INF641. Introduction to the Verification of Neural Networks

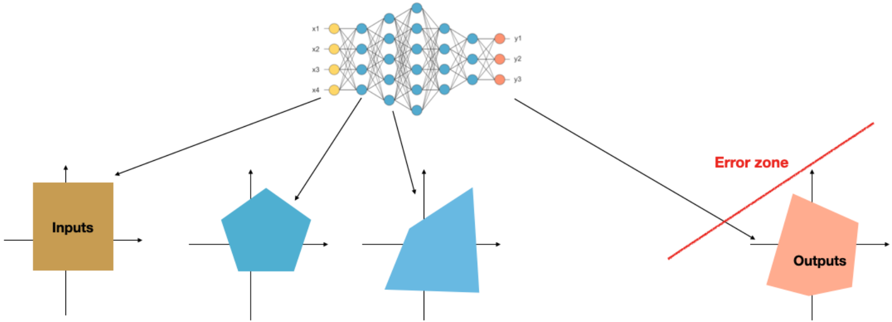
Lecture 2. Abstraction-based verification II

Eric Goubault and Sylvie Putot

Remember last week: Reachability Analysis for Neural Networks

Need efficient and accurate abstraction and transformers for propagation in networks:

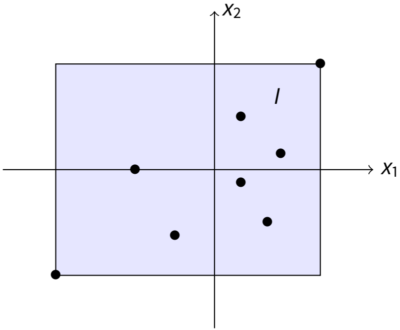
- Affine transformers
- Nonlinear activation functions
- Fixed-point computations/convergence for recurrent networks (skipped here)



Numerical abstract domains

We have seen:

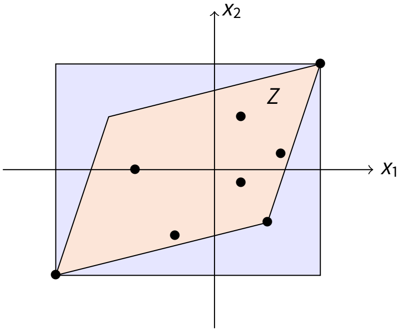
- Intervals/Boxes/Hyperrectangles (synonymous)



Numerical abstract domains

We have seen:

- Intervals/Boxes/Hyperrectangles (synonymous)
- Zonotopes

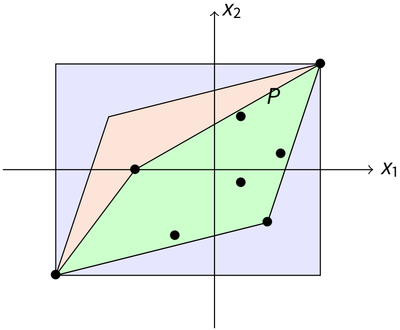


Numerical abstract domains

We have seen:

- Intervals/Boxes/Hyperrectangles (synonymous)
- Zonotopes

Now let us see Convex Polyhedra.



Outline for today

1. Deterministic abstractions for neural network analysis
   - Boxes, Zonotopes, Polyhedra
   - A word on other abstractions (see also Lecture 3 for non-convex abstractions)
   - Finish Lab Session 1
2. Probabilistic verification
   - Sets of Probabilities: P-boxes and Dempster-Shafer structures
   - Arithmetic on P-boxes and Probabilistic Affine forms
   - Lab Session 2

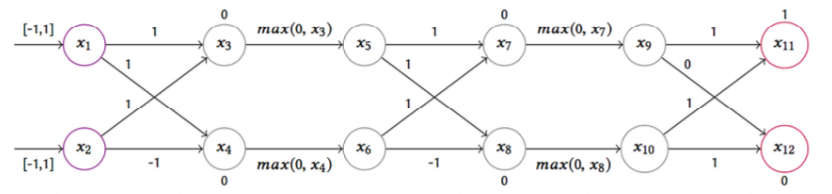
Convex Polyhedra abstractions

The problem is similar to what we have already seen:

Example

Proving specifications such as

- Two inputs: $x_1 \in [-1, 1]$ and $x_2 \in [-1, 1]$, two outputs x_{11} and x_{12}
- Specification: $\forall x_1, x_2 \in [-1, 1]$, we always have $x_{11} \geq x_{12}$ (classification problem)



The Convex Polyhedra abstraction ([Cousot& Halbwachs 1979])

Abstraction by Polyhedra P for Program Analysis usually rely on a double description:

- Constraint representation : an intersection of a finite number of closed half spaces of the form $a^T x \leq \beta$ and a finite number of subspaces of the form $d^T x = \xi$, i.e.

$$P = \{x \in \mathbb{R}^n | Ax \leq b \text{ and } Dx = e\}$$

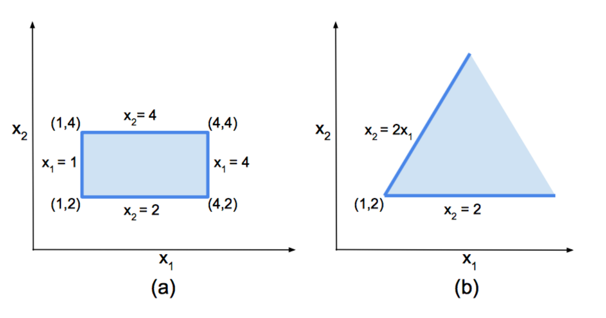
- Generator representation : a convex hull of a finite set of vertices v_i, a finite set of rays r_j and a finite set of lines z_k, i.e. $x \in P$ iff :

$$x = \sum_{i=1}^{u} \lambda_i v_i + \sum_{j=1}^{v} \mu_i r_i + \sum_{i=1}^{w} \nu_i z_i$$

  where $\lambda_i, \mu_i \geq 0$ and $\sum_{i=1}^{u} \lambda_i = 1$.

Chernikova's algorithm is used to convert between the above representations (but this has worst case exponential complexity!)

Example of the double description



In equations

- Left : $C = \{-x_1 \leq -1, x_1 \leq 4, -x_2 \leq -2, x_2 \leq 4\}$ or $G = \{V = \{(1,2),(1,4),(4,2),(4,4)\}, R = \emptyset, Z = \emptyset\}$
- Right : $C = \{-x_2 \leq -2, x_2 \leq 2x_1\}$ or $G = \{V = \{(1,2)\}, R = \{(1,2),(1,0)\}, Z = \emptyset\}$.

Abstract operators

Order-theoretic operations

- Join : $P \cup Q$ is the convex hull of P and Q (easy with the vertex representation)
- Meet : $P \cap Q$ is obtained using the constraint representation, by concatenating the constraints of P and Q
- Inclusion : $P \subseteq Q$ is implemented using LP (linear programming). For each constraint $\sum a_i x_i \leq b$ in P, compute $\mu = max \sum a_i x_i$ subject to constraints of Q : if $\mu > b$ the inclusion does not hold

Arithmetic operations

- Linear assignments $x = L$: add a new variable x to P and the constraint $x - L = 0$ (then use Chernikova for getting the vertex set representation)
- Non linear assignments : generally by linearization

The DeepPoly convex relaxation

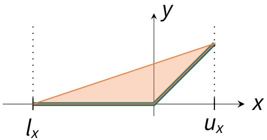
Ref. An Abstract Domain for Certifying Neural Networks, G. SIngh, T. Gehr, M. Puschel, M. Vechev, in POPL 2019

- A restriction of Polyhedra to ensure scalability (bounds the number of constraints to $2n$ where n is the number of variables)
- Affine transforms are exact (and easy)
- Custom convex relaxations for activation functions
- Generally more accurate but more costly than Zonotopes

For each node (or variable) x_i:

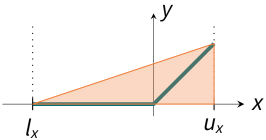
- upper and lower bounds: $x_i \leq u_i$ and $\boldsymbol{x}_i \geq l_i$
- two polyhedral constraints $x_i \leq \sum_j u_{ij} x_j + u_{i0}$ and $x_i \geq \sum_j l_{ij} x_j + l_{i0}$ where the x_j only refer to "previous" variables in the network.

Abstract Transformers: ReLU activation

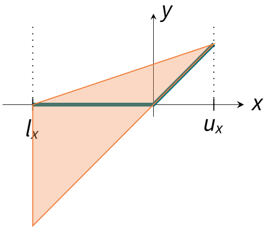


Optimal

Convex transformer (triangle abstraction)



DeepPoly transformer 1



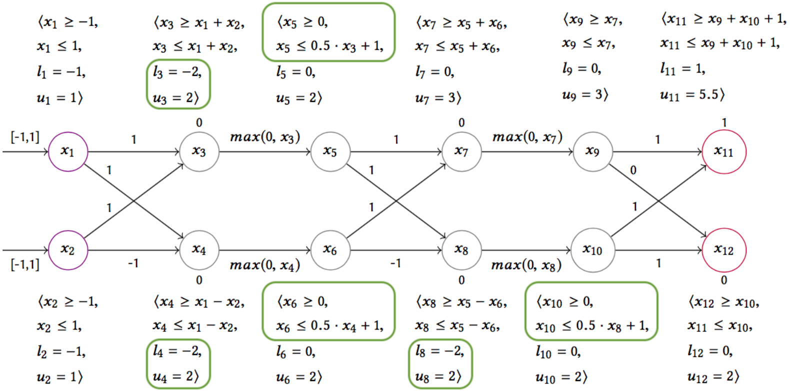
DeepPoly transformer 2

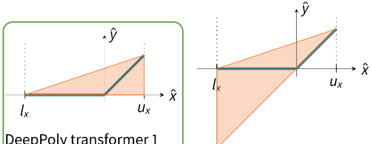
- Upper constraint

$$y \leq \lambda x + \mu,$$

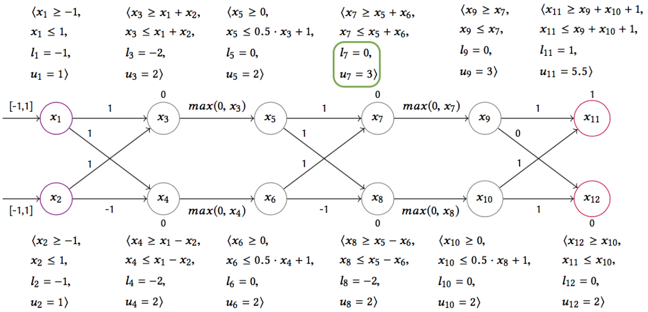
  with $\lambda = \frac{u_x}{(u_x - l_x)}$ and $\mu = \frac{-l_x u_x}{(u_x - l_x)}$.
- Optimal (triangle) transformer contains two lower polyhedral constraints for y, which is not allowed by the restricted domain
- Choice between RELU transformers 1 or 2 depends on area (heuristic): both are smaller area-wise than the Zonotope transformer

Analysis by DeepPoly on the example: ReLU transformer





Analysis by DeepPoly on the example: affine transformers

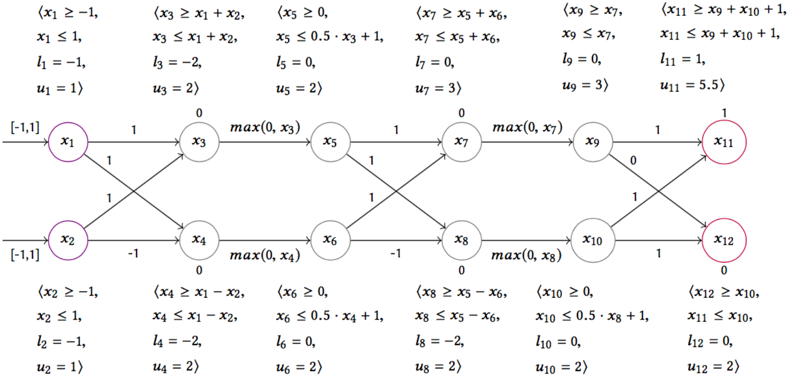


Precise bounds (useful for ReLU) by backsubstitution on the polyhedral constraints:

$$
\begin{array}{ccccc}
x_5 + x_6 & \leq & x_7 & \leq & x_5 + x_6 \\
0 & \leq & x_7 & \leq & 0.5x_3 + 0.5x_4 + 2 \\
0 & \leq & x_7 & \leq & 0.5(x_1 + x_2) + 0.5(x_1 - x_2) + 2 \\
0 & \leq & x_7 & \leq & x_1 + 2 \leq 3
\end{array}
$$

Checking the specification

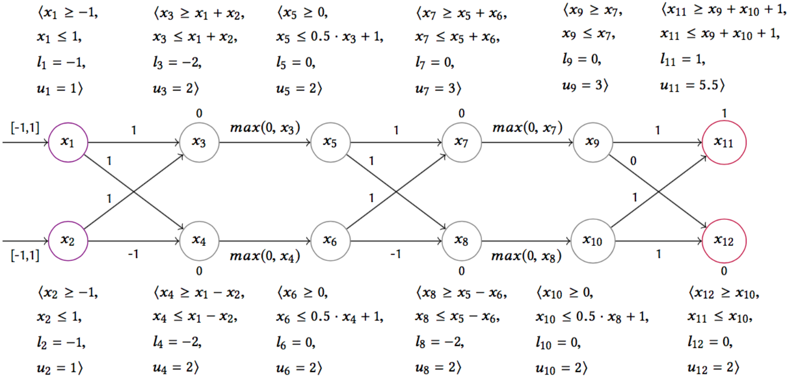
Check whether $\forall i_1, i_2 \in [-1,1] \times [-1,1], x_{11} \geq x_{12}$ (robustness of classification) ?



Checking the specification

Check whether $\forall i_1, i_2 \in [-1,1] \times [-1,1], x_{11} \geq x_{12}$ (robustness of classification) ?

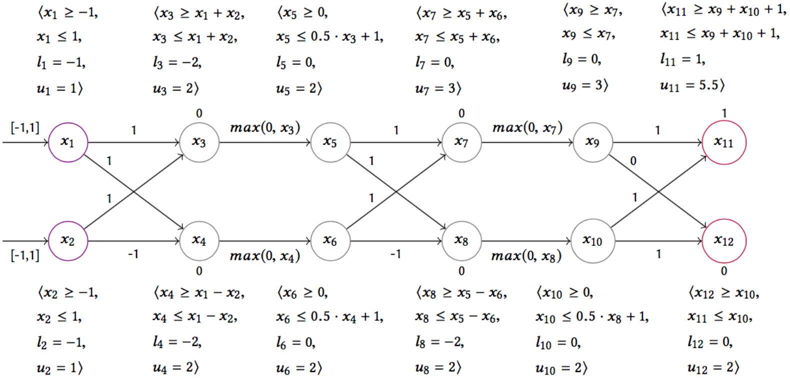
- bounds on x_{11} and x_{12} ?



Checking the specification

Check whether $\forall i_1, i_2 \in [-1,1] \times [-1,1], x_{11} \geq x_{12}$ (robustness of classification) ?

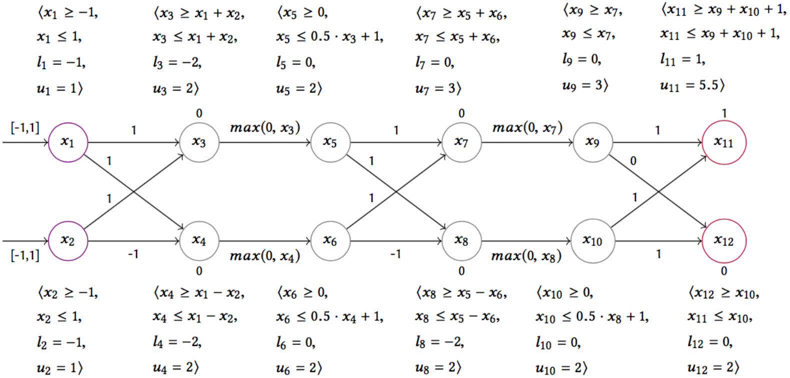
- bounds on x_{11} and x_{12} ? inconclusive: $x_{11} \geq 1 \wedge x_{12} \leq 2 \implies x_{11} - x_{12} \geq -1$



Checking the specification

Check whether $\forall i_1, i_2 \in [-1,1] \times [-1,1], x_{11} \geq x_{12}$ (robustness of classification) ?

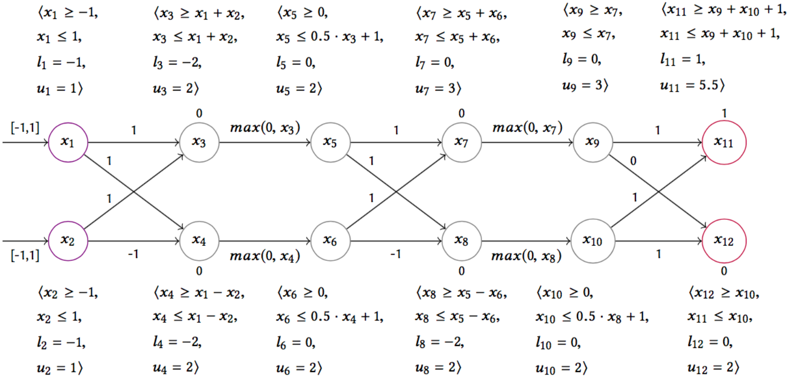
- bounds on x_{11} and x_{12} ? inconclusive: $x_{11} \geq 1 \wedge x_{12} \leq 2 \implies x_{11} - x_{12} \geq -1$
- backsubstitution on $x_{11} - x_{12}$, possibly up to 1st layer:



Checking the specification

Check whether $\forall i_1, i_2 \in [-1,1] \times [-1,1], x_{11} \geq x_{12}$ (robustness of classification) ?

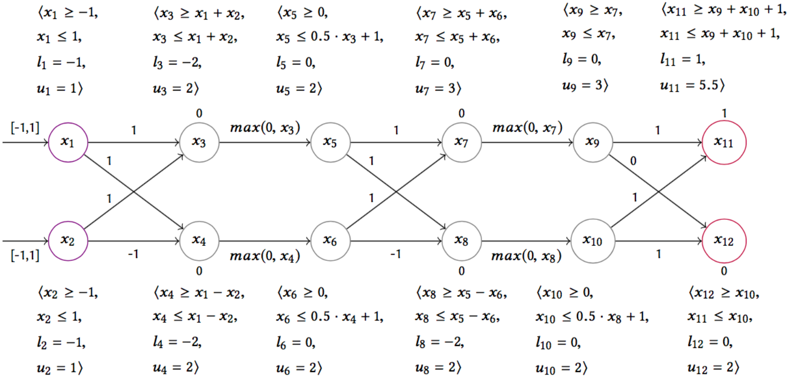
- bounds on x_{11} and x_{12} ? inconclusive: $x_{11} \geq 1 \wedge x_{12} \leq 2 \implies x_{11} - x_{12} \geq -1$
- backsubstitution on $x_{11} - x_{12}$, possibly up to 1st layer: $x_{11} - x_{12} \geq x_9 + x_{10} + 1 - x_{10} \geq l_9 + 1 = 1$. Proved



Checking the specification

Check whether $\forall i_1, i_2 \in [-1,1] \times [-1,1], x_{11} \geq x_{12}$ (robustness of classification) ?

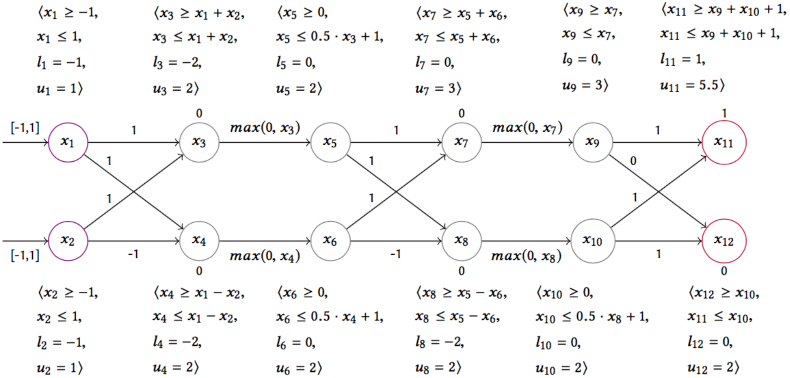
- bounds on x_{11} and x_{12} ? inconclusive: $x_{11} \geq 1 \wedge x_{12} \leq 2 \implies x_{11} - x_{12} \geq -1$
- backsubstitution on $x_{11} - x_{12}$, possibly up to 1st layer: $x_{11} - x_{12} \geq x_9 + x_{10} + 1 - x_{10} \geq l_9 + 1 = 1$. Proved
- What if inconclusive ?



Checking the specification

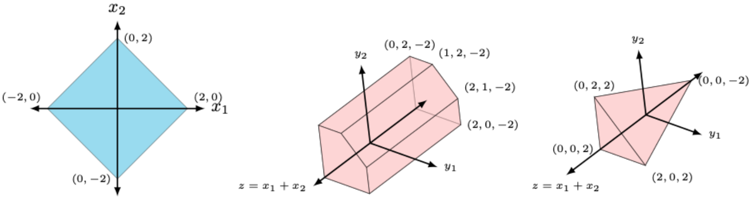
Check whether $\forall i_1, i_2 \in [-1,1] \times [-1,1], x_{11} \geq x_{12}$ (robustness of classification) ?

- bounds on x_{11} and x_{12} ? inconclusive: $x_{11} \geq 1 \wedge x_{12} \leq 2 \implies x_{11} - x_{12} \geq -1$
- backsubstitution on $x_{11} - x_{12}$, possibly up to 1st layer: $x_{11} - x_{12} \geq x_9 + x_{10} + 1 - x_{10} \geq l_9 + 1 = 1$. Proved
- What if inconclusive ? Try prove the contrary: falsification



To go further: refining existing abstractions

- RefineZono: combined zonotope abstraction and MILP encoding: *Boosting Robustness Certification of Neural Networks, ICLR 2019, G Singh, T Gehr, M Püschel, M Vechev*
  - mixing abstraction and optimization (see next week for optimization based approaches)
- Star sets: *Star-based reachability analysis of deep neural networks, Tran et al., FM 2019*.
  - extension of Zonotopes, can be as expressive as Polyhedra
- k-Relu: how to abstract jointly multiple Relus *Beyond the single neuron convex barrier for neural network certification, NeurIPS 2019, G Singh, R Ganvir, M Pschel, and M Vechev.*

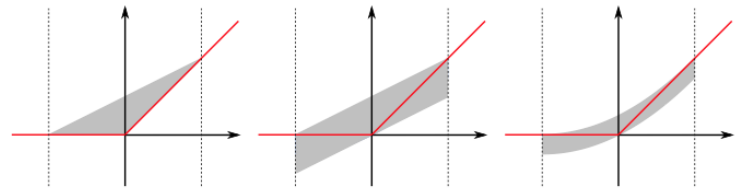


(a) Input shape (b) 1-ReLU (c) 2-ReLU

To go further: other abstractions

Non-convex abstractions

- Polynomial Zonotopes *Open- and Closed-Loop Neural Network Verification using Polynomial Zonotopes, N. Kochdumper, C. Schilling, M. Althoff, and S. Bak, 2022*



- Max-plus (or Tropical) Polyhedra: *Static analysis of ReLU neural networks with tropical polyhedra, E. Goubault, S. Palumby, S. Putot, L. Rustenholz and S. Sankaranarayanan, SAS 2021*
  - does not rely on convexification of ReLU: ReLU is a tropical polyhedron (see next week)

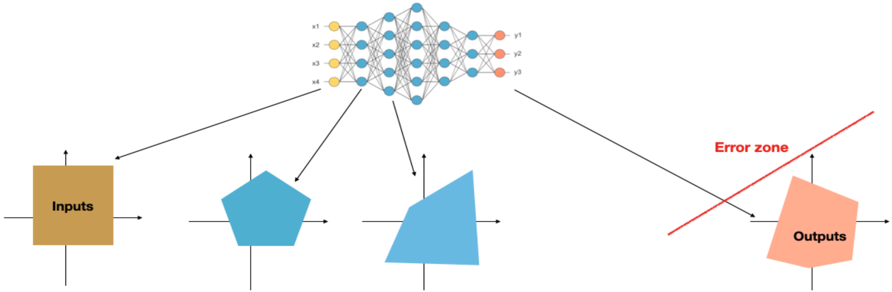
Outline for today

1. Deterministic abstractions for neural network analysis
   - Boxes, Zonotopes, Polyhedra
   - A word on other abstractions (see also Lecture 3 for non-convex abstractions)
   - Finish Lab Session 1
2. Probabilistic verification
   - Sets of Probabilities: P-boxes and Dempster-Shafer structures
   - Arithmetic on P-boxes and Probabilistic Affine forms
   - Lab Session 2

Reachability Analysis for Neural Network Verification

Robustness and input/output properties:

- Need to be proved for (possibly large) sets of network inputs
- Can be specified as preconditions/postconditions expressed in linear arithmetic



Qualitative verification: property proven true or unknown

Quantitative Neural Network Verification

Motivation

- Provide additional information on property satisfaction compared to SAT/UNKNOWN Often need quantitative, probabilistic guarantees on safety, security, reliability, performance, resource usage, etc, for instance
  - transportation: probability of a failure in a time interval should be less than 0.00001
  - neural network robustness: requiring no adversarial examples may be too strict, want high probability that local perturbations result in same classification result
- Exploit knowledge of probabilistic information on inputs
  - can be probabilistic but imprecisely known, e.g.:
    - Gaussian variable $\mathcal{N}(\mu, \sigma^2)$ with uncertain mean $\mu \in [\underline{\mu}, \overline{\mu}]$ and variance $\sigma^2 \in [\underline{\sigma^2}, \overline{\sigma^2}]$
    - Uniform variable $\mathcal{U}(a, b)$ with uncertain range (a and b uncertain)
  - example: noise due to sensor $V + \varepsilon$ with $V \in [a, b]$, ε a random variable

Problem Statement: propagating imprecise probabilities

Problem (Probability bounds analysis)

Given a ReLU network f and a constrained probabilistic input set

$$\mathcal{X} = \{X \in \mathbb{R}^{h_0} \mid CX \leq d \wedge \underline{F}(x) \leq \boldsymbol{P}(X \leq x) \leq \overline{F}(x), \forall x\}$$

where $\underline{F}$ and $\overline{F}$ are two cumulative distribution functions, compute a constrained probabilistic output set $\mathcal{Y}$ guaranteed to contain $\{f(X), X \in \mathcal{X}\}$.
For $X \in \mathbb{R}^n$, we note $\boldsymbol{P}(X \leq x) := \boldsymbol{P}(X_1 \leq x_1 \wedge X_2 \leq x_2 \ldots \wedge X_n \leq x_n)$

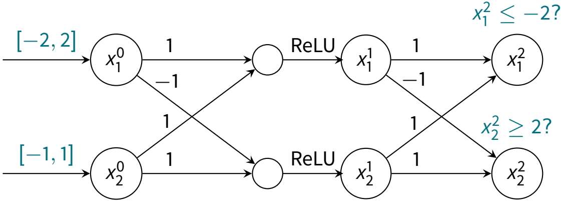
Problem (Quantitative property verification)

Given a ReLU network f, a constrained probabilistic input set $\mathcal{X}$ and a linear safety property $Hy \leq w$, bound the probability of the network output vector y satisfying this property.

Toy illustrating example: 2-layers ReLU network

$$A_1 = A_2 = \begin{bmatrix} 1 & -1 \\ 1 & 1. \end{bmatrix}, b_1 = b_2 = \begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}.$$

$$\begin{aligned} x^1 &= \sigma(A_1 x^0 + b_1) = \sigma(x_1^0 - x_2^0, x_1^0 + x_2^0) \\ x^2 &= A_2 x^1 + b_2 \end{aligned}$$



Property:

- Qualitative: if $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$, does output satisfy $x_1^2 \leq -2 \wedge x_2^2 \geq 2$?
- Quantitative:
  - $\boldsymbol{P}\left(x_1^2 \leq -2 \wedge x_2^2 \geq 2 \;\middle|\; x_1^0 \in \mathcal{U}(-2, 2) \wedge x_2^0 \in \mathcal{U}(-1, 1)\right)$?
  - $\boldsymbol{P}\left(x_1^2 \leq -2 \wedge x_2^2 \geq 2 \;\middle|\; x_1^0 \in \mathcal{N}(0, [0.5, 0.66]) \wedge x_2^0 \in \mathcal{N}([0, 1], 0.33)\right)$?

Outline

- Imprecise probabilities: P-boxes and Dempster-Shafer Interval Structures (DSI)
  - Representations of sets of probability distributions
  - Generalize both probabilistic and non deterministic (interval) computations
- ReLU neural network analysis by DSI
- Mitigating the wrapping effect of intervals using zonotopes
  - Probabilistic Zonotopes
  - Zonotopic Dempster-Shafer Structures (DSZ)
- Evaluation

Representation of imprecise probabilities: P-box

Definition (P-box for a real-valued random variable X)

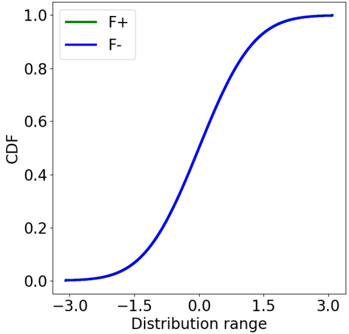
Given two (lower and upper) CDF (Cumulative Distribution Functions) $\underline{F}$ and $\overline{F}$ from $\mathbb{R}$ to $\mathbb{R}^+$ s.t. $\forall x \in \mathbb{R}, \underline{F}(x) \leq \overline{F}(x)$, the p-box $[\underline{F}, \overline{F}]$ represents the set of probability distributions for X s.t.

$$\forall x \in \mathbb{R}, \underline{F}(x) \leq \boldsymbol{P}(X \leq x) \leq \overline{F}(x).$$

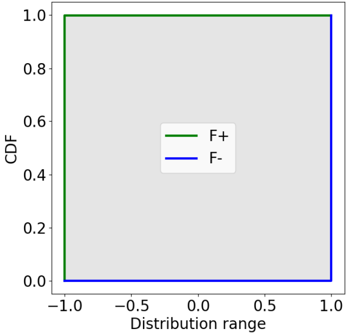
- Ferson S, Kreinovich V, Ginzburg L, Myers D, Sentz K, Constructing probability boxes and Dempster–Shafer structures. Tech. Rep. SAND2002-4015, 2003
- Williamson and Downs, Probabilistic Arithmetic I: Numerical Methods for Calculating Convolutions and Dependency Bounds, Journal of Approximate Reasoning, 1990

P-box examples (Julia library ProbabilityBoundsAnalysis.jl)l

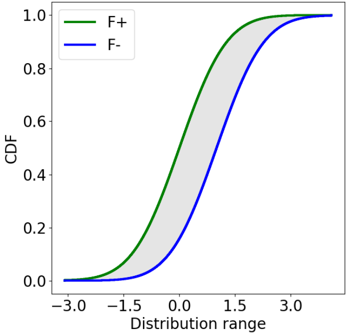
Sets of probability distributions on X (CDF form) such that $\forall x, F^{-}(x) \leq \boldsymbol{P}(X \leq x) \leq F^{+}(x)$:



normal(0,1)



makepbox(interval(-1,1))



normal(interval(0,1),1)

Generalize probabilistic and non deterministic (interval) information

Dempster-Shafer Interval structures (DSI)

A discrete version of P-boxes:

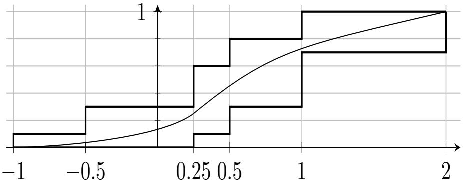
- Focal elements $t \in T$ (sets of values, here Intervals) with probability $w : T \to \mathbb{R}^+$

$t \in T$	[-1,0.25]	[-0.5,0.5]	[0.25,1]	[0.5,1]	[0.5,2]	[1,2]
$w(t)$	0.1	0.2	0.3	0.1	0.1	0.2

- Represents the set of probability distributions P on X such that:

$$\forall x \in [-1, -0.5],\ P(X \leq x) \leq 0.1\,,$$
$$\forall x \in [-0.5, 0.25],\ P(X \leq x) \leq 0.1 + 0.2\,,$$
$$\forall x \in [0.25, 0.5],\ 0.1 \leq P(X \leq x) \leq 0.1 + 0.2 + 0.3\,,$$

etc.



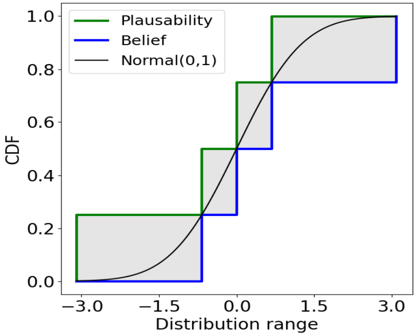
- They define Belief function Bel and Plausibility function Pl from $\wp(E)$ to $\mathbb{R}$:

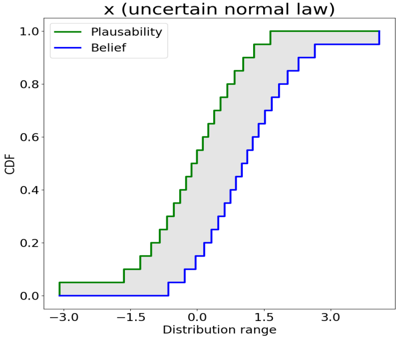
$$Bel(S) = \sum_{t \in T, t \subseteq S} w(t) \leq P(S) \leq \sum_{t \in T, t \cap S \neq \emptyset} w(t) = Pl(S)$$

From P-boxes to Dempster-Shafer Interval structures

Given a P-box $(\underline{F}, \overline{F})$

- Take lower and upper approximation by stair functions
- Deduce focal elements (intervals) and weights

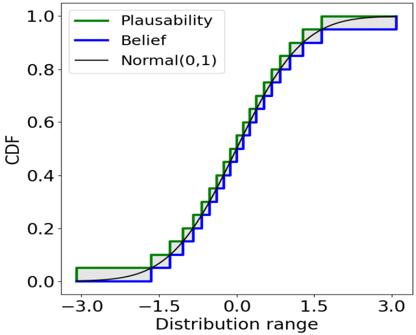


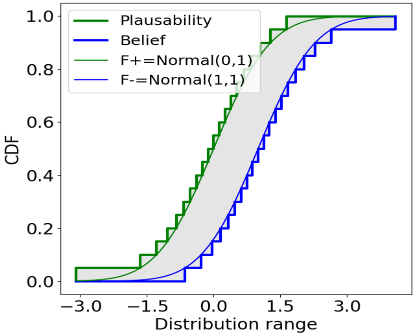


From P-boxes to Dempster-Shafer Interval structures

Given a P-box $(\underline{F}, \overline{F})$

- Take lower and upper approximation by stair functions
- Deduce focal elements (intervals) and weights

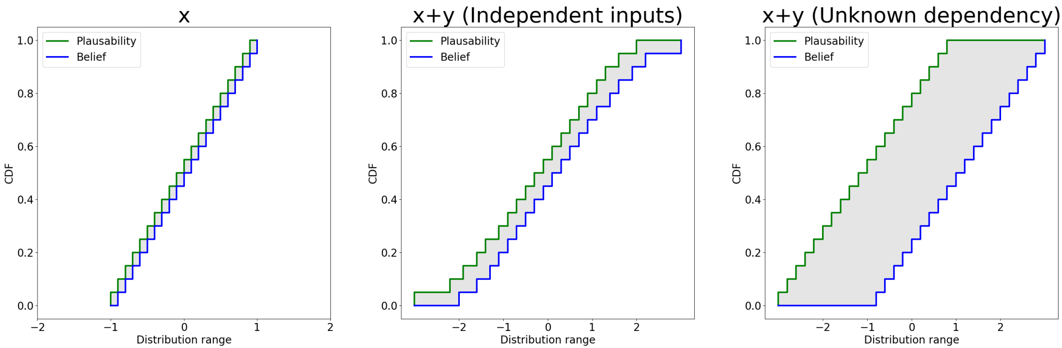




Arithmetic on DSI structures

DSI structures can be propagated through arithmetic operations:

- 2 cases: independent inputs / unknown dependency
- relying on interval arithmetic / Frechet inequalities
- conservative approximations



Arithmetic on DS structures: $z = x\Box y$ ($\Box$=+,-,$\times$,/ etc.)

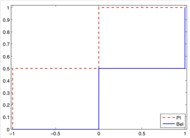
Independent variables x, y

- x (resp. y) given by focal elements T^x (resp. T^y) and weights w^x (resp. w^y)
- $T^z = \{t^x \Box t^y \mid t^x \in T^X, t^y \in T^y\}$ and $w^z(t^x \Box t^y) = w^x(t^x)w^y(t^y)$ (and renormalize)

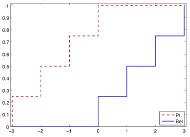
Example

- $T^x = \{[-1,0],[0,1]\}, w^x([-1,0]) = w^x([0,1]) = \frac{1}{2}$ (approximation of uniform distribution on [-1,1])
- $T^y = \{[-2,0],[0,2]\}, w^y([-2,0]) = w^y([0,2]) = \frac{1}{2}$

$x; y$	[-2,0], $\frac{1}{2}$	[0,2], $\frac{1}{2}$
[-1,0], $\frac{1}{2}$	[-3,0], $\frac{1}{4}$	[-1,2], $\frac{1}{4}$
[0,1], $\frac{1}{2}$	[-2,1], $\frac{1}{4}$	[0,3], $\frac{1}{4}$



CDF of x



CDF of $x + y$

Arithmetic on DSS for unknown dependencies (here $\square = +$)

- DS for x (similarly for y) given on $T^x = \{[a_i^x, b_i^x] \mid i = 1, \ldots, n\}$ by $w^x([a_i^x, b_i^x]) = w_i^x$
- Compute P-boxes for $z = x + y$ by LP using Frechet inequalities

Compute the stair functions given by values at $a_k^x + a_l^y, b_k^x + b_l^y$:

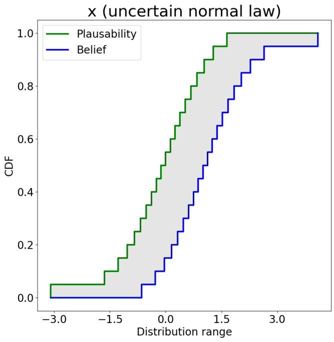
$$\overline{F}_z(a_k^x + a_l^y) = \min\left(\inf_{a_i^x + a_j^y = a_k^x + a_l^y} \sum_{i' \leq i} w_{i'}^x + \sum_{j' \leq j} w_{j'}^y, 1\right)$$

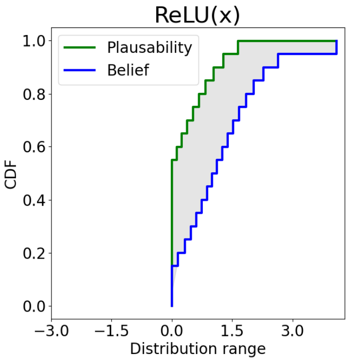
$$\underline{F}_z(b_k^x + b_l^y) = \max\left(\sup_{b_i^x + b_j^y = b_k^x + b_l^y} \sum_{i' \leq i} w_{i'}^x + \sum_{j' \leq j} w_{j'}^y - 1, 0\right)$$

ReLU

Lemma (ReLU of a DSI)

Given X represented by the DSI $\{\langle \boldsymbol{x_i}, w_i \rangle, i \in [1, n]\}$, then the CDF of $Y = \sigma(X) = \max(0, X)$ is included in the DSI $\{\langle \boldsymbol{y_i}, w_i \rangle, i \in [1, n]\}$ with $y_i = [\max(0, \underline{x_i}), \max(0, \overline{x_i})]$.





ReLU neural network analysis by DSI

Input: d^0 a h_0-dimensional vector of DSI

for $k = 0$ to $L - 1$ **do**
  for $l = 1$ to h_{k+1} **do**
    $d_l^{k+1} \leftarrow \sigma(\sum_{j=1}^{h_k} a_{lj}^k d_j^k + b_l^k)$ $\triangleright$ *Affine transform and ReLU - Dependency graph useful for choosing the right DSI operations (indep. or unknown dep.) in affine transforms*
  end for
end for
return *$(d^L$,cdf$(Hd^L, w))$* $\triangleright$ *Vector of DSI for the output layer and probability bounds for property $Hz \leq w$*

Illustration on the toy example

Input $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$ with Uniform law on inputs

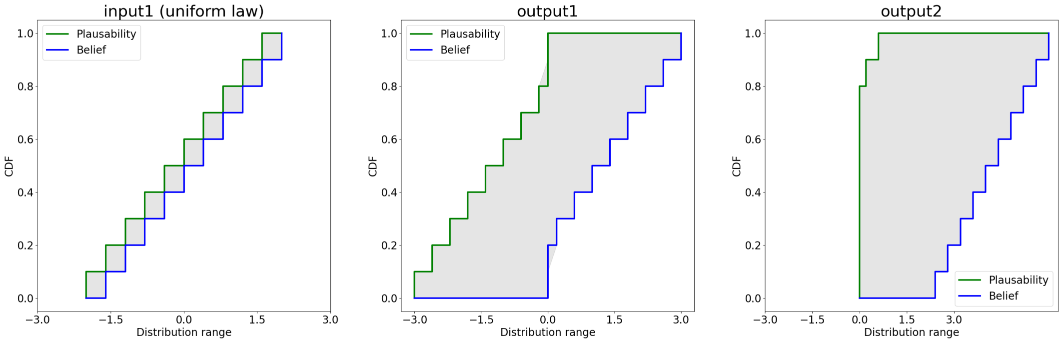
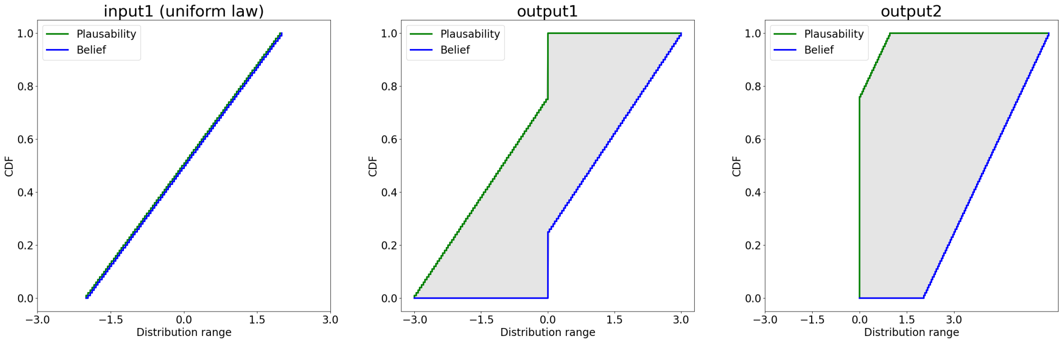


Illustration on the toy example

Input $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$ with Uniform law on inputs



Finer discretization refines the approximation but the ranges are unchanged

Illustration on the toy example

Input $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$ with Normal law on inputs

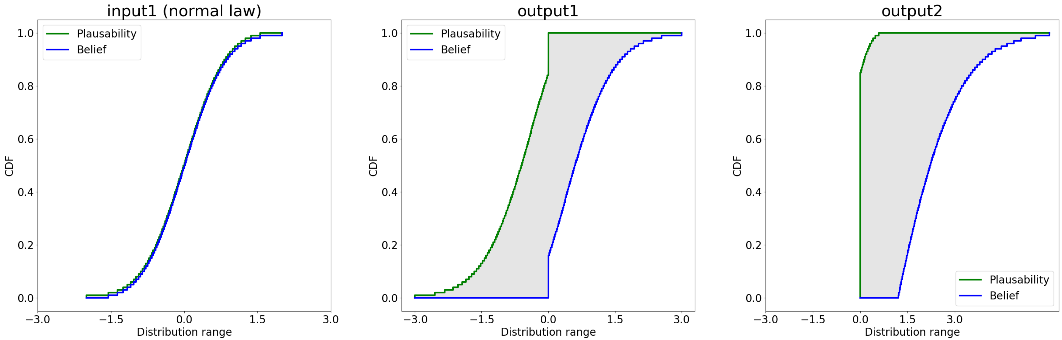
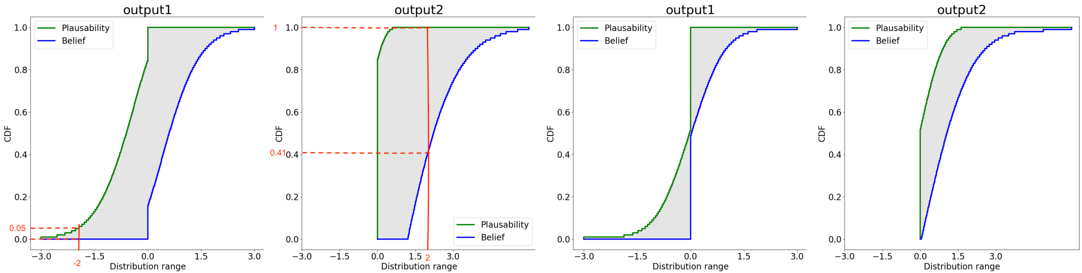


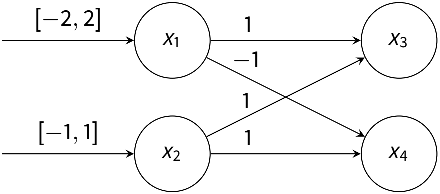
Illustration on the toy example

Unknown dependency on inputs vs independent inputs



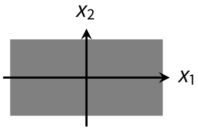
$\boldsymbol{P}(z_1 \leq -2) \in [0, 0.05]$ $\boldsymbol{P}(z_2 \geq 2) \in [0, 0.59]$ $\boldsymbol{P}(z_1 \leq -2) \in [0, 0.01]$ $\boldsymbol{P}(z_2 \geq 2) \in [0, 0.2]$

Wrapping effect: example of the first affine layer



Initial domain:

$$-2 \leq x_1 \leq 2$$
$$-1 \leq x_2 \leq 1$$

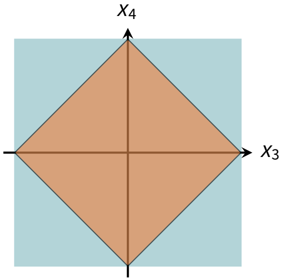


Exact domain:

$$x_3 = x_1 - x_2$$
$$x_4 = x_1 + x_2$$
$$x_1, x_2 \in [-1, 1]$$

Using Intervals/Boxes:

$$-3 \leq x_3 \leq 3$$
$$-3 \leq x_4 \leq 3$$
$$x_1, x_2 \in [-1, 1]$$



The optimal affine transformers for boxes are not exact. Zonotope transformers are !

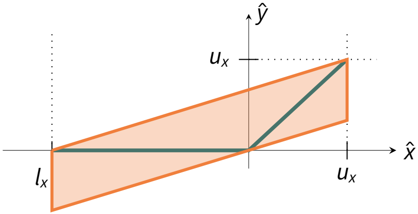
Zonotopes and neural network reachability analysis

Definition (Zonotope)

An n-dimensional zonotope $\mathcal{Z}$ with center $c \in R^n$ and a vector $\Gamma = \left[g_1 \dots g_p\right] \in \mathbb{R}^{n,p}$ of p generators $g_j \in \mathbb{R}^n$ for $j = 1, \dots, p$ is defined as $\mathcal{Z} = \langle c, \Gamma \rangle = \{c + \Gamma\varepsilon \mid \|\varepsilon\|_\infty \leq 1\}$.

Zonotopes are closed under affine transformations: for $A \in \mathbb{R}^{m,n}$ and $b \in \mathbb{R}^m$ we define $A\mathcal{Z} + b = \langle Ac + b, A\Gamma \rangle$ as the m-dimensional resulting zonotope.

RELU transformer: conservative approximation



Two solutions for zonotopic probabilistic NN analysis

Main idea: encode as much deterministic dependencies as possible by affine forms, and avoid/delay Dempster-Shafer arithmetic whenever possible

Probabilistic zonotopes (or probabilistic affine forms)

- Zonotopic network analysis starting from the support of input distribution
- Probabilistic interpretation: noise symbols are DSI instead of intervals

Dempster-Shafer Zonotopic structures (DSZ)

- Dempster-Shafer structures with zonotopic focal elements
- A refinement of probabilistic zonotopes, which fully exploits the DSI input discretization in the NN analysis
- Restricted to independent inputs

NN analysis by DSZ (independent inputs)

Input: d^0 a h_0-dimensional vector of DSI

$d^0_{\mathcal{Z}} = \left\{ \langle \mathcal{Z}^0_{i_1 \dots i_{h_0}}, w^0_{1,i_1} \dots w^0_{h_0,i_{h_0}} \rangle,\ (i_1, \dots, i_{h_0}) \in [1,n]^{h_0} \right\} \leftarrow$ dsi-to-dsz(d^0)
for $k = 0$ to $L-1$ **do**
  for $(i_1, i_2, \dots, i_{h_0}) \in [1,n]^{h_0}$ **do**
    $\mathcal{Z}^{k+1}_{i_1 \dots i_{h_0}} \leftarrow \sigma(A^k \mathcal{Z}^k_{i_1 \dots i_{h_0}} + b^k)$ ▷ Independent zonotopic analyzes (can be done in parallel)
  end for
end for
$d^L_{\mathcal{Z}} = \left\{ \langle \mathcal{Z}^L_{i_1 \dots i_{h_0}}, w^0_{1,i_1} \dots w^0_{h_0,i_{h_0}} \rangle,\ (i_1, \dots, i_{h_0}) \in [1,n]^{h_0} \right\}$
$d^L \leftarrow$ dsz-to-dsi($d^L_{\mathcal{Z}}$)
return (d^L,cdf(($Hd^L_{\mathcal{Z}}, w$))) ▷ Property bounds computed by direct evaluation of the CDF on the zonotopic focal elements

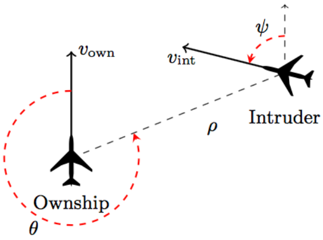
Comparing DSI, Prob. Zonotopes and DSZ: toy example

Table 1: Probability bounds for the toy example, independent inputs.

Law	DSI			Prob. Zono.			DSZ		
(#FE)	$\boldsymbol{P}(x_1^2 \leq -2)$	$\boldsymbol{P}(x_2^2 \geq 2)$	time	$\boldsymbol{P}(x_1^2 \leq -2)$	$\boldsymbol{P}(x_2^2 \geq 2)$	time	$\boldsymbol{P}(x_1^2 \leq -2)$	$\boldsymbol{P}(x_2^2 \geq 2)$	time
$U(2)$	$[0, 0.5]$	$[0, 1]$	$< e^{-3}$	$[0, 0.5]$	$[0, 1]$	$< e^{-3}$	$[0, 0.25]$	$[0, 0.5]$	$< e^{-3}$
$U(10)$	$[0, 0.2]$	$[0, 0.7]$	e^{-3}	$[0, 0.3]$	$[0, 0.8]$	e^{-3}	$[0, 0.03]$	$[0.2, 0.3]$	$< e^{-3}$
$U(10^2)$	$[0, 0.07]$	$[0.05, 0.52]$	0.022	$[0, 0.26]$	$[0, 0.76]$	0.013	$[0, 0.0014]$	$[0.25, 0.26]$	0.026
$U(10^3)$	$[0, 0.063]$	$[0.062, 0.502]$	2.4	$[0, 0.251]$	$[0, 0.751]$	1.2	$[0, 3.e^{-6}]$	$[0.25, 0.251]$	3
$N(10)$	$[0, 0.017]$	$[0, 0.277]$	e^{-3}	$[0, 0.1]$	$[0, 1]$	e^{-3}	$[0, 0.01]$	$[0, 0.1]$	$< e^{-3}$
$N(10^2)$	$[0, 0.004]$	$[0, 0.186]$	0.022	$[0, 0.07]$	$[0, 0.94]$	0.013	$[0, 4.e^{-4}]$	$[0.06, 0.07]$	0.026
$N(10^3)$	$[0, 0.004]$	$[0.003, 0.182]$	2.4	$[0, 0.067]$	$[0, 0.934]$	1.2	$[6e^{-5}, 1.1e^{-4}]$	$[0.066, 0.067]$	3

- For independent inputs, DSZ always more precise.

ACAS Xu: collision avoidance systems for civil aircrafts (FAA)



- Bounded (vector) inputs in [lb,ub], components follow independent Gaussian distributions with $\mu = (ub + lb)/2$ and $\sigma = (ub - m)/3$
- Properties: $P_2 : y_1 > y_2 \wedge y_1 > y_3 \wedge y_1 > y_4 \wedge y_1 > y_5$
  $P_3/P_4 : y_1 < y_2 \wedge y_1 < y_3 \wedge y_1 < y_4 \wedge y_1 < y_5$
- (Manual) Input discretization: [5, 80, 50, 6, 5] for P_2, [5, 20, 1, 6, 5] for P_3 and P_4

Prop	Net	DSZ	
		P	time
2	1-6	[0, 0.01999]	46.4
2	2-2	[0.00423 0.0809]	47.9
2	2-9	[0, 0.0774684]	51.0
2	3-1	[0.0165, 0.08787]	43.8
2	3-6	[0.0167, 0.1111]	52.4
2	3-7	[6e-05, 0.1361]	43.7
2	4-1	[1e-05, 0.05353]	40.9
2	4-7	[0.0129, 0.1056]	44.4
2	5-3	[0, 0.03939]	40.0
3	1-7	[1, 1]	0.25
4	1-9	[1, 1]	0.2