Understanding and Improving Fast Adversarial Training against l_0 Bounded Perturbations

Xuyang Zhong [1] Yixiao Huang [1] Chen Liu [1]

[1]City University of Hong Kong

{xuyang.zhong, yixiao.huang}@my.cityu.edu.hk chen.liu@cityu.edu.hk

Unique Challenges in Fast l_0 Adversarial Training

Fast adversarial training is efficient but usually encounters **catastrophic overfitting (CO)** – The model trained by 1-step attack, e.g., sPGD, shows zero robustness to a stronger attack, e.g., sAA.

Most methods designed for other l_p norms ($p \geq 1$) turn out **ineffective** at all in the l_0 scenario.

Table 1. Comparison between existing CO mitigation methods and multi-step method (sTRADES) in robust accuracy (%) by sAA. The target sparsity level $\epsilon = 20$.

Method	ATTA	Free-AT	GA	Fast-BAT	FLC Pool	N-AAER	N-LAP	NuAT	sTRADES
Robust Acc.	*0.0*	8.9	*0.0*	14.1	*0.0*	*0.1*	*0.0*	51.9	61.7

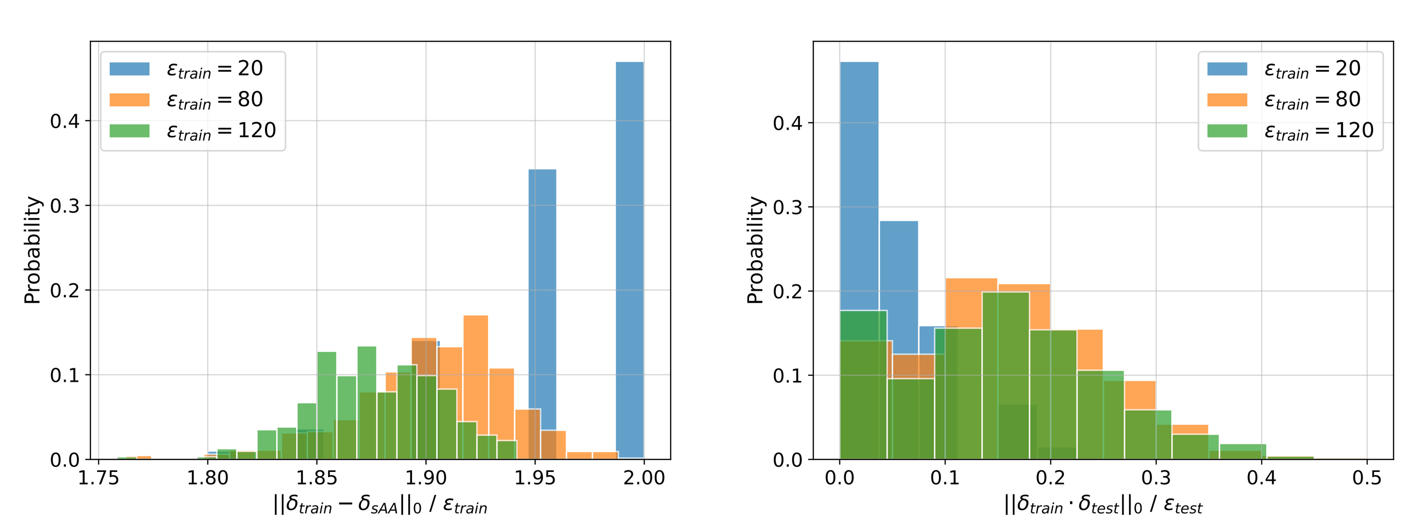
CO in l_0 adversarial training is primarily due to **sub-optimal perturbation locations rather than magnitudes:**

(1) We cannot find successful adversarial examples through simple interpolations.

Table 2. Robust accuracy of the models obtained by 1-step sAT against the interpolation between perturbations generated by 1-step sPGD and clean examples, where α denotes the interpolation factor, i.e., $\boldsymbol{x}_{interp} = \boldsymbol{x} + \alpha \cdot \boldsymbol{\delta}$.

α	0.0	0.2	0.4	0.6	0.8	1.0	**sAA**
$\epsilon_{train} = 20$	77.5	**69.1**	80.4	88.0	90.2	90.4	**0.0**
$\epsilon_{train} = 40$	70.2	**64.3**	79.8	87.4	89.6	89.6	**0.0**
$\epsilon_{train} = 120$	32.5	**24.5**	41.5	65.2	72.8	67.6	**0.0**

(2) Perturbations generated by 1-step sPGD are almost completely different from those generated by sAA in location.



(a) Location Difference (b) Location Overlapping

Figure 1. Visualization of location difference and location overlapping.

The sub-optimal location issue can be mitigated to some extent by multi-ϵ strategy. However, a larger ϵ_{train}, in turn, leads to **unstable training and degraded clean accuracy**. To address this challenge, we investigate the loss landscape of l_0 adversarial training.

Analysis on the Smoothness of Adversarial Loss

If the model's output logits $\{f_i\}_{i=0}^{K-1}$ is Lipschitz continuous and smooth w.r.t. model parameter $\boldsymbol{\theta}$ and input $\boldsymbol{x}$.

Theorem 1 (Lipschitz continuity of adversarial loss function)

$$\forall \boldsymbol{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \quad \|\mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_1) - \mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_2)\| \leq A_{\boldsymbol{\theta}} \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|, \tag{1}$$

The constant $A_{\boldsymbol{\theta}} = 2\sum_{i \in \mathcal{S}_+} y_i L_{\boldsymbol{\theta}}$ where $\mathcal{S}_+ = \{i \mid y_i \geq 0, h_i(\boldsymbol{x} + \boldsymbol{\delta}_1, \boldsymbol{\theta}_2) > h_i(\boldsymbol{x} + \boldsymbol{\delta}_1, \boldsymbol{\theta}_1)\}$, $\boldsymbol{\delta}_1 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta})$ and $\boldsymbol{\delta}_2 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta})$. h is the output probability after softmax.

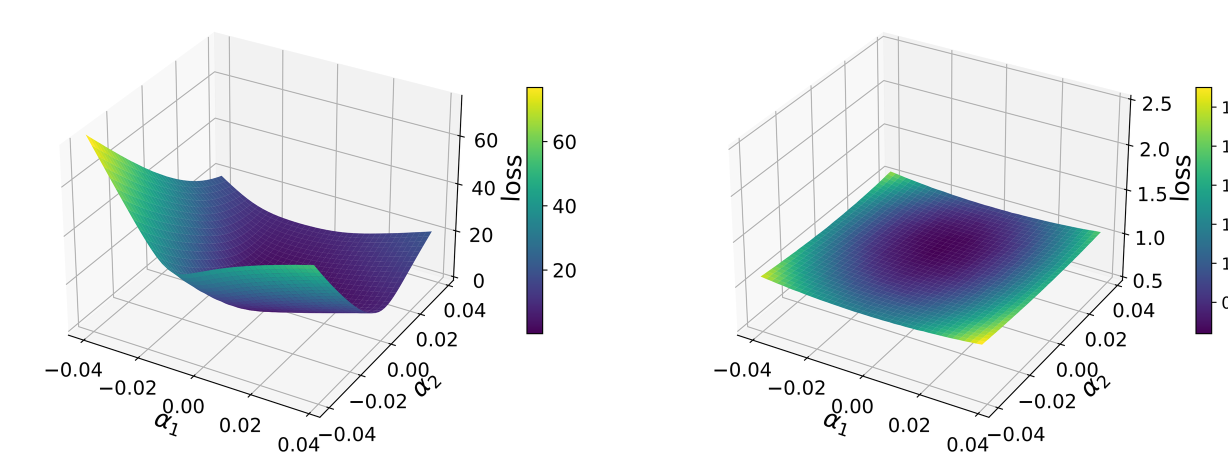
Theorem 2 (Lipschitz smoothness of adversarial loss function)

$$\forall \boldsymbol{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \quad \|\nabla_{\boldsymbol{\theta}} \mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_1) - \nabla_{\boldsymbol{\theta}} \mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta}_2)\| \leq A_{\boldsymbol{\theta\theta}} \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\| + B_{\boldsymbol{\theta\delta}} \tag{2}$$

The constants $A_{\boldsymbol{\theta\theta}} = L_{\boldsymbol{\theta\theta}}$ and $B_{\boldsymbol{\theta\delta}} = L_{\boldsymbol{\theta x}} \|\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2\| + 4L_{\boldsymbol{\theta}}$ where $\boldsymbol{\delta}_1 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}_1)$ and $\boldsymbol{\delta}_2 \in \arg\max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta}_2)$.

The upper bound of $\|\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2\|$ in the l_0 case is significantly larger than other cases, **indicating a more craggy loss landscape in l_0 adversarial training.**

Numerical results also validate the conclusions in theoretical analyses.



(a) $\mathcal{L}_\epsilon^{(0)}$, $\epsilon_{train} = 1$ (b) $\mathcal{L}_\epsilon^{(1)}$, $\epsilon_{train} = 24$

Figure 2. The loss landscape of $\mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta} + \alpha_1 \boldsymbol{v}_1 + \alpha_2 \boldsymbol{v}_2)$ where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the eigenvectors associated with the top 2 eigenvalues of $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}_\epsilon(\boldsymbol{x}, \boldsymbol{\theta})$, respectively.

Recipe: Soft Label and Trade-off Loss Function

1. Let $\boldsymbol{y}_h \in \{0, 1\}^K$ and $\boldsymbol{y}_s \in (0, 1)^K$ denote the hard and soft label, respectively. We find that **soft label $\boldsymbol{y}_s$ leads to a reduced first-order Lipschitz constant**.
2. Introduce a trade-off loss function $\mathcal{L}_{\epsilon,\alpha}(\boldsymbol{x}, \boldsymbol{\theta}) = (1 - \alpha)\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) + \alpha \max_{\boldsymbol{\delta} \in \mathcal{S}_\epsilon(\boldsymbol{x})} \mathcal{L}(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{\theta})$, where $\alpha \in [0, 1]$ is the interpolation factor. We find that **trade-off loss function can improve the Lipschitz smoothness**.

Experiments

We try different techniques incorporating soft labels or/and trade-off loss function, and name the best combination **Fast-LS-l_0**, i.e., 1-step sTRADES + SAT + N-FGSM.

Compared to traditional CO-mitigation methods, Fast-LS-l_0 **successfully mitigate CO in the l_0 case**, and **greatly narrow down the performance gaps between 1-step and multi-step adversarial training**. Additionally, our method can improve the performance of multi-step adversarial training.

Table 3. Robust accuracy (%) against sparse attacks. **(a)** PreActResNet-18 trained on **CIFAR-10**, where the attack sparsity level $\epsilon = 20$. **(b)** ResNet-34 trained on **ImageNet-100**, where $\epsilon = 200$. CornerSearch (CS) is not evaluated due to its high computational complexity. Cost times are recorded on one NVIDIA RTX 6000 Ada.

(a) **CIFAR-10,** $\epsilon = 20$

Model	Time Cost	Clean	Black CS	Black RS	White SAIF	White σ-zero	White sPGD$_p$	White sPGD$_u$	sAA
Multi-step									
sAT	5.3 h	84.5	52.1	36.2	76.6	79.8	75.9	75.3	36.2
+S&N	5.5 h	80.8	64.1	61.1	76.1	78.7	76.8	75.1	61.0
sTRADES	5.5 h	89.8	69.9	61.8	84.9	85.9	84.6	81.7	61.7
+S&N	5.4 h	82.2	66.3	66.1	77.1	77.0	74.1	72.2	**65.5**
One-step									
Fast-LS-l_0	0.8 h	82.5	69.3	65.4	75.7	73.7	67.2	67.7	**63.0**

(b) **ImageNet-100,** $\epsilon = 200$

Model	Time Cost	Clean	Black RS	White SAIF	White σ-zero	White sPGD$_p$	White sPGD$_u$	sAA
Multi-step								
sAT	325 h	86.2	61.4	69.0	78.6	78.0	77.8	61.2
+S&N	336 h	83.0	75.0	76.4	80.8	78.8	79.2	74.8
sTRADES	359 h	84.8	76.0	77.4	81.6	80.6	81.4	75.8
+S&N	360 h	82.4	78.2	79.2	80.0	78.2	79.8	**77.8**
One-step								
Fast-LS-l_0	44 h	82.4	76.8	75.4	74.0	74.6	74.6	**72.4**

Takeaway Messages

1. Catastrophic overfitting (CO) in fast l_0 AT arises from **sub-optimal perturbation locations**. Although multi-ϵ strategy can mitigate this issue to some extent, it leads to unstable training.
2. We prove that the **adversarial loss landscape is more craggy in l_0 cases**. In this regard, **soft labels** and the **trade-off loss function** can be used to provably smooth the adversarial loss landscape.
3. Experiments show that our method can not only mitigate CO issue but also improve the performance of multi-step adversarial training.

Extension - Structured Sparse Perturbation

Our work "**Sparse-PGD: A Unified Framework for Sparse Adversarial Perturbations Generation**" was recently accepted by **TPAMI**. Please scan the QR code for details.