Revisiting and Advancing Fast Adversarial Training through the Lens of Bi-Level Optimization

Yihua Zhang[1,*], Guanhua Zhang[2,*], Prashant Khanduri[3], Mingyi Hong[3], Shiyu Chang[2], Sijia Liu[1,4]
[1]Michigan State University, [2]UC Santa Barbara, [3]University of Minnesota, [4]MIT-IBM Watson Lab

ICML International Conference On Machine Learning

Paper Code

➢Motivations

How to design a 'fast' AT with improved stability, mitigated catastrophic overfitting, and theoretical guarantees?

➢Fast Robust Training: Not Enough!

- ❖ Low Stability;
- ❖ Robust Catastrophic Overfitting;
- ❖ Robustness at Cost of Sharp Drop in Accuracy.

➢Bi-level Optimization: A Fresh Perspective!

❖ Standard min-max formulation for adversarial training:

$$\min_{\theta} \mathrm{E}_{(\boldsymbol{x},t)\sim D}\left[\max_{|\delta|_\infty\le\epsilon} \ell_{\mathrm{tr}}(\boldsymbol{\theta};\boldsymbol{x}+\boldsymbol{\delta},t)\right]$$

❖ Bi-level Optimization (BLO) framework:

$$\min_{\theta} \ell_{\mathrm{tr}}(\theta,\boldsymbol{\delta}^*(\theta))$$

$$\text{s.t.}\quad \boldsymbol{\delta}^*(\boldsymbol{\theta}) = \mathrm{argmin}_{\boldsymbol{\delta}\in C}\ \ell_{\mathrm{atk}}(\theta,\delta)$$

✓ Customizable lower-level objectives: $\ell_{\mathrm{atk}} \neq -\ell_{\mathrm{tr}}$!

✓ BLO is a generalization form of min-max optimization

✓ Presence of implicit gradient (IG): the 'fingerprint' of BLO

The upper-level gradient calculation:

$$\frac{\mathrm{d}\ell_{\mathrm{tr}}(\boldsymbol{\theta},\boldsymbol{\delta}^*(\boldsymbol{\theta}))}{\mathrm{d}\theta} = \frac{\partial\ell_{\mathrm{tr}}(\boldsymbol{\theta},\boldsymbol{\delta}^*(\boldsymbol{\theta}))}{\partial\theta} + \underbrace{\frac{\mathrm{d}\boldsymbol{\delta}^*(\boldsymbol{\theta})^T}{\mathrm{d}\theta}}_{\text{IG}} \frac{\partial\ell_{\mathrm{tr}}(\boldsymbol{\theta},\boldsymbol{\delta}^*(\boldsymbol{\theta}))}{\partial\delta}$$

IG: $\boldsymbol{\delta}^*(\boldsymbol{\theta})$ is an implicit function of $\boldsymbol{\theta}$

✓ BLO is hard to solve, lower-level constraint makes it harder!

➢ Customize Your Attack Loss : Fast-BAT!

❖ Linearization at $\boldsymbol{z}$ with quadratic regularization:

$$\ell'_{\mathrm{atk}}(\boldsymbol{\theta},\boldsymbol{\delta}) = <\nabla_{\boldsymbol{\delta}=\boldsymbol{z}}\ \ell_{\mathrm{atk}}(\boldsymbol{\theta},\boldsymbol{\delta}),\boldsymbol{\delta}-\boldsymbol{z}> + \left(\frac{\lambda}{2}\right)||\boldsymbol{\delta}-\boldsymbol{z}||_2^2$$

❖ Fast-BAT Algorithm

✓ One-step lower-level solver: **unique**, **efficient**, **closed-form**

$$\boldsymbol{\delta}^*(\boldsymbol{\theta}) = \mathrm{Proj}_C(\boldsymbol{z} - (1/\lambda)\nabla_{\boldsymbol{\delta}}\ell_{\mathrm{atk}}(\boldsymbol{\theta},\boldsymbol{z}))$$

✓ Tractable IG with KKT conditions and Hessian-free assumption.

$$\frac{\mathrm{d}\boldsymbol{\delta}^*(\boldsymbol{\theta})^T}{\mathrm{d}\theta} = -\left(\frac{1}{\lambda}\right)\nabla_{\theta\delta}\ell_{\mathrm{atk}}(\theta,\delta^*)H_C$$

➢Fast-AT[1]: Live by 'Sign', Die from 'Sign' !

❖ Fast BAT with **gradient sign-based** linearization:

$$\ell'_{\mathrm{atk}}(\boldsymbol{\theta},\boldsymbol{\delta}) = <\mathrm{sign}(\nabla_{\boldsymbol{\delta}}\ \ell_{\mathrm{atk}}(\boldsymbol{\theta},\boldsymbol{z})),\boldsymbol{\delta}-\boldsymbol{z}> + \left(\frac{\lambda}{2}\right)||\boldsymbol{\delta}-\boldsymbol{z}||_2^2$$

❖ Fast-AT Algorithm

✓ One-step lower-level solver: **unique**, **efficient**, **closed-form**

$$\boldsymbol{\delta}^*(\boldsymbol{\theta}) = \mathrm{Proj}_C(\boldsymbol{z} - (1/\lambda)\,\mathrm{sign}(\nabla_{\boldsymbol{\delta}}\ell_{\mathrm{atk}}(\boldsymbol{\theta},\boldsymbol{z})))$$

✓ IG degrades to zero! BLO degrades to MMO!

$$\frac{\mathrm{d}\boldsymbol{\delta}^*(\boldsymbol{\theta})^T}{\mathrm{d}\theta} = 0$$

❖ Why not use Fast-AT?

✓ Sign operation is non-smooth that affects training stability!

✓ Sign operation is sub-optimal for linearization!

Table 1. Performance overview of proposed FAST-BAT vs. the baselines FAST-AT [1] and FAST-AT-GA [2] on various datasets with PreActResNet-18.

Method	SA (%) ($\epsilon = 8/255$)	RA-PGD (%) ($\epsilon = 8/255$)	RA-AA (%) ($\epsilon = 8/255$)	SA (%) ($\epsilon = 16/255$)	RA-PGD (%) ($\epsilon = 16/255$)	RA-AA (%) ($\epsilon = 16/255$)	Time (s/epoch)
CIFAR-10, PreActResNet-18							
FAST-AT	**82.39**±0.14	45.49±0.21	41.87±0.15	44.15±7.27	21.83±1.32	12.49±0.33	23.1
FAST-AT-GA	79.71±0.24	47.27±0.22	43.24±0.27	58.29±1.32	26.01±0.16	17.97±0.33	75.3
FAST-BAT	79.97±0.12	**48.83**±0.17	**45.19**±0.12	**68.16**±0.25	**27.69**±0.16	**18.79**±0.24	61.4
CIFAR-100, PreActResNet-18							
FAST-AT	**52.62** ±0.18	24.66±0.21	21.72±0.17	21.32±3.27	8.62±1.03	6.22±0.61	23.8
FAST-AT-GA	50.06±0.27	24.97±0.23	21.82±0.21	32.51±1.27	12.27±0.36	9.43±0.19	77.1
FAST-BAT	50.19±0.21	**26.49**±0.20	**23.97**±0.15	**39.29**±0.53	**13.97**±0.17	**11.32**±0.22	61.6
Tiny-ImageNet, PreActResNet-18							
FAST-AT	41.37±3.08	17.05±3.25	12.31±2.73	31.38±0.19	5.42±2.17	3.13±0.24	284.6
FAST-AT-GA	45.52±0.24	20.39±0.19	16.25±0.17	29.17±0.32	6.79±0.27	4.27±0.15	592.7
FAST-BAT	**45.80**±0.22	**21.97**±0.21	**17.64**±0.15	**33.78**±0.23	**8.83**±0.22	**5.52**±0.14	572.4

Table 2. Performance comparison on CIFAR-10 with various model architectures.

Model	Method	SA(%) ($\epsilon = 8/255$)	RA-PGD(%) ($\epsilon = 8/255$)	SA(%) ($\epsilon = 16/255$)	RA-PGD(%) ($\epsilon = 16/255$)
PARN-50	FAST-AT	73.15±6.10	41.03±2.99	43.86±4.31	22.08±0.27
	FAST-AT-GA	77.40±0.81	46.16±0.98	42.28±6.69	22.87±1.25
	PGD-2-AT	**83.53**±0.17	46.17±0.59	68.88±0.39	22.37±0.41
	FAST-BAT	78.91±0.68	**49.18**±0.35	**69.01**±0.19	**24.55**±0.06
WRN-16-8	FAST-AT	84.39±0.46	45.80±0.57	49.39±2.17	21.99±0.41
	FAST-AT-GA	81.51±0.38	48.29±0.20	45.95±13.65	23.10±3.90
	PGD-2-AT	**85.52**±0.14	45.47±0.14	**72.11**±0.33	23.61±0.16
	FAST-BAT	81.66±0.54	**49.93**±0.36	68.12±0.47	**25.63**±0.44

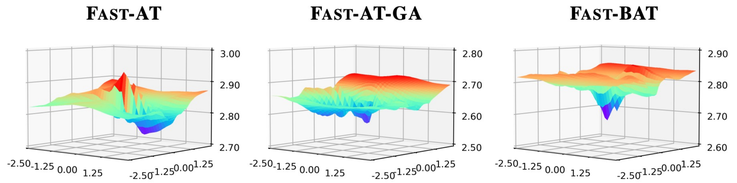

Figure 1. Visualization of adversarial loss landscapes (ResNet18, CIFAR10)

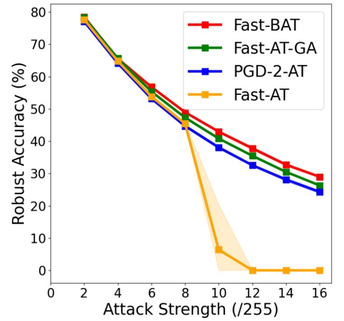

Figure 2. Robustness against different training attack strengths.

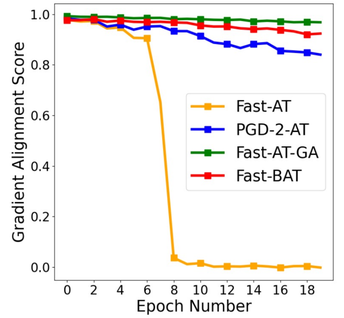

Figure 3. Training curves of Gradient Alignment score. Fast-BAT achieves 'gradient alignment' for free!

ICML WORKSHOP

[1] Eric Wong et al. "Fast is Better Than Free: Revisiting Adversarial Training." ICLR 2020.
[2] Maksym Andriushchenko et al. "Understanding and Improving Fast Adversarial Training." NeurIPS 2020.

Contact: {zhan1908, liusiji5}@msu.edu, {guanhua, shiyu}@ucsb.edu, {khand095,mhong}@umn.edu