Towards Efficient Training and Evaluation Robust Models against l_0 Bounded Adversarial Perturbation

Xuyang Zhong, Yixiao Huang, Chen Liu*

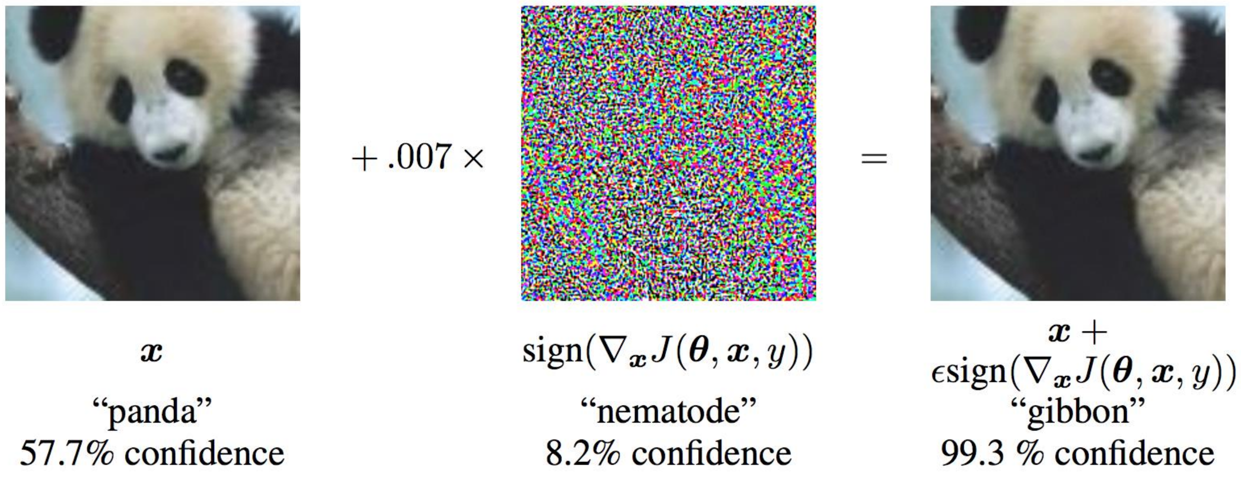
xuyang.zhong@my.cityu.edu.hk, chen.liu@cityu.edu.hk

City University of Hong Kong, University of Michigan

ICML 2024, Vienna, Austria

Introduction

Deep neural network is vulnerable to some imperceptible adversarial perturbations



[[1412.6572] Explaining and Harnessing Adversarial Examples (arxiv.org)]

Methods

$$\max_{\|\boldsymbol{\delta}\|_0 \le k, 0 \le \boldsymbol{x}+\boldsymbol{\delta} \le 1} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{\delta}) = \max_{\boldsymbol{p} \in \mathcal{S}_{\boldsymbol{p}}, \boldsymbol{m} \in \mathcal{S}_{\boldsymbol{m}}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}+\boldsymbol{p} \odot \boldsymbol{m})$$

- Decompose the l_0 bounded perturbation $\boldsymbol{\delta}$ into a magnitude tensor $\boldsymbol{p} \in \mathbb{R}^{h \times w \times c}$ and a sparsity mask $\boldsymbol{m} \in \{0,1\}^{h \times w \times 1}$
- $\mathcal{S}_{\boldsymbol{p}} = \{\boldsymbol{p} \in \mathbb{R}^{h \times w \times c} \mid 0 \le \boldsymbol{x} + \boldsymbol{p} \le 1\}$
- $\mathcal{S}_{\boldsymbol{m}} = \{\boldsymbol{m} \in \{0,1\}^{h \times w \times 1} \mid \|\boldsymbol{m}\|_{\boldsymbol{0}} \le k\}$
- We update $\boldsymbol{p}$ and $\boldsymbol{m}$ separately

Methods—Update $\boldsymbol{p}$

$$\boldsymbol{p} \longleftarrow \Pi_{\mathcal{S}_{\boldsymbol{p}}} \left(\boldsymbol{p} + \alpha \cdot \mathtt{sign}(\nabla_{\boldsymbol{p}} \mathcal{L}(\theta, \boldsymbol{x} + \boldsymbol{p} \odot \boldsymbol{m}))\right)$$

- Standard l_∞-bounded PGD to update the magnitude tensor $\boldsymbol{p}$
- $\prod_{\mathcal{S}_{\boldsymbol{p}}}$ is to clip $\boldsymbol{p}$ such that $0 \leq \boldsymbol{x} + \boldsymbol{p} \leq 1$

Methods—Update $\boldsymbol{m}$

$$\widetilde{\boldsymbol{m}} \longleftarrow \widetilde{\boldsymbol{m}} + \beta \cdot \nabla_{\widetilde{\boldsymbol{m}}}\mathcal{L}/||\nabla_{\widetilde{\boldsymbol{m}}}\mathcal{L}||_2,$$
$$\boldsymbol{m} \longleftarrow \Pi_{\mathcal{S}_{\boldsymbol{m}}}(\sigma(\widetilde{\boldsymbol{m}}))$$

- Instead updating a discrete $\boldsymbol{m}$, we update its continuous alternative $\widetilde{\boldsymbol{m}} \in \mathbb{R}^{h\times w\times 1}$
- Use l_2-bounded PGD to update $\widetilde{\boldsymbol{m}}$
- Project $\widetilde{\boldsymbol{m}}$ to the feasible set $\mathcal{S}_{\boldsymbol{m}}$ to get $\boldsymbol{m}$ before multiplying it with $\boldsymbol{p}$
- $\prod_{\mathcal{S}_{\boldsymbol{m}}}$ is to set the *k*–largest elements to 1 and the rest to 0
- σ denotes the sigmoid function

Methods—Sparse-PGD (sPGD)

Algorithm 1 Sparse-PGD

Input: Clean image: $\boldsymbol{x} \in [0,1]^{h\times w\times c}$; Model parameters: $\boldsymbol{\theta}$; Max iteration number: T; Tolerance: t; l_0 budget: k; Step size: α, β; Small constant: $\gamma = 2 \times 10^{-8}$

Random initialize $\boldsymbol{p}$ and $\widetilde{\boldsymbol{m}}$

for $i = 0, 1, ..., T-1$ **do**

    $\boldsymbol{m} = \Pi_{\mathcal{S}_m}(\sigma(\widetilde{\boldsymbol{m}}))$

    Calculate the loss $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x} + \boldsymbol{p} \odot \boldsymbol{m})$

    if unprojected **then**

        $g_{\boldsymbol{p}} = \nabla_{\boldsymbol{\delta}}\mathcal{L} \odot \sigma(\widetilde{\boldsymbol{m}})$ $\{\boldsymbol{\delta} = \boldsymbol{p} \odot \boldsymbol{m}\}$

    else

        $g_{\boldsymbol{p}} = \nabla_{\boldsymbol{\delta}}\mathcal{L} \odot \boldsymbol{m}$

    end if

    $g_{\widetilde{\boldsymbol{m}}} = \nabla_{\boldsymbol{\delta}}\mathcal{L} \odot \boldsymbol{p} \odot \sigma'(\widetilde{\boldsymbol{m}})$

    $\boldsymbol{p} = \Pi_{\mathcal{S}_p}(\boldsymbol{p} + \alpha \cdot \mathtt{sign}(g_{\boldsymbol{p}}))$

    $\boldsymbol{d} = g_{\widetilde{\boldsymbol{m}}}/(||g_{\widetilde{\boldsymbol{m}}}||_2)$ **if** $||g_{\widetilde{\boldsymbol{m}}}||_2 \geq \gamma$ **else** 0

    $\boldsymbol{m}_{old}, \widetilde{\boldsymbol{m}} = \boldsymbol{m}, \widetilde{\boldsymbol{m}} + \beta \cdot \boldsymbol{d}$

    if attack succeeds **then**

        break

    end if

    if $||\Pi_{\mathcal{S}_m}(\sigma(\widetilde{\boldsymbol{m}})) - \boldsymbol{m}_{old}||_0 \leq 0$ for t consecutive iters **then**

        Random initialize $\widetilde{\boldsymbol{m}}$

    end if

end for

Output: Perturbation: $\boldsymbol{\delta} = \boldsymbol{p} \odot \boldsymbol{m}$

Methods

- **Sparse-AutoAttack (sAA)**: A parameter-free ensemble of both sPGD and black-box attack for comprehensive robustness evaluation against l_0 bounded perturbations

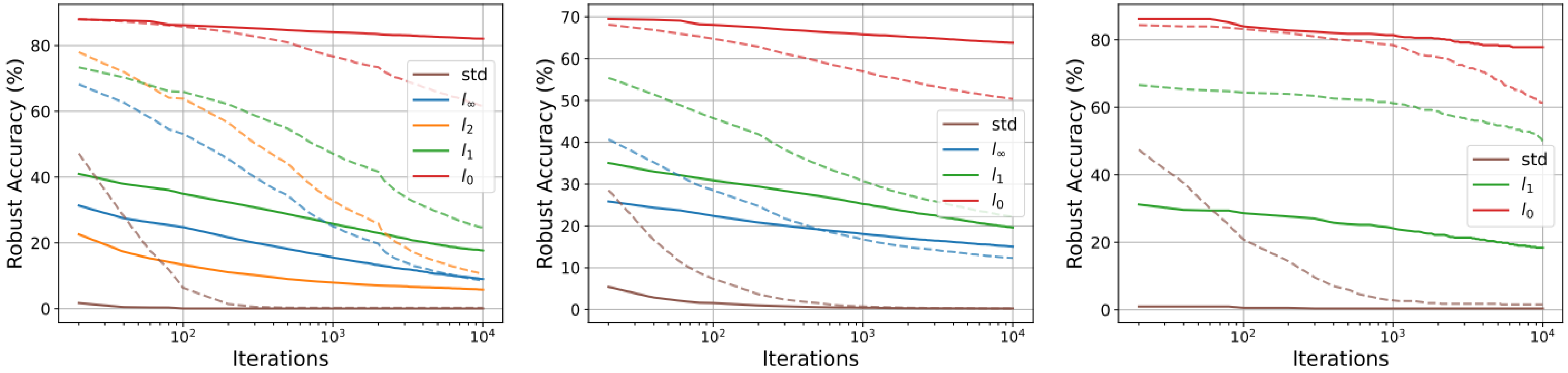
- **Adversarial training**: Build models against sparse perturbations. We incorporate sPGD in the framework of vanilla adversarial training (Madry et al., 2017) and TRADES (Zhang et al., 2019) and name corresponding methods **sAT** and **sTRADES**.

Experiments

Table 1. Robust accuracy of various models on different attacks that generate l_0 bounded perturbations, where the sparsity level $k = 20$. The models are trained on **CIFAR-10**. Note that we report results of Sparse-RS (RS) with fine-tuned hyperparameters, which outperforms its original version in Croce et al. (2022). CornerSearch (CS) is evaluated on 1000 samples due to its high computational complexity.

Model	Network	Clean	Black-Box		White-Box					sAA
			CS	RS	SF	PGD_0	SAIF	$\textbf{sPGD}_{\text{proj}}$	$\textbf{sPGD}_{\text{unproj}}$	
Vanilla	RN-18	93.9	1.2	0.0	17.5	0.4	3.2	0.0	0.0	**0.0**
l_∞-adv. trained, $\epsilon = 8/255$										
GD	PRN-18	87.4	26.7	6.1	52.6	25.2	40.4	9.0	15.6	**5.3**
PORT	RN-18	84.6	27.8	8.5	54.5	21.4	42.7	9.1	14.6	**6.7**
DKL	WRN-28	92.2	33.1	7.0	54.0	29.3	41.1	9.9	15.8	**6.1**
DM	WRN-28	92.4	32.6	6.7	49.4	26.9	38.5	9.9	15.1	**5.9**
l_2-adv. trained, $\epsilon = 0.5$										
HAT	PRN-18	90.6	34.5	12.7	56.3	22.5	49.5	9.1	8.5	**7.2**
PORT	RN-18	89.8	30.4	10.5	55.0	17.2	48.0	6.3	5.8	**4.9**
DM	WRN-28	95.2	43.3	14.9	59.2	31.8	59.6	13.5	12.0	**10.2**
FDA	WRN-28	91.8	43.8	18.8	64.2	25.5	57.3	15.8	19.2	**14.1**
l_1-adv. trained, $\epsilon = 12$										
l_1-APGD	PRN-18	80.7	32.3	25.0	65.4	39.8	55.6	17.9	18.8	**16.9**
Fast-EG-l_1	PRN-18	76.2	35.0	24.6	60.8	37.1	50.0	18.1	18.6	**16.8**
l_0-adv. trained, $k = 20$										
PGD_0-A	PRN-18	77.5	16.5	2.9	62.8	56.0	47.9	9.9	21.6	**2.4**
PGD_0-T	PRN-18	90.0	24.1	4.9	85.1	61.1	67.9	27.3	37.9	**4.5**
sAT	PRN-18	84.5	52.1	36.2	81.2	78.0	76.6	75.9	75.3	**36.2**
sTRADES	PRN-18	89.8	69.9	61.8	88.3	86.1	84.9	84.6	81.7	**61.7**

Experiments



(a) CIFAR-10, $k = 20$ (b) CIFAR-100, $k = 10$ (c) ImageNet-100, $k = 200$

Comparison between sPGD and Sparse-RS attack under different iterations

Solid: sPGD
Dashed: a strong black-box attack Sparse-RS

Conclusion

1. We propose an effective and efficient attack algorithm called sparse-PGD (sPGD) to generate l_0 bounded adversarial perturbation.
2. We propose an ensemble of sparse attacks called sparse-AutoAttack (sAA) for reliable robustness evaluation against l_0 bounded perturbation.
3. We conduct extensive experiments to demonstrate that our attack methods achieve impressive performance in terms of both effectiveness and efficiency.