Static analysis of ReLU neural networks with tropical polyhedra

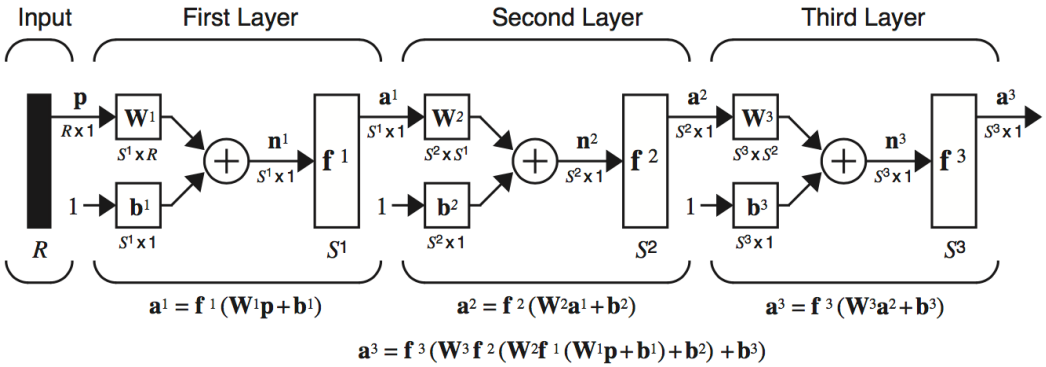
Eric Goubault, Sébastien Palumby, Sylvie Putot, Louis Rustenholz*, Sriram Sankaranarayanan

Ecole Polytechnique, CNRS, IP-Paris and **University of Colorado Boulder**
* (Now **Universidad Politécnica de Madrid (UPM)** and **IMDEA Software Institute**)

SAS (COVID Track), December 7th, 2022
Initially at SAS, October 19th, 2021

Neural nets - quick recap and setup here

Feedforward neural nets - multi-layer perceptron



- Transfer functions are non-linear :
  - differentiable: "SoftPlus", sigmoid, (hyperbolic) tan-sigmoid etc.
  - non-differentiable: the ReLU function $f(x) = max(x, 0)$, max-pool etc.

We concentrate here on specificities of non-differentiable activation functions, ReLU here.

Properties

We develop here a range analysis, the basic block for proving some specifications correct and proving robustness properties

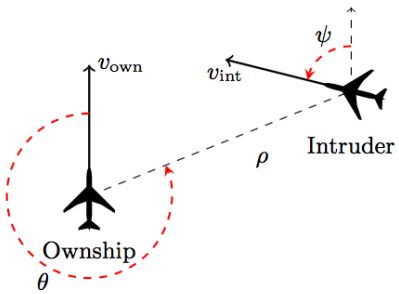
Training by e.g. backpropagation algorithms (not part of this talk - some interesting verification issues there, still, e.g. concerning convergence, propagation of uncertainties see our former work on these issues!)

Static analysis of neural nets – Specifications?

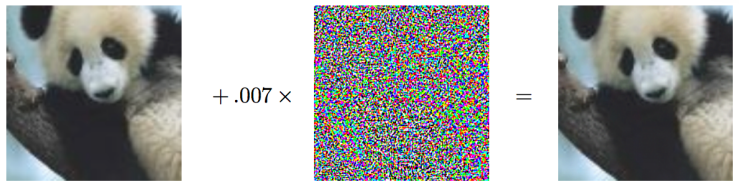
What properties could we want to check?

- Verification on "all" possible executions, before being actually used.
- In general very difficult to specify what a neural net has to compute.
- Nevertheless:
  - There are cases in which we may have some specifications, e.g. for classification or in the context of neural net based control of a CPS.
  - Robustness issues, very important and accessible to formal verification. For example, prove the absence of adversarial examples, e.g. formalised with Lipschitz continuity.

  All based on more basic range (set-based) analysis.
  - Could be used for verifying heuristical robust training techniques, or for training itself.
  - Probabilistic properties are another interesting class of specifications.



ACAS Xu



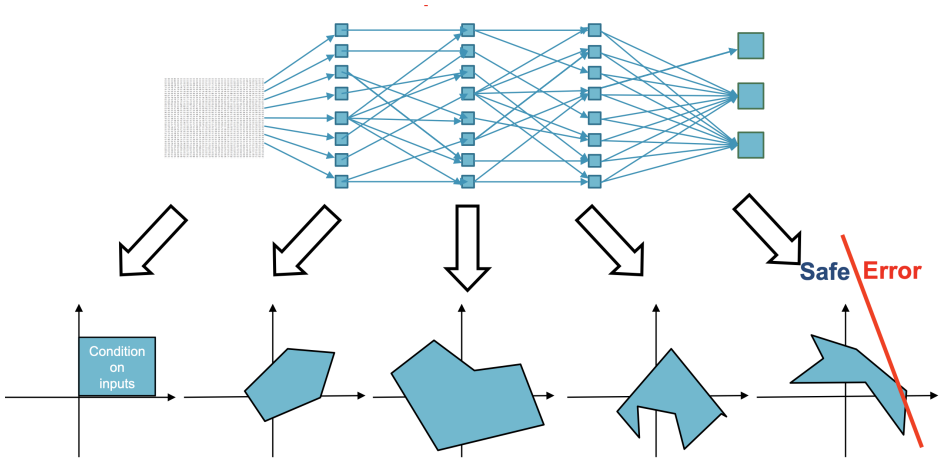
Adversarial examples

(from "Explaining and Harnessing Adversarial Examples", Goodfellow, Shlens & Szegedy, 2015)

Abstract interpretation based methods for neural nets

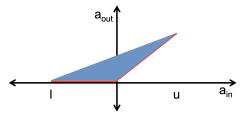
Propagate abstract sets of states through layers (seen as program control points)

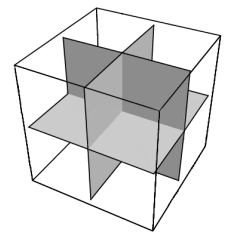
E.g. polyhedral/zonotopic methods of AI2/deepZ/deepPoly etc.



(from "Fast and Effective Robustness Certification", G. Singh, T. Gehr, M. Mirman, M. Püschel, M. Vechev, NIPS 2018)

Inherent non-convexity makes verification NP-hard (combinatorial blowup) $\Rightarrow$ abstraction using a "tractable" class of sets ; But convexification leads to too many "false alarms"





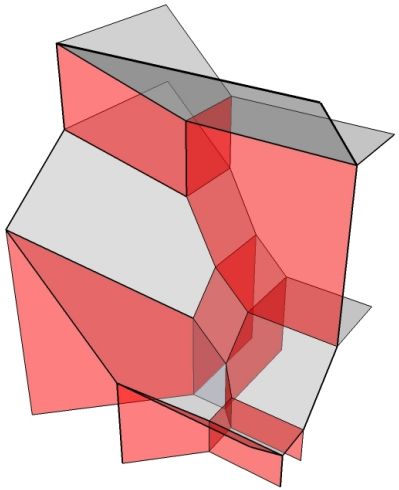
ReLU networks

What do they compute?

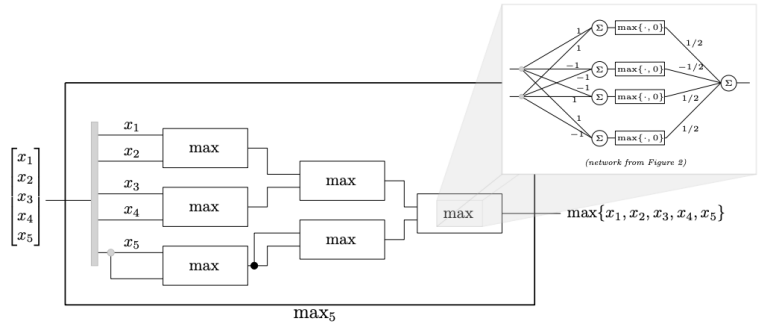
Computes tropical rational functions!

i.e. $max_j(\sum_i \alpha^i_j x_i) - max_k(\sum_l \beta^l_k x_l)$

See e.g. Liwen Zhang, Gregory Naitzat and Lek-Heng Lim, "Tropical Geometry of Deep Neural Networks", 2018. (also min-max functions $min_i max_j(\sum_l \alpha^l_{i,j} x_l)$, also continuous piecewise linear functions, see e.g. Rockafellar)



Decision boundary / Tropical hypersurface



Similar for max-pool

Tropical algebra

Tropical semi-ring

Classical ring $\mathbb{R}$ with ordinary sum and multiplications replaced by max-plus semiring $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$:

- Addition $x \oplus y := max(x, y)$
- Multiplication $x \otimes y := x + y$

Properties

- Almost a ring: neutral elements $\mathbb{1} := 0$ for $\otimes$, $\mathbb{0} = -\infty$ for $\oplus$, inverse for $\otimes$ on $\mathbb{R}_{max} \backslash \{\mathbb{0}\}$ but not for $\oplus$
- Fits in with the usual order $\leq$ on $\mathbb{R}$, extended to $\mathbb{R}_{max}$: $x \leq y$ if and only if $x \oplus y = y$.

Very versatile, used in algebraic geometry, timed event systems etc.

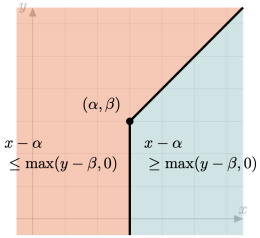
Tropical half-space

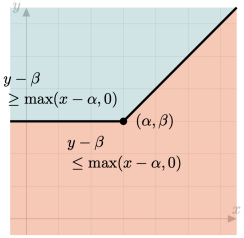
Similar to classical half-spaces, defined as the set of points satisfying

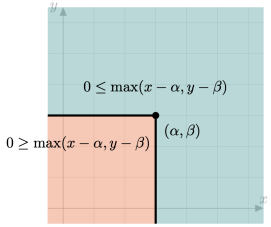
$$\bigoplus_{1 \leq i \leq d} a_i \otimes x_i \oplus c \leq \bigoplus_{1 \leq i \leq d} b_i \otimes x_i \oplus d$$

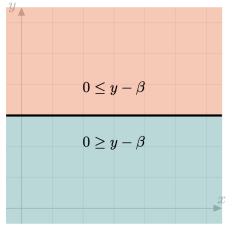
Some examples from tropical geometry

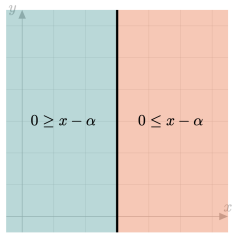
Lines and half-spaces in $\mathbb{R}^2$

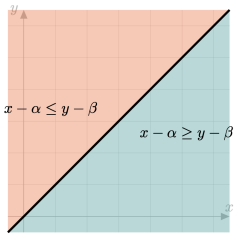




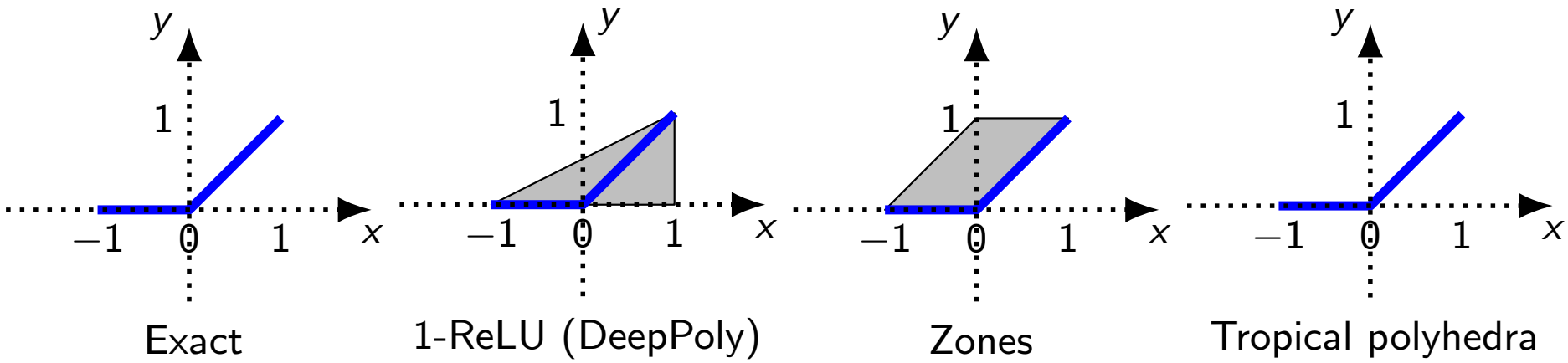








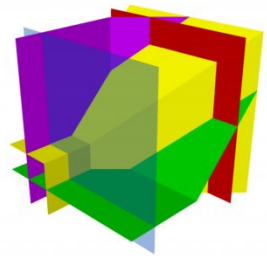
ReLU $x \to max(x, 0) = x \oplus 0$ is tropically linear!



Tropical double description method

Constraints

$max(\alpha_0, max(\alpha_1 + v_1, \ldots, \alpha_d + v_d)) \leq max(\beta_0, max(\beta_1 + v_1, \ldots, \beta_d + v_d))$

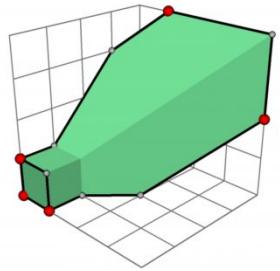


Convex combination of extreme points and generators

$x \in C$ (a max-plus polyhedron) can be expressed as the sum of the convex combination of n extreme points $p_1, \ldots, p_n \in C$ and p extreme generators (rays) $g_1, \ldots, g_p$ with $n + p \leq d + 1$:

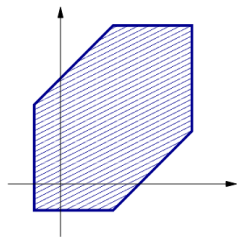
$$x = max\{max_{i=1}^{n}(\lambda_i + p_i), max_{j=1}^{p} g_j\}$$

(with $max(\lambda_1, \ldots, \lambda_n) = 0$ - the tropical unit - all λs are $\geq -\infty$ - the tropical 0).



Relationship with zones and disjunctive analyses

- Max-plus polyhedra are disjunctions of zones (polytopes with "faces at 0 or 45°")
- Each max-invariant expresses a disjunction of zone invariants over the variables v_i: $max(\alpha_0, max(\alpha_1 + v_1, \ldots, \alpha_d + v_d)) \leq max(\beta_0, max(\beta_1 + v_1, \ldots, \beta_d + v_d))$ is equivalent to
  $\bigvee_{1 \leq i \leq d, \beta_i \neq -\infty} \left[\left(\wedge_{1 \leq j \leq d} \alpha_j - \beta_i \leq v_i - v_j \right) \wedge (\alpha_0 - \beta_i \leq v_i) \right] \vee \left[\wedge_{1 \leq i \leq d, \alpha_i \neq -\infty} v_i \leq \beta_0 - \alpha_i \right]$



See e.g. Xavier Allamigeon, Stéphane Gaubert, Eric Goubault, "The Tropical Double Description Method", STACS 2010.

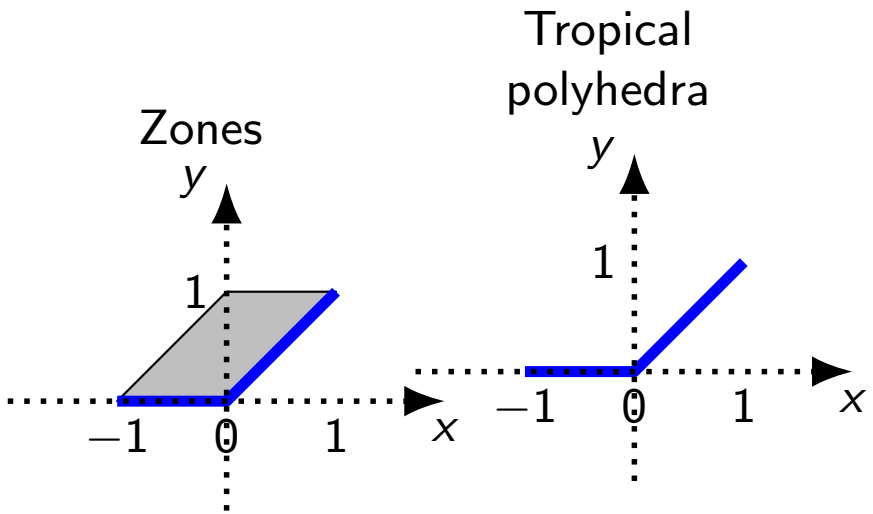
From tropical polyhedra to zones (see e.g. SAS 2008)

- Tropical polyhedron $\mathcal{H}$ with p extreme generators and rays $A_1, \ldots, A_p$,
- Put in homogeneous coordinates in $\mathbb{R}^{n+1}$ by adding as last component 0 to the coordinates of the extreme generators, and $-\infty$ to the last component, for extreme rays
- A matrix of generators for $\mathcal{H}$ stripped out of rows consisting only of $-\infty$ entries
- A/A residuated matrix:

$$(A/A)_{i,j} = \min_{1 \leq k \leq p} a_{i,k} - a_{j,k}$$

Then, smallest zone containing $\mathcal{H}$:

$$x_i - x_j \geq (A/A)_{i,j} \qquad \forall i, j = 1, \ldots, n$$
$$(A/A)_{i,n+1} \leq x_i \leq -(A/A)_{n+1,i} \qquad \forall i = 1, \ldots, n$$



Extreme points $A_1 = (-1, 0)$

and $A_2 = (1, 1)$

From zones to tropical polyhedra (this paper!)

Internal tropical representation of closed zones

- Let $H_{ext} \subset \mathbb{R}^n$ be the n-dimensional zone defined by the conjunction of the $(n+1)^2$ inequalities

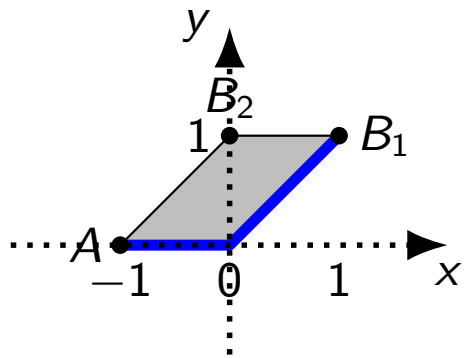
$$\bigwedge_{0 \leq i,j \leq n} (x_i - x_j \leq c_{i,j}),$$

where $\forall i, j \in [0, n]$, $c_{i,j} \in \mathbb{R} \cup \{+\infty\}$.

- Assume that this representation is *closed*, then H_{ext} is equal to the tropical polyhedron H_{int} defined, with internal representation, as the tropical convex hull of the following extreme points (and no extreme ray):

$$A = (a_i)_{1 \leq i \leq n} := (-c_{0,1}, \ldots, -c_{0,n})$$
$$B_k = (b_{ki})_{1 \leq i \leq n} := (c_{k,0} - c_{k,1}, \ldots, c_{k,0} - c_{k,n}),$$
$$k=1, \ldots, n,$$

Zones/Tropical polyhedron



Abstract operations for the analysis of ReLU networks

Operations

- Transfer functions for
  - ReLU layers,
  - Linear layers.
- Will be convenient to have join, meet, some conversions.
- Test against affine guards.

Abstract operations for the analysis of ReLU networks

Operations

- Transfer functions for
  - ReLU layers,
  - Linear layers.
- Will be convenient to have join, meet, some conversions.
- Test against affine guards.

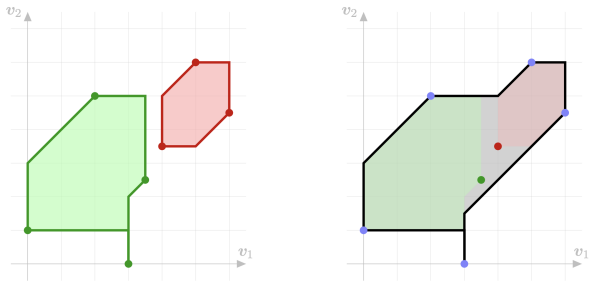
- ReLU layers are linear in the tropical sense: the abstraction is exact and easy to compute.

Abstract operations for the analysis of ReLU networks

Operations

- Transfer functions for
  - ReLU layers,
  - Linear layers.
- Will be convenient to have join, meet, some conversions.
- Test against affine guards.

- ReLU layers are linear in the tropical sense: the abstraction is exact and easy to compute.
- Many operations available from previous work, see e.g. Allamigeon, Gaubert, Goubault, "Inferring Min and Max Invariants Using Max-Plus Polyhedra", SAS 2008

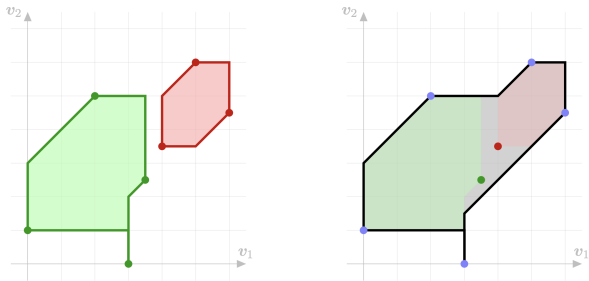


Abstract operations for the analysis of ReLU networks

Operations

- Transfer functions for
  - ReLU layers,
  - Linear layers.
- Will be convenient to have join, meet, some conversions.
- Test against affine guards.

- ReLU layers are linear in the tropical sense: the abstraction is exact and easy to compute.
- Many operations available from previous work, see e.g. Allamigeon, Gaubert, Goubault, "Inferring Min and Max Invariants Using Max-Plus Polyhedra", SAS 2008

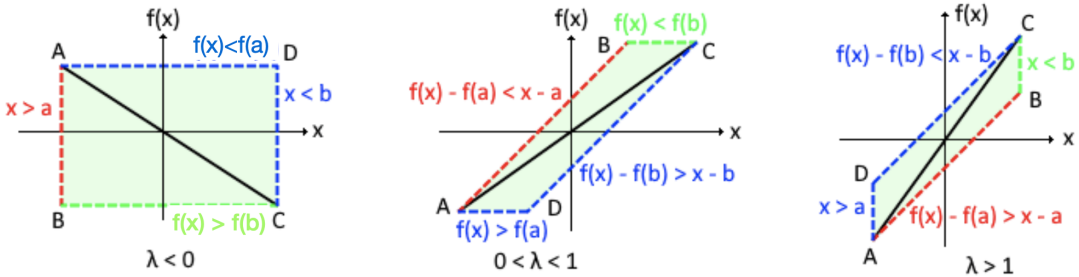


- But it is harder to deal with the linear layer.
  Compared to other analyses, **we approach the problem from the other end.**

Abstraction of linear layers: zones to zones

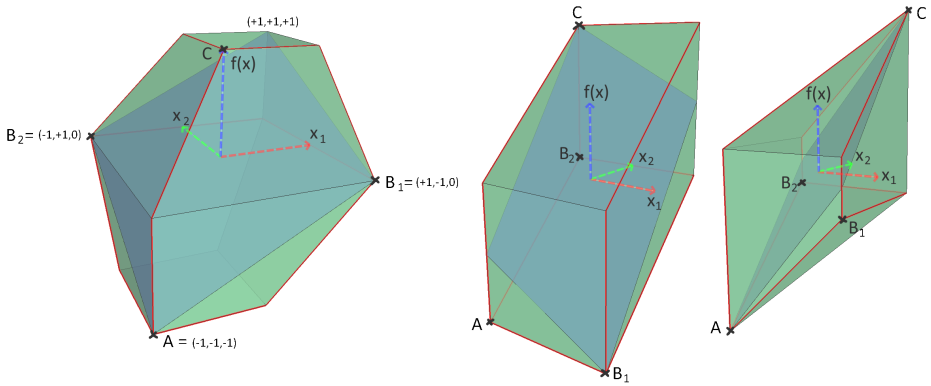
For simplicity, we don't do directly the best abstraction from tropical polyhedra to tropical polyhedra, but simply the best zone abstraction, and make conversions.

Example (dimension 1). Consider $f(x) = \lambda x + b : \mathbb{R} \to \mathbb{R}$, on an interval.



Example (dimension 2). Consider $f : \mathbb{R}^2 \to \mathbb{R}$.

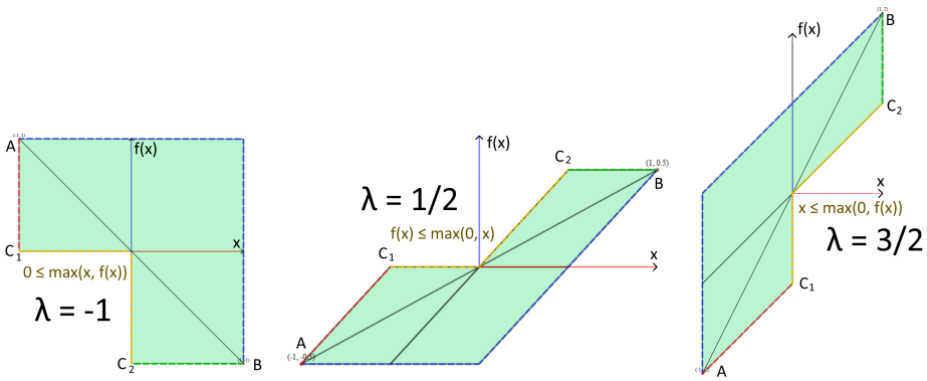
6 cases, depending on the values of λ_1 and λ_2. Represented as tropical polyhedra using only 4 extreme points and 4 inequalities (instead of at least 8 and 6 in the classical case).



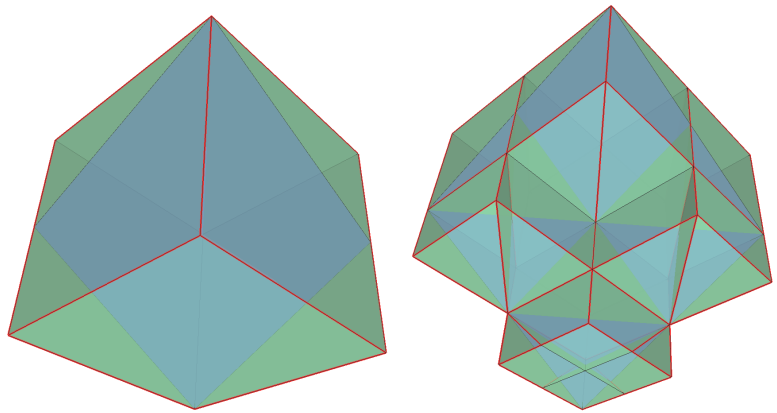
Abstraction of linear layers: (union of) zones to (union of) zones

To recover precision, we use **subdivisions** of the input.

Example (dimension 1).



Example (dimension 2).



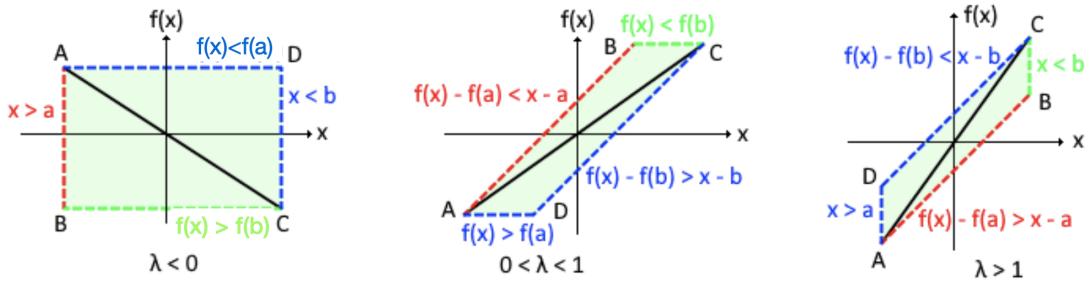
Abstraction of linear layers: refining the domain

Let's go back to the 1-dimensional example.

Consider $f(x) = \lambda x + b : \mathbb{R} \to \mathbb{R}$, on an interval

Three cases

- $\lambda < 0$ (case $\delta_{i,j} = 0$), best abstraction for the graph of f is a square (cannot encode any dependency between $f(x)$ and x)
- $0 \leq \lambda \leq 1$, middle picture
- $\lambda > 1$ right hand side.



- To increase precision, we go from the domain of $(\max, +)$ polyhedra, made of *zones*, to $(\max, +, -)$ **polyhedra**, made of *octagons*.
- We had the tropical monomials "x^0" and "x^1".
  By adding "x^{-1}", we recover some negative slopes.
- More in the paper. Future work: higher degrees, semi-algebraic tropical varieties.

Optimal approximation of a linear layer by a zone/tropical polyhedron

From the best zone abstraction (Proposition 3 of the paper) $\mathcal{H}_f$ of graph $\mathcal{G}_f = \{(x_1, \ldots, x_m, y_1, \ldots, y_n) \mid \underline{x}_j \leq x_j \leq \overline{x}_j,\ y_i = f_i(x_1, \ldots, x_m)\} \subseteq \mathbb{R}^{m+n}$ of the linear function $f : \mathbb{R}^m \to \mathbb{R}^n$ ($i \in [1, n]$ and $j \in [1, m]$):

Theorem 1: external representation of the tropical abstraction

Using a system of $m + n + 1$ affine tropical inequalities:

$$\begin{array}{rcl} \max(x_1 - \overline{x}_1, \ldots, x_m - \overline{x}_m, y_1 - M_1, \ldots, y_n - M_n) & \leq & 0 \\ \max(0, y_1 - M_1 + \delta_{1,j}, \ldots, y_n - M_n + \delta_{n,j}) & \leq & x_j - \underline{x}_j \\ \max(0, x_1 - \overline{x}_1 + \delta_{i,1}, \ldots, x_n - \overline{x}_n + \delta_{i,n}, y_1 - d_{i,1}, \ldots, y_n - d_{i,n}) & \leq & y_i - m_i \end{array}$$

Theorem 2: internal representation of the tropical abstraction

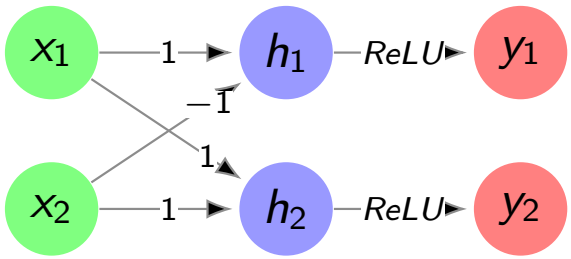
$\mathcal{H}_f$ is the tropical convex hull of $m + n + 1$ extreme points:

$$\begin{array}{rcl} A & = & (\underline{x}_1, \ldots, \underline{x}_m, m_1, \ldots, m_n) \\ B_1 & = & (\overline{x}_1, \underline{x}_2, \ldots, \underline{x}_m, m_1 + \delta_{1,1}, \ldots, m_n + \delta_{n,1}) \ldots \\ B_m & = & (\underline{x}_1, \ldots, \underline{x}_{m-1}, \overline{x}_m, m_1 + \delta_{1,m}, \ldots, m_n + \delta_{n,m}) \\ C_1 & = & (\underline{x}_1 + \delta_{1,1}, \ldots, \underline{x}_m + \delta_{1,m}, M_1, c_{1,2}, \ldots, c_{1,n}) \ldots \\ C_n & = & (\underline{x}_1 + \delta_{n,1}, \ldots, \underline{x}_m + \delta_{n,m}, c_{n,1}, \ldots, c_{n,n-1}, M_n) \end{array}$$

An illustration of our static analysis

Simple neural net

Input layer | Linear layer | Output ReLU layer



- Neural network with 2 inputs x_1 and x_2 in [-1,1] and 2 outputs.
- Linear layer with biases $[-1, 1]$, defined by
  $h_1 = x_1 - x_2 - 1$,
  $h_2 = x_1 + x_2 + 1$
- Followed by a ReLU layer with neurons y_1 and y_2;
  $y_1 = max(0, x_1 - x_2 - 1)$ and
  $y_2 = max(0, x_1 + x_2 + 1)$.

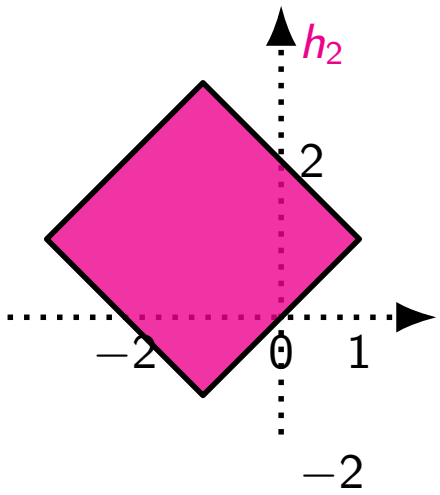
Ranges

We want to check two properties on this simple neural network

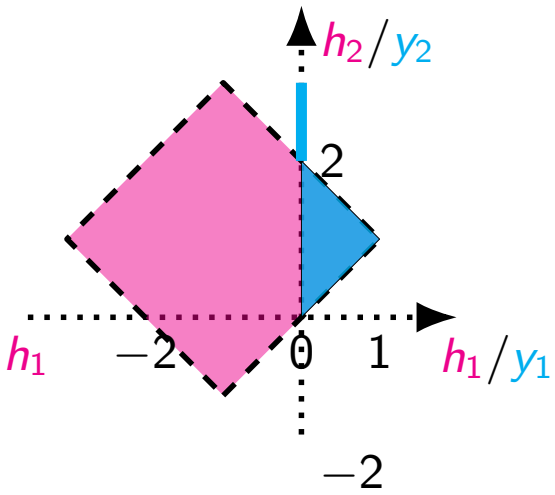
(P_1): the input is always classified as belonging to the class identified by neuron y_2, i.e. we always have $y_2 \geq y_1$

(P_2): (robustness property) in the neighborhood [-0.25,0.25] of 0 for x_1, whatever x_2 in [-1,1], the output y_1 is never above threshold 0.5

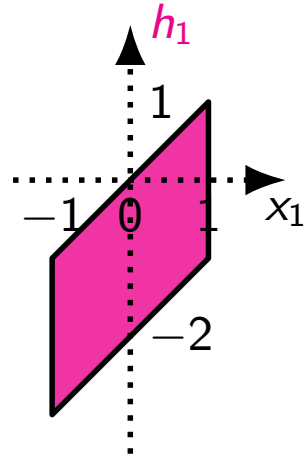
Exact ranges



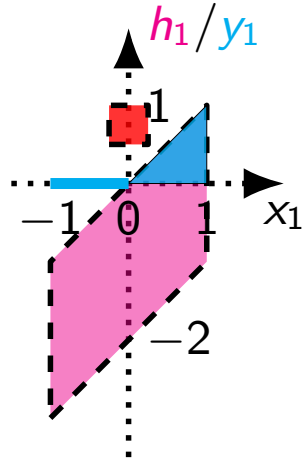
(h_1, h_2)



(h_1, h_2); (y_1, y_2)



(x_1, h_1)

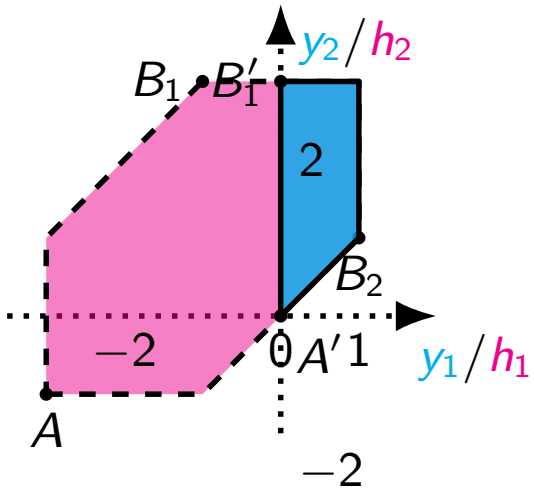


(x_1, h_1); (x_1, y_1)

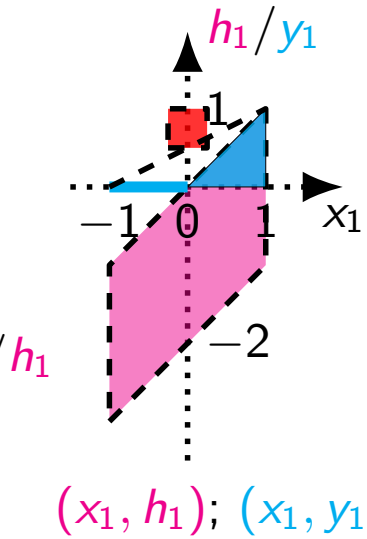
(P_2) is the complement of the red square

Abstractions

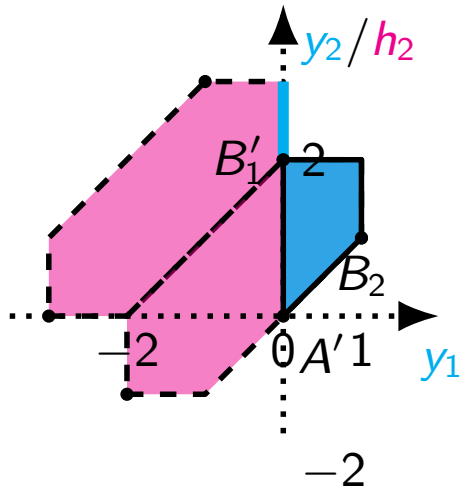
Abstractions



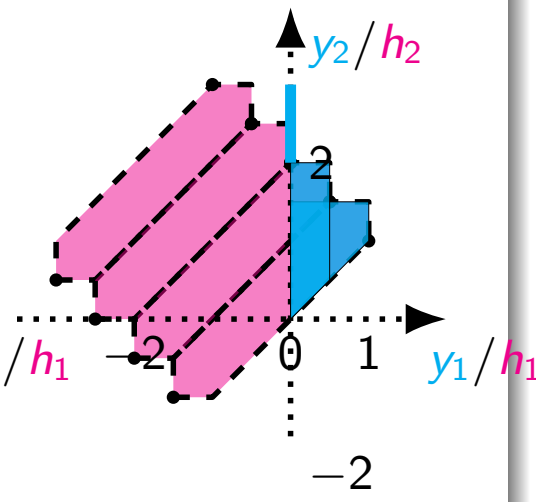
With zone, non-subdivided tropical polyhedra



(x_1, h_1); (x_1, y_1)
Dashed lines enclose the classical convexification



Once subdivided tropical polyhedra



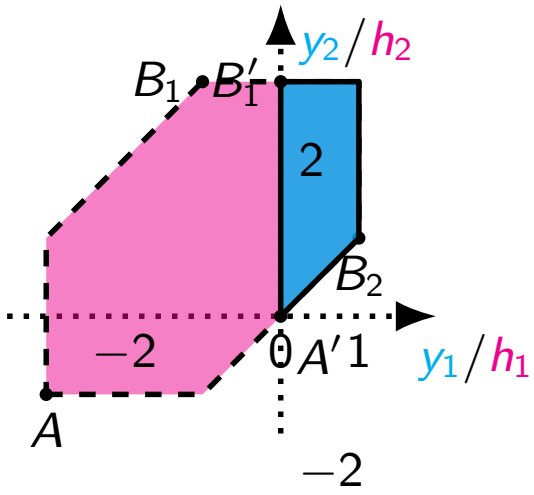
Twice subdivided tropical polyhedra

Abstraction not exact

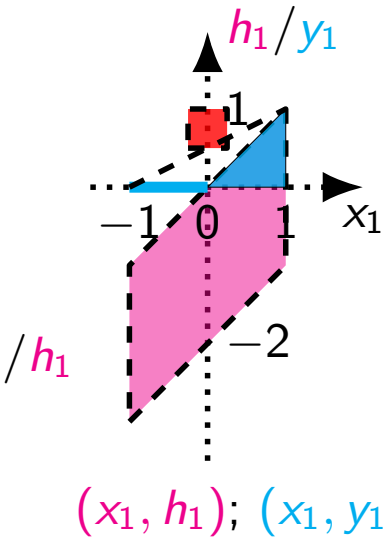
- However, cyan region above diagonal, proving (P_1) - also (P_2), second figure
- Abstraction has an area 2.5 times larger than the exact range, (tropical linearisation)

Abstractions

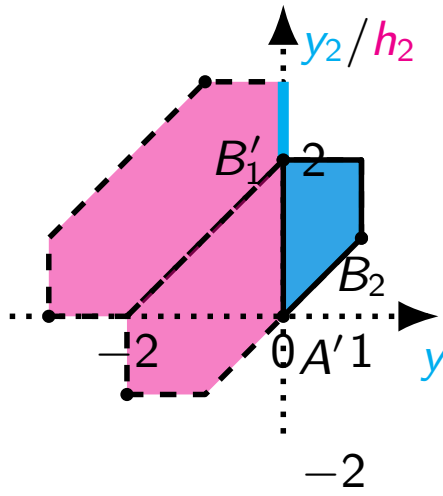
Abstractions



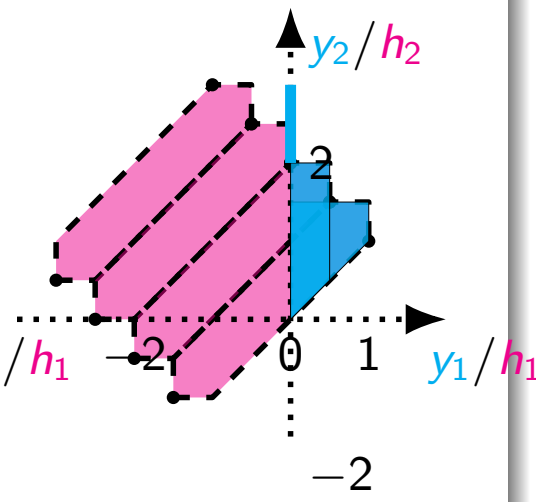
With zone, non-subdivided tropical polyhedra



(x_1, h_1); (x_1, y_1)
Dashed lines enclose the classical convexification



Once subdivided tropical polyhedra



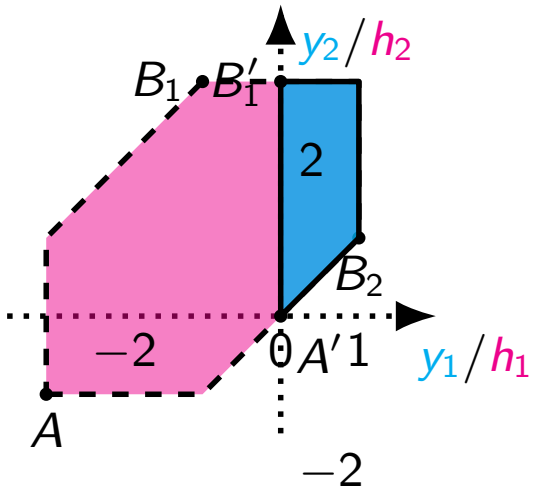
Twice subdivided tropical polyhedra

Abstraction not exact

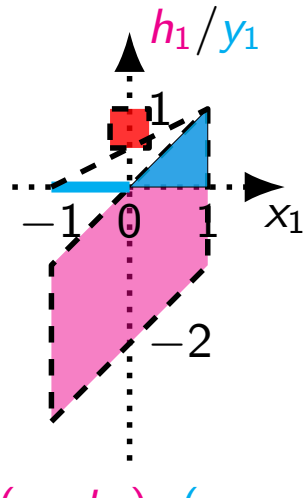
- Subdividing x_2 in 2 ($[-1, 0]$, $[0, 1]$): 1.5 times the exact area
- Subdividing further: area 1.25 times the exact one (general case: $1 + \frac{1}{k}$ for $k \geq 2$)

Abstractions

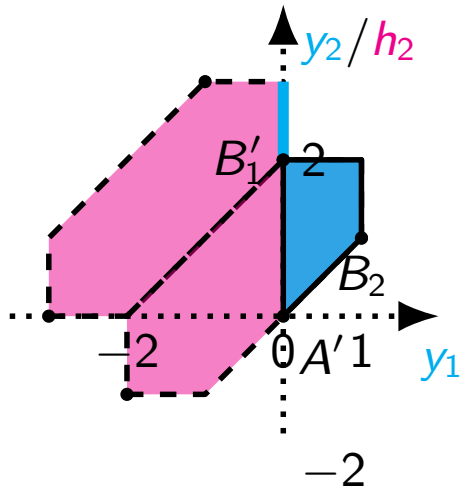
Abstractions



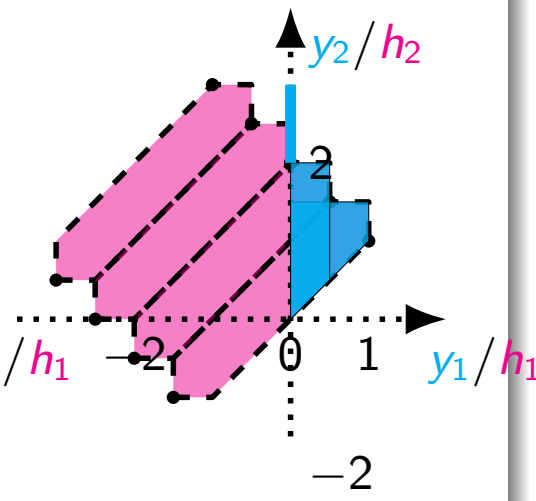
With zone, non-subdivided tropical polyhedra



(x_1, h_1); (x_1, y_1)
Dashed lines enclose the classical convexification



Once subdivided tropical polyhedra



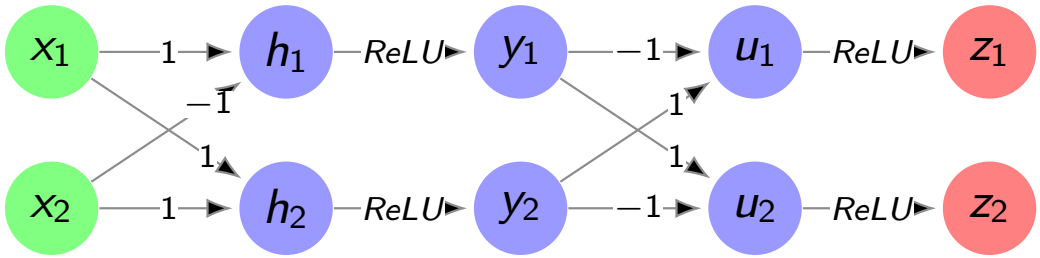
Twice subdivided tropical polyhedra

Abstraction not exact

- Tropical polyhedra are unions of zones: 1, 2 and 3 zone(s) resp. (left to right)
- Last one generated by 5 extreme points

Multi-layer abstraction, an example (more in the paper!)

Input layer | Linear layer | Hidden ReLU layer | Linear layer 2 | Output ReLU layer 2



(biases are $b_1 = -1$, $b_2 = 1$, $c_1 = -1$ and $c_2 = 1$)

New linear layer:

$$u_1 = y_2 - y_1 - 1$$
$$u_2 = y_1 - y_2 + 1$$

Output neurons:

$$z_1 = max(0, u_1)$$
$$= max(0, y_2 - y_1 - 1)$$
$$z_2 = max(0, u_2)$$
$$= max(0, y_1 - y_2 + 1)$$

- Enclosing cube for first layer L_1 (y_1, y_2): $[0, 1] \times [0, 3]$
- Intersection of values obtained on 2nd layer with embedding for 1st layer: refined bounds

$$-2 \leq u_1 - y_1 \leq 2$$
$$-5 \leq u_2 - y_2 \leq 1$$

Checking properties on ReLU neural nets

Affine specifications $h(x,y) = \sum_{i=1}^{m} h_i x_i + \sum_{j=1}^{n} h'_j y_j + c$

- where x_i, resp. y_j are the input, resp. output neurons,
- for all input values in $[-1,1]$, do we have $h(x) \geq 0$?

Method 1

- Derive the smallest zone Z containing $\mathcal{H}$, tropical polyhedron abstracting the input output relation of the neural net
- Solve linear programming problem $m = \min_{x,y \in Z} h(x,y)$

Enough for checking (P_1) in our example but not (P_2).

Method 2

- Solve $m = \min_{x,y \in \mathcal{H}} h(x,y)$ (not convex) optimisation problem, in any sense (tropical nor classical). This could be encoded as MILP (e.g. Gurobi, future work)

Will prove (P_2).

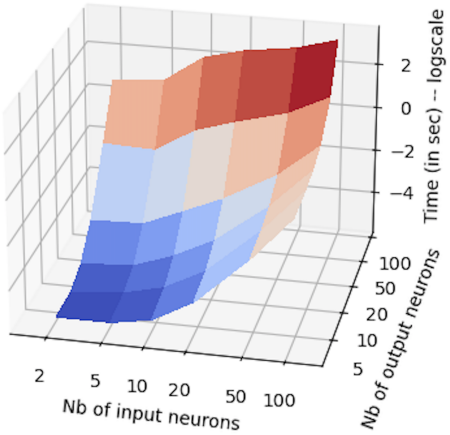
Execution times (internal and double description) on sample networks.

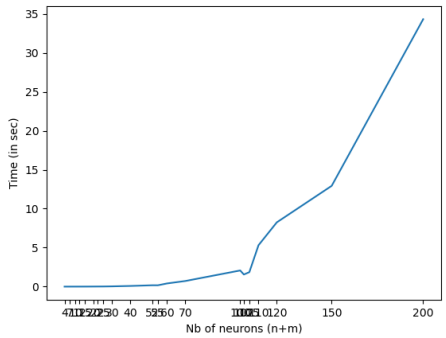
Example	# inp.	# out.	# hid.	# neur.	intern. (s)	double (s)
`running`	2	2	0	4	0.006	1.83
`running2`	2	2	1	6	0.011	4.34
`multi`	2	8	1	13	0.005	3.9
`krelu`	2	2	0	4	0.011	1.94
tora_controller	4	1	1	6	0.005	14.57
tora_controller_1	4	1	1	105	0.75	815.12
quadcopter_control_3	18	1	1	49	0.009	102.54
quadcopter_control_1	18	1	1	69	0.2	469.77
quad_controller	18	1	1	20	0.005	14
car_nn_controller_2	4	2	1	506	104.75	–
car_nn_controller_1	4	2	1	506	88.8	–
ex	2	1	5	59	0.195	1682.28

(examples from Sherlock, mostly)

Conclusion

- First step towards specific approximations of tropical rational functions (=ReLU networks), by linearisation. Enough for getting good precision, further work using e.g. tropical polynomials
- First experiments paves the way to help tackle the curse of dimensionality using extra ingredients (e.g. external or internal representation only)





x-axis is the total number of neurons, y-axis is time.

  - various number of inputs and ouputs for neural nets with one hidden layer only.
  - linear layers are generated randomly, with weights between -2 and 2.

  For 100 inputs and 100 neurons in the hidden layer, the full pipeline (checking the linear specification in particular) took about 35 seconds, among which the tropical polyhedron analysis took 6 seconds.
- Many open problems
  - Efficient double representation algorithm?
  - Possibilities and limits of the tropical world for non-convex and disjunctive analysis