Towards Stable and Efficient Adversarial Training against l_1 Bounded Adversarial Attacks

Yulun Jiang*[1], Chen Liu*[2], Zhichao Huang[3], Mathieu Salzmann[1], Sabine Süsstrunk[1]

[1] EPFL, [2] City University of Hong Kong, [3] ByteDance

ICML 2023

* Equal Contribution

Background

Network parameterized by $\theta \in \mathbb{R}^n$, the training set $\{\boldsymbol{x}_i\}_{i=1}^N$, the loss function $\mathcal{L}$, the adversarial budget $\mathcal{S}_\epsilon^{(p)} := \{\Delta | \|\Delta\|_p \leq \epsilon\}$, we solve the robust learning problem.

$$\min_\theta \sum_{i=1}^{N} \max_{\Delta \in \mathcal{S}_\epsilon^{(p)}} \mathcal{L}(\theta, \boldsymbol{x}_i + \Delta) \tag{1}$$

Background

Network parameterized by $\theta \in \mathbb{R}^n$, the training set $\{\boldsymbol{x}_i\}_{i=1}^N$, the loss function $\mathcal{L}$, the adversarial budget $\mathcal{S}_\epsilon^{(p)} := \{\Delta | \|\Delta\|_p \leq \epsilon\}$, we solve the robust learning problem.

$$\min_\theta \sum_{i=1}^N \max_{\Delta \in \mathcal{S}_\epsilon^{(p)}} \mathcal{L}(\theta, \boldsymbol{x}_i + \Delta) \tag{1}$$

To generate adversarial examples, we maximize $\mathcal{L}(\theta, \boldsymbol{x}_i + \Delta) \simeq \mathcal{L}(\theta, \boldsymbol{x}_i) + \langle \Delta, \nabla \mathcal{L} \rangle$

Background

Network parameterized by $\theta \in \mathbb{R}^n$, the training set $\{\boldsymbol{x}_i\}_{i=1}^N$, the loss function $\mathcal{L}$, the adversarial budget $\mathcal{S}_\epsilon^{(p)} := \{\Delta | \|\Delta\|_p \leq \epsilon\}$, we solve the robust learning problem.

$$\min_\theta \sum_{i=1}^{N} \max_{\Delta \in \mathcal{S}_\epsilon^{(p)}} \mathcal{L}(\theta, \boldsymbol{x}_i + \Delta) \tag{1}$$

To generate adversarial examples, we maximize $\mathcal{L}(\theta, \boldsymbol{x}_i + \Delta) \simeq \mathcal{L}(\theta, \boldsymbol{x}_i) + \langle \Delta, \nabla \mathcal{L} \rangle$

- $p = \infty$

$$\Delta \leftarrow \Pi_{\mathcal{S}_\epsilon^{(\infty)}} (\Delta + \alpha \ \mathrm{sign}(\nabla_\Delta \mathcal{L}))$$

- $p = 2$

$$\Delta \leftarrow \Pi_{\mathcal{S}_\epsilon^{(2)}} (\Delta + \alpha \ \nabla_\Delta \mathcal{L} / \|\nabla_\Delta \mathcal{L}\|_2)$$

Background

Things get more difficult in the case of l_1 adversarial budget.

Background

Things get more difficult in the case of l_1 adversarial budget.

- Theoretically, **one-hot coordinate descent**

$$\Delta \leftarrow \Pi_{\mathcal{S}_\epsilon^{(1)}} \left(\Delta + \alpha \ \mathbf{1}(i = j_{max})\right), \ j_{max} = \arg\max_i |\nabla_\Delta \mathcal{L}|_i$$

Background

Things get more difficult in the case of l_1 adversarial budget.

- Theoretically, **one-hot coordinate descent**

$$\Delta \leftarrow \Pi_{\mathcal{S}_\epsilon^{(1)}} \left(\Delta + \alpha \ \mathbf{1}(i = j_{max}) \right), \ j_{max} = \arg\max_i |\nabla_\Delta \mathcal{L}|_i$$

- Empirically, **K-hot coordinate descent**

$$\Delta \leftarrow \Pi_{\mathcal{S}_\epsilon^{(1)}} \left(\Delta + \alpha / K \ \mathbf{1}(i \in \mathcal{S}_{max}) \right)$$

$$\mathcal{S}_{max} = \{ i | i \text{ is among the top K coordinates of } \nabla_\Delta \mathcal{L} \text{ in absolute magnitude} \}$$

Motivation & Challenges

Motivation:

- We aims to design **stable** and **efficient** adversarial training against l_1 bounded adversarial attacks.

Motivation & Challenges

Motivation:

- We aims to design **stable** and **efficient** adversarial training against l_1 bounded adversarial attacks.

Challenges:

- **Stability**: Catastrophic overfitting happens more frequently in the l_1 cases.
- **Efficiency**: The complexity of the SOTA method in the l_1 cases is much higher than those in the l_2 and l_∞ cases.
- Existing efficient robust learning methods are proposed for the l_2 or l_∞ adversarial budgets, naively extending them to the l_1 cases yields suboptimal performance.

Analysis

Key take away: **coordinate descent contributes to catastrophic overfitting.**

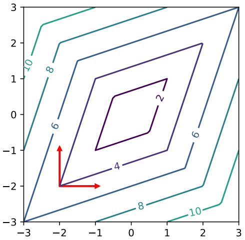

Figure: An example of coordinate descent trapped in suboptimality with non-smooth functions: at the point $(-2, -2)$ of the function $2 \times |x - y| + |x + y|$.

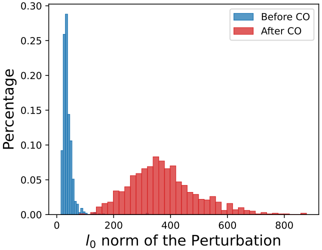

Figure: Distributions of the l_0 norm of the perturbations generated by AutoAttack (AA) before and after catastrophic overfitting (CO).

Method

Generate l_1 bounded perturbations by Euclidean geometry, i.e., no coordinate descent.

- $\Delta \leftarrow \Pi_{\mathcal{S}_\epsilon^{(1)}} (\Delta + \alpha \, \nabla_\Delta \mathcal{L} / \|\nabla_\Delta \mathcal{L}\|_2)$.
- Perturbations updated by Euclidean geometry but projected to l_1 budgets.
- One step attack with random initialization to improve efficiency.
- α is chosen that one step update by Euclidean geometry can cover the area of what coordinate descent can explore, i.e., $\alpha = \sqrt{\epsilon}$.
- Multi-ϵ trick to encourage adversarial example exploration during training.

Advantages:

- Efficient and stable, free of catastrophic overfitting.
- No memory overhead, scalable to large dataset.
- No more hyper-parameters, no need for finetuning.

Results

Method	CIFAR10 ($\epsilon = 12$)		CIFAR100 ($\epsilon = 6$)		ImageNet100 ($\epsilon = 72$)	
	AA (%)	Time (h)	AA (%)	Time (h)	AA (%)	Time (h)
AutoPGD	55.77	2.58	42.18	2.58	-	-
FGSM-RS	36.29	0.76	33.23	0.71	36.64	22.12
ATTA	46.57	0.67	33.74	0.68	-	-
AdaAT	31.84	0.88	28.64	0.84	28.62	26.96
Grad-Align	36.38	1.52	33.19	1.52	-	-
N-FGSM	44.21	0.65	35.79	0.66	30.28	23.53
NuAT	48.35	1.01	36.46	1.05	45.82	29.18
Fast-EG-l_1	**50.27**	0.67	**38.03**	0.67	**46.74**	22.11

Table: Robust accuracy (in %) evaluated by AutoAttack (AA) and training time in hours when we run different methods on CIFAR10, CIFAR100, and ImageNet100. Hyper-parameters of baselines are finetuned. The results of AutoPGD, ATTA and Grad-Align on ImageNet100 are not reported because of prohibitively-high computational or memory overhead.

Thank You!

Full Paper

Code