Star Set based Reachability Analysis of Neural Networks with differing Layers and Activation Functions

Hana Masara
Supervision: Prof. Dr. Erika Ábrahám, László Antal

LuFG Theory of Hybrid Systems

03.07.2023

RWTH AACHEN UNIVERSITY

Outline

Outline

Feedforward Neural Network

Feedforward Neural Network

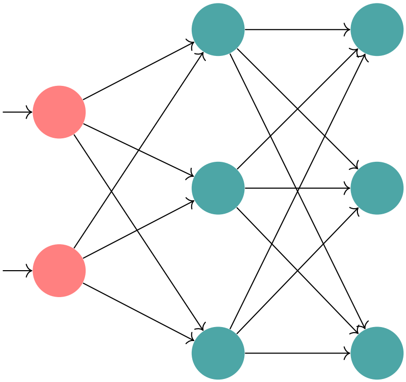
- FFN transmits information forward through:

Feedforward Neural Network

- FFN transmits information forward through:
  - Input layer

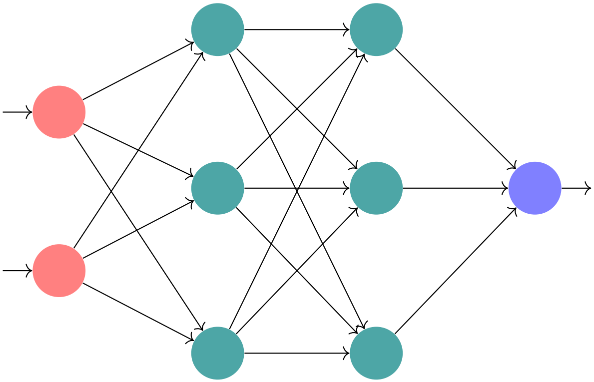
Feedforward Neural Network

- FFN transmits information forward through:
  - Input layer
  - One or multiple hidden layers



Feedforward Neural Network

- FFN transmits information forward through:
  - Input layer
  - One or multiple hidden layers
  - Output layer



Feedforward Neural Network

- Given an input vector x

Feedforward Neural Network

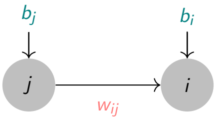
- Given an input vector x
- Three components define the output:

Feedforward Neural Network

- Given an input vector x
- Three components define the output:
  - The weight matrix W^k between two layers $k - 1$ and k

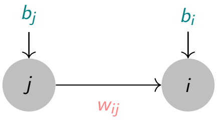
Feedforward Neural Network

- Given an input vector x
- Three components define the output:
  - The weight matrix W^k between two layers $k-1$ and k
  - The bias vectors b^k of the k^{th} layer



Feedforward Neural Network

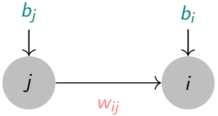
- Given an input vector x
- Three components define the output:
  - The weight matrix W^k between two layers $k-1$ and k
  - The bias vectors b^k of the k^{th} layer



  - The activation function f

Feedforward Neural Network

- Given an input vector x
- Three components define the output:
  - The weight matrix W^k between two layers $k-1$ and k
  - The bias vectors b^k of the k^{th} layer



  - The activation function f
  - Overall y_i give the output of a neuron i by:

$$y_i = f(b_i + \sum_{j=1}^{n} w_{ij} x_j)$$

Star Set Representation

- A star set $\langle c, V, P \rangle$ where:

Star Set Representation

- A star set $\langle c, V, P \rangle$ where:
  - $c \in \mathbb{R}^n$ the center

Star Set Representation

- A star set $\langle c, V, P \rangle$ where:
  - $c \in \mathbb{R}^n$ the center
  - $V = \{v_1, \ldots, v_m\} \subseteq \mathbb{R}^n$ the basis vertors

Star Set Representation

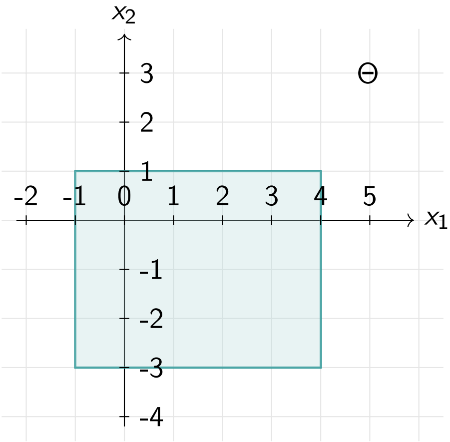
- A star set $\langle c, V, P \rangle$ where:
  - $c \in \mathbb{R}^n$ the center
  - $V = \{v_1, \ldots, v_m\} \subseteq \mathbb{R}^n$ the basis vertors
  - $P : \mathbb{R}^m \to \{\perp, \top\}$ a predicate of $P(\alpha) \triangleq C\alpha \leq d$

Star Set Representation

- A star set $\langle c, V, P\rangle$ where:
  - $c \in \mathbb{R}^n$ the center
  - $V = \{v_1, \ldots, v_m\} \subseteq \mathbb{R}^n$ the basis vertors
  - $P : \mathbb{R}^m \rightarrow \{\perp, \top\}$ a predicate of $P(\alpha) \triangleq C\alpha \leq d$
- The set of states:

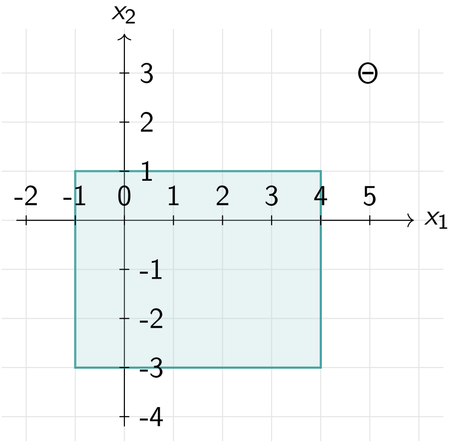
$$[\![\Theta]\!] = \{x \mid x = c + \sum_{i=1}^{m}(\alpha_i v_i) \text{ such that } P(\alpha) = \top)\}$$

Affine Mapping of a Star

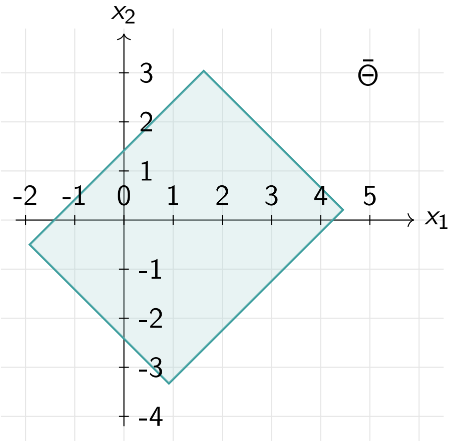


$\langle c, V, P \rangle$

Affine Mapping of a Star



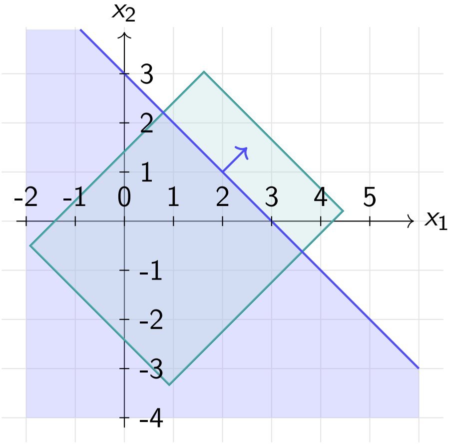
$\langle c, V, P \rangle$



$\langle \bar{c}, \bar{V}, \bar{P} \rangle$

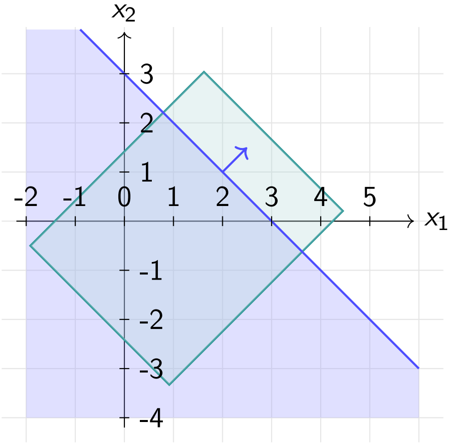
$$\bar{c} = Wc + b, \ \ \bar{V} = \{Wv_1, \dots, Wv_m\}, \ \ \bar{P} \equiv P$$

Star and Half-space Intersection

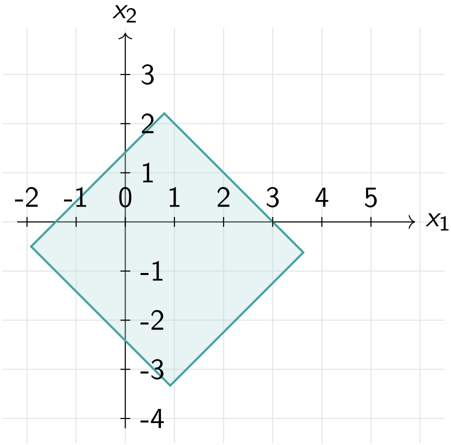


$\Theta = \langle c, V, P \rangle$ and $\mathscr{H} \triangleq \{x \mid Hx \leq g\}$

Star and Half-space Intersection



$\Theta = \langle c, V, P \rangle$ and $\mathscr{H} \triangleq \{x \mid Hx \leq g\}$

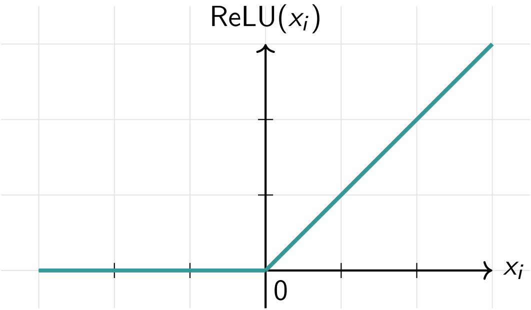


$\bar{\Theta} = \Theta \cap \mathscr{H} = \langle c, V, \bar{P} \rangle$ with $\bar{P} = P \wedge P'$, where
$P'(\alpha) \triangleq (H \times V)\alpha \leq g - H \times c, \;\; V = [v_1, \ldots, v_m]$

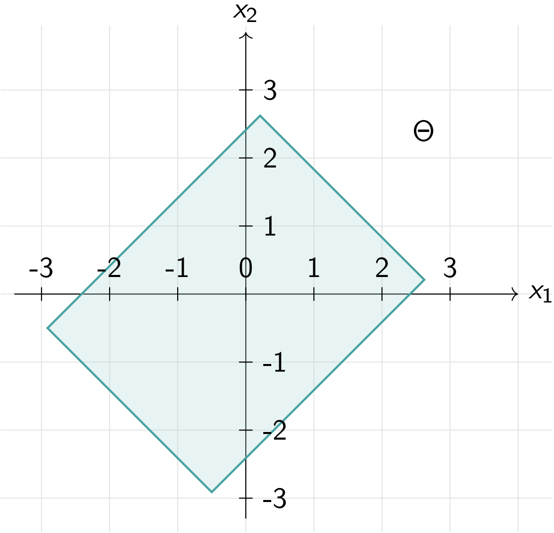
Reachability Analysis of FNNs with ReLU

- Given the input x,

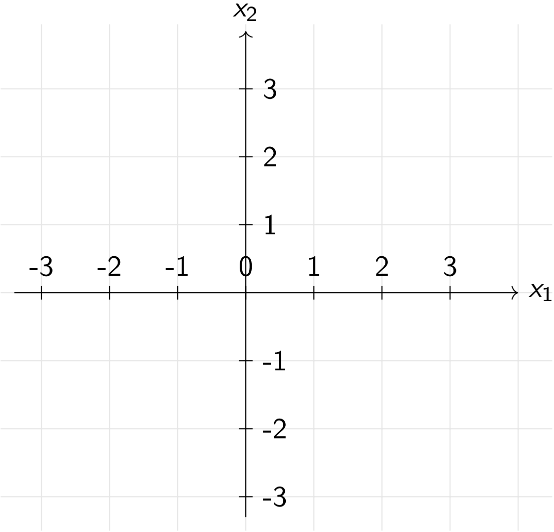
$$ReLU(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} = max(0, x)$$



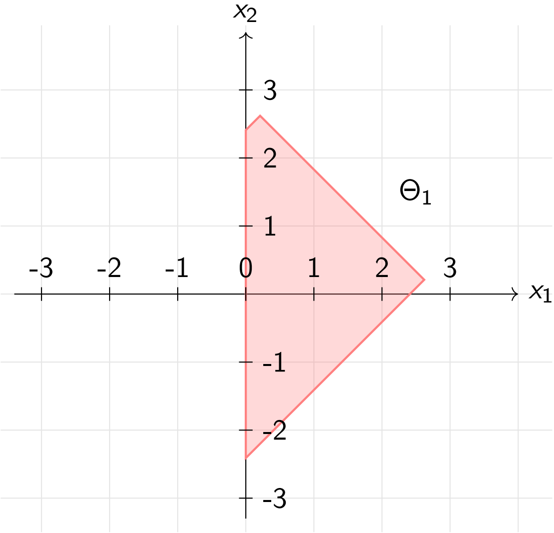
Exact Analysis



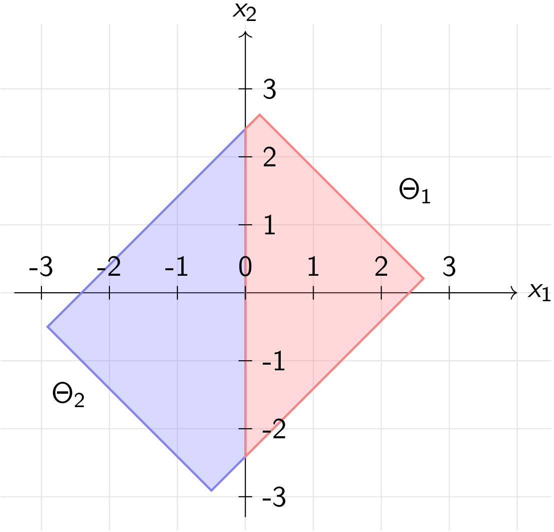
Exact Analysis



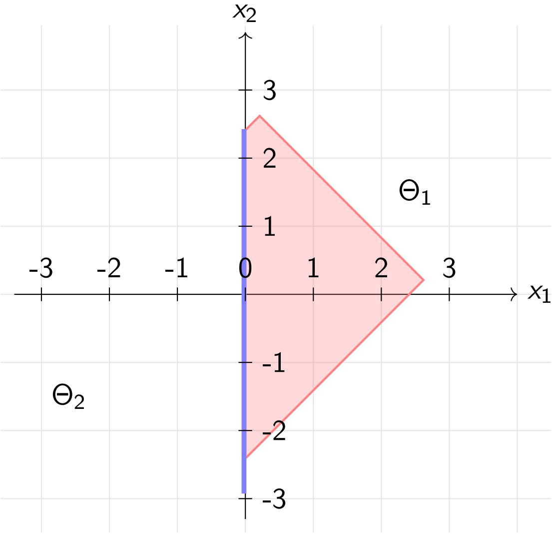
Exact Analysis



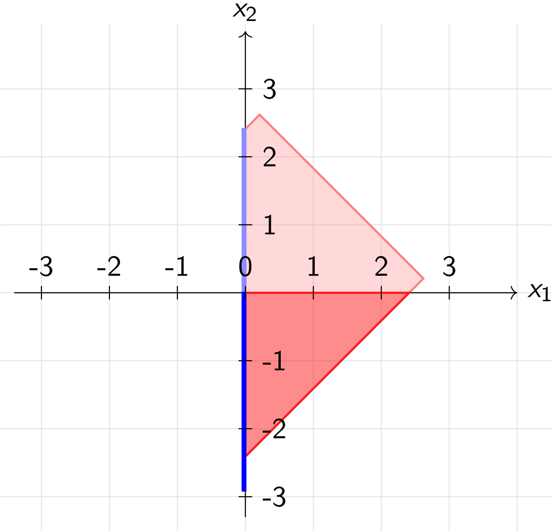
Exact Analysis



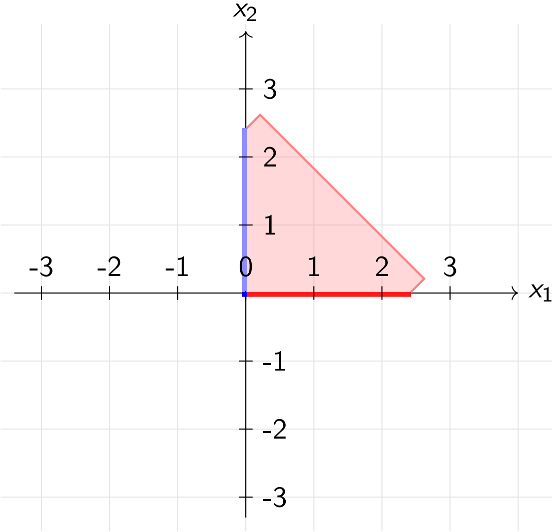
Exact Analysis



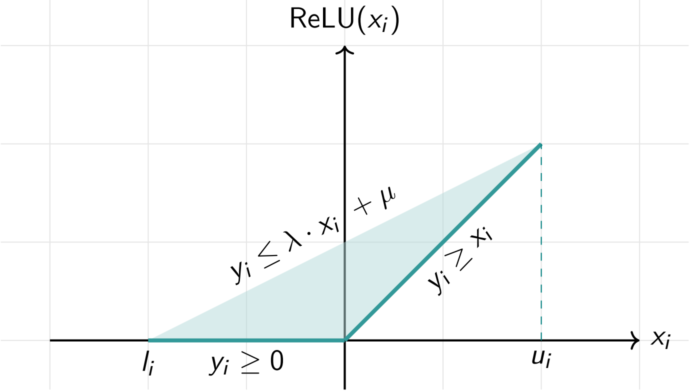
Exact Analysis



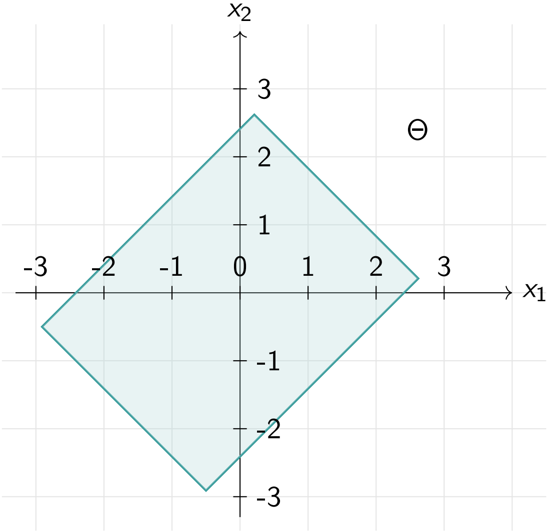
Exact Analysis



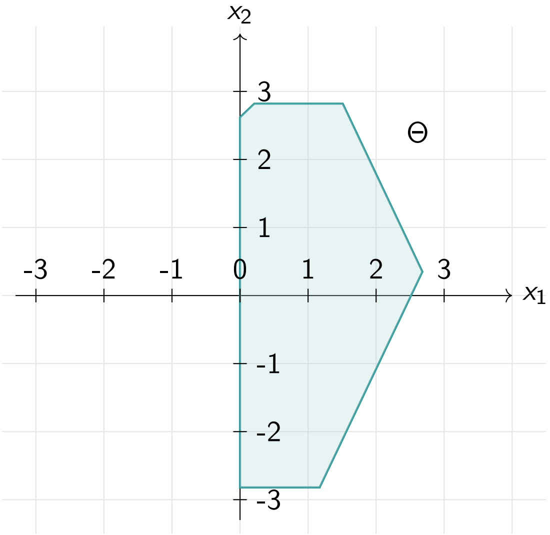
Over-approximate Analysis Convex Relaxation



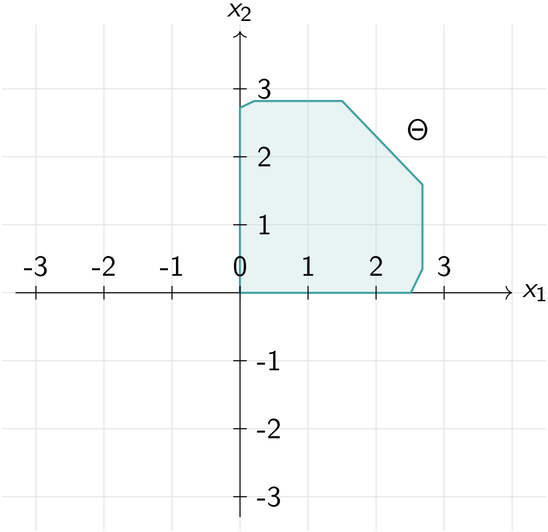
Over-approximate Analysis



Over-approximate Analysis



Over-approximate Analysis



Outline

Motivation

Neural networks process and learn from complex, non-linear datasets

$\Rightarrow$ Non-linear activation functions to prevent linearity

$\Rightarrow$ Verifying neural networks to ensure their reliability, robustness, and safety

Methodology and Implementation

We investigate star set based reachability analysis of . . .

Methodology and Implementation

We investigate star set based reachability analysis of ...

- Leaky ReLU Layer

Methodology and Implementation

We investigate star set based reachability analysis of . . .

- Leaky ReLU Layer
- Hard Tanh Layer

Methodology and Implementation

We investigate star set based reachability analysis of . . .

- Leaky ReLU Layer
- Hard Tanh Layer
- Hard Sigmoid Layer

Methodology and Implementation

We investigate star set based reachability analysis of ...

- Leaky ReLU Layer
- Hard Tanh Layer
- Hard Sigmoid Layer
- Unit Step Function Layer

Methodology and Implementation

We investigate star set based reachability analysis of . . .

- Leaky ReLU Layer
- Hard Tanh Layer
- Hard Sigmoid Layer
- Unit Step Function Layer

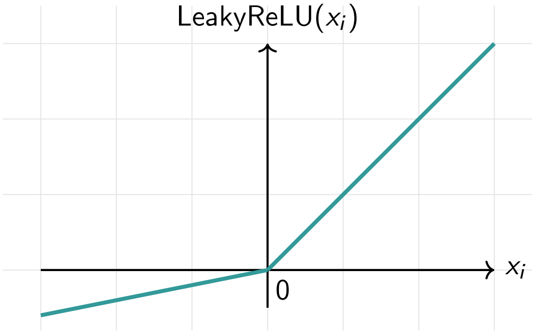
for exact and over-approximation on bounded/unbounded sets.

Leaky ReLU Function

- Given the input x,

$$LeakyReLU(x) = \begin{cases} x, & x > 0 \\ \gamma \cdot x, & x \leq 0 \end{cases} = max(\gamma \cdot x, x)$$

where $\gamma \in (0, 1)$

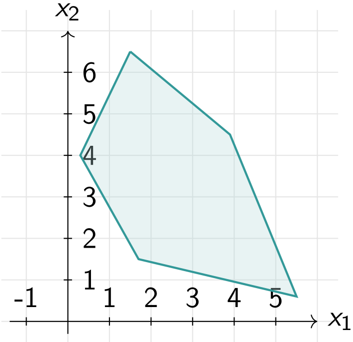


Exact Analysis

Case 1 | Case 2 | Case 3

Exact Analysis

Case 1

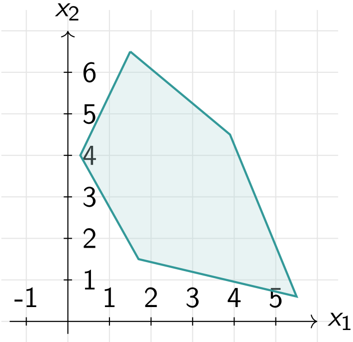


Case 2

Case 3

Exact Analysis

Case 1

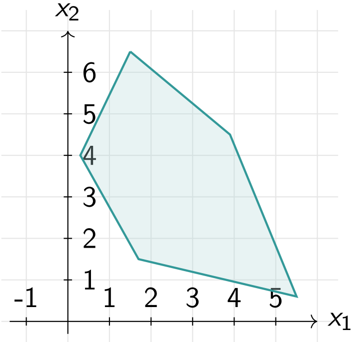


Case 2

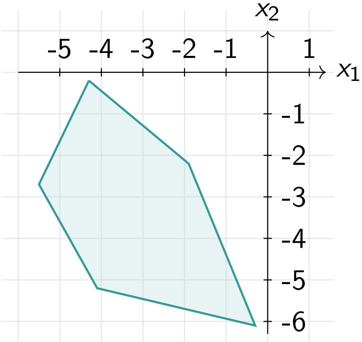
Case 3

Exact Analysis

Case 1



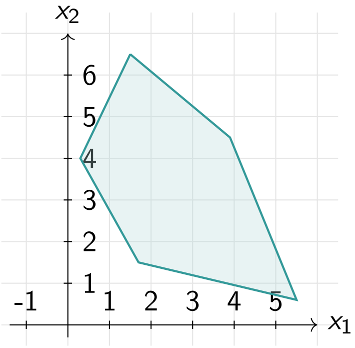
Case 2



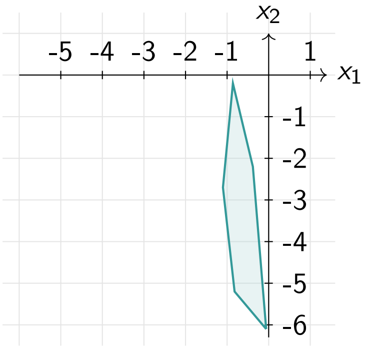
Case 3

Exact Analysis

Case 1



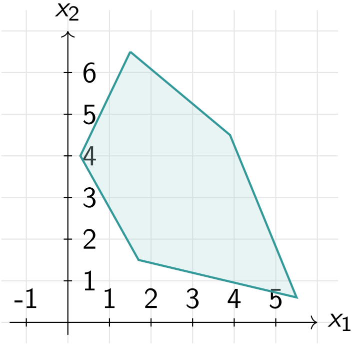
Case 2



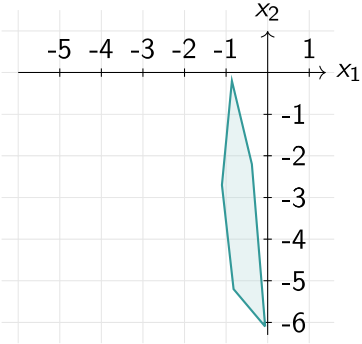
Case 3

Exact Analysis

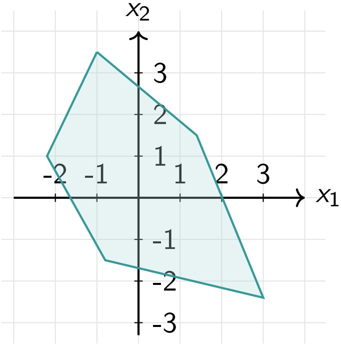
Case 1



Case 2

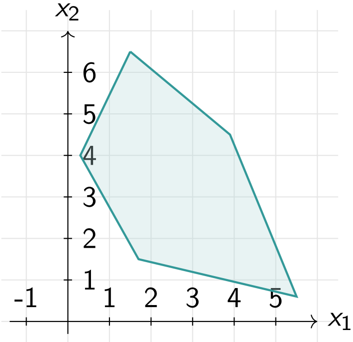


Case 3

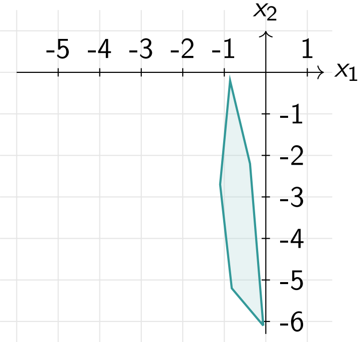


Exact Analysis

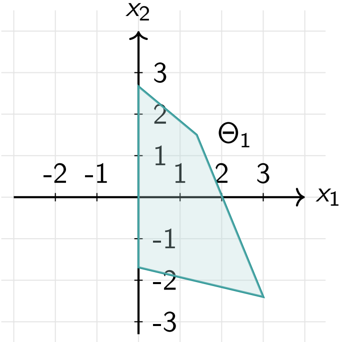
Case 1



Case 2

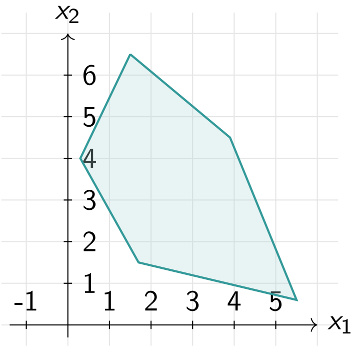


Case 3

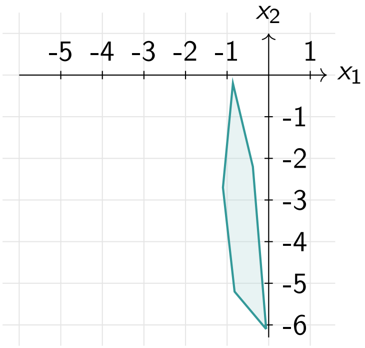


Exact Analysis

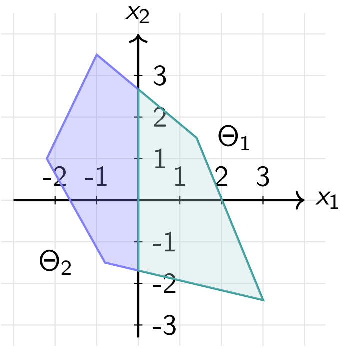
Case 1



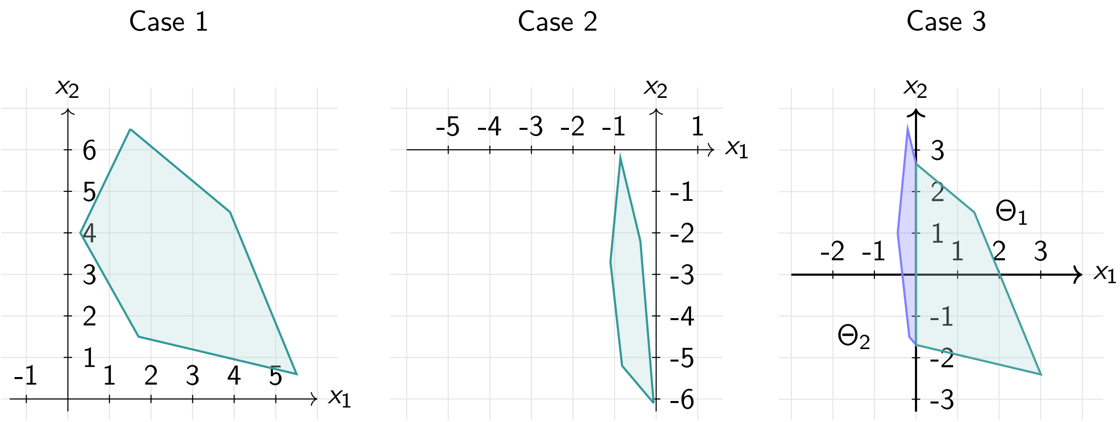
Case 2



Case 3

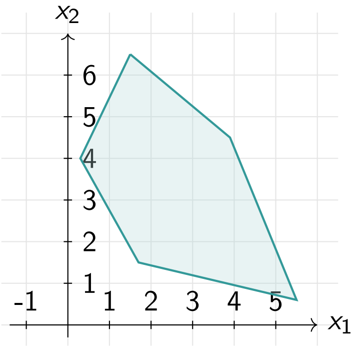


Exact Analysis

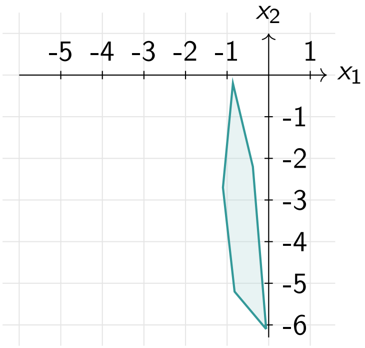


Exact Analysis

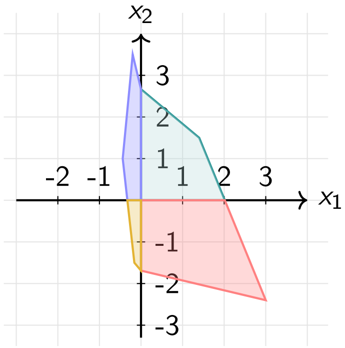
Case 1



Case 2

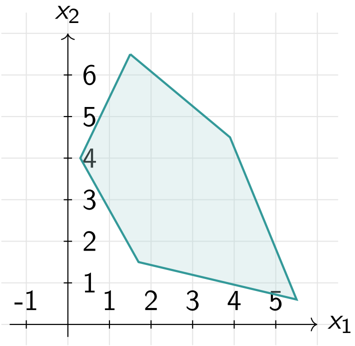


Case 3

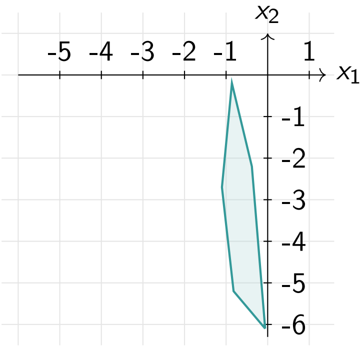


Exact Analysis

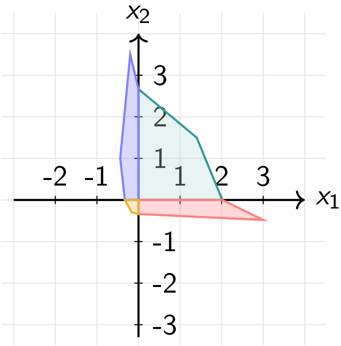
Case 1



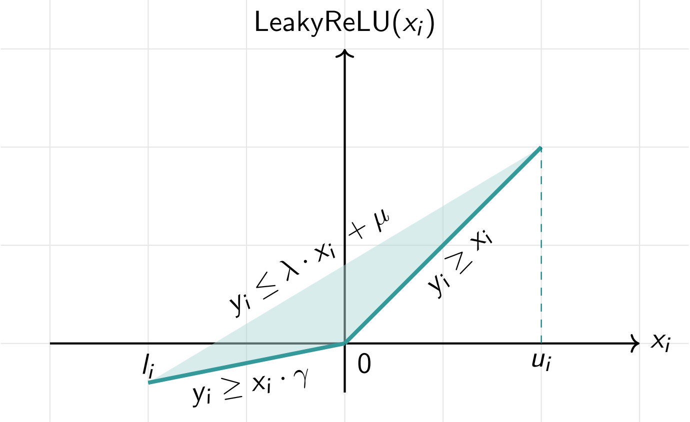
Case 2



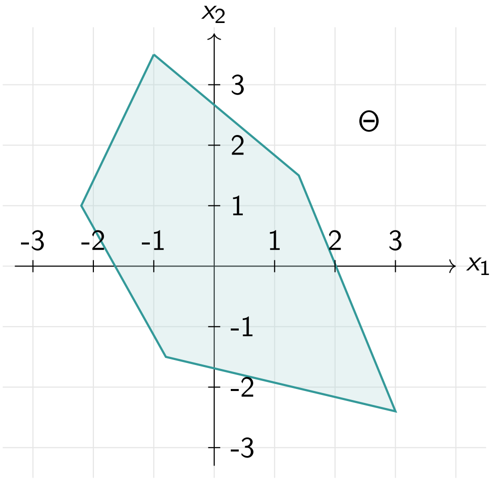
Case 3



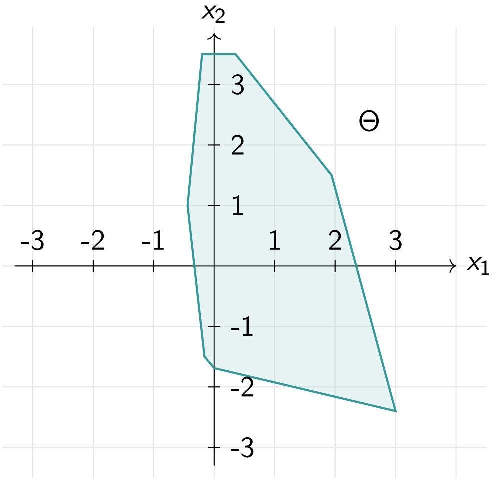
Over-approximate Analysis Convex Relaxation



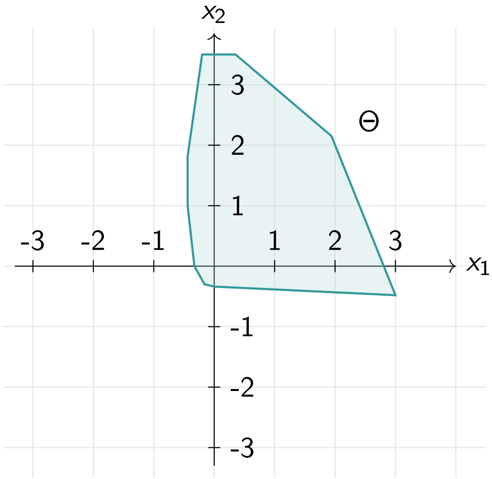
Over-approximate Analysis



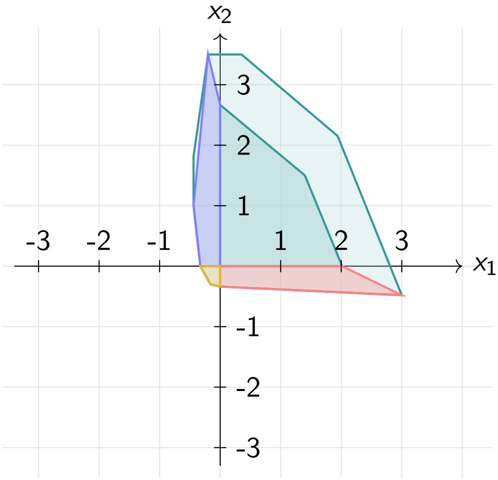
Over-approximate Analysis



Over-approximate Analysis



Comparison

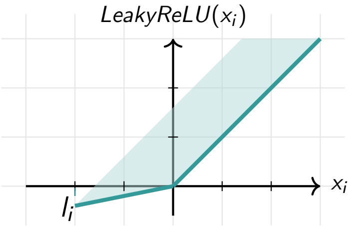


Unbounded Analysis

Case 1 | Case 2 | Case 3

Unbounded Analysis

No upper bound

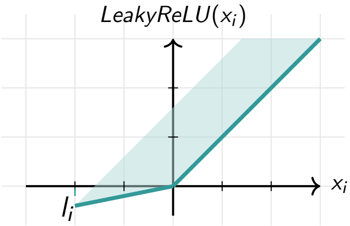


Case 2

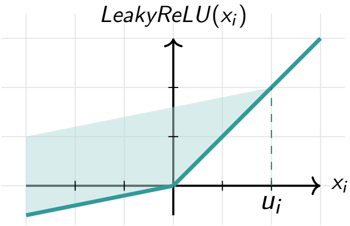
Case 3

Unbounded Analysis

No upper bound



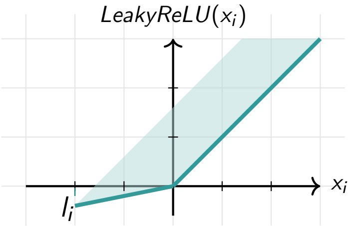
No lower bound



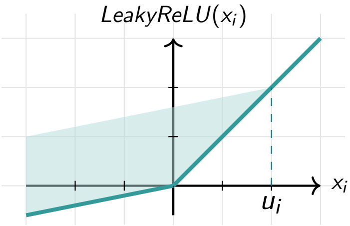
Case 3

Unbounded Analysis

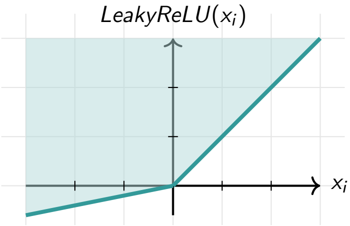
No upper bound



No lower bound



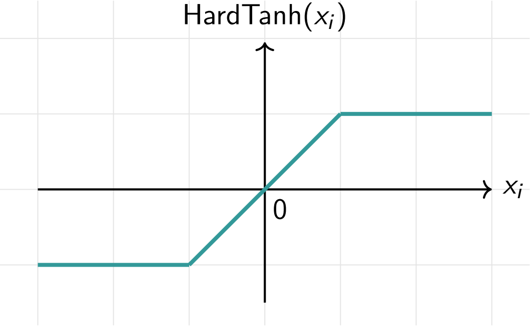
No bounds



Hard Tanh Function

- For the input x,

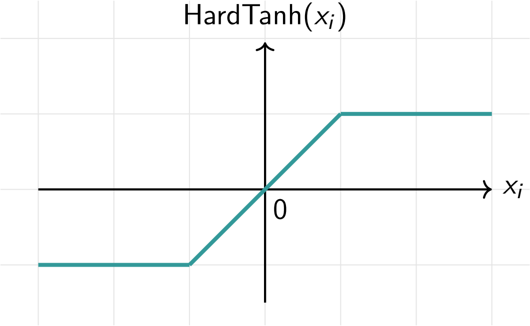
$$HardTanh(x) = \begin{cases} -1 & x < -1 \\ 1 & x > 1 \\ x & -1 \leq x \leq 1 \end{cases}$$



Hard Tanh Function

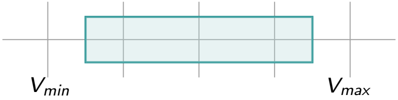
- For the input x,

$$HardTanh(x) = \begin{cases} V_{min}, & x < V_{min} \\ V_{max}, & x > V_{max} \\ x, & V_{min} \leq x \leq V_{max} \end{cases}$$



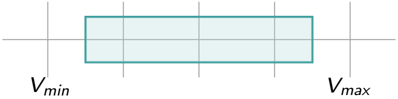
Exact Analysis

Exact Analysis

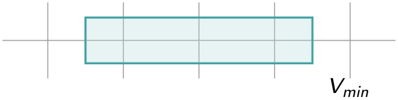


$\Rightarrow$ Remain the same

Exact Analysis

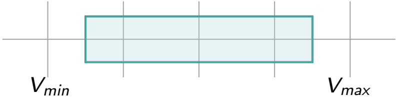


$\Rightarrow$ Remain the same

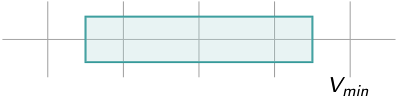


$\Rightarrow$ Project onto V_{min}

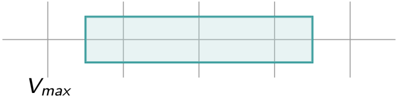
Exact Analysis



$\Rightarrow$ Remain the same

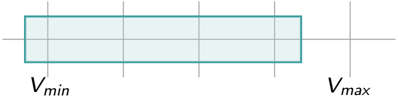


$\Rightarrow$ Project onto V_{min}



$\Rightarrow$ Project onto V_{max}

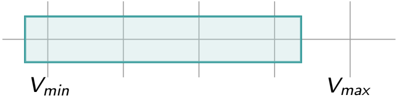
Exact Analysis



$\Rightarrow$ Decompose into two subsets

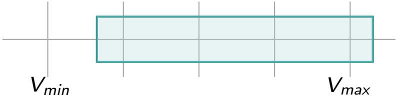
$$\Theta_1 = \Theta \wedge (V_{min} \leq x_i \leq V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i < V_{min})$$

Exact Analysis



$\Rightarrow$ Decompose into two subsets

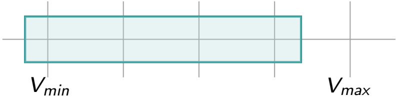
$$\Theta_1 = \Theta \wedge (V_{min} \leq x_i \leq V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i < V_{min})$$



$\Rightarrow$ Decompose into two subsets

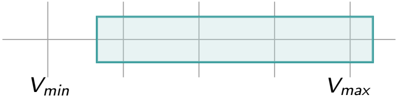
$$\Theta_1 = \Theta \wedge (V_{min} \leq x_i \leq V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i > V_{max})$$

Exact Analysis



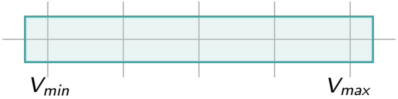
$\Rightarrow$ Decompose into two subsets

$$\Theta_1 = \Theta \wedge (V_{min} \leq x_i \leq V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i < V_{min})$$



$\Rightarrow$ Decompose into two subsets

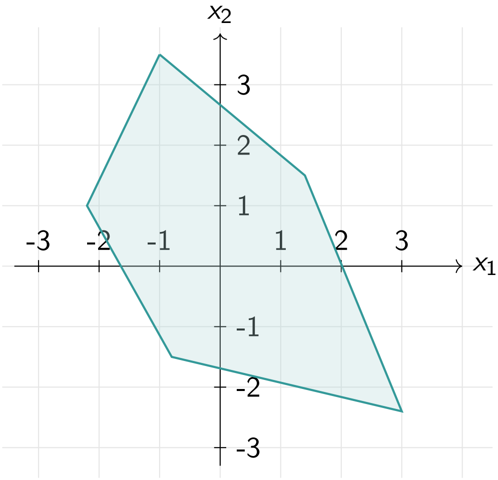
$$\Theta_1 = \Theta \wedge (V_{min} \leq x_i \leq V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i > V_{max})$$



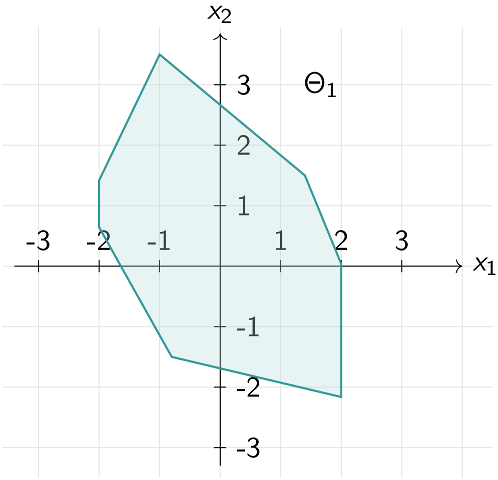
$\Rightarrow$ Decompose into three subsets

$$\Theta_1 = \Theta \wedge (V_{min} \leq x_i \leq V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i < V_{min})$$
$$\Theta_3 = \Theta \wedge (x_i > V_{max})$$

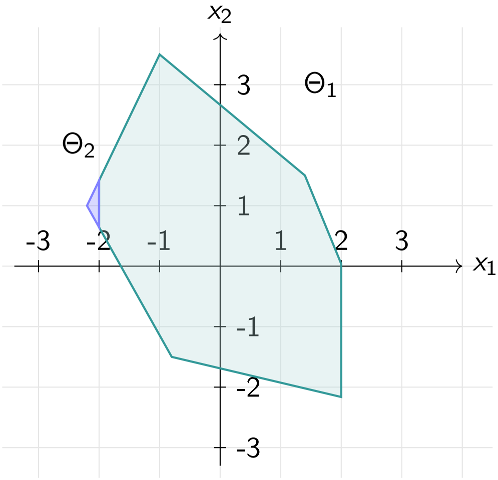
Exact Analysis



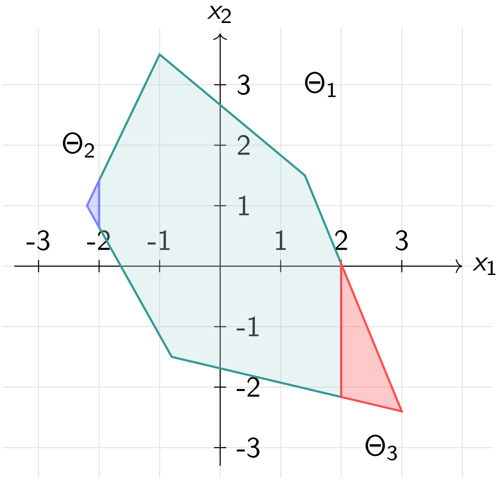
Exact Analysis



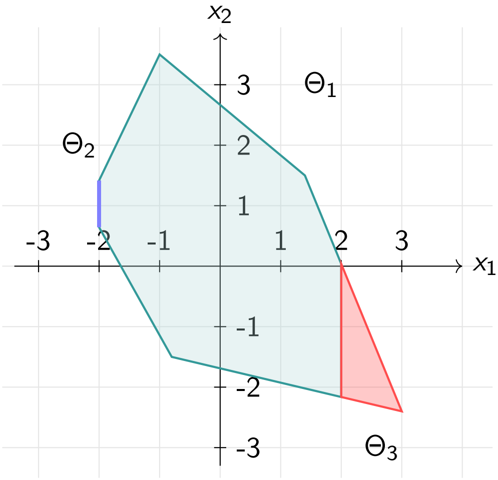
Exact Analysis



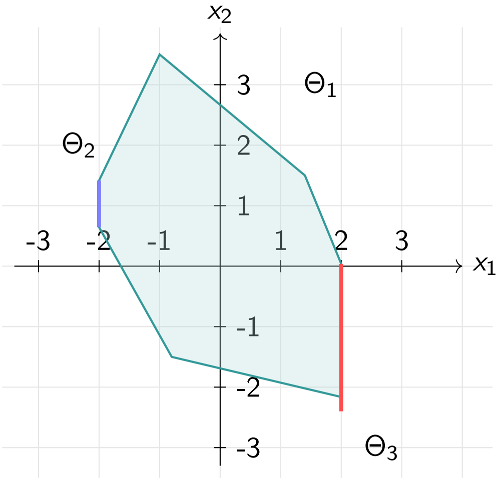
Exact Analysis



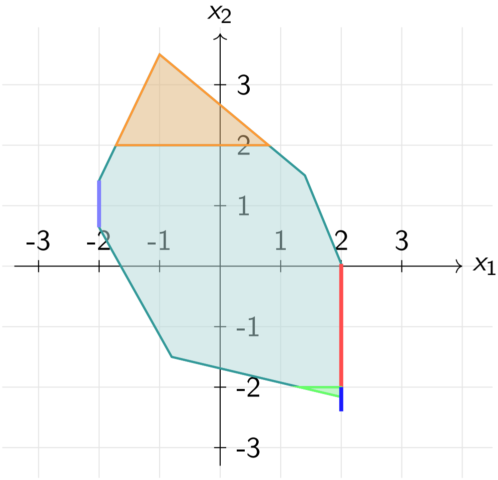
Exact Analysis



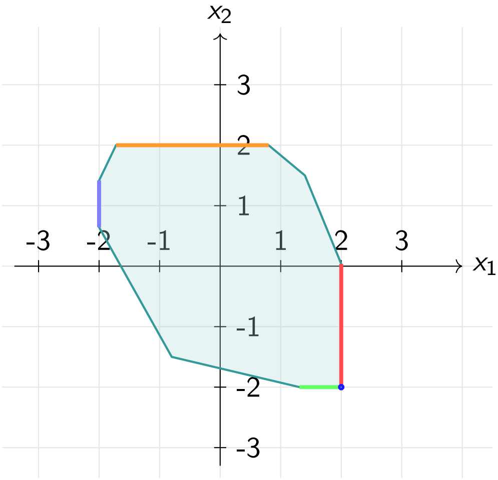
Exact Analysis



Exact Analysis



Exact Analysis



Over-approximate Analysis Convex Relaxation

Case 1, 2, and 3 equivalent to exact analysis

Over-approximate Analysis Convex Relaxation

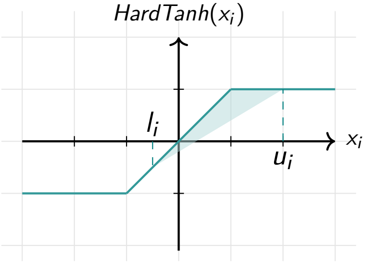
Case 4

Case 5

Case 6

Over-approximate Analysis Convex Relaxation

Case 4

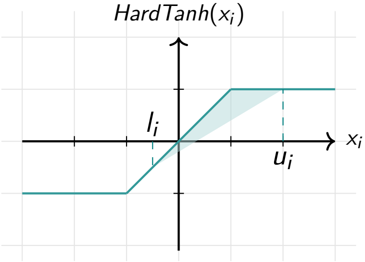


Case 5

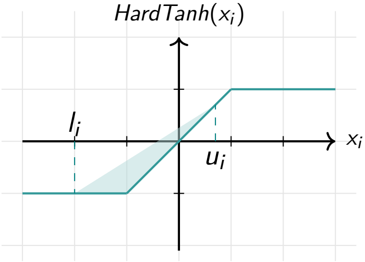
Case 6

Over-approximate Analysis Convex Relaxation

Case 4



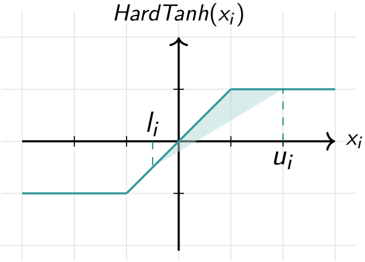
Case 5



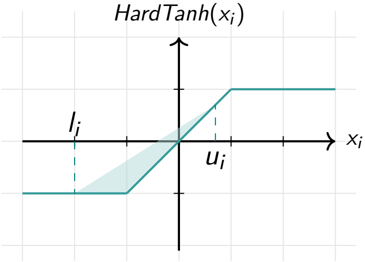
Case 6

Over-approximate Analysis Convex Relaxation

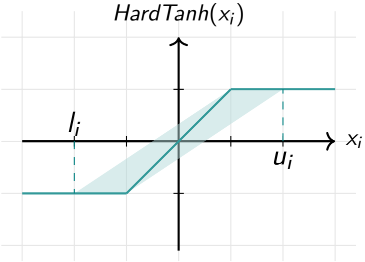
Case 4



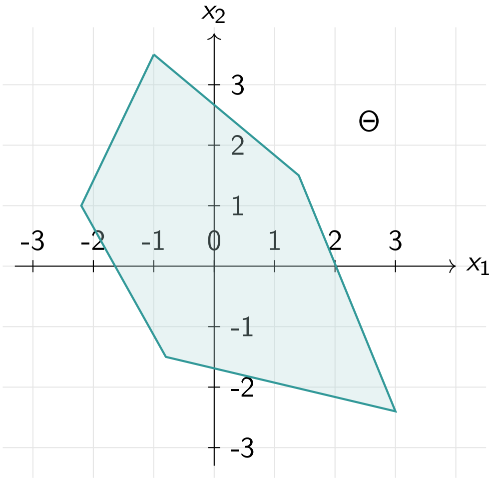
Case 5



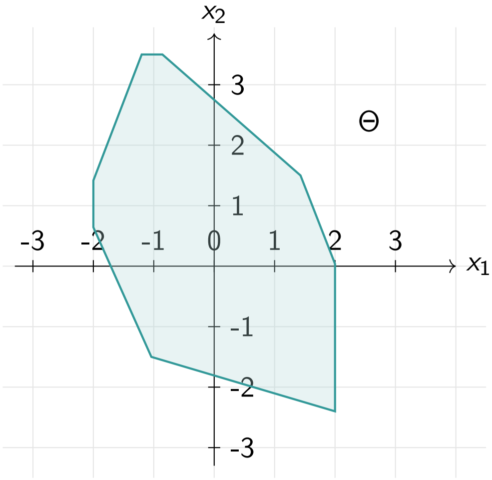
Case 6



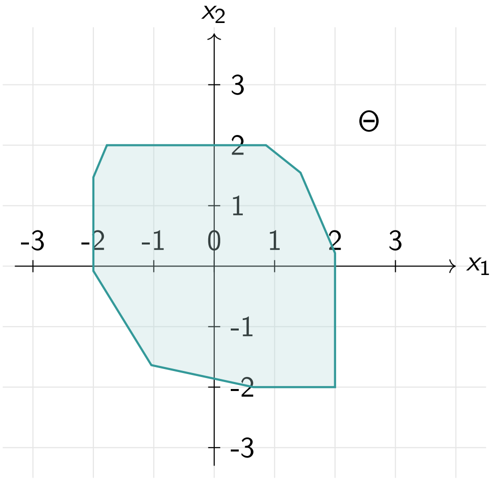
Over-approximate Analysis



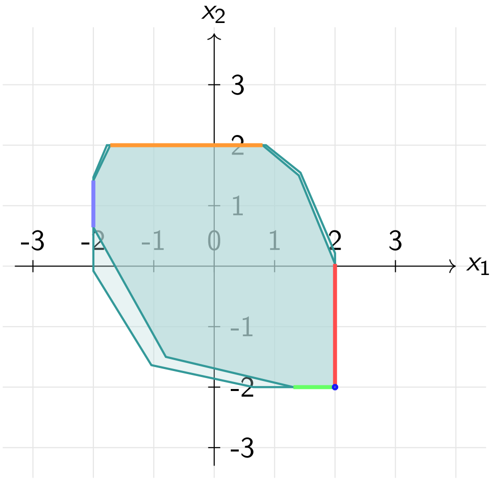
Over-approximate Analysis



Over-approximate Analysis



Comparison



Unbounded Analysis

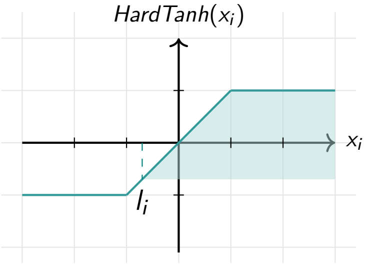
Case 1

Case 2

Case 3

Unbounded Analysis

No upper bound

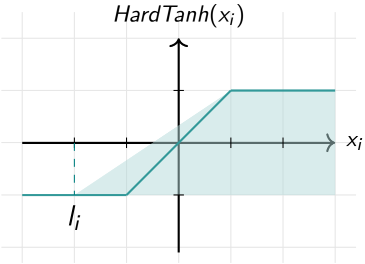


Case 2

Case 3

Unbounded Analysis

No upper bound

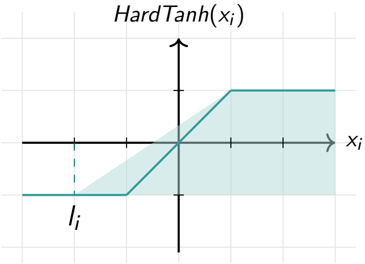


Case 2

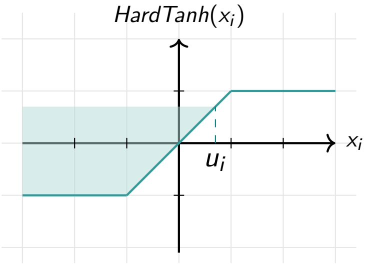
Case 3

Unbounded Analysis

No upper bound



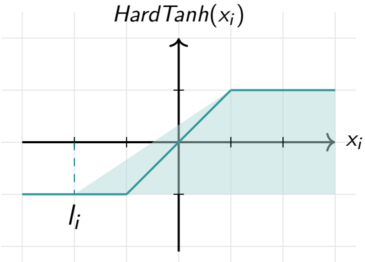
No lower bound



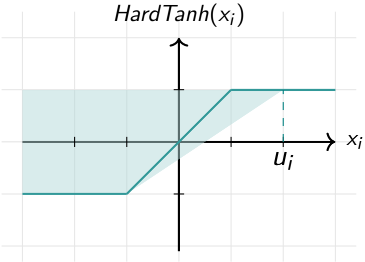
Case 3

Unbounded Analysis

No upper bound



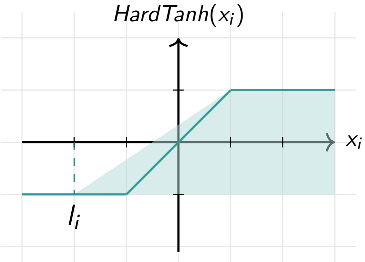
No lower bound



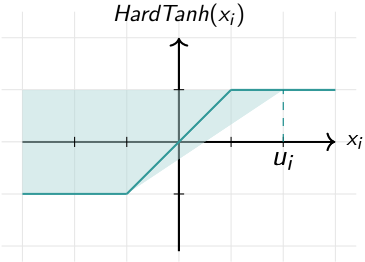
Case 3

Unbounded Analysis

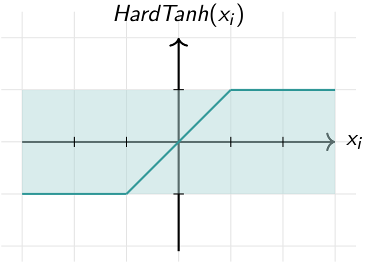
No upper bound



No lower bound



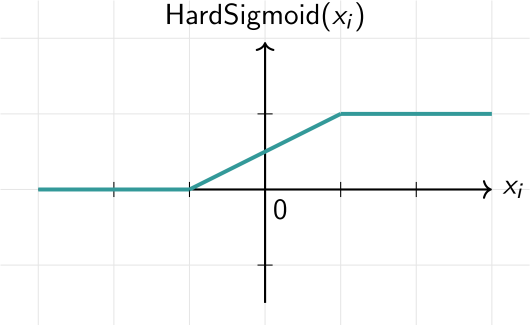
No bounds



Hard Sigmoid Function

- For the input x,

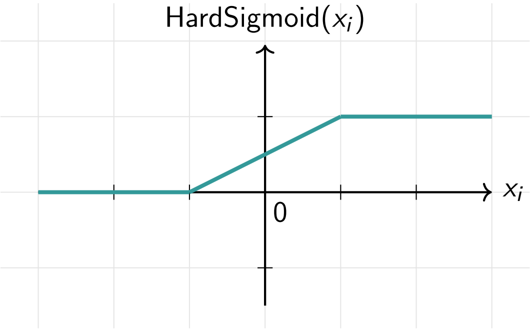
$$HardSigmoid(x) = \begin{cases} 0 & x \leq -1 \\ 1 & x \geq 1 \\ \frac{1}{2} \cdot x + \frac{1}{2} & -1 < x < 1 \end{cases} = max(0, min(\frac{1}{2} \cdot x + \frac{1}{2}))$$



Hard Sigmoid Function

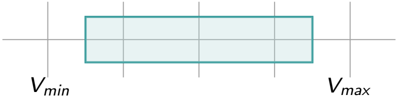
- For the input x,

$$HardSigmoid(x) = \begin{cases} 0 & x \leq V_{min} \\ 1 & x \geq V_{max} \\ \frac{1}{V_{max} - V_{min}} \cdot x + \frac{V_{min}}{V_{min} - V_{max}} & V_{min} < x < V_{max} \end{cases}$$



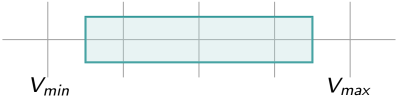
Exact Analysis

Exact Analysis

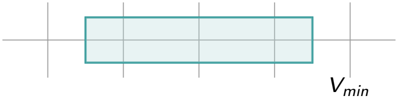


$\Rightarrow$ Correspond to the function

Exact Analysis

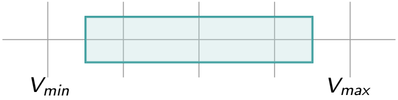


$\Rightarrow$ Correspond to the function

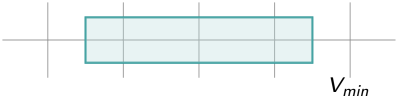


$\Rightarrow$ Project onto 0

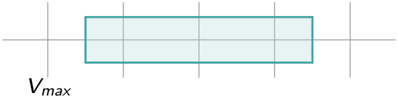
Exact Analysis



$\Rightarrow$ Correspond to the function

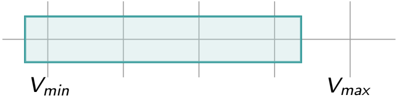


$\Rightarrow$ Project onto 0



$\Rightarrow$ Project onto 1

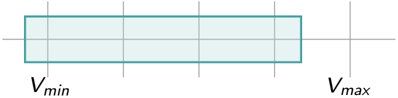
Exact Analysis



$\Rightarrow$ Decompose into two subsets

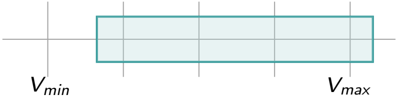
$$\Theta_1 = \Theta \wedge (V_{min} < x_i < V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i \leq V_{min})$$

Exact Analysis



$\Rightarrow$ Decompose into two subsets

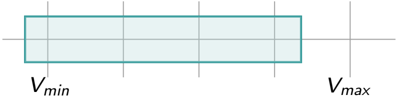
$$\Theta_1 = \Theta \wedge (V_{min} < x_i < V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i \leq V_{min})$$



$\Rightarrow$ Decompose into two subsets

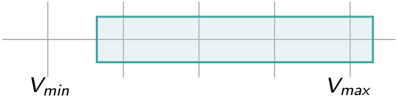
$$\Theta_1 = \Theta \wedge (V_{min} < x_i < V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i \geq V_{max})$$

Exact Analysis



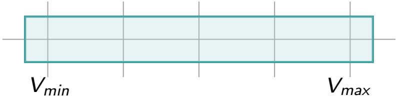
$\Rightarrow$ Decompose into two subsets

$$\Theta_1 = \Theta \wedge (V_{min} < x_i < V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i \leq V_{min})$$



$\Rightarrow$ Decompose into two subsets

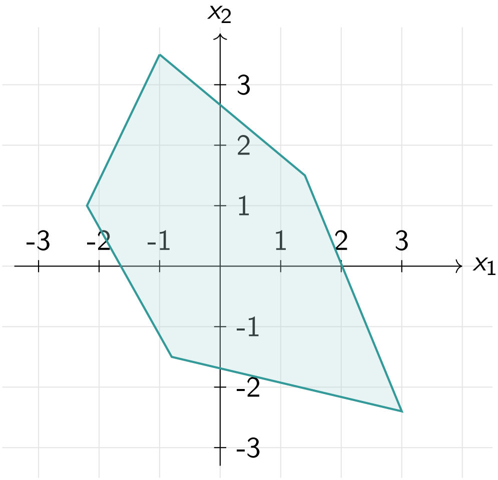
$$\Theta_1 = \Theta \wedge (V_{min} < x_i < V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i \geq V_{max})$$



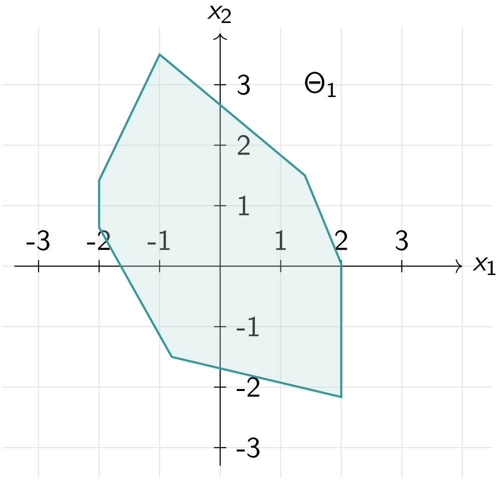
$\Rightarrow$ Decompose into three subsets

$$\Theta_1 = \Theta \wedge (V_{min} < x_i < V_{max})$$
$$\Theta_2 = \Theta \wedge (x_i \leq V_{min})$$
$$\Theta_3 = \Theta \wedge (x_i \geq V_{max})$$

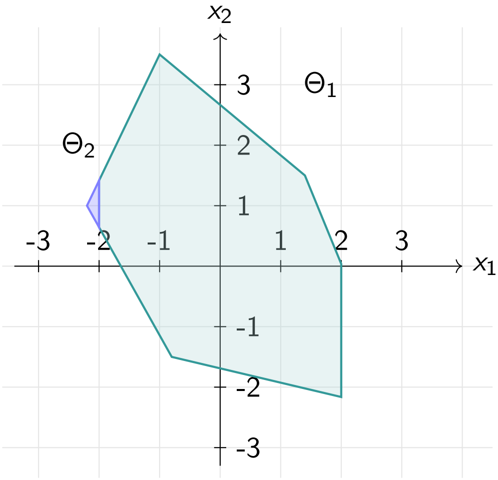
Exact Analysis



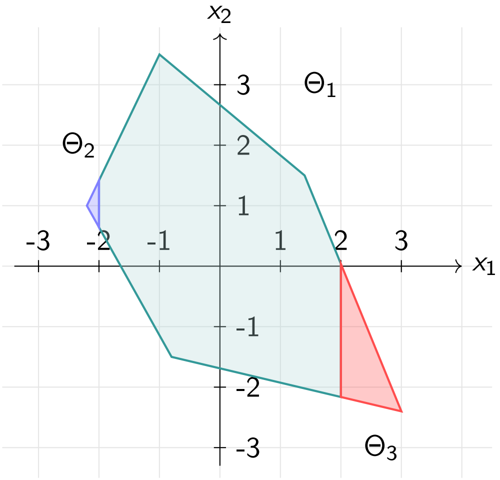
Exact Analysis



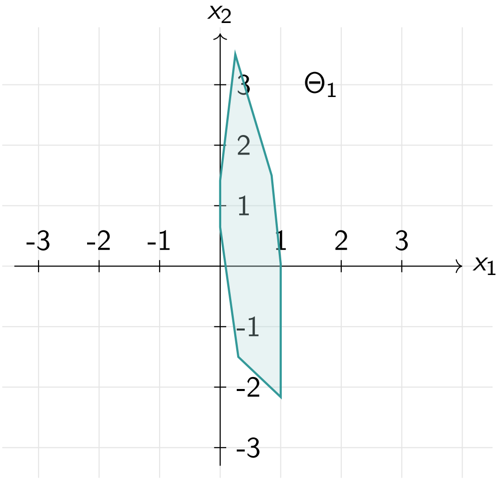
Exact Analysis



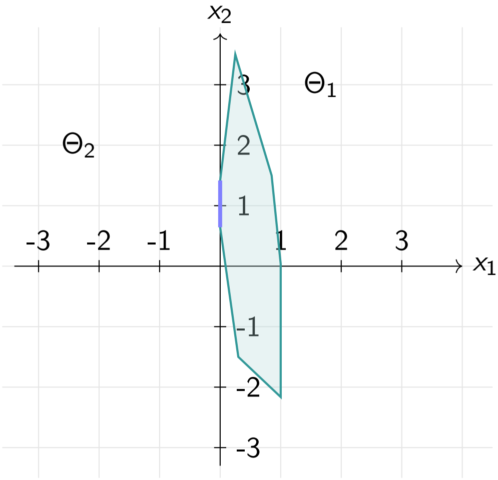
Exact Analysis



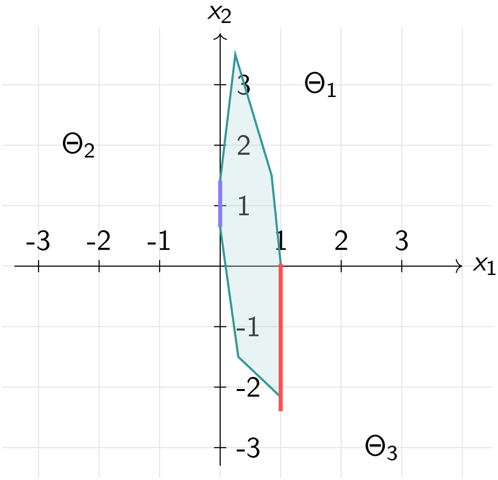
Exact Analysis



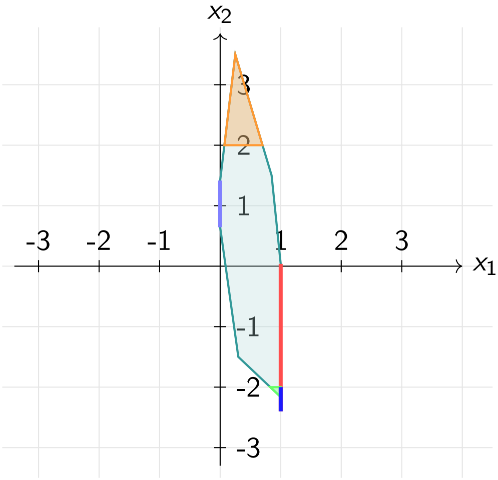
Exact Analysis



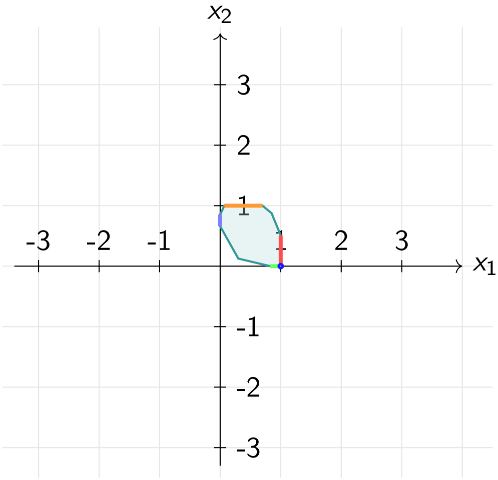
Exact Analysis



Exact Analysis



Exact Analysis

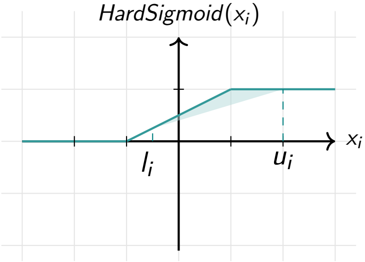


Over-approximate Analysis Convex Relaxation

Case 1 Case 2 Case 3

Over-approximate Analysis Convex Relaxation

Case 1

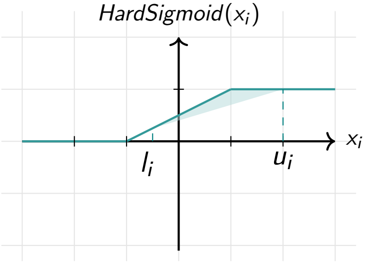


Case 2

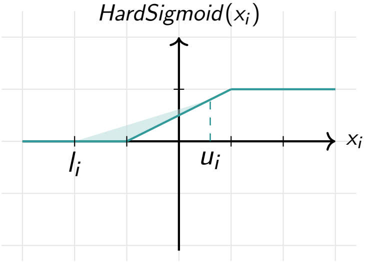
Case 3

Over-approximate Analysis Convex Relaxation

Case 1



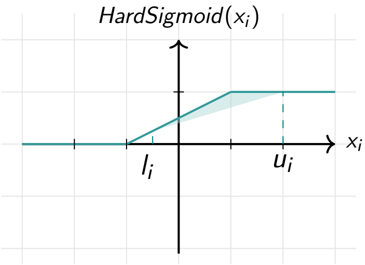
Case 2



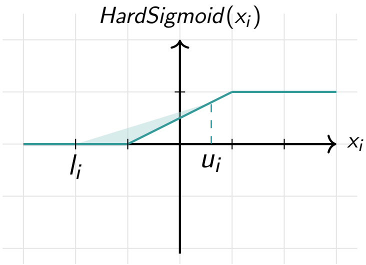
Case 3

Over-approximate Analysis Convex Relaxation

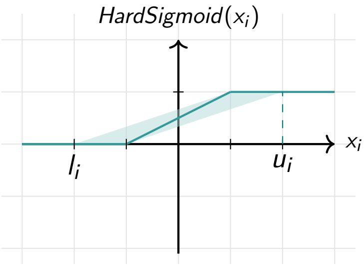
Case 1



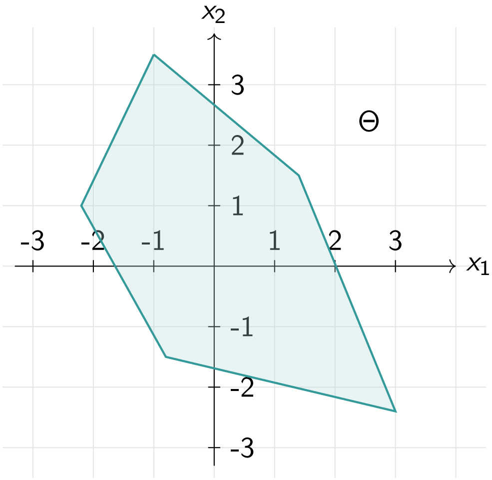
Case 2



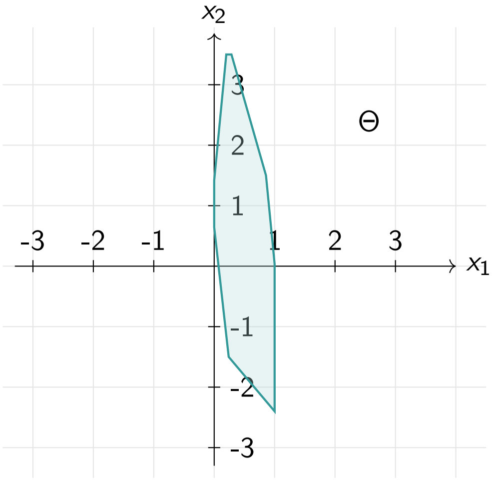
Case 3



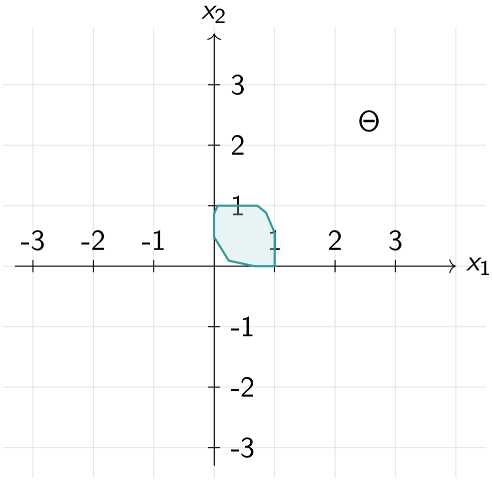
Over-approximate Analysis



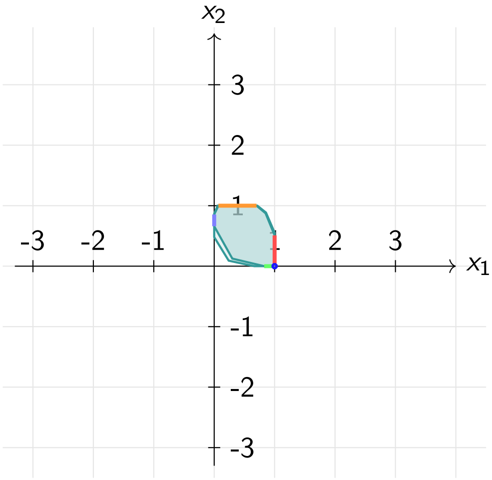
Over-approximate Analysis



Over-approximate Analysis



Comparison



Unbounded Analysis

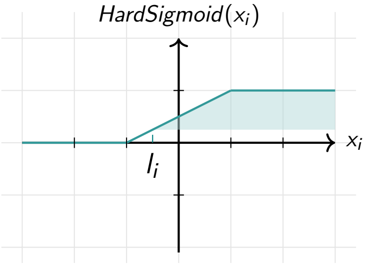
Case 1

Case 2

Case 3

Unbounded Analysis

No upper bound

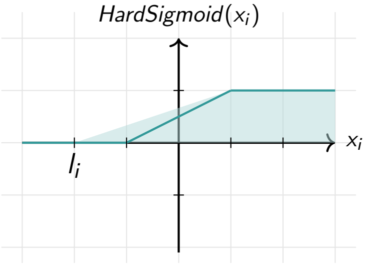


Case 2

Case 3

Unbounded Analysis

No upper bound

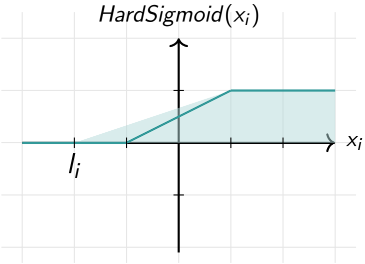


Case 2

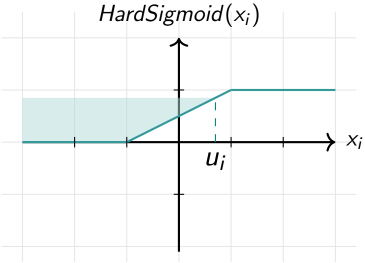
Case 3

Unbounded Analysis

No upper bound



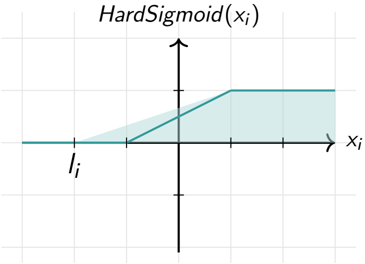
No lower bound



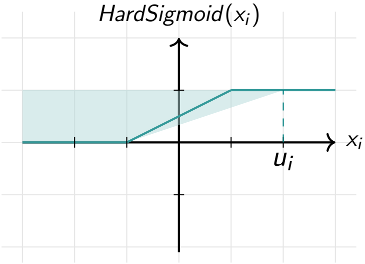
Case 3

Unbounded Analysis

No upper bound

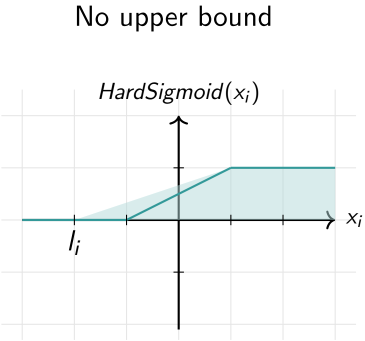


No lower bound



Case 3

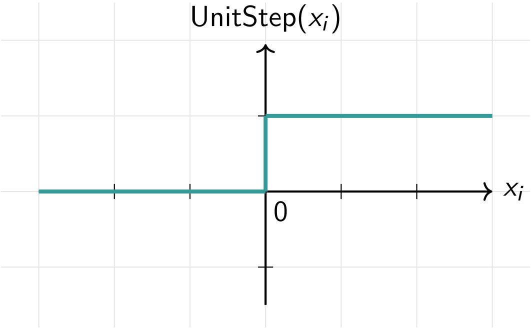
Unbounded Analysis



Unit step Function

- For the input x,

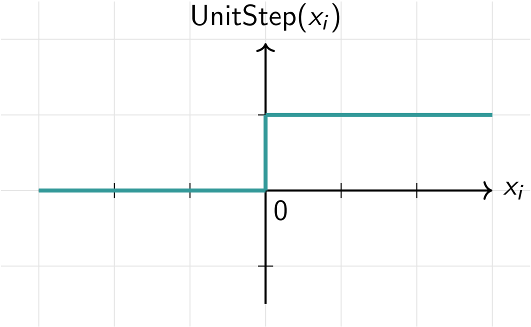
$$UnitStep(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$



Unit step Function

- For the input x,

$$unitStep(x) = \begin{cases} V_{min} & x < V \\ V_{max} & x \geq V \end{cases}$$



Exact Analysis

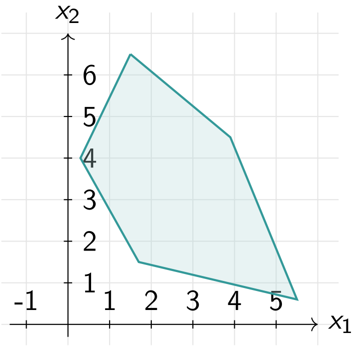
Case 1

Case 2

Case 3

Exact Analysis

Case 1

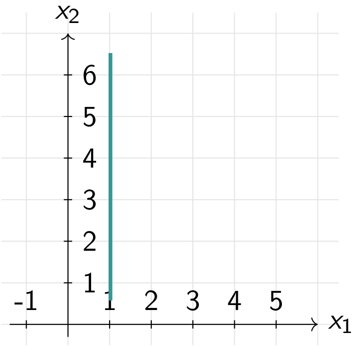


Case 2

Case 3

Exact Analysis

Case 1

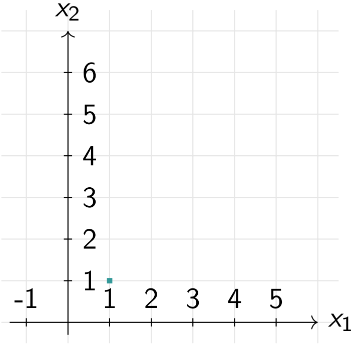


Case 2

Case 3

Exact Analysis

Case 1

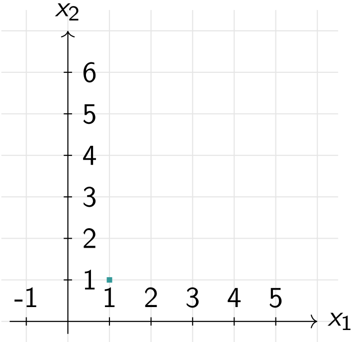


Case 2

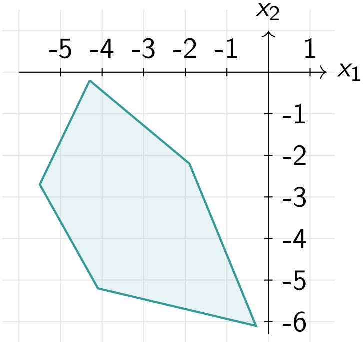
Case 3

Exact Analysis

Case 1



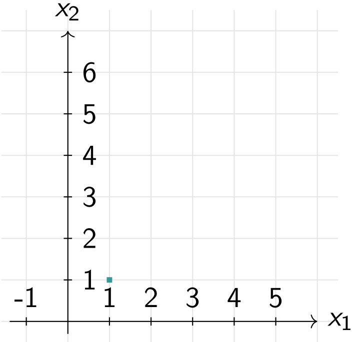
Case 2



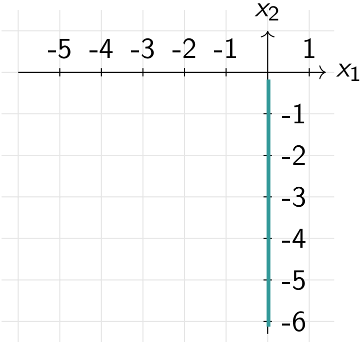
Case 3

Exact Analysis

Case 1



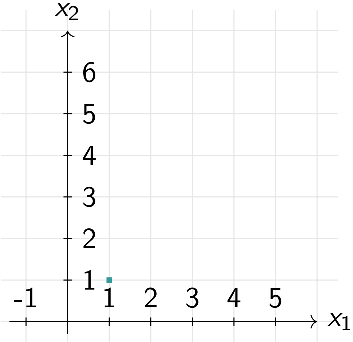
Case 2



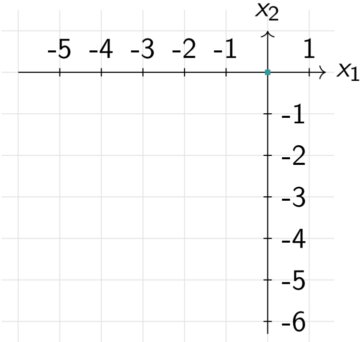
Case 3

Exact Analysis

Case 1



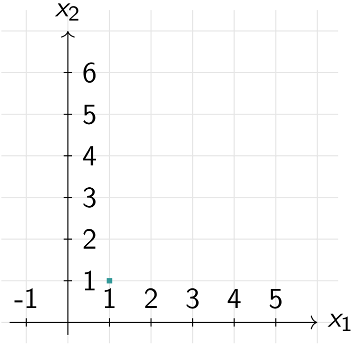
Case 2



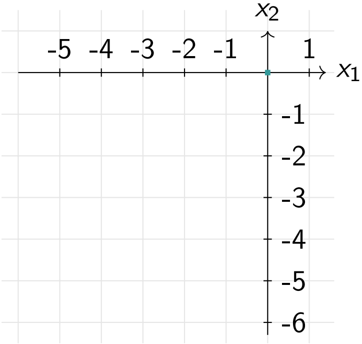
Case 3

Exact Analysis

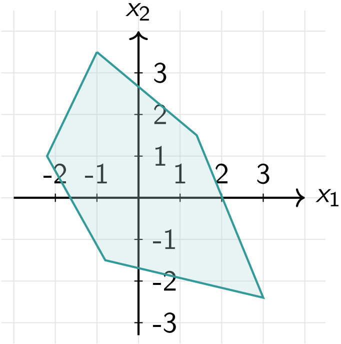
Case 1



Case 2

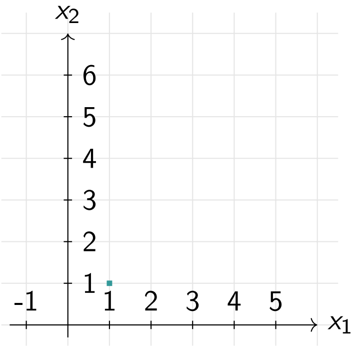


Case 3

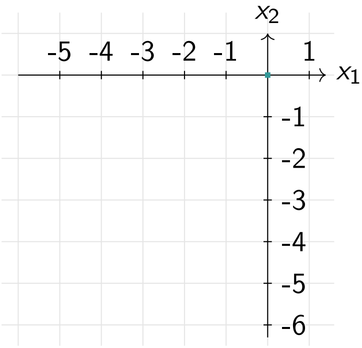


Exact Analysis

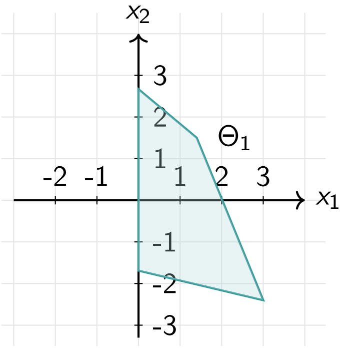
Case 1



Case 2

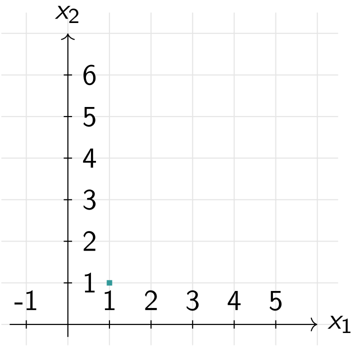


Case 3

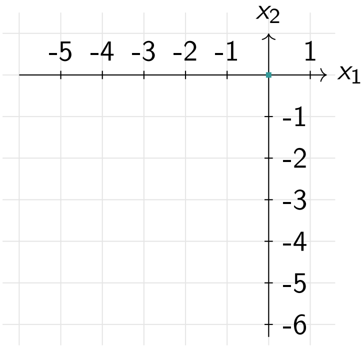


Exact Analysis

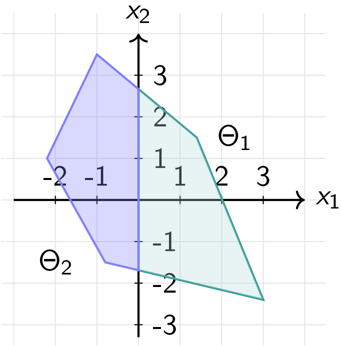
Case 1



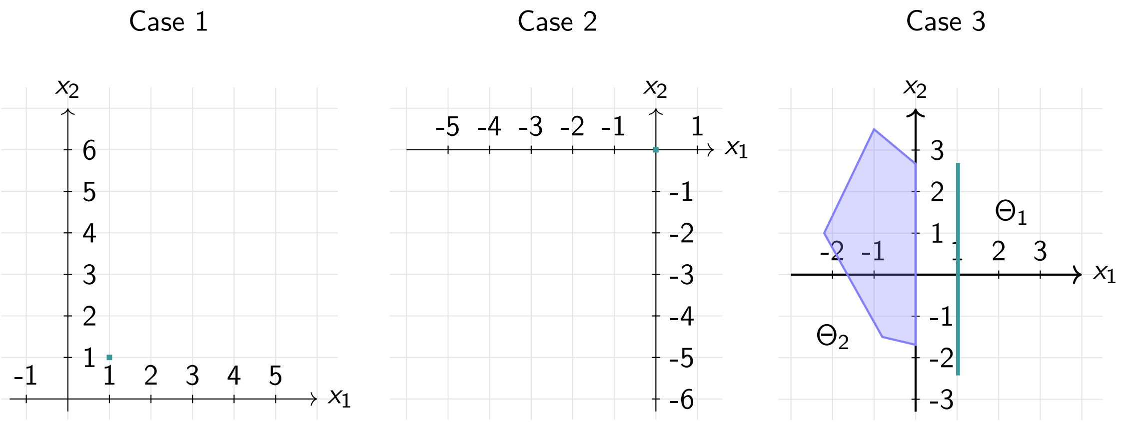
Case 2



Case 3

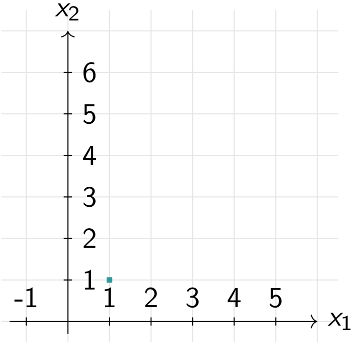


Exact Analysis

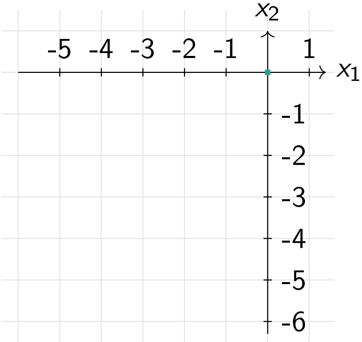


Exact Analysis

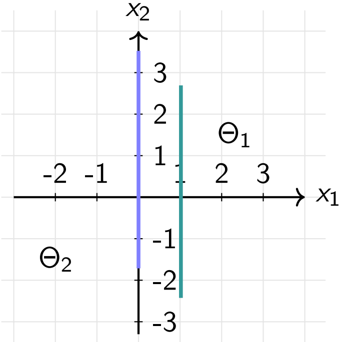
Case 1



Case 2

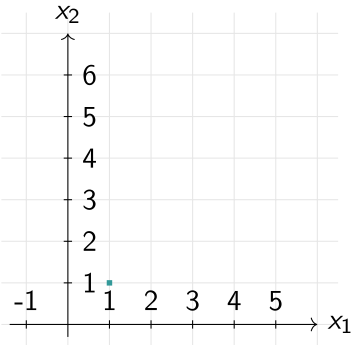


Case 3

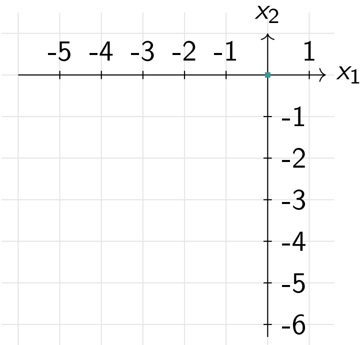


Exact Analysis

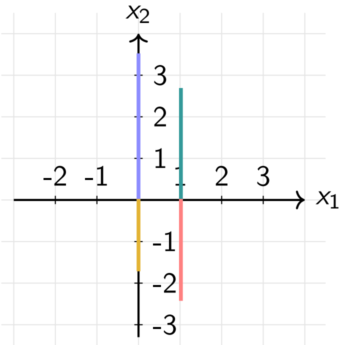
Case 1



Case 2

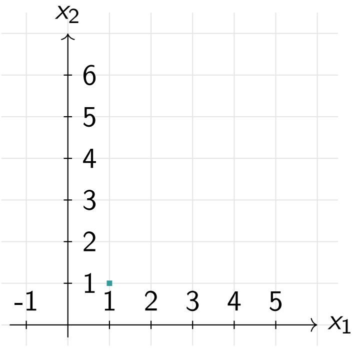


Case 3

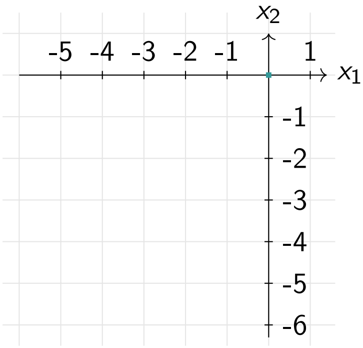


Exact Analysis

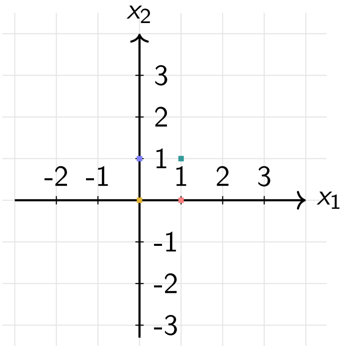
Case 1



Case 2

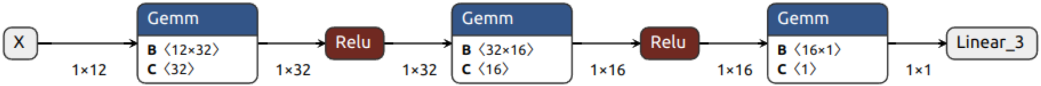


Case 3



ONNX Parser

- Open Neural Network Exchange (ONNX)
- Standard for representing deep learning models



Outline

ACAS Xu

- Airborne Collision Avoidance System Xu (ACAS Xu)
- A set of 45 feedforward neural networks, each with
  - 5 inputs,
  - 5 outputs,
  - 7 fully connected layers,
  - 300 neurons
- 10 properties to test the networks

ACAS Xu

For the evaluation, we

- compute the reachable set
- check the safety verification
- check the computation (RT) and the safety verification time (VT)

ACAS Xu

From the results,

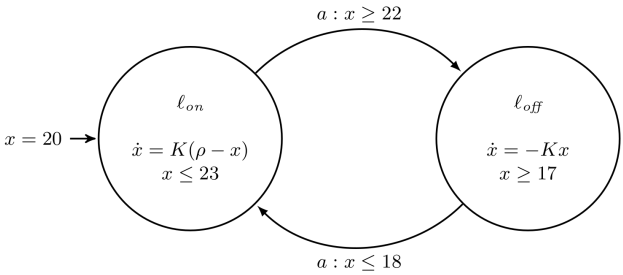
- the star set approach is, on average, faster than the Reluplex

properties	Exact	Overapproximation
	AVG RT(s)	AVG RT(s)
ϕ_1	29402.5067	2049.4807
ϕ_2	44631.003	1775.056
ϕ_3	254.60	10.38
ϕ_4	140.7655	9.4855
Sum	74428.8752	3844.4022

Average computation results for properties $\phi_1, \phi_2, \phi_3, \phi_4$

Thermostat

- Maintains room temperature between $17^\circ C$ and $23^\circ C$
- Neural network, with
  - 2 input neurons,
  - 2 hidden layers,
  - 1 output



The hybrid automaton model of the thermostat

Drones

- Drone takeoff and hover at chosen altitude
- A set of 8 feedforward neural networks

Architecture	Network ID	Neurons
Two layers	AC1	32, 16
	AC2	64, 32
	AC3	128, 64
	AC4	256, 128
Three layers	AC5	32, 16, 8
	AC6	64, 32, 16
	AC7	128, 64, 32
	AC8	256, 128, 64

Drones

For the evaluation, we check

- the computation time (RT)
- the safety verification time (VT)

Exact		Overapproximation	
RT(s)	VT(s)	RT(s)	VT(s)
232.37	8.86	4.94	1.37

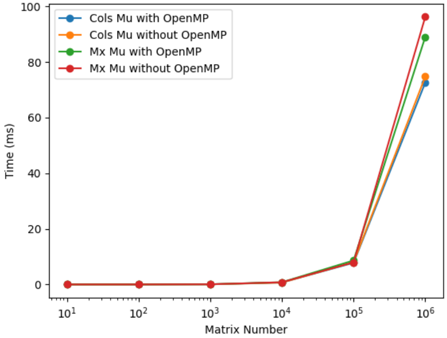
Average computation and verification time results

Sonar Binary Classifier

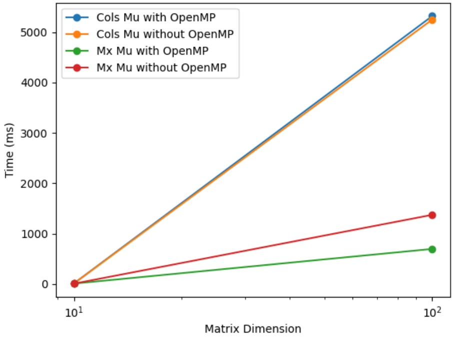
- Binary classification, distinguishing rocks from metal cylinders
- Neural network, with
  - 60 inputs,
  - 1 output,
  - 2 layers
- We check the local robustness of the neural network

Experimental Results

- Experiments to improve the running time of our algorithms, with
  - matrix multiplications vs. column multiplications



Run time with increasing matrix numbers



Run time with increasing matrix dimensions

Outline

Conclusion

- Reachability analysis
  - provides valuable insights into complex network behavior
  - ensures network safety and reliability
- Developed various activation functions, includes
  - exact analysis
  - over-approximate analysis

  for bounded and unbounded cases.
- ONNX parser for easier integration in Hypro

Questions?

References I

[] *ONNX*. https://onnx.ai/. [Accessed: May 30, 2023].

[] *OpenMP*. https://www.openmp.org/. [Accessed : May 29, 2023].

[AG21] Sushma Priya Anthadupula and Manasi Gyanchandani. "A Review and Performance Analysis of Non-Linear Activation Functions in Deep Neural Networks". In: *Int. Res. J. Mod. Eng. Technol. Sci* (2021).

[BD17] Stanley Bak and Parasara Sridhar Duggirala. "Simulation-Equivalent Reachability of Large Linear Systems with Inputs". In: *Computer Aided Verification*. Ed. by Rupak Majumdar and Viktor Kunčak. 2017, pp. 401–420.

[Bro22] Jason Brownlee. *Binary Classification Tutorial with the Keras Deep Learning Library*. https://machinelearningmastery.com/binary-classification-tutorial-with-the-keras-deep-learning-library/. [Accessed : June 1, 2023]. 2022.

[CBD15] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. "Binaryconnect: Training deep neural networks with binary weights during propagations". In: *Advances in neural information processing systems* 28 (2015).

References II

[Col+11] Ronan Collobert et al. *Natural Language Processing (almost) from Scratch*. 2011.

[Col04] Ronan Collobert. "Large Scale Machine Learning". In: (2004).

[GJ+10] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. [Accessed : May 29, 2023]. 2010.

[Gui+22] Dario Guidotti et al. *Evaluating Reachability Algorithms for Neural Networks on NeVer2*. Sept. 2022.

[Jia23] Ruoran Gabriela Jiang. "Verifying ai-controlled hybrid systems". MA thesis. RWTH Aachen University, 2023.

[JKO19] Kyle D. Julian, Mykel J. Kochenderfer, and Michael P. Owen. "Deep Neural Network Compression for Aircraft Collision Avoidance Systems". In: *Journal of Guidance, Control, and Dynamics* 42.3 (Mar. 2019), pp. 598–608. DOI: 10.2514/1.g003724. URL: https://doi.org/10.2514%2F1.g003724.

References III

[Kat+17] Guy Katz et al. "Reluplex: An efficient SMT solver for verifying deep neural networks". In: *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*. Springer. 2017, pp. 97–117.

[Kum19] Kumar, Niranjan. *Deep Learning: Feedforward Neural Networks Explained.* `https://medium.com/hackernoon/deep-learning-feedforward-neural-networks-explained-\c34ae3f084f1`. [Accessed: May 03, 2023]. 2019.

[Nwa+18] Chigozie Nwankpa et al. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 2018.

[Sch+17] Stefan Schupp et al. "HyPro: A C++ Library of State Set Representations for Hybrid Systems Reachability Analysis". In: *NASA Formal Methods*. Ed. by Clark Barrett, Misty Davies, and Temesghen Kahsai. 2017, pp. 288–294. ISBN: 978-3-319-57287-1.

References IV

[Sin+19] Gagandeep Singh et al. "An abstract domain for certifying neural networks". In: *Proceedings of the ACM on Programming Languages* 3 (Jan. 2019), pp. 1–30. DOI: 10.1145/3290354.

[SKP97] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal. "Introduction to multi-layer feed-forward neural networks". In: *Chemometrics and Intelligent Laboratory Systems* 39.1 (1997), pp. 43–62. ISSN: 0169-7439. DOI: doi.org/10.1016/S0169-7439(97)00061-0.

[Tra20] Dung Tran. "Verification of Learning-enabled Cyber-Physical Systems". PhD thesis. Vanderbilt University Graduate School, Aug. 2020.

[Xu+20] Jin Xu et al. "Reluplex made more practical: Leaky ReLU". In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. 2020, pp. 1–7. DOI: 10.1109/ISCC50000.2020.9219587.

[Yu+20] Yongbin Yu et al. "RMAF: Relu-Memristor-Like Activation Function for Deep Learning". In: *IEEE Access* 8 (2020), pp. 72727–72741. DOI: 10.1109/ACCESS.2020.2987829.

Appendix A

Lemma

The worst-case complexity of the number of stars in the reachable set of an N-neurons FNN is $\mathcal{O}(2^N)$.

Lemma

The worst-case complexity of the number of constraints of a star in the reachable set of an N-neurons FNN is $\mathcal{O}(N)$.

Lemma

The worst-case complexity of the number of variables and constraints in the reachable set of an N-neurons FNN is $N + m_0$ *and* $3N + n_0$, *respectively, where* m_0 *is the number of variables and* n_0 *the number of linear constraints of the predicate of the input set.*

Appendix B

Lemma

The worst-case complexity of the number of stars in the reachable set of an N-neurons FNN is $\mathcal{O}(3^N)$.

Lemma

The worst-case complexity of the number of constraints of a star in the reachable set of an N-neurons FNN is $\mathcal{O}(2N)$.

Lemma

The worst-case complexity of the number of variables and constraints in the reachable set of an N-neuron FNN is $N + m_0$ *and* $4N + n_0$, *where respectively* m_0 *is the number of variables and* n_0 *the number of linear constraints of the predicate of the input set.*

Appendix C

Lemma

The worst-case complexity of the number of stars in the reachable set of an N-neurons FNN is $\mathcal{O}(3^N)$.

Lemma

The worst-case complexity of the number of constraints of a star in the reachable set of an N-neurons FNN is $\mathcal{O}(2N)$.

Lemma

The worst-case complexity of the number of variables and constraints in the reachable set of an N-neurons FNN is $N + m_0$ *and* $4N + n_0$, *where respectively* m_0 *is the number of variables and* n_0 *the number of linear constraints of the predicate of the input set.*

Appendix D

Lemma

The worst-case complexity of the number of stars in the reachable set of an N-neurons FNN is $\mathcal{O}(2^N)$.

Lemma

The worst-case complexity of the number of constraints of a star in the reachable set of an N-neurons FNN is $\mathcal{O}(N)$.