Adversarial Training for FREE!

Ali Shafahi[1], Mahyar Najibi[1], Amin Ghiasi[1], Zheng Xu[1], John Dickerson[1], Christoph Studer[2], Larry Davis[1], Gavin Taylor[3] , Tom Goldstein[1]

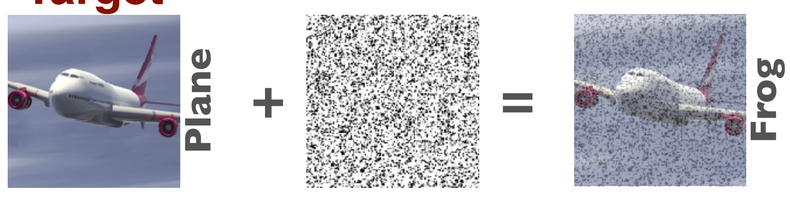
[1]University of Maryland, [2]Cornell University, and [3]US Naval Academy

Overview

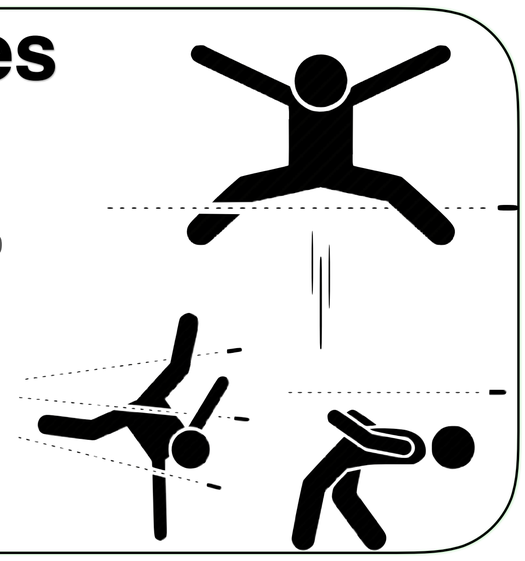
Adversarial training (using PGD attacks) is one of the best defenses and wins nearly all defense competitions. But it's slow, taking 5-50X longer than regular training. This makes it nearly intractable for large problems like ImageNet.

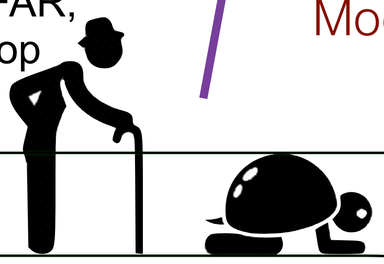
We present a method that adversarially trains with no added cost beyond regular training. Our "free" method gets comparable results to adversarial training on CIFAR, and can adversarially train ImageNet on a desktop computer in just a day!

Adversarial examples



Modify target images at ***test*** time





Adversarial Training

Algorithm: K-PGD

Input: Training samples $\boldsymbol{X}$, perturbation bound ϵ, step-size ϵ_s, PGD iterations $\boldsymbol{K}$, learning-rate τ

for epoch = 1 … N **do**
  for minibatch $B \subset X$ **do**
    Build x_{adv} forall $x \in B$ with PGD:
    $x_{adv} \leftarrow x + r; \quad r \leftarrow U(-\epsilon, \epsilon)$
    for k = 1 … K **do**
      $g_{adv} \leftarrow \nabla_x l(x_{adv}, y, \theta)$
      $x_{adv} \leftarrow \text{clip}(x_{adv} + \epsilon_s \cdot \text{sign}(g_{adv}), x - \epsilon, x + \epsilon)$
    end for
    update θ with SGD:
    $g_\theta \leftarrow \mathbb{E}_{(x,y)\in B}[\nabla_\theta l(x_{adv}, y, \theta)]$
    $\theta \leftarrow \theta - \tau g_\theta$
  end for
end for

Backprop K times

Backprop **1** time

Adversarial Training for Free!

Algorithm: Free-m

Input: Training samples $\boldsymbol{X}$, perturbation bound ϵ, learning-rate τ, *replay* $\boldsymbol{m}$

$\delta \leftarrow 0$
for epoch = 1 … N/m **do**
  for minibatch $B \subset X$ **do**
    for i = 1 … m **do**
      Update θ with SGD:
      $g_\theta \leftarrow \mathbb{E}_{(x,y)\in B}[\nabla_\theta l(x + \delta, y, \theta)]$
      $g_{adv} \leftarrow \nabla_x l(x + \delta, y, \theta)$
      $\theta \leftarrow \theta - \tau g_\theta$
      Use grads calculated at min step for updating δ
      $\delta \leftarrow \text{clip}(\delta + \epsilon \cdot \text{sign}(g_{adv}), -\epsilon, +\epsilon)$
    end for
  end for
end for

Backprop just 1 time!

Results: Free vs K-PGD

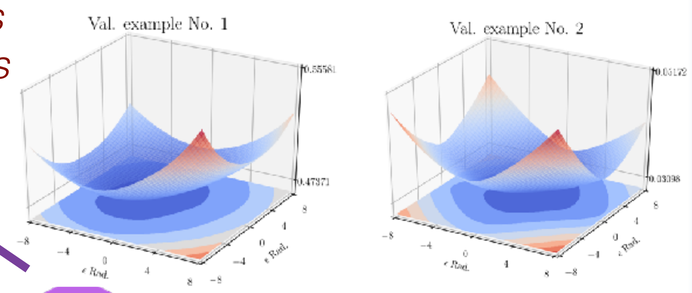
Model & Training	Evaluated Against				Train time (minutes)
	Natural Images	PGD-10	PGD-50	PGD-100	
ResNet-50 – Free $m = 4$	60.206%	32.768%	31.878%	31.816%	**3,016**
ResNet-101 – Free $m = 4$	63.340%	35.388%	34.402%	34.328%	5,122
ResNet-152 – Free $m = 4$	**64.446%**	36.992%	36.044%	35.994%	7,526
ResNet-50 – 2-PGD trained	64.134%	**37.172%**	**36.352%**	**36.316%**	10,435

ImageNet ($\epsilon = 4$)

ImageNet

Free-m has interpretable grads

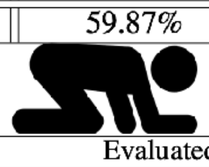
Free-m has smooth loss surface



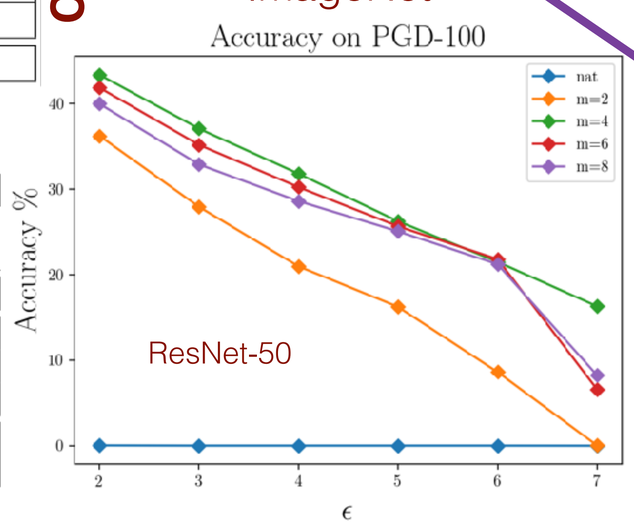
Training	Evaluated Against			Training Time (minutes)
	Natural Images	PGD-20	PGD-100	
Natural	**78.84%**	0.00%	0.00%	811
Free $m = 4$	65.28%	20.64%	20.15%	**767**
Free $m = 6$	64.87%	23.68%	23.18%	791
Free $m = 8$	62.13%	**25.88%**	**25.58%**	780
Free $m = 10$	59.27%	25.15%	24.88%	776
Madry *et al.* (7-PGD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

CIFAR-100 ($\epsilon = 8$)

CIFAR-10 ($\epsilon = 8$)



Training	Evaluated Against					Training Time (minutes)
	Natural Images	PGD-20	PGD-100	CW-100	10 restart PGD-20	
Natural	**95.01%**	0.00%	0.00%	0.00%	0.00%	**780**
Free $m = 4$	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
Madry *et al.* (7-PGD trained)	87.25%	45.84%	45.29%	46.52%	45.53%	5418



Pytorch

ImageNet

Tensorflow

CIFAR