uai2022

IBM Research · MICHIGAN STATE UNIVERSITY

Distributed Adversarial Training to Robustify Deep Neural Networks at Scale

Gaoyuan Zhang[1*], Songtao Lu[1*], Yihua Zhang[2], Xiangyi Chen[3], Pin-Yu Chen[1], Quanfu Fan[1], Lee Martie [1], Lior Horesh [1], Mingyi Hong[3], Sijia Liu[1,2]

[1]IBM Research, [2]Michigan State University, [3]University of Minnesota

Paper Code

Motivations

- ❖ Current accelerated AT algorithms all suffer from various problems.
- ❖ Distributed ML is effective for standard training.

Research Question

How to scale up Adversarial Training with distributed machine learning to large models and datasets?

DAT: A Distributed AT Framework

- ❖ We present a general algorithmic framework for Distributed AT.
- ❖ We provide convergence analysis of DAT in general non-convex settings.
- ❖ Experiments in various AT settings
  - ✓ robust training on ImageNet
  - ✓ semi-supervised AT
  - ✓ certified robust training
  - ✓ robust pretrain + finetuning

Problem Formulation

❖ Adversarial Training

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

❖ **Distributed Adversarial Training**

$$\text{minimize}_{\boldsymbol{\theta}} \; \frac{1}{M}\sum_{i=1}^{M} f_i(\theta, D_i)$$

$$f_i =: \mathrm{E}_{(\boldsymbol{x},t)\sim D_i}[\lambda \ell_{\mathrm{tr}}(\boldsymbol{\theta}; x, t) + \max_{\|\boldsymbol{\delta}\|_\infty\le\epsilon} \phi(\boldsymbol{\theta}; \boldsymbol{x}+\boldsymbol{\delta}, t)]$$

f_i	Local cost function at i-th worker
D_i	Local dataset at i-th worker, $D = \cup_{i=1}^{M} D_i$
$\{\boldsymbol{x}+\boldsymbol{\delta}: \|\boldsymbol{\delta}\|_\infty \le \epsilon\}$	ℓ_∞ adversarial attack with strength ϵ
λ	Balance between training loss and robustness.

Algorithmic Framework of DAT

Algorithm 1 Meta-version of DAT (Alg. A1 in Supplement)

```
1:  for Worker i = 1, 2, ..., M do                      ▷ Block 1
2:      Sample-wise attack generation (A1)
3:      Local gradient computation (A2)
4:      Worker-server communication
5:  end for
6:  Gradient aggregation at server (A3)                 ▷ Block 2
7:  Server-worker communication
8:  for Worker i = 1, 2, ..., M do                      ▷ Block 3
9:      Model parameter update (A4)
10: end for
```

Large-batch Challenge

❖ Adversarial training suffers from performance degradation with large batches.

❖Solution: Layer-wise Adaptive Learning Rate

$$\theta_{t+1,i} = \theta_{t,i} - \tau(\|\theta_{t,i}\|_2)\cdot\eta_t \frac{u_{t,i}}{\|u_{t,i}\|_2}$$

$$\tau(\|\theta_{t,i}\|_2) = \min(\max((\|\theta_{t,i}\|_2, c_l), c_u)$$

Gradient Quantization

❖ Reduce computational costs by using fewer bits to store gradient information.

Challenges in Convergence Analysis

❖ Non-linear error coupling from:

- ✓ gradient estimation
- ✓ gradient quantization
- ✓ LALR
- ✓ inner maximization oracle

Table 1. DAT (in gray color) on (ImageNet, ResNet-50) compared with baselines in Standard Accuracy (TA), Robust Accuracy against PGD (RA) and AutoAttack (AA), communication time per epoch (C) and total training time per epoch (T). 'p × q' represents '# nodes × # GPUs per node'.

Method	ImageNet, ResNet-50						
	$p \times q$	Batch size	TA (%)	RA (%)	AA (%)	C (s)	T (s)
AT	1 × 6	512	62.70	40.38	37.46	NA	6022
DAT-PGD w/o LALR	6 × 6	6 × 512	57.09	34.02	30.98	865	1932
DAT-PGD	6 × 6	6 × 512	63.75	38.45	36.04	898	1960
Fast AT	1 × 6	512	58.99	40.78	37.18	NA	1544
DAT-FGSM w/o LALR	6 × 6	6 × 512	55.04	35.03	32.16	863	1080
DAT-FGSM	6 × 6	6 × 512	58.02	40.27	36.02	859	1109

Table 2. Certified accuracy (%) of smooth classifiers on (CIFAR-10, ResNet-18) versus ℓ_2 radii.

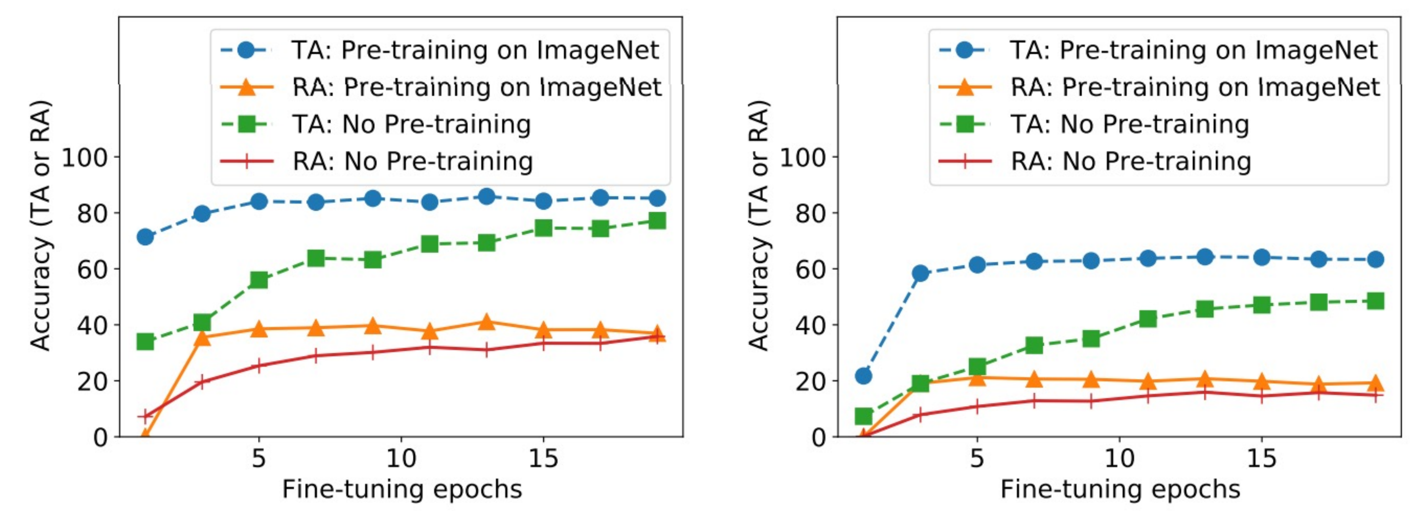
Method	Smooth classifier ($\sigma = 0.12$)						
	$r=0.05$	$r=0.1$	$r=0.15$	$r=0.2$	$r=0.3$	$r=0.4$	$r=0.5$
Baseline ($N=2$)	0.832	0.804	0.762	0.728	0.654	0.545	0
DAT ($N=20$)	0.838	0.812	0.784	0.748	0.661	0.550	0
Method	Smooth classifier ($\sigma = 0.25$)						
	$r=0.05$	$r=0.1$	$r=0.15$	$r=0.2$	$r=0.3$	$r=0.4$	$r=0.5$
Baseline ($N=2$)	0.752	0.730	0.708	0.678	0.625	0.562	0.498
DAT ($N=20$)	0.764	0.748	0.716	0.688	0.632	0.566	0.514

Table 3. DAT with semi-supervision using ResNet-18 or Wide ResNet-28-10 under CIFAR-10 + 500K unlabeled Tiny Images.

Method	ResNet-18, batch size 12 × 2048				
	TA (%)	RA (%)	AA (%)	C(s)	T(s)
DAT-PGD	87.00	47.34	45.23	86	451
DAT-FGSM	88.00	45.84	43.19	86	124
Method	Wide ResNet-28-10, batch size 12 × 128				
	TA (%)	RA (%)	AA (%)	C(s)	T(s)
DAT-PGD	89.37	62.06	58.35	302	1020
DAT-FGSM	89.52	61.24	57.65	302	674

Table 4. Effect of gradient quantization on the performance of DAT for various numbers of bits. The training and evaluation settings on (ImageNet, ResNet-50) are consistent with Table 1. The new performance metric 'Data trans. (MB)' represents data transmitted per iteration in the unit MB.

Method	ImageNet, ResNet-50				
	# bits	TA (%)	RA (%)	C (s)	Data trans. (MB)
DAT-PGD	32	63.75	38.45	898	2924
DAT-PGD	16	61.77	38.40	850	1462
DAT-PGD	8	56.53	37.90	592	731
DAT-PGD	8 (2-sided)	53.09	34.59	1091	244
DAT-PGD (HPC)	32	63.43	38.55	15	1074
DAT-FGSM	32	58.02	40.27	859	2924
DAT-FGSM	16	54.71	39.29	849	1462
DAT-FGSM	8	50.11	36.38	594	731
DAT-FGSM	8 (2-sided)	48.27	33.20	1013	244
DAT-FGSM (HPC)	32	57.60	41.70	15	310



(a) Finetuning over CIFAR-10 (b) Finetuning over CIFAR-100

Figure 2. Fine-tuning ResNet-50 (pre-trained on ImageNet) under CIFAR-10 and CIFAR-100. Adversarial training on CIFAR from scratch is also presented. Here DAT-PGD is used for both pre-training and fine-tuning at 6 computing nodes.

Contact: {gaoyuan.zhang, songtao}@ibm.com, {zhan1908}@msu.edu