Semialgebraic Optimization for Lipschitz Constants of ReLU Networks

Tong Chen

tchen@laas.fr

joint work with J.-B. Lasserre, V. Magron and E. Pauwels

August 11, 2021

Outline

Deep learning: neural network and its robustness

From deep learning to polynomial optimization

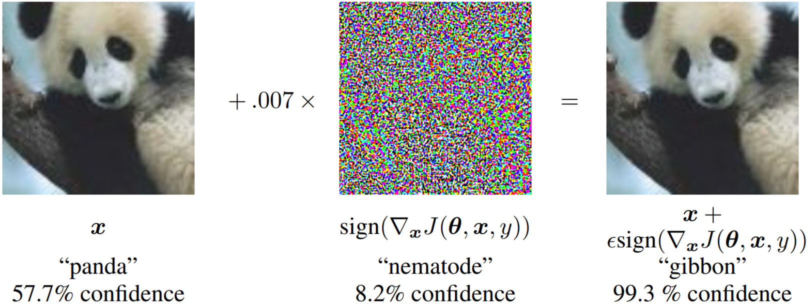
Lipschitz constant of neural network

Heuristic relaxation for nearly sparse POP

Numerical results

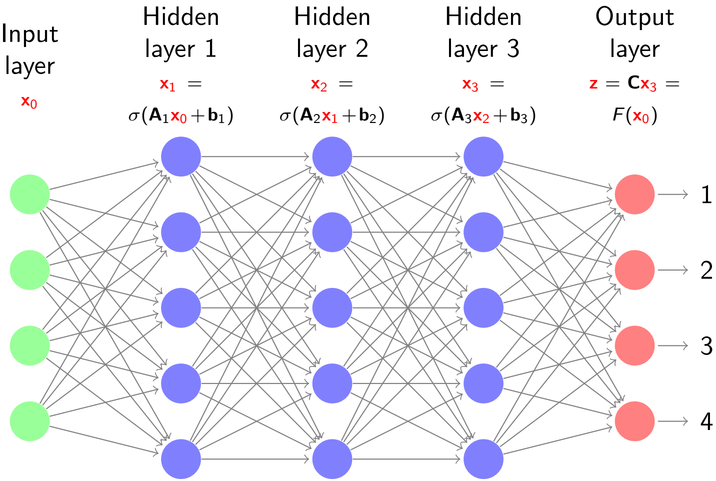
Outline

Robustness of neural network



Adversarial example of neural network, Ian Goodfellow et al., 2015.

Architecture of neural networks



Fully-connected neural network F with activation function σ.

Mathematical interpretation

For a network F with L hidden layers and K labels:

- Output of input $\mathbf{x}_0$: $F(\mathbf{x}_0) = \mathbf{C}\mathbf{x}_L, \mathbf{x}_i = \sigma(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i), i = 1, \ldots, L$.
- Prediction of input $\mathbf{x}_0$: $y(\mathbf{x}_0) = \arg\max_{k=1,\ldots,K} F(\mathbf{x}_0)_k$.
- Fix an input $\bar{\mathbf{x}}_0$, the network F is ε-**robust** (w.r.t. norm $\|\cdot\|$) at $\bar{\mathbf{x}}_0$: for any input $\mathbf{x}_0$ such that $\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\| \leq \varepsilon$,

$$y(\mathbf{x}_0) = y(\bar{\mathbf{x}}_0),$$

$$\Updownarrow$$

$$F(\mathbf{x}_0)_k \leq F(\mathbf{x}_0)_{y(\bar{\mathbf{x}}_0)}, \forall k \neq y(\bar{\mathbf{x}}_0),$$

$$\Updownarrow$$

$$F(\mathbf{x}_0)_k - F(\mathbf{x}_0)_{y(\bar{\mathbf{x}}_0)} \leq 0, \forall k \neq y(\bar{\mathbf{x}}_0),$$

- Robustness verification: maximize $F(\mathbf{x}_0)_k - F(\mathbf{x}_0)_{y(\bar{\mathbf{x}}_0)}, \forall k \neq y(\mathbf{x}_0)$.

Optimization reformulation

Fix an input $\bar{\mathbf{x}}_0$ and label $k \neq y(\bar{\mathbf{x}}_0)$:

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

$$\Updownarrow$$

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

where $\mathbf{c} = \mathbf{C}_k - \mathbf{C}_{y(\bar{\mathbf{x}}_0)}$.

Outline

Robustness certification problem

- Take $\sigma(x) = \mathrm{ReLU}(x) = \max(0, x)$.
- Take $\| \cdot \| = \| \cdot \|_p$ for $p = 2, \infty$.

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

Semialgebraicity of L_p norm and ReLU function

L_p norm for $p = 2, \infty$:

- $\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\|_2 \leq \varepsilon \Leftrightarrow (\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T(\mathbf{x}_0 - \bar{\mathbf{x}}_0) \leq \varepsilon^2$
- $\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\|_\infty \leq \varepsilon \Leftrightarrow (\mathbf{x}_0 - \bar{\mathbf{x}}_0)^2 \leq \varepsilon^2$

ReLU function:

- $u = \mathrm{ReLU}(x) \Leftrightarrow u(u - x) = 0, u \geq x, u \geq 0$

POP (Raghunathan et al, 2018):

$$\begin{aligned} \max \quad & \mathbf{c}\mathbf{x}_L \\ \text{s.t.} \quad & \begin{cases} \mathbf{x}_i(\mathbf{x}_i - \mathbf{A}_i\mathbf{x}_{i-1} - \mathbf{b}_i) = 0, \mathbf{x}_i \geq \mathbf{A}_i\mathbf{x}_{i-1} + \mathbf{b}_i, \mathbf{x}_i \geq 0, i = 1, \ldots, L \\ (\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T(\mathbf{x}_0 - \bar{\mathbf{x}}_0) \leq \varepsilon^2 \qquad (\text{or } (\mathbf{x}_0 - \bar{\mathbf{x}}_0)^2 \leq \varepsilon^2) \end{cases} \end{aligned}$$

Outline

Why Lipschitz constant?

- Lipschitz constant implies robustness: let L_1 be the Lipschitz constant of $F(\cdot)_k$ and L_2 the Lipschitz constant of $F(\cdot)_{y(\bar{\mathbf{x}}_0)}$,

$$\begin{aligned}
& F(\mathbf{x}_0)_k - F(\mathbf{x}_0)_{y(\bar{\mathbf{x}}_0)} \\
=& F(\mathbf{x}_0)_k - F(\bar{\mathbf{x}}_0)_k + F(\bar{\mathbf{x}}_0)_k - F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} + F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} - F(\mathbf{x}_0)_{y(\bar{\mathbf{x}}_0)} \\
\leq& |F(\mathbf{x}_0)_k - F(\bar{\mathbf{x}}_0)_k| + F(\bar{\mathbf{x}}_0)_k - F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} + |F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} - F(\mathbf{x}_0)_{y(\bar{\mathbf{x}}_0)}| \\
\leq& L_1\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\| + L_2\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\| + F(\bar{\mathbf{x}}_0)_k - F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} \\
\leq& (L_1 + L_2)\varepsilon + F(\bar{\mathbf{x}}_0)_k - F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)}
\end{aligned}$$

- $(L_1 + L_2)\varepsilon + F(\bar{\mathbf{x}}_0)_k - F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} < 0 \Rightarrow \varepsilon$-robust.
- Lipschitz training, Lipschitz bounded network.

Lipschitz constant of a general function

Let $f : \mathcal{X} \to \mathbb{R}$ be a function defined on $\mathcal{X} \subseteq \mathbb{R}^n$.

- $L_f^{\|\cdot\|} = \inf\{L : \forall \mathbf{x}, \mathbf{y} \in \mathcal{X}, |f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|\}$

If $\mathcal{X}$ is convex, f is differentiable,

- $L_f^{\|\cdot\|} = \sup\{\|\nabla_{\mathbf{x}} f\|_* : \mathbf{x} \in \mathcal{X}\} = \sup\{\mathbf{t}^T \nabla_{\mathbf{x}} f : \|\mathbf{t}\| \leq 1, \mathbf{x} \in \mathcal{X}\}$

Lipschitz constant of neural network

Let $F : \mathcal{X} \to \mathbb{R}^K$ be a fully-connected neural network.

- Fix a label $k \in \{1, \dots, K\}$.
- Let $f(\mathbf{x}_0) = F(\mathbf{x}_0)_k = \mathbf{C}_k \mathbf{x}_L =: \mathbf{c}^T \mathbf{x}_L, \mathbf{x}_i = \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i)$.
- By the Chain Rule (formal calculation):

$$
\begin{aligned}
\nabla_{\mathbf{x}_0} f &= \prod_{i=1}^{L} \nabla_{\mathbf{x}_{i-1}} \mathbf{x}_i \cdot \mathbf{c} = \prod_{i=1}^{L} \nabla_{\mathbf{x}_{i-1}} \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i) \cdot \mathbf{c} \\
&= \prod_{i=1}^{L} \nabla_{\mathbf{x}_{i-1}} (\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i) \cdot \nabla_{\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i} \mathrm{ReLU} \cdot \mathbf{c} \\
&= \prod_{i=1}^{L} \mathbf{A}_i^T \cdot \nabla_{\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i} \mathrm{ReLU} \cdot \mathbf{c} \\
&= \prod_{i=1}^{L} \mathbf{A}_i^T \cdot \mathrm{diag}(\partial \mathrm{ReLU}(\mathbf{A}_i \mathbf{x}_{i-1} + \mathbf{b}_i)) \cdot \mathbf{c}
\end{aligned}
$$

Lipschitz constant of neural network

- Recall: if f is differentiable,

$$L_f^{\|\cdot\|} = \sup\{\|\nabla_{\mathbf{x}} f\|_* : \mathbf{x} \in \mathcal{X}\} = \sup\{\mathbf{t}^T \nabla_{\mathbf{x}} f : \|\mathbf{t}\| \leq 1, \mathbf{x} \in \mathcal{X}\}$$

- For neural network, $f(\mathbf{x}_0) = \mathbf{C}_k \mathbf{x}_L$ is not differentiable, but if we define $\partial\mathrm{ReLU}(x) = 0$ for $x < 0$, 1 for $x > 0$, and $\{0, 1\}$ for $x = 0$,

$$L_f^{\|\cdot\|} \leq \sup\{\|\nabla_{\mathbf{x}_0} f\|_* : \mathbf{x}_0 \in \mathcal{X}\} = \sup\{\mathbf{t}^T \nabla_{\mathbf{x}_0} f : \|\mathbf{t}\| \leq 1, \mathbf{x}_0 \in \mathcal{X}\}$$

- For robustness certification, an upper bound of Lipschitz constant is enough: if $\tilde{L}_1 \geq L_1, \tilde{L}_2 \geq L_2$,

$$(\tilde{L}_1 + \tilde{L}_2)\varepsilon + F(\bar{\mathbf{x}}_0)_k - F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} < 0$$

$$\Downarrow$$

$$(L_1 + L_2)\varepsilon + F(\bar{\mathbf{x}}_0)_k - F(\bar{\mathbf{x}}_0)_{y(\bar{\mathbf{x}}_0)} < 0$$

Optimization reformulation

- Semialgebraicity of $\partial\mathrm{ReLU}$:

$$u = \partial\mathrm{ReLU}(x) \Leftrightarrow u(u-1) = 0, (u - \frac{1}{2})x \geq 0$$

- Upper bound of Lipschitz constant of $f(\mathbf{x}_0) = \mathbf{c}^T\mathbf{x}_L$:

$$\max \quad \mathbf{t}^T \nabla_{\mathbf{x}_0} f = \mathbf{t}^T \cdot \prod_{i=1}^{L} \mathbf{A}_i^T \cdot \mathrm{diag}(\mathbf{u}_i) \cdot \mathbf{c}$$

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

Outline

Nearly sparse POP

- Take F as a 1-hidden layer network with parameters $\mathbf{A} \in \mathbb{R}^{p \times p}, \mathbf{b} \in \mathbb{R}^{p}, \mathbf{C} \in \mathbb{R}^{K \times p}$
- Take $\|\cdot\| = \|\cdot\|_\infty$
- Fix an input $\bar{\mathbf{x}}$, a label k and let $\mathbf{c} = \mathbf{C}_k^T$
- Upper bound of Lipschitz constant of $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x}_1$,

$$\max \quad \mathbf{t}^T \cdot \mathbf{A}^T \cdot \mathrm{diag}(\mathbf{u}) \cdot \mathbf{c}$$
$$\text{s.t.} \begin{cases} \mathbf{u}(\mathbf{u}-1) = 0, (\mathbf{u}-1/2)(\mathbf{A}\mathbf{x}+\mathbf{b}) \geq 0 \\ \mathbf{t}^2 \leq 1, (\mathbf{x}-\bar{\mathbf{x}})^2 \leq \varepsilon^2 \end{cases}$$

- Dense constraints: $(\mathbf{u} - 1/2)(\mathbf{A}\mathbf{x} + \mathbf{b}) \geq 0$.

Approach 1: standard Lasserre's relaxation

- POP:

$$\begin{aligned} \max \quad & \mathbf{t}^T \cdot \mathbf{A}^T \cdot \mathrm{diag}(\mathbf{u}) \cdot \mathbf{c} \\ \text{s.t.} \quad & \begin{cases} \mathbf{u}(\mathbf{u}-1) = 0, (\mathbf{u}-1/2)(\mathbf{A}\mathbf{x}+\mathbf{b}) \geq 0 \\ \mathbf{t}^2 \leq 1, (\mathbf{x}-\bar{\mathbf{x}})^2 \leq \varepsilon^2 \end{cases} \end{aligned}$$

- Cliques:

$$I = \{x_1, \ldots, x_p, u_1, \ldots, u_p\}, J_i = \{u_1, \ldots, u_p, t_i\}, i = 1, \ldots, p$$

- ρ_1 := 1st-order sparse Lasserre's relaxation, ρ_2 := 2nd-order sparse Lasserre's relaxation.

Approach 1: standard Lasserre's relaxation

- 2nd-order sparse Lasserre's relaxation:

$$\rho_2 = \max \quad L_{\mathbf{y}}(\mathbf{t}^T \cdot \mathbf{A}^T \cdot \mathrm{diag}(\mathbf{u}) \cdot \mathbf{c})$$
$$\text{s.t.} \begin{cases} \mathbf{M}_2(\mathbf{y}, I) \succeq 0, \mathbf{M}_2(\mathbf{y}, J_i) \succeq 0, L_{\mathbf{y}}(1) = 1; \\ \mathbf{M}_1(u_i(u_i - 1)\mathbf{y}, J_i) = 0, \\ \mathbf{M}_1((u_i - 1/2)(\mathbf{A}_i\mathbf{x} + b_i)\mathbf{y}, I) \succeq 0; \\ \mathbf{M}_1((1 - t_i^2)\mathbf{y}, J_i) \succeq 0; \\ \mathbf{M}_1((\varepsilon^2 - (x_i - \bar{x}_i)^2)\mathbf{y}, I) \succeq 0. \end{cases}$$

- $|I| = 2p$, $\mathbf{M}_2(\mathbf{y}, I)$ of size $\binom{2p+2}{2} = (p+1)(2p+1) = O(p^2)$.
- $|J_i| = p+1$, $\mathbf{M}_2(\mathbf{y}, J_i)$ of size $\binom{p+3}{2} = (p+3)(p+2)/2 = O(p^2)$.

Approach 2: heuristic relaxation

- Trick 1: reduce the size of the cliques:

$$I = \{x_1, \dots, x_p, u_1, \dots, u_p\} \longrightarrow \{x_i\}$$
$$J_i = \{u_1, \dots, u_p, t_i\} \longrightarrow \{u_i, t_i\}$$

  Note: these cliques **no longer** satisfies the RIP condition.

- Trick 2: reduce the order of localizing matrices w.r.t. dense constraints:

$$\mathbf{M}_1((u_i - 1/2)(\mathbf{A}_i\mathbf{x} + b_i)\mathbf{y}, I)$$
$$\longrightarrow \mathbf{M}_0((u_i - 1/2)(\mathbf{A}_i\mathbf{x} + b_i)\mathbf{y}, I) = L_{\mathbf{y}}((u_i - 1/2)(\mathbf{A}_i\mathbf{x} + b_i))$$

- Trick 3: Add a full 1st-order moment matrix $\mathbf{M}_1(\mathbf{y})$ to make the problem feasible.

Approach 2: heuristic relaxation

- 2nd-order heuristic relaxation:

$$h_2 = \max \quad L_{\mathbf{y}}(\mathbf{t}^T \cdot \mathbf{A}^T \cdot \mathrm{diag}(\mathbf{u}) \cdot \mathbf{c})$$

$$\text{s.t.} \begin{cases} \boxed{\mathbf{M}_1(\mathbf{y}) \succeq 0}, \mathbf{M}_2(\mathbf{y}, \boxed{\{x_i\}}) \succeq 0, \mathbf{M}_2(\mathbf{y}, \boxed{\{u_i, t_i\}}) \succeq 0, L_{\mathbf{y}}(1) = 1; \\ \mathbf{M}_1(u_i(u_i - 1)\mathbf{y}, \boxed{\{u_i, t_i\}}) = 0, \\ \boxed{L_{\mathbf{y}}((u_i - 1/2)(\mathbf{A}_i\mathbf{x} + b_i)) \succeq 0}; \\ \mathbf{M}_1((1 - t_i^2)\mathbf{y}, \boxed{\{u_i, t_i\}}) \succeq 0; \\ \mathbf{M}_1((\varepsilon^2 - (x_i - \bar{x}_i)^2)\mathbf{y}, \boxed{\{x_i\}}) \succeq 0. \end{cases}$$

- $\rho_1 \leq h_2 \leq \rho_2$.

Outline

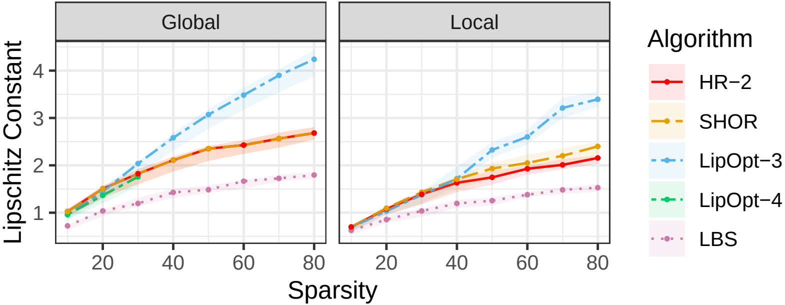
Several POP-based approaches

- Solving POPs reduces to find efficient positivity certificates:

Certificates	Types	Algorithms	Applications
Krivine-Stengle	LP	**LipOpt-3/4**	Lipschitz constant (Latorre et al., 2020)
Shor	SDP	**SDP-cert**	Certification (Raghunathan et al., 2018)
Putinar	SDP	**HR-2** (ours)	Lipschitz constant (Chen et al., 2020)

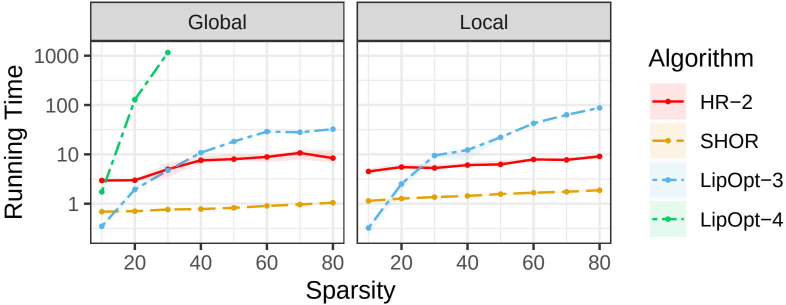
- Our contribution: **HR-2**, a *sparse* version of degree-4 Lasserre's relaxation adapted to deep learning applications, provides significant better results than **LipOpt-3/4**.

Lipschitz Constant of Neural Networks



Upper bounds of Lipschitz constants of random $(80, 80)$ networks

Lipschitz Constant of Neural Networks



Running time of each algorithm of random $(80, 80)$ networks

Robustness Certification

- 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
HR-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 SDP-NN (784, 500) network by **HR-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%

Conclusion

- **HR-2** is an intermediate relaxation between the 1st and 2nd Lasserre's relaxation.
- **HR-2** provides valid upper bounds of Lipschitz constant of neural network.
- **HR-2** is based on SDP, hence relies on SDP solver. This is the main reason that the heuristic approach does **NOT** scale.