ÉCOLE POLYTECHNIQUE

INSTITUT POLYTECHNIQUE DE PARIS

# A Zonotopic Dempster-Shafer Approach to the Quantitative Verification of Neural Networks

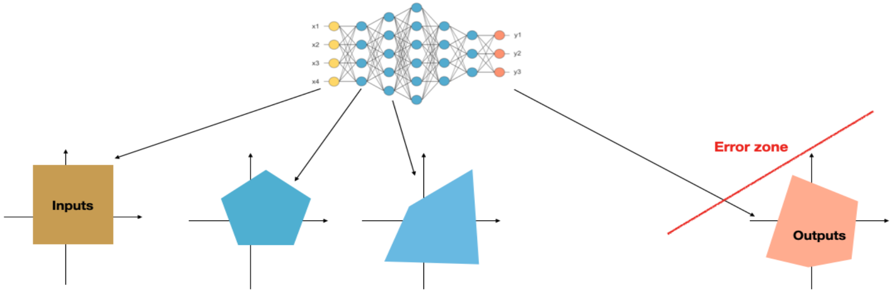
Eric Goubault and Sylvie Putot

FM 2024, September 9-13, Milan, Italy

# 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
- 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 $V \in [a, b]$ , $\varepsilon$ a random variable

With respect to most closely related work: Quantitative verification for neural networks using Probstars, Tran, H.D., Choi, S., Okamoto, H., Hoxha, B., Fainekos, G., Prokhorov, D., HSCC 2023

- inputs are arbitrary distributions (extending the Gaussian distribution hypothesis)
- our approach gives fully guaranteed probability bounds

# Problem Statement: propagating imprecise probabilities

**Problem (Probability bounds analysis)**

*Given a ReLU network $f$ and a constrained probabilistic input set*

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

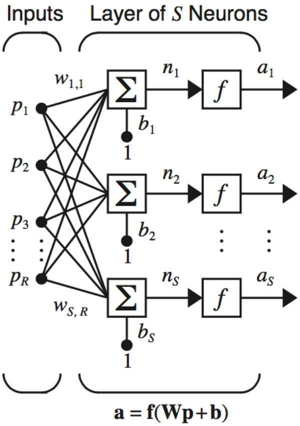
*where $\underline{F}$ and $\overline{F}$ are two cumulative distribution functions, compute a constrained probabilistic output set $\mathcal{Y}$ guaranteed to contain $\{f(X), X \in \mathcal{X}\}$.*
*For $X \in \mathbb{R}^n$, we note $\boldsymbol{P}(X \leq x) := \boldsymbol{P}(X_1 \leq x_1 \wedge X_2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

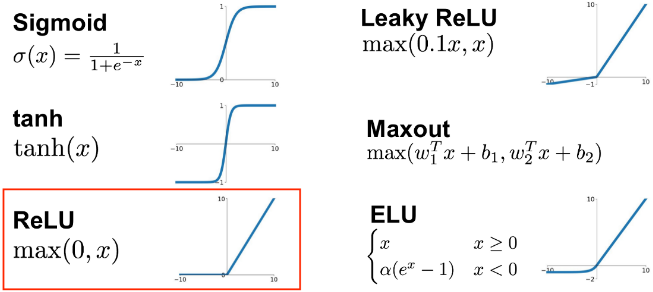
**Problem (Quantitative property verification)**

*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.*

# Feedforward ReLU neural network

Each layer consists in a linear transform followed by a non linear activation function:



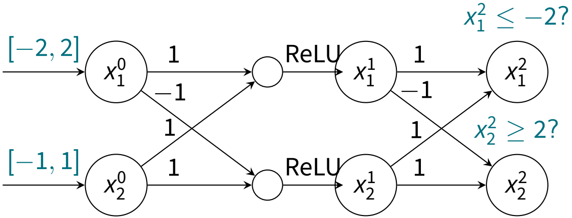




We focus on the classical ReLU activation function

# 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

# Imprecise probabilities: P-boxes and Dempster-Shafer structures

# Representation of imprecise probabilities: P-box

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

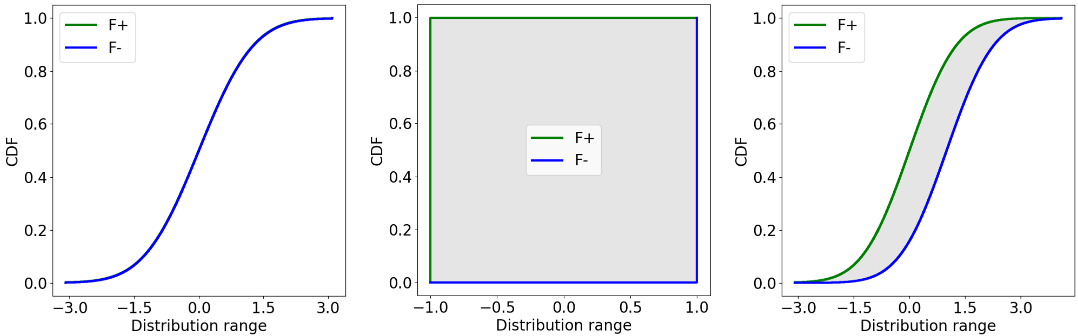
Given two (lower and upper) CDF (Cumulative Distribution Functions) $\underline{F}$ and $\overline{F}$ from $\mathbb{R}$ to $\mathbb{R}^+$ s.t. $\forall x \in \mathbb{R}, \underline{F}(x) \leq \overline{F}(x)$, the p-box $[\underline{F}, \overline{F}]$ represents the set of probability distributions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

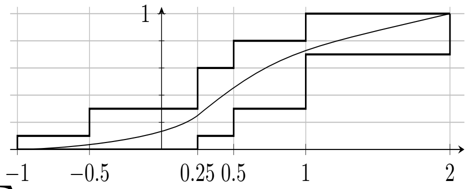
- Focal elements $t \in T$ (sets of values, here Intervals) with probability $w : T \to \mathbb{R}^+$

| $t \in T$ | [-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.*

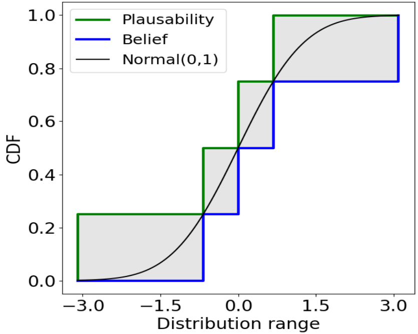


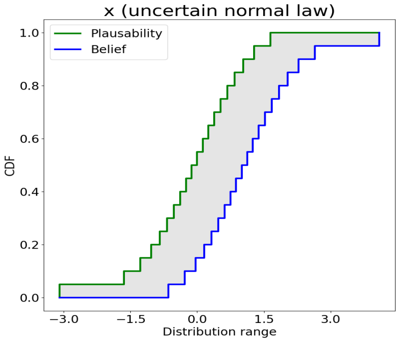

$$\sum_{t \in T, t \subseteq S} w(t) \leq P(S) \leq \sum_{t \in T, t \cap S \neq \emptyset} w(t)$$

# 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

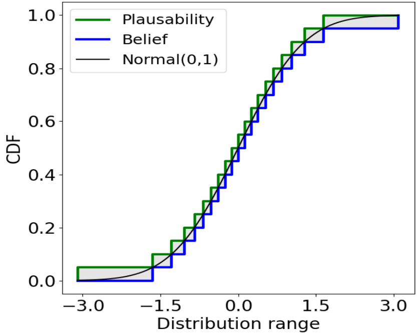


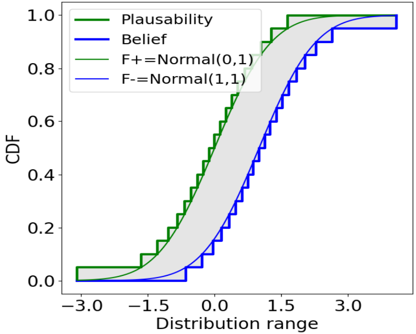



# From P-boxes to Dempster-Shafer Interval structures

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

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

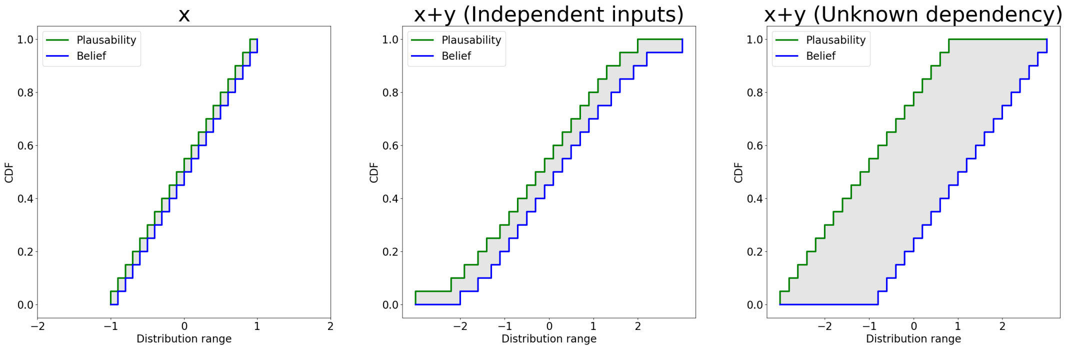


# Arithmetic on DSI structures

DSI structures can be propagated through arithmetic operations:

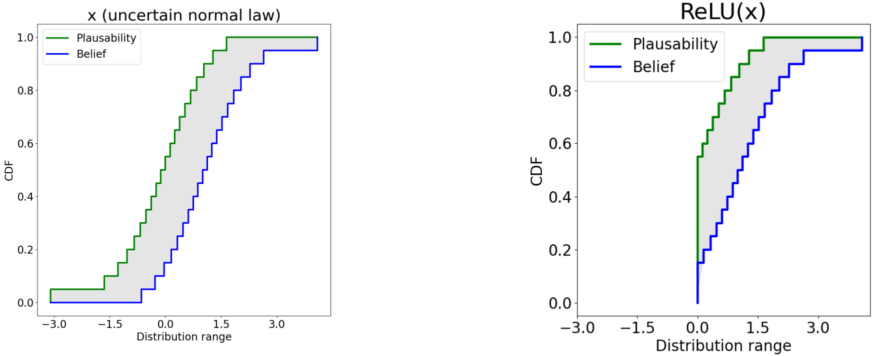
- 2 cases: independent inputs / unknown dependency
- relying on interval arithmetic / Frechet inequalities
- conservative approximations



# ReLU

**Lemma (ReLU of a DSI)**

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

    $d_l^{k+1} \leftarrow \sigma(\sum_{j=1}^{h_k} a_{lj}^k d_j^k + b_l^k)$ ▷ *Affine transform and ReLU - Dependency graph useful for choosing the right DSI operations (indep. or unknown dep.) in affine transforms*

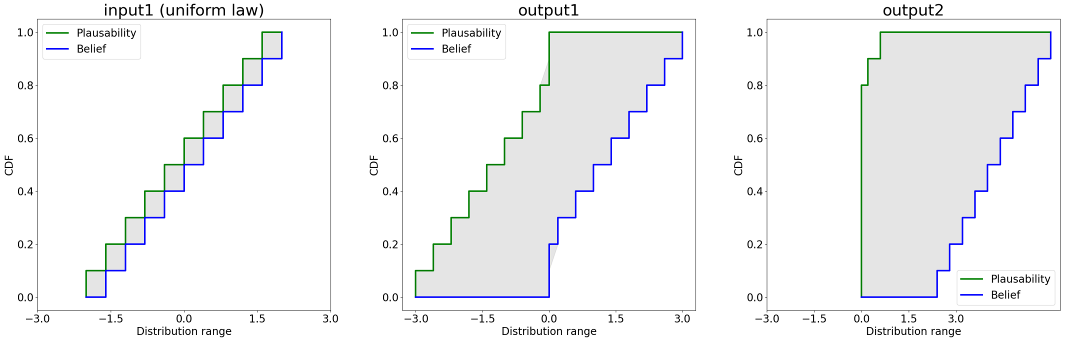
  ***end for***

***end for***

***return*** *$(d^L$,cdf$(Hd^L, w))$* ▷ *Vector of DSI for the output layer and probability bounds for property $Hz \leq w$*

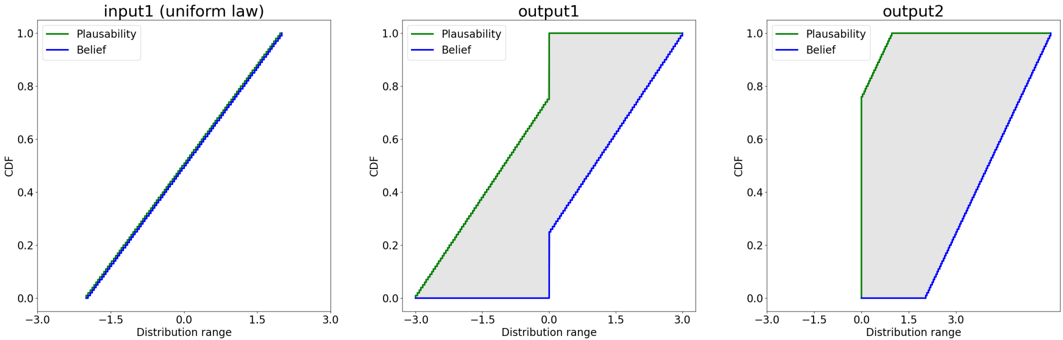
# Illustration on the toy example

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

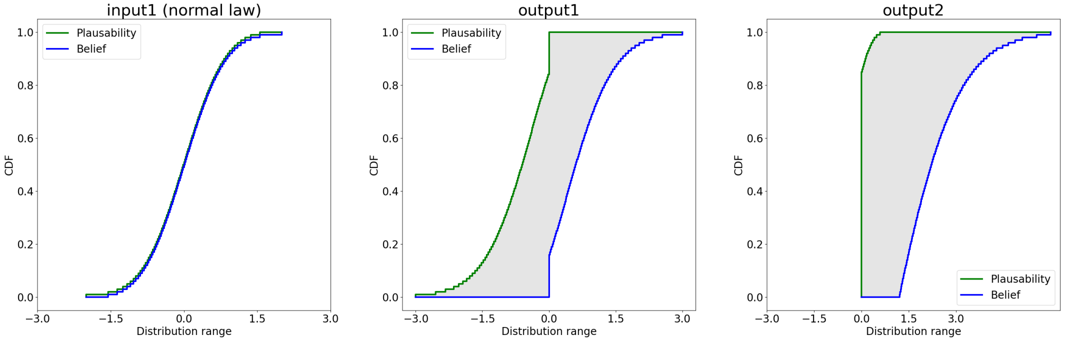
Input $x^0 = \begin{bmatrix} x_1^0 & x_2^0 \end{bmatrix}^\top \in [-2, 2] \times [-1, 1]$ with Uniform law on inputs




Finer discretization refines the approximation but the ranges are unchanged

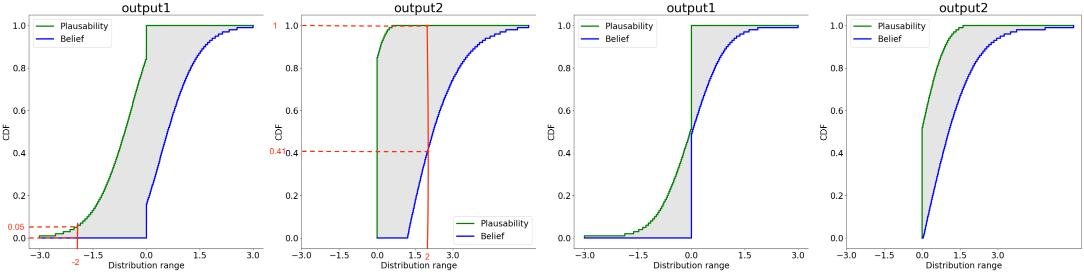
# Illustration on the toy example

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



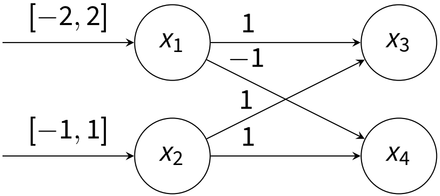
# Illustration on the toy example

## Unknown dependency on inputs vs independent inputs




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

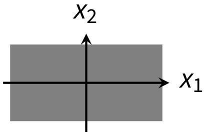
# Wrapping effect: example of the first affine layer




Initial domain:

$$-2 \leq x_1 \leq 2$$

$$-1 \leq x_2 \leq 1$$




Exact domain:

$$x_3 = x_1 - x_2$$

$$x_4 = x_1 + x_2$$

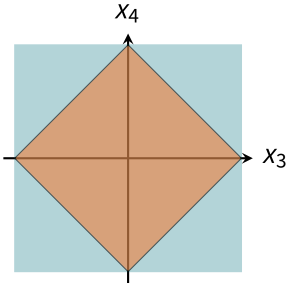
$$x_1, x_2 \in [-1, 1]$$

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 !

# Mitigating the wrapping effect: Probabilistic zonotopes and Dempster-Shafer Zonotopic Structures (DSZ)

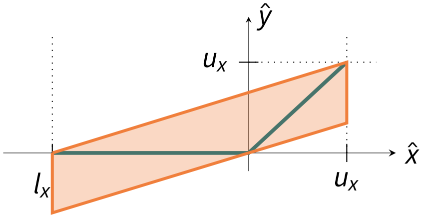
# Zonotopes and neural network reachability analysis

**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

## 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
- inspired from [Adje et al 2013] A. Adjé, O. Bouissou, J. Goubault-Larrecq, E. Goubault, S. Putot: Static Analysis of Programs with Imprecise Probabilistic Inputs. VSTTE 2013: 22-47

## 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
- Currently restricted to independent inputs

# NN analysis by DSZ (independent inputs)

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

$d^0_{\mathcal{Z}} = \{\langle \mathcal{Z}^0_{i_1 \dots i_{h_0}}, w^0_{1,i_1} \dots w^0_{h_0,i_{h_0}} \rangle,\ (i_1, \dots, i_{h_0}) \in [1,n]^{h_0}\} \leftarrow$ 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**

$d^L_{\mathcal{Z}} = \{\langle \mathcal{Z}^L_{i_1 \dots i_{h_0}}, w^0_{1,i_1} \dots w^0_{h_0,i_{h_0}} \rangle,\ (i_1, \dots, i_{h_0}) \in [1,n]^{h_0}\}$

$d^L \leftarrow$ 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

# Evaluation

# Implementation and Evaluation

## Julia implementation

- available from `https://github.com/sputot/DSZAnalysis` or `https://doi.org/10.5281/zenodo.12519084`.
- uses the LazySets and the NeuralVerification package for zonotopic NN analysis
- uses the ProbabilityBoundsAnalysis package for P-boxes / DSI analysis

## Examples and evaluation

- Toy example - corrected Table 1 in the paper (thanks to the RE reviewers !)
- ACAS Xu airplanes collision avoidance example
- Rocket lander example

# Comparing DSI, Prob. Zonotopes and DSZ: toy example

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

| Law (#FE) | DSI $P(x_1^2 \leq -2)$ | DSI $P(x_2^2 \geq 2)$ | DSI time | Prob. Zono. $P(x_1^2 \leq -2)$ | Prob. Zono. $P(x_2^2 \geq 2)$ | Prob. Zono. time | DSZ $P(x_1^2 \leq -2)$ | DSZ $P(x_2^2 \geq 2)$ | DSZ 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.
- In the paper, detailed calculation for the 3 approaches in the case of 2 focal elements.

# Comparisons to the state of the art

*[Tran et al 23] Quantitative Verification for Neural Networks using ProbStars, Tran et al, HSCC 2023*

**Examples**

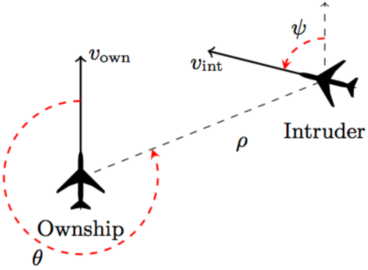
- ACAS Xu airplanes collision avoidance
- Rocket lander

**Inputs and configuration**

- Bounded (vector) inputs in [lb,ub], components follow independent Gaussian distributions with $\mu = (ub + lb)/2$ and $\sigma = (ub - m)/3$
- Timings and results given for [Tran et al 23] are from their paper:
  - parallelized (between 1 and 8 cores) and on a slightly stronger computer than ours
  - we reproduced a few analyzes: approx 7 to 10 times slower than their results with 1 core, approx 1.5 to 3.5 with 4 and 8 cores

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

- Produces aicraft advisory (clear-of-conflict, weak right, weak left, strong right, etc.)
- Array of 45 DNNs by discretizing $\tau$ and $a_{prev}$ ; each has 5 inputs ($\rho, \theta, \psi$, $v_{own}$ and $v_{int}$) and 5 outputs (score for each advisory).
- Fully connected ReLU feeedforward networks with 5 inputs, 6 hidden layers, 5 outputs




Properties:

$P_2 : y_1 > y_2 \wedge y_1 > y_3 \wedge y_1 > y_4 \wedge y_1 > y_5$

$P_3/P_4 : y_1 < y_2 \wedge y_1 < y_3 \wedge y_1 < y_4 \wedge y_1 < y_5$

*Reluplex : An Efficient SMT Solver for Verifying Deep Neural Networks, Katz et al, CAV 2017.*

# Comparing DSZ and ProbStars Prob. bounds on ACAS Xu

- (Manual) Input discretization: [5, 80, 50, 6, 5] for $P_2$, [5, 20, 1, 6, 5] for $P_3$ and $P_4$

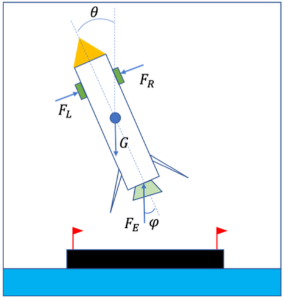
| Prop | Net | DSZ | | Probstar $p_f = e^{-5}$ | | Probstar $p_f = 0$ | |
|---|---|---|---|---|---|---|---|
| | | ***P*** | time | ***P*** | time | ***P*** | time |
| 2 | 1-6 | [0, 0.01999] | 46.4 | [2.8e-06,0.05283] | 206.7 | 1.87224e-05 | 1424 |
| 2 | 2-2 | [0.00423 0.0809] | 47.9 | [0.0195,0.094] | 299.0 | 0.0353886 | 2102.5 |
| 2 | 2-9 | [0, 0.0774684] | 51.0 | [0.000255,0.107] | 504.5 | 0.000997678 | 4561.2 |
| 2 | 3-1 | [0.0165, 0.08787] | 43.8 | [0.0305, 0.07263] | 202.7 | 0.044535 | 1086.4 |
| 2 | 3-6 | [0.0167, 0.1111] | 52.4 | [0.02078,0.1069] | 452.0 | 0.0335763 | 5224.4 |
| 2 | 3-7 | [6e-05, 0.1361] | 43.7 | [0.002319,0.075] | 331.1 | 0.00404731 | 2598 |
| 2 | 4-1 | [1e-05, 0.05353] | 40.9 | [0.00104,0.07162] | 305.3 | 0.00231247 | 1870.7 |
| 2 | 4-7 | [0.0129, 0.1056] | 44.4 | [0.02078,0.1081] | 418.9 | 0.04095 | 3407.8 |
| 2 | 5-3 | [0, 0.03939] | 40.0 | [1.59e-09,0.0326] | 139.7 | 1.81747e-09 | 418.8 |
| 3 | 1-7 | [1, 1] | 0.25 | [0.9801,0.9804] | 4.7 | 0.976871 | 3.6 |
| 4 | 1-9 | [1, 1] | 0.2 | [0.9796,0.98] | 3.6 | 0.989244 | 3.6 |

# Comparing DSZ and ProbStars Prob. bounds on ACAS Xu

- (Manual) Input discretization: [5, 80, 50, 6, 5] for $P_2$, [5, 20, 1, 6, 5] for $P_3$ and $P_4$
- Basic (sensitivity-based) discretization refinement algo: $P_2$, net 1-6, refining [1,1,1,1,1] to ensure width(prob interval)$\leq$ 0.05: 112s and [5, 81, 38, 5, 5], probability in [0, 0.0276]

| Prop | Net | DSZ | | Probstar $p_f = e^{-5}$ | | Probstar $p_f = 0$ | |
|---|---|---|---|---|---|---|---|
| | | ***P*** | time | ***P*** | time | ***P*** | time |
| 2 | 1-6 | [0, 0.01999] | 46.4 | [2.8e-06,0.05283] | 206.7 | 1.87224e-05 | 1424 |
| 2 | 2-2 | [0.00423 0.0809] | 47.9 | [0.0195,0.094] | 299.0 | 0.0353886 | 2102.5 |
| 2 | 2-9 | [0, 0.0774684] | 51.0 | [0.000255,0.107] | 504.5 | 0.000997678 | 4561.2 |
| 2 | 3-1 | [0.0165, 0.08787] | 43.8 | [0.0305, 0.07263] | 202.7 | 0.044535 | 1086.4 |
| 2 | 3-6 | [0.0167, 0.1111] | 52.4 | [0.02078,0.1069] | 452.0 | 0.0335763 | 5224.4 |
| 2 | 3-7 | [6e-05, 0.1361] | 43.7 | [0.002319,0.075] | 331.1 | 0.00404731 | 2598 |
| 2 | 4-1 | [1e-05, 0.05353] | 40.9 | [0.00104,0.07162] | 305.3 | 0.00231247 | 1870.7 |
| 2 | 4-7 | [0.0129, 0.1056] | 44.4 | [0.02078,0.1081] | 418.9 | 0.04095 | 3407.8 |
| 2 | 5-3 | [0, 0.03939] | 40.0 | [1.59e-09,0.0326] | 139.7 | 1.81747e-09 | 418.8 |
| 3 | 1-7 | [1, 1] | 0.25 | [0.9801,0.9804] | 4.7 | 0.976871 | 3.6 |
| 4 | 1-9 | [1, 1] | 0.2 | [0.9796,0.98] | 3.6 | 0.989244 | 3.6 |

# Rocket lander




- feedforward neural networks with 9 inputs, 3 outputs, and 5 hidden layers with 20 ReLU neurons per layer
- $P_1$: when $-20^\circ \leq \theta \leq -6^\circ, \omega < 0\,, \phi' \leq 0$, $F_S' \leq 0$, the output action should be $\phi < 0$ or $F_S < 0$: the agent should prevent the rocket from tilting to the right. ($P_2$ similar)

*Neural Network Repair with Reachability Analysis. Yang et al, FORMATS 2022*

*Quantitative Verification for Neural Networks using ProbStars, Tran et al, HSCC 2023*

# Comparing DSZ and ProbStars Prob. bounds: rocket lander

- Input discretization: [7, 12, 10, 17, 9, 7, 1, 1, 2, 1, 1]

| Prop | Net | DSZ | | Probstar $p_f = 1e-5$ | | Probstar $p_f = 0$ | |
|---|---|---|---|---|---|---|---|
| | | ***P*** | time | ***P*** | time | ***P*** | time |
| 1 | 0 | [0, 0.03387] | 77.8 | [4.15e-09, 0.06748] | 1158.6 | 7.978e-08 | 5903.7 |
| 2 | 0 | [0, 0.01352] | 83.7 | [0,0.1053] | 2216 | 0 | 13132.7 |
| 1 | 1 | [0, 0.01985] | 80.5 | [0,0.0536] | 1229.7 | 8.68e-08 | 5163.9 |
| 2 | 1 | [0, 0.00055] | 69.1 | [0, 0.0161751] | 448.5 | 0 | 1495.6 |

# Conclusion and Future Work

- DSI/DSZ for ReLU networks generalize state of the art of NN probabilistic verification
  - more efficient than state of the art but still scalability issues
  - efficiency given by the input layer discretization size
- Future work:
  - Distributions with unbounded support (handled for DSI, extension to DSZ more of a technical/implementation issue)
  - Other initial abstractions/discretizations
    - avoid inefficient initial staircase / DSI discretization step
    - handle multivariate input distributions (lift independence hypothesis for DSZ)

We are looking for a motivated PhD student !