Training Provably Robust Models by Polyhedral Envelope Regularization

Chen Liu, Mathieu Salzmann, Sabine Süsstrunk

École Polytechnique Fédérale de Lausanne

November 23, 2020

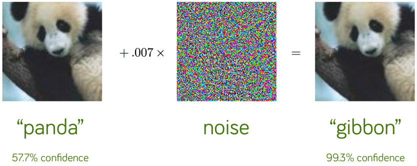
Overview

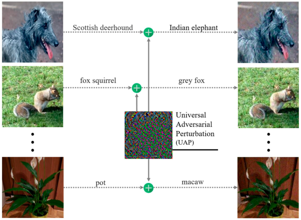
Content

Introduction

Existence of Adversarial Example

- State-of-the-art deep learning models are vulnerable to adversarial attacks.
- Imperceptible attack. [1] Sparse attack. [2] Universal attack [3].





[1] "Explaining and harnessing adversarial examples." ICLR 2014.

[2] "One pixel attack for fooling deep neural networks." IEEE Transactions on Evolutionary Computation (2019).

[3] "Universal adversarial perturbations." CVPR 2017.

Definition (Robustness Problem)

Given a classification model $f(\theta, \mathbf{x}) : \Theta \times \mathbb{R}^H \rightarrow \mathbb{R}^K$ parameterized by θ, data points drawn from the distribution $(\mathbf{x}, y) \sim \mathcal{D}$ and loss function $\mathcal{L}$, robustness problem is formulation as follows:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \max_{\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x})} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{1}$$

where $\mathcal{S}_\epsilon(\mathbf{x})$ is called the adversarial budget.

Definition (Robustness Problem)

Given a classification model $f(\theta, \mathbf{x}) : \Theta \times \mathbb{R}^H \rightarrow \mathbb{R}^K$ parameterized by θ, data points drawn from the distribution $(\mathbf{x}, y) \sim \mathcal{D}$ and loss function $\mathcal{L}$, robustness problem is formulation as follows:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \max_{\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x})} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{1}$$

where $\mathcal{S}_\epsilon(\mathbf{x})$ is called the adversarial budget.

- $\mathcal{S}_\epsilon(\mathbf{x})$ is often defined by $\{\mathbf{x}' | \|\mathbf{x}' - \mathbf{x}\| \leq \epsilon\}$

Definition (Robustness Problem)

Given a classification model $f(\theta, \mathbf{x}) : \Theta \times \mathbb{R}^H \to \mathbb{R}^K$ parameterized by θ, data points drawn from the distribution $(\mathbf{x}, y) \sim \mathcal{D}$ and loss function $\mathcal{L}$, robustness problem is formulation as follows:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \max_{\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x})} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{1}$$

where $\mathcal{S}_\epsilon(\mathbf{x})$ is called the adversarial budget.

- $\mathcal{S}_\epsilon(\mathbf{x})$ is often defined by $\{\mathbf{x}' | \|\mathbf{x}' - \mathbf{x}\| \leq \epsilon\}$
- Attack algorithm solves the inner maximization problem.
- Defense algorithm solves the outer minimization problem.

$$\max_{\|\mathbf{x}-\mathbf{x}'\|_\infty<\epsilon} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{2}$$

[4] "Explaining and harnessing adversarial examples." ICLR 2014.

[5] "Towards deep learning models resistant to adversarial attacks." ICLR 2018.

$$\max_{\|\mathbf{x}-\mathbf{x}'\|_\infty<\epsilon} \mathcal{L}(f(\theta,\mathbf{x}'),y) \tag{2}$$

- Fast Gradient Sign Method (FGSM) [4].

$$\mathbf{x}' \leftarrow \mathbf{x} + \epsilon \mathrm{sign}(\nabla_{\mathbf{x}}\mathcal{L}(f(\theta,\mathbf{x}'),y)) \tag{3}$$

[4] "Explaining and harnessing adversarial examples." ICLR 2014.

[5] "Towards deep learning models resistant to adversarial attacks." ICLR 2018.

$$\max_{\|\mathbf{x}-\mathbf{x}'\|_\infty<\epsilon} \mathcal{L}(f(\theta,\mathbf{x}'),y) \tag{2}$$

- Fast Gradient Sign Method (FGSM) [4].

$$\mathbf{x}' \leftarrow \mathbf{x} + \epsilon\mathrm{sign}(\nabla_{\mathbf{x}}\mathcal{L}(f(\theta,\mathbf{x}'),y)) \tag{3}$$

- Projected Gradient Descent (PGD) [5] $\sim$ iterative fast gradient sign method.

$$\mathbf{x}^{(t+1)} \leftarrow \Pi_{\{\mathbf{x}' | \|\mathbf{x}'-\mathbf{x}\| \leq \epsilon\}} \left[\mathbf{x}^{(t)} + \alpha\mathrm{sign}(\nabla_{\mathbf{x}}\mathcal{L}(f(\theta,\mathbf{x}^{(t)}),y))\right] \tag{4}$$

[4] "Explaining and harnessing adversarial examples." ICLR 2014.

[5] "Towards deep learning models resistant to adversarial attacks." ICLR 2018.

$$\min_{\theta} \max_{\|\mathbf{x}-\mathbf{x}'\|_\infty<\epsilon} \mathcal{L}(f(\theta,\mathbf{x}'),y) \tag{5}$$

[6] "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples." ICML 2018.

$$\min_{\theta} \max_{\|\mathbf{x}-\mathbf{x}'\|_\infty<\epsilon} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{5}$$

- Empirical defense algorithm: estimate the inner maximization problem by its lower bound.
  - Most effective method: PGD adversarial training. [6]

[6] "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples." ICML 2018.

$$\min_{\theta} \max_{\|\mathbf{x}-\mathbf{x}'\|_\infty < \epsilon} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{5}$$

- Empirical defense algorithm: estimate the inner maximization problem by its lower bound.
  - Most effective method: PGD adversarial training. [6]
- Provably defense algorithm: solve the inner maximization problem exactly or estimate by its upper bound.
  - Convexize loss function. Linear approximation. Mixed integer programming. Random input smoothing e.t.c.

[6] "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples." ICML 2018.

$$\min_{\theta} \max_{\|\mathbf{x}-\mathbf{x}'\|_\infty<\epsilon} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{5}$$

- Empirical defense algorithm: estimate the inner maximization problem by its lower bound.
  - Most effective method: PGD adversarial training. [6]
- Provably defense algorithm: solve the inner maximization problem exactly or estimate by its upper bound.
  - Convexize loss function. Linear approximation. Mixed integer programming. Random input smoothing e.t.c.
- Robust certification: the input neighbor region guaranteed to be adversary-free.
- Evaluation metrics:

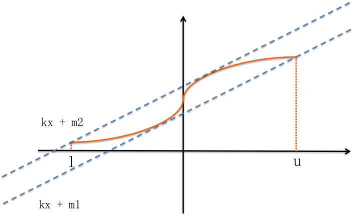
  Clean Accuracy $\geq$ Empirical Robust Accuracy $\geq$ Robust Accuracy $\geq$ Certified Robust Accuracy

[6] "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples." ICML 2018.

Content

Regularization based on Geometric Envelope

Linear Approximation



- Given any nonlinear function $\sigma(\mathbf{x})$ with bounded input $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$, we can introduce one diagonal matrix $\mathbf{D}$ and two vectors $\mathbf{m}_1$, $\mathbf{m}_2$:

$$\mathbf{D}\mathbf{x} + \mathbf{m}_1 \leq \sigma(\mathbf{x}) \leq \mathbf{D}\mathbf{x} + \mathbf{m}_2$$

- Equivalently, $\forall \mathbf{x} : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$, we have $\mathbf{D}, \mathbf{m}_1, \mathbf{m}_2$ and $\exists \mathbf{m} : \mathbf{m}_1 \leq \mathbf{m} \leq \mathbf{m}_2$, such that

$$\sigma(\mathbf{x}) = \mathbf{D}\mathbf{x} + \mathbf{m}$$

- Recall the N-layer neural network.

$$
\begin{aligned}
\mathbf{z}^{(i+1)} &= \mathbf{W}^{(i)}\hat{\mathbf{z}}^{(i)} + \mathbf{b}^{(i)} \quad & i = 1, 2, ..., N-1 \\
\hat{\mathbf{z}}^{(i)} &= \sigma(\mathbf{z}^{(i)}) \quad & i = 2, 3, ..., N-1
\end{aligned}
\tag{6}
$$

[7] "Towards fast computation of certified robustness for relu networks." ICML 2018

[8] "Efficient neural network robustness certification with general activation functions." NeurIPS 2018

Regularization based on Geometric Envelope

Model Linearization

- Recall the N-layer neural network.

$$\begin{aligned} \mathbf{z}^{(i+1)} &= \mathbf{W}^{(i)}\hat{\mathbf{z}}^{(i)} + \mathbf{b}^{(i)} \qquad i = 1, 2, ..., N-1 \\ \hat{\mathbf{z}}^{(i)} &= \sigma(\mathbf{z}^{(i)}) \qquad\qquad\quad i = 2, 3, ..., N-1 \end{aligned} \tag{6}$$

- We can linearize the output of each layer.

$$\begin{aligned} \mathbf{z}^{(i)} &= \mathbf{W}^{(i-1)}(\sigma(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x}+\mathbf{m}^{(1)})+\mathbf{b}^{(1)})...)+\mathbf{b}^{i-2}))+\mathbf{b}^{(i-1)} \\ &= \mathbf{W}^{(i-1)}(\mathbf{D}^{(i-1)}(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x}+\mathbf{m}^{(1)})+\mathbf{b}^{(1)})...)+\mathbf{b}^{(i-2)})+\mathbf{m}^{(i-1)})+\mathbf{b}^{(i-1)} \\ &= \left(\Pi_{j=2}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(1)}\mathbf{x} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{b}^{(h)} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(h)}\mathbf{m}^{(h)} \end{aligned} \tag{7}$$

[7] "Towards fast computation of certified robustness for relu networks." ICML 2018

[8] "Efficient neural network robustness certification with general activation functions." NeurIPS 2018

Regularization based on Geometric Envelope

Model Linearization

- Recall the N-layer neural network.

$$
\begin{aligned}
\mathbf{z}^{(i+1)} &= \mathbf{W}^{(i)}\hat{\mathbf{z}}^{(i)} + \mathbf{b}^{(i)} \qquad i = 1, 2, ..., N-1 \\
\hat{\mathbf{z}}^{(i)} &= \sigma(\mathbf{z}^{(i)}) \qquad\qquad\quad i = 2, 3, ..., N-1
\end{aligned}
\tag{6}
$$

- We can linearize the output of each layer.

$$
\begin{aligned}
\mathbf{z}^{(i)} &= \mathbf{W}^{(i-1)}(\sigma(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x}+\mathbf{m}^{(1)})+\mathbf{b}^{(1)})...)+\mathbf{b}^{i-2}))+\mathbf{b}^{(i-1)} \\
&= \mathbf{W}^{(i-1)}(\mathbf{D}^{(i-1)}(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x}+\mathbf{m}^{(1)})+\mathbf{b}^{(1)})...)+\mathbf{b}^{(i-2)})+\mathbf{m}^{(i-1)})+\mathbf{b}^{(i-1)} \\
&= \left(\Pi_{j=2}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(1)}\mathbf{x} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{b}^{(h)} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(h)}\mathbf{m}^{(h)}
\end{aligned}
\tag{7}
$$

- Bound for $\{\mathbf{m}^{(h)}\}_{h=1}^{i-1}$ $\rightarrow$ bounds for $\mathbf{z}^{(i)}$ $\rightarrow$ bound for $\mathbf{m}^{(i)}$
- Iteratively estimate the bounds for $\{\mathbf{z}^{(i)}\}_{i=2}^{N}$ [7] [8]

[7] "Towards fast computation of certified robustness for relu networks." ICML 2018

[8] "Efficient neural network robustness certification with general activation functions." NeurIPS 2018

Proposition (Model Linearization)

Given a classification model $f(\theta, \mathbf{x}) : \Theta \times \mathbb{R}^H \rightarrow \mathbb{R}^K$ *parameterized by* θ*, a data point* $(\mathbf{x}, y)$ *and a pre-defined adversarial budget* $\mathcal{S}_\epsilon(\mathbf{x})$*,* $\exists \mathbf{W} \in \mathbb{R}^{H \times K}, \mathbf{b} \in \mathbb{R}^K$ *such that*

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), f(\theta, \mathbf{x}') - f(\theta, \mathbf{x}')_y \leq \mathbf{W}\mathbf{x}' + \mathbf{b} \tag{8}$$

[9] "Towards fast computation of certified robustness for relu networks." ICML 2018

[10] "Efficient neural network robustness certification with general activation functions." NeurIPS 2018

[11] "On the effectiveness of interval bound propagation for training verifiably robust models." 2018

[12] "Provable defenses against adversarial examples via the convex outer adversarial polytope" ICML 2018.

Proposition (Model Linearization)

Given a classification model $f(\theta, \mathbf{x}) : \Theta \times \mathbb{R}^H \to \mathbb{R}^K$ *parameterized by* θ*, a data point* $(\mathbf{x}, y)$ *and a pre-defined adversarial budget* $\mathcal{S}_\epsilon(\mathbf{x})$*,* $\exists \mathbf{W} \in \mathbb{R}^{H\times K}, \mathbf{b} \in \mathbb{R}^K$ *such that*

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), f(\theta, \mathbf{x}') - f(\theta, \mathbf{x}')_y \leq \mathbf{W}\mathbf{x}' + \mathbf{b} \tag{8}$$

- Method to calculate $\mathbf{W}, \mathbf{b}$: Fast-Lin [9], CROWN [10], IBP-inspired [11].
- We can further bound $\mathbf{W}\mathbf{x}' + \mathbf{b} \leq \mathbf{v}, \forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x})$. If $\mathcal{L}$ is softmax cross entropy loss, then we have $\max_{\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x})} \mathcal{L}(f(\theta, \mathbf{x}'), y) \leq \mathcal{L}(\mathbf{v}, y)$. Minimizing the RHS allows us to train provable models (KW [12]).

[9] "Towards fast computation of certified robustness for relu networks." ICML 2018

[10] "Efficient neural network robustness certification with general activation functions." NeurIPS 2018

[11] "On the effectiveness of interval bound propagation for training verifiably robust models." 2018

[12] "Provable defenses against adversarial examples via the convex outer adversarial polytope" ICML 2018.

Methodology

Geometric Interpretation

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), f(\theta, \mathbf{x}') - f(\theta, \mathbf{x}')_y \leq \mathbf{W}\mathbf{x}' + \mathbf{b} \tag{9}$$

Methodology

Geometric Interpretation

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), \forall i \in [K], f(\theta, \mathbf{x}')_i - f(\theta, \mathbf{x}')_y \leq \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \qquad (9)$$

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), \forall i \in [K], f(\theta, \mathbf{x}')_i - f(\theta, \mathbf{x}')_y \leq \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0 \qquad (9)$$

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), \forall i \in [K], f(\theta, \mathbf{x}')_i - f(\theta, \mathbf{x}')_y \leq \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0 \tag{9}$$

- If $\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}) \cap \{\mathbf{x}' | \forall i, \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0\}$, then $\mathbf{x}'$ is guaranteed to have the same prediction as $\mathbf{x}$.

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), \forall i \in [K], f(\theta, \mathbf{x}')_i - f(\theta, \mathbf{x}')_y \leq \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0 \tag{9}$$

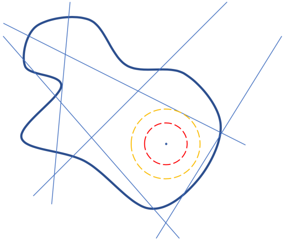
- If $\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}) \cap \{\mathbf{x}' | \forall i, \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0\}$, then $\mathbf{x}'$ is guaranteed to have the same prediction as $\mathbf{x}$.
- $\{\mathbf{x}' | \forall i, \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0\}$ forms a polyhedron in $\mathbb{R}^H$ space and is an envelope of the model's decision boundary.

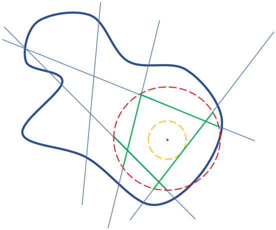
Methodology

Geometric Interpretation

$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), \forall i \in [K], f(\theta, \mathbf{x}')_i - f(\theta, \mathbf{x}')_y \leq \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0 \tag{9}$$

- If $\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}) \cap \{\mathbf{x}' | \forall i, \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0\}$, then $\mathbf{x}'$ is guaranteed to have the same prediction as $\mathbf{x}$.
- $\{\mathbf{x}' | \forall i, \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0\}$ forms a polyhedron in $\mathbb{R}^H$ space and is an envelope of the model's decision boundary.
- Geometric interpretation: when ϵ is too big or too small.





$$\forall \mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x}), \forall i \in [K], f(\theta, \mathbf{x}')_i - f(\theta, \mathbf{x}')_y \leq \mathbf{W}_i \mathbf{x}' + \mathbf{b}_i \leq 0 \qquad (10)$$

Theorem (Theoretical Robustness Guarantee)

Given a classification model $f(\theta, \mathbf{x}) : \Theta \times \mathbb{R}^H \rightarrow \mathbb{R}^K$ *parameterized by* θ *and linear bounds in Equation 10, we assume adversarial budget is defined based on* l_p *norm:* $\mathcal{S}_\epsilon(\mathbf{x}) = \{\mathbf{x}' | \|\mathbf{x}' - \mathbf{x}\|_p \leq \epsilon\}$, *then there is no adversarial example inside an* l_p *norm ball of radius* d *centered around* $\mathbf{x}$, *with* $d = \min_{i \in [K]} \{\epsilon, d_i\}$, $d_i = \max \left\{0, -\frac{\mathbf{W}_i \mathbf{x} + \mathbf{b}_i}{\|\mathbf{W}_i\|_q}\right\}$, *where* l_q *is the dual norm of* l_p *,i.e.* $\frac{1}{p} + \frac{1}{q} = 1$.

- This theorem is too pessimistic, as the attack can not perturb the image out of domain $[0, 1]^H$.
- If we constrain the perturbed images inside $[0, 1]^H$, the certified bound should be larger.

- This theorem is too pessimistic, as the attack can not perturb the image out of domain $[0,1]^H$.
- If we constrain the perturbed images inside $[0,1]^H$, the certified bound should be larger.
- We need to calculate the distance between the clean input $\mathbf{x}$ and set $\mathcal{S} = \cup_{i\in[K]}\{\mathbf{x}'|\mathbf{W}_i\mathbf{x}' + \mathbf{b}_i > 0\} \cap [0,1]^H$, which is the minimum distance to $\mathcal{S}_i = \{\mathbf{x}'|\mathbf{W}_i\mathbf{x}' + \mathbf{b}_i > 0\} \cap [0,1]^H$ over $i \in [K]$.

$$
\begin{aligned}
&\min_{\mathbf{x}'} \|\mathbf{x} - \mathbf{x}'\|_p \\
&s.t. 0 \leq \mathbf{x}' \leq 1 \\
&\mathbf{W}_i\mathbf{x}' + \mathbf{b}_i > 0
\end{aligned}
\tag{11}
$$

$$\min_{\mathbf{x}'} \|\mathbf{x} - \mathbf{x}'\|_p$$
$$s.t. 0 \leq \mathbf{x}' \leq 1 \tag{12}$$
$$\mathbf{W}_i \mathbf{x}' + \mathbf{b}_i > 0$$

- Convex objective with linear constraints.
- To satisfy $\mathbf{W}_i \mathbf{x}' + \mathbf{b}_i > 0$, the solution to minimize $\|\mathbf{x} - \mathbf{x}'\|_p$ is:

$$\tilde{\mathbf{x}}' = \mathbf{x} - \frac{\mathbf{W}_i \mathbf{x} + \mathbf{b}_i}{\|\mathbf{W}_i\|_q^q} \mathbf{W}_{\mathrm{i}} |\mathbf{W}_i|^{\frac{q}{p}} \tag{13}$$

- Greedy algorithm to find points satisfying $0 \leq \mathbf{x}' \leq 1$: check if elements of $\tilde{\mathbf{x}}'$ satisfying the constraint, for those that don't, clip them to 0 or 1 and keep them fixed in the next iteration.

$$\min_{\mathbf{x}'} \|\mathbf{x} - \mathbf{x}'\|_p \quad s.t. 0 \leq \mathbf{x}' \leq 1 \quad \mathbf{W}_i\mathbf{x}' + \mathbf{b}_i > 0 \tag{14}$$

$$\tilde{\mathbf{x}}' = \mathbf{x} - \frac{\mathbf{W}_i\mathbf{x} + \mathbf{b}_i}{\|\mathbf{W}_i\|_q^q}\mathbf{W}_\mathrm{i}|\mathbf{W}_i|^{\frac{q}{p}} \tag{15}$$

- Given $\mathbf{W}_i$, $\mathbf{b}_i$, $\mathbf{x}$
- Frozen dimension $\mathcal{S}^{(f)} = \emptyset$
- Calculate $\tilde{\mathbf{x}}'$ based on 15
- While $0 \leq \tilde{\mathbf{x}}' \leq 1$ not satisfied:
  - Update $\mathcal{S}^{(f)} = \mathcal{S}^{(f)} \cup \{j|\tilde{\mathbf{x}}'_j < 0\} \cup \{j|\tilde{\mathbf{x}}'_j > 1\}$
  - Clip $\tilde{\mathbf{x}}' = \mathrm{clip}(\tilde{\mathbf{x}}', \min = 0, \max = 1)$
  - Update $\tilde{\mathbf{x}}'$ based on 15 with $\tilde{\mathbf{x}}_j, j \in \mathcal{S}^{(f)}$ fixed
- Return solution $\|\tilde{\mathbf{x}}' - \mathbf{x}\|_p$

Corollary (Optimality Guarantee)

The greedy algorithm is guaranteed to find the optimum of problem 14.

- We call this method Polyhedral Envelope Certification (PEC).
- Advantages:
  - Almost no overhead.
  - Finer-grained certified bounds.
  - Fast convergence when searching for optimal bounds by binary search.

Definition

Given the certified bound $\tilde{d}$ by PEC, we define the Polyhedral Envelope Regularization (PER) based on hinge-loss.

$$PER(\mathbf{x}, y, \theta) = \max\left\{0, 1 - \frac{\tilde{d}}{\epsilon}\right\} \tag{16}$$

Definition

Given the certified bound $\tilde{d}$ by PEC, we define the Polyhedral Envelope Regularization (PER) based on hinge-loss.

$$PER(\mathbf{x}, y, \theta) = \max\left\{0, 1 - \frac{\tilde{d}}{\epsilon}\right\} \tag{16}$$

- Training objective of PER: $\mathcal{L}(f(\mathbf{x}, \theta), y) + \gamma PER(\mathbf{x}, y, \theta)$
- We can combine PER with adversarial training: $\mathcal{L}(f(\mathbf{x}', \theta), y) + \gamma PER(\mathbf{x}', y, \theta)$, where $\mathbf{x}'$ is found by PGD.
- We can use sub-sampling to decrease the complexity of PER: $\mathcal{L}(f(\mathbf{x}, \theta), y) + \gamma PER(\bar{\mathbf{x}}, \bar{y}, \theta)$, where $(\bar{\mathbf{x}}, \bar{y})$ is sub-sampled from a mini-batch $(\mathbf{x}, y)$.

Content

Experiment
Settings

- Models: FC1 for MNIST, LeNet for MNIST and CIFAR10.
- 7 baselines: normal training (plain), PGD adversarial training (at), KW [13], MMR, MMR+at [14], IBP [15], C-IBP [16].
- 8 evaluation metric: clean test accuracy (CTE), PGD robust accuracy (PGD), incomplete certified robust accuracy by Fast-Lin / CROWN (CRE Lin / CRE CROWN) and by IBP (CRE IBP), complete certified robust accuracy (CRW MIP) [17], average certified bounds by Fast-Lin [18]/ CROWN [19] (ACB KW / ACB CRO) and IBP (ACB IBP), average certified bounds by PEC (ACB PEC).

[13] "Provable defenses against adversarial examples via the convex outer adversarial polytope." ICML 2018.

[14] "Provable robustness of relu networks via maximization of linear regions." AISTATS 2019.

[15] "On the effectiveness of interval bound propagation for training verifiably robust models." ICCV 2019.

[16] "Towards stable and efficient training of verifiably robust neural networks." ICLR 2020.

[17] "Training for faster adversarial robustness verification via inducing reLU stability." ICLR 2019.

[18] "Towards fast computation of certified robustness for relu networks." ICML 2018

[19] "Efficient neural network robustness certification with general activation functions." NeurIPS 2018.

Experiment

Results for ReLU Network

Methods	CTE (%)	PGD (%)	CRE Lin (%)	CRE IBP (%)	CRE MIP (%)	ACB Lin	ACB IBP	ACB PEC
				MNIST - FC1, ReLU, l_∞, $\epsilon = 0.1$				
plain	1.99	98.37	100.00	100.00	100.00	0.0000	0.0000	0.0000
at	1.42	9.00	97.94	100.00	100.00	0.0021	0.0000	0.0099
KW	2.26	8.59	12.91	69.20	10.90	0.0871	0.0308	0.0928
IBP	1.65	9.67	87.27	**15.20**	12.36	0.0127	**0.0848**	0.0705
C-IBP	1.98	9.50	67.39	**14.45**	11.39	0.0326	**0.0855**	0.0800
MMR	2.11	17.82	33.75	99.88	24.90	0.0663	0.0001	0.0832
MMR+at	2.04	10.39	17.64	95.09	14.10	0.0824	0.0049	0.0905
C-PER	**1.60**	**7.45**	**11.71**	92.89	**7.69**	**0.0883**	0.0071	**0.0935**
C-PER+at	1.81	7.73	12.90	99.90	8.22	0.0871	0.0001	0.0925
I-PER	**1.60**	**6.28**	**11.96**	93.33	**8.10**	**0.0880**	0.0067	**0.0934**
I-PER+at	**1.54**	7.15	13.96	98.55	8.48	0.0868	0.0014	0.0927
				MNIST - CNN, ReLU, l_∞, $\epsilon = 0.1$				
plain	1.28	85.75	100.00	100.00	100.00	0.0000	0.0000	0.0000
at	1.02	4.75	91.91	100.00	100.00	0.0081	0.0000	0.0189
KW	1.21	3.03	**4.44**	100.00	4.40	**0.0956**	0.0000	**0.0971**
IBP	1.51	4.43	23.89	**8.13**	5.23	0.0761	**0.0919**	0.0872
C-IBP	1.85	4.28	10.72	**6.91**	4.83	0.0893	**0.0931**	0.0928
MMR	1.65	6.07	11.56	100.00	6.10	0.0884	0.0000	0.0928
MMR+at	1.19	3.35	9.49	100.00	3.60	0.0905	0.0000	0.0939
C-PER	1.44	3.44	5.13	100.00	3.62	0.0949	0.0000	0.0965
C-PER+at	**0.50**	**2.02**	4.85	100.00	**2.21**	0.0952	0.0000	0.0969
I-PER	1.03	2.40	4.64	99.55	2.52	**0.0954**	0.0004	0.0967
I-PER+at	**0.48**	**1.29**	**4.61**	99.94	**1.47**	**0.0954**	0.0001	**0.0971**
				CIFAR10 - CNN, ReLU, l_∞, $\epsilon = 2/255$				
plain	24.62	86.29	100.00	100.00	100.00	0.0000	0.0000	0.0000
at	27.04	48.53	85.36	100.00	88.50	0.0011	0.0000	0.0015
KW	39.27	46.60	**53.81**	99.98	**48.00**	**0.0036**	0.0000	**0.0040**
IBP	46.74	56.38	61.81	**67.58**	58.80	0.0030	**0.0025**	0.0034
C-IBP	58.32	63.56	66.28	**69.10**	65.44	0.0026	**0.0024**	0.0029
MMR	34.59	57.17	69.28	100.00	61.00	0.0024	0.0000	0.0032
MMR+at	35.36	49.27	59.91	100.00	54.20	0.0031	0.0000	0.0037
C-PER	39.21	50.98	57.45	99.98	52.70	0.0033	0.0000	0.0038
C-PER+at	**28.87**	**43.55**	**56.59**	100.00	48.43	**0.0034**	0.0000	**0.0040**
I-PER	29.34	51.54	64.34	99.98	54.87	0.0028	0.0000	0.0036
I-PER+at	**26.66**	**43.35**	57.72	100.00	**47.87**	0.0033	0.0000	**0.0040**

TABLE I: Full results of 11 training schemes and 8 evaluation schemes for ReLU networks under l_∞ attacks. The best and the second best results among provably robust training methods (plain and at excluded) are bold. In addition, the best results are underlined.

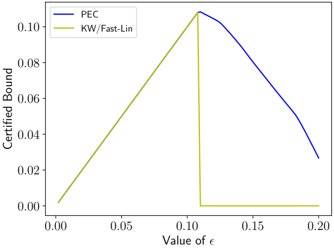
Experiment

Results for Non-ReLU Network

Methods	CTE (%)	PGD (%)	CRE CRO (%)	CRE IBP (%)	ACB CRO	ACB IBP	ACB PEC
			MNIST - FC1, Sigmoid, l_∞, $\epsilon = 0.1$				
plain	2.04	97.80	100.00	100.00	0.0000	0.0000	0.0000
at	1.78	10.05	98.52	100.00	0.0015	0.0000	0.0055
IBP	2.06	10.58	44.14	13.65	0.0559	0.0863	0.0846
C-IBP	2.88	9.83	26.04	**12.51**	0.0740	**0.0875**	0.0886
C-PER	**1.97**	7.55	12.15	84.76	0.0879	0.0152	**0.0930**
C-PER+at	2.16	**7.12**	**11.87**	88.06	**0.0881**	0.0119	0.0927
I-PER	2.15	8.35	12.79	86.99	0.0872	0.0130	0.0926
I-PER+at	2.45	8.05	12.36	88.94	0.0876	0.0111	0.0923
			MNIST - FC1, Tanh, l_∞, $\epsilon = 0.1$				
plain	2.00	97.80	100.00	100.00	0.0000	0.0000	0.0000
at	1.28	8.89	99.98	100.00	0.0000	0.0000	0.0001
IBP	**2.04**	9.84	31.81	13.02	0.0682	0.0870	0.0864
C-IBP	2.75	9.57	20.10	**11.80**	0.0799	**0.0882**	0.0894
C-PER	2.19	7.71	11.55	57.81	0.0885	0.0422	**0.0934**
C-PER+at	2.30	**7.45**	**11.39**	56.74	**0.0886**	0.0433	0.0930
I-PER	2.21	8.51	12.23	55.53	0.0878	0.0445	0.0929
I-PER+at	2.46	7.87	12.04	66.04	0.0880	0.0340	0.0929

TABLE II: Full results of 8 training schemes and 7 evaluation schemes for sigmoid and tanh networks under l_∞ attacks. The best results among provably robust training methods (plain and at excluded) are bold and underlined.

- To search for the optimal certified bound ϵ, Fast-Lin / CROWN adjust their target by binary search.
- PEC has a finer-grained certified bound, so needs fewer iterations than baselines.



Iterations used for Searching Optimal ϵ

Methods	MNIST-FC1, ReLU, l_∞			MNIST-CNN, ReLU, l_∞			CIFAR10-CNN, ReLU, l_∞		
	T_{Lin}	T_{PEC}	$\frac{T_{PEC}}{T_{Lin}}$	T_{Lin}	T_{PEC}	$\frac{T_{PEC}}{T_{Lin}}$	T_{Lin}	T_{PEC}	$\frac{T_{PEC}}{T_{Lin}}$
plain		9.85	0.8207		10.56	0.8804		9.33	0.9331
at		10.77	0.8972		11.39	0.9489		9.12	0.9128
KW		8.48	0.7066		11.61	0.9674		8.43	0.8432
MMR	12	8.04	0.6703	12	10.68	0.8897	10	8.05	0.8053
MMR+at		7.68	0.6402		11.22	0.9351		8.45	0.8450
C-PER		9.34	0.7780		11.17	0.9305		8.61	0.8606
C-PER+at		9.38	0.7816		11.74	0.9784		8.68	0.8681

TABLE IV: Number of steps of bound calculation for the optimal ϵ in Fast-Lin (T_{Lin}) and PEC (T_{PEC}) for ReLU networks under l_∞ attacks. Note that T_{Lin} is a constant for different models given the original interval $[\underline{\epsilon}, \bar{\epsilon}]$.

Content

- Consider a N-layer network with m, k, n as input, output and hidden dimensions. ($n >> k, m$)
- overhead of PEC / PER: $\mathcal{O}(km)$ (Negligible compared with model linearization)

Methods	Complexity
PGD	$\mathcal{O}(Nn^2)$
Fast-Lin / CROWN	$\mathcal{O}(N^2n^3)$
KW	$\mathcal{O}(N^2n^3)$
MMR / MMR+at	$\mathcal{O}(Nn^2m)$
IBP	$\mathcal{O}(Nn^2)$
C-IBP	$\mathcal{O}(Nn^3)$
I-PER / I-PER+at	$\mathcal{O}(Nn^2m)$
C-PER / C-PER+at	$\mathcal{O}(N^2n^3)$

TABLE V: Complexity of different methods on an N-layer neural network model with k-dimensional output and m-dimensional input. Each hidden layer has n neurons.

Analysis

Prevent Over-regularization

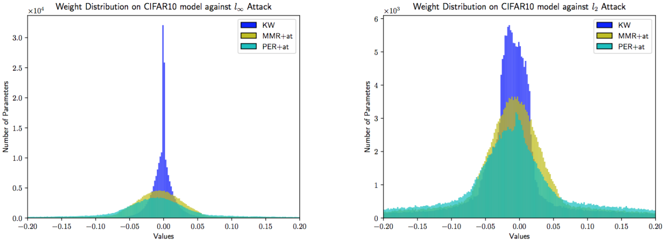


Figure 6: Parameter value distributions of CIFAR10 models trained against l_∞ and l_2 attacks. The Euclidean norm of parameter for KW, MMR+at, PER+at model against l_∞ attack is 18.08, 38.36 and 94.63 respectively. For models against l_2 attack, the corresponding Euclidean norm is 71.34, 62.97 and 141.77 respectively.

Analysis

Prevent Over-regularization

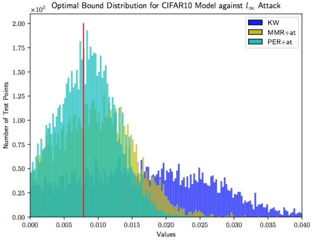


Figure 7: The distribution of optimal certified bounds of CIFAR10 models trained against l_∞ and l_2 attacks. The target bounds are marked as a red vertical line. ($2/255$ for l_∞ cases and 0.1 for l_2 cases.)

Content

Conclusion

- Contributions
  - Geometric interpretation of certified bounds.
  - Certification method with finer-grained certified bounds. (PEC)
  - Geometric inspired training method for provable robust model. (PER)
- Limitations
  - Scalability to bigger models.

Thank You!