Semialgebraic Representation of ReLU Networks and Their Applications to Robustness Certification

Tong Chen, Jean-Bernard Lasserre, Victor Magron, Edouard Pauwels

LAAS-CNRS, MIT, IRIT, ANITI

Introduction

Robustness verification of neural network F: input $\mathbf{x}_0$, perturbation threshold ϵ.

$$\max_{\mathbf{x}} \quad \|F(\mathbf{x}) - F(\mathbf{x}_0)\|, \quad \text{s.t. } \|\mathbf{x}-\mathbf{x}_0\| \le \varepsilon^2$$

General methodology:

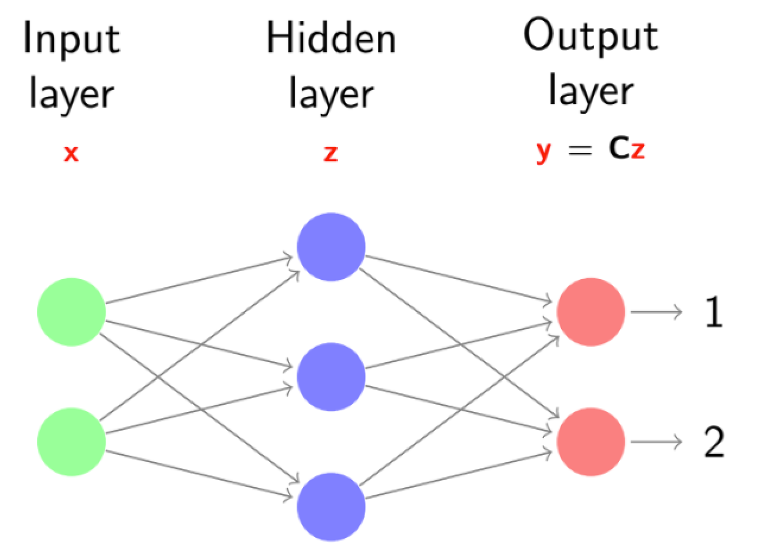
- exact semi-algebraic representation + polynomial optimization + SDP relaxation.

Contributions:

- Estimating Lipschitz constant of relu networks [3];
- SDP relaxation taylored to NN certification [5].
- Robustness certification of implicit networks [4].

Structure of neural networks

Input layer: $\mathbf{x}$, hidden layer: $\mathbf{z}$, output layer: $\mathbf{y}$.



Activation function: $\sigma(x) = \text{ReLU}(x) = \max\{0, x\}$

- Deep neural network (DNN weights $\mathbf{A}, \mathbf{b}$):

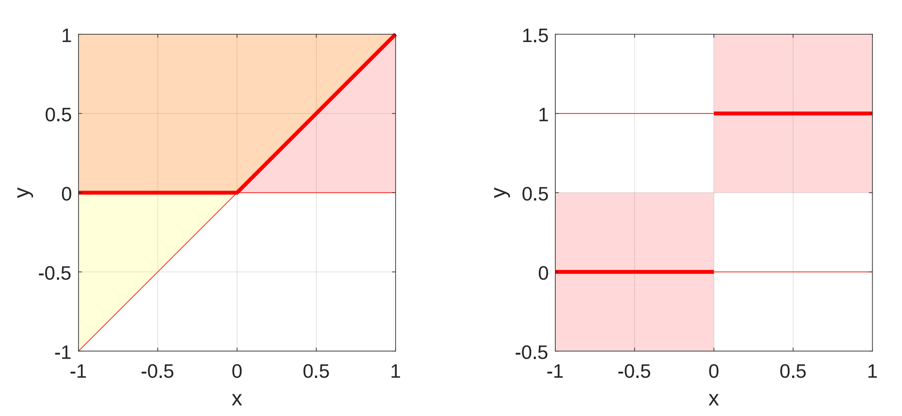
$$\mathbf{z} = \text{ReLU}(\mathbf{A}\mathbf{x} + \mathbf{b})$$

- Monotone equilibrium model (monDEQ, weights $\mathbf{W}, \mathbf{U}, \mathbf{u}$):

$$\mathbf{z} = \text{ReLU}(\mathbf{W}\mathbf{z} + \mathbf{U}\mathbf{x} + \mathbf{u})$$

Semialgebraicity of ReLU, ∂ReLU

- $y = \text{ReLU}(x) \quad \Leftrightarrow \quad y(y-x) = 0,\ y \ge x, y \ge 0.$
- $y = \partial\text{ReLU}(x) \quad \Leftrightarrow \quad y(y-1) = 0,\ (y - \frac{1}{2})x \ge 0$



Hidden layer in DNN:

$$\mathbf{z}(\mathbf{z} - (\mathbf{A}\mathbf{x}+\mathbf{b})) = 0,\ \mathbf{z} \ge (\mathbf{A}\mathbf{x}+\mathbf{b}),\ \mathbf{z} \ge 0.$$

Hidden in monDEQ:

$$\mathbf{z}(\mathbf{z} - \mathbf{W}\mathbf{z} - \mathbf{U}\mathbf{x} - \mathbf{u}) = 0,\ \mathbf{z} \ge \mathbf{W}\mathbf{z} + \mathbf{U}\mathbf{x} + \mathbf{u},\ \mathbf{z} \ge 0.$$

Polynomial optimization

Polynomial optimization problem (POP):

$$\max_{\mathbf{x}\in\mathbb{R}^n} \{f(\mathbf{x}) : g_i(\mathbf{x}) \ge 0, i = 1, \ldots, p\}$$

where f and g_i are polynomials in variable $\mathbf{x} \in \mathbb{R}^n$.

Semidefinite Programming (SDP)

Semidefinite programming (SDP):

Input: $\mathbf{C} \in \mathbb{S}^n$, $\mathbf{A}_k \in \mathbb{S}^n$, $b_k \in \mathbb{R}$, $k = 1, \ldots, m$.

$$\min_{\mathbf{X}\in\mathbb{S}^n} \{\langle \mathbf{C}, \mathbf{X}\rangle_{\mathbb{S}^n} : \langle \mathbf{A}_k, \mathbf{X}\rangle_{\mathbb{S}^n} = b_k, k = 1, \ldots, m; \mathbf{X} \succeq 0\}$$

where $\mathbb{S}^n$ is the space of real symmetric $n \times n$ matrices, and $\langle \cdot, \cdot \rangle_{\mathbb{S}^n}$ is the Frobenius scalar product in $\mathbb{S}^n$.

From POP to SDP: Lasserre's relaxation

Set $\omega_i = \lceil \deg g_i / 2 \rceil$ and choose $d \in \mathbb{N}$ a degree bound:

$$\max_{\mathbf{y}: L_{\mathbf{y}}(1)=1} \{L_{\mathbf{y}}(f) : \mathbf{M}_d(\mathbf{y}) \succeq 0, \mathbf{M}_{d-\omega_i}(g_i\mathbf{y}) \succeq 0\}.$$

This is an SDP with:

- $p+1$ positive semidefinite matrices.
- of size $\binom{n+2d}{2d}$, $\binom{n+2(d-\omega_i)}{2(d-\omega_i)}$.

Main features:

- For any d: certified POP upper bound.
- As $d \to \infty$: converges to POP solution.

Main challenge for NN certification: scalability.

Lipschitz model of DNN

Fix input $\mathbf{x}_0$ and perturbation ε.

$$\max_{\mathbf{x},\mathbf{s},\mathbf{t}} \quad \mathbf{t}^T\mathbf{A}^T\text{diag}(\mathbf{s})\mathbf{c}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{s}(\mathbf{s}-1) = 0, \\ (\mathbf{s} - \frac{1}{2})(\mathbf{A}\mathbf{x}+\mathbf{b}) \ge 0; \\ \mathbf{t}^2 \le 1, \\ \|\mathbf{x} - \mathbf{x}_0\|^2 \le \varepsilon^2. \end{cases}$$

Robustness model of monDEQ

Fix input $\mathbf{x}_0$ and perturbation ε.

$$\max_{\mathbf{x},\mathbf{z}} \quad \mathbf{c}^T\mathbf{z}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{z}(\mathbf{z} - \mathbf{W}\mathbf{z} - \mathbf{U}\mathbf{x} - \mathbf{u}) = 0, \\ \mathbf{z} \ge \mathbf{W}\mathbf{z} + \mathbf{U}\mathbf{x} + \mathbf{u}, \\ \mathbf{z} \ge 0, \\ (\mathbf{x}_0 - \bar{\mathbf{x}}_0 + \varepsilon)(\mathbf{x}_0 - \|\mathbf{x} - \mathbf{x}_0\|^2 \le \varepsilon^2. \end{cases}$$

Numerical results: Lipschitz constant of DNNs

Upper bounds of Lipschitz constant and running time of various methods for SDP-NN network:

	Our SDP method [3]	Shor	LP method [1]
Bound	14.56	17.85	OfM
Time	12246	2869	OfM

OfM: out of memory.

Numerical results: Robustness certification of monDEQs

Ratio of certified examples among the first 100 test examples of MNIST dataset for robustness model.

Norm	ε	Our method [4]	Pabbaraju et. al.[2]
L_2	0.1	99%	91%
L_∞	0.1	0%	0%
	0.05	24%	0%
	0.01	99%	24%

Conclusion

Advantage:

- guaranteed upper bounds for Lipschitz constant and certifying robustness of neural networks;
- significantly improves state of the art.

Main challenge and future research:

- scalability.
- limitation of SDP solvers.

References

[1] Latorre et. al. Lipschitz constant estimation of neural networks via sparse polynomial optimization. ICLR 2020.

[2] Pabbaraju et. al. Estimating lipschitz constants of monotone deep equilibrium models. ICLR 2021.

[3] C. L. M. P. Semialgebraic optimization for lipschitz constants of relu networks. NeurIPS 2020.

[4] C. L. M. P. Semialgebraic representation of monotone deep equilibrium models and applications to certification. NeurIPS 2021.

[5] C. L. M. P. A sublevel moment-sos hierarchy for polynomial optimization. *Computational Optimization and Applications*, 81(1):31–66, 2022.

ANITI
ARTIFICIAL & NATURAL INTELLIGENCE TOULOUSE INSTITUTE

LAAS CNRS