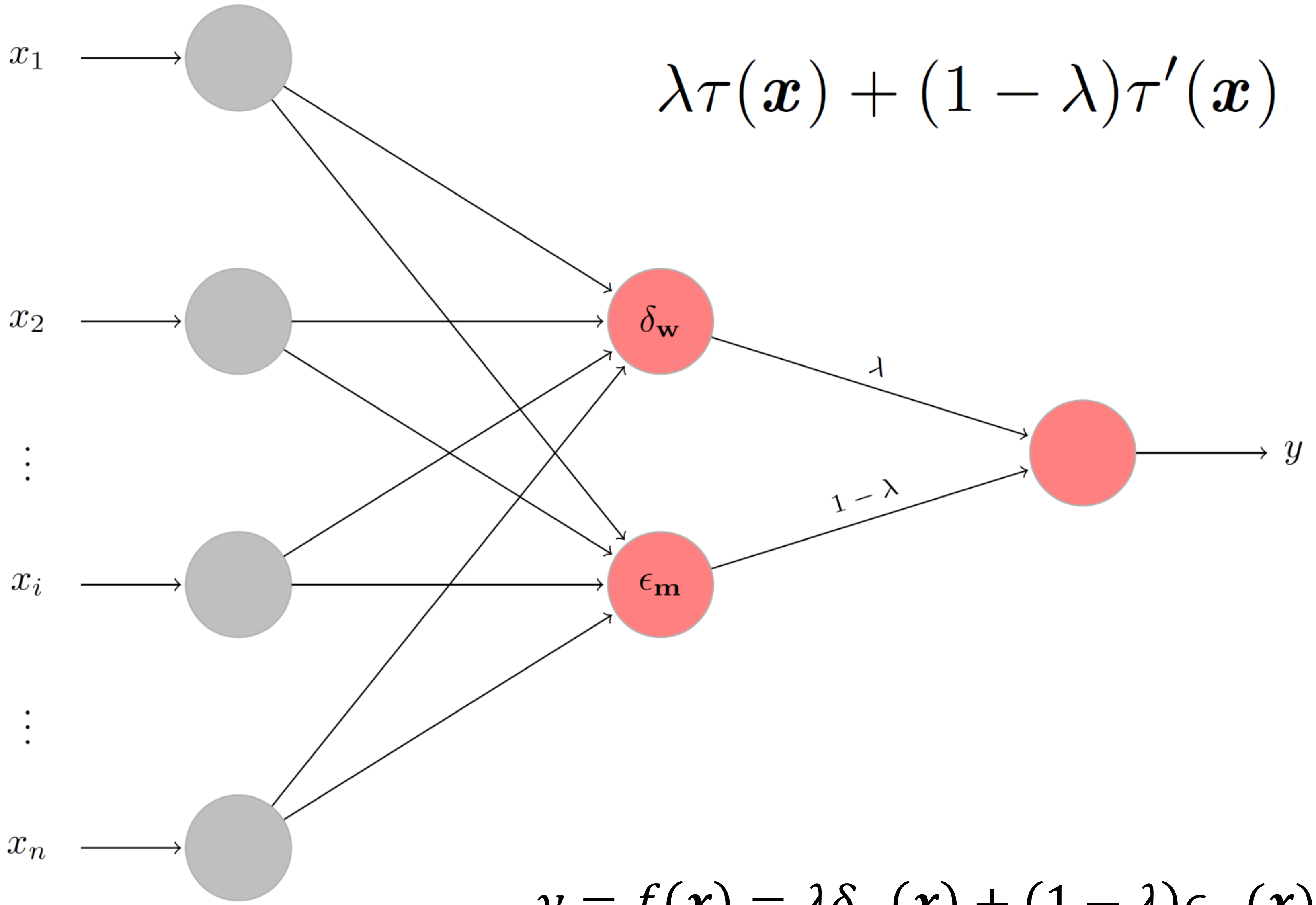
Classification of initially separable Gaussian data with randomly flipped labels 20%:

...... : No regularization (DCCP)

---- : Regularization (Weighted DCCP)



Dilation-Erosion Perceptron (DEP)



$$\lambda\tau(\boldsymbol{x}) + (1-\lambda)\tau'(\boldsymbol{x})$$

$$y = f(\boldsymbol{x}) = \lambda\delta_w(\boldsymbol{x}) + (1-\lambda)\epsilon_m(\boldsymbol{x})$$

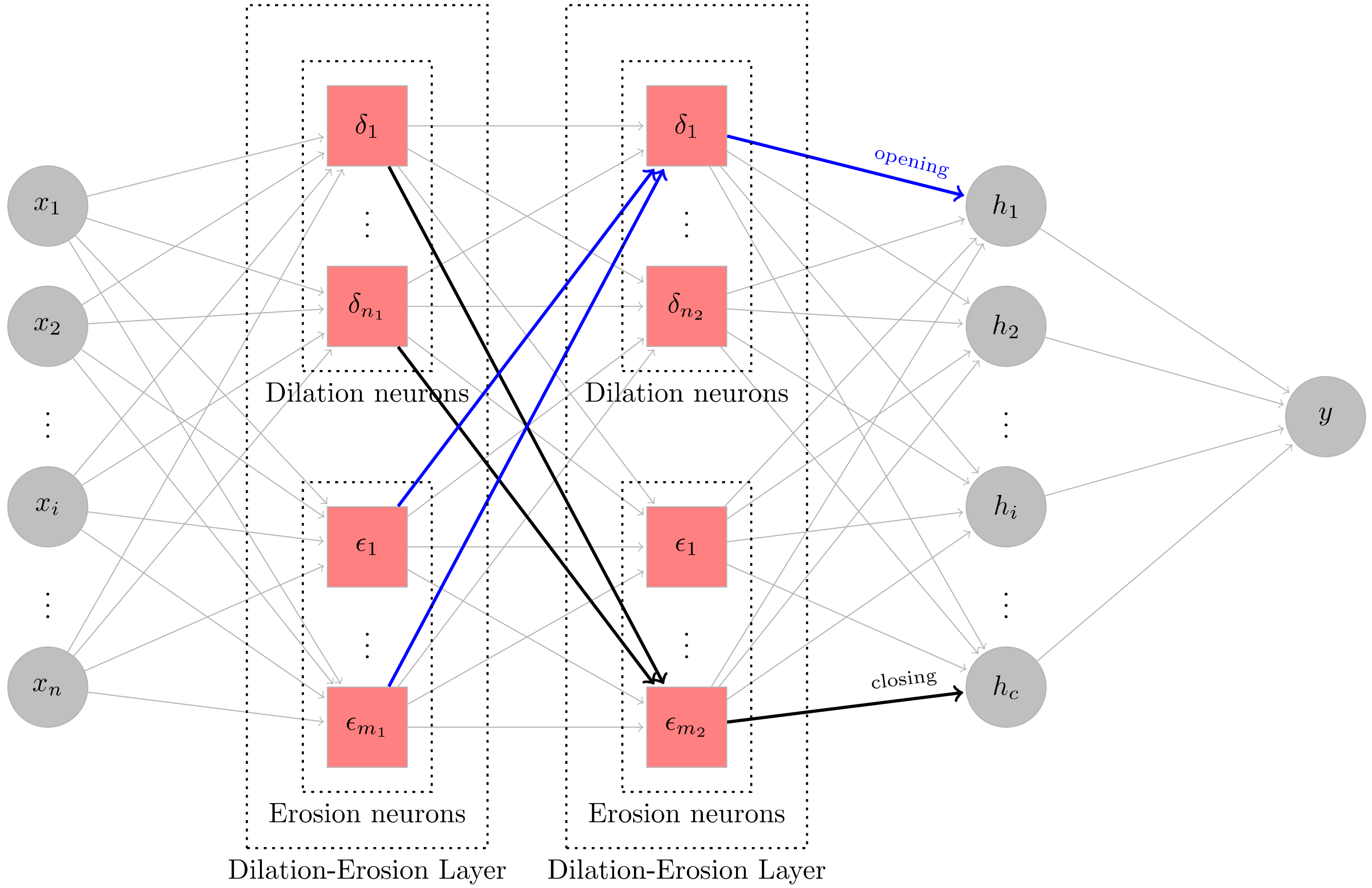
Dilation-Erosion Perceptron Training

$$y = f(\boldsymbol{x}) = \lambda\delta_w(\boldsymbol{x}) + (1-\lambda)\epsilon_m(\boldsymbol{x}) = \lambda\delta_w(\boldsymbol{x}) - (1-\lambda)[-\epsilon_m(\boldsymbol{x})]$$
$$= convex - (-concave)$$
$$= convex - convex$$

Training as Difference-of-Convex Optimization via Convex-Concave Programming

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} v_i \max\{0, \xi_i\} \\ \text{subject to} \quad & \lambda\delta_{\mathrm{w}}(\mathbf{x}_i) + (1-\lambda)\varepsilon_{\mathrm{m}}(\mathbf{x}_i) \geq -\xi_i \quad \forall \mathbf{x}_i \in \mathcal{P}, \\ & \lambda\delta_{\mathrm{w}}(\mathbf{x}_i) + (1-\lambda)\varepsilon_{\mathrm{m}}(\mathbf{x}_i) \leq +\xi_i \quad \forall \mathbf{x}_i \in \mathcal{N} \end{aligned}$$

Dense Morphological Networks



Dense Morphological Network with 2 hidden layers [similar to Mondal et al. 2019]

Focus on Sparsity [Dimitriadis & Maragos 2021] → Apply ℓ_1 Pruning

Experiments: Pruning Dense MNN vs MLP-ReLU

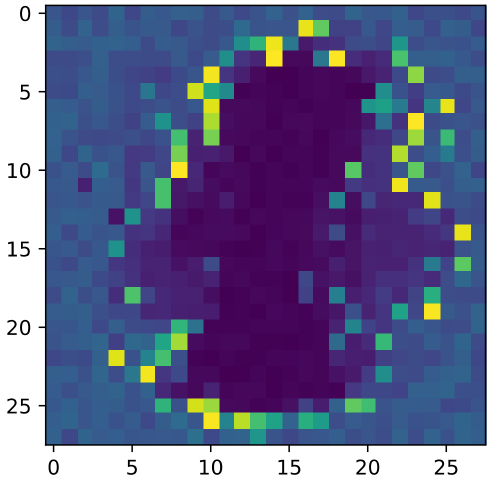
	p	Adaptive Momentum Estimation δ	Adaptive Momentum Estimation ε	Adaptive Momentum Estimation (δ, ε)	Adaptive Momentum Estimation FF-ReLU	Stochastic Gradient Descent δ	Stochastic Gradient Descent ε	Stochastic Gradient Descent (δ, ε)	Stochastic Gradient Descent FF-ReLU
MNIST	100%	97.62	96.17	97.95	98.13	94.86	93.36	96.07	98.16
	75%	97.62	96.18	97.93	98.15	94.86	93.36	96.07	98.12
	50%	97.62	96.22	97.90	98.17	94.86	93.37	96.07	98.08
	25%	97.62	96.09	97.87	97.51	94.86	93.40	96.06	98.01
	10%	97.62	95.78	97.74	93.38	94.86	93.38	96.09	96.67
	7.5%	97.62	95.42	97.76	90.17	94.86	93.38	96.10	95.56
	5%	97.62	94.51	97.66	83.39	94.86	93.40	96.10	92.96
	2.5%	97.62	93.43	97.37	68.93	94.86	93.39	96.09	80.48
	1%	97.62	91.17	97.08	44.22	94.86	93.38	96.08	58.07
FashionMNIST	100%	86.31	86.82	88.32	88.82	82.06	85.23	86.21	87.79
	75%	86.30	86.81	88.30	88.88	82.00	85.23	86.21	87.75
	50%	86.22	86.80	88.33	88.18	82.05	85.25	86.20	87.19
	25%	85.95	86.85	88.31	82.15	81.90	85.26	86.28	84.35
	10%	85.58	86.27	88.05	65.89	81.67	85.27	86.23	73.22
	7.5%	85.47	86.15	87.99	57.93	81.63	85.27	86.21	63.95
	5%	85.37	85.81	87.76	49.12	81.52	85.24	86.22	47.73
	2.5%	84.91	85.47	87.56	42.48	81.14	85.26	86.22	38.84
	1%	81.14	84.86	86.85	28.13	80.68	85.27	86.18	35.46

Table: Accuracy of pruned networks on the MNIST and FashionMNIST datasets.

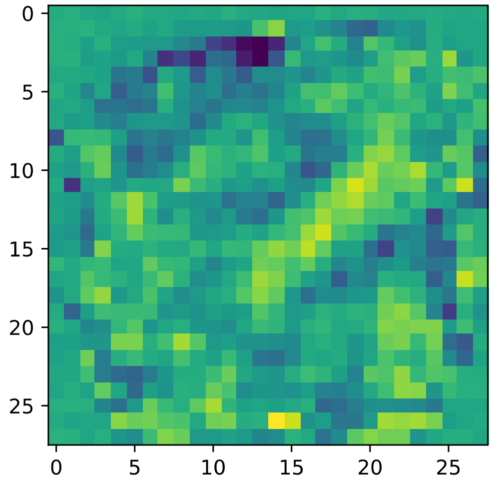
Models: $\delta \to$ only dilation neurons, $\varepsilon \to$ only erosion, $(\delta, \varepsilon) \to$ split equally, FF-ReLU $\to$ FeedForward NN with ReLU.

shades of red showcase the degree of (severe) deterioration in accuracy green indicates the absence of performance loss

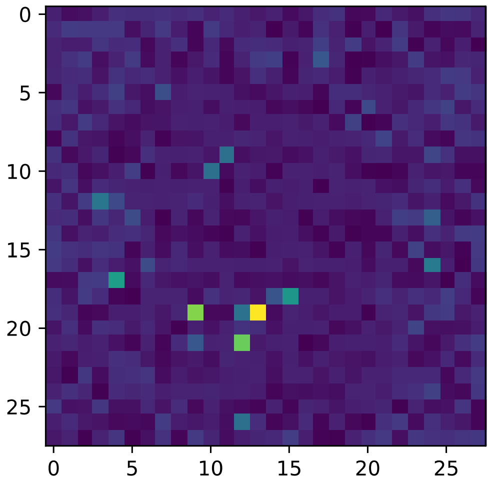
Qualitative Perspectives on Sparsity



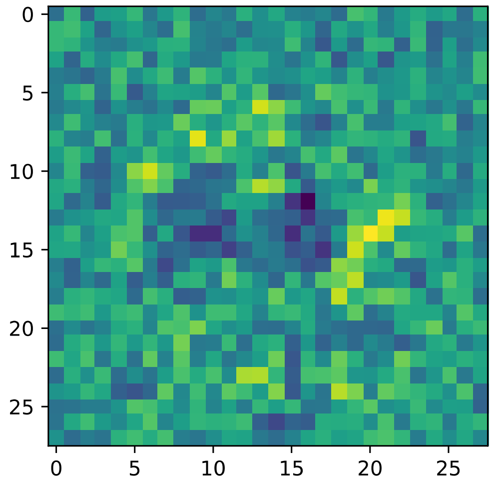
$(\delta, \epsilon) - Adam$



FF−ReLU $- Adam$



$(\delta, \epsilon) - SGD$



FF-ReLU $-SGD$

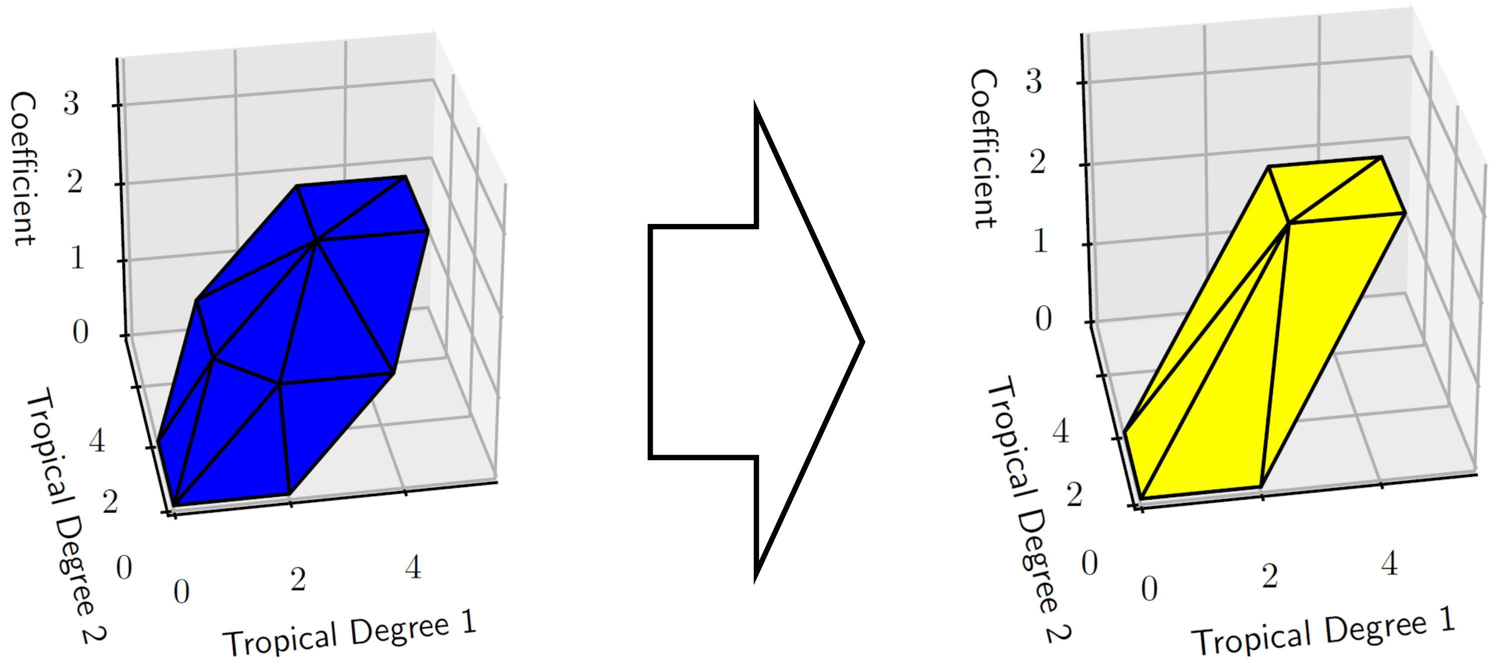
Examples of hidden layer activations for various NN models (MNIST dataset)

Minimization of Neural Nets via Tropical Division

References:

- G. Smyrnis, P. Maragos and G. Retsinas, "*MaxPolynomial Division With Application to Neural Network Simplification*", Proc. ICASSP 2020.
- G. Smyrnis and P. Maragos, "*Multiclass Neural Network Minimization Via Tropical Newton Polytope Approximation*", Proc. ICML 2020.

General idea for Geometric NN Minimization



Original Network Polytope

Approximate Network Polytope

Reminder: Tropical Polynomials and Newton Polytopes

Tropical Semiring $(\mathbb{R}_{\max}, \vee, +)$ $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$

$$a \vee b = \max(a, b) \qquad a + b = a + b$$

Real coefficients

Tropical Polynomials

$$f(\boldsymbol{x}) = \max_{i \in [n]} \{\boldsymbol{a}_i^T \boldsymbol{x} + b_i\}$$

Newton Polytopes

$$\text{Newt}\,(f) = \text{conv}\,\{\boldsymbol{a}_i : i \in [n]\}$$

$$\text{ENewt}\,(f) = \text{conv}\,\{(\boldsymbol{a}_i, b_i) : i \in [n]\}$$

Polytope computation

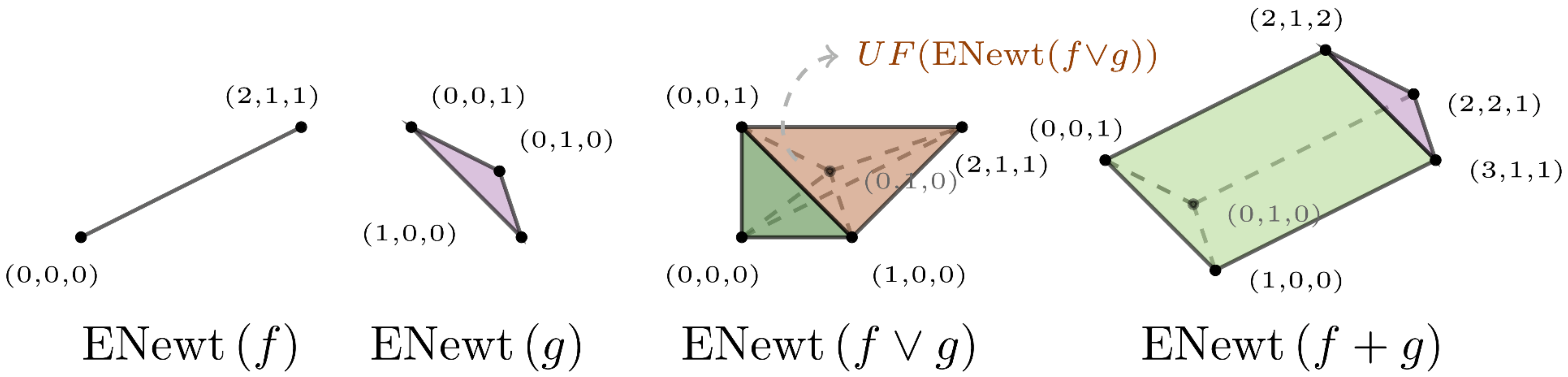
$$\text{ENewt}\,(f \vee g) = \text{conv}\{\text{ENewt}\,(f) \cup \text{ENewt}\,(g)\}$$

$$\text{ENewt}\,(f + g) = \text{ENewt}\,(f) \oplus \text{ENewt}\,(g)$$

Example: Polytope Computation

$$f(x, y) = \max(2x + y + 1, 0)$$

$$g(x.y) = \max(x, y, 1)$$



$$f \vee g = \max(2x + y + 1, 0, x, y, 1)$$

$$f + g = \max(x, y, 1, 3x + y + 1, 2x + 2y + 1, 2x + y + 2)$$

Max-polynomial Division

Problem: Assume we have two max-polynomials $p(\boldsymbol{x}), d(\boldsymbol{x})$ (dividend and divisor). We want to find two max-polynomials $q(\boldsymbol{x}), r(\boldsymbol{x})$ (quotient and remainder) such that:

$$p(\boldsymbol{x}) = \max(q(\boldsymbol{x}) + d(\boldsymbol{x}), r(\boldsymbol{x}))$$

However! The above is not always feasible (non-trivially).

Approximate Division: We relax the requirements, so that the polynomials we want to find satisfy:

$$p(\boldsymbol{x}) \geq \max(q(\boldsymbol{x}) + d(\boldsymbol{x}), r(\boldsymbol{x}))$$

We also require that $q(\boldsymbol{x}), r(\boldsymbol{x})$ satisfy the above maximally.

Algorithm for Approximate Maxpolynomial Division

1. Let C be the set of possible vectors $\boldsymbol{c}$ by which we can h-shift $\mathrm{Newt}(d)$ (each of which corresponds to a linear term in q).
2. We raise the shifted version of $\mathrm{ENewt}(d)$ as high as possible so that it still lies below $\mathrm{ENewt}(p)$, and we mark the vertical shift as q_c.
3. We set the quotient equal to:

$$q(\boldsymbol{x}) = \max_{\boldsymbol{c} \in C}(q_{\boldsymbol{c}} + \boldsymbol{c}^T \boldsymbol{x})$$

and add all terms not covered by an h-shift $\boldsymbol{c}$ to the remainder $r(\boldsymbol{x})$.

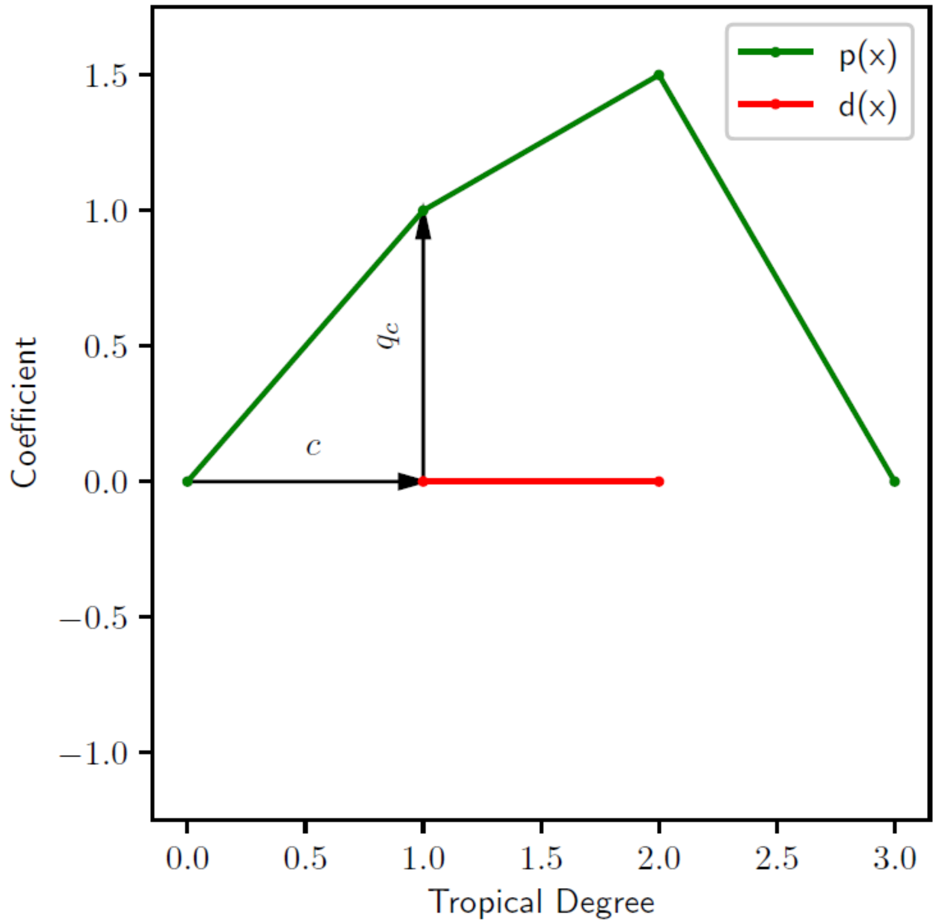


Figure: Division Method
Division of $p(x) = \max(3x, 2x + 1.5, x + 1, 0)$ by $d(x) = \max(x, 0)$.

Division Example

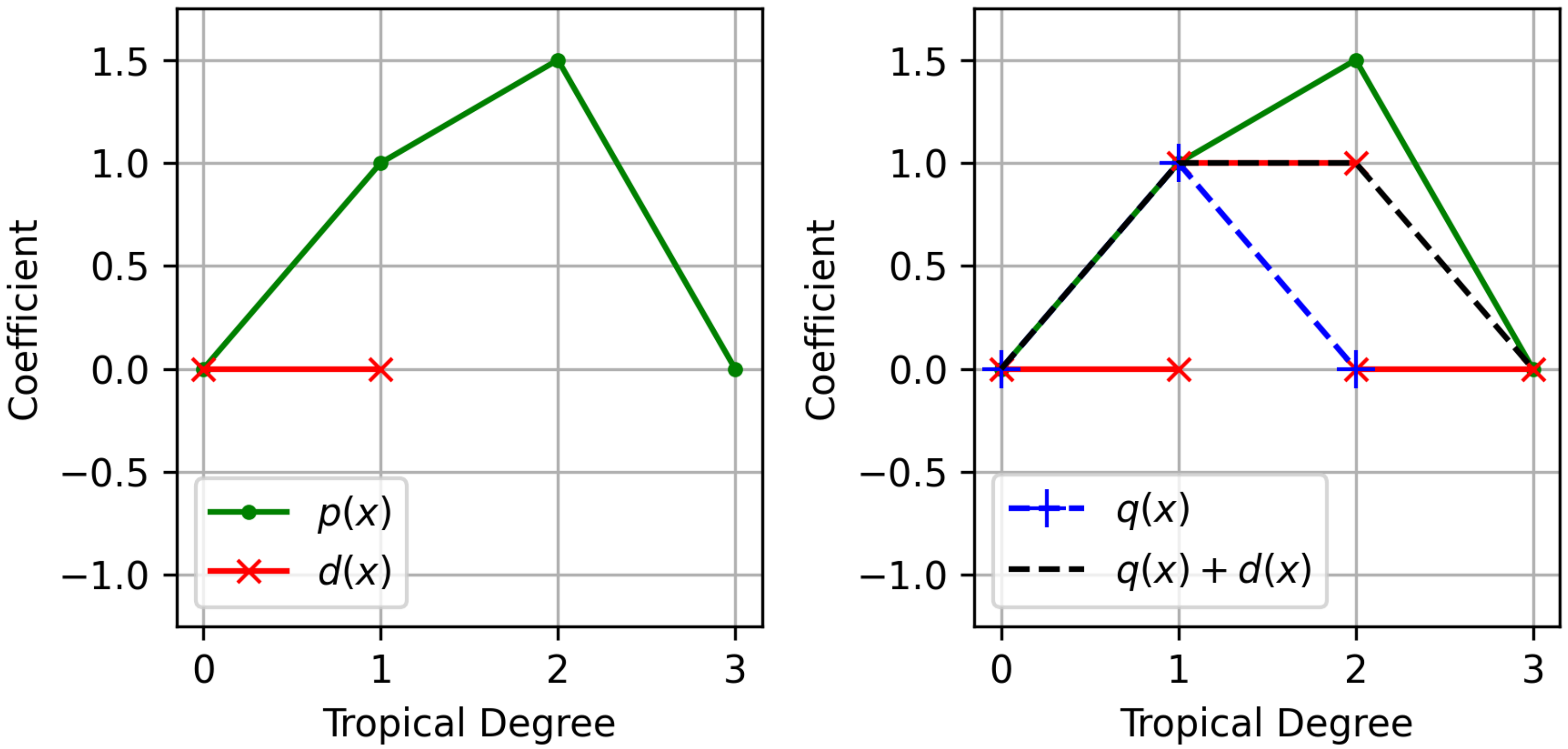


Figure: Division of $p(x) = \max(3x, 2x + 1.5, x + 1, 0)$ by $d(x) = \max(x, 0)$.

Note: The Newton Polytope of the divisor is raised as much as possible, but it cannot match the polytope of the dividend exactly. Thus, only 3 out of the 4 vertices are perfectly matched.

Application to Neural Network Minimization

General idea: Our algorithm seeks to minimize the network by matching the most important vertices of the ENewton Polytopes of its maxpolynomials.

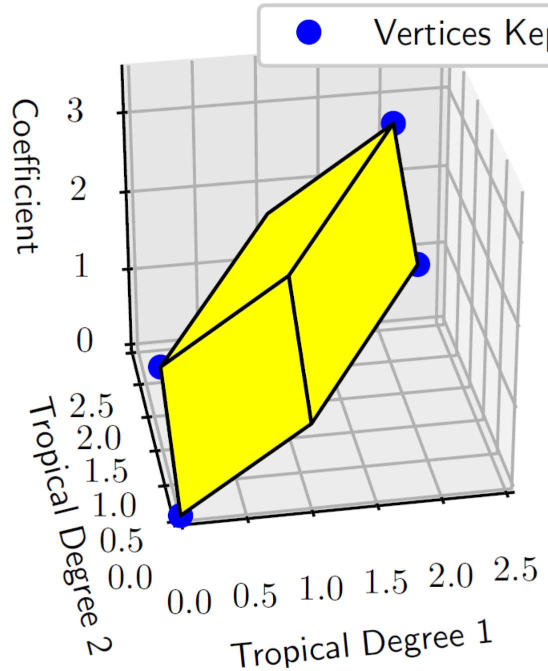
2-layer 1-output NN:

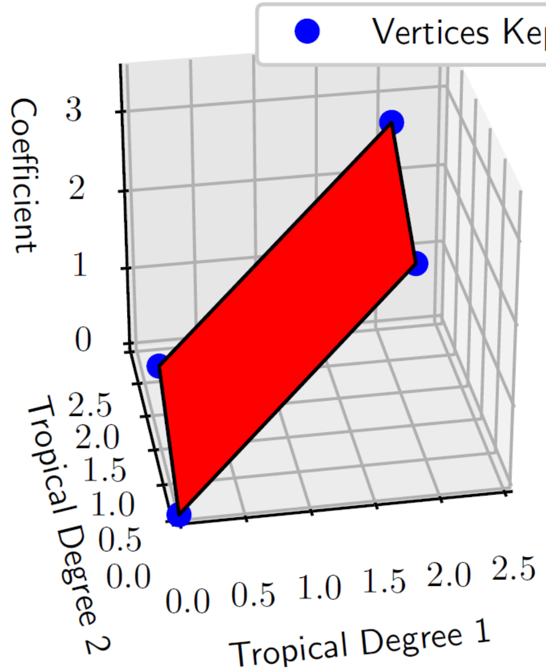
The NN considered is the difference of two maxpolynomials.

For each of the two (+,-) maxpolynomials $p(\boldsymbol{x})$ of the network, we first find a divisor $d(\boldsymbol{x})$. This is done by:

Finding the most important vertices of $\mathrm{ENewt}(p)$, via the weights of the network (based on which combination of neurons is activated).

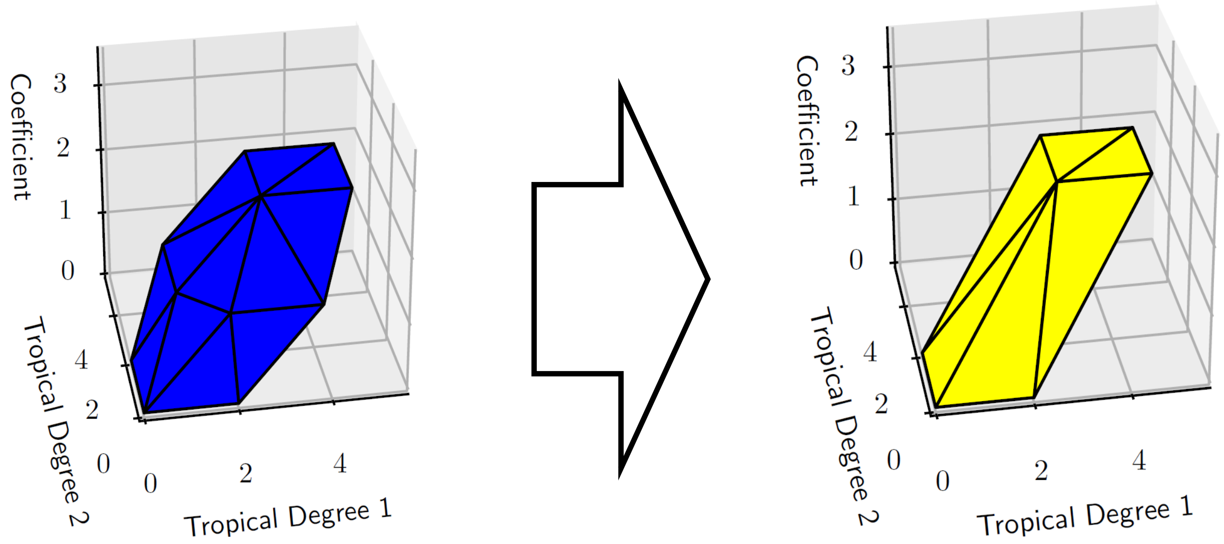
Method for Single Output Neuron





- Final polytope (right) is precisely under the original (left).
- The process is a “smoothing” of the original polytope.
  (From the 8 vertices of the original-yellow polytope we keep only the 4 blue which comprise the vertices of the final-red polytope.)

Properties of Trop. Div. Approximation Method

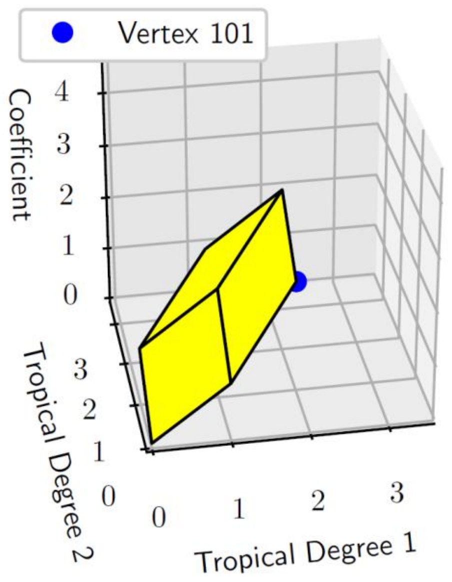


Original Network Polytope   *Approximate Network Polytope*

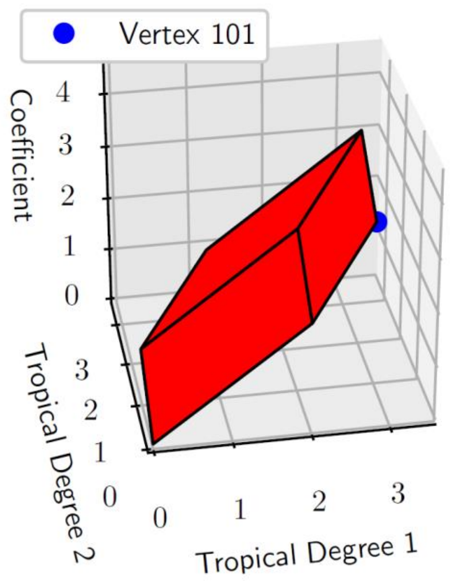
1. Approximate polytope contains only vertices of the original.
2. The input samples activating the chosen vertices have the same output in the two networks.
3. At least $\frac{N}{\sum_{j=0}^{d}\binom{n}{j}} O(\log n')$ samples retain their output

(N is # of samples, n and n' the # of neurons in hidden layer before and after the approximation).
Note: this is not a tight bound.

Extension with Multiple Output Neurons



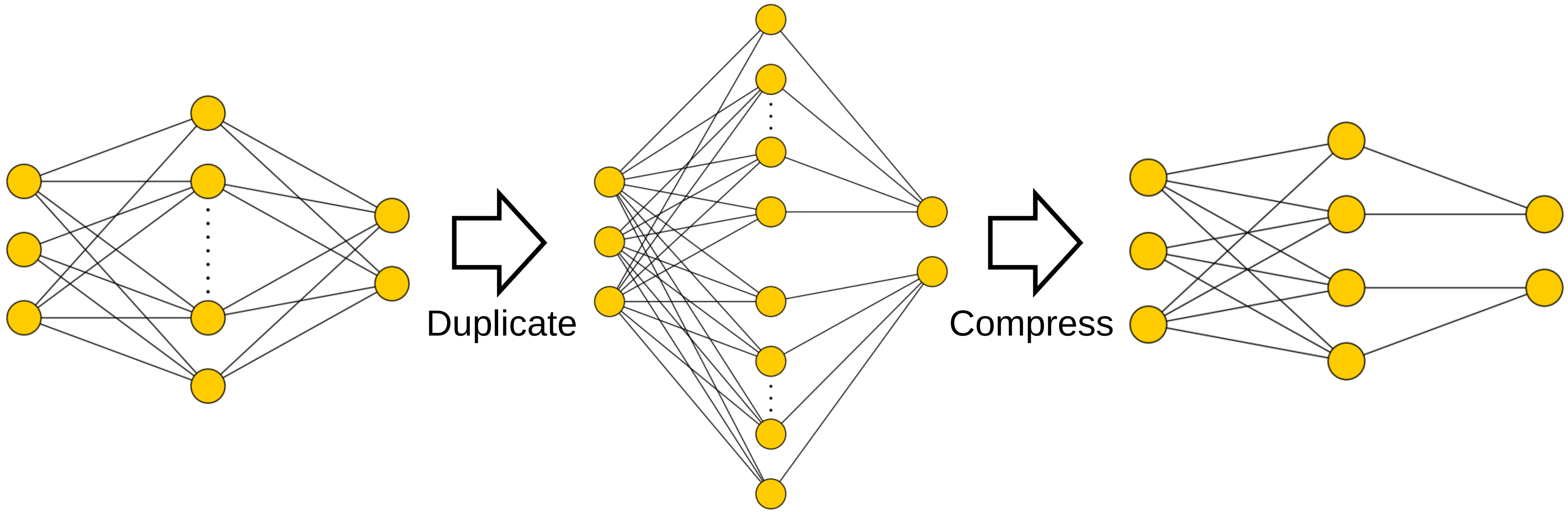
Upper hull of polytope, Neuron 1



Upper hull of polytope, Neuron 2

- <u>What we have:</u> Multiple polytopes (one pair for each output neuron), interconnected (Minkowski sums of same hidden neurons but with different scaling weights).
- <u>What we want:</u> Simultaneous approximation of all polytopes.

Trop. Div. method for Multiple Outputs: One-Vs-All Framework



Experiments: Trop. Division NN Minimization

Neurons Kept	TropDiv Method, Avg. Accuracy	TropDiv Method, St. Deviation
Original	98.604	0.027
75%	**96.560**	**1.245**
50%	**96.392**	**1.177**
25%	**95.154**	**2.356**
10%	**93.748**	**2.572**
5%	92.928	2.589

MNIST Dataset

Neurons Kept	TropDiv Method, Avg. Accuracy	TropDiv Method, St. Deviation
Original	88.658	0.538
75%	**83.556**	2.885
50%	**83.300**	2.799
25%	**82.224**	2.845
10%	**80.430**	3.267

Fashion-MNIST Dataset

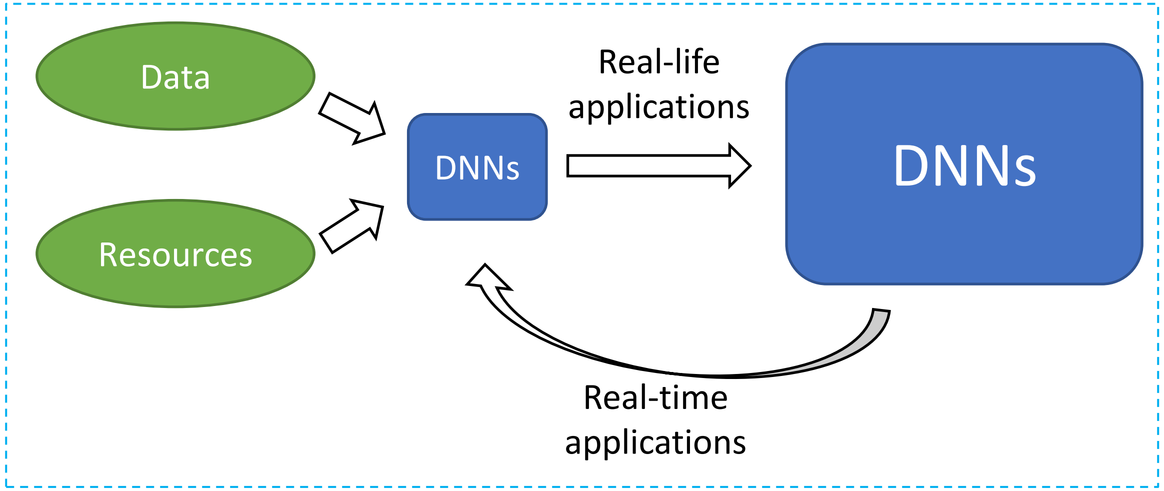
[G. Smyrnis & P. Maragos, "*Multiclass Neural Net Minimization, Tropical Newton Polytope Approximation*", ICML 2020]

Minimization of Neural Nets via Newton Polytope Approximation

Reference:

- P. Misiakos, G. Smyrnis, G. Retsinas and P. Maragos, "*Neural Network Approximation based on Hausdorff distance of Tropical Zonotopes*", Proc. ICLR 2022.
- K. Fotopoulos, P. Maragos and P. Misiakos, "*Structured Neural Network Compression Using Tropical Geometry*", ArXiv 2024.

Neural Network Compression



SoA architectures improve accuracy by adding complexity!

✓ *e.g. Increasing depth/width/connectivity*

Optimize/compress a model with respect to:

■ **#parameters** ■ **FLOPS**

■ *memory footprint* ■ *parallelization*

Solutions:

Bottleneck layers, Shared Weights, Tensor Decomposition, *Quantization, Pruning/Sparsification*

Pruning: Find weights/neurons with the least contribution

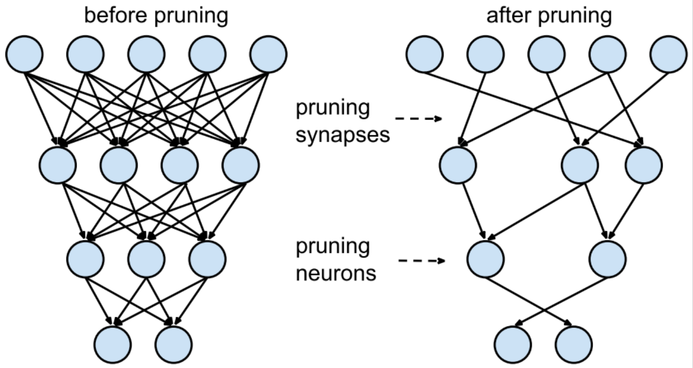
✓ Pruning individual weights vs channels/neurons

Two notable approaches:

- Minimum magnitude
- Minimum inducing error

Iterative process:

1) Prune 2) Re-train



S. Han et al. “Learning both weights and connections for efficient neural network”, NIPS 2015

Pruning via Zonotope Approximation

P. Misiakos,…, P. Maragos, “Neural Network Approximation based on Hausdorff Distance of Tropical Zonotopes”, ICLR, 2022

Approximately equal polytopes ⇒ *Approximately* equivalent polynomials

ReLU NNs ≡ **Tropical rational maps**

[*Zhang et al., 2018*]

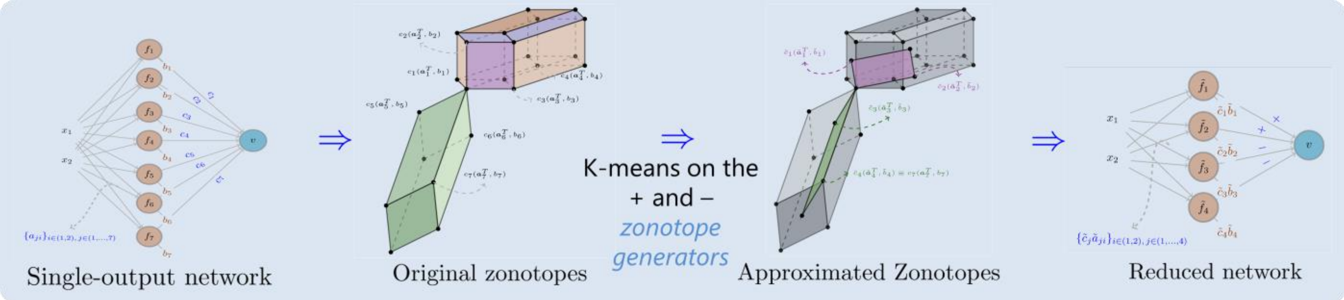
Polynomials & Polytopes equivalence

[*Charisopoulos and Maragos, 2018*]

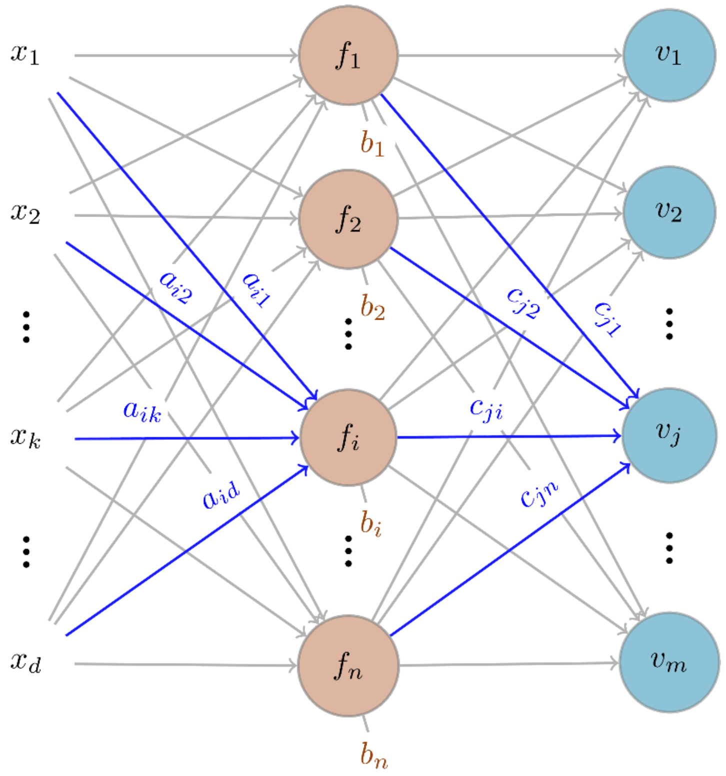
Theorem: NN with 1 hidden layer *Hausdorff distance of zonotopes*

$$\max_{x \in \mathcal{B}} \|v(\boldsymbol{x}) - \tilde{v}(\boldsymbol{x})\|_1 \leq \rho \cdot \left(\sum_{j=1}^{m} \mathcal{H}\left(P_j, \tilde{P}_j\right) + \mathcal{H}\left(Q_j, \tilde{Q}_j\right) \right)$$

Positive and negative *zonotopes:* $P_j = \mathrm{ENewt}\,(p_j)$ $Q_j = \mathrm{ENewt}\,(q_j)$



Neural Network Tropical Geometry: Polynomials



1 hidden layer with ReLU activations

$i-$ th hidden layer neuron

$$f_i(\boldsymbol{x}) = \max\left(\boldsymbol{a}_i^T \boldsymbol{x} + b_i, 0\right)$$

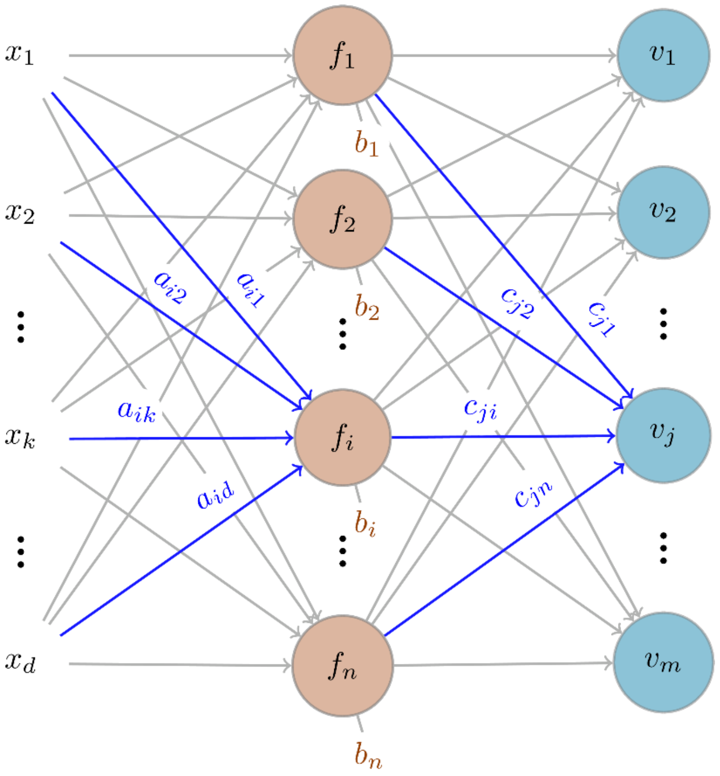
Tropical *polynomial*

$j-$ th output layer neuron

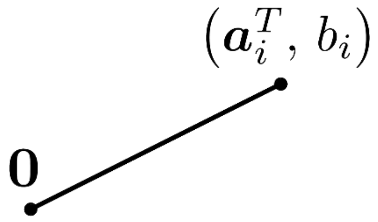
$$\begin{aligned} v_j(\boldsymbol{x}) &= \sum_{i=1}^{n} c_{ji} f_i(x) \\ &= \sum_{c_{ji}>0} |c_{ji}| f_i(\boldsymbol{x}) - \sum_{c_{ji}<0} |c_{ji}| f_i(\boldsymbol{x}) \\ &= p_j(\boldsymbol{x}) - q_j(\boldsymbol{x}) \end{aligned}$$

Tropical *rational* function

Neural Network Tropical Geometry: Polytopes



$$f_i(\boldsymbol{x}) = \max\left(\boldsymbol{a}_i^T \boldsymbol{x} + b_i, 0\right)$$



$\mathrm{ENewt}\,(f_i)$ *is a linear segment*

$$v_j(\boldsymbol{x}) = \sum_{c_{ji}>0} |c_{ji}| f_i(\boldsymbol{x}) - \sum_{c_{ji}<0} |c_{ji}| f_i(\boldsymbol{x})$$
$$= p_j(\boldsymbol{x}) - q_j(\boldsymbol{x})$$

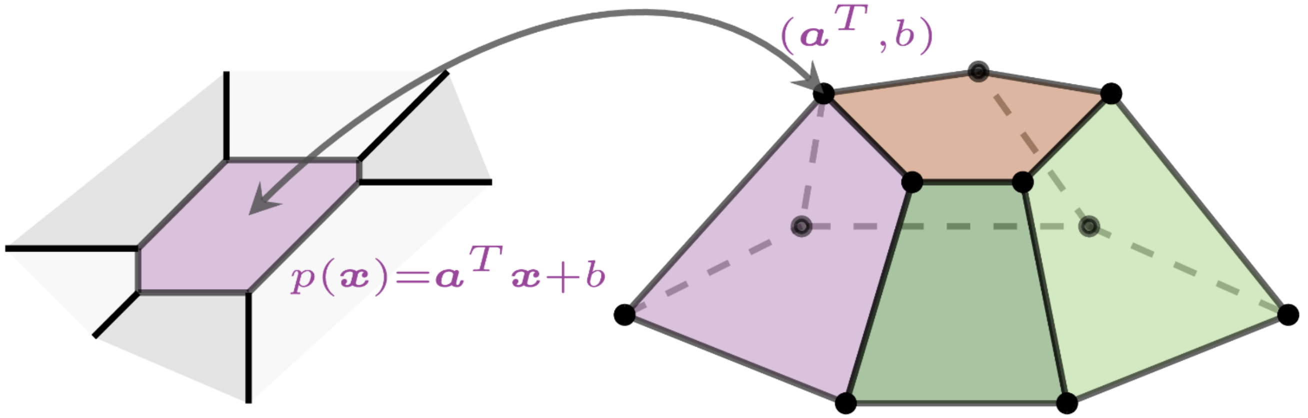
$P_j = \mathrm{ENewt}\,(p_j)$

$Q_j = \mathrm{ENewt}\,(q_j)$

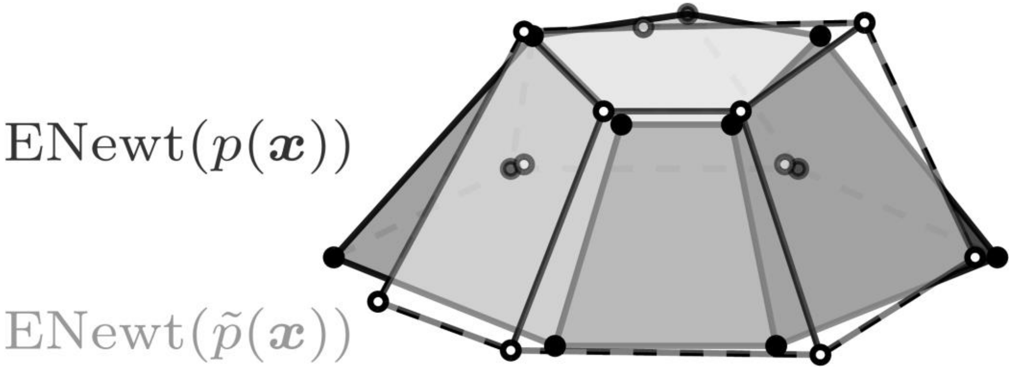
Positive and Negative *zonotopes – or polytopes for deeper NNs*

$c_{ji}\left(\boldsymbol{a}_i^T, b_i\right)$ *Generators* of the zonotopes

Approximate Extended Newton Polytopes



linear regions $\overset{1-1}{\longleftrightarrow}$ *vertices of the upper envelope of the extended Newton polytope*



$\overset{?}{\Longrightarrow}$ $p(\boldsymbol{x}) \approx \tilde{p}(\boldsymbol{x})$

Approximate extended Newton polytopes

Approximate tropical polynomials

Approximating Tropical Polynomials

Proposition Let $p, \tilde{p} \in \mathbb{R}_{\max}[\boldsymbol{x}]$ and consider the polytopes $P = \mathrm{ENewt}(p), \tilde{P} = \mathrm{ENewt}(\tilde{p})$. Then,

$$\max_{x \in \mathcal{B}} |p(\boldsymbol{x}) - \tilde{p}(\boldsymbol{x})| \leq \rho \cdot \mathcal{H}\left(P, \tilde{P}\right)$$

Hausdorff distance of polytopes

Neural Network Approximation Theorem

Theorem: Consider two neural networks $v, \tilde{v}$ with output size m and $P_j, Q_j, \tilde{P}_j, \tilde{Q}_j$ be the positive and negative extended Newton polytopes of $v, \tilde{v}$ respectively. Then,

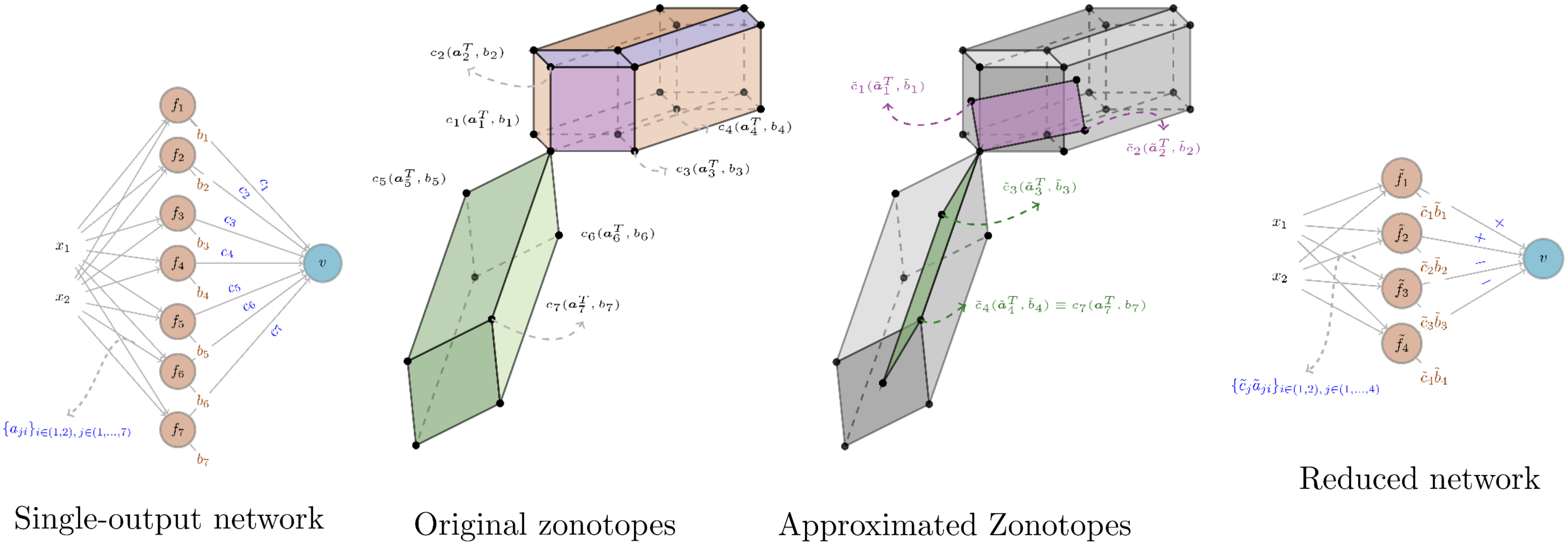
$$\max_{x \in \mathcal{B}} \|v(\boldsymbol{x}) - \tilde{v}(\boldsymbol{x})\|_1 \leq \rho \cdot \left(\sum_{j=1}^{m} \mathcal{H}\left(P_j, \tilde{P}_j\right) + \mathcal{H}\left(Q_j, \tilde{Q}_j\right) \right)$$

Approximately equal polytopes $\Rightarrow$ *Approximately* equivalent networks

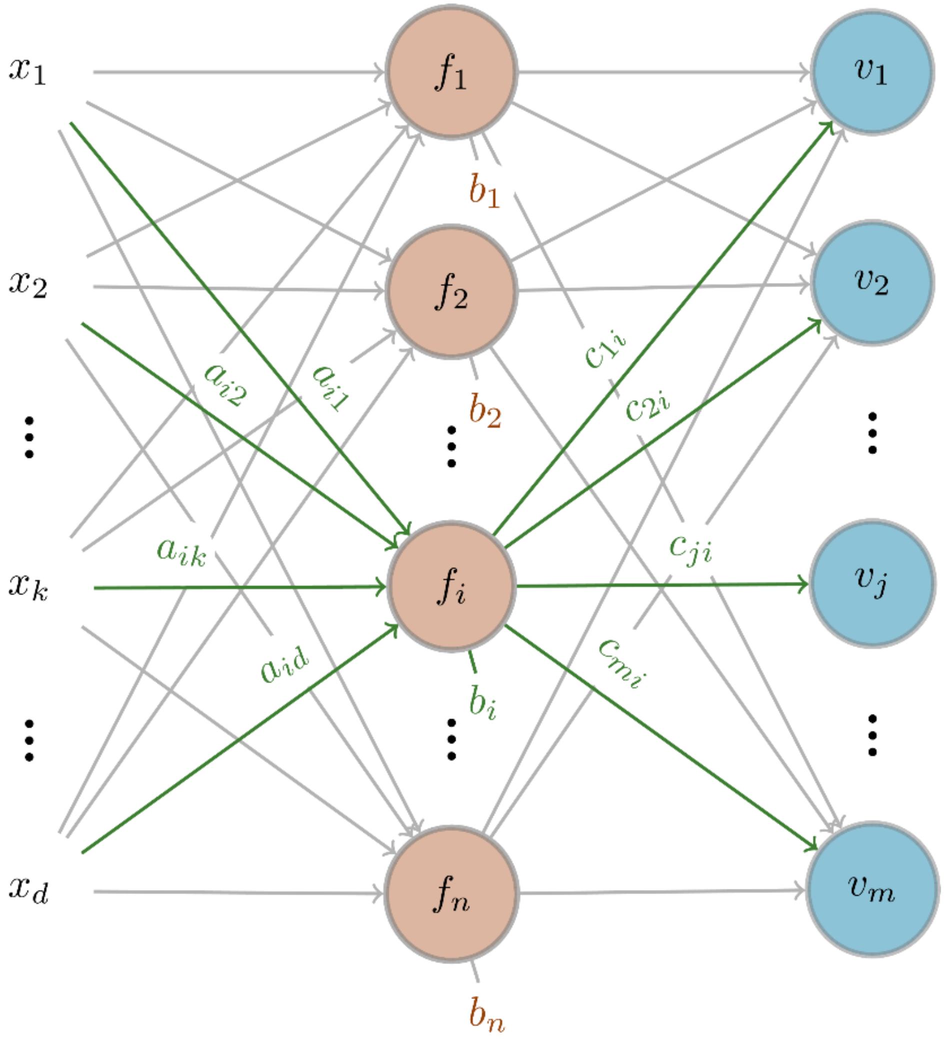
[P. Misiakos, G. Smyrnis, G. Retsinas and P. M., Proc. ICLR 2022.]

Zonotope K-Means

K-means on the positive and negative *zonotope generators*



Neural Path K-means



Generalization for multi-output networks

K-means on the vectors associated with the *neural paths*

Performance Results: Comparison with tropical division

Binary Classification Experiments

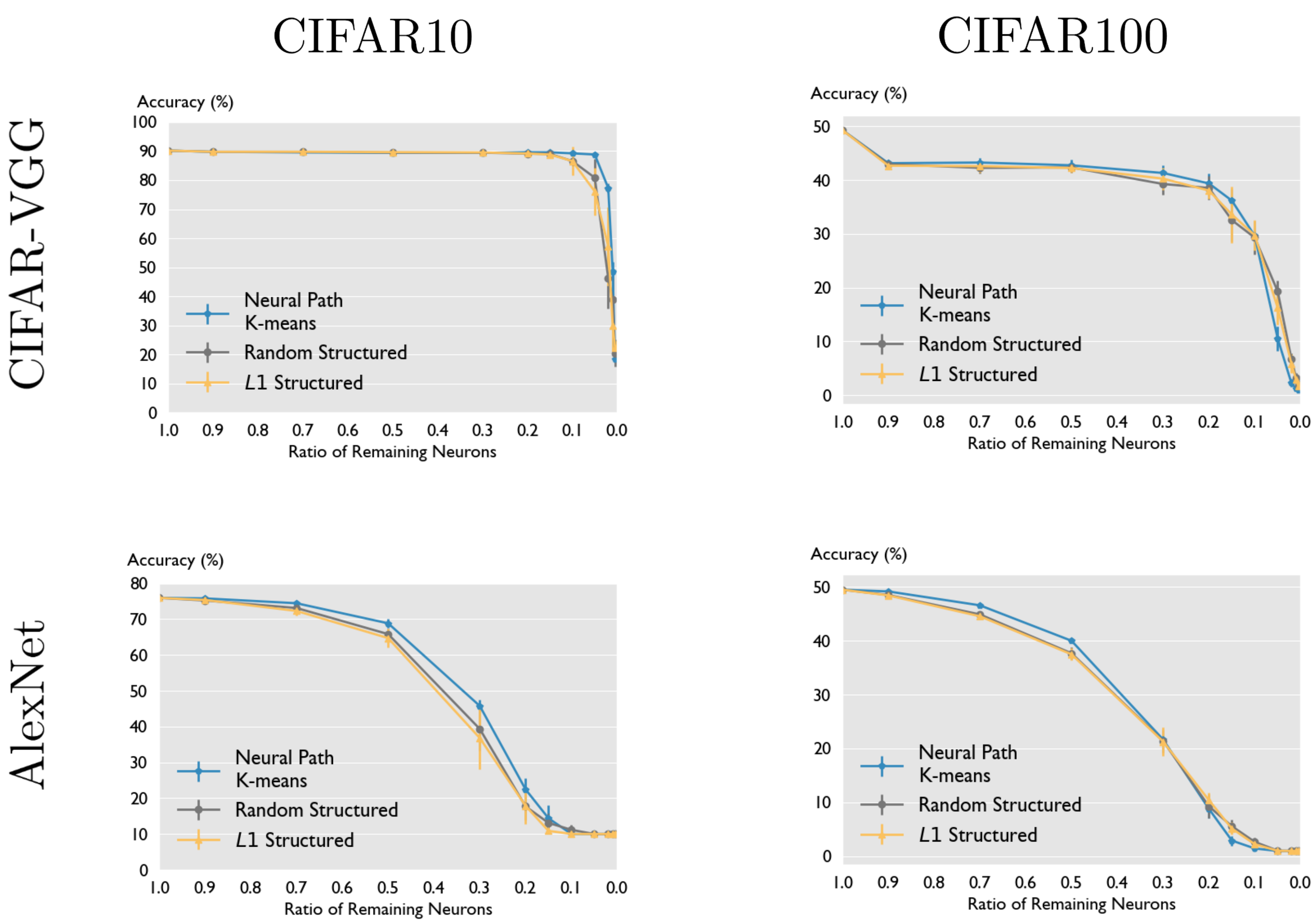
Percentage of Remaining Neurons	MNIST 3/5			MNIST 4/9		
	Smyrnis et al., 2020	Zonotope K-means	Neural Path K-means	Smyrnis et al., 2020	Zonotope K-means	Neural Path K-means
100% (Original)	99.18 ± 0.27	99.38 ± 0.09	99.38 ± 0.09	99.53 ± 0.09	99.53 ± 0.09	99.53 ± 0.09
5%	99.12 ± 0.37	99.42 ± 0.07	99.25 ± 0.04	98.99 ± 0.09	99.52 ± 0.09	99.48 ± 0.15
1%	99.11 ± 0.36	99.39 ± 0.05	99.32 ± 0.03	99.01 ± 0.09	99.46 ± 0.05	99.35 ± 0.17
0.5%	99.18 ± 0.36	99.41 ± 0.05	99.22 ± 0.11	98.81 ± 0.09	99.35 ± 0.24	98.84 ± 1.18
0.3%	99.18 ± 0.36	99.25 ± 0.37	99.19 ± 0.41	98.81 ± 0.09	98.22 ± 1.38	98.22 ± 1.33

Multiclass Classification Experiments

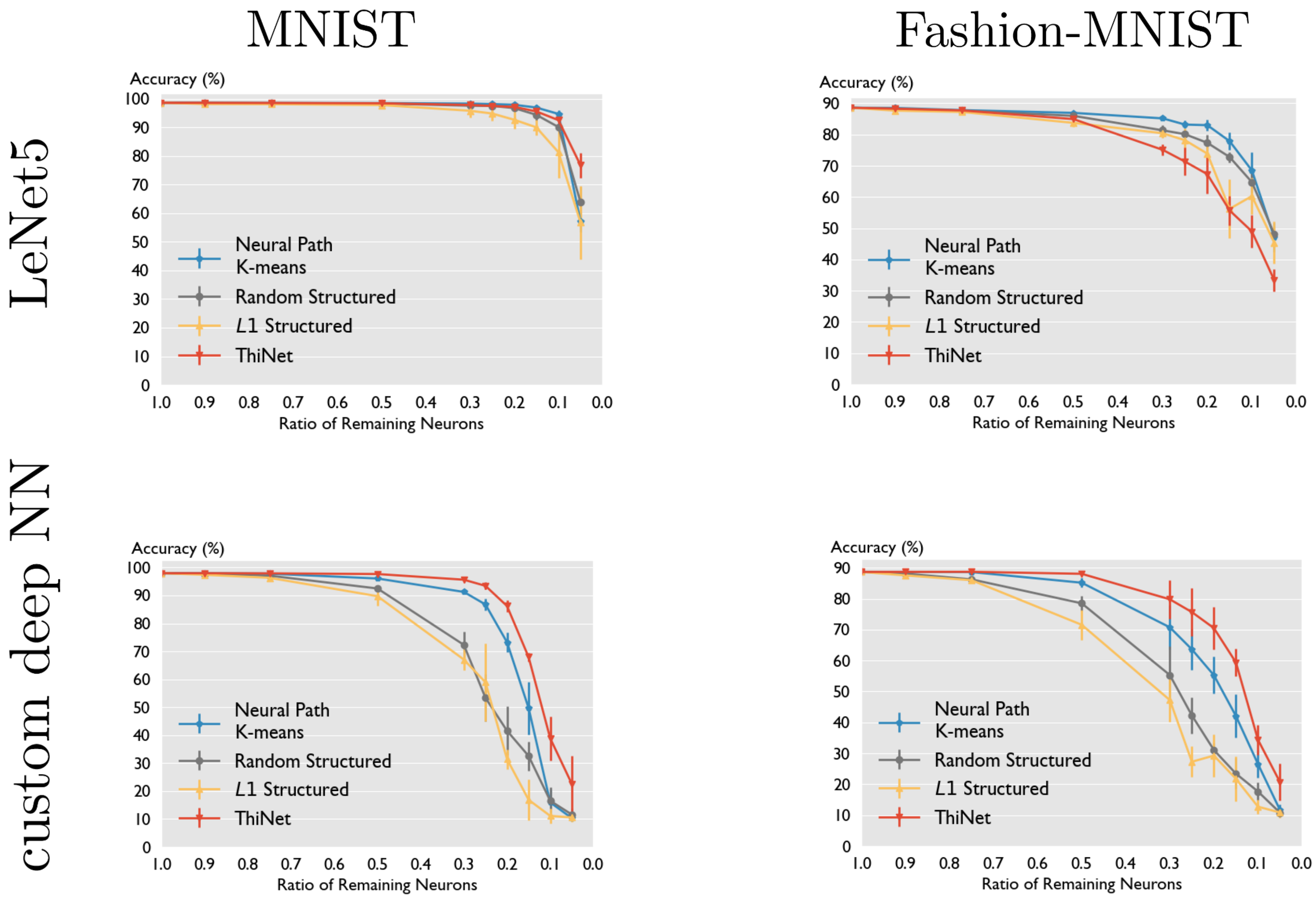
Percentage of Remaining Neurons	MNIST		Fashion-MNIST	
	Smyrnis and Maragos, 2020	Neural Path K-means	Smyrnis and Maragos, 2020	Neural Path K-means
100% (Original)	98.60 ± 0.03	98.61 ± 0.11	88.66 ± 0.54	89.52 ± 0.19
50%	96.39 ± 1.18	98.13 ± 0.28	83.30 ± 2.80	88.22 ± 0.32
25%	95.15 ± 2.36	98.42 ± 0.42	82.22 ± 2.85	86.67 ± 1.12
10%	93.48 ± 2.57	96.89 ± 0.55	80.43 ± 3.27	86.04 ± 0.94
5%	92.93 ± 2.59	96.31 ± 1.29	–	83.68 ± 1.06

[P. Misiakos, G. Smyrnis, G. Retsinas and P. M., "*Neural Network Approximation based on Hausdorff Distance of Tropical Zonotopes*", Proc. ICLR 2022]

Comparison with Baselines

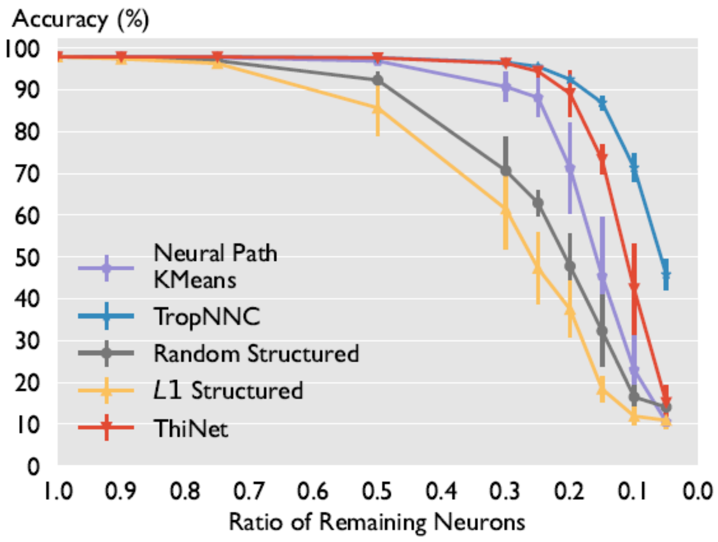


Comparison with ThiNet and Baselines

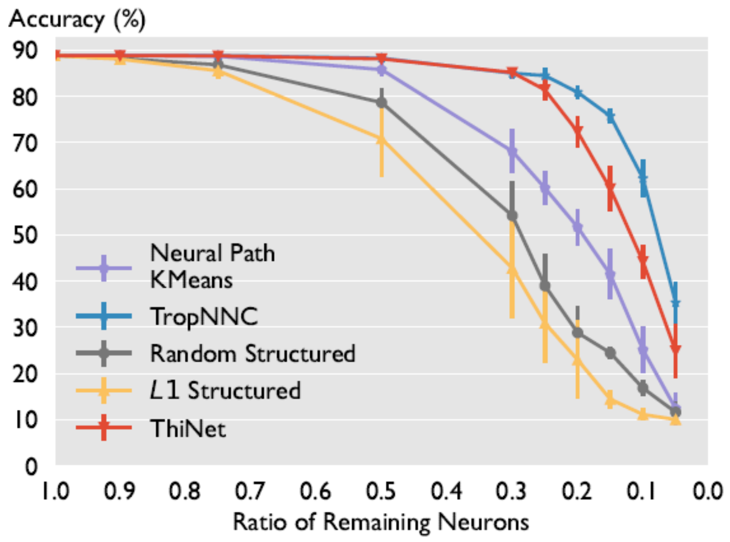


[P. Misiakos, G. Smyrnis, G. Retsinas and P. M., “*Neural Network Approximation based on Hausdorff Distance of Tropical Zonotopes*”, Proc. ICLR 2022]

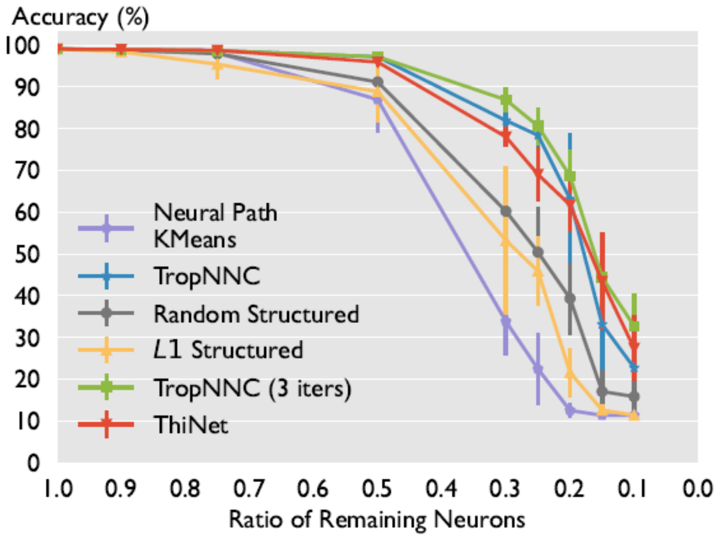
Recent Results on Compressing Linear/Conv ReLU NNs on (F)MINST db



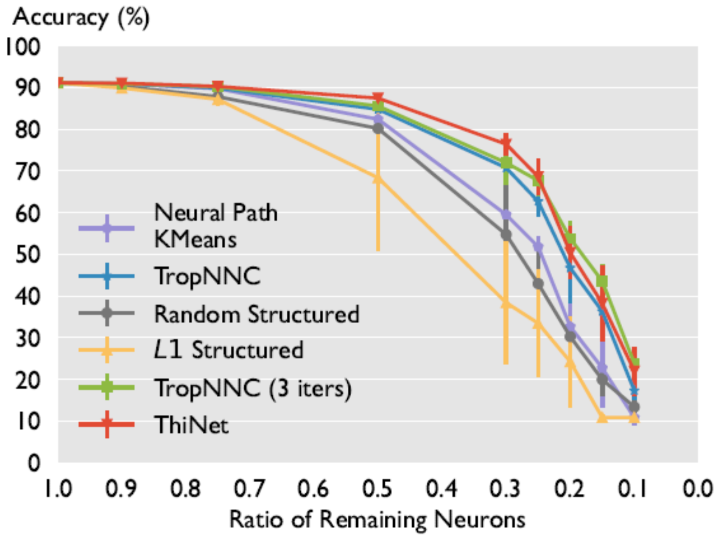
(a) deepNN, MNIST



(b) deepNN, F-MNIST



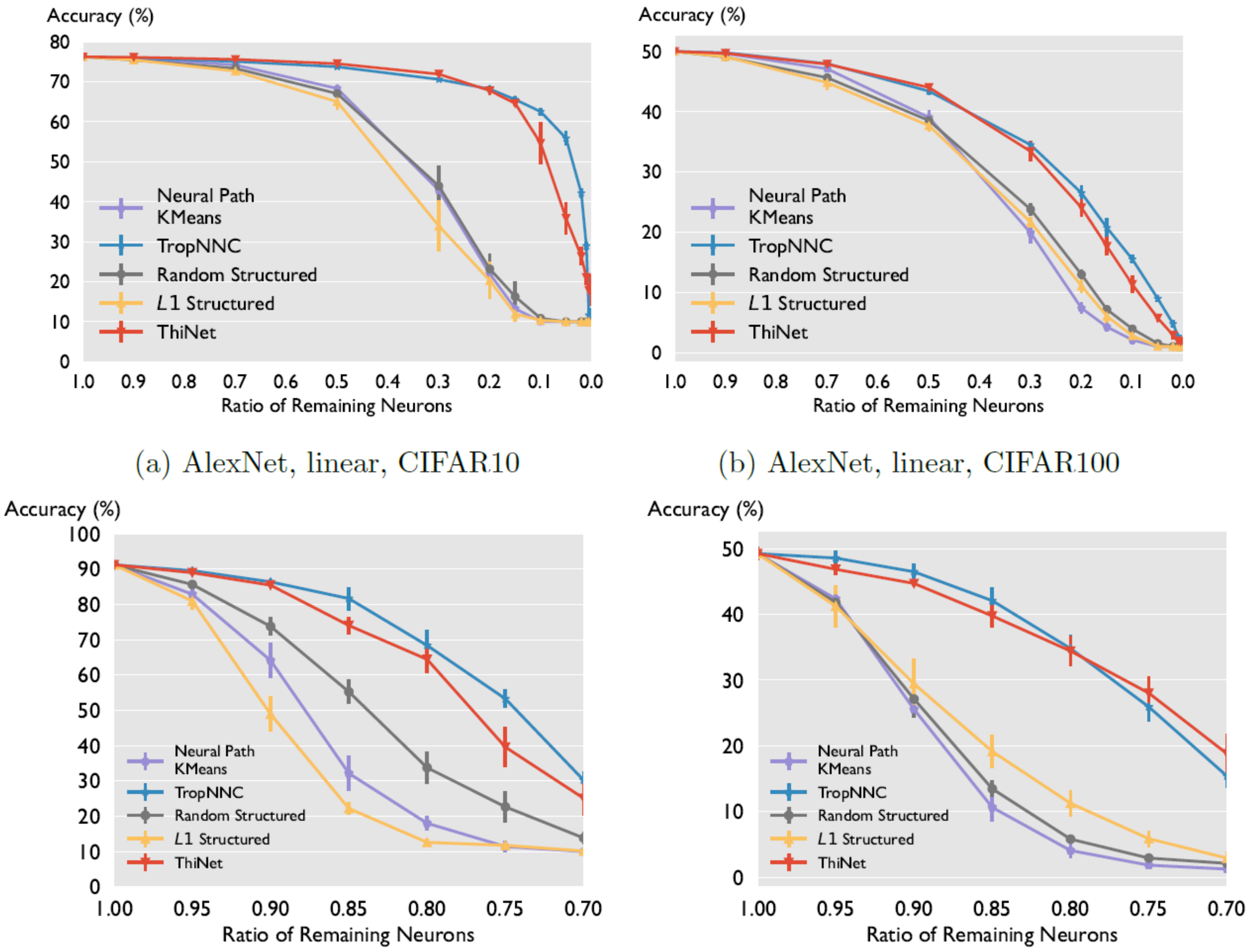
(c) deepCNN2D, MNIST



(d) deepCNN2D, F-MNIST

[K. Fotopoulos, P.M., P. Misiakos, ArXiv 2024.]

Recent Results on Compressing Linear/Conv ReLU NNs on CIFAR db



(a) AlexNet, linear, CIFAR10

(b) AlexNet, linear, CIFAR100

(c) VGG, conv., CIFAR10

(d) VGG, conv., CIFAR100

[K. Fotopoulos, P.M., P. Misiakos, ArXiv 2024.]

Tropical Regression and Piecewise-Linear Surface Fitting

Main References:

- P. Maragos and E. Theodosis, "*Multivariate Tropical Regression and Piecewise-Linear Surface Fitting*", *Proc. ICASSP*, 2020.
- P. Maragos, V. Charisopoulos and E. Theodosis, "*Tropical Geometry and Machine Learning*", *Proceedings of the IEEE*, 2021.

Related:

- A. Magnani and S. Boyd, "*Convex piecewise-linear fitting*," Optim. Eng., 2009.
- J. Hook, "*Linear regression over the max-plus semiring: Algorithms and applications*," ArXiv 2017.
- A. Ghosh et al., "*Max-Affine Regression: Parameter Estimation for Gaussian Designs*", IEEE T-Info. Theory, 2022.

Optimal Regression for Fitting Euclidean vs Tropical Lines

Problem: Fit a curve to data $(x_i, y_i), \quad i = 1, \ldots, m$

Euclidean:

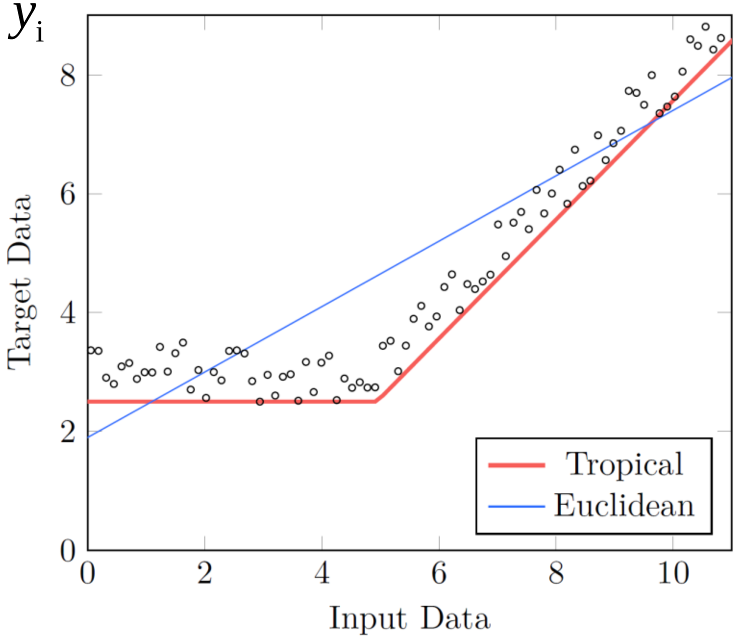
Fit a straight line $y = ax + b$ by minimizing ℓ_2-norm of error:

$$a = \frac{\sum x_i y_i - \left(\sum x_i\right)\left(\sum y_i\right)/m}{\sum (x_i)^2 - \left(\sum x_i\right)^2 / m}, \quad b = \frac{1}{m}\sum_i y_i - ax_i$$

Tropical:

Fit a tropical line $y = \max(a + x, b)$ by minimizing some ℓ_p-norm of error:

Greatest Subsolution: $a = \mathrm{MIN}_i \, y_i - x_i \; , \; b = \mathrm{MIN}_i \, y_i$



Solve Max-plus Equations

- **Problems**:
  (1) Exact problem: Solve $\delta_A(\mathbf{x}) = \mathbf{A} \boxplus \mathbf{x} = \mathbf{b}, \quad \mathbf{A} \in \overline{\mathbb{R}}^{m\times n}, \quad \mathbf{b} \in \overline{\mathbb{R}}^m$

  (2) Approximate Constrained: Min $\|\mathbf{A} \boxplus \mathbf{x} - \mathbf{b}\|_{p=1\dots\infty}$ s.t. $\mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}$

- **Theorem**: (a) The **greatest (sub)solution** of (1) and unique solution of (2) is

$$\hat{\mathbf{x}} = \varepsilon_A(\mathbf{b}) = \mathbf{A}^* \boxplus' \mathbf{b} = [\bigwedge_i b_i - a_{ij}], \quad \mathbf{A}^* \triangleq -\mathbf{A}^T$$

  and yields the **Greatest Lower Estimate (GLE)** of data $\mathbf{b}$:

  Lattice Projection: $\delta_A(\varepsilon_A(\mathbf{b})) = \mathbf{A} \boxplus (\mathbf{A}^* \boxplus' \mathbf{b}) \leq \mathbf{b}$

  (b) **Min Max Absolute Error (MMAE) unconstrained unique solution**:

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} + \mu, \quad \mu = \|\mathbf{A} \boxplus \hat{\mathbf{x}} - \mathbf{b}\|_\infty / 2$$

- **Geometry**: Operators δ, ε are vector dilation and erosion, and the GLE $\mathbf{b} \mapsto \delta\varepsilon(\mathbf{b})$ is an opening (lattice projection).

- **Complexity**: $O(mn)$

Sparse solutions: [Tsiamis & Maragos 2019], [Tsilivis et al. 2021]

Optimally Fitting Tropical Lines to Data

Problem: Fit a tropical line $y = \max(a + x, b)$ to noisy data (x_i, f_i), $i = 1, \ldots, m$, where $f_i = y_i + \text{error}$ by minimizing $\ell_{1,\ldots,\infty}$ norm of error:

Greatest Subsolution (GLE): $\hat{w} = (\hat{a}, \hat{b})$, $\hat{a} = \underset{i}{\text{MIN}}\ f_i - x_i$, $\hat{b} = \underset{i}{\text{MIN}}\ f_\text{i}$

Min Max Abs. Error (MMAE) Solution: $\tilde{w} = \hat{w} + \mu$, $\mu = \|\text{GLE error}\|_\infty /2$

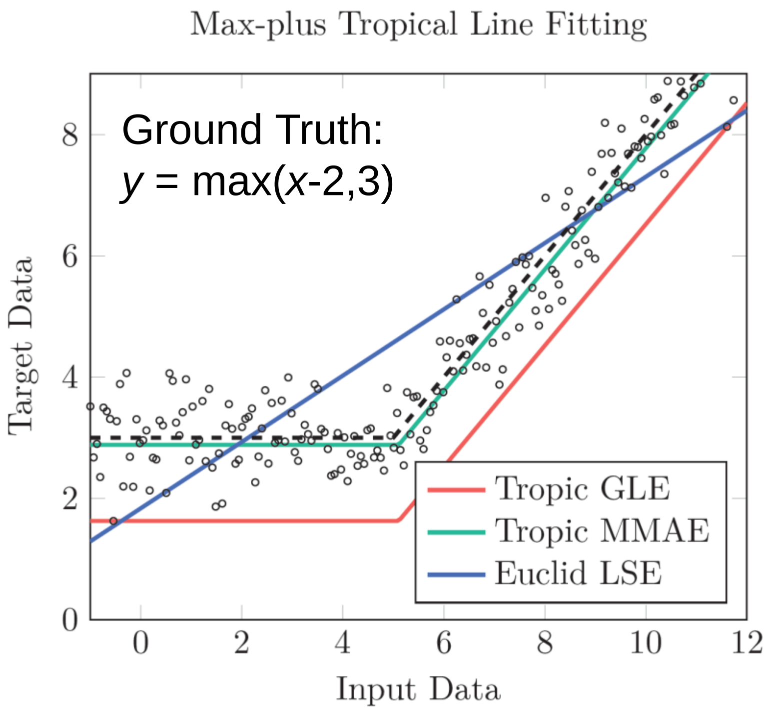
$$\underbrace{\begin{bmatrix} x_1 & 0 \\ \vdots & \vdots \\ x_m & 0 \end{bmatrix}}_{\mathbf{X}} \boxplus \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}}_{\mathbf{f}} \Longrightarrow \underbrace{\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}}_{\hat{\mathbf{w}}} = \underbrace{\begin{bmatrix} \bigwedge_i f_i - x_i \\ \bigwedge_i f_i \end{bmatrix}}_{\mathbf{X}^* \boxplus' \mathbf{f}}$$

Numerical Examples of Optimally Fitting Tropical Lines to Data

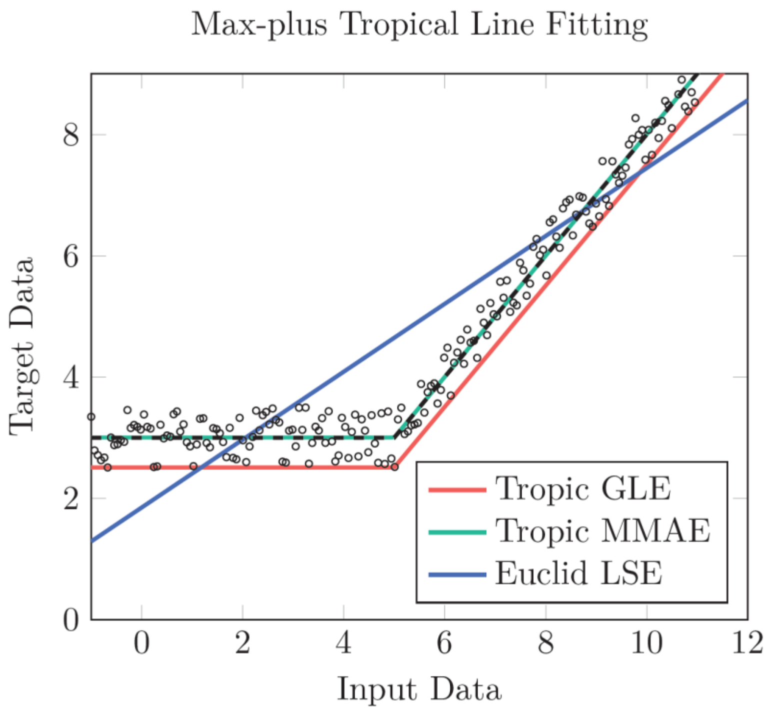
Problem: Fit a tropical line $y = \max(a + x, b)$ to noisy data (x_i, f_i), $i = 1, \ldots, m = 200$, where $f_i = y_i + \text{error}$ by minimizing $\ell_{1,\ldots,\infty}$ of error:

Greatest Subsolution (GLE): $\hat{w} = (\hat{a}, \hat{b})$, $\hat{a} = \underset{i}{\mathrm{MIN}}\; f_i - x_i$, $\hat{b} = \underset{i}{\mathrm{MIN}}\; f_i$

Min Max Abs. Error (MMAE) Solution: $\tilde{w} = \hat{w} + \mu$, $\mu = \| \text{GLE error} \|_\infty /2$



(a) T-line with Gaussian Noise



(b) T-line with Uniform Noise

Optimal Fitting 1D Max-Plus Tropical Polynomials to Data

We wish to fit a tropical polynomial $f(x)$ to given data $(x_i, f_i) \in \mathbb{R}^2$, $i = 1, ..., m$,

$$f(x) = \max(a_0 x + b_0, a_1 x + b_1, a_2 x + b_2, \ldots, a_K x + b_K) = \bigvee_{k=0}^{K} a_k x + b_k$$

where $a_k \in \mathbb{Z}$, $b_k \in \mathbb{R}$, and $f_i = f(x_i) + \text{error}$, by minimizing the ℓ_1 error norm. For example, if $a_k = k - 1$ we have a K-degree tropical polynomial curve:

$$f(x) = \max(b_0, x + b_1, 2x + b_2, \ldots, Kx + b_K)$$

The equations to solve for finding the optimal parameters **b** become:

$$\underbrace{\begin{bmatrix} a_0 x_1 & a_1 x_1 & a_2 x_1 & \cdots & a_K x_1 \\ a_0 x_2 & a_1 x_2 & a_2 x_2 & \cdots & a_K x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_0 x_m & a_1 x_m & a_2 x_m & \cdots & a_K x_m \end{bmatrix}}_{\mathbf{X}} \boxplus \underbrace{\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}}_{\mathbf{f}}$$

Optimal solution for minimum ℓ_1 error

$$\begin{bmatrix} \hat{b}_0 \\ \hat{b}_1 \\ \vdots \\ \hat{b}_K \end{bmatrix} = \hat{\mathbf{b}} = \mathbf{X}^* \boxplus' \mathbf{f} = \begin{bmatrix} -a_0 x_1 & -a_0 x_2 & \cdots & -a_0 x_m \\ -a_1 x_1 & -a_1 x_2 & \cdots & -a_1 x_m \\ \vdots & \vdots & \vdots & \vdots \\ -a_K x_1 & -a_K x_2 & \cdots & -a_K x_m \end{bmatrix} \boxplus' \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} = \begin{bmatrix} \bigwedge_{i=1}^{m} f_i - a_0 x_i \\ \bigwedge_{i=1}^{m} f_i - a_1 x_i \\ \vdots \\ \bigwedge_{i=1}^{m} f_i - a_K x_i \end{bmatrix}$$

Optimal Fitting Max-Plus Tropical Planes to Data

Problem: Fit a tropical plane $z = \max(a + x, b + y, c)$ to noisy data (x_i, y_i, f_i),
where $f_i = z_i$+error, $i = 1, \ldots, m = 100$, by minimizing $\ell_{1,\ldots,\infty}$ norm of error:

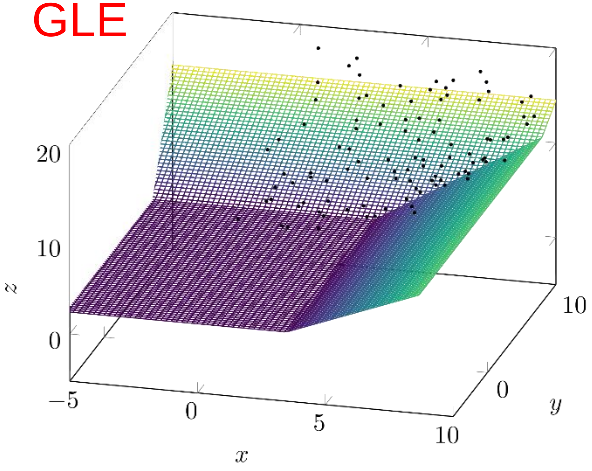
Greatest Subsolution (GLE): $\hat{w} = (\hat{a}, \hat{b}, \hat{c})$

Min Max Abs. Error (MMAE) Solution: $\tilde{w} = \hat{w} + \mu, \quad \mu = \| \text{GLE error} \|_\infty /2$

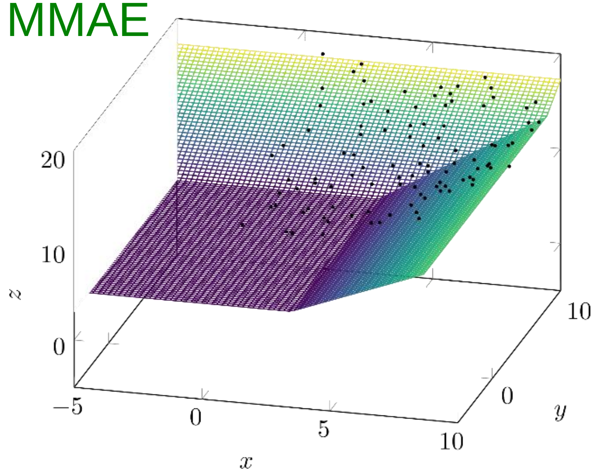
$$\underbrace{\begin{bmatrix} x_1 & y_1 & 0 \\ \vdots & \vdots & \vdots \\ x_m & y_m & 0 \end{bmatrix}}_{\mathbf{X}} \boxplus \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix}}_{\mathbf{f}} \Longrightarrow \underbrace{\begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix}}_{\hat{\mathbf{w}}} = \underbrace{\begin{bmatrix} \bigwedge_i f_i - x_i \\ \bigwedge_i f_i - y_i \\ \bigwedge_i f_i \end{bmatrix}}_{\mathbf{X}^* \boxplus' \mathbf{f}}$$

Ground Truth:
$z = \max(x + 5, y + 7, 9)$
Noise: $N(0,1)$

GLE



MMAE



Optimal Fitting 2D Higher-degree Tropical Polynomials to Data

Data (noisy paraboloid):

3D tuples $(x_i, y_i, f_i) \in \mathbb{R}^3$

$f_i = x_i^2 + y_i^2 + \varepsilon_i,$

$(x_i, y_i) \sim \text{Unif}[-1,1]$

$\varepsilon_i \sim \mathcal{N}(0, 0.25^2)$

Model:

Fit K-rank 2D trop. polynomial

$$p(x,y) = \text{MAX}_{k=1}^{K} \{a_k x + b_k y + c_k\}$$

by minimizing error $\| f_i - p(x_i, y_i) \|_\infty$

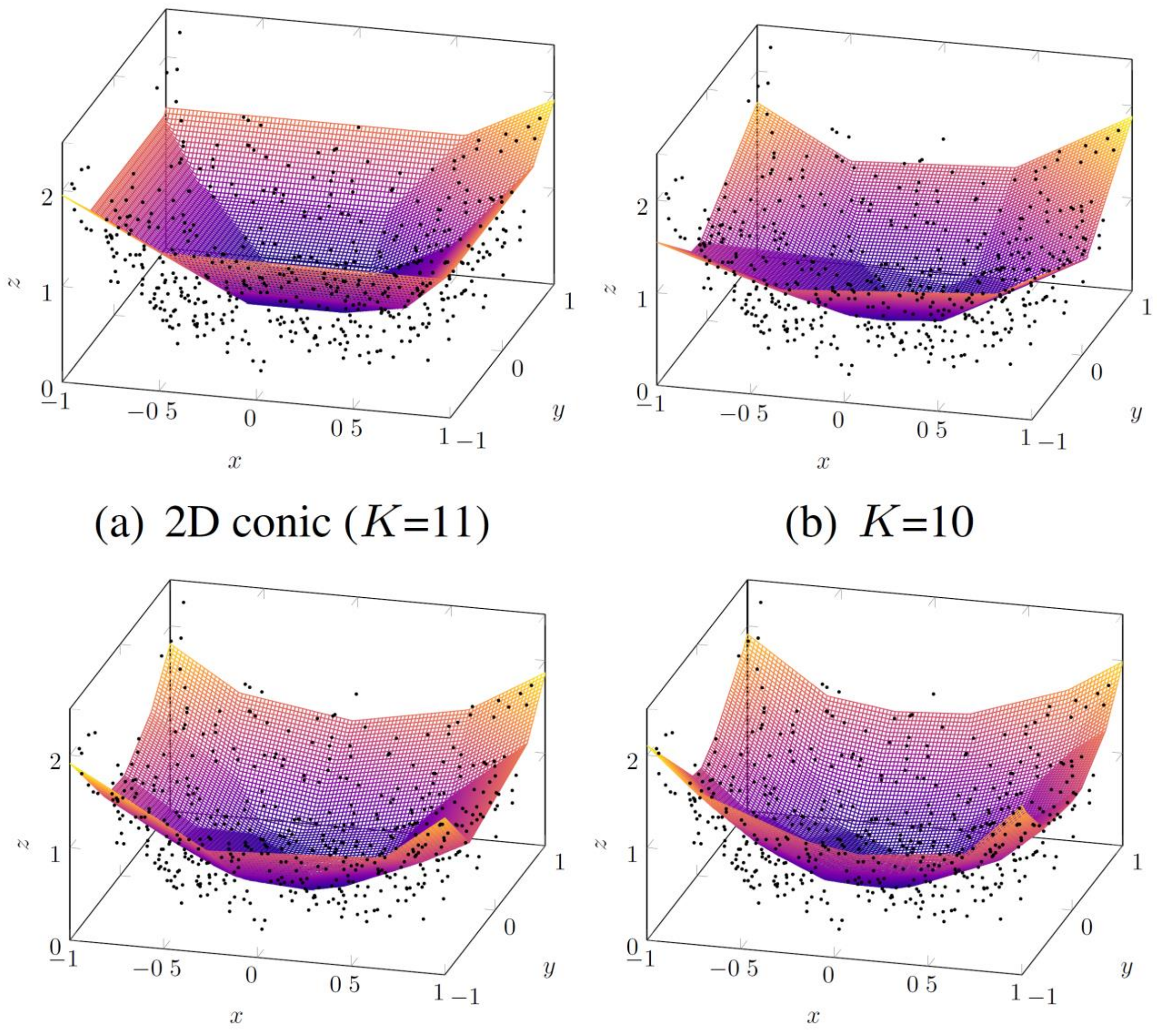
Estimation algorithm:

$K-$means on data gradients $\to a_k, b_k$

solve max-plus eqns $\to c_k$

Complexity: $\approx$ Linear

O(#data, #dimensions)



(a) 2D conic (K=11) (b) K=10

(c) K=25 (d) K=100

Optimal Solutions of Max-Plus Equations and Sparsity

References:

- A. Tsiamis and P. Maragos, "*Sparsity in Max-plus Algebra*", Discrete Events Dynamic Systems, 2019.
- N. Tsilivis, A. Tsiamis and P. Maragos, "*Sparsity in Max-plus Algebra And Applications in Multivariate Convex Regression*", ICASSP, 2021.
- N. Tsilivis, A. Tsiamis and P. Maragos, "*Sparse Approximate Solutions to Max-Plus Equations*", Int'l Conf. Discrete Geometry and Mathematical Morphology, 2021.
- N. Tsilivis, A. Tsiamis and P. Maragos, "*Toward a Sparsity Theory on Weighted Lattices*", Journal of Mathematical Imaging and Vision, 2022.

Sparsest Solution to Max-Plus Equation

[Tsiamis & Maragos, DEDS 2019]

- A sparse vector $\boldsymbol{x} \in \mathbb{R}^n_{\max}$ has many $-\infty$ elements.
- Let supp$(\boldsymbol{x})$ be the support (the set of finite indices)
- We solve the following problems:

Exact solution

$$\min_{\boldsymbol{x} \in \mathbb{R}^n_{\max}} \quad |\text{supp}(\boldsymbol{x})|$$
$$\text{subject to} \quad \boldsymbol{A} \boxplus \boldsymbol{x} = \boldsymbol{b}$$

Approximate solution

$$\min_{x \in \mathbb{R}^n_{\max}} \quad |\text{supp}(\boldsymbol{x})|$$
$$\text{subject to} \quad \|\boldsymbol{b} - \boldsymbol{A} \boxplus \boldsymbol{x}\|_1 \leq \epsilon$$
$$\boldsymbol{A} \boxplus \boldsymbol{x} \leq \boldsymbol{b}$$

- NP-complete problem (~minimum set cover). Use greedy algorithms.
- Submodularity tools provide suboptimality bounds.
- Extensions to other Lp norms [Tsilivis, Tsiamis & Maragos, DGMM 2021]

Sparsest Solution to Max-Plus Equation – General Norms

- Extensions to other Lp norms [Tsilivis, Tsiamis & Maragos, DGMM 2021]

$$\min_{\mathbf{x} \in \mathbb{R}^n_{\max}} |\mathrm{supp}(\mathbf{x})|, \ \text{s.t.} \ \|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}\|_p^p \leq \epsilon, \qquad \mathbf{A} \boxplus \mathbf{x} \leq \mathbf{b}. \tag{4}$$

- Greedy algorithm, as in p=1 – similar analysis.
- Provides heuristic for sparse solutions without the monotonicity constraint:

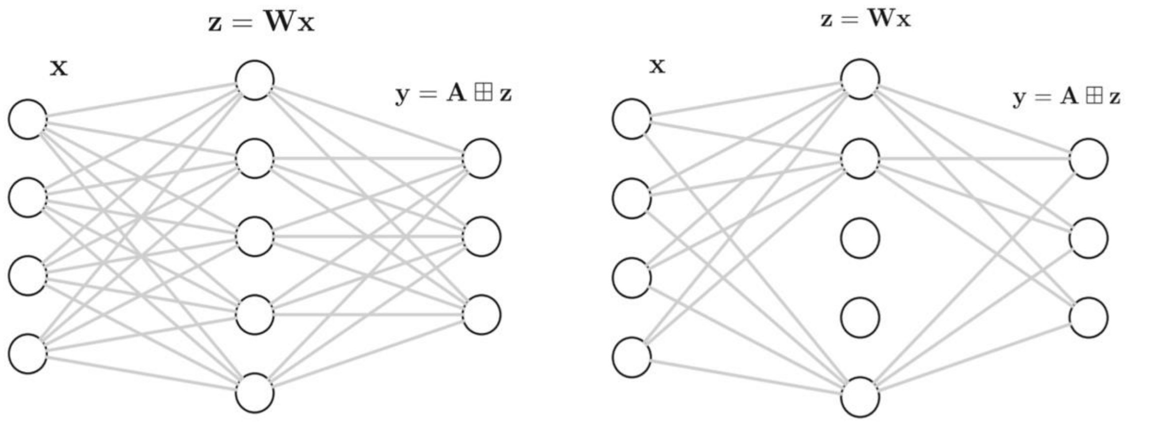
$$\mathbf{x}_{\mathrm{SMMAE}} = \mathbf{x}^* + \frac{\|\mathbf{b} - \mathbf{A} \boxplus \mathbf{x}^*\|_\infty}{2},$$

where $\mathbf{x}^*$ *is a solution of problem (4) with fixed* (p, ϵ).

- **Best** approximation error among all vectors with same *support*.
- Applications:
  - Morphological Neural Networks Minimization
  - Convex Regression

Morphological Neural Networks Minimization

- Sparse Solutions to Max-Plus Equations: neuron pruning in Morphological Neural Networks.



(a) A simple Max-plus block with $d = 4, n = 5, k = 3$.

(b) The same Max-plus block, after pruning two neurons from its first layer.

- Experiments on image classification datasets:

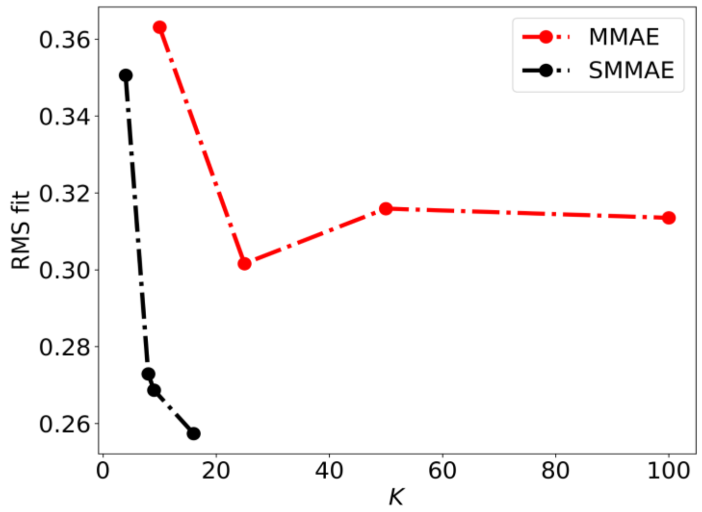
Same performance,
Less neurons

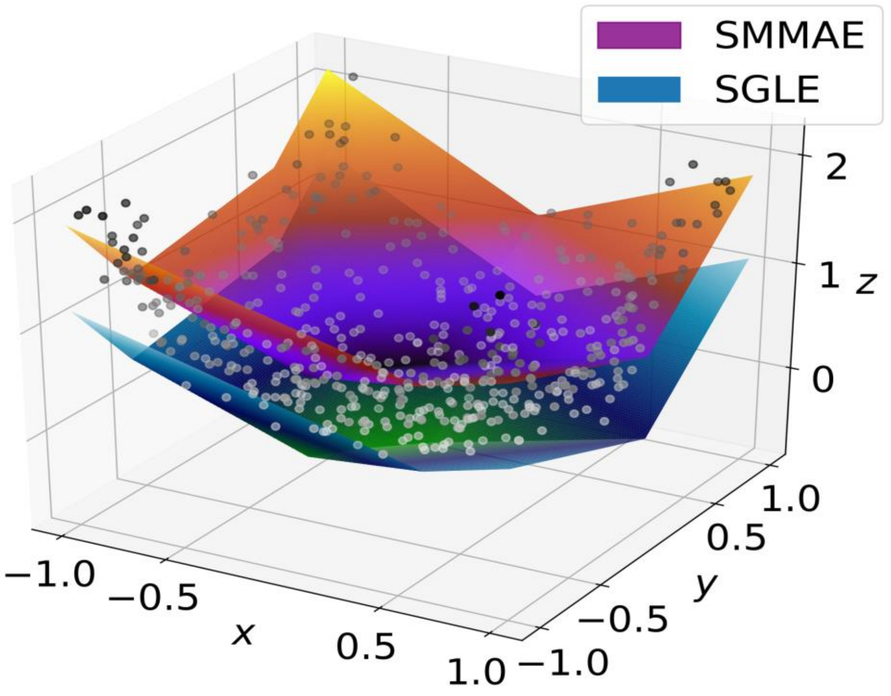
	MNIST		FashionMNIST	
	64	128	64	128
Full model	92.21	92.17	79.27	83.37
Pruned ($n = 10$)	92.21	92.17	79.27	83.37

[Tsilivis, Tsiamis & Maragos, DGMM 2021]

Multivariate Convex Regression

- Convex functions as piecewise linear $$p(\mathbf{x}) = \bigvee_{k=1}^{K} \mathbf{a}_k^{\intercal}\mathbf{x} + b_k,$$
- Approximation from data by solving max-plus systems of equations.
- Sparsity = Few affine regions.
- Improved results over non-sparse approximation:



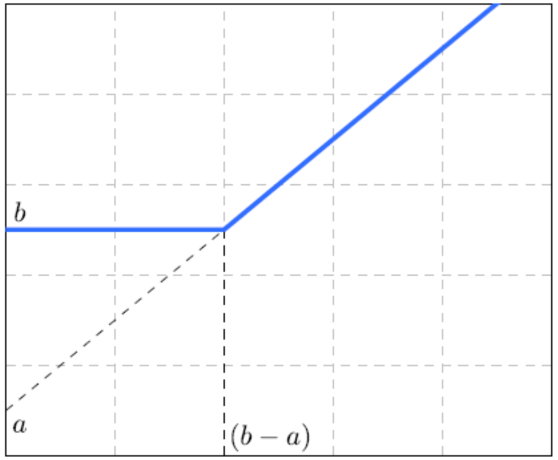


[Tsilivis, Tsiamis & Maragos, ICASSP 2021]

Generalized Tropical Versions of Lines & Planes over Max-* Algebras

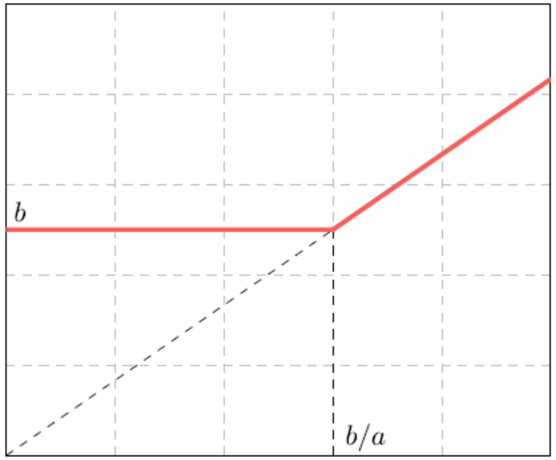
Max-plus Tropical Line

$$y = \max(a + x, b)$$



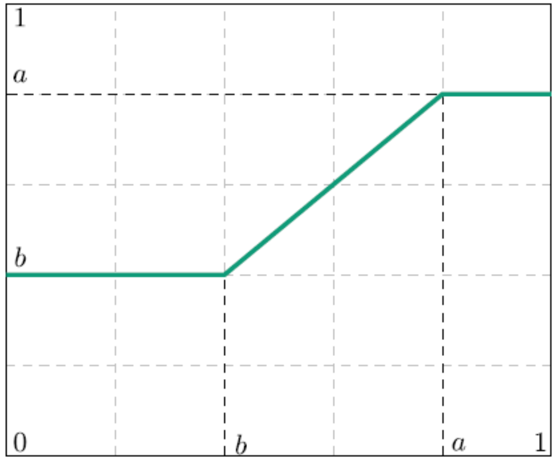
Max-product Tropical Line

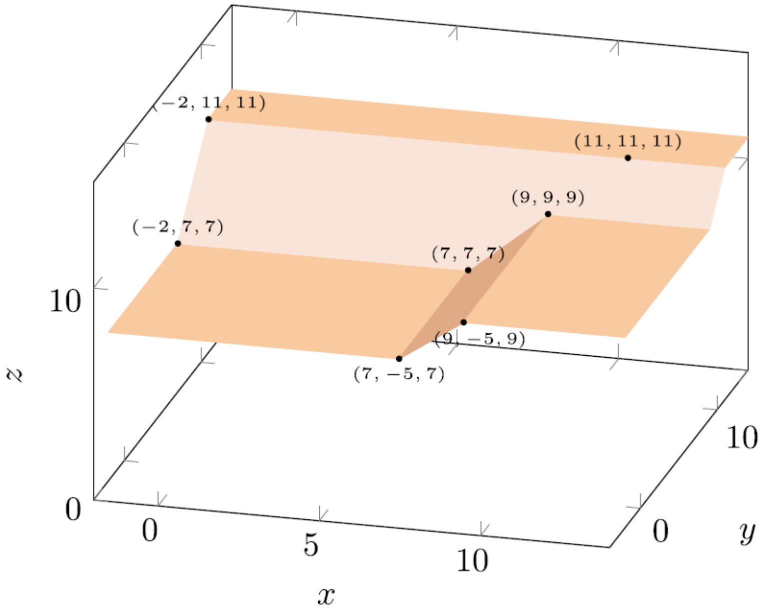
$$y = \max(a \cdot x, b)$$



Max-min Tropical Line

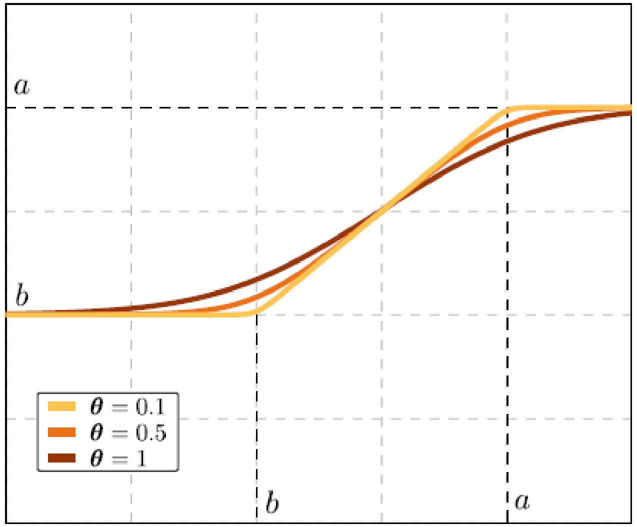
$$y = \max(\min(a, x), b)$$





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

SoftMax-SoftMin Tropical Line



Conclusions

- **Tropical Geometry**, and its underlying **max-plus algebra**, provide principled and insightful tools for analysis of NNs with PWL activations and other ML systems.

- **NNs** with nonlinear max/min-plus nodes: similar performance and superior compression ability compared to linear counterparts. Trained via CCP or SGD/Adam.

- **Tropical Regression**: Tropical Polynomials for multidimensional data fitting using PWL functions. Low-complexity algorithm from optimal solutions of max-plus eqns.

- **NN Minimization**: TG offers effective and insightful tools for compression of NNs.

- **Future work**: deeper networks, nonconvex settings, more general functions using max-* algebra on weighted lattices. Tropical Approximation: theory & applications.

For more information, demos, and current results:

http://robotics.ntua.gr and http://cvsp.cs.ntua.gr