PhD defense: Robustness Verification of ReLU Networks using Polynomial Optimization

Tong Chen
POP team, LAAS-CNRS
Supervisors: E. Pauwels, V. Magron and J.-B. Lasserre

December 2, 2022

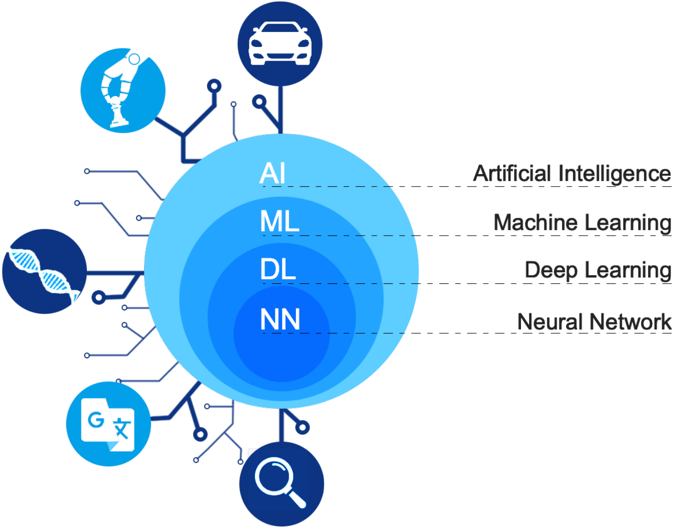
Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
3. Part III: Robustness Verification (Chapter 5)
4. Conclusion and future works

Outline

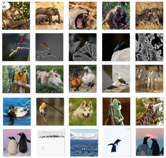
1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
3. Part III: Robustness Verification (Chapter 5)
4. Conclusion and future works

Artificial intelligence (AI) and neural network (NN)



NN for classification

Dataset with labels



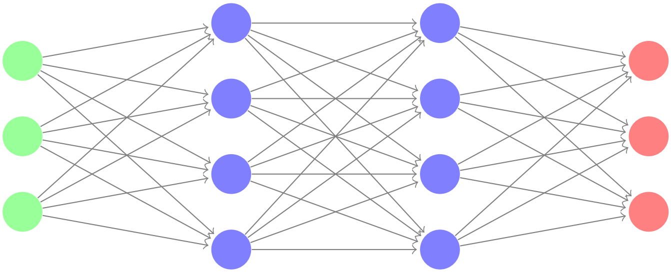
$$\xrightarrow{\text{training}} \text{NN} \xrightarrow{\text{classification}} \begin{cases} \text{cat} \\ \text{dog} \\ \text{tiger} \\ \text{panda} \\ \dots \end{cases}$$

What is this?

$$\xrightarrow{\text{input}} \text{NN} \xrightarrow{\text{output}} \text{This is a panda!}$$

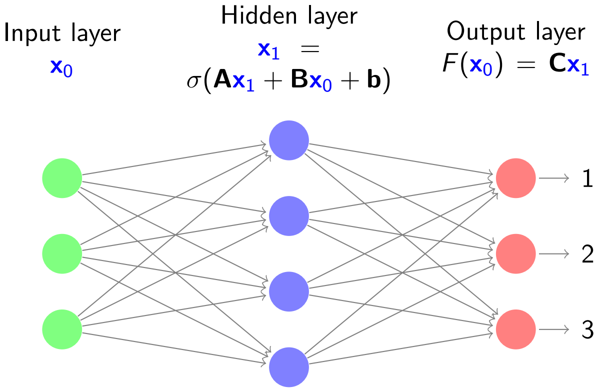
Deep neural network (DNN)

Input layer	Hidden layer 1	Hidden layer 2	Output layer
$\mathbf{x}_0$	$\mathbf{x}_1 = \sigma(\mathbf{A}_1\mathbf{x}_0 + \mathbf{b}_1)$	$\mathbf{x}_2 = \sigma(\mathbf{A}_2\mathbf{x}_1 + \mathbf{b}_2)$	$F(\mathbf{x}_0) = \mathbf{C}\mathbf{x}_3$



Fully-connected DNN with $\sigma(x) = \mathrm{ReLU}(x) = \max(x, 0)$.

Monotone operator equilibrium network (monDEQ)

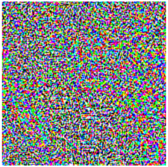


Fully-connected monDEQ with $\sigma(x) = \mathrm{ReLU}(x) = \max(x, 0)$.

Adversarial example

This is a panda!

+



=

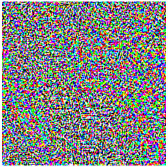
This is a gibbon!

Ref: [Goodfellow15].

Adversarial example

This is a panda!

+

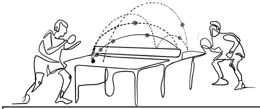


=

This is a gibbon!

Ref: [Goodfellow15].

Attack v.s. defense



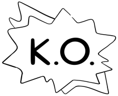
adversarial example

attack

defend

adversarial training

$\rightsquigarrow$

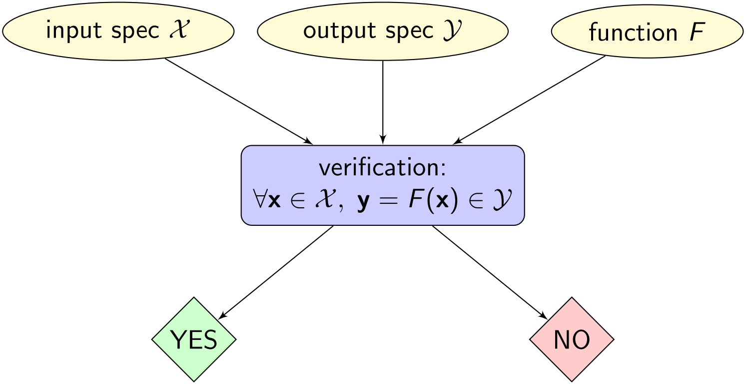


(sound) verification

train

(provable) defense

Neural network verification: input-output satisfiability



An example: robustness verification of NN

- $F : \mathcal{X} \to \mathbb{R}^K$, classification;

An example: robustness verification of NN

- $F : \mathcal{X} \to \mathbb{R}^K$, classification;
- $F_k := F(\cdot)_k, \; y(\mathbf{x}_0) = \arg\max_k F_k(\mathbf{x}_0)$;

An example: robustness verification of NN

- $F : \mathcal{X} \to \mathbb{R}^K$, classification;
- $F_k := F(\cdot)_k, \; y(\mathbf{x}_0) = \arg\max_k F_k(\mathbf{x}_0)$;
- Fix $\bar{\mathbf{x}}$, take $\mathbf{B}(\bar{\mathbf{x}}, \varepsilon, \|\cdot\|_p) := \{\mathbf{x} : \|\mathbf{x} - \bar{\mathbf{x}}\|_p \leq \varepsilon\}$.

An example: robustness verification of NN

- $F : \mathcal{X} \to \mathbb{R}^K$, classification;
- $F_k := F(\cdot)_k,\ y(\mathbf{x}_0) = \arg\max_k F_k(\mathbf{x}_0)$;
- Fix $\bar{\mathbf{x}}$, take $\mathbf{B}(\bar{\mathbf{x}}, \varepsilon, \|\cdot\|_p) := \{\mathbf{x} : \|\mathbf{x} - \bar{\mathbf{x}}\|_p \leq \varepsilon\}$.

ε-robust w.r.t. L_p norm at $\bar{\mathbf{x}}$

$$\forall \mathbf{x}_0 \in \mathbf{B}(\bar{\mathbf{x}}, \varepsilon, \|\cdot\|_p),\ y_0 := y(\mathbf{x}_0) = y(\bar{\mathbf{x}}) =: \bar{y},$$

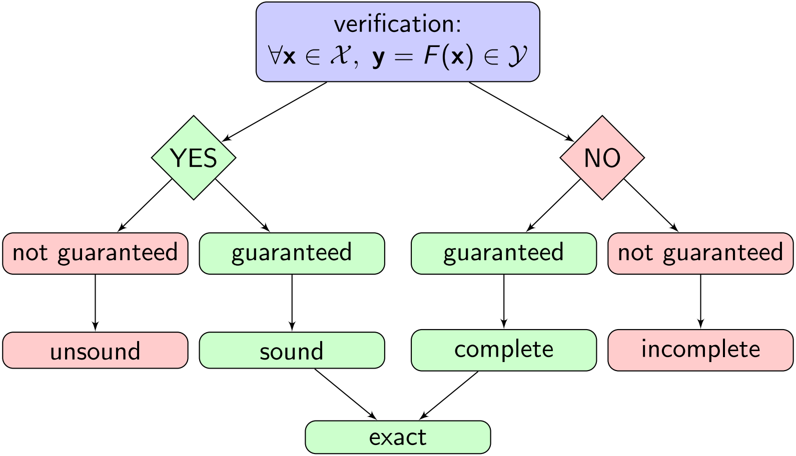
$$\Updownarrow$$

$$F_k(\mathbf{x}_0) < F_{\bar{y}}(\mathbf{x}_0),\ \forall k \neq \bar{y},$$

$$\Updownarrow$$

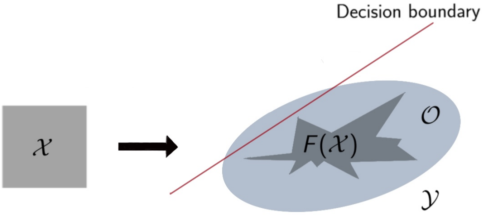
$$F_k(\mathbf{x}_0) - F_{\bar{y}}(\mathbf{x}_0) < 0,\ \forall k \neq \bar{y}.$$

Completeness and soundness



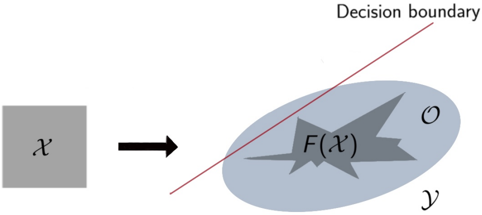
Completeness and soundness: examples

- **sound (not complete)** approach: find $\mathcal{O} \supseteq F(\mathcal{X})$, verify $\mathcal{O} \subseteq \mathcal{Y}$;

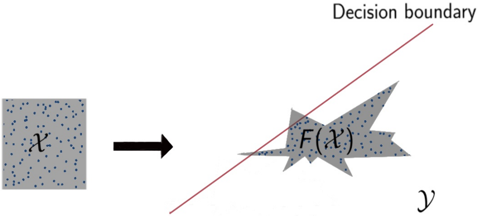


Completeness and soundness: examples

- **sound (not complete)** approach: find $\mathcal{O} \supseteq F(\mathcal{X})$, verify $\mathcal{O} \subseteq \mathcal{Y}$;



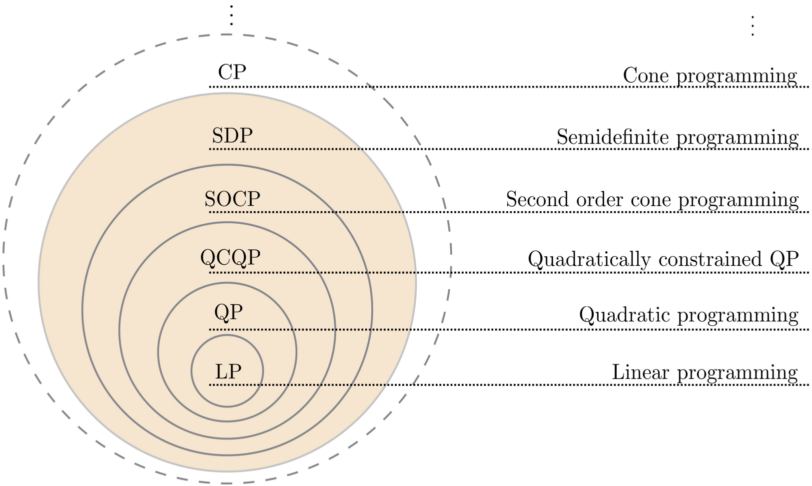
- **complete (not sound)** approach: find $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} = F(\mathbf{x}) \notin \mathcal{Y}$.



History of neural network verification

	(complete verifiers)	
DNN $<$ 100 neurons	SAT, MILP	2017
	(incomplete verifiers)	
	BP, convex relaxation	2018
	$+$ branch and bound	2020
CNN $>$ 100K neurons	$+$ GPU B&B	now

Incomplete approach: convex relaxation



Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
   - POP and Lasserre's relaxation
   - Sublevel relaxation and applications
3. Part III: Robustness Verification (Chapter 5)
4. Conclusion and future works

Polynomial optimization problem (POP)

For $f, g_i \in \mathbb{R}[\mathbf{x}]$, consider

$$f^* = \min_{\mathbf{x}} \quad f(\mathbf{x}) \qquad \text{(POP)}$$
$$\text{s.t.} \quad g_i(\mathbf{x}) \geq 0, \; i = 1, \ldots, p.$$

Polynomial optimization problem (POP)

For $f, g_i \in \mathbb{R}[\mathbf{x}]$, consider

$$f^* = \min_{\mathbf{x}} \quad f(\mathbf{x}) \qquad \text{(POP)}$$
$$\text{s.t.} \quad g_i(\mathbf{x}) \geq 0, \ i = 1, \ldots, p.$$

- Non-convex, NP-hard $\longrightarrow$ **relax** it!

Sum-of-square (SOS) relaxation

$$\left(-\frac{1}{2} = \right) \qquad \min_{\mathbf{x} \in \mathbb{R}^2} \quad \{ f(\mathbf{x}) = x_1 x_2 : g(\mathbf{x}) = 1 - x_1^2 - x_2^2 \geq 0 \}$$

Sum-of-square (SOS) relaxation

$$\left(-\frac{1}{2}=\right) \qquad \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^2 - x_2^2 \geq 0\}$$

$$\geq \max_{\lambda\geq 0} \underbrace{\min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) - \lambda \cdot g(\mathbf{x})\}}_{\text{Lagrangian relaxation}}$$

Sum-of-square (SOS) relaxation

$$
\begin{aligned}
\left(-\frac{1}{2} = \right) \qquad & \min_{\mathbf{x} \in \mathbb{R}^2} \quad \{ f(\mathbf{x}) = x_1 x_2 : g(\mathbf{x}) = 1 - x_1^2 - x_2^2 \geq 0 \} \\
& \geq \max_{\lambda \geq 0} \underbrace{\min_{\mathbf{x} \in \mathbb{R}^2} \quad \{ f(\mathbf{x}) - \lambda \cdot g(\mathbf{x}) \}}_{\text{Lagrangian relaxation}} \\
& = \max_{\lambda \geq 0} \max_{\mu \in \mathbb{R}} \quad \{ \mu : \mu \leq f(\mathbf{x}) - \lambda \cdot g(\mathbf{x}) \}
\end{aligned}
$$

Sum-of-square (SOS) relaxation

$$\left(-\frac{1}{2}=\right) \qquad \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1 x_2 : g(\mathbf{x}) = 1 - x_1^2 - x_2^2 \geq 0\}$$

$$\geq \max_{\lambda\geq 0} \underbrace{\min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) - \lambda \cdot g(\mathbf{x})\}}_{\text{Lagrangian relaxation}}$$

$$= \max_{\lambda\geq 0} \max_{\mu\in\mathbb{R}} \quad \{\mu : \mu \leq f(\mathbf{x}) - \lambda \cdot g(\mathbf{x})\}$$

$$= \max_{\lambda\geq 0,\ \mu\in\mathbb{R}} \quad \{\mu : f - \mu - \lambda \cdot g \geq 0\} \qquad \longrightarrow \text{hard!}$$

Sum-of-square (SOS) relaxation

$$
\begin{aligned}
\left(-\frac{1}{2}=\right) \qquad & \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1 x_2 : g(\mathbf{x}) = 1 - x_1^2 - x_2^2 \geq 0\} \\
\geq\ & \max_{\lambda\geq 0} \underbrace{\min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) - \lambda \cdot g(\mathbf{x})\}}_{\text{Lagrangian relaxation}} \\
=\ & \max_{\lambda\geq 0} \max_{\mu\in\mathbb{R}} \quad \{\mu : \mu \leq f(\mathbf{x}) - \lambda \cdot g(\mathbf{x})\} \\
=\ & \max_{\lambda\geq 0,\ \mu\in\mathbb{R}} \quad \{\mu : f - \mu - \lambda \cdot g \geq 0\} \qquad \longrightarrow \text{hard!} \\
\geq\ & \max_{\lambda\geq 0, \mu\in\mathbb{R}} \quad \{\mu : f - \mu - \lambda \cdot g = \text{SOS}\} \qquad \longrightarrow \text{easy! (SDP!)}
\end{aligned}
$$

Sum-of-square (SOS) relaxation

$$
\begin{aligned}
\left(-\frac{1}{2}=\right) \qquad & \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^2 - x_2^2 \geq 0\} \\
& \geq \max_{\lambda\geq 0} \underbrace{\min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) - \lambda \cdot g(\mathbf{x})\}}_{\text{Lagrangian relaxation}} \\
& = \max_{\lambda\geq 0} \max_{\mu\in\mathbb{R}} \quad \{\mu : \mu \leq f(\mathbf{x}) - \lambda \cdot g(\mathbf{x})\} \\
& = \max_{\lambda\geq 0,\ \mu\in\mathbb{R}} \quad \{\mu : f - \mu - \lambda \cdot g \geq 0\} \qquad \longrightarrow \text{hard!} \\
& \geq \max_{\lambda\geq 0, \mu\in\mathbb{R}} \quad \{\mu : f - \mu - \lambda \cdot g = \text{SOS}\} \qquad \longrightarrow \text{easy! (SDP!)}
\end{aligned}
$$

$$
\underbrace{x_1x_2}_{f} - \underbrace{\left(-\frac{1}{2}\right)}_{\mu} = \underbrace{\left(\frac{x_1+x_2}{\sqrt{2}}\right)^2}_{\text{SOS}} + \underbrace{\frac{1}{2}}_{\lambda} \cdot \underbrace{(1 - x_1^2 - x_2^2)}_{g}.
$$

SOS relaxation: dense [Lasserre01]

Given (POP). For $d \geq 1$:

$$\rho_d := \max_{\sigma_i, \lambda} \quad \lambda$$

$$\text{s.t.} \quad \begin{cases} f - \lambda = \sigma_0 + \sum_{i=1}^{p} \sigma_i \cdot g_i, \\ \sigma_0 \text{ is SOS of deg} \leq 2d, \\ \sigma_i \text{ is SOS of deg} \leq 2(d - \omega_i), \end{cases}$$

where $\omega_i = \lceil \deg(g_i)/2 \rceil$.

SOS relaxation: dense [Lasserre01]

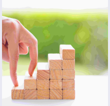
Given (POP). For $d \geq 1$:

$$\rho_d := \max_{\sigma_i, \lambda} \quad \lambda$$

$$\text{s.t.} \quad \begin{cases} f - \lambda = \sigma_0 + \sum_{i=1}^{p} \sigma_i \cdot g_i, \\ \sigma_0 \text{ is SOS of deg} \leq 2d, \\ \sigma_i \text{ is SOS of deg} \leq 2(d - \omega_i), \end{cases}$$

where $\omega_i = \lceil \deg(g_i)/2 \rceil$.

- SDP, **convex**;
- primal-dual pair: moment-SOS relaxation;
- add $M - \|\mathbf{x}\|_2^2 \geq 0$, then $\rho_d \uparrow f^*$ as $d \to \infty$.



Computational complexity

- size of matrix: $\binom{n+d}{d} = O(n^d)$;
- $n = 100,\ d = 2 \longrightarrow 10000$, not possible;
- scales only for $n < 50$.

Computational complexity

- size of matrix: $\binom{n+d}{d} = O(n^d)$;
- $n = 100, \ d = 2 \longrightarrow 10000$, not possible;
- scales only for $n < 50$.

sparsity!

Layer structure induces sparsity

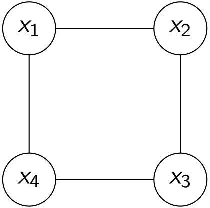
total # of variables: $n = 10 + 10 = 20$, take $d = 2$

size and number of matrices in different cases

	dense	sparse					
		correlative	term				sublevel
size	[231]	[78]	[21,	12,	2,	1]	$\mathbf{l} = [l_i]$
number	[1]	[10]	[1,	10,	100,	90]	$\mathbf{q} = [q_i]$

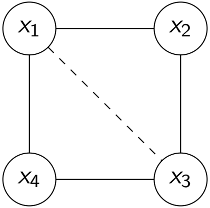
Correlative sparsity pattern (CSP)

$$\min_{\mathbf{x}\in\mathbb{R}^2} \left\{ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 : 1 - x_i^2 - x_j^2 \geq 0 \right\}.$$



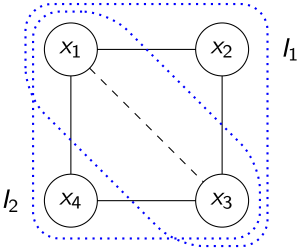
Correlative sparsity pattern (CSP)

$$\min_{\mathbf{x}\in\mathbb{R}^2} \left\{ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 : 1 - x_i^2 - x_j^2 \geq 0 \right\}.$$



Correlative sparsity pattern (CSP)

$$\min_{\mathbf{x}\in\mathbb{R}^2} \left\{ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 : 1 - x_i^2 - x_j^2 \geq 0 \right\}.$$



CSP graph, matrix size $= \binom{3+d}{d}$

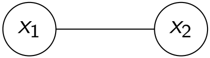
Term sparsity pattern (TSP) [Wang19]

$$\min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0\}.$$

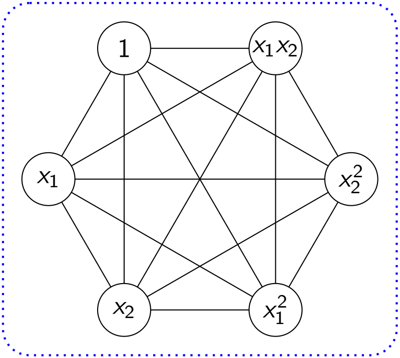
Term sparsity pattern (TSP) [Wang19]

$$\min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0\}.$$

(CSP)



I_1

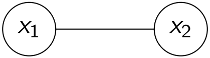


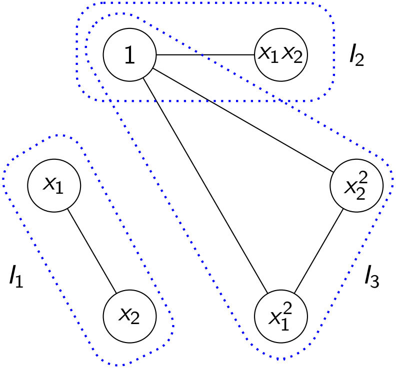
CSP graph, matrix size $= 6$

Term sparsity pattern (TSP) graph [Wang19]

$$\min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0\}.$$

(TSP)





TSP graph, matrix size $= 3$

Sparse polynomial optimization

A sparse POP is stated as follows

$$\begin{aligned} f^* = \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) = \sum f_i(\mathbf{x}_{I_k}) \\ \text{s.t.} \quad & g_i(\mathbf{x}_{I_k}) \geq 0, \end{aligned} \tag{SparsePOP}$$

where I_k are maximal cliques in the chordal extension.

SOS relaxation: sparse [Waki05; Lasserre06; Wang19]

Given (SparsePOP). For $d \geq 1$:

$$\varphi_d := \max_{\lambda} \quad \lambda$$

$$\text{s.t.} \quad \begin{cases} f(\mathbf{x}) - \lambda = \sum_{k=1}^{m} \left(\sigma_{0,k}(\mathbf{x}_{I_k}) + \sum_{i \in J_k} \sigma_{i,k}(\mathbf{x}_{I_k}) \cdot g_i(\mathbf{x}_{I_k}) \right), \\ \sigma_{0,k} \text{ is SOS of deg} \leq 2d, \\ \sigma_{i,k} \text{ is SOS of deg} \leq 2(d - \omega_i), \end{cases}$$

where $\omega_i = \lceil \deg(g_i)/2 \rceil$.

SOS relaxation: sparse [Waki05; Lasserre06; Wang19]

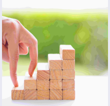
Given (SparsePOP). For $d \geq 1$:

$$\varphi_d := \max_{\lambda} \quad \lambda$$

$$\text{s.t.} \quad \begin{cases} f(\mathbf{x}) - \lambda = \sum_{k=1}^{m} \left(\sigma_{0,k}(\mathbf{x}_{I_k}) + \sum_{i \in J_k} \sigma_{i,k}(\mathbf{x}_{I_k}) \cdot g_i(\mathbf{x}_{I_k}) \right), \\ \sigma_{0,k} \text{ is SOS of deg} \leq 2d, \\ \sigma_{i,k} \text{ is SOS of deg} \leq 2(d - \omega_i), \end{cases}$$

where $\omega_i = \lceil \deg(g_i)/2 \rceil$.

- SDP, **convex**;
- primal-dual pair: moment-SOS relaxation;
- chordal, add $M_k - \|\mathbf{x}_{I_k}\|_2^2 \geq 0$, then $\varphi_d \uparrow f^*$ as $d \to \infty$.



Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
   - POP and Lasserre's relaxation
   - Sublevel relaxation and applications
3. Part III: Robustness Verification (Chapter 5)
4. Conclusion and future works

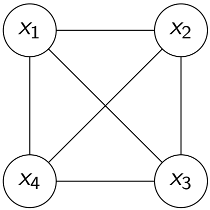
T. Chen, J.-B. Lasserre, V. Magron, E. Pauwels (2022). A Sublevel Moment-SOS Hierarchy for Polynomial Optimization, *Computational Optimization and Applications.*

Sublevel: subgraph from CSP graph

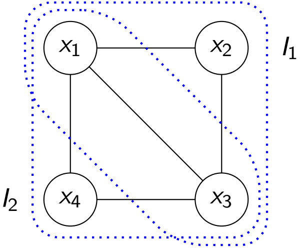
$$(-2 =) \qquad \min_{\mathbf{x} \in \mathbb{R}^2} \left\{ x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_1 : 1 - x_i^2 - x_j^2 \geq 0 \right\}.$$

Sublevel: subgraph from CSP graph

$$(-2 =) \qquad \min_{\mathbf{x}\in\mathbb{R}^2} \quad \left\{ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 : 1 - x_i^2 - x_j^2 \geq 0 \right\}.$$



$\longrightarrow$

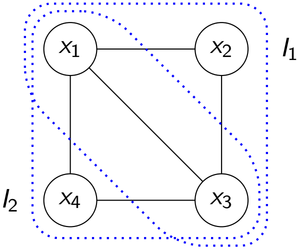


complete graph $\rightsquigarrow -2$

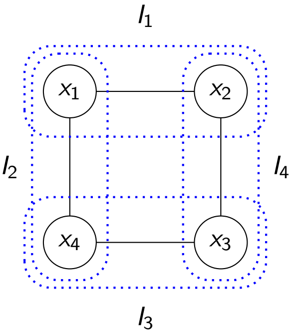
CSP subgraph $\rightsquigarrow -2$

Sublevel: subgraph from CSP graph

$$(-2 =) \qquad \min_{\mathbf{x}\in\mathbb{R}^2} \quad \left\{ x_1x_2 + x_2x_3 + x_3x_4 + x_4x_1 : 1 - x_i^2 - x_j^2 \geq 0 \right\}.$$



$\longrightarrow$



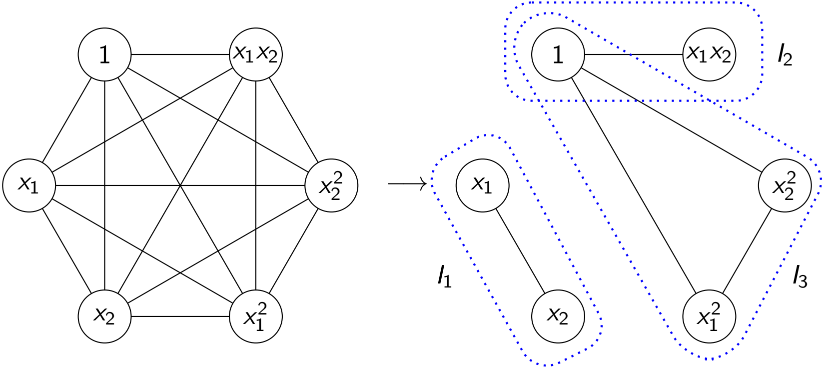
CSP subgraph $\rightsquigarrow -2$ sublevel (2,1) $\rightsquigarrow -2$

Sublevel: subgraph from TSP graph

$$\left(-\frac{1}{\sqrt{2}}=\right) \qquad \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0\}.$$

Sublevel: subgraph from TSP graph

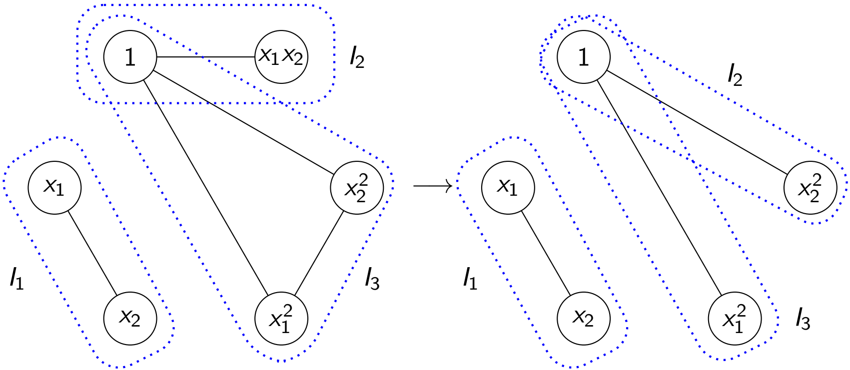
$$\left(-\frac{1}{\sqrt{2}} = \right) \qquad \min_{\mathbf{x} \in \mathbb{R}^2} \quad \{ f(\mathbf{x}) = x_1 x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0 \}.$$



complete graph $\rightsquigarrow -\frac{1}{\sqrt{2}}$ TSP subgraph $\rightsquigarrow -\frac{1}{\sqrt{2}}$

Sublevel: subgraph from TSP graph

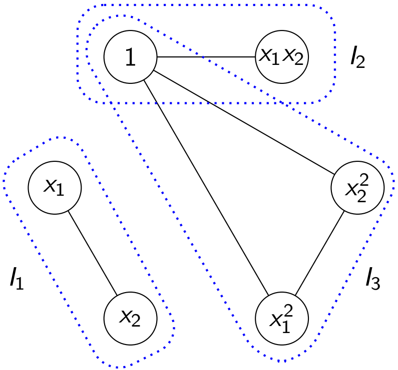
$$\left(-\frac{1}{\sqrt{2}} = \right) \qquad \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0\}.$$



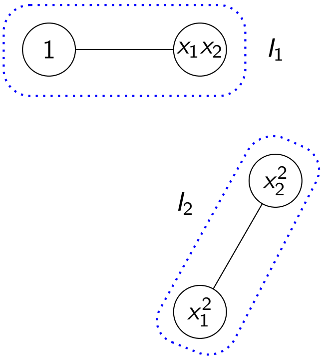
TSP graph $\rightsquigarrow -\frac{1}{\sqrt{2}}$ sublevel (2,3) $\rightsquigarrow -\frac{1}{\sqrt{2}}$

Sublevel: subgraph from TSP graph

$$\left(-\frac{1}{\sqrt{2}} = \right) \qquad \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0\}.$$



$\longrightarrow$

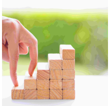


TSP graph $\rightsquigarrow -\frac{1}{\sqrt{2}}$ sublevel (2,2) $\rightsquigarrow -\frac{1}{\sqrt{2}}$

Sublevel structure: order free & clique free

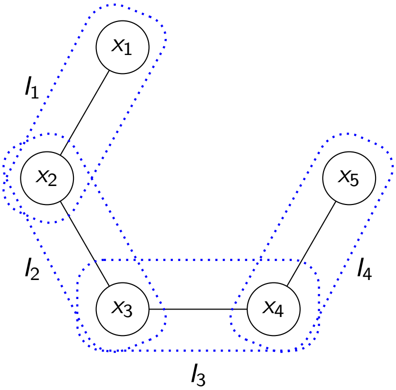
$$\mathcal{A} = \{x_1,\ x_2,\ \ldots,\ x_n\},\ \text{level } l,\ \text{depth } q.$$

$$\begin{tikzcd}
 & l=0 & l=1 & l=2 & \cdots & l=n \\
q=1 & \varnothing \arrow[r] \arrow[d] & \boxed{\{x_i\}} \arrow[r] \arrow[d] & \boxed{\{x_j, x_k\}} \arrow[r] \arrow[d] & \cdots \arrow[r] & \mathcal{A} \arrow[d] \\
q=2 & \varnothing \arrow[r] \arrow[d] & \boxed{\begin{matrix}\{x_{i_1}\}\\ \{x_{i_2}\}\end{matrix}} \arrow[r] \arrow[d] & \boxed{\begin{matrix}\{x_{j_1}, x_{k_1}\}\\ \{x_{j_2}, x_{k_2}\}\end{matrix}} \arrow[r] \arrow[d] & \cdots \arrow[r] & \mathcal{A} \arrow[d] \\
q=3 & \varnothing \arrow[r] \arrow[d] & \boxed{\begin{matrix}\{x_{i_1}\}\\ \{x_{i_2}\}\\ \{x_{i_3}\}\end{matrix}} \arrow[r] \arrow[d] & \boxed{\begin{matrix}\{x_{j_1}, x_{k_1}\}\\ \{x_{j_2}, x_{k_2}\}\\ \{x_{j_3}, x_{k_3}\}\end{matrix}} \arrow[r] \arrow[d] & \cdots \arrow[r] & \mathcal{A} \arrow[d] \\
\vdots & \vdots & \vdots & \vdots & & \vdots
\end{tikzcd}$$

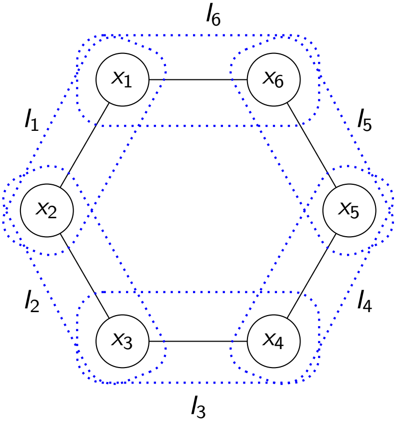


Question: How do we choose the cliques?

$\mathcal{A} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, level $l = 2$, depth $q = 4$.



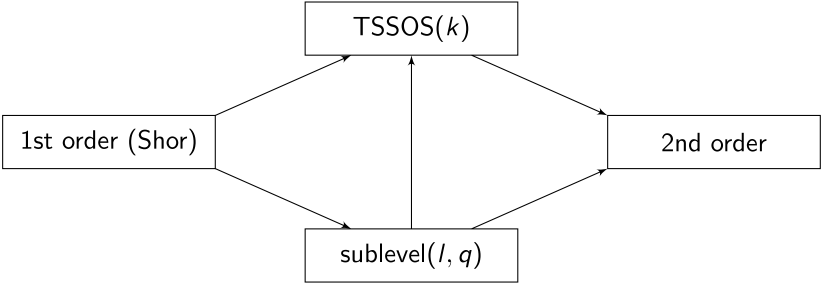
ordered heuristic $O(l, q)$



cyclic heuristic $C(l)$

Moment-SOS relaxations for (non-convex) QCQP

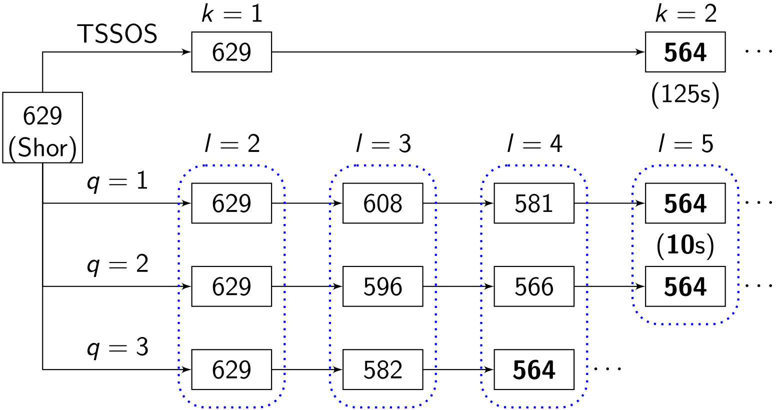
Apply sublevel structure for each polynomial constraint.



A Max-Cut example of size 800

Instance G11 from: http://web.stanford.edu/∼yyye/yyye/Gset/.

Apply $O(l, q)$ to: $\quad \max \quad \{\mathbf{x}^T \mathbf{L} \mathbf{x} : x_i^2 = 1\}. \qquad (= 564)$



A united view of moment-SOS relaxations

- Lasserre's relaxation [Lasserre01, 06]:

$$\underbrace{d\text{-th order}}_{I:\text{ all }0} \longrightarrow \underbrace{(d+1)\text{-th sublevel}}_{(I,q)} \longrightarrow \underbrace{(d+1)\text{-th order}}_{I:\text{ all }n}$$

- multi-order relaxation [Josz18]:

$$\underbrace{\text{Shor}}_{I:\text{ all }0} \longrightarrow \underbrace{\text{multi-order}}_{I:\text{ some }0,\text{ some }n;\ q=1} \longrightarrow \underbrace{\text{2nd order}}_{I:\text{ all }n}$$

- partial relaxation [Campos22]:

$$\underbrace{\text{Shor}}_{I:\text{ all }0} \longrightarrow \underbrace{\text{partial}}_{I:\text{ some }0,\text{ some }s\ll n;\ q=1} \longrightarrow \underbrace{\text{2nd order}}_{I:\text{ all }n}$$

- TSSOS [Wang19]:

$$\underbrace{\text{Shor}}_{I:\text{ all }0} \longrightarrow \underbrace{\text{TSSOS}}_{(I,q)\text{ for higher degree monomial basis}} \longrightarrow \underbrace{\text{2nd order}}_{I:\text{ all }n}$$

Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
3. Part III: Robustness Verification (Chapter 5)
   - Lipschitz constant estimation
   - Algorithms and experiments
   - Other models of robustness verification
4. Conclusion and future works

T. Chen, J.-B. Lasserre, V. Magron, E. Pauwels (2022). Semialgebraic Optimization for Lipschitz Constants of ReLU networks, 34th Conference on Neural Information Processing Systems (NeurIPS 2020).

T. Chen, J.-B. Lasserre, V. Magron, E. Pauwels (2022). Semialgebraic Representation of Monotone Deep Equilibrium Models and Applications to Certification, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
3. Part III: Robustness Verification (Chapter 5)
   - Lipschitz constant estimation
   - Algorithms and experiments
   - Other models of robustness verification
4. Conclusion and future works

Lipschitz constant controls robustness

- Let $f : \mathcal{X} \to \mathbb{R}$:

$$C_f^p = \inf_{\mathbf{x},\mathbf{y} \in \mathcal{X}} \{C : |f(\mathbf{x}) - f(\mathbf{y})| \le C \cdot \|\mathbf{x} - \mathbf{y}\|_p\}.$$

Lipschitz constant controls robustness

- Let $f : \mathcal{X} \to \mathbb{R}$:

$$C_f^p = \inf_{\mathbf{x},\mathbf{y} \in \mathcal{X}} \{C : |f(\mathbf{x}) - f(\mathbf{y})| \leq C \cdot \|\mathbf{x} - \mathbf{y}\|_p\}.$$

- Let $C_k, C_{\bar{y}}$ be the Lip. const. of $F_k, F_{\bar{y}}$ resp.,

$$F_k(\mathbf{x}_0) - F_{\bar{y}}(\mathbf{x}_0) \leq (C_k + C_{\bar{y}})\varepsilon + F_k(\bar{\mathbf{x}}) - F_{\bar{y}}(\bar{\mathbf{x}}) := \alpha(C_k, C_{\bar{y}}, \varepsilon).$$

Lipschitz constant controls robustness

- Let $f : \mathcal{X} \to \mathbb{R}$:

$$C_f^p = \inf_{\mathbf{x},\mathbf{y} \in \mathcal{X}} \{C : |f(\mathbf{x}) - f(\mathbf{y})| \leq C \cdot \|\mathbf{x} - \mathbf{y}\|_p\}.$$

- Let $C_k, C_{\bar{y}}$ be the Lip. const. of $F_k, F_{\bar{y}}$ resp.,

$$F_k(\mathbf{x}_0) - F_{\bar{y}}(\mathbf{x}_0) \leq (C_k + C_{\bar{y}})\varepsilon + F_k(\bar{\mathbf{x}}) - F_{\bar{y}}(\bar{\mathbf{x}}) := \alpha(C_k, C_{\bar{y}}, \varepsilon).$$

- $\alpha(C_k, C_{\bar{y}}, \varepsilon) < 0 \implies \varepsilon$-robust.

Lipschitz constant computation

- If $\mathcal{X}$ is convex, f is smooth:

$$C_f^p = \sup_{\mathbf{x}\in\mathcal{X}} \|\nabla f(\mathbf{x})\|_p^* = \sup_{\mathbf{x}\in\mathcal{X}} \{\mathbf{t}^T \nabla f(\mathbf{x}) : \|\mathbf{t}\|_p \leq 1\}.$$

Lipschitz constant computation

- If $\mathcal{X}$ is convex, f is smooth:

$$C_f^p = \sup_{\mathbf{x}\in\mathcal{X}} \|\nabla f(\mathbf{x})\|_p^* = \sup_{\mathbf{x}\in\mathcal{X}} \{\mathbf{t}^T \nabla f(\mathbf{x}) : \|\mathbf{t}\|_p \leq 1\}.$$

- ⚠ ReLU network is **NON-smooth**:

$$\mathbf{x}_0 \xrightarrow{\text{linear}} \mathbf{z}_1 \xrightarrow{\boxed{\text{ReLU}}} \mathbf{x}_1 \xrightarrow{\text{linear}} \cdots \xrightarrow{\text{linear}} \mathbf{z}_L \xrightarrow{\boxed{\text{ReLU}}} \mathbf{x}_L \xrightarrow{\text{linear}} F(\mathbf{x}_0)$$

Lipschitz constant estimation: for DNN

Let $F : \mathcal{X} \to \mathbb{R}^K$ be a fully-connected DNN. Fix a label k.

$$\mathbf{x}_0 \xrightarrow{\text{linear}} \mathbf{z}_1 \xrightarrow{\text{ReLU}} \mathbf{x}_1 \xrightarrow{\text{linear}} \cdots \xrightarrow{\text{linear}} \mathbf{z}_L \xrightarrow{\text{ReLU}} \mathbf{x}_L \xrightarrow{\text{linear}} \mathbf{C}_k \mathbf{x}_L = F_k(\mathbf{x}_0)$$

Lipschitz constant estimation: for DNN

Let $F : \mathcal{X} \to \mathbb{R}^K$ be a fully-connected DNN. Fix a label k.

$$\mathbf{x}_0 \xrightarrow{\text{linear}} \mathbf{z}_1 \xrightarrow{\text{ReLU}} \mathbf{x}_1 \xrightarrow{\text{linear}} \cdots \xrightarrow{\text{linear}} \mathbf{z}_L \xrightarrow{\text{ReLU}} \mathbf{x}_L \xrightarrow{\text{linear}} \mathbf{C}_k \mathbf{x}_L = F_k(\mathbf{x}_0)$$

- Apply chain rule to F_k:

$$\begin{aligned} G_{F_k}(\mathbf{x}_0) &:= \left(\mathcal{J}^C_{\mathbf{x}_L}(\mathbf{z}_L) \cdot \mathcal{J}^C_{\mathbf{z}_L}(\mathbf{x}_{L-1}) \cdots \mathcal{J}^C_{\mathbf{x}_1}(\mathbf{z}_1) \cdot \mathcal{J}^C_{\mathbf{z}_1}(\mathbf{x}_0)\right)^T \cdot \mathbf{C}_k \\ &= \left(\prod_{i=1}^{L} \mathbf{A}_i^T \cdot \mathrm{diag}(\partial \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i)) \right) \cdot \mathbf{C}_k. \end{aligned}$$

Lipschitz constant estimation: for DNN

Let $F : \mathcal{X} \to \mathbb{R}^K$ be a fully-connected DNN. Fix a label k.

$$\mathbf{x}_0 \xrightarrow{\text{linear}} \mathbf{z}_1 \xrightarrow{\text{ReLU}} \mathbf{x}_1 \xrightarrow{\text{linear}} \cdots \xrightarrow{\text{linear}} \mathbf{z}_L \xrightarrow{\text{ReLU}} \mathbf{x}_L \xrightarrow{\text{linear}} \mathbf{C}_k \mathbf{x}_L = F_k(\mathbf{x}_0)$$

- Apply chain rule to F_k:

$$\begin{aligned} G_{F_k}(\mathbf{x}_0) &:= \left(\mathcal{J}^C_{\mathbf{x}_L}(\mathbf{z}_L) \cdot \mathcal{J}^C_{\mathbf{z}_L}(\mathbf{x}_{L-1}) \cdots \mathcal{J}^C_{\mathbf{x}_1}(\mathbf{z}_1) \cdot \mathcal{J}^C_{\mathbf{z}_1}(\mathbf{x}_0)\right)^T \cdot \mathbf{C}_k \\ &= \left(\prod_{i=1}^{L} \mathbf{A}_i^T \cdot \mathrm{diag}(\partial \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i))\right) \cdot \mathbf{C}_k. \end{aligned}$$

- Upper bound of Lipschitz constant:

$$L^p_{F_k} \leq \sup_{\mathbf{x}_0 \in \mathcal{X}} \{\mathbf{t}^T \cdot G_{F_k}(\mathbf{x}_0) : \|\mathbf{t}\|_p \leq 1\}.$$

Polynomial optimization reformulation

- Upper bound of Lipschitz constant for DNN:

$$\max \quad \mathbf{t}^T \cdot \left(\prod_{i=1}^{L} \mathbf{A}_i^T \cdot \mathrm{diag}(\mathbf{y}_i) \right) \cdot \mathbf{C}_k$$

$$\text{s.t.} \begin{cases} \|\mathbf{t}\|_p \leq 1; \\ \|\mathbf{x}_0 - \bar{\mathbf{x}}\|_p \leq \varepsilon; \\ \mathbf{y}_i = \partial \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i), \ i = 1, \ldots, L; \\ \mathbf{x}_i = \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i), \ i = 2, \ldots, L. \end{cases}$$

Polynomial optimization reformulation

- Upper bound of Lipschitz constant for DNN:

$$\max \quad \mathbf{t}^T \cdot \left(\prod_{i=1}^{L} \mathbf{A}_i^T \cdot \mathrm{diag}(\mathbf{y}_i) \right) \cdot \mathbf{C}_k$$

$$\text{s.t.} \begin{cases} \|\mathbf{t}\|_p \leq 1; \\ \|\mathbf{x}_0 - \bar{\mathbf{x}}\|_p \leq \varepsilon; \\ \mathbf{y}_i = \partial \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i), \ i = 1, \ldots, L; \\ \mathbf{x}_i = \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i), \ i = 2, \ldots, L. \end{cases}$$

- Non-convex.

Lipschitz constant estimation: for monDEQ

Let $F : \mathcal{X} \to \mathbb{R}^K$ be a fully-connected monDEQ. Fix a label k.

$$\mathbf{x}_0 \xrightarrow{\text{linear+ReLU}} \mathbf{x}_1 = \mathrm{ReLU}(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_0 + \mathbf{b}) \xrightarrow{\text{linear}} \mathbf{C}_k\mathbf{x}_1 = F_k(\mathbf{x}_0)$$

- Apply chain rule to F_k:

$$G_{F_k}(\mathbf{x}_0) := \underbrace{\left(\mathcal{J}^C_{\mathbf{x}_1}(\mathbf{x}_0)\right)}_{\text{technical}}{}^T \cdot \mathbf{C}_k.$$

Lipschitz constant estimation: for monDEQ

Let $F : \mathcal{X} \to \mathbb{R}^K$ be a fully-connected monDEQ. Fix a label k.

$$\mathbf{x}_0 \xrightarrow{\text{linear+ReLU}} \mathbf{x}_1 = \mathrm{ReLU}(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_0 + \mathbf{b}) \xrightarrow{\text{linear}} \mathbf{C}_k\mathbf{x}_1 = F_k(\mathbf{x}_0)$$

- Apply chain rule to F_k:

$$G_{F_k}(\mathbf{x}_0) := \big(\underbrace{\mathcal{J}^C_{\mathbf{x}_1}(\mathbf{x}_0)}_{\text{technical}}\big)^T \cdot \mathbf{C}_k.$$

- Upper bound of Lipschitz constant:

$$L^p_{F_k} \leq \sup_{\mathbf{x}_0 \in \mathcal{X}} \{\mathbf{t}^T \cdot G_{F_k}(\mathbf{x}_0) : \|\mathbf{t}\|_p \leq 1\}.$$

Polynomial optimization reformulation

- Upper bound of Lipschitz constant for monDEQ:

$$\max \quad \mathbf{t}^T \cdot \mathbf{B}^T \cdot \mathrm{diag}(\mathbf{y}) \cdot \mathbf{r} \qquad \text{(Lip-monDEQ)}$$

$$\text{s.t.} \begin{cases} \|\mathbf{t}\|_p \le 1; \\ \|\mathbf{x}_0 - \bar{\mathbf{x}}\|_p \le \varepsilon; \\ \mathbf{r} - \mathbf{A}^T \cdot \mathrm{diag}(\mathbf{y}) \cdot \mathbf{r} = \mathbf{C}_k; \\ \mathbf{y} = \partial \mathrm{ReLU}(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_0 + \mathbf{b}); \\ \mathbf{x}_1 = \mathrm{ReLU}(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_0 + \mathbf{b}). \end{cases}$$

Polynomial optimization reformulation

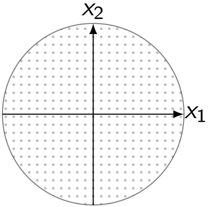
- Upper bound of Lipschitz constant for monDEQ:

$$\max \quad \mathbf{t}^T \cdot \mathbf{B}^T \cdot \mathrm{diag}(\mathbf{y}) \cdot \mathbf{r} \qquad \text{(Lip-monDEQ)}$$

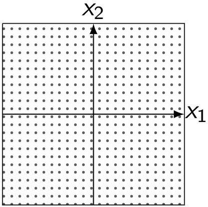
$$\text{s.t.} \begin{cases} \|\mathbf{t}\|_p \leq 1; \\ \|\mathbf{x}_0 - \bar{\mathbf{x}}\|_p \leq \varepsilon; \\ \mathbf{r} - \mathbf{A}^T \cdot \mathrm{diag}(\mathbf{y}) \cdot \mathbf{r} = \mathbf{C}_k; \\ \mathbf{y} = \partial \mathrm{ReLU}(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_0 + \mathbf{b}); \\ \mathbf{x}_1 = \mathrm{ReLU}(\mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{x}_0 + \mathbf{b}). \end{cases}$$

- Non-convex.

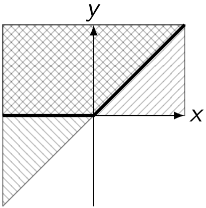
Key: semialgebraicity



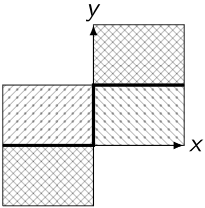
L_2 norm: $\|\mathbf{x}\|_2 \leq 1$



L_∞ norm: $\|\mathbf{x}\|_\infty \leq 1$



$y = \mathrm{ReLU}(x)$



$y \in \partial\mathrm{ReLU}(x)$

Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
3. Part III: Robustness Verification (Chapter 5)
   - Lipschitz constant estimation
   - Algorithms and experiments
   - Other models of robustness verification
4. Conclusion and future works

Algorithms

For DNN

- **LipOpt-3/4** [Latorre20]: 3rd-/4th-degree LP relaxation;
- **SHOR**: Shor's relaxation;
- **Sub-2**: 2nd-order sublevel relaxation;
- **LBS**: lower bound by random sampling.

Algorithms

For DNN

- **LipOpt-3/4** [Latorre20]: 3rd-/4th-degree LP relaxation;
- **SHOR**: Shor's relaxation;
- **Sub-2**: 2nd-order sublevel relaxation;
- **LBS**: lower bound by random sampling.

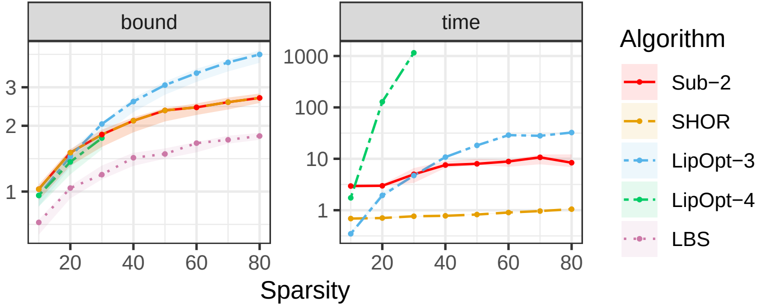
For monDEQ

- **Pab**: analytical upper bound by [Pabbaraju21];
- **SHOR**: Shor's relaxation.

Random (80,80) DNN

$$
\begin{bmatrix}
* & * & * & * & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & * & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
* & * & * & * & * & * & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & * & * & * & * & * & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & * & * & * & * & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & * & * & * & * & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & * & * & * & * & * & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & * & * & * & * & * & * \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & * & * & * & * & * & * \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & * & * & * & * & * \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & * & * & * & *
\end{bmatrix}
$$

Random (80,80) DNN



MNIST (784, 500) DNN

Upper bounds and running time

	Sub-2	**SHOR**	**LipOpt-3**	**LBS**
bound	14.56	17.85	OfM	9.69
time (s)	12246	2869	OfM	-

MNIST (784, 87) monDEQ

Upper bounds and solving time

	L_2		L_∞	
	bound	time (s)	bound	time (s)
Pab	4.80	-	824.14	-
SHOR	4.67	1756	108.84	1898

Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
3. Part III: Robustness Verification (Chapter 5)
   - Lipschitz constant estimation
   - Algorithms and experiments
   - Other models of robustness verification
4. Conclusion and future works

Models for robustness verification

$$\left\{\begin{array}{c} \text{Lipschitz model} \\ \boxed{\max \quad \|G_{F_k}(\mathbf{x}_0)\|_*} \\ \\ \text{robustness model} \\ \boxed{\max \quad F_k(\mathbf{x}_0) - F_{\bar{y}}(\mathbf{x}_0)} \\ \\ \text{ellipsoid model} \\ \boxed{\min \;\; \mathrm{Vol}(\mathcal{E}) \to \max \;\; \log\det(\mathbf{Q})} \end{array}\right. \qquad \text{s.t.} \begin{cases} L_p \\ \mathrm{ReLU} \\ \partial\mathrm{ReLU} \\ \text{linear} \end{cases}$$

Comparison of three models

Take (p_0, p_1) monDEQ for K-classification, we need to verify N examples.

Comparison of three models

Take (p_0, p_1) monDEQ for K-classification, we need to verify N examples.

	robustness model	ellipsoid model	Lipschitz model
# of variables	$p_0 + p_1$	$p_0 + p_1$	$2p_0 + 3p_1$
# of experiments	N	N	K

Numerical results: for MNIST (784, 87) monDEQ

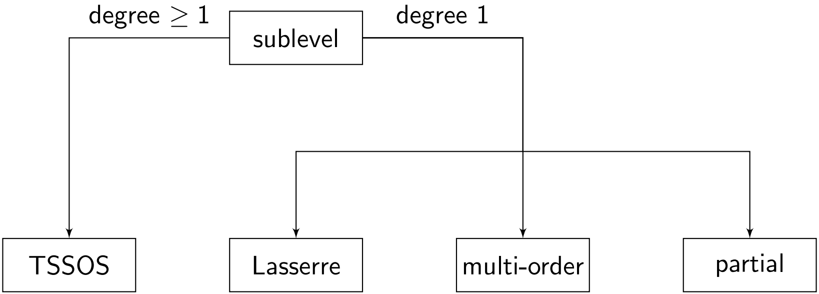
Ratio of verified examples (per 100) and running time.

norm	perturbation	robustness model (1350s/ex., 37.5h)	ellipsoid model (500s/ex., 14h)	Lipschitz model (0.5h)
L_2	0.1	99%	99%	91%
L_∞	0.01	99%	92%	24%

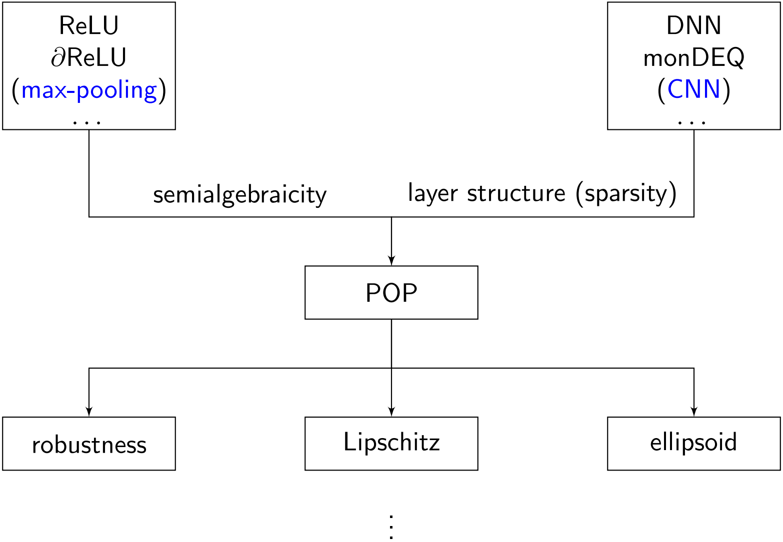
Outline

1. Part I: Neural Network Verification (Chapter 1-2)
2. Part II: Moment-SOS Relaxation (Chapter 3-4)
3. Part III: Robustness Verification (Chapter 5)
4. Conclusion and future works

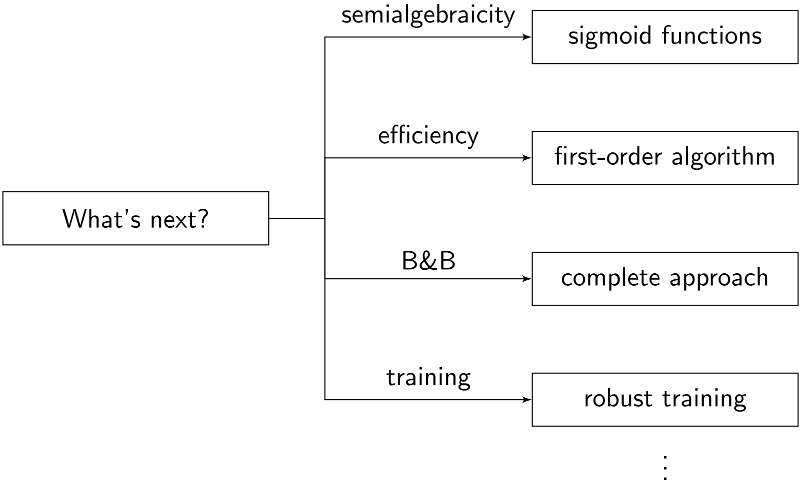
Conclusion: optimization perspective



Conclusion: deep learning perspective



Future works



Thank you!

Pros and cons of SDP relaxation

- Tightness: SDP $>$ QP $>$ LP $>$ BP;
- Efficiency: SDP $<$ QP $<$ LP $<$ BP;
- Adaptivity (norms, layer structures): SDP $>$ BP.

Table 3: **Verified accuracy (%)** and avg. per-example verification time (s) on 7 models from SDP-FO [9]. CROWN/DeepPoly are fast but loose bound propagation based methods, and they cannot be improved with more running time. SDP-FO uses stronger semidefinite relaxations, which can be very slow and sometimes has convergence issues. PRIMA, a concurrent work, is the state-of-the-art relaxation barrier breaking method; we did not include kPoly and OptC2V because they are weaker than PRIMA (see Table 2).

Dataset	Model	CROWN/DeepPoly		SDP-FO [9]*		PRIMA [26]		β-CROWN FSB		Upper
$\epsilon = 0.3$ and $\epsilon = 2/255$		Verified%	Time (s)	Ver.%	Time(s)	Ver.%	Time(s)	Ver.%	Time(s)	bound
MNIST	CNN-A-Adv	1.0	0.1	43.4	>20h	44.5	135.9	**70.5**	21.1	76.5
CIFAR	CNN-B-Adv	21.5	0.5	32.8	>25h	38.0	343.6	**46.5**	32.2	65.0
	CNN-B-Adv-4	43.5	0.9	46.0	>25h	53.5	43.8	**54.0**	11.6	63.5
	CNN-A-Adv	35.5	0.6	39.6	>25h	41.5	4.8	**44.0**	5.8	50.0
	CNN-A-Adv-4	41.5	0.7	40.0	>25h	45.0	4.9	**46.0**	5.6	49.5
	CNN-A-Mix	23.5	0.4	39.6	>25h	37.5	34.3	**41.5**	49.6	53.0
	CNN-A-Mix-4	38.0	0.5	47.8	>25h	48.5	7.0	**50.5**	5.9	57.5

* SDP-FO results are directly from their paper due to its very long running time (>20h per example). † PRIMA experiments were done using commit 396dc7a, released on June 4, 2021. PRIMA and β-CROWN FSB results are on the same set of 200 examples (first 200 examples of CIFAR-10 dataset) and we don't run verifiers on examples that are classified incorrectly or can be attacked by a 200-step PGD. β-CROWN uses 1 GPU and 1 CPU; PRIMA uses 1 GPU and 20 CPUs.

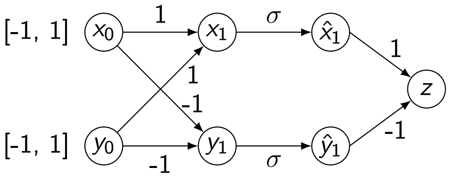
Table 1: Adversarial robustness of MNIST classifiers to perturbations of ϵ in a l_∞-norm. We run each algorithm on the first 100 test set samples from the MNIST dataset. The times are expressed in seconds.

Network	ϵ	Certified Accuracy ↑				Average Time [s] ↓			
		β-CROWN	HQ-CRAN	PRIMA	GPUPoly	β-CROWN	HQ-CRAN	PRIMA	GPUPoly
PGD-2x[20]	$2/255$	88%	88%	88%	88%	0.017	0.029	0.073	0.002
	$4/255$	88%	88%	88%	88%	0.017	0.157	0.075	0.005
	$8/255$	81%	81%	81%	81%	0.181	1.474	0.115	0.008
	$16/255$	52%	52%	32%	27%	1.820	6.331	0.369	0.013
MLP-2x[20]	$2/255$	97%	97%	97%	97%	0.078	0.099	0.022	0.004
	$4/255$	96%	96%	96%	96%	0.063	0.533	0.028	0.004
	$8/255$	78%	78%	76%	60%	0.384	3.791	0.244	0.006
	$16/255$	26%	26%	8%	3%	2.140	18.487	0.544	0.022

Some software for NN verification

- α, β-CROWN (bound propagation);
- ERAN (abstract interpretation);
- SDP-FO (first-order SDP);
- Gurobi Machine Learning (adversarial attack);
- NCVX (adversarial attack).

Complete approach: SAT/SMT

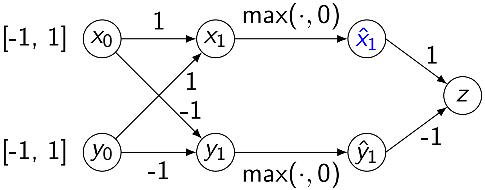


Property: $z > 0$

Satisfiability problem [Katz17, Ehlers17, Bunel18]

$$
\begin{aligned}
-1 \leq x_0 \leq 1, &\quad -1 \leq y_0 \leq 1, \\
x_1 = x_0 + y_0, &\quad y_1 = -x_0 - y_0, \\
\hat{x}_1 = \sigma(x_1), &\quad \hat{y}_1 = \sigma(y_1), \\
z = \hat{x}_1 &- \hat{y}_1, \\
z &\leq 0.
\end{aligned}
$$

Complete approach: MILP



Property: $z > 0$

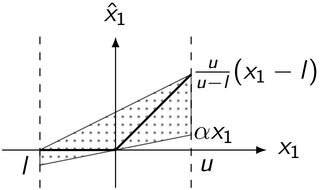
Mixed integer linear programming formulation [Tjeng19]

$$\hat{x}_1 \geq 0, \quad \hat{x}_1 \leq u \cdot \delta, \qquad \delta \in \{0, 1\}.$$
$$\hat{x}_1 \geq x_1, \quad \hat{x}_1 \leq x_1 - l \cdot (1 - \delta),$$

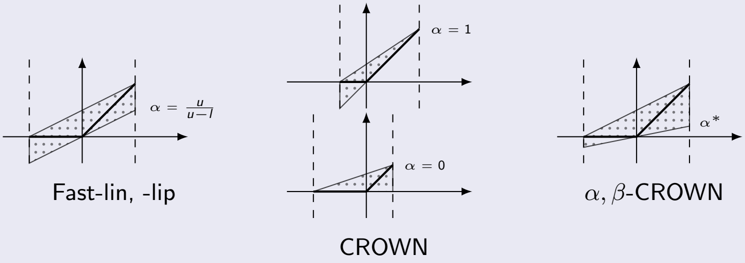
Mixed integer linear programming formulation [Lomuscio17, Cheng17]

$$\hat{x}_1 \geq 0, \quad \hat{x}_1 \leq M \cdot \delta, \qquad \delta \in \{0, 1\}.$$
$$\hat{x}_1 \geq x_1, \quad \hat{x}_1 \leq x_1 - M \cdot (1 - \delta),$$

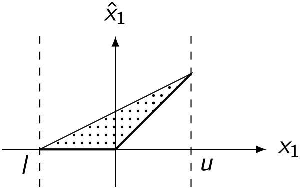
Incomplete approach: bound propagation



CROWN family [Weng18, Zhang18, Xu21, Wang21]



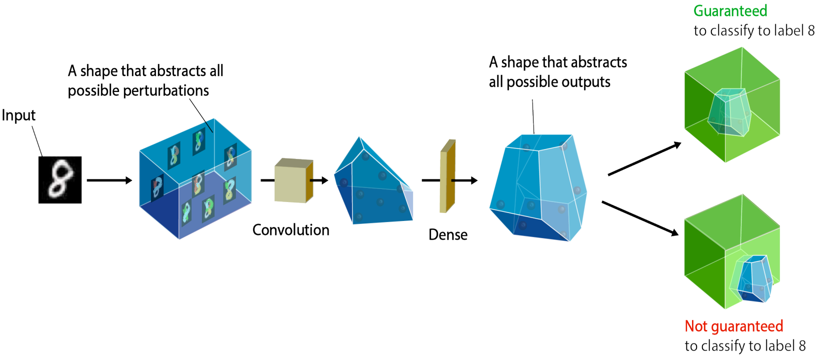
Incomplete approach: triangular relaxation



Linear programming formulation [Ehlers17, Wong18]

$$\hat{x}_1 \geq 0, \quad \hat{x}_1 \geq x_1, \quad \hat{x}_1 \leq \frac{u}{u-l}(x_1 - l).$$

Incomplete approach: abstract interpretation (AI)



Incomplete approach: Lagrangian dual

Primal network $\quad \mathbf{x}_0 \xrightarrow{\mathbf{A}} \mathbf{x}_1 \xrightarrow{\sigma} \hat{\mathbf{x}}_1$

Dual network $\quad \mathbf{v}_0 \xleftarrow{\mathbf{A}^T} \mathbf{v}_1 \xleftarrow{\sigma^*} \hat{\mathbf{v}}_1$

$$\langle \sigma^*(\mathbf{v}_1), \mathbf{x}_1 \rangle = \langle \mathbf{v}_1, \sigma(\mathbf{x}_1) \rangle$$

Primal

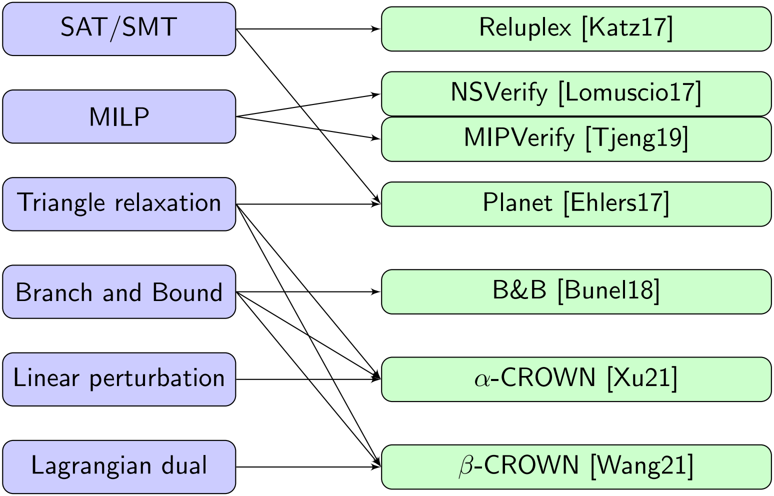
$$\min_{\mathbf{x}_0, \mathbf{x}_1, \hat{\mathbf{x}}_1} \quad \{\mathbf{c}^T \hat{\mathbf{x}}_1 : \hat{\mathbf{x}}_1 = \sigma(\mathbf{x}_1),\ \mathbf{x}_1 = \mathbf{A}\mathbf{x}_0 + \mathbf{b}\}.$$

Dual

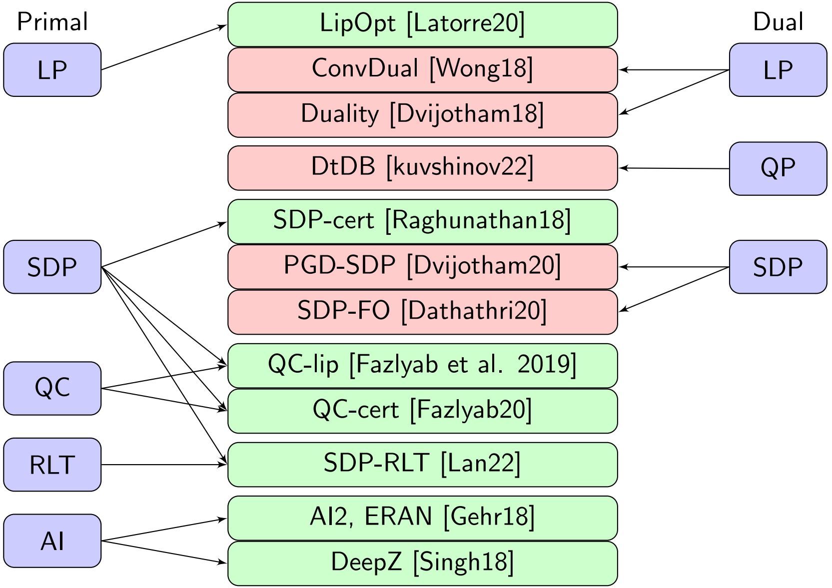
$$\begin{aligned} & \max_{\mathbf{v}_0, \mathbf{v}_1, \hat{\mathbf{v}}_1} \min_{\mathbf{x}_0, \mathbf{x}_1, \hat{\mathbf{x}}_1} \quad \mathbf{c}^T \hat{\mathbf{x}}_1 - \mathbf{v}_1^T \cdot (\hat{\mathbf{x}}_1 - \sigma(\mathbf{x}_1)) - \hat{\mathbf{v}}_1^T \cdot (\mathbf{x}_1 - \mathbf{A}\mathbf{x}_0 - \mathbf{b}) \\ = & \max_{\mathbf{v}_0, \mathbf{v}_1, \hat{\mathbf{v}}_1} \min_{\mathbf{x}_0, \mathbf{x}_1, \hat{\mathbf{x}}_1} \quad \hat{\mathbf{x}}_1^T \cdot (\mathbf{c} - \mathbf{v}_1) + \mathbf{x}_1^T \cdot (\sigma^*(\mathbf{v}_1) - \hat{\mathbf{v}}_1) + \mathbf{x}_0^T \mathbf{A}^T \hat{\mathbf{v}}_1 + \mathbf{b}^T \hat{\mathbf{v}}_1. \end{aligned}$$

- **convex** in dual variables;
- algorithm is **anytime**;
- optimize **independently**.

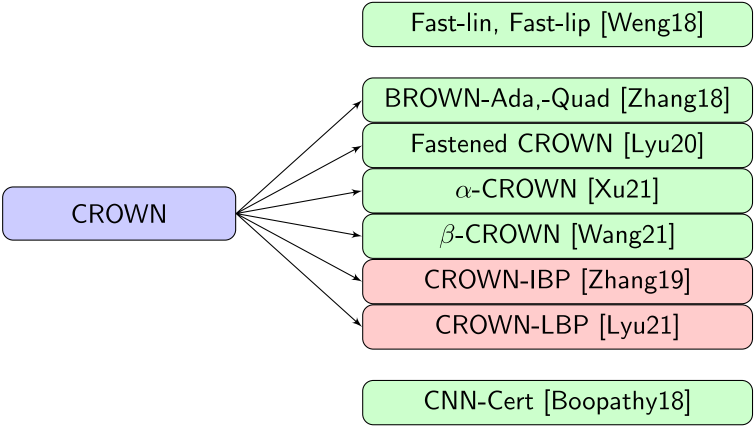
Complete approaches



Incomplete approaches: convex relaxations

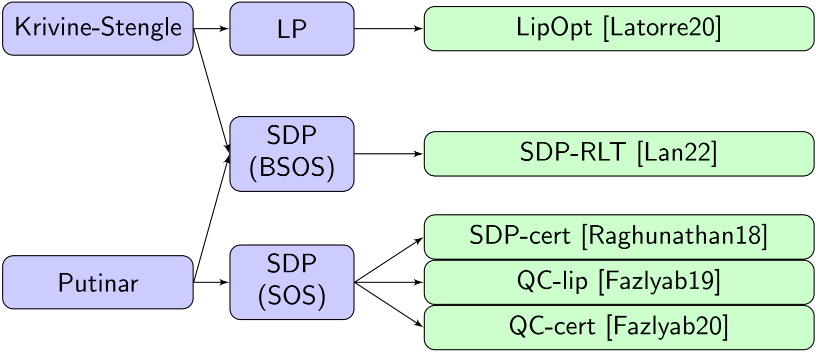


Incomplete approaches: bound propagation (BP)



- SOTA: α, β-CROWN wins the **Top Highest Score Award** in VNN-COMP21 (https://sites.google.com/view/vnn2021).

Positiveness-based approaches: LP/SDP relaxations



Some links

- ICML **W**orkshop on **F**ormal **V**erification of **M**achine **L**earning (WFVML 2022): https://sites.google.com/view/vnn2022.
- International **V**erification of **N**eural **N**etworks **COM**petition (VNN-COMP22): https://www.ml-verification.com/.

Odd-One-Out:

1. alpha-beta-CROWN: 777.07
2. VeriNet: 708.47
3. oval: 588.96 (GPU) Oxford
4. ERAN: 586.88
5. Marabou: 340.92
6. Debona: 209.05
7. venus2: 194.57
8. nnenum: 189.79
9. nnv: 59.16
10. NeuralVerification.jl: 48.06
11. Neural-Network-Reach: 25.45
12. DNNF: 24.99
13. randgen: 1.85

Voting:

1. alpha-beta-CROWN: 776.67
2. VeriNet: 709.21
3. ERAN: 588.71 (GPU) ETH / Illinois
4. oval: 588.38
5. Marabou: 302.14
6. Debona: 208.7
7. venus2: 194.56
8. nnenum: 194.21
9. nnv: 59.05
10. NeuralVerification.jl: 48.06
11. DNNF: 24.93
12. Neural-Network-Reach: 20.08
13. randgen: 1.84

Zoom in: TSSOS v.s. sublevel

$$-\frac{1}{\sqrt{2}} = \min_{\mathbf{x}\in\mathbb{R}^2} \quad \{f(\mathbf{x}) = x_1x_2 : g(\mathbf{x}) = 1 - x_1^4 - x_2^4 \geq 0\}.$$

- term sparsity:

$$f + \frac{1}{\sqrt{2}} = \sigma_{01}(x_1, x_2)^{(2)} + \sigma_{02}(1, x_1x_2)^{(2)} + \sigma_{03}(1, x_1^2, x_2^2)^{(2)} + \sigma_1^{(0)} \cdot g.$$

- sublevel structure:

$$2\sqrt{2}f + 2 = \underbrace{(1 + \sqrt{2}x_1x_2)^2}_{\sigma_{01}(1,x_1x_2)^{(2)}} + \underbrace{(x_1^2 - x_2^2)^2}_{\sigma_{02}(x_1^2,x_2^2)^{(2)}} + \underbrace{1}_{\sigma_1^{(0)}} \cdot g.$$

$$2\sqrt{2}f + 2 = \underbrace{\sqrt{2}(x_1 + x_2)^2}_{\sigma_{01}(x_1,x_2)^{(2)}} + \underbrace{(\frac{\sqrt{2}}{2} - x_1^2)^2}_{\sigma_{02}(1,x_1^2)^{(2)}} + \underbrace{(\frac{\sqrt{2}}{2} - x_2^2)^2}_{\sigma_{03}(1,x_2^2)^{(2)}} + \underbrace{1}_{\sigma_1^{(0)}} \cdot g.$$

Max-Cut problems

$$\max_{\mathbf{x}\in\{1,-1\}^n} \quad \mathbf{x}^T\mathbf{L}\mathbf{x}$$

Subset design for $(l, 1)$

Ord$(l,1)$: set $\{x_i, \ldots, x_{i+l-1}\}$.

	upper bound	#var	$l = 0$ (Shor) / 4 / 6 / 8, $q = 1$			
			upper bounds			
g_200	4472.3 (TSSOS)	5005	4584.6	4353.3	4228.1	**4132.2**
G32	1398 (best known)	2000	1567.6	1433.4	1415.9	**1415.9**

MIQCQP problems

$$\min_{\mathbf{x}\in\{0,1\}^{100}} \quad \{\mathbf{x}^T\mathbf{Q}_0\mathbf{x} + \mathbf{b}_0^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{b}_i^T\mathbf{x} \leq c_i,\ \ i = 1, \dots, p\}$$

Subset design for $(l, 1)$

- $\mathbf{Ord}(l, 1)$ for $x_i \in \{0, 1\}$ and $\mathbf{x}^T\mathbf{x} + \mathbf{b}_i^T\mathbf{x} \leq c_i$: set $\{x_i, \dots, x_{i+l-1}\}$;
- $\mathbf{Ord}(l, 1)$ for $\mathbf{A}\mathbf{x} = \mathbf{b}$: set $\{x_1, \dots, x_l\}$.

	solution	#var	$l = 0$ (Shor) / 4 / 6 / 8, $q = 1$			
			lower bounds			
gka1d	-6333	100	-6592.7	-6475.3	-6403.1	**-6369.6**

Matrix-product technique

If $\mathbf{J} \in \mathcal{J}^{C}_{\mathbf{x}_1}(\mathbf{x}_0)$, it is easy to show that $\mathbf{J} = \mathrm{diag}(\mathbf{s}) \cdot (\mathbf{A} \cdot \mathbf{J} + \mathbf{B})$.
Then

$$\mathbf{J} = (\mathbf{I} - \mathrm{diag}(\mathbf{s}) \cdot \mathbf{A})^{-1} \cdot \mathrm{diag}(\mathbf{s}) \cdot \mathbf{B}.$$

Denote by $\mathbf{r}^T = \mathbf{C}_k^T \cdot (\mathbf{I} - \mathrm{diag}(\mathbf{s}) \cdot \mathbf{A})^{-1}$, then

$$\mathbf{r} - \mathbf{A}^T \cdot \mathrm{diag}(\mathbf{s}) \cdot \mathbf{r} = \mathbf{C}_k.$$

Hence

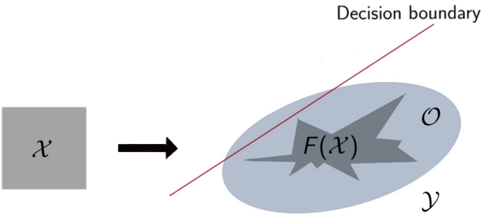
$$\mathbf{C}_k^T \cdot \mathbf{J} = \mathbf{C}_k^T \cdot (\mathbf{I} - \mathrm{diag}(\mathbf{s}) \cdot \mathbf{A})^{-1} \cdot \mathrm{diag}(\mathbf{s}) \cdot \mathbf{B} = \mathbf{r}^T \cdot \mathrm{diag}(\mathbf{s}) \cdot \mathbf{B}.$$

Hence the objective $\mathbf{t}^T \mathbf{J}^T \mathbf{C}_k = \mathbf{t}^T \mathbf{B}^T \mathrm{diag}(\mathbf{s}) \cdot \mathbf{r}$.

Basic idea

Over-approximation of the output domain

find good shape $\mathcal{E} \supseteq F(\mathcal{X})$ (eg., polytope, zonotope, ellipsoid), verify $\mathcal{E} \subseteq \mathcal{Y}$ instead of $F(\mathcal{X}) \subseteq \mathcal{Y}$.



Ellipsoidal propagation

Ellipsoidal propagation

Ellipsoid

An *ellipsoid* in $\mathbb{R}^n$ has the form

$$\mathcal{E} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{Q}\mathbf{x} + \mathbf{q}\|_2 \leq 1\}.$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{q} \in \mathbb{R}^n$.

Minimum volume ellipsoid

$$\min_{\mathcal{E}} \quad \{\mathrm{Vol}(\mathcal{E}) : F(\mathcal{X}) \subseteq \mathcal{E}\}.$$

Moment-SOS relaxation

Log-det maximization problem

$$\max_{\mathbf{Q}\in\mathbb{R}^{K\times K},\, \mathbf{q}\in\mathbb{R}^{K}} \quad \log\det(\mathbf{Q}) \tag{Ellip}$$

$$\text{s.t.} \begin{cases} \mathbf{x}_0 \in \mathcal{X}, & \text{(input constraint)} \\ \mathbf{x}_i = \mathrm{ReLU}(\mathbf{A}_i\mathbf{x}_{i-1} + \mathbf{b}_i), & \text{(ReLU constraint)} \\ 1 - \|\mathbf{Q}\mathbf{x}_L + \mathbf{q}\|_2^2 \geq 0. & \text{(output constraint)} \end{cases}$$

Moment-SOS relaxation to (Ellip)

$$\max_{\mathbf{Q}\in\mathbb{R}^{n\times n},\, \mathbf{q}\in\mathbb{R}^{n}} \quad \log\det(\mathbf{Q})$$

$$\text{s.t.} \quad 1 - \|\mathbf{Q}\mathbf{x}_L + \mathbf{q}\|_2^2 = \sigma_0 + \sigma_1 \cdot \text{(input constr)} + \sigma_2 \cdot \text{(ReLU constr)}$$

Algorithms

$$1 - \|\mathbf{Q}\mathbf{x}_L + \mathbf{q}\|_2^2 = \sigma_0 + \sigma_1 \cdot (\text{input constr}) + \sigma_2 \cdot (\text{ReLU constr}).$$

Subset design

- **Ord**$(l, 1)$ for input constraint: set $\{x_0^k, \ldots, x_0^{k+l-1}\}$;
- **Cyc**(l) for ReLU constraint: set

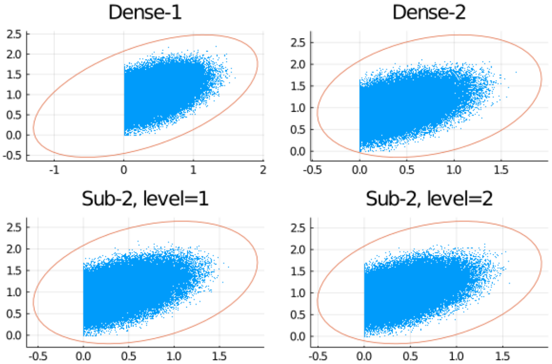
$$\{x_i^1, \ldots, x_i^l; x_{i+1}^{(j)}\}, \ldots, \{x_i^{p_i}, \ldots, x_i^{p_i+l-1}; x_{i+1}^{(j)}\}.$$

Algorithms

- **Dense-d**: d-th order dense relaxation for $d = 1, 2$;
- **Sub-2**: 2nd-order sublevel relaxation.

Random (20,2) DNN

	Dense-1	Dense-2	Sub-2	
			level 1	level 2
value	0.48	**0.62**	**0.60**	**0.61**
time (s)	0.06	216.39	**5.96**	**8.15**



MNIST (784,87) MON

Ratios of verified examples.

Norm	ε	Lipschitz	ellipsoid
L_2	0.1	91%	99%
L_∞	0.01	24%	92%

Robustness problem

Fix $\bar{\mathbf{x}}$ and $k \neq \bar{y}$:

$$\max \quad F_k(\mathbf{x}_0) - F_{\bar{y}}(\mathbf{x}_0) = (\mathbf{C}_k - \mathbf{C}_{\bar{y}})\mathbf{x}_N$$
$$\text{s.t.} \begin{cases} \mathbf{x}_i = \sigma(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i), \ i = 1, \dots, N, \\ \|\mathbf{x}_0 - \bar{\mathbf{x}}\| \leq \varepsilon. \end{cases}$$

$$\Updownarrow$$

$$\max \quad \mathbf{c}\mathbf{x}_N \tag{Cert}$$
$$\text{s.t.} \begin{cases} \mathbf{x}_i = \sigma(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i), \ i = 1, \dots, N, \\ \|\mathbf{x}_0 - \bar{\mathbf{x}}\| \leq \varepsilon. \end{cases}$$

where $\mathbf{c} = \mathbf{C}_k - \mathbf{C}_{\bar{y}}$.

- **NP-hard** for $\sigma = \mathrm{ReLU}$.

Verification criterions

Robustness model (Cert) $\longrightarrow \delta_k^{cert}$

Criterion I: $\delta_k^{cert} < 0$, for all $k \neq \bar{y}$.

Lipschitz model (Lip) $\longrightarrow L_k$

Criterion II: $F_{\bar{y}}(\bar{\mathbf{x}}) - F_k(\bar{\mathbf{x}}) \geq (L_k + L_{\bar{y}})\varepsilon$, for all $k \neq \bar{y}$.

Ellipsoid model (Ellip) $\longrightarrow \mathcal{E}$

Define $\delta_k^{ellip} := \max\{x^{(i)} - x^{(\bar{y})} : \mathbf{x} \in \mathcal{E}\}$

Criterion III: $\delta_k^{ellip} < 0$ for all $k \neq \bar{y}$.

Criterion I-III $\Longrightarrow$ ε-robust.

Numerical results: for MNIST (784, 500) DNN

- Ratios of certified examples of a well-trained $(80, 80)$ network:

ε	0.01	0.02	0.03	0.04	0.05	0.06	0.07
Sub-2	87.51%	75.02%	62.46%	49.89%	37.22%	24.36%	8.15%
LipOpt-3	69.03%	37.84%	4.78%	0.15%	0%	0%	0%

- Ratios of certified examples of the MNIST network (784, 500) network by **Sub-2**:

ε	0.01	0.02	0.04	0.06	0.08	0.1
Ratios	98.80%	97.24%	92.84%	87.10%	78.34%	67.63%