Improving the affordability of robustness training for DNNs

Sidharth Gupta [1] Parijat Dube [2] Ashish Verma [2]

[1]University of Illinois at Urbana-Champaign [2]IBM Research

CVPR 2020 Workshop on Adversarial Machine Learning in Computer Vision

ILLINOIS
Coordinated Science Lab
COLLEGE OF ENGINEERING

Deep neural networks are sensitive

- Deep neural network classifiers can give almost **0% accuracy** when the test data is perturbed by an imperceptible amount

Deep neural networks are sensitive

- Deep neural network classifiers can give almost **0% accuracy** when the test data is perturbed by an imperceptible amount

$\boldsymbol{x}$

$\boldsymbol{x}_{adv}$

Deep neural networks are sensitive

- Deep neural network classifiers can give almost **0% accuracy** when the test data is perturbed by an imperceptible amount

$\boldsymbol{x}$

$\boldsymbol{x}_{adv}$

- Clean data, $\boldsymbol{x}$, is correctly classified
- Adversarial samples, $\boldsymbol{x}_{adv}$, are incorrectly classified
- $\boldsymbol{x}_{adv}$ are ℓ_∞ perturbed samples: $\|\boldsymbol{x} - \boldsymbol{x}_{adv}\|_\infty \leq \frac{8}{255} \approx 0.03$

Building robust models

- Regular adversarial training (RAT) is popular:

  $\min_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta}); \; \rho(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_\infty \leq \epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), y)\right]$

Building robust models

- Regular adversarial training (RAT) is popular:
  $\min_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta})$; $\rho(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_\infty \leq \epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), y)\right]$
  1. Replace data with adversarial counterparts

Building robust models

- Regular adversarial training (RAT) is popular:

  $\min_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta});\ \rho(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_\infty \le \epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), y)\right]$

  1. Replace data with adversarial counterparts
  2. Update model with adversarial counterparts

Building robust models

- Regular adversarial training (RAT) is popular:

  $\min_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta}); \; \rho(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_\infty \leq \epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), y) \right]$

  1. Replace data with adversarial counterparts
  2. Update model with adversarial counterparts

- Projected Gradient Descent (PGD) for maximization

  $$\boldsymbol{x}^{t+1} = \Pi_{\epsilon-\text{ball}} \left(\boldsymbol{x}^t + \alpha \, \text{sign}(\nabla_{\boldsymbol{x}^t} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}^t), y)) \right)$$

  Projects to an ϵ-ball around $\boldsymbol{x}$ after every iteration

Building robust models is expensive

- Regular adversarial training (RAT) is popular:

  $\min_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta});\ \rho(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_\infty \leq \epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), y)\right]$

  1. Replace data with adversarial counterparts
  2. Update model with adversarial counterparts

- Projected Gradient Descent (PGD) for maximization

  $$\boldsymbol{x}^{t+1} = \Pi_{\epsilon-\text{ball}}\left(\boldsymbol{x}^t + \alpha\, \text{sign}(\nabla_{\boldsymbol{x}^t} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}^t), y))\right)$$

  Projects to an ϵ-ball around $\boldsymbol{x}$ after every iteration

Each PGD iteration requires a forward and backward pass of the model which is **computationally expensive**

Building robust models is expensive

- Regular adversarial training (RAT) is popular:

  $\min_{\boldsymbol{\theta}} \rho(\boldsymbol{\theta})$; $\rho(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\tilde{\boldsymbol{x}}-\boldsymbol{x}\|_\infty \le \epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), y)\right]$

  1. Replace data with adversarial counterparts
  2. Update model with adversarial counterparts

- Projected Gradient Descent (PGD) for maximization

  $$\boldsymbol{x}^{t+1} = \Pi_{\epsilon-\text{ball}}\left(\boldsymbol{x}^t + \alpha\,\text{sign}(\nabla_{\boldsymbol{x}^t}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}^t), y))\right)$$

  Projects to an ϵ-ball around $\boldsymbol{x}$ after every iteration

Each PGD iteration requires a forward and backward pass of the model which is **computationally expensive**

Model architecture	Natural training	Regular adversarial training
ResNet-50	1.1 hours	6.8 hours
WideResNet-28x10	2.2 hours	14.7 hours

CIFAR-10 for 155 epochs: 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$

Usefulness of initial adversaries in RAT

- Generated adversarial samples depend on the evolving model parameters which are randomly initialized

Usefulness of initial adversaries in RAT

- Generated adversarial samples depend on the evolving model parameters which are randomly initialized
- Final model parameters are very different $\Rightarrow$ Initial training samples are very different from adversaries that the final model will face

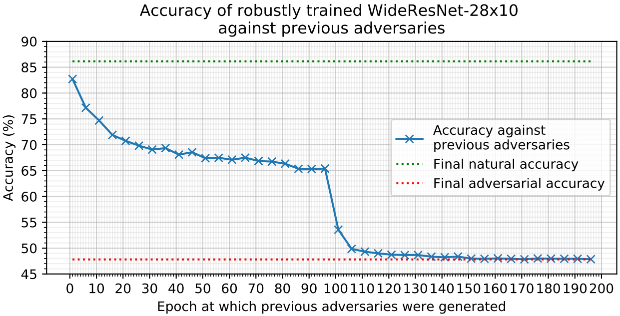
Usefulness of initial adversaries in RAT

- Generated adversarial samples depend on the evolving model parameters which are randomly initialized
- Final model parameters are very different $\Rightarrow$ Initial training samples are very different from adversaries that the final model will face
- Generating initial samples adds computational overhead

Usefulness of initial adversaries in RAT

- Generated adversarial samples depend on the evolving model parameters which are randomly initialized
- Final model parameters are very different $\Rightarrow$ Initial training samples are very different from adversaries that the final model will face
- Generating initial samples adds computational overhead

- Perform RAT (learning rate drop after epoch 100)
- Test final model with adversaries generated from model parameters at previous epochs
- CIFAR-10 with 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$



Delayed adversarial training (DAT)

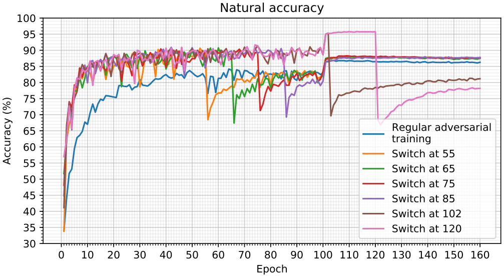
- Adversarial samples are computationally expensive to generate and not useful in the initial training phase

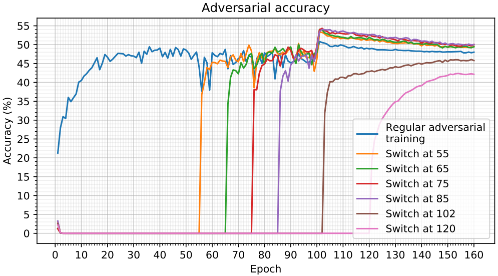
Delayed adversarial training (DAT)

- Adversarial samples are computationally expensive to generate and not useful in the initial training phase
- Use free **natural** training samples initially until model stabilizes

Delayed adversarial training (DAT)

- Adversarial samples are computationally expensive to generate and not useful in the initial training phase
- Use free **natural** training samples initially until model stabilizes

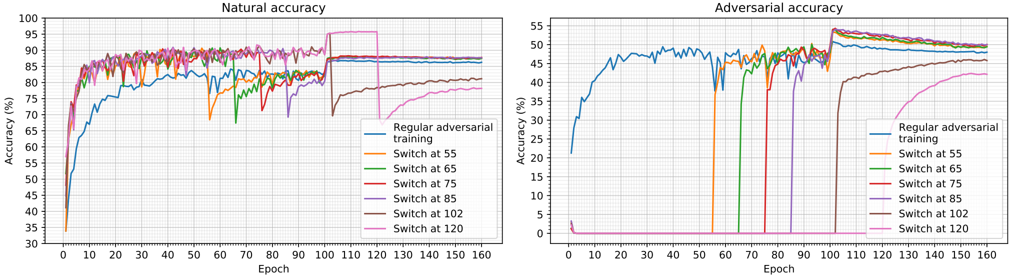




CIFAR-10; WideResNet-28x10; 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$; learning rate drop after epoch 100

Delayed adversarial training (DAT)

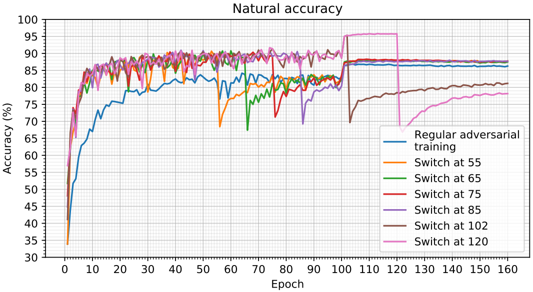
- Adversarial samples are computationally expensive to generate and not useful in the initial training phase
- Use free **natural** training samples initially until model stabilizes

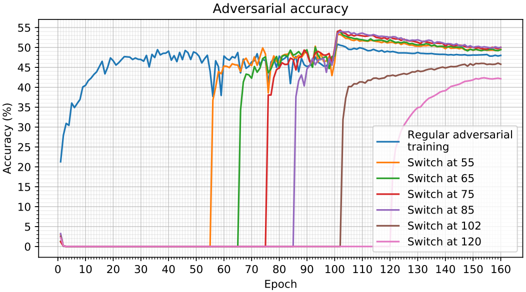


CIFAR-10; WideResNet-28x10; 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$; learning rate drop after epoch 100

- Automated switching: Training loss on natural samples stabilizes before first learning rate drop $\Rightarrow$ Switch from natural to adversarial samples when loss stabilizes before first learning rate drop

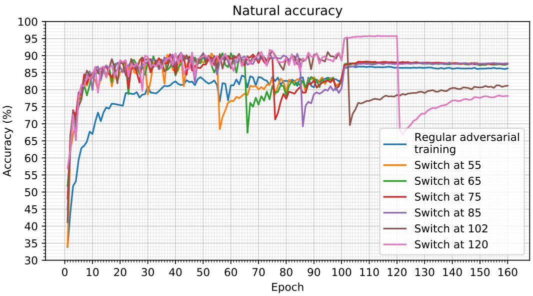
Delayed adversarial training (DAT) helps generalization

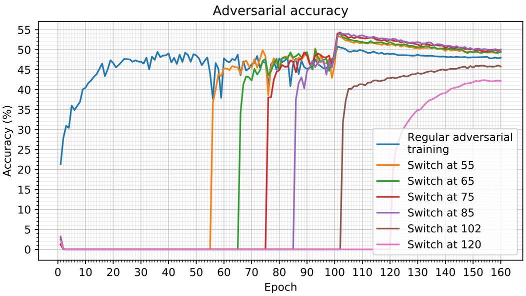




CIFAR-10; WideResNet-28x10; 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$; learning rate drop after epoch 100

Delayed adversarial training (DAT) helps generalization





CIFAR-10; WideResNet-28x10; 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$; learning rate drop after epoch 100

- DAT trained models show higher test accuracy
- Models are not overfitting to adversarial samples of little relevance in the initial phase of training
- Along with the higher accuracy, we observe a higher training loss with DAT which indicates better generalization

Training times and test accuracy against adversaries of training strength

		CIFAR-10: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
WideResNet-28x10	*RAT*				
	DAT				
	RAT early stop				
	DAT early stop				
ResNet-18	*RAT*				
	DAT				
	RAT early stop				
	DAT early stop				
		CIFAR-100: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
ResNet-50	*RAT*				
	DAT				
		MNIST: 40-step PGD, $\epsilon=0.3$, $\alpha=0.01$ on less powerful system			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
Two-layer CNN	*RAT*				
	DAT				

Training times and test accuracy against adversaries of training strength

		CIFAR-10: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
WideResNet-28x10	*RAT*	14.7 hours	46.9%	48.5%	86.8%
	DAT	7.8 hours		49.7%	87.9%
	RAT early stop	10.9 hours	62.4%	49.2%	87.1%
	DAT early stop	4.1 hours		53.6%	87.9%
ResNet-18	*RAT*				
	DAT				
	RAT early stop				
	DAT early stop				
		CIFAR-100: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
ResNet-50	*RAT*				
	DAT				
		MNIST: 40-step PGD, $\epsilon=0.3$, $\alpha=0.01$ on less powerful system			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
Two-layer CNN	*RAT*				
	DAT				

Training times and test accuracy against adversaries of training strength

		Training time	Time saved	Adversarial accuracy	Natural accuracy
		CIFAR-10: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
WideResNet-28x10	*RAT*	14.7 hours	**46.9%**	48.5%	86.8%
	DAT	7.8 hours		49.7%	87.9%
	RAT early stop	10.9 hours	**62.4%**	49.2%	87.1%
	DAT early stop	4.1 hours		**53.6%**	**87.9**%
ResNet-18	*RAT*	2.5 hours	**36.0%**	37.0%	**73.8%**
	DAT	1.6 hours		40.4%	72.8%
	RAT early stop	1.9 hours	**52.6%**	40.6%	71.0%
	DAT early stop	0.9 hours		**41.1%**	69.9%
		CIFAR-100: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
ResNet-50	*RAT*				
	DAT				
		MNIST: 40-step PGD, $\epsilon=0.3$, $\alpha=0.01$ on less powerful system			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
Two-layer CNN	*RAT*				
	DAT				

Training times and test accuracy against adversaries of training strength

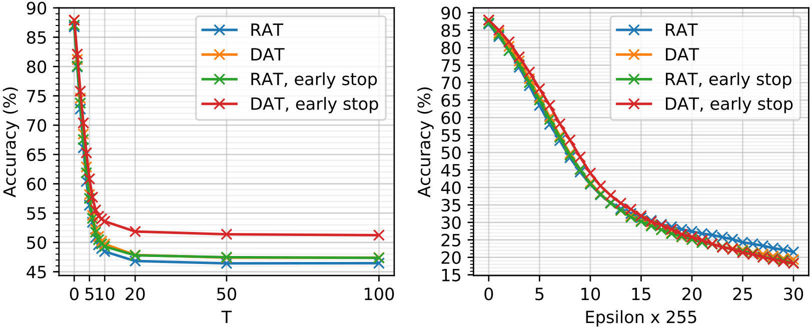
		CIFAR-10: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
WideResNet-28x10	*RAT*	14.7 hours	**46.9%**	48.5%	86.8%
	DAT	7.8 hours		49.7%	87.9%
	RAT early stop	10.9 hours	**62.4%**	49.2%	87.1%
	DAT early stop	4.1 hours		**53.6%**	**87.9**%
ResNet-18	*RAT*	2.5 hours	**36.0%**	37.0%	**73.8%**
	DAT	1.6 hours		40.4%	72.8%
	RAT early stop	1.9 hours	**52.6%**	40.6%	71.0%
	DAT early stop	0.9 hours		**41.1%**	69.9%
		CIFAR-100: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
ResNet-50	*RAT*	6.9 hours	**42.0%**	15.2%	44.2%
	DAT	4.0 hours		15.2%	**46.6%**
		MNIST: 40-step PGD, $\epsilon=0.3$, $\alpha=0.01$ on less powerful system			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
Two-layer CNN	*RAT*				
	DAT				

Training times and test accuracy against adversaries of training strength

		CIFAR-10: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
WideResNet-28x10	*RAT*	14.7 hours	46.9%	48.5%	86.8%
	DAT	7.8 hours		49.7%	87.9%
	RAT early stop	10.9 hours	62.4%	49.2%	87.1%
	DAT early stop	4.1 hours		53.6%	87.9%
ResNet-18	*RAT*	2.5 hours	36.0%	37.0%	73.8%
	DAT	1.6 hours		40.4%	72.8%
	RAT early stop	1.9 hours	52.6%	40.6%	71.0%
	DAT early stop	0.9 hours		41.1%	69.9%
		CIFAR-100: 10-step PGD, $\epsilon=8/255$, $\alpha=2/255$			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
ResNet-50	*RAT*	6.9 hours	42.0%	15.2%	44.2%
	DAT	4.0 hours		15.2%	46.6%
		MNIST: 40-step PGD, $\epsilon=0.3$, $\alpha=0.01$ on less powerful system			
		Training time	*Time saved*	*Adversarial accuracy*	*Natural accuracy*
Two-layer CNN	*RAT*	2.2 hours	13.6%	91.4%	98.2%
	DAT	1.9 hours		91.9%	98.2%

Generalization to attacks of different strength to training strength

1. Keep ϵ-ball size fixed and vary PGD steps (T)
2. Keep T fixed and vary ϵ



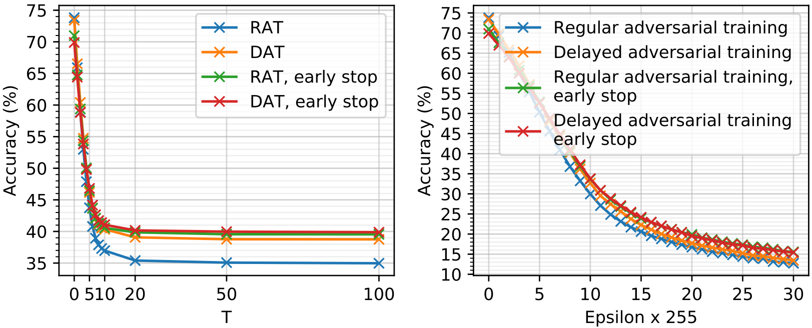
CIFAR-10

WideResNet-28x10

Training adversary strength: 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$

Generalization to attacks of different strength to training strength

1. Keep ϵ-ball size fixed and vary PGD steps (T)
2. Keep T fixed and vary ϵ



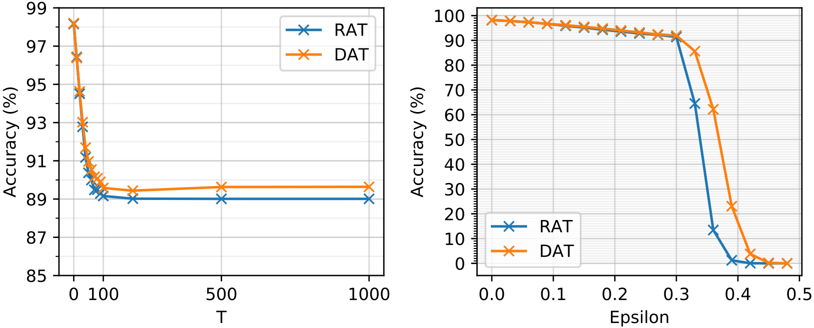
CIFAR-10

ResNet-18

Training adversary strength: 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$

Generalization to attacks of different strength to training strength

1. Keep ϵ-ball size fixed and vary PGD steps (T)
2. Keep T fixed and vary ϵ



MNIST

Two-layer CNN

Training adversary strength: 40-step PGD, $\epsilon = 0.3$, $\alpha = 0.01$

Black-box attack performance

- Test against attacks from independently trained copies of the network
- CIFAR-10; WideResNet-28x10; 10-step PGD, $\epsilon = 8/255$, $\alpha = 2/255$
- Independent copies trained with 1) Natural training; 2) Regular adversarial training (RAT); 3) Delayed adversarial training (DAT)

	White-box	**Independent natural**	**Independent RAT**	**Independent DAT**
Accuracy	49.7%	86.7%	69.4%	65.7%

Summary

- There is an initial phase of adversarial training where adversarial samples are of little relevance
- These samples are expensive to generate
- Delayed adversarial training (DAT) uses natural samples in the initial phase to save time
- DAT achieves comparable accuracy

Check out our full paper for more details and more experimental results