Adversarial Training for Free!

60212252 컴퓨터공학과 한우혁

목차

Introduction

- Deep learning – excellent performance / robustness to adversarial attacks
- Robustness - detecting and rejecting adversarial examples, or adversarial training
- Cost problem of adversarial training(update network parameters, produce adversarial images) -> 3~30 times longer
- Certified Defenses - demonstrated for small networks, low-res datasets, and relatively small perturbation budgets.
- Adversarial training remains among the most trusted defenses, but it is nearly intractable on large-scale problems.

Introduction - contributions

- A fast adversarial training algorithm for generating robust models with little to no additional cost compared to natural training
- **Idea**: updating both model parameters and image perturbations using a single simultaneous backward pass instead of computing separate gradients for each update step
- A speedup of 3–30× compared to previous methods(K-PGD)
- Application of the algorithm to large-scale ImageNet classification on a single workstation with four P100 GPUs within approximately two days, achieving 40% accuracy against non-targeted PGD attacks
- High competitiveness compared to previous approaches

Non-targeted Adversarial Examples

- **non-targeted** / targeted
- non-targeted attack objective

$$\max_{\delta} l(x+\delta, y, \theta), \text{ subject to } |\delta|_p \leq \epsilon,$$

$$||v||_p = \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}$$

Non-targeted Adversarial Examples

- non-targeted adversarial example

- evaluating the robustness of models & adversarial training

Non-targeted Adversarial Examples

- FGSM(Fast Gradient Sign Method) – Goodfellow

- sign of the gradients to construct an adversarial example in one iteration

$$x_{adv} = x + \epsilon \cdot \text{sign}\big(\nabla_x l(x, y, \theta)\big).$$

Non-targeted Adversarial Examples

- BIM(Basic Iterative Method) – Kurakin
  - iterative version of FGSM

- PGD(Projected Gradient Descent)
  - the most powerful first-order attacks
  - uniform random noise as initialization -> gradient masking
  - K – number of iteration

  -> strength of attacks & computation time for generating adversarial example

  -> complete forward and backward pass is needed to compute the gradient of the loss

- PGD-K attack

Adversarial Training

- producing adversarial examples and injecting them into training data
- robustness - strength of the adversarial example
- 2017 Mardy - propose training on multi-step PGD adversaries -> state-of-the-art robustness levels against `∞ attacks on MNIST, CIFAR-10 dataset

Adversarial Training

- K-PGD adversarial training algorithm

- inner loop(constructs adversarial examples by PGD-K)
  - $\nabla_x l(x_{\text{adv}}, y, \theta)$ - similar cost for updating network parameters
  - compare natural training -> K-PGD K+1 times more compute

- outer loop(using minibatch SGD on the generated example)

“Free” Adversarial Training

- K-PGD generally slow
- 2017 Mardy – Wide Resnet 7PGD training CIFAR-10 4days on Titan X GPU
- So, propose free adversarial training, which has a negligible complexity overhead compared to natural training
  - Reuse backward pass gradient
    -> but performs only one backward pass
  - train on the same minibatch m times in a row

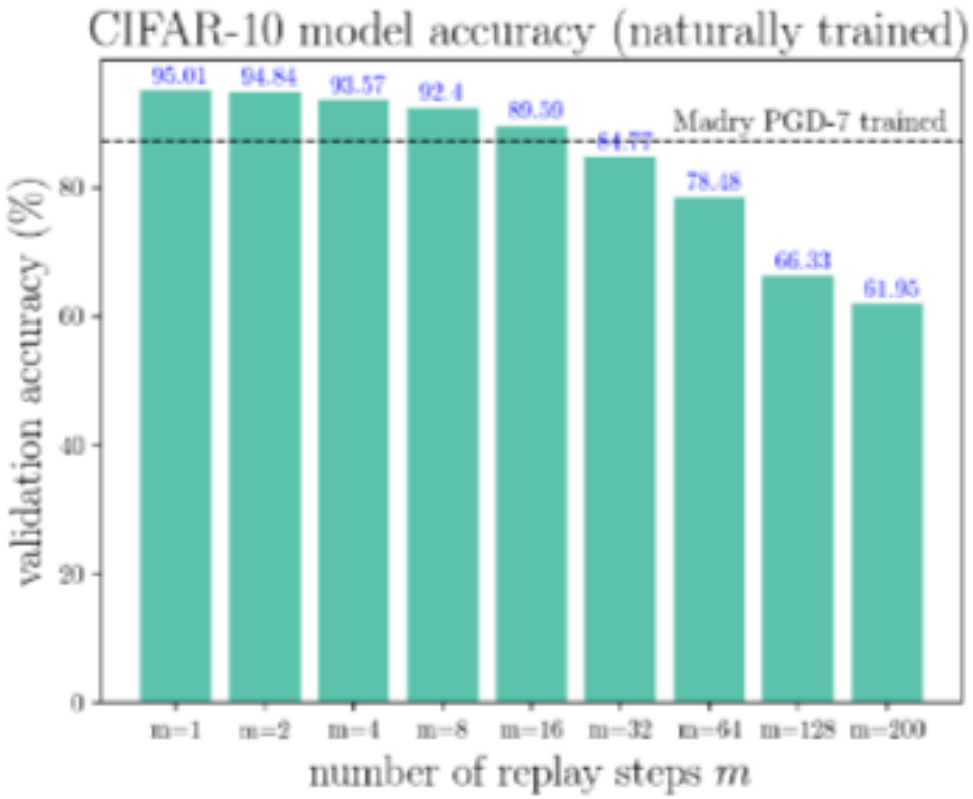
"Free" Adversarial Training

Algorithm 1 "Free" Adversarial Training (Free-m)

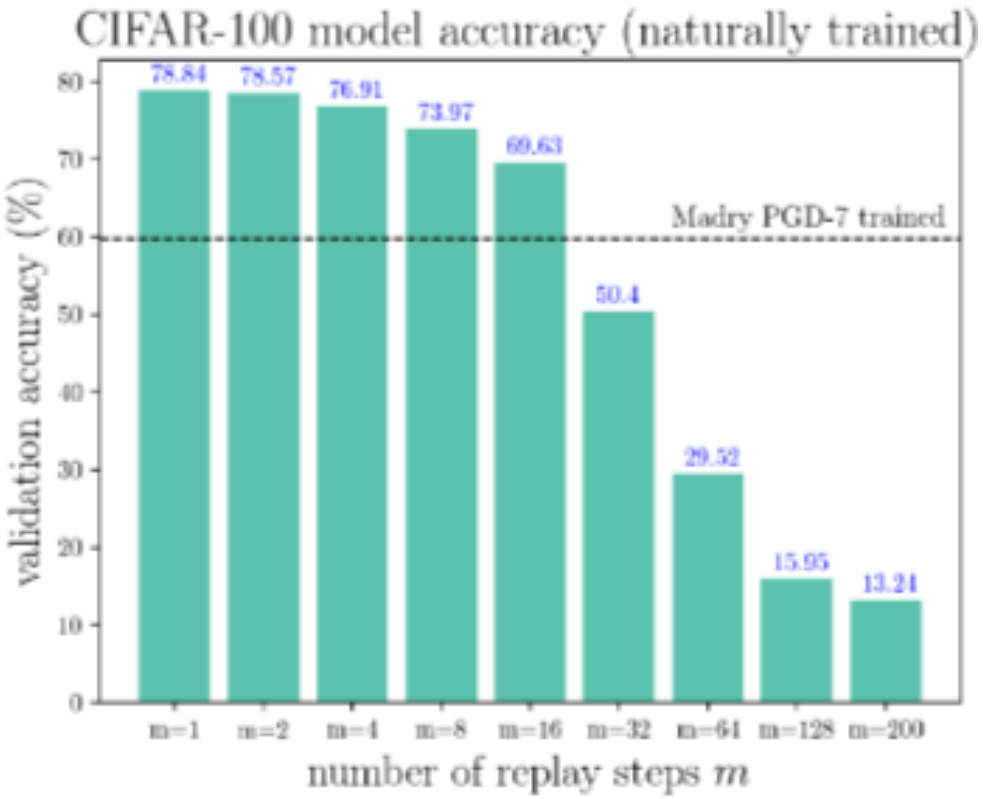
Require: Training samples X, perturbation bound ϵ, learning rate τ, hop steps m

Initialize θ
$\delta \leftarrow 0$
for epoch $= 1 \ldots N_{ep}/m$ **do**
    for minibatch $B \subset X$ **do**
        for i $= 1 \ldots m$ **do**
            Update θ with stochastic gradient descent
                $g_\theta \leftarrow \mathbb{E}_{(x,y)\in B}[\nabla_\theta \, l(x+\delta, y, \theta)]$
                $g_{adv} \leftarrow \nabla_x \, l(x+\delta, y, \theta)]$
                $\theta \leftarrow \theta - \tau g_\theta$
            Use gradients calculated for the minimization step to update δ
                $\delta \leftarrow \delta + \epsilon \cdot \text{sign}(g_{adv})$
                $\delta \leftarrow \text{clip}(\delta, -\epsilon, \epsilon)$
        end for
    end for
end for

“Free” Adversarial Training



(a) CIFAR-10 sensitivity to m



(b) CIFAR-100 sensitivity to m

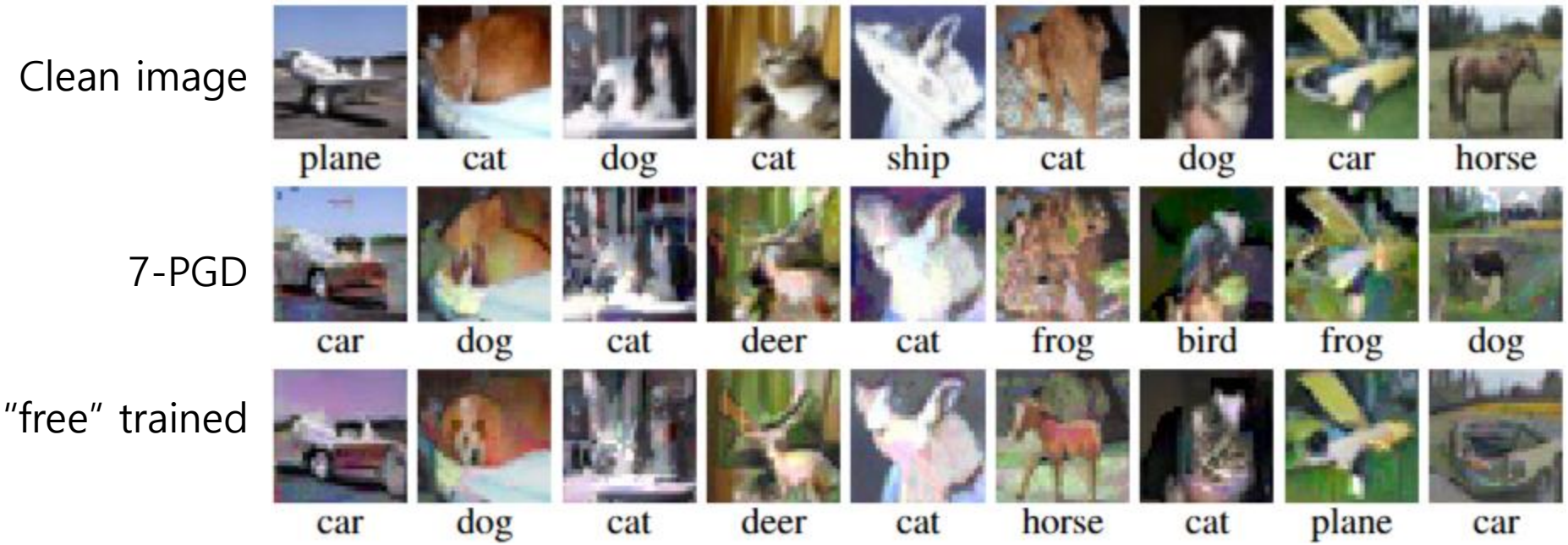
Robust models on CIFAR-10 and 100

Training	Evaluated Against					Train Time (min)
	Nat. Images	PGD-20	PGD-100	CW-100	10 restart PGD-20	
Natural	**95.01%**	0.00%	0.00%	0.00%	0.00%	**780**
Free $m = 2$	91.45%	33.92%	33.20%	34.57%	33.41%	816
Free $m = 4$	87.83%	41.15%	40.35%	41.96%	40.73%	800
Free $m = 8$	85.96%	**46.82%**	**46.19%**	**46.60%**	**46.33%**	785
Free $m = 10$	83.94%	46.31%	45.79%	45.86%	45.94%	785
7-PGD trained	87.25%	45.84%	45.29%	46.52%	45.53%	5418

Training	Evaluated Against			Training Time (minutes)
	Natural Images	PGD-20	PGD-100	
Natural	**78.84%**	0.00%	0.00%	811
Free $m = 2$	69.20%	15.37%	14.86%	816
Free $m = 4$	65.28%	20.64%	20.15%	**767**
Free $m = 6$	64.87%	23.68%	23.18%	791
Free $m = 8$	62.13%	**25.88%**	**25.58%**	780
Free $m = 10$	59.27%	25.15%	24.88%	776
Madry *et al.* (2-PGD trained)	67.94%	17.08%	16.50%	2053
Madry *et al.* (7-PGD trained)	59.87%	22.76%	22.52%	5157

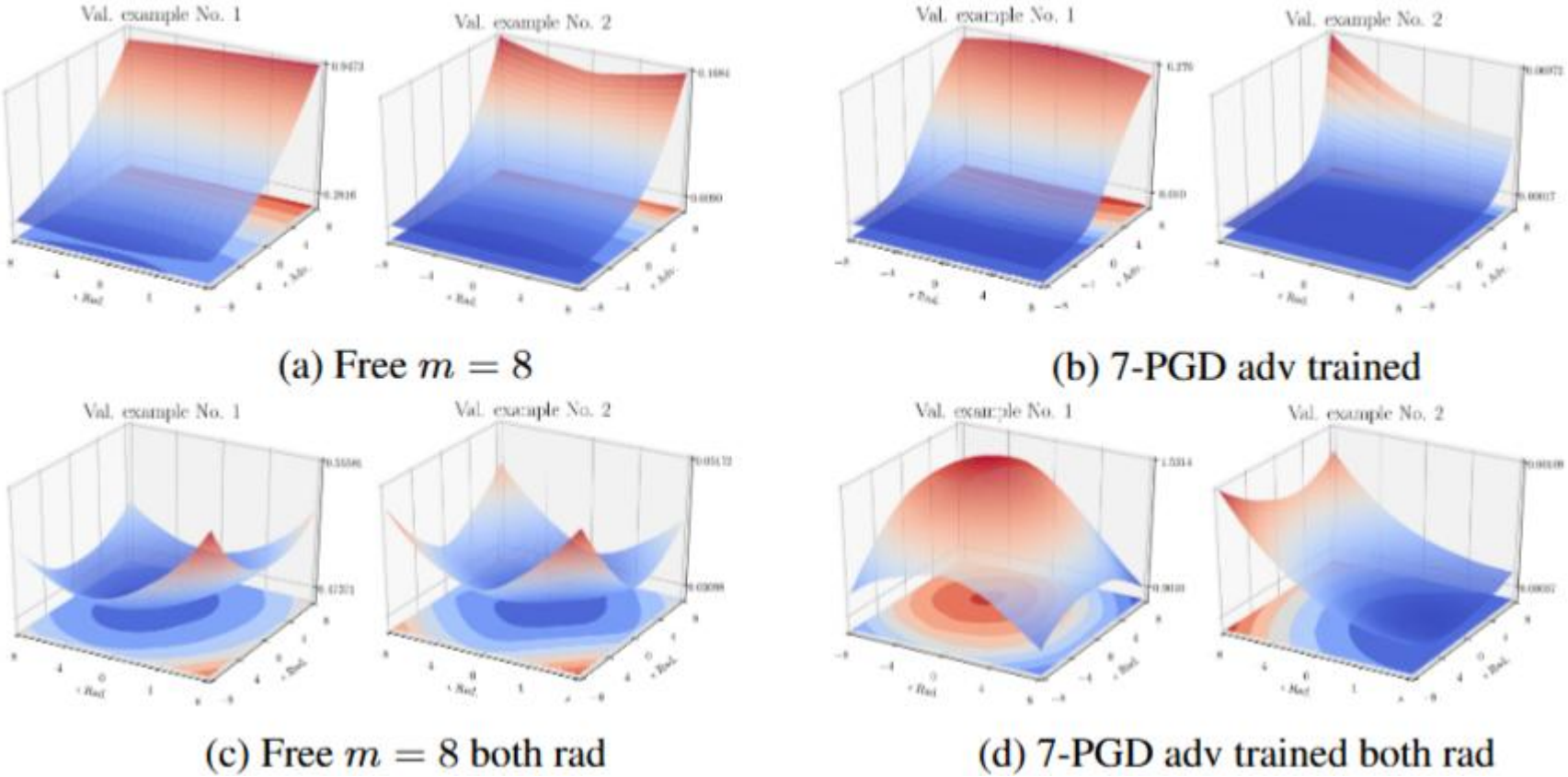
Does “free” training behave like standard adversarial training

- interpretability of gradients and the flattness of loss surface
- Generative behavior for largely perturbed example
- trained to resist L_∞attacks with $\epsilon = 8$



Does “free” training behave like standard adversarial training

- Smooth and flattened loss surface



(a) Free $m = 8$

(b) 7-PGD adv trained

(c) Free $m = 8$ both rad

(d) 7-PGD adv trained both rad

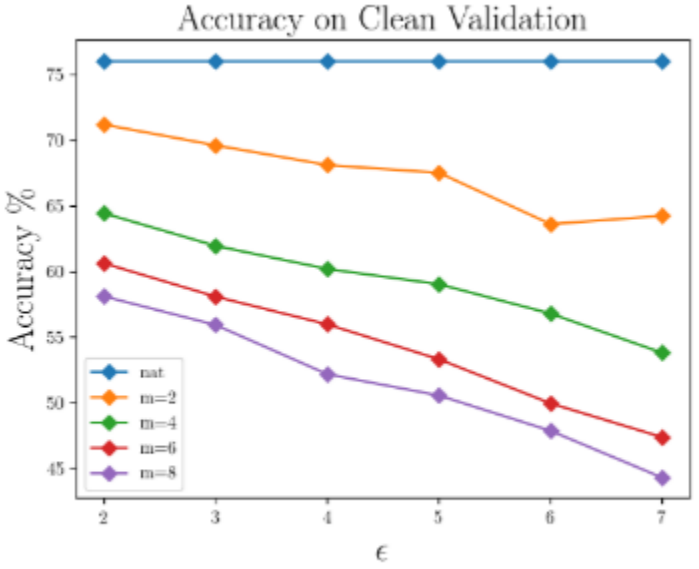
Robust ImageNet classifiers

- large image classification dataset of over 1 million high-res images and 1000 class
  -> high computational cost

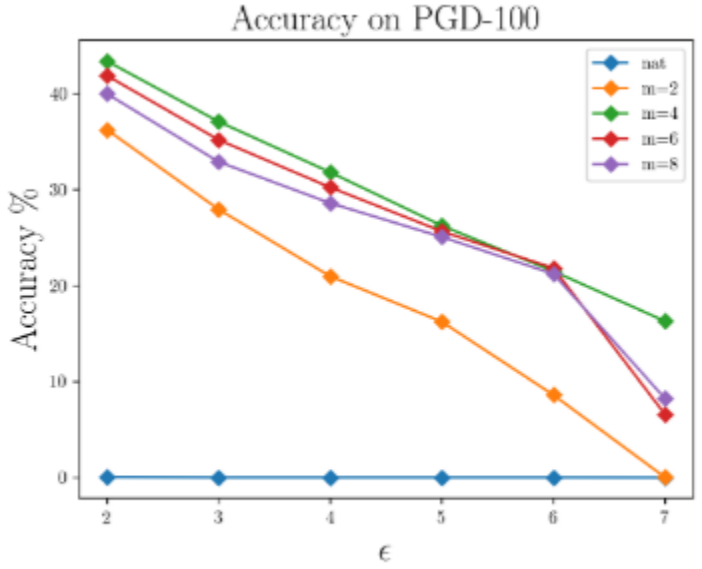
- Algorithm – non targeted adversarial training
  - single workstation with four P100 GPUs, ResNet-50 below 50 hours

Robust ImageNet classifiers

Training	Evaluated Against			
	Natural Images	PGD-10	PGD-50	PGD-100
Natural	**76.038%**	0.166%	0.052%	0.036%
Free $m = 2$	71.210%	37.012%	36.340%	36.250%
Free $m = 4$	64.446%	**43.522%**	**43.392%**	**43.404%**
Free $m = 6$	60.642%	41.996%	41.900%	41.892%
Free $m = 8$	58.116%	40.044%	40.008%	39.996%



(a) Clean



(b) PGD-100

Robust ImageNet classifiers

- Comparison with PGD-trained models

Model & Training	Evaluated Against				Train time (minutes)
	Natural Images	PGD-10	PGD-50	PGD-100	
RN50 – Free $m = 4$	60.206%	32.768%	31.878%	31.816%	**3016**
RN50 – 2-PGD trained	**64.134%**	**37.172%**	**36.352%**	**36.316%**	10,435

Architecture	Evaluated Against			
	Natural Images	PGD-10	PGD-50	PGD-100
ResNet-50	60.206%	32.768%	31.878%	31.816%
ResNet-101	63.340%	35.388%	34.402%	34.328%
ResNet-152	**64.446%**	**36.992%**	**36.044%**	**35.994%**

Conclusion

- Adversarial training is a well-studied method that boosts the robustness and interpretability of neural networks.
  -> but, high cost

- So, present “FREE” training with cost nearly equal natural
  -> need modest compute resources