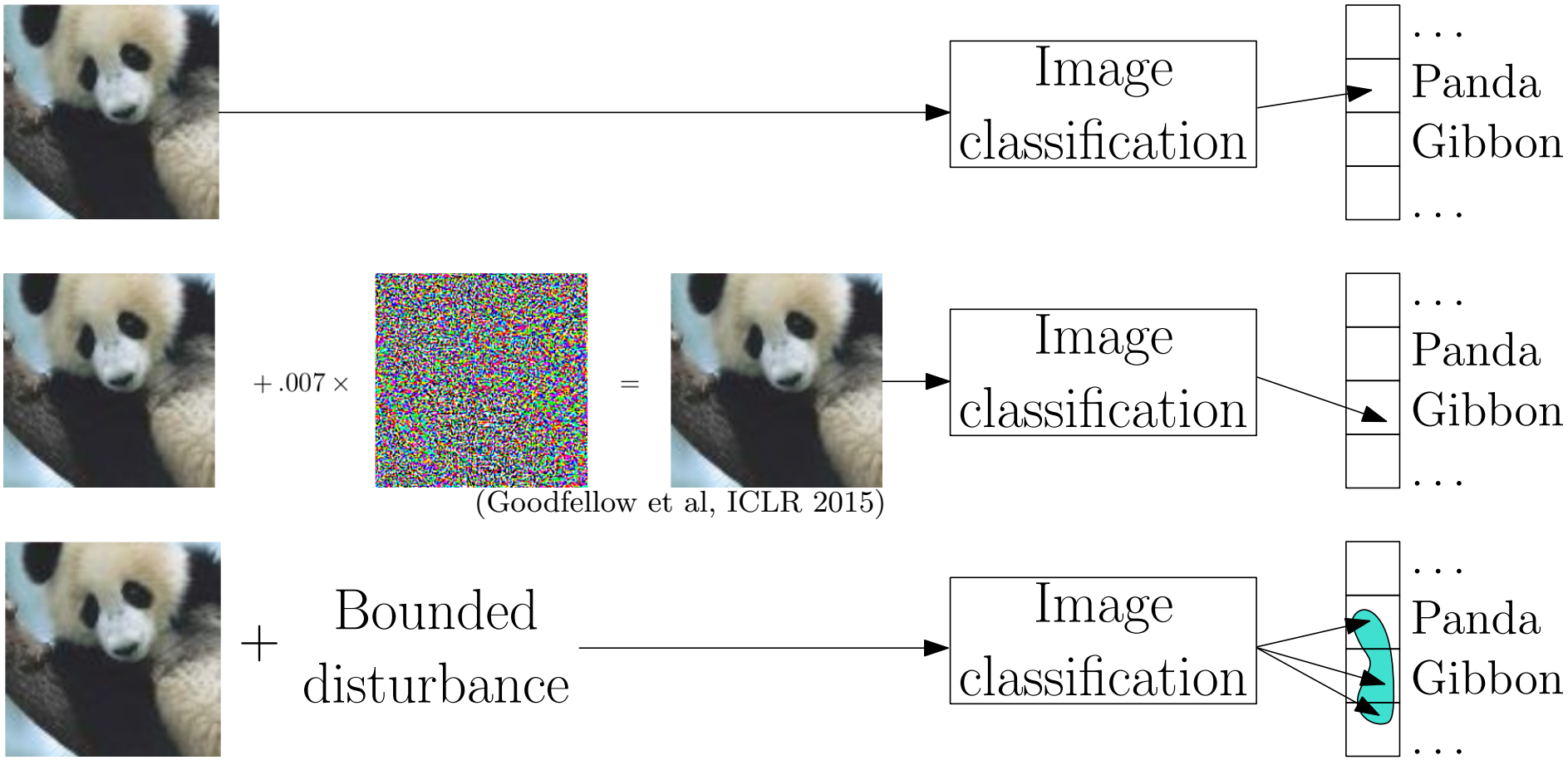
Reachability analysis of neural networks using mixed monotonicity

Pierre-Jean Meyer

Université Gustave Eiffel

7th of December 2022

Motivation: robustness of image classification

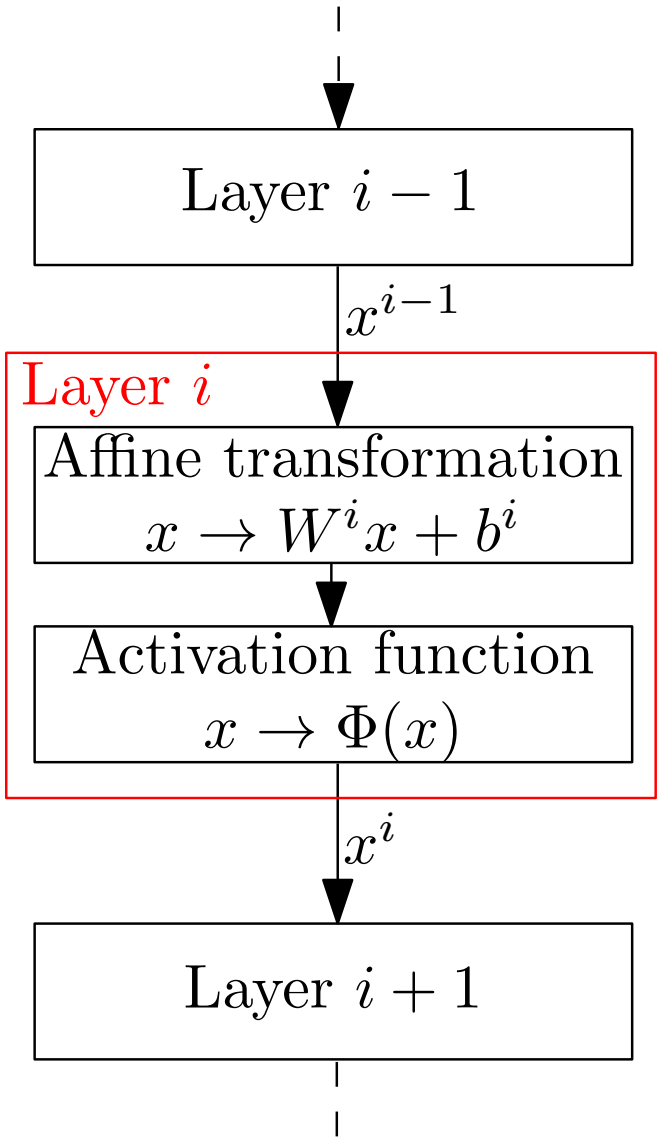


Reachability analysis: does the set of all reachable outputs spans more than one class ?

Neural networks

L-layer feedforward neural network

$$x^i = \Phi(W^i x^{i-1} + b^i),\ \forall i$$



Neural networks

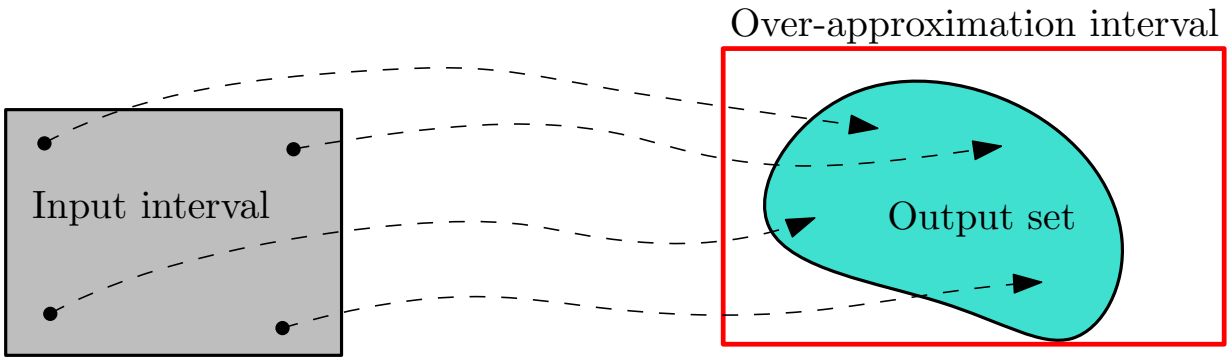
L-layer feedforward neural network

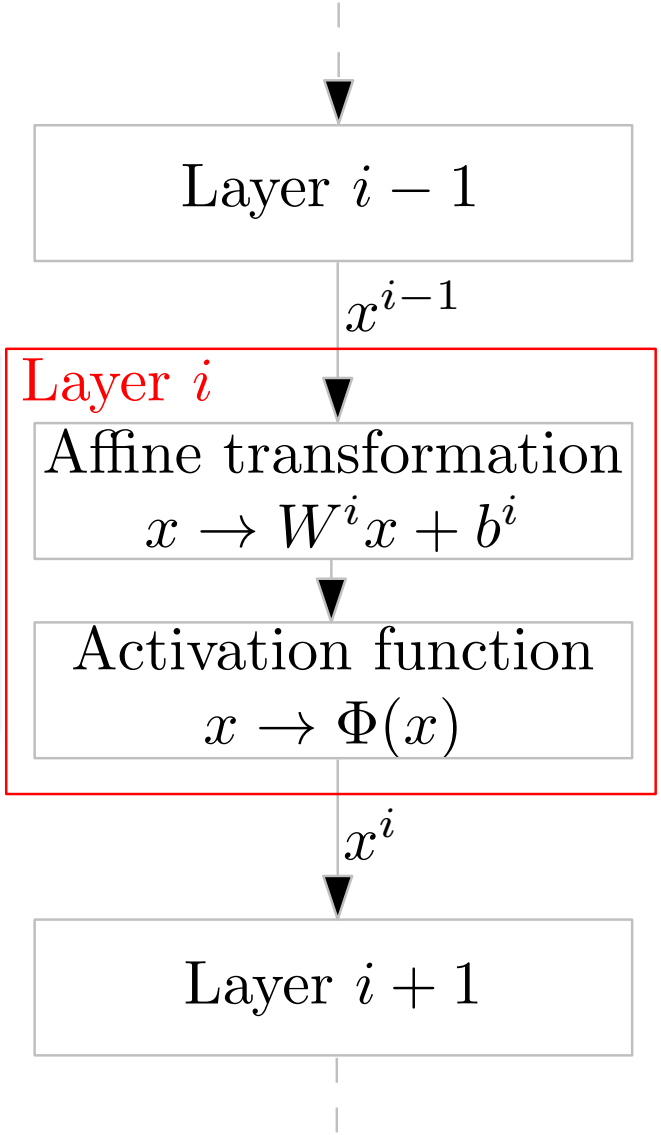
$$x^i = \Phi(W^i x^{i-1} + b^i), \ \forall i$$

Reachability problem

Given the input interval $[\underline{x^0}, \overline{x^0}]$, find an interval over-approximation $[\underline{x^L}, \overline{x^L}]$ of the output set:

$$\{x^L \mid x^0 \in [\underline{x^0}, \overline{x^0}]\} \subseteq [\underline{x^L}, \overline{x^L}].$$





Mixed-monotonicity reachability analysis

$y = f(x),\ x \in \mathbb{R}^n,\ y \in \mathbb{R}^m$

Assumption

$$f'(x) \in [\underline{J}, \overline{J}] \subseteq \mathbb{R}^{m \times n},\ \forall x \in [\underline{x}, \overline{x}]$$

Let J^* be the center of $[\underline{J}, \overline{J}]$

Proposition

$$f_i(x) \in \left[f_i(\underline{z}^i) - |\alpha^i|(\overline{x} - \underline{x}), f_i(\overline{z^i}) + |\alpha^i|(\overline{x} - \underline{x}) \right],\ \forall i \in \{1, \dots, m\}$$

where $\underline{z}^i, \overline{z^i} \in \mathbb{R}^n$ and $\alpha^i \in \mathbb{R}^{1 \times n}$ such that $\forall j \in \{1, \dots, n\}$,

$$(\underline{z}^i_j, \overline{z^i}_j, \alpha^i_j) = \begin{cases} (\underline{x}_j, \overline{x}_j, \min(0, \underline{J}_{ij})) & \text{if } J^*_{ij} \geq 0, \\ (\overline{x}_j, \underline{x}_j, \max(0, \overline{J}_{ij})) & \text{if } J^*_{ij} \leq 0. \end{cases}$$

Bounding the derivative of the network

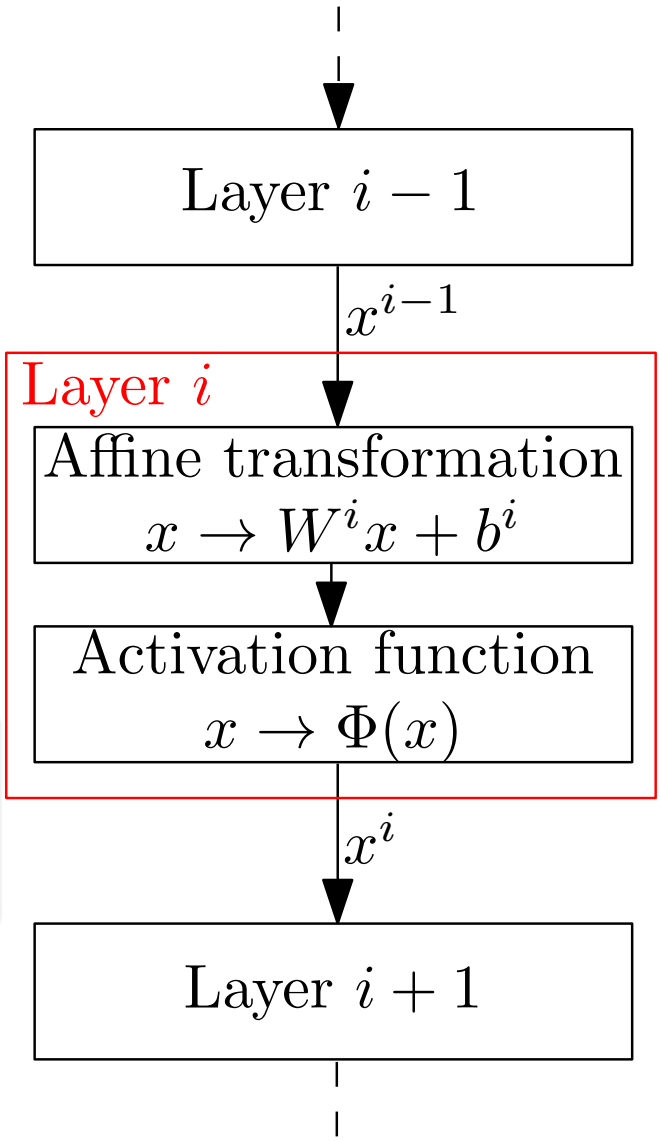
Goal: bounding the derivative of the network

Layer i:

$$x^i = \Phi(W^i x^{i-1} + b^i)$$

Derivative of layer i:

$$\Phi'(W^i x^{i-1} + b^i) * W^i$$



Proposition

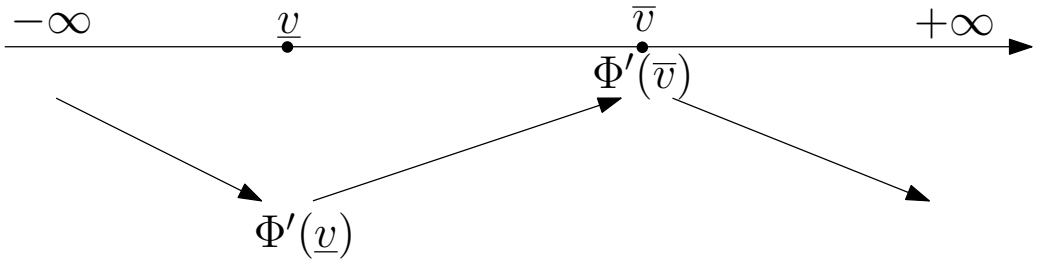
If Φ is continuous, the derivative of the whole neural network is bounded.

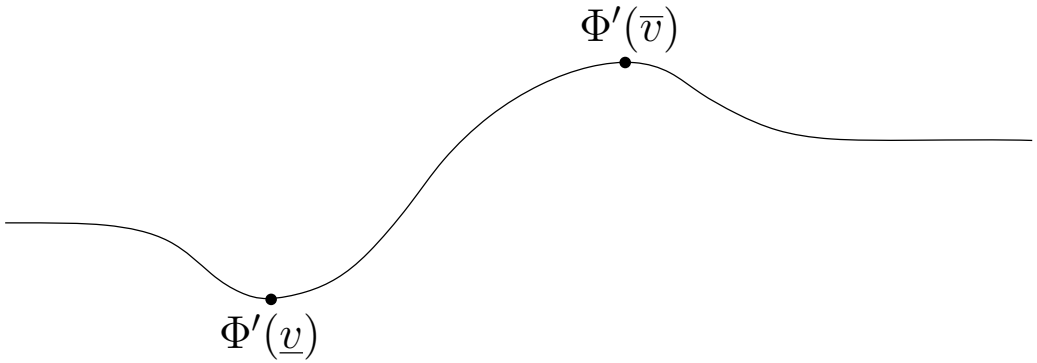
Main challenge: how to compute the bounds on the derivative of the activation function ?

Bounding the derivative of the activation function

Subclass of activation functions whose derivative is:

- non-increasing until global min $\Phi'(\underline{v})$
- non-decreasing until global max $\Phi'(\overline{v})$
- non-increasing

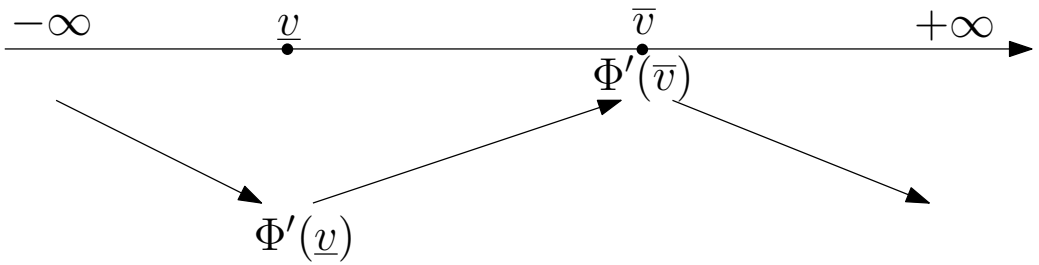




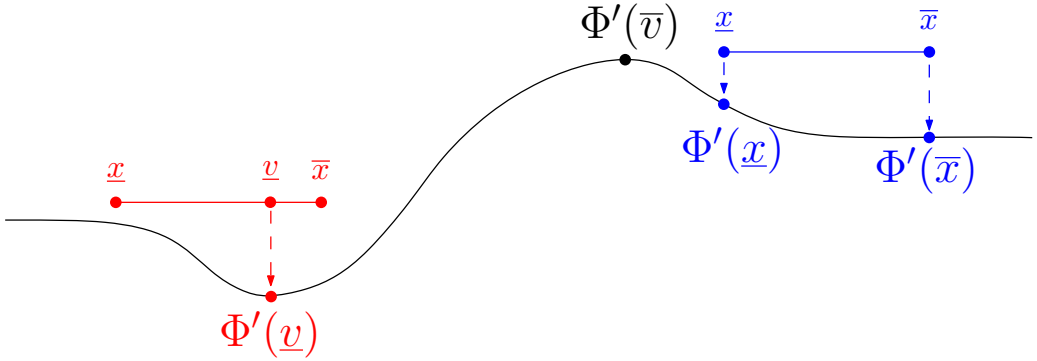
Bounding the derivative of the activation function

Subclass of activation functions whose derivative is:

- non-increasing until global min $\Phi'(\underline{v})$
- non-decreasing until global max $\Phi'(\overline{v})$
- non-increasing



Bounding function for Φ' created just from knowing $\underline{v}$ and $\overline{v}$



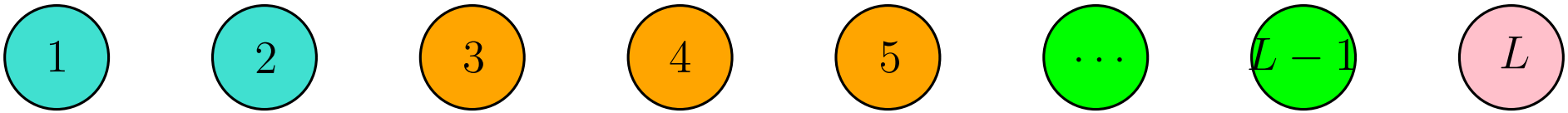
$$\min_{x \in [\underline{x}, \overline{x}]} \Phi'(x) = \begin{cases} \Phi'(\underline{v}) & \text{if } \underline{v} \in [\underline{x}, \overline{x}] \\ \min(\Phi'(\underline{x}), \Phi'(\overline{x})) & \text{else} \end{cases}$$

Intermediate decompositions

One subsystem per layer

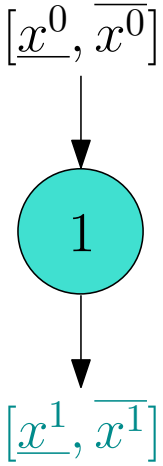
Whole network as a single system

Intermediate decomposition



Challenge: which decomposition gives the tightest results ?

Main algorithm



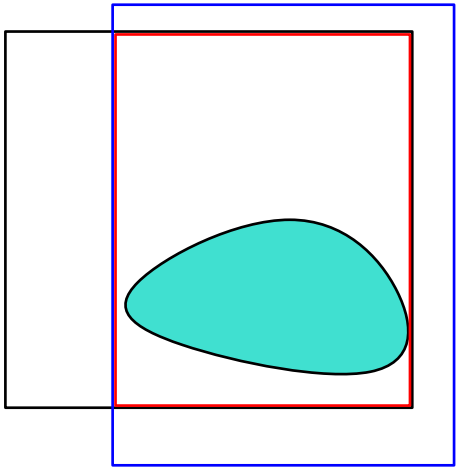
Partial networks ending at layer *i*

- 1 for layer 1

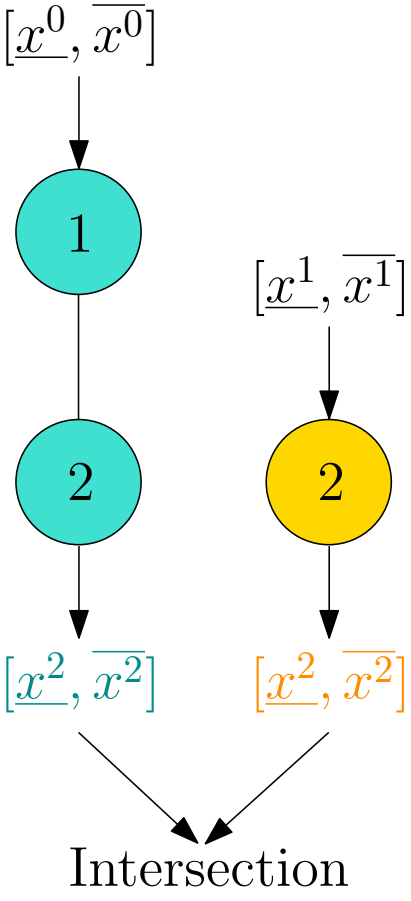
Main algorithm

Partial networks ending at layer i

- 1 for layer 1
- 2 for layer 2



Intersection of two over-approximations
$\rightarrow$ tighter over-approximation



Main algorithm

Partial networks ending at layer i

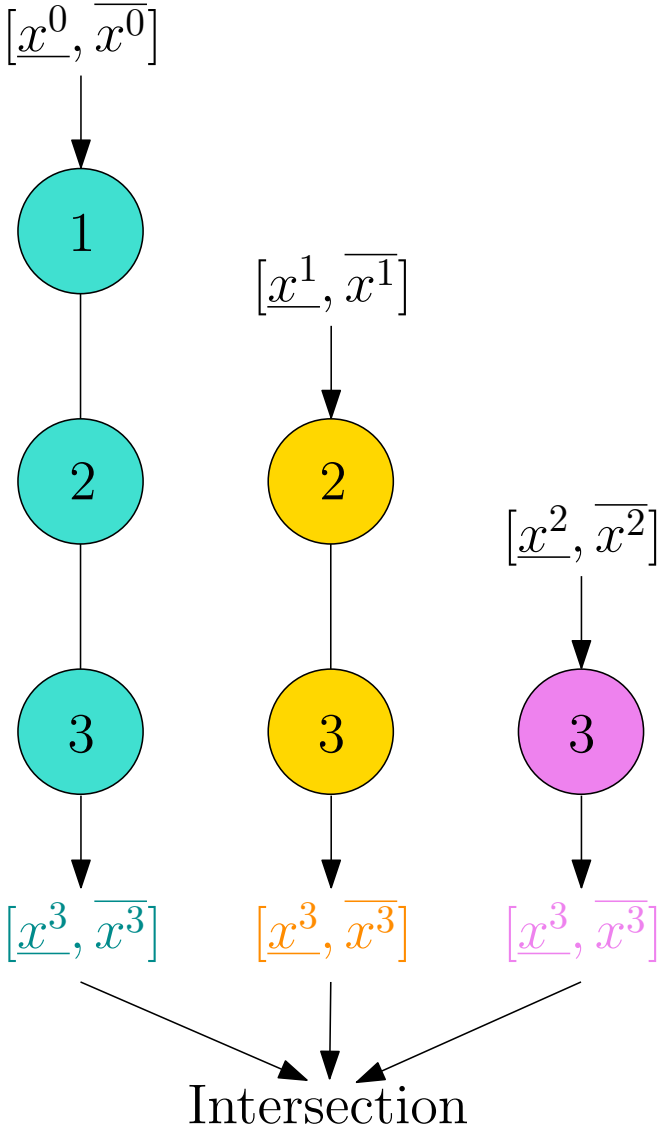
- 1 for layer 1
- 2 for layer 2
- 3 for layer 3
- ...

Complexity in the network depth L

Reachability analysis applied to

$$\frac{L(L+1)}{2} \text{ partial networks}$$

Tighter output bounds than with any specific layer decomposition

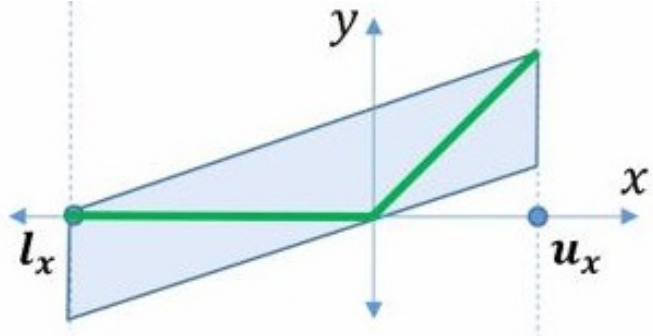


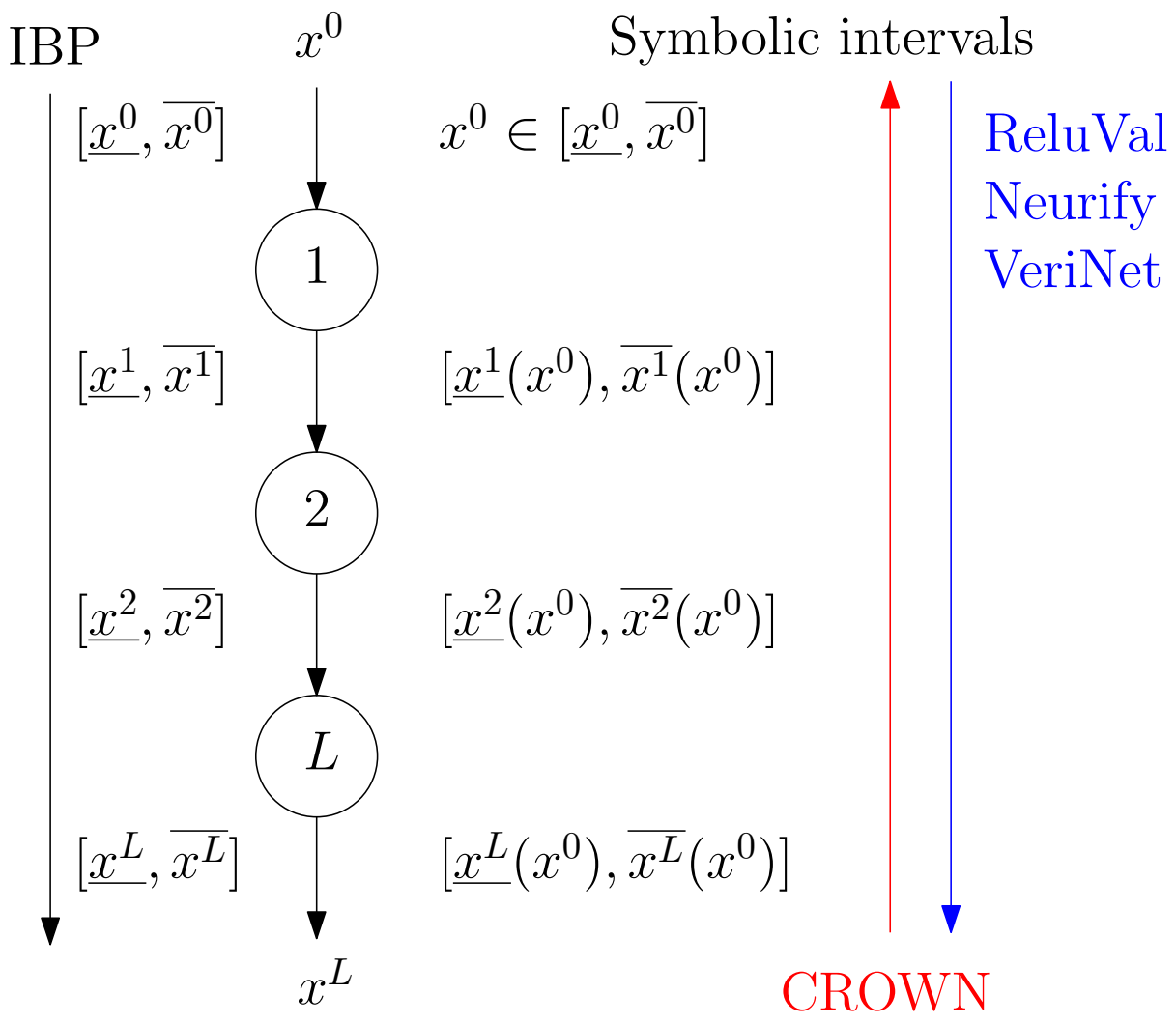
Numerical comparisons

Compared methods

- IBP
- ReluVal
- Neurify
- VeriNet
- CROWN

Linear relaxations of activation functions





Existing benchmarks (2021 VNN competition)

MNIST dataset

Input width: 784 (28×28 pixels)

↓

Hidden layers

AF	ReLU	Sigmoid
Width	256	200
Depth	2	6

↓

Output width: 10

Comparison on 250 **pictures**

- Faster than ReluVal/Neurify
- Tighter than IBP
- VeriNet/CROWN fail on Sigmoid

Method	**ReLU**	**Sigmoid**
IBP	100%	100%
ReluVal	80%	-
Neurify	72.4%	-
VeriNet	70.4%	100%
CROWN	70.4%	100%

Random networks - Setup

Limitations of benchmarks

- Only 2 specific networks
- Only popular activations

	Small NN	**Large NN**
# of NN	10000	1000
Depth	1-5	5-10
Input width	1-10	500-1000
Hidden width	1-30	100-200
Output width	1-10	10-50

Activation	**ReLU** PW affine	**TanH** S-shaped	**ELU** Monotone	**SiLU** Non-monotone
ReluVal				
Neurify				
VeriNet				
CROWN				
IBP				
Mixed-mono				

Random networks - Results

Computation time

- IBP/VeriNet: fastest + time per neuron is size-independent
- Small NN: Mixed-monotonicity is the slowest
- Large NN: ReluVal/Neurify are the slowest

Width of over-approximation

- Always tighter than (or equal to) IBP
- None of the other 5 is always better than the others
- Large ReLU NN: mostly looser
- Large TanH NN: mostly equal (approximate saturations)
- Large ELU NN
  - tighter than VeriNet in 98% cases (79% strictly)
  - CROWN always fails (too conservative)

Conclusions

New reachability method for neural networks

- Generality: applicable to **any continuous activation function**
  $\rightarrow$ can consider new and more performant non-monotone activation
- Performances: good **complementarity** with other tools
- Complexity: main limitation, especially on smaller networks

Perspectives

- Combine the reachability analysis with iterative refinement
  $\rightarrow$ full verification tool
- Reachability with respect to uncertainty on the network parameters
  $\rightarrow$ safe training, network repair

Contact: `pierre-jean.meyer@univ-eiffel.fr`

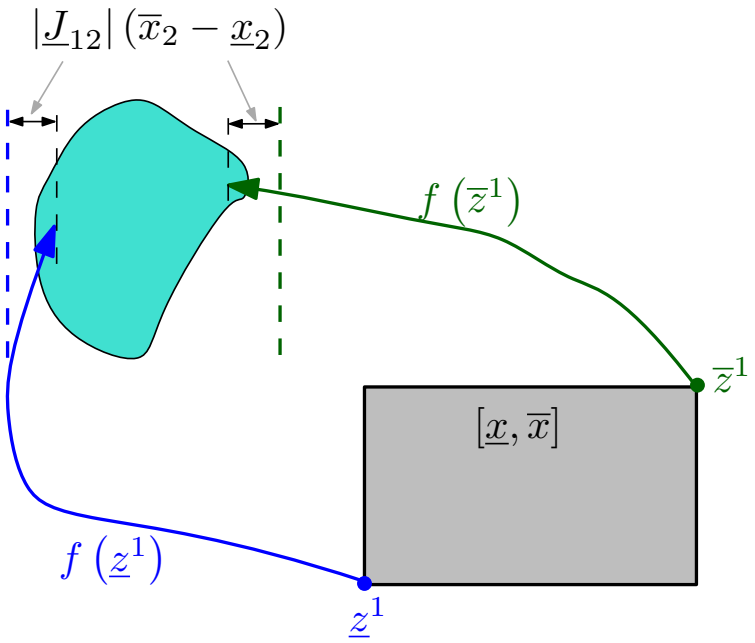
Mixed monotonicity - Illustration

$$sign\,(J^*) = \begin{pmatrix} + & + \\ - & + \end{pmatrix} \qquad \begin{array}{rcl} \alpha^1 & = & (0 \quad \underline{J}_{12}) \\ \alpha^2 & = & (0 \quad 0) \end{array}$$

Dimension 1:

$$\underline{f_1}([\underline{x}, \overline{x}]) = f_1(\underline{z}^1) - |\underline{J}_{12}|(\overline{x}_2 - \underline{x}_2)$$
$$\overline{f_1}([\underline{x}, \overline{x}]) = f_1(\overline{z}^1) + |\underline{J}_{12}|(\overline{x}_2 - \underline{x}_2)$$

$$\underline{z}^1 = \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix} \quad \overline{z}^1 = \begin{pmatrix} \overline{x}_1 \\ \overline{x}_2 \end{pmatrix}$$



Mixed monotonicity - Illustration

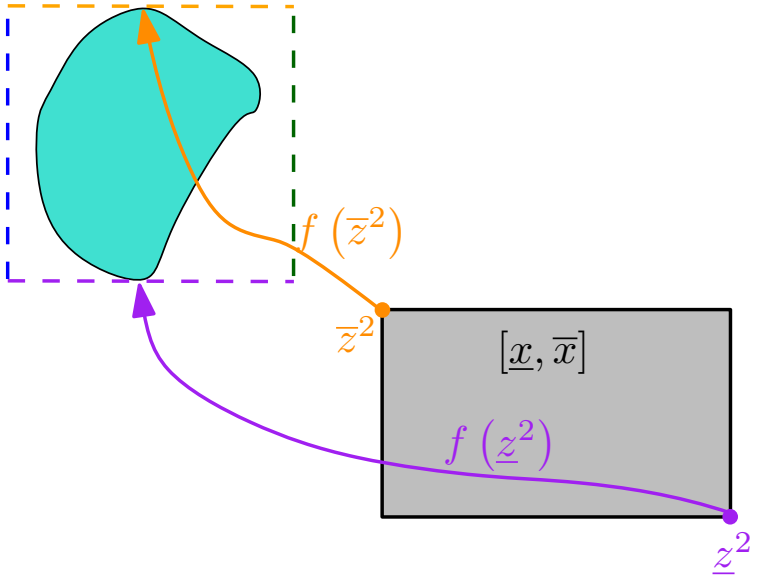
$$sign\,(J^*) = \begin{pmatrix} + & + \\ - & + \end{pmatrix} \qquad \begin{array}{rcl} \alpha^1 & = & (0 \quad \underline{J}_{12}) \\ \alpha^2 & = & (0 \quad 0) \end{array}$$

Dimension 2:

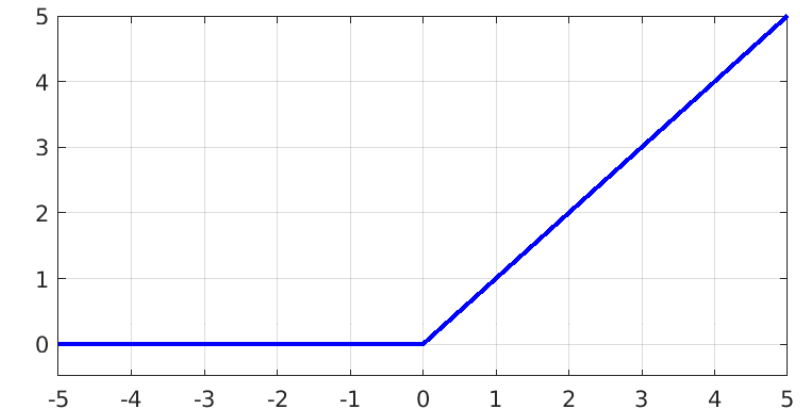
$\underline{f_2}([\underline{x}, \overline{x}]) = f_2(\underline{z}^2) + 0$

$\overline{f_2}([\underline{x}, \overline{x}]) = f_2(\overline{z}^2) + 0$

$$\underline{z}^2 = \begin{pmatrix} \overline{x}_1 \\ \underline{x}_2 \end{pmatrix} \quad \overline{z}^2 = \begin{pmatrix} \underline{x}_1 \\ \overline{x}_2 \end{pmatrix}$$



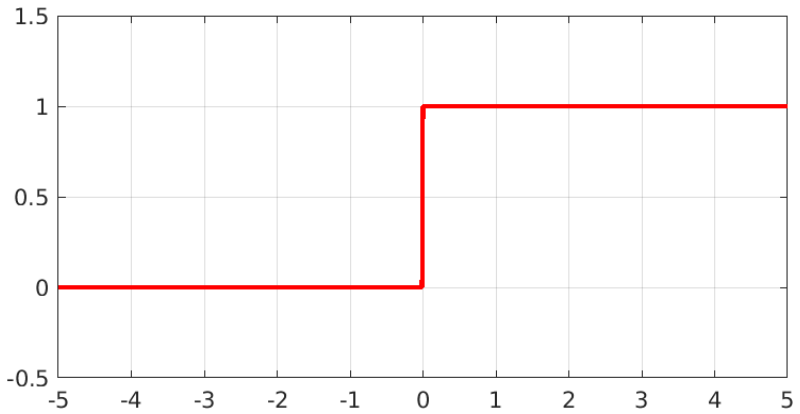
Activation functions - Examples



Activation function

ReLU (piecewise affine)

$$\Phi(x) = \max(0, x)$$



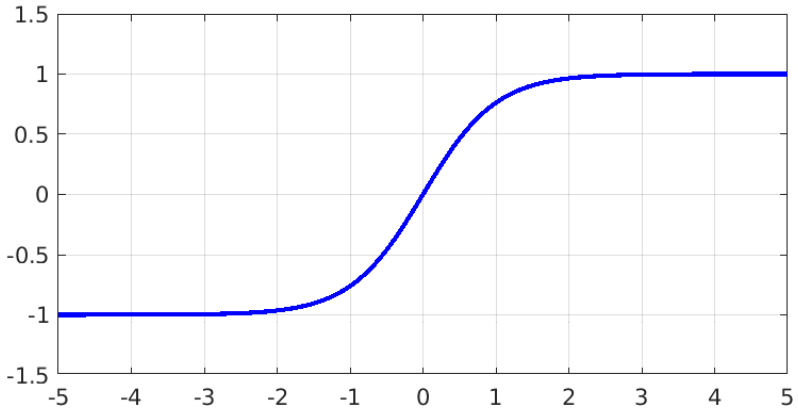
Derivative

$$\Phi'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

$$\underline{v} = -\infty \qquad \Phi'(\underline{v}) = 0$$

$$\overline{v} = +\infty \qquad \Phi'(\overline{v}) = 1$$

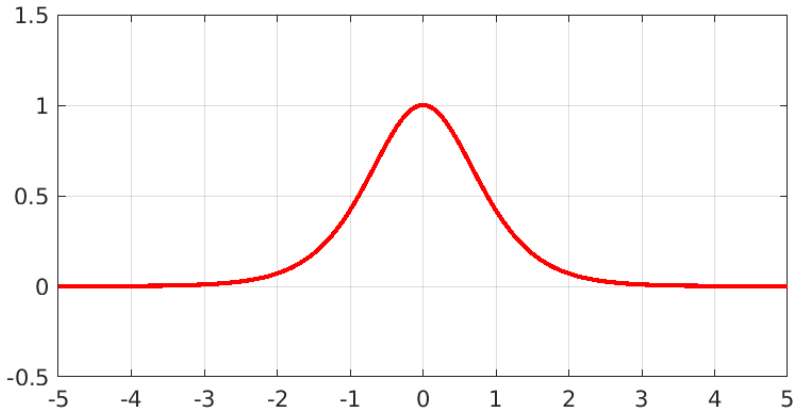
Activation functions - Examples



Activation function

Hyperbolic tangent (S-shaped)

$$\Phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



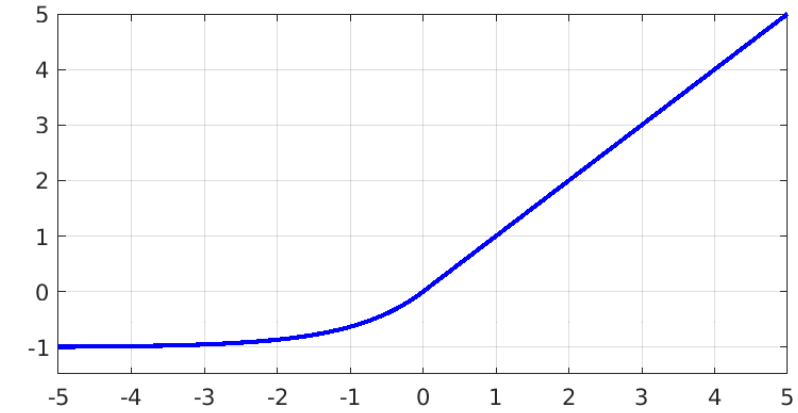
Derivative

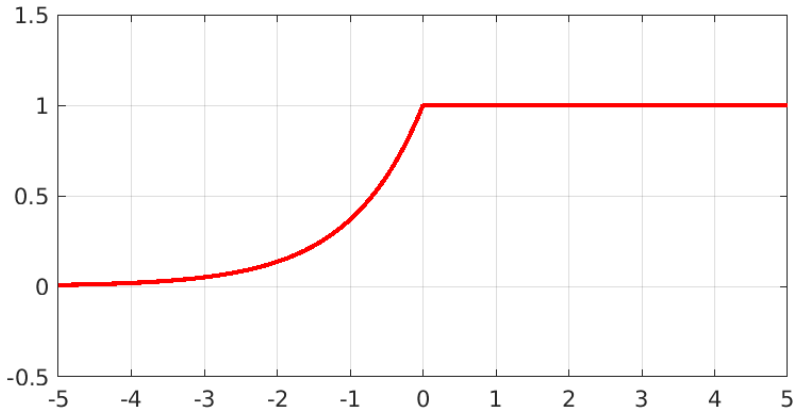
$$\Phi'(x) = 1 - \Phi(x)^2$$

$$\underline{v} = -\infty \qquad \Phi'(\underline{v}) = 0$$

$$\overline{v} = 0 \qquad \Phi'(\overline{v}) = 1$$

Activation functions - Examples





Activation function

ELU (monotone)

$$\Phi(x) = \begin{cases} e^x - 1 & \text{if } x < 0 \\ x & \text{if } x > 0 \end{cases}$$

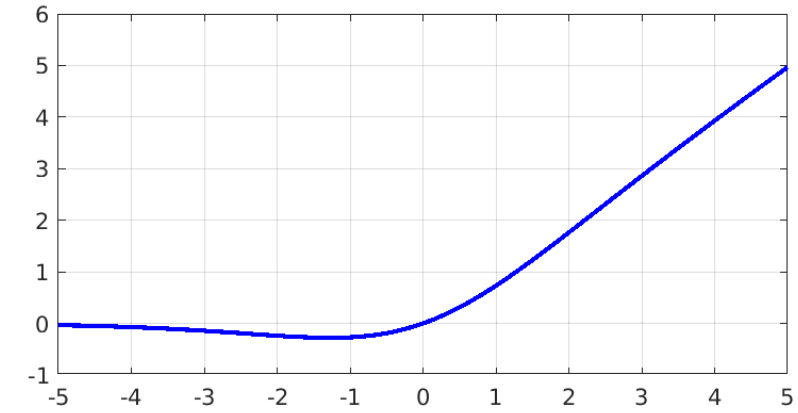
Derivative

$$\Phi'(x) = \begin{cases} e^x & \text{if } x \leq 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

$$\underline{v} = -\infty \qquad \Phi'(\underline{v}) = 0$$

$$\overline{v} = +\infty \qquad \Phi'(\overline{v}) = 1$$

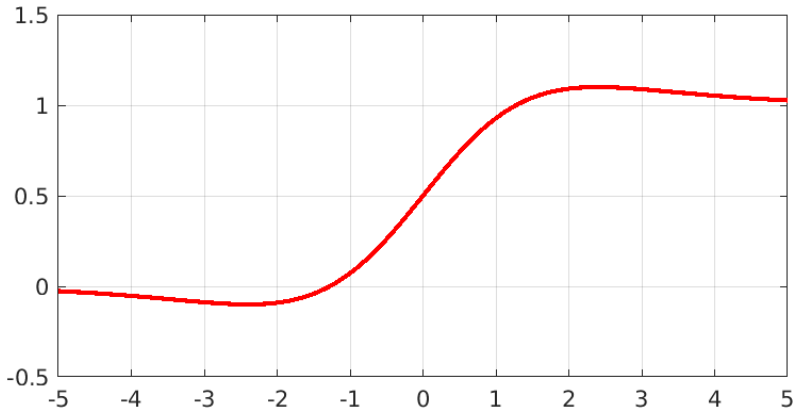
Activation functions - Examples



Activation function

SiLU (non-monotone)

$$\Phi(x) = \frac{x}{1 + e^{-x}}$$



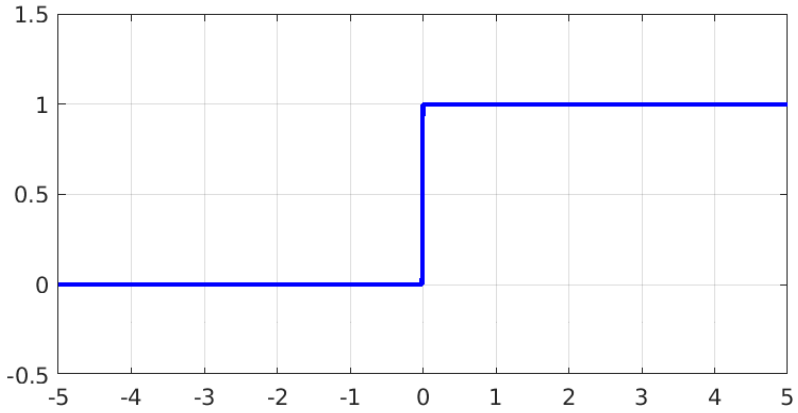
Derivative

$$\Phi'(x) = \frac{1 + e^{-x} + xe^{-x}}{(1 + e^{-x})^2}$$

$$\underline{v} = -2.4 \qquad \Phi'(\underline{v}) = -0.1$$

$$\overline{v} = 2.4 \qquad \Phi'(\overline{v}) = 1.1$$

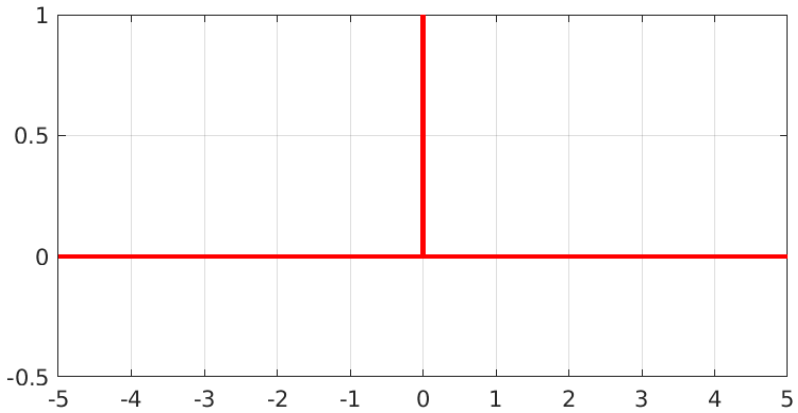
Activation functions - Examples



Activation function

Binary step (discontinuous)

$$\Phi(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

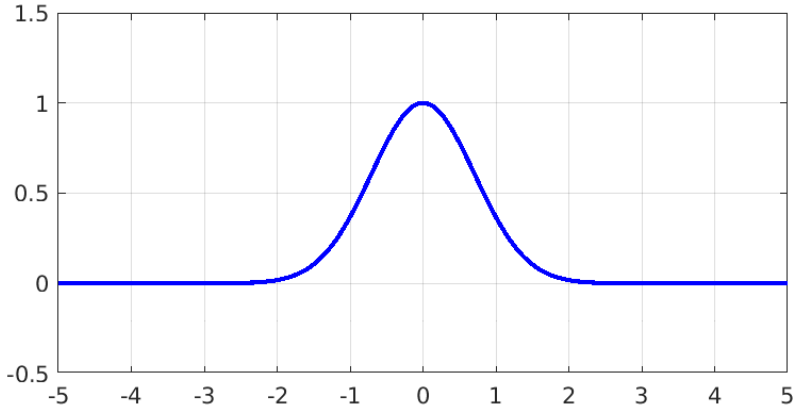


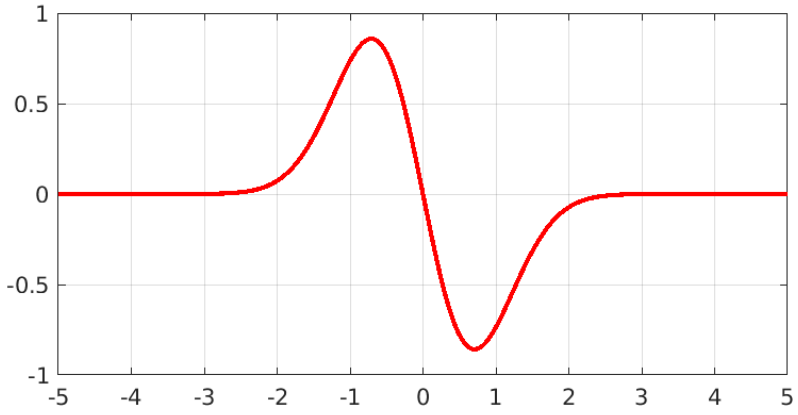
Derivative

$$\Phi'(x) = \begin{cases} 0 & \text{if } x \neq 0 \\ +\infty & \text{if } x = 0 \end{cases}$$

Unbounded

Activation functions - Examples





Activation function

Gaussian (non-monotone)

$$\Phi(x) = e^{-x^2}$$

Derivative

$$\Phi'(x) = -2xe^{-x^2}$$

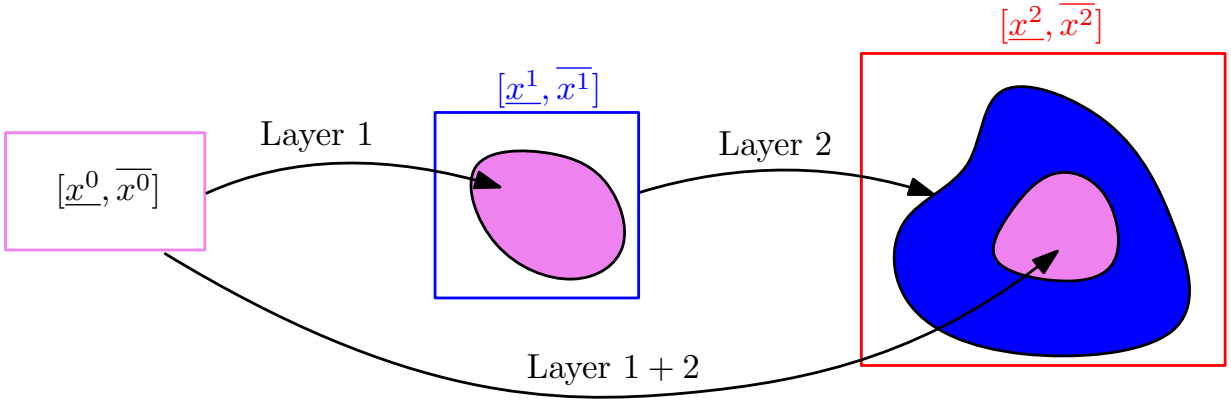
Only $-\Phi'$ has the correct shape

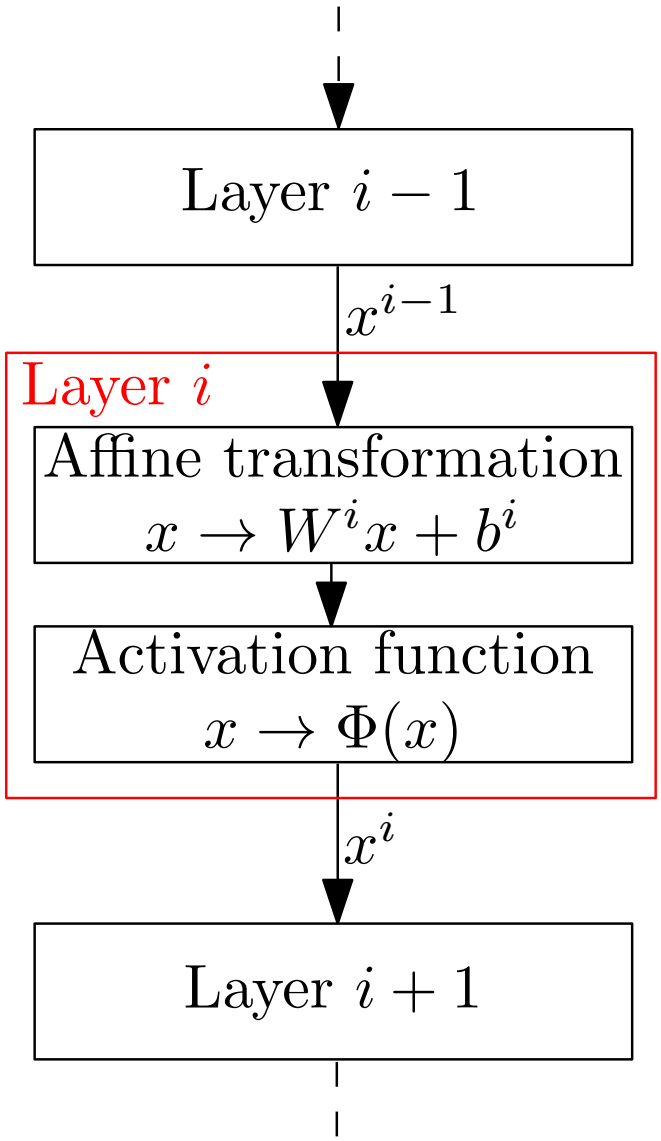
Reachability analysis: layer by layer

Layer 1

- Input interval: $[\underline{x^0}, \overline{x^0}]$
- Layer description: $x^1 = \Phi(W^1 x^0 + b^1)$
- Layer derivative: $\Phi'(W^1 x^0 + b^1) * W^1$
- Output over-approximation: $[\underline{x^1}, \overline{x^1}]$

Strength: good approximation for isolated layer
Weakness: lose input dependency



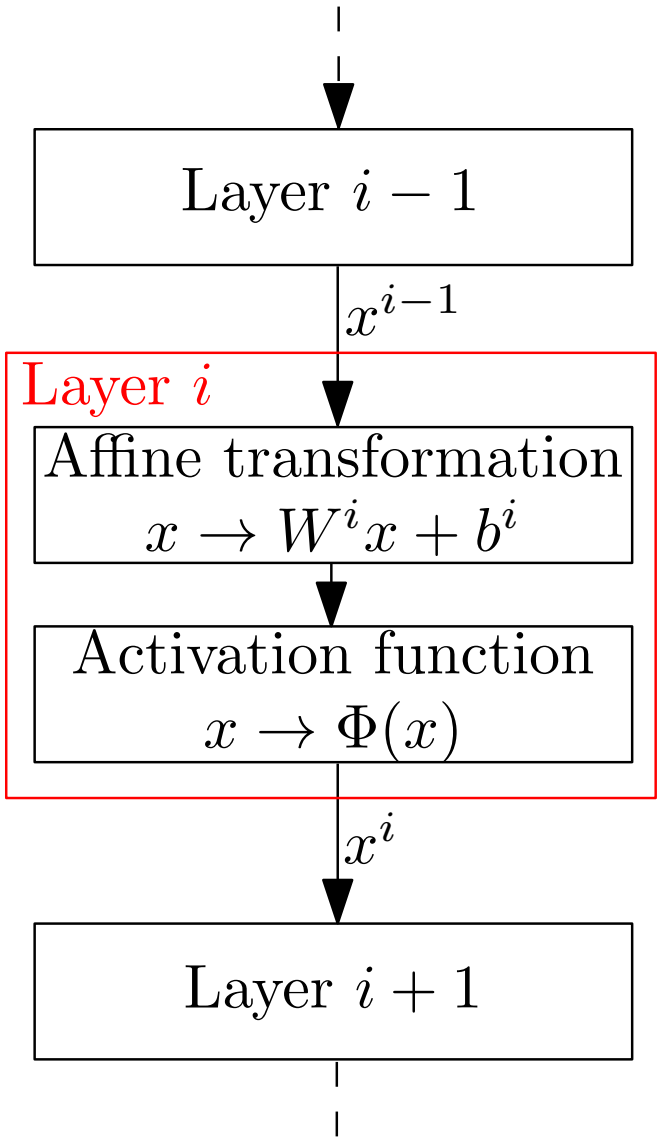


Reachability analysis: whole network

Whole neural network

- Input interval: $[\underline{x^0}, \overline{x^0}]$
- System: $x^i = \Phi(W^i x^{i-1} + b^i), \ \forall i$
- Derivative: $\prod_{i=1}^{L} \Phi'(W^i x^{i-1} + b^i) * W^i$
- Output over-approximation: $[\underline{x^L}, \overline{x^L}]$

Strength: preserve input dependency
Weakness: conservative derivative bounds



VNN benchmarks - computation time

Method	**ReLU**-2	**ReLU**-4	**ReLU**-6	**Sigmoid**-6
IBP	0.012	0.019	0.025	0.016
ReluVal	42	62	85	-
Neurify	43	62	84	-
VeriNet	0.012	0.019	0.026	0.017
CROWN	0.78	4.1	10	5.7
Mixed-Monotonicity	15	36	69	39

Table: Average computation time per network in s

VNN benchmarks - width

Method	**ReLU**-2	**ReLU**-4	**ReLU**-6	**Sigmoid**-6
IBP	100%	100%	100%	100%
ReluVal	80%	74.8%	65.6%	-
Neurify	72.4%	72%	58.8%	-
VeriNet	70.4%	54%	15.6%	100%
CROWN	70.4%	54%	15.6%	100%

Table: Proportion of networks where our bounds are tighter or equal to others

Small random networks - computation time

Method	**ReLU**	**TanH**	**ELU**	**SiLU**
IBP	12	18	11	(13)
ReluVal	29	-	-	-
Neurify	27	-	-	-
VeriNet	14	33	(25)	-
CROWN	199	213	(177)	-
Mixed-Monotonicity	591	462	550	543

Table: Average computation time (per neuron in the network) in μs

Small random networks - width

Method	ReLU	TanH	ELU	SiLU
IBP	100%	100%	100%	(100%)
ReluVal	68%	-	-	-
Neurify	46%	-	-	-
VeriNet	43%	32%	(40%)	-
CROWN	43%	31%	(38%)	-

Table: Proportion of networks where our bounds are tighter or equal to others

Method	ReLU	TanH	ELU	SiLU
IBP	73%	71%	79%	(79%)
ReluVal	36%	-	-	-
Neurify	16%	-	-	-
VeriNet	12%	11%	(19%)	-
CROWN	12%	10%	(17%)	-

Table: Proportion of networks where our bounds are strictly tighter than others

Large random networks - computation time

Method	**ReLU**	**TanH**	**ELU**	**SiLU**
IBP	0.016	0.018	0.016	(0.018)
ReluVal	44	-	-	-
Neurify	44	-	-	-
VeriNet	0.034	0.05	(0.037)	-
CROWN	4.8	4.8	(4.8)	-
Mixed-Monotonicity	33	29	34	33

Table: Average computation time (per neuron in the network) in *ms*

Large random networks - width

Method	ReLU	TanH	ELU	SiLU
IBP	100%	100%	100%	(100%)
ReluVal	100%	-	-	-
Neurify	2%	-	-	-
VeriNet	0.1%	100%	(98%)	-
CROWN	0.1%	100%	(100%)	-

Table: Proportion of networks where our bounds are tighter or equal to others

Method	ReLU	TanH	ELU	SiLU
IBP	0%	0%	0%	(0%)
ReluVal	0%	-	-	-
Neurify	0%	-	-	-
VeriNet	0%	0%	(79%)	-
CROWN	0%	2%	(100%)	-

Table: Proportion of networks where our bounds are strictly tighter than others