On Certifying Non-uniform Bounds against Adversarial Attacks

Chen Liu[†], Ryota Tomioka[‡], Volkan Cevher[†]

†École Polytechnique Fédérale de Lausanne
‡Microsoft Research Cambridge

June 11th, 2019

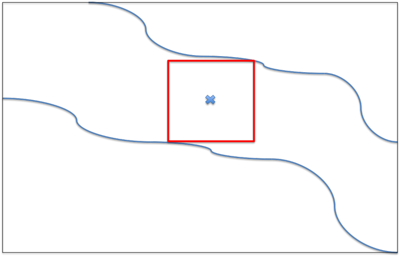
Background

Problem (Certification Problem)

Given the label set $\mathcal{C}$, a classification model $f : \mathbb{R}^n \to \mathcal{C}$ and an input data point $\mathbf{x} \in \mathbb{R}^n$, we would like to find the largest neighborhood $\mathcal{S}$ around $\mathbf{x}$ such that $f(\mathbf{x}) = f(\mathbf{x}') \ \forall \mathbf{x}' \in \mathcal{S}$.

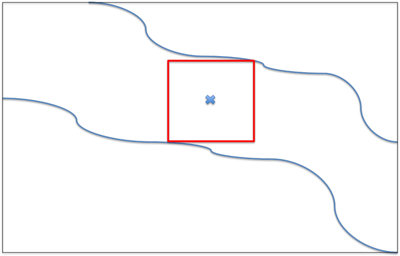
- Set $\mathcal{S}$ is called adversarial budget and $\mathbf{x} \in \mathcal{S}$.

Motivation

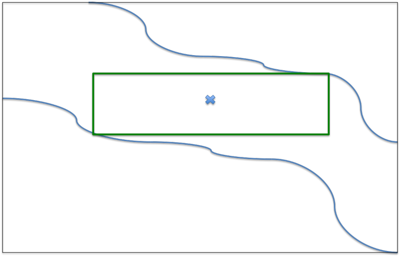


$$\mathcal{S}_\epsilon^{(p)}(\mathbf{x}) = \{\mathbf{x}' = \mathbf{x} + \epsilon\mathbf{v} | \|\mathbf{v}\|_p \leq 1\}$$
$$\epsilon \in \mathbb{R}$$

Motivation

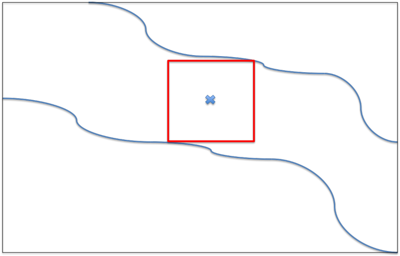


$$\mathcal{S}_{\epsilon}^{(p)}(\mathbf{x}) = \{\mathbf{x}' = \mathbf{x} + \epsilon \mathbf{v} | \|\mathbf{v}\|_p \leq 1\}$$
$$\epsilon \in \mathbb{R}$$

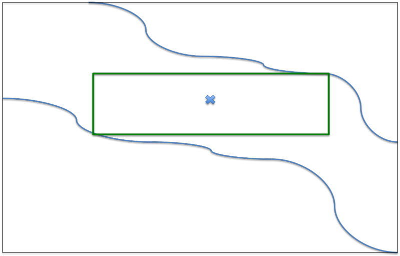


$$\mathcal{S}_{\boldsymbol{\epsilon}}^{(p)}(\mathbf{x}) = \{\mathbf{x}' = \mathbf{x} + \boldsymbol{\epsilon} \odot \mathbf{v} | \|\mathbf{v}\|_p \leq 1\}$$
$$\boldsymbol{\epsilon} \in \mathbb{R}^n$$

Motivation



$$\mathcal{S}_{\epsilon}^{(p)}(\mathbf{x}) = \{\mathbf{x}' = \mathbf{x} + \epsilon\mathbf{v} \,|\, \|\mathbf{v}\|_p \leq 1\}$$
$$\epsilon \in \mathbb{R}$$



$$\mathcal{S}_{\boldsymbol{\epsilon}}^{(p)}(\mathbf{x}) = \{\mathbf{x}' = \mathbf{x} + \boldsymbol{\epsilon} \odot \mathbf{v} \,|\, \|\mathbf{v}\|_p \leq 1\}$$
$$\boldsymbol{\epsilon} \in \mathbb{R}^n$$

Advantages of non-uniform bounds:

- Larger overall volumes.
- Quantitative metric of feature robustness.

Formulation

- A N-layer fully connected neural network, parameterized by $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=1}^{N-1}$

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

Formulation

- A N-layer fully connected neural network, parameterized by $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=1}^{N-1}$

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

- Given a model $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}$ and a data point $\mathbf{x}$ labeled as $c \in \mathcal{C}$, we want to

$$\begin{aligned} \min_{\boldsymbol{\epsilon}} \quad & \left\{ -\sum_{j=0}^{n_1-1} \log \epsilon_j \right\} \\ \hat{\mathbf{z}}^{(1)} &\in \mathcal{S}_{\boldsymbol{\epsilon}}(\mathbf{x}) \\ \mathbf{z}^{(i+1)} &= \mathbf{W}^{(i)}\hat{\mathbf{z}}^{(i)} + \mathbf{b}^{(i)} \quad & i = 1, 2, ..., N-1 \\ \hat{\mathbf{z}}^{(i)} &= \sigma(\mathbf{z}^{(i)}) \quad & i = 2, 3, ..., N-1 \\ z_c^{(N)} - z_j^{(N)} &\geq \delta \quad & j = 0, 1, ..., n_N - 1; j \neq c \end{aligned} \tag{2}$$

Formulation

- A N-layer fully connected neural network, parameterized by $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=1}^{N-1}$

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

- Given a model $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}$ and a data point $\mathbf{x}$ labeled as $c \in \mathcal{C}$, we want to

$$
\begin{aligned}
\min_{\boldsymbol{\epsilon}} \quad & \left\{ -\sum_{j=0}^{n_1-1} \log \epsilon_j \right\} \\
\hat{\mathbf{z}}^{(1)} &\in \mathcal{S}_{\boldsymbol{\epsilon}}(\mathbf{x}) \\
\mathbf{z}^{(i+1)} &= \mathbf{W}^{(i)}\hat{\mathbf{z}}^{(i)} + \mathbf{b}^{(i)} && i = 1, 2, ..., N-1 \\
\hat{\mathbf{z}}^{(i)} &= \sigma(\mathbf{z}^{(i)}) && i = 2, 3, ..., N-1 \\
z_c^{(N)} - z_j^{(N)} &\geq \delta && j = 0, 1, ..., n_N - 1; j \neq c
\end{aligned}
\tag{2}
$$

- Generally intractable (at least NP-complete)! [Weng et al. 18]

Formulation

- A N-layer fully connected neural network, parameterized by $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}_{i=1}^{N-1}$

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

- Given a model $\{\mathbf{W}^{(i)}, \mathbf{b}^{(i)}\}$ and a data point $\mathbf{x}$ labeled as $c \in \mathcal{C}$, we want to

$$\begin{aligned} &\min_{\boldsymbol{\epsilon}} \quad \left\{ -\sum_{j=0}^{n_1-1} \log \epsilon_j \right\} \\ &\hat{\mathbf{z}}^{(1)} \in \mathcal{S}_{\boldsymbol{\epsilon}}(\mathbf{x}) \\ &\mathbf{z}^{(i+1)} = \mathbf{W}^{(i)}\hat{\mathbf{z}}^{(i)} + \mathbf{b}^{(i)} && i = 1, 2, ..., N-1 \\ &\hat{\mathbf{z}}^{(i)} = \sigma(\mathbf{z}^{(i)}) && i = 2, 3, ..., N-1 \\ &l_c^{(N)} - u_j^{(N)} \geq \delta && j = 0, 1, ..., n_N - 1; j \neq c \end{aligned} \tag{2}$$

- Generally intractable (at least NP-complete)! [Weng et al. 18]
- Relax the output logits!

Optimization

- $\mathbf{l}^{(N)}$ and $\mathbf{u}^{(N)}$ are differentiable w.r.t. ϵ.

Optimization

- $\mathbf{l}^{(N)}$ and $\mathbf{u}^{(N)}$ are differentiable w.r.t. $\boldsymbol{\epsilon}$.
- The relaxation problem is tractable

$$\min_{\boldsymbol{\epsilon},\mathbf{y}\geq 0} \left\{ -\sum_{j=0}^{n_1-1} \log \epsilon_j \right\} \quad \text{s.t. } l_c^{(N)} - \mathbf{u}_{j\neq c}^{(N)} - \delta = \mathbf{y} \tag{3}$$

Optimization

- $\mathbf{l}^{(N)}$ and $\mathbf{u}^{(N)}$ are differentiable w.r.t. $\boldsymbol{\epsilon}$.
- The relaxation problem is tractable

$$\min_{\boldsymbol{\epsilon},\mathbf{y}\geq 0} \left\{ -\sum_{j=0}^{n_1-1} \log \epsilon_j \right\} \quad s.t.\ l_c^{(N)} - \mathbf{u}_{j\neq c}^{(N)} - \delta = \mathbf{y} \tag{3}$$

- The problem can be solved by Augmented Lagrangian Method

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\epsilon},\mathbf{y}\geq 0} -\left(\sum_{j=0}^{n_1-1} \log \epsilon_j \right) + \langle \boldsymbol{\lambda}, \mathbf{v} - \mathbf{y} \rangle + \frac{\rho}{2} \|\mathbf{v} - \mathbf{y}\|_2^2 \tag{4}$$

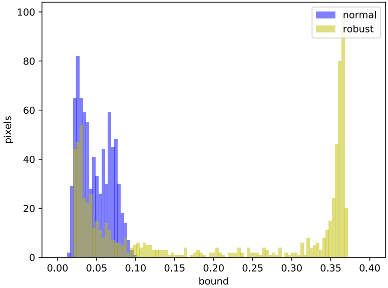
  - $\mathbf{v}$ is defined as $l_c^{(N)} - \mathbf{u}_{j\neq c}^{(N)} - \delta$

Dataset	Architecture	Training Method	Uniform	Non-uniform	Ratio
MNIST	100-100-100	-	0.0295	0.0349	1.183
		PGD, $\tau = 0.1$	0.0692	0.1678	2.425
	300-300-300	-	0.0309	0.0350	1.133
		PGD, $\tau = 0.1$	0.0507	0.1404	2.769
	500-500-500	-	0.0319	0.0360	1.129
		PGD, $\tau = 0.1$	0.0436	0.1167	2.677
Fashion-MNIST	1024-1024-1024	-	0.0397	0.0518	1.305
		PGD, $\tau = 0.1$	0.0446	0.1134	2.543
SVHN	1024-1024-1024	-	0.0022	0.0072	3.273
		PGD, $\tau = 0.1$	0.0054	0.0281	5.204

Table: Average of uniform and non-uniform bounds in the test sets.

- Larger volumes covered by non-uniform bounds, especially for robust models.

Experiments
Robustness and Feature Selection



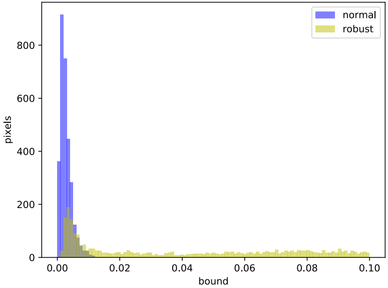


Figure: Examples of distributions of bounds for normal and robust models among all pixels. (Left: MNIST, Right: SVHN)

- Features of very large bounds → Features dropped

- We can visualize bounding map $\epsilon \in \mathbb{R}^n$ like an input data point.
- The bounding maps demonstrate better interpretability of robust models.

Figure: Left: between digit 1 and 7. Right: between digit 3 and 8. Lighter pixels mean smaller bounds.

More Details

- Welcome to Poster #63
- Code on GitHub: Certify_Nonuniform_Bounds

More Details