Defense on Image Recognition

Todo List

- Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In 2019 IEEE Winter Conference on Applications ofComputer Vision (WACV), pp. 1300–1307. IEEE, 2019.
- Jamie Hayes. On visible adversarial perturbations & digital watermarking. In Proceedings ofthe IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1597–1604, 2018.
- Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420, 2018.
- Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. arXiv preprint arXiv:1810.12715, 2018.

- Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In International Conference on Machine Learning, pp. 3575–3583, 2018.
- Alexander Levine and Soheil Feizi. Robustness certificates for sparse adversarial attacks by randomized ablation. arXiv preprint arXiv:1911.09272, 2019.
- Gotta Catch'Em All: Using Honeypots to Catch Adversarial Attacks on Neural Networks
- Certified adversarial robustness via randomized smoothing

Towards Deep Learning Models Resistant to Adversarial Attacks

> 由于时间原因，该文章的笔记借鉴自 "前人分享"（链接见下）。

Contribution

1. Madry.
2. 建模了对抗训练过程；
3. 使用 PGD生成的对抗样本 来做**对抗训练**；

Notes

1. ☆ 问题建模，**从优化的角度来看模型鲁棒性问题。**深度学习中，我们经常根据下面这个目标来训练我们的网络：即我们希望我们训练得到的模型在训练样本**上**的经验损失能够达到最小。

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[L(\theta, x, y)\right]$$

   但是这样的训练目标，使得模型容易受到对抗样本的攻击。故作者将对抗样本的攻击防御问题总结为以下公式，该问题原文中作者称为 **鞍点问题（saddle point problem）**，即我们希望我们训练得到的模型在训练样本**周围**的经验损失能够达到最小。

$$\min_{\theta} \rho(\theta), \quad \text{where} \quad \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y)\right]$$

   建模完问题以后，那么以前的对抗样本领域的工作就可以进行简单地分类：（稍微有点绕）

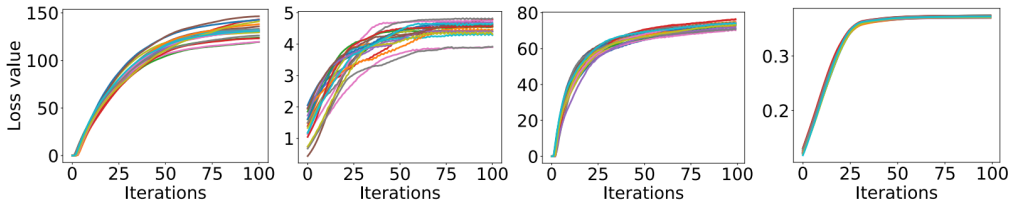
   - 提出一个好的对抗攻击算法，来寻找使得（内层）经验损失最大化的扰动；
   - 提出一个鲁棒性好的模型，来使得（外层）最小化（内层的最大的）经验损失；

2. 文章中作者采用投影梯度下降算法（PGD）来生成对抗样本：（作者的代码中，在生成对抗样本之前，会添加均匀分布的随机噪声，然后再生成对抗样本）

$$x^{t+1} = \Pi_{x+\mathcal{S}}\left(x^t + \alpha\,\mathrm{sgn}(\nabla_x L(\theta, x, y))\right)$$

3. 实验发现：

   (1) Loss 下降趋势和对抗样本算法迭代轮数的关系：无论是原始模型还是使用对抗训练得到的模型，两者使用 PGD算法 生成对抗样本时，随着迭代轮数的上升，样本的 loss 都会上升，且到最后趋于收敛；



(a) MNIST Standard training (b) MNIST Adversarial training (c) CIFAR10 Natural training (d) CIFAR10 Adversarial training

Figure 1: Cross-entropy loss values while creating an adversarial example from the MNIST and CIFAR10 evaluation datasets. The plots show how the loss evolves during 20 runs of projected gradient descent (PGD). Each run starts at a uniformly random point in the ℓ_∞-ball around the same natural example (additional plots for different examples appear in Figure 11). The adversarial loss plateaus after a small number of iterations. The optimization trajectories and final loss values are also fairly clustered, especially on CIFAR10. Moreover, the final loss values on adversarially trained networks are significantly smaller than on their standard counterparts.

(2) Loss 分布的差异：于原始模型相比，在对抗训练得到的模型上生成对抗样本，得到的loss更小，更集中且没有异常值；

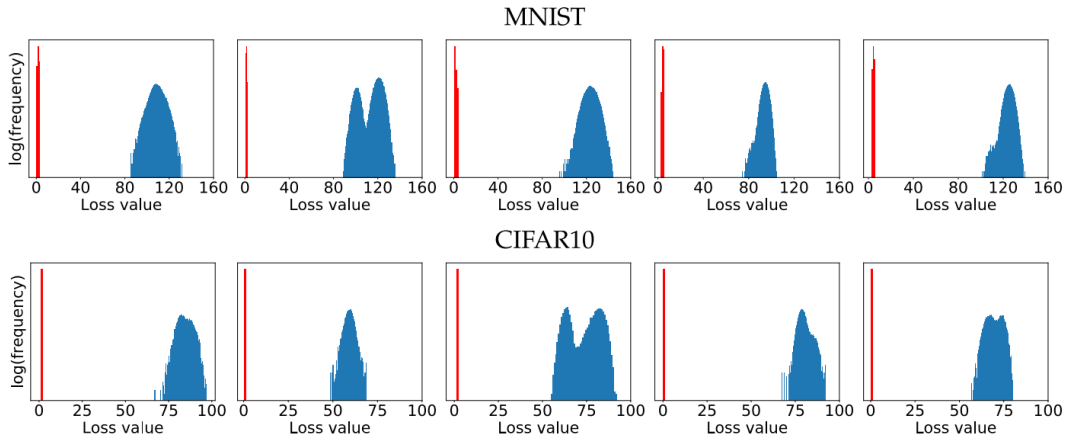
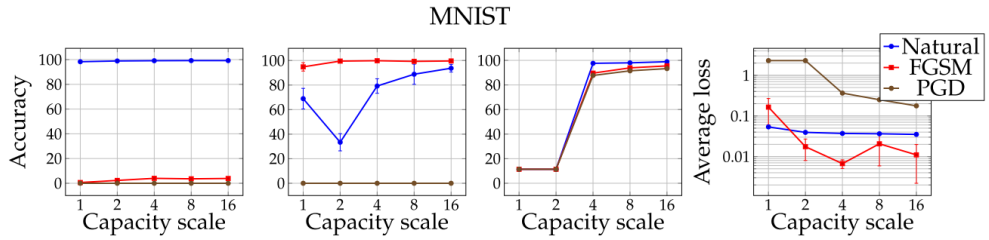


Figure 2: Values of the local maxima given by the cross-entropy loss for five examples from the MNIST and CIFAR10 evaluation datasets. For each example, we start projected gradient descent (PGD) from 10^5 uniformly random points in the ℓ_∞-ball around the example and iterate PGD until the loss plateaus. The blue histogram corresponds to the loss on a standard network, while the red histogram corresponds to the adversarially trained counterpart. The loss is significantly smaller for the adversarially trained networks, and the final loss values are very concentrated without any outliers.

(3) 鲁棒性与模型规模的关系：相对而言，模型越复杂，鲁棒性也越好。同时，经过对抗训练的模型，在原始任务上会有一定的损失，是因为出现了如"过拟合"的现象，使得模型在测试集上面的效果并不好；



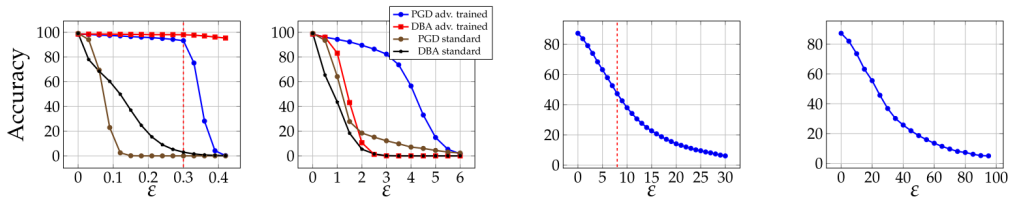
CIFAR10

	Simple	Wide	Simple	Wide	Simple	Wide	Simple	Wide
Natural	92.7%	95.2%	87.4%	90.3%	79.4%	87.3%	0.00357	0.00371
FGSM	27.5%	32.7%	90.9%	95.1%	51.7%	56.1%	0.0115	0.00557
PGD	0.8%	3.5%	0.0%	0.0%	43.7%	45.8%	1.11	0.0218

(a) Standard training (b) FGSM training (c) PGD training (d) Training Loss

Figure 4: The effect of network capacity on the performance of the network. We trained MNIST and CIFAR10 networks of varying capacity on: (a) natural examples, (b) with FGSM-made adversarial examples, (c) with PGD-made adversarial examples. In the first three plots/tables of each dataset, we show how the standard and adversarial accuracy changes with respect to capacity for each training regime. In the final plot/table, we show the value of the cross-entropy loss on the adversarial examples the networks were trained on. This corresponds to the value of our saddle point formulation (2.1) for different sets of allowed perturbations.

(4) 范数限制的影响：l_∞ 范数比 l_2 范数成功的扰动量要小；（这个对比合理吗？）



(a) MNIST, ℓ_∞-norm (b) MNIST, ℓ_2-norm (c) CIFAR10, ℓ_∞-norm (d) CIFAR10, ℓ_2-norm

Figure 6: Performance of our adversarially trained networks against PGD adversaries of different strength. The MNIST and CIFAR10 networks were trained against $\varepsilon = 0.3$ and $\varepsilon = 8$ PGD ℓ_∞ adversaries respectively (the training ε is denoted with a red dashed lines in the ℓ_∞ plots). In the case of the MNIST adversarially trained networks, we also evaluate the performance of the Decision Boundary Attack (DBA) [4] with 2000 steps and PGD on standard and adversarially trained models. We observe that for ε less or equal to the value used during training, the performance is equal or better. For MNIST there is a sharp drop shortly after. Moreover, we observe that the performance of PGD on the MNIST ℓ_2-trained networks is poor and significantly overestimates the robustness of the model. This is potentially due to the threshold filters learned by the model masking the loss gradients (the decision-based attack does not utilize gradients).

Links

- 论文链接：Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks[J]. arXiv preprint arXiv:1706.06083, 2017.
- 论文代码 - mnist：https://github.com/MadryLab/mnist_challenge
- 论文代码 - cifar10：https://github.com/MadryLab/cifar10_challenge

> 从代码上来看，作者提供的模型的输入是 $32 * 32 * 3$ 大小的，这样经过压缩的输入维度，是否导致了对抗样本算法难以实现呢？或者说生成对抗样本的过程能否利用一下这个特点？

- 论文模型：https://github.com/MadryLab/cifar10_challenge
- 前人分享：https://zhuanlan.zhihu.com/p/45684812

Theoretically Principled Trade-off between Robustness and Accuracy

Contribution

1. Trades.
2. 从理论上证明了 成功率和鲁棒性 对于分类问题来说是一个权衡利弊的问题；（虽然我并不太关注这个证明）
3. 在理论的基础上，提出了新的对抗训练的损失函数；

Notes

1. 文章思想：（证明部分直接忽略不看）

   (1) 文章整体的思想如下图，即为在保证模型分类准确的前提下，希望模型的边界能够离这些真实的样本远一些：

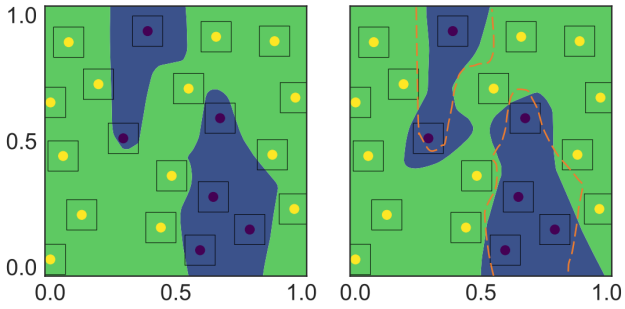
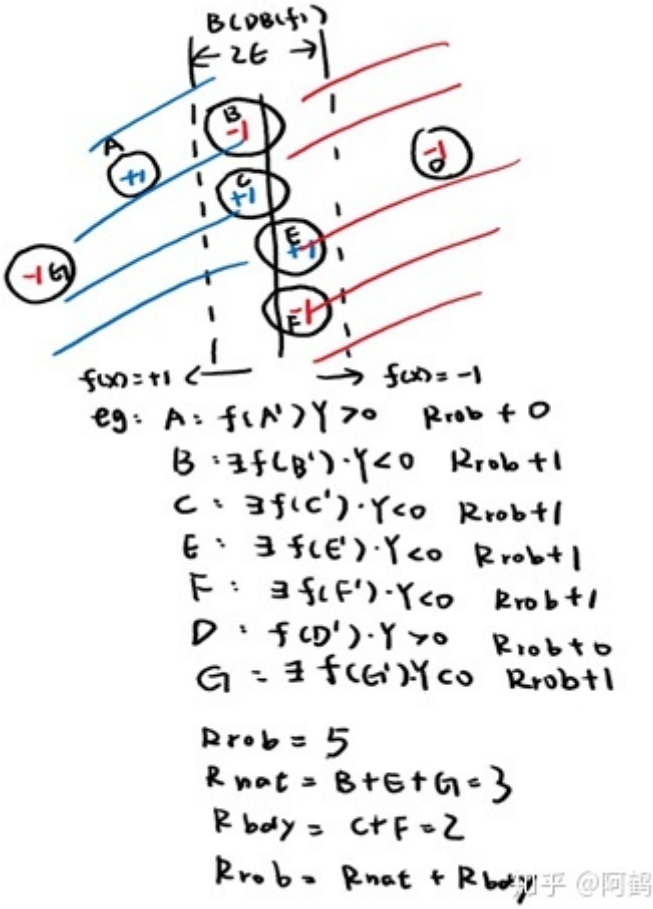


Figure 1: **Left figure:** decision boundary learned by natural training method. **Right figure:** decision boundary learned by our adversarial training method, where the orange dotted line represents the decision boundary in the left figure. It shows that both methods achieve zero natural training error, while our adversarial training method achieves better robust training error than the natural training method.

   (2) 文章中提到了 鲁棒误差（Robust Error）、自然误差（Natural Error）和 边界误差（Boundary Error）：

   - 鲁棒误差：
     $\mathcal{R}_{\mathrm{rob}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}\{\exists \boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon) \text{ s.t. } f(\boldsymbol{X}')Y \leq 0\}$
   - 自然误差：$\mathcal{R}_{\mathrm{nat}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}\{f(\boldsymbol{X})Y \leq 0\}$
   - 边界误差：
     $\mathcal{R}_{\mathrm{bdy}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbf{1}\{\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon), f(\boldsymbol{X})Y > 0\}$
   - 三者关系：$\mathcal{R}_{\mathrm{rob}}(f) = \mathcal{R}_{\mathrm{nat}}(f) + \mathcal{R}_{\mathrm{bdy}}(f)$

     理解起来有点困难，借鉴前人分享的图（链接见下）：



> 画个示意图：假设中间是决策边界，A点~G点是x，外面的圈圈是给定的扰动，虚线是决策边界的边界。

2. 最优化问题：

$$\min_f \mathbb{E}\Big\{ \underbrace{\phi(f(\boldsymbol{X})Y)}_{\text{for accuracy}} + \underbrace{\max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X})f(\boldsymbol{X}')/\lambda)}_{\text{regularization for robustness}} \Big\}$$

左半部分用来保证模型的**准确性**，而右半部分则用来保证模型的**鲁棒性**；上式其实表述的是一个二分类的问题，对于多分类问题，作者修改上式为：

$$\min_f \mathbb{E}\left\{ \mathcal{L}(f(\boldsymbol{X}), \boldsymbol{Y}) + \max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)} \mathcal{L}(f(\boldsymbol{X}), f(\boldsymbol{X}'))/\lambda \right\}$$

其中 $\mathcal{L}(\cdot,\cdot)$ 为交叉熵损失函数；

3. 训练方法：

Algorithm 1 Adversarial training by TRADES

Input: Step sizes η_1 and η_2, batch size m, number of iterations K in inner optimization, network architecture parametrized by θ
Output: Robust network f_θ
Randomly initialize network f_θ, or initialize network with pre-trained configuration
repeat
  Read mini-batch $B = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ from training set
  for $i = 1, ..., m$ (in parallel) **do**
    $\boldsymbol{x}_i' \leftarrow \boldsymbol{x}_i + 0.001 \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is the Gaussian distribution with zero mean and identity variance
    for $k = 1, ..., K$ **do**
      $\boldsymbol{x}_i' \leftarrow \Pi_{\mathbb{B}(\boldsymbol{x}_i, \epsilon)}(\eta_1 \mathsf{sign}(\nabla_{\boldsymbol{x}_i'} \mathcal{L}(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i'))) + \boldsymbol{x}_i')$, where Π is the projection operator
    end for
  end for
  $\theta \leftarrow \theta - \eta_2 \sum_{i=1}^{m} \nabla_\theta [\mathcal{L}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) + \mathcal{L}(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i'))/\lambda]/m$
until training converged

首先在样本上面添加高斯随机噪声（cifar10-challenge 中用的是均匀分布随机噪声），然后再利用 PGD 生成对抗样本（这里希望对抗样本的概率分布和原始概率分布能够尽可能的不同），最后微调模型。

4. 实验：

(0) 实验评估：使用两个成功率来评估方法的好坏，为了说明实验方法即保证较高的成功率，又有很好的鲁棒性，我们应该希望实验得到的数据 $\mathcal{A}_{rob}$ and $\mathcal{A}_{nat}$ 都比较大；

$$\mathcal{A}_{rob}(f) = 1 - \mathcal{R}_{rob}(f)$$
$$\mathcal{A}_{nat}(f) = 1 - \mathcal{R}_{nat}(f)$$

(1) 参数 λ 的作用：

- 参数设置：

  MNIST setup. We use the CNN which has two convolutional layers, followed by two fully-connected layers. The output size of the last layer is 10. We set perturbation $\epsilon = 0.1$, perturbation step size $\eta_1 = 0.01$, number of iterations $K = 20$, learning rate $\eta_2 = 0.01$, batch size $m = 128$, and run 50 epochs on the training dataset. To evaluate the robust error, we apply FGSM^k (white-box) attack with 40 iterations and 0.005 step size. The results are in Table 4.

  CIFAR10 setup. We apply ResNet-18 [HZRS16] for classification. The output size of the last layer is 10. We set perturbation $\epsilon = 0.031$, perturbation step size $\eta_1 = 0.007$, number of iterations $K = 10$, learning rate $\eta_2 = 0.1$, batch size $m = 128$, and run 100 epochs on the training dataset. To evaluate the robust error, we apply FGSM^k (white-box) attack with 20 iterations and the step size is 0.003. The results are in Table 4.

- 实验结果：在 MNIST 上面对原始分类成功率的影响不大，但是对 CIFAR10 的影响还是比较大的。所以后面的实验中，作者分别使用 $\lambda = 1$ or $\lambda = 6$，分别保证成功率和鲁棒性；

  Table 4: Sensitivity of regularization hyperparameter λ on MNIST and CIFAR10 datasets.

	MNIST		CIFAR10	
$1/\lambda$	$\mathcal{A}_{\text{rob}}(f)$ (%)	$\mathcal{A}_{\text{nat}}(f)$ (%)	$\mathcal{A}_{\text{rob}}(f)$ (%)	$\mathcal{A}_{\text{nat}}(f)$ (%)
0.1	91.09 ± 0.0385	99.41 ± 0.0235	26.53 ± 1.1698	91.31 ± 0.0579
0.2	92.18 ± 0.0450	99.38 ± 0.0094	37.71 ± 0.6743	89.56 ± 0.2154
0.4	93.21 ± 0.0660	99.35 ± 0.0082	41.50 ± 0.3376	87.91 ± 0.2944
0.6	93.87 ± 0.0464	99.33 ± 0.0141	43.37 ± 0.2706	87.50 ± 0.1621
0.8	94.32 ± 0.0492	99.31 ± 0.0205	44.17 ± 0.2834	87.11 ± 0.2123
1.0	94.75 ± 0.0712	99.28 ± 0.0125	44.68 ± 0.3088	87.01 ± 0.2819
2.0	95.45 ± 0.0883	99.29 ± 0.0262	48.22 ± 0.0740	85.22 ± 0.0543
3.0	95.57 ± 0.0262	99.24 ± 0.0216	49.67 ± 0.3179	83.82 ± 0.4050
4.0	95.65 ± 0.0340	99.16 ± 0.0205	50.25 ± 0.1883	82.90 ± 0.2217
5.0	95.65 ± 0.1851	99.16 ± 0.0403	50.64 ± 0.3336	81.72 ± 0.0286

(2) 横向对比 - 白盒攻击：

- 参数设置：

MNIST setup. We use the CNN architecture in [CW17] with four convolutional layers, followed by three fully-connected layers. We set perturbation $\epsilon = 0.3$, perturbation step size $\eta_1 = 0.01$, number of iterations $K = 40$, learning rate $\eta_2 = 0.01$, batch size $m = 128$, and run 100 epochs on the training dataset.

CIFAR10 setup. We use the same neural network architecture as [MMS+18], i.e., the wide residual network WRN-34-10 [ZK16]. We set perturbation $\epsilon = 0.031$, perturbation step size $\eta_1 = 0.007$, number of iterations $K = 10$, learning rate $\eta_2 = 0.1$, batch size $m = 128$, and run 100 epochs on the training dataset.

in [ZSLG16, KGB17, RDV17] on the CIFAR10 dataset as they are also regularization based methods. For MNIST dataset, we apply FGSM^k (white-box) attack with 40 iterations and the step size is 0.01. For CIFAR10 dataset, we apply FGSM^k (white-box) attack with 20 iterations and the step size is 0.003, under which the

- 实验结果：

Table 5: Comparisons of TRADES with prior defense models under white-box attacks.

Defense	Defense type	Under which attack	Dataset	Distance	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$
[BRRG18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	0%
[MLW+18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	5%
[DAL+18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	0%
[SKN+18]	gradient mask	[ACW18]	CIFAR10	0.031 (ℓ_∞)	-	9%
[NKM17]	gradient mask	[ACW18]	CIFAR10	0.015 (ℓ_∞)	-	15%
[WSMK18]	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	27.07%	23.54%
[MMS+18]	robust opt.	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	87.30%	**47.04%**
[ZSLG16]	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	94.64%	0.15%
[KGB17]	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	85.25%	45.89%
[RDV17]	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	95.34%	0%
TRADES ($1/\lambda = 1$)	regularization	FGSM^{1,000} (PGD)	CIFAR10	0.031 (ℓ_∞)	88.64%	48.90%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{1,000} (PGD)	CIFAR10	0.031 (ℓ_∞)	84.92%	**56.43%**
TRADES ($1/\lambda = 1$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	88.64%	49.14%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{20} (PGD)	CIFAR10	0.031 (ℓ_∞)	84.92%	**56.61%**
TRADES ($1/\lambda = 1$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	88.64%	59.10%
TRADES ($1/\lambda = 6$)	regularization	DeepFool (ℓ_∞)	CIFAR10	0.031 (ℓ_∞)	84.92%	61.38%
TRADES ($1/\lambda = 1$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	88.64%	84.41%
TRADES ($1/\lambda = 6$)	regularization	LBFGSAttack	CIFAR10	0.031 (ℓ_∞)	84.92%	81.58%
TRADES ($1/\lambda = 1$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	88.64%	51.26%
TRADES ($1/\lambda = 6$)	regularization	MI-FGSM	CIFAR10	0.031 (ℓ_∞)	84.92%	57.95%
TRADES ($1/\lambda = 1$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	88.64%	84.03%
TRADES ($1/\lambda = 6$)	regularization	C&W	CIFAR10	0.031 (ℓ_∞)	84.92%	81.24%
[SKC18]	gradient mask	[ACW18]	MNIST	0.005 (ℓ_2)	-	55%
[MMS+18]	robust opt.	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.36%	96.01%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{1,000} (PGD)	MNIST	0.3 (ℓ_∞)	99.48%	95.60%
TRADES ($1/\lambda = 6$)	regularization	FGSM^{40} (PGD)	MNIST	0.3 (ℓ_∞)	99.48%	96.07%
TRADES ($1/\lambda = 6$)	regularization	C&W	MNIST	0.005 (ℓ_2)	99.48%	99.46%

(3) 横向对比 - 黑盒攻击：

- 参数设置：

MNIST setup. We use the CNN architecture in [CW17] with four convolutional layers, followed by three fully-connected layers. We set perturbation $\epsilon = 0.3$, perturbation step size $\eta_1 = 0.01$, number of iterations $K = 40$, learning rate $\eta_2 = 0.01$, batch size $m = 128$, and run 100 epochs on the training dataset.

CIFAR10 setup. We use the same neural network architecture as [MMS+18], i.e., the wide residual network WRN-34-10 [ZK16]. We set perturbation $\epsilon = 0.031$, perturbation step size $\eta_1 = 0.007$, number of iterations $K = 10$, learning rate $\eta_2 = 0.1$, batch size $m = 128$, and run 100 epochs on the training dataset.

For both datasets, we use FGSM^k (black-box) method to attack various defense models. For MNIST dataset, we set perturbation $\epsilon = 0.3$ and apply FGSM^k (black-box) attack with 40 iterations and the step size is 0.01. For CIFAR10 dataset, we set $\epsilon = 0.031$ and apply FGSM^k (black-box) attack with 20 iterations and the step size is 0.003. Note that the setup is the same as the setup specified in Section 5.3.1. We summarize our

- 实验结果：

Table 6: Comparisons of TRADES with prior defenses under black-box FGSM^{40} attack on the MNIST dataset. The models inside parentheses are source models which provide gradients to adversarial attackers. We provide the average cross-entropy loss value $\mathcal{L}(f(\boldsymbol{X}), \boldsymbol{Y})$ of each defense model in the bracket. The defense model 'Madry' is the same model as in the antepenultimate line of Table 5. The defense model 'TRADES' is the same model as in the penultimate line of Table 5.

Defense Model	Robust Accuracy $\mathcal{A}_{\text{rob}}(f)$	
Madry	97.43% [0.0078484]	(Natural)
TRADES	**97.63%** [0.0075324]	(Natural)
Madry	97.38% [0.0084962]	(Ours)
TRADES	**97.66%** [0.0073532]	(Madry)

Table 7: Comparisons of TRADES with prior defenses under black-box FGSM[20] attack on the CIFAR10 dataset. The models inside parentheses are source models which provide gradients to adversarial attackers. We provide the average cross-entropy loss value of each defense model in the bracket. The defense model 'Madry' is implemented based on [MMS+18], and the defense model 'TRADES' is the same model as in the 11th line of Table 5.

Defense Model	Robust Accuracy $\mathcal{A}_{\text{rob}}(f)$	
Madry	84.39% [0.0519784]	(Natural)
TRADES	**87.60%** [0.0380258]	(Natural)
Madry	66.00% [0.1252672]	(Ours)
TRADES	**70.14%** [0.0885364]	(Madry)

Links

- 论文链接：Zhang H, Yu Y, Jiao J, et al. Theoretically principled trade-off between robustness and accuracy[C]//International Conference on Machine Learning. PMLR, 2019: 7472-7482.
- 论文代码：https://github.com/yaodongyu/TRADES
- 前人分享：https://zhuanlan.zhihu.com/p/337989683

Unlabeled Data Improves Adversarial Robustness

Contribution

1. RST.
2. 在 TRADES 的基础上修改了损失函数，添加了无标签数据；
3. 文章给我的感觉是，就是利用更多数据来训练网络，不过这个数据可以是无标签的数据，这样的话就不需要进行大量的认为标注；

Notes

1. 训练方法：（证明部分直接忽略不看）

Meta-Algorithm 1 Robust self-training

Input: Labeled data $(x_1, y_1, \ldots, x_n, y_n)$ and unlabeled data $(\tilde{x}_1, \ldots, \tilde{x}_{\tilde{n}})$

Parameters: Standard loss L_{standard}, robust loss L_{robust} and unlabeled weight w

1: Learn $\hat{\theta}_{\text{intermediate}}$ by minimizing $\sum_{i=1}^{n} L_{\text{standard}}(\theta, x_i, y_i)$

2: Generate pseudo-labels $\tilde{y}_i = f_{\hat{\theta}_{\text{intermediate}}}(\tilde{x}_i)$ for $i = 1, 2, \ldots \tilde{n}$

3: Learn $\hat{\theta}_{\text{final}}$ by minimizing $\sum_{i=1}^{n} L_{\text{robust}}(\theta, x_i, y_i) + w \sum_{i=1}^{\tilde{n}} L_{\text{robust}}(\theta, \tilde{x}_i, \tilde{y}_i)$

   - 首先使用有标签数据训练网络，这里使用 standard loss 为：

$$L_{\text{standard}}(\theta, x, y) = -\log p_\theta(y \mid x)$$

   - 使用训练好的网络，标记无标签的数据；
   - 使用有标签和"自标签"的数据继续训练网络，这里使用 robust loss 为：

$$L_{\text{robust}}(\theta, x, y) = L_{\text{standard}}(\theta, x, y) + \beta L_{\text{reg}}(\theta, x),$$
$$\text{where} \quad L_{\text{reg}}(\theta, x) := \max_{x' \in \mathcal{B}_\epsilon^p(x)} D_{\text{KL}}(p_\theta(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x'))$$

     这里又到了经典的如何拟合 $L_{reg}(\theta, x)$ 项（因为寻找邻域内的最大值太困难），作者提出了两种方法：

     - **Adversarial Training**：使用 PGD 获取邻域最大值

$$L_{\text{reg}}^{\text{adv}}(\theta, x) := D_{\text{KL}}(p_\theta(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x'_{\text{PG}}[x]))$$

- **Stability Training**：使用 高斯分布 采样邻域的值

$$L_{\text{reg}}^{\text{stab}}(\theta, x) := \mathbb{E}_{x' \sim \mathcal{N}(x, \sigma^2 I)} D_{\text{KL}}(p_\theta(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x'))$$

使用 Stability Training 的网络在测试的时候也做了改变，**模型输出的是 高斯分布 采样邻域中的可能性最大的分类**

$$g_\theta(x) := \underset{y \in \mathcal{Y}}{\text{argmax}} \; q_\theta(y \mid x), \quad \text{where} \quad q_\theta(y \mid x) := \mathbb{P}_{x' \sim \mathcal{N}(x, \sigma^2 I)}(f_\theta(x') = y)$$

2. 实验：

(0) 实验参数：

- 数据集：大致意思就是从 **80M的CIFAR10-TINY** 中为每个类挑选出 50K 张图片组成一个 500K 大小的无标签数据集；

To obtain unlabeled data distributed similarly to the CIFAR-10 images, we use the 80 Million Tiny Images (80M-TI) dataset [46], of which CIFAR-10 is a manually labeled subset. However, most images in 80M-TI do not correspond to CIFAR-10 image categories. To select relevant images, we train an 11-way classifier to distinguish CIFAR-10 classes and an 11^{th} "non-CIFAR-10" class using a Wide ResNet 28-10 model [54] (the same as in our experiments below). For each class, we select additional 50K images from 80M-TI using the trained model's predicted scores[2]—this is our 500K images unlabeled which we add to the 50K CIFAR-10 training set when performing `RST`. We provide a detailed description of the data sourcing process in Appendix B.6.

- 模型 & 训练参数：

Architecture. We use a Wide ResNet 28-10 architecture, as in [29] and similarly to [56], who use a 34-10 variant.

Robust self-training. We set the regularization weight $\beta = 6$ as in [56]. We implicitly set the unlabeled data weight to $w = 50\text{K}/500\text{K} = 0.1$ by composing every batch from equal parts labeled and unlabeled data.

Adversarial self-training. We compute x_{PG} exactly as in [56], with step size 0.007, 10 iterations and $\epsilon = 8/255$.

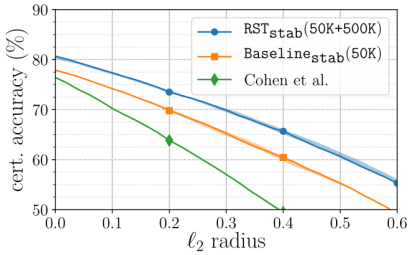
Stability training. We set the additive noise variance to $\sigma = 0.25$. We perform the certification using the randomized smoothing protocol described in [9], with parameters $N_0 = 100$, $N = 10^4$, $\alpha = 10^{-3}$ and noise variance $\sigma = 0.25$.

(1) 经验性防御（heuristic defense）：主要关注 L_∞ 攻击；（✩ 针对作者提供的这种方法，作者还自己对 PGD 做了修改以达到最好的攻击成功率，这一点在做攻击的时候可以借鉴一下。）

Model	PG_{Madry}	$\text{PG}_{\text{TRADES}}$	PG_{Ours}	CW [7]	Best attack	No attack
RST_{adv}(50K+500K)	63.1	63.1	62.5	64.9	**62.5** ±0.1	**89.7** ±0.1
TRADES [56]	55.8	56.6	55.4	65.0	55.4	84.9
Adv. pre-training [18]	57.4	58.2	57.7	-	57.4†	87.1
Madry et al. [29]	45.8	-	-	47.8	45.8	87.3
Standard self-training	-	0.3	0	-	0	96.4

Table 1: **Heuristic defense.** CIFAR-10 test accuracy under different optimization-based ℓ_∞ attacks of magnitude $\epsilon = 8/255$. Robust self-training (RST) with 500K unlabeled Tiny Images outperforms the state-of-the-art robust models in terms of robustness as well as standard accuracy (no attack). Standard self-training with the same data does not provide robustness. †: A projected gradient attack with 1K restarts reduces the accuracy of this model to 52.9%, evaluated on 10% of the test set [18].

(2) 证明性防御（certified defense）：同时关注 L_∞ 和 L_2 攻击；



(a)

Model	ℓ_∞ acc. at $\epsilon = \frac{2}{255}$	Standard acc.
RST$_{stab}$(50K+500K)	**63.8** ± 0.5	**80.7** ± 0.3
Baseline$_{stab}$(50K)	58.6 ± 0.4	77.9 ± 0.1
Wong et al. (single) [50]	53.9	68.3
Wong et al. (ensemble) [50]	**63.6**	64.1
IBP [17]	50.0	70.2

(b)

Figure 1: **Certified defense.** Guaranteed CIFAR-10 test accuracy under all ℓ_2 and ℓ_∞ attacks. Stability-based robust self-training with 500K unlabeled Tiny Images (RST$_{stab}$(50K+500K)) outperforms stability training with only labeled data (Baseline$_{stab}$(50K)). **(a)** Accuracy vs. ℓ_2 radius, certified via randomized smoothing [9]. Shaded regions indicate variation across 3 runs. Accuracy at ℓ_2 radius 0.435 implies accuracy at ℓ_∞ radius $2/255$. **(b)** The implied ℓ_∞ certified accuracy is comparable to the state-of-the-art in methods that directly target ℓ_∞ robustness.

(3) SVHN（Street View House Numbers）实验：实验的结果表明，数据的标签对于模型的鲁棒性影响并不大；（我这里比较好奇的是，在 SVHN 这个任务上作者提出的方法并没有优于 $Baseline_{adv}(604K)$ 的效果）

Model	PG$_{0ur}$	No attack
Baseline$_{adv}$(73K)	75.3 ± 0.4	94.7 ± 0.2
RST$_{adv}$(73K+531K)	86.0 ± 0.1	97.1 ± 0.1
Baseline$_{adv}$(604K)	86.4 ± 0.2	97.5 ± 0.1

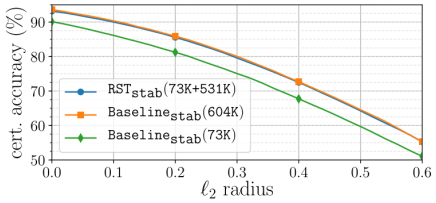


Figure 2: SVHN test accuracy for robust training without the extra data, with unlabeled extra (self-training), and with the labeled extra data. Left: Adversarial training and accuracies under ℓ_∞ attack with $\epsilon = 4/255$. Right: Stability training and certified ℓ_2 accuracies as a function of perturbation radius. Most of the gains from extra data comes from the unlabeled inputs.

Links

- 论文链接：Carmon Y, Raghunathan A, Schmidt L, et al. Unlabeled data improves adversarial robustness[J]. arXiv preprint arXiv:1905.13736, 2019.
- 论文代码：https://github.com/yaircarmon/semisup-adv

Defense Against Adversarial Attacks Using Feature Scattering-based Adversarial Training

Contribution

1. 使用 Wasserstein Distance 来生成对抗样本；

Notes

1. Background：

(1) Adversarial Training，可以被形式化为 $min - max$ 问题：

$$\min_{\boldsymbol{\theta}} \left[\max_{\mathbf{x}' \in \mathcal{S}_{\mathbf{x}}} \mathcal{L}(\mathbf{x}', y; \boldsymbol{\theta}) \right]$$

其中内层最大值经常用对抗样本生成来近似，如使用单步的 $FGSM$ 算法，或者是多步的 PGD 算法（算法第一次迭代时首先会添加上一个随机噪声，然后再迭代生成对抗样本），PGD 算法公式如下：

$$\mathbf{x}^{t+1} = \mathcal{P}_{\mathcal{S}_{\mathbf{x}}}\left(\mathbf{x}^t + \alpha \cdot \operatorname{sign}\left(\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^t, y; \boldsymbol{\theta})\right)\right)$$

(2) 对抗训练的问题：

- 标签泄露（Label Leaking）：生成的对抗扰动本身和目标类别是密切相关的，导致训练好的模型看到测试集的对抗扰动便知道目标分类，而与原始的图片无关；
- 梯度隐藏（Gradient Masking）：指的是训练后的模型学习到了尽可能生成一些无用的梯度（这个和我想象中的"梯度隐藏"的概念不同，从作者的解释来看是模型学习到了这种生成无用梯度的能力，可能是我没有真正理解这个含义，因为我感觉即使是梯度隐藏也是可以起到防御对抗攻击的效果的）；

2. 最优运输理论（Optimal Transport Theory）

(1) 参考链接：

- 【数学】Wasserstein Distance
- 机器学习工具（二）Notes of Optimal Transport
- 令人拍案叫绝的Wasserstein GAN

(2) Wasserstein Distance的优点：**Wasserstein距离相比KL散度、JS散度的优越性在于，即便两个分布没有重叠，Wasserstein距离仍然能够反映它们的远近**

3. 训练方法：

Algorithm 1 Feature Scattering-based Adversarial Training

Input: dataset S, training epochs K, batch size n, learning rate γ, budget ϵ, attack iterations T
for $k = 1$ **to** K **do**
  for random batch $\{\mathbf{x}_i, y_i\}_{i=1}^n \sim S$ **do**
    initialization: $\boldsymbol{\mu} = \sum_i u_i \delta_{\mathbf{x}_i}$, $\boldsymbol{\nu} = \sum_i v_i \delta_{\mathbf{x}'_i}$, $\mathbf{x}'_i \sim B(\mathbf{x}_i, \epsilon)$
    feature scattering (maximizing feature matching distance $\mathcal{D}$ *w.r.t.* $\boldsymbol{\nu}$):
    for $t = 1$ **to** T **do**
      $\cdot\ \mathbf{x}'_i \leftarrow \mathcal{P}_{\mathcal{S}_\mathbf{x}}\big(\mathbf{x}'_i + \epsilon \cdot \text{sign}\big(\nabla_{\mathbf{x}'_i} \mathcal{D}(\boldsymbol{\mu}, \boldsymbol{\nu})\big)\big) \quad \forall i = 1, \cdots, n, \quad \boldsymbol{\nu} = \sum_i v_i \delta_{\mathbf{x}'_i}$
    end for
    adversarial training (updating model parameters):
    $\cdot\ \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \cdot \frac{1}{n} \sum_{i=1}^n \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}'_i, y_i; \boldsymbol{\theta})$
  end for
end for
Output: model parameter $\boldsymbol{\theta}$.

我的理解：

- 简单的来看，作者提出的方法和已有的方法的不同之处在于使用了 Wasserstein Distance 作为生成对抗样本时的损失函数；
- 思考一下，变换了这个损失函数的作用有可能那么大么？从 Wasserstein Distance 自身的有点来看，它的优点主要是体现在两个分布没有重叠时，还能够有效地指导网络训练。所以从对抗样本的生成角度来看，Wasserstein Distance 对于较大扰动时的对抗样本生成是更加有效的，因为此时两个样本的解码概率分布可能已经差异十分大了，用 KL 散度并不能更好地指导生成对抗样本；**概括一下，能够指导生成更多样化的对抗样本，从而更好地进行对抗训练**；
- 再思考一下，这篇文章作者一直是从 feature 这个角度来说的，其实我一直不太明白作者这样说有什么意义，但是我们揣测一下的话，因为在利用 Wasserstein Distance 生成对抗样本的时候，其实是要利用到一整个batch的样本来调整扰动的，这也是这个方法和前面方法的不同之处；**概括一下，用到了batch来生成对抗样本**；

方法示意图：

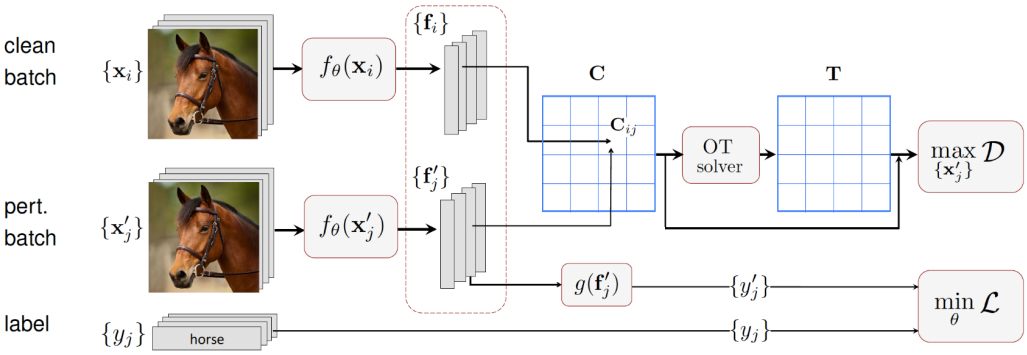


Figure 1: **Feature Scattering-based Adversarial Training Pipeline**. The adversarial perturbations are generated collectively by feature scattering, *i.e.*, maximizing the feature matching distance between the clean samples $\{\mathbf{x}_i\}$ and the perturbed samples $\{\mathbf{x}'_j\}$. The model parameters are updated by minimizing the cross-entropy loss using the perturbed images $\{\mathbf{x}'_j\}$ as the training samples.

4. 实验：

(0) 实验参数：

- 数据集 & 基准方法：

[illegible][3] We conduct extensive experiments across several benchmark datasets including CIFAR10 [31], CIFAR100 [31] and SVHN [42]. We use Wide ResNet (WRN-28-10) [68] as the network structure following [36]. We compare the performance of the proposed method with a number of baseline methods, including: *i)* the model trained with standard approach using clean images (**`Standard`**) [31], *ii)* PGD-based approach from Madry *et al.* (**`Madry`**) [36], which is one of the most effective defense method [2], *iii)* another recent method performs adversarial training with both image and label adversarial perturbations (**`Bilateral`**) [61]. [illegible] 0.01 for SVHN.

- 测试参数：

and Sinkhorn algorithm [12] with regularization of 0.01 is used. For *testing*, model robustness is evaluated by approximately computing an upper bound of robustness on the test set, by measuring the accuracy of the model under different adversarial attacks, including white-box FGSM [24], PGD [36], CW [8] (CW-loss [8] within the PGD framework) attacks and variants of black-box attacks.

(1) 白盒攻击结果：

Models	Clean	Accuracy under White-box Attack ($\epsilon = 8$)								
		FGSM	PGD10	PGD20	PGD40	PGD100	CW10	CW20	CW40	CW100
`Standard`	**95.6**	36.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
`Madry`	85.7	54.9	45.1	44.9	44.8	44.8	45.9	45.7	45.6	45.4
`Bilateral`	91.2	70.7	–	57.5	–	55.2	–	56.2	–	53.8
`Proposed`	90.0	**78.4**	**70.9**	**70.5**	**70.3**	**68.6**	**62.6**	**62.4**	**62.1**	**60.6**

Table 1: Accuracy comparison of the **`Proposed`** approach with **`Standard`**, **`Madry`** [36] and **`Bilateral`** [61] methods on CIFAR10 under different threat models.

Models	Clean	White-box Attack ($\epsilon = 8$)				
		FGSM	PGD20	PGD100	CW20	CW100
`Standard`	**97.2**	53.0	0.3	0.1	0.3	0.1
`Madry`	93.9	68.4	47.9	46.0	48.7	47.3
`Bilateral`	94.1	69.8	53.9	50.3	–	48.9
`Proposed`	96.2	**83.5**	**62.9**	**52.0**	**61.3**	**50.8**

Models	Clean	White-box Attack ($\epsilon = 8$)				
		FGSM	PGD20	PGD100	CW20	CW100
`Standard`	**79.0**	10.0	0.0	0.0	0.0	0.0
`Madry`	59.9	28.5	22.6	22.3	23.2	23.0
`Bilateral`	68.2	60.8	26.7	25.3	–	22.1
`Proposed`	73.9	**61.0**	**47.2**	**46.2**	**34.6**	**30.6**

Table 2: Accuracy comparison on (a) SVHN and (b) CIFAR100.

Links

- 论文链接：Zhang H, Wang J. Defense against adversarial attacks using feature scattering-based adversarial training[J]. arXiv preprint arXiv:1907.10764, 2019.
- 论文代码：https://github.com/Haichao-Zhang/FeatureScatter

Using Pre-Training Can Improve Model Robustness and Uncertainty

Contribution

1. 验证了预训练的作用：不仅可以提高模型收敛速率，而且可以提高模型的鲁棒性和不确定性；

Links

- 论文链接：Hendrycks D, Lee K, Mazeika M. Using pre-training can improve model robustness and uncertainty[C]//International Conference on Machine Learning. PMLR, 2019: 2712-2721.
- 论文代码： https://github.com/hendrycks/pre-training

Boosting Adversarial Training with Hypersphere Embedding

Contribution

1. 修改了loss函数，希望生成对抗样本的时候能够只学习角度的变换；

Notes

1. 思想：作者希望在生成对抗样本的时候，尽可能地旋转样本的角度，而不是缩放大小；

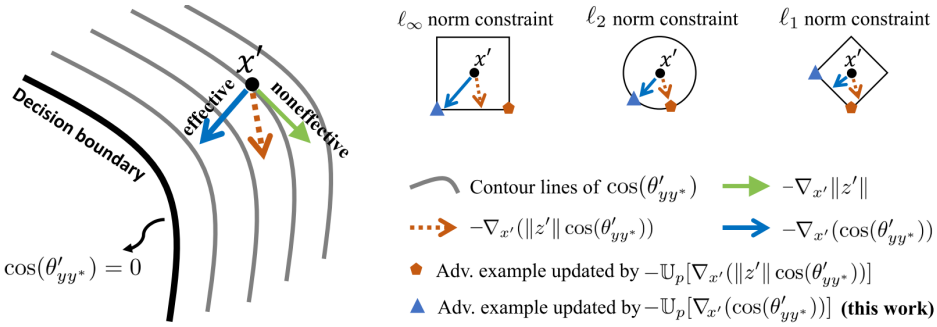


Figure 1: Intuitive illustration in the input space. When applying FN in $\mathcal{L}_{\mathrm{A}}$, the adversary can take more effective update steps to move x' across the decision boundary defined by $\cos(\theta'_{yy^*}) = 0$.

2. 方法：对参数和模型提取出来的特征，在最后一个softmax层之前做normalization，然后计算loss；

Note that in Eq. (1) there is $\mathbf{W}^\top z = (W_1^\top z, \cdots, W_L^\top z)$, and $\forall l \in [L]$, the inner product $W_l^\top z = \|W_l\|\|z\| \cos(\theta_l)$, where $\theta_l = \angle(W_l, z)$.[1] Then the WN and FN operations can be denoted as

$$\textbf{WN operation: } \widetilde{W}_l = \frac{W_l}{\|W_l\|}; \textbf{ FN operation: } \widetilde{z} = \frac{z}{\|z\|}, \tag{4}$$

Let $\cos\boldsymbol{\theta} = (\cos(\theta_1), \cdots, \cos(\theta_L))$ and $\widetilde{\mathbf{W}}$ be the weight matrix after executing WN on each column vector W_l. Then, the output predictions of the DNNs with HE become

$$\widetilde{f}(x) = \mathbb{S}(\widetilde{\mathbf{W}}^\top \widetilde{z}) = \mathbb{S}(\cos\boldsymbol{\theta}), \tag{5}$$

where no bias vector b exists in $\widetilde{f}(x)$ [42, 75]. In contrast, the **AM operation** is only performed in the training phase, where $\widetilde{f}(x)$ is fed into the CE loss with a margin m [76], formulated as

$$\mathcal{L}_{\mathrm{CE}}^m(\widetilde{f}(x), y) = -1_y^\top \log \mathbb{S}(s \cdot (\cos\boldsymbol{\theta} - m \cdot 1_y)). \tag{6}$$

该处的loss实现方法和 CosFace 相同。

3. 实验：

Table 2: Classification accuracy (%) on **CIFAR-10** under the *white-box* threat model. The perturbation $\epsilon = 0.031$, step size $\eta = 0.003$. We highlight the best-performance model under each attack.

Defense	Clean	PGD-20	PGD-500	MIM-20	FGSM	DeepFool	C&W	FeaAtt.	FAB
PGD-AT	86.75	53.97	51.63	55.08	59.70	57.26	84.00	52.38	51.23
PGD-AT+**HE**	86.19	59.36	57.59	60.19	63.77	61.56	84.07	52.88	**54.45**
ALP	87.18	52.29	50.13	53.35	58.99	59.40	84.96	49.55	50.54
ALP+**HE**	**89.91**	57.69	51.78	58.63	65.08	**65.19**	**87.86**	48.64	51.86
TRADES	84.62	56.48	54.84	57.14	61.02	60.70	81.13	55.09	53.58
TRADES+**HE**	84.88	**62.02**	**60.75**	**62.71**	**65.69**	60.48	81.44	**58.13**	53.50

Table 3: Validation of combining FastAT and FreeAT with HE and m-HE on **CIFAR-10**. We report the accuracy (%) on clean and PGD, as well as the total training time (min).

Defense	Epo.	Clean	PGD-50	Time
FastAT	30	**83.80**	46.40	*11.38*
FastAT+**HE**	30	82.58	52.55	*11.48*
FastAT+**m-HE**	30	83.14	**53.49**	*11.49*
FreeAT	10	77.21	46.14	*15.78*
FreeAT+**HE**	10	76.85	50.98	*15.87*
FreeAT+**m-HE**	10	**77.59**	**51.85**	*15.91*

Table 4: Top-1 classification accuracy (%) on **ImageNet** under the *white-box* threat model.

Model	Method	Clean	PGD-10	PGD-50
ResNet-50	FreeAT	60.28	32.13	31.39
	FreeAT+**HE**	**61.83**	**40.22**	**39.85**
ResNet-152	FreeAT	65.20	36.97	35.87
	FreeAT+**HE**	**65.41**	**43.24**	**42.60**
WRN-50-2	FreeAT	64.18	36.24	35.38
	FreeAT+**HE**	**65.28**	**43.83**	**43.47**
WRN-101-2	FreeAT	66.15	39.35	38.23
	FreeAT+**HE**	**66.37**	**45.35**	**45.04**

Links

- 论文链接：[Pang T, Yang X, Dong Y, et al. Boosting adversarial training with hypersphere embedding[J]. arXiv preprint arXiv:2002.08619, 2020.]
- 论文代码：[https://github.com/ShawnXYang/AT_HE]

Overfitting in adversarially robust deep learning

Contribution

1. 使用大量的实验来验证：对抗训练的模型需要设置一个 `early stop`，并且这种模型过拟合的问题并不能通过现有的一些抑制过拟合的方法来解决；

Notes

1. 实验 1：过拟合会对对抗训练的鲁棒性产生负面影响；
    - 在 l_∞ 对抗攻击下：

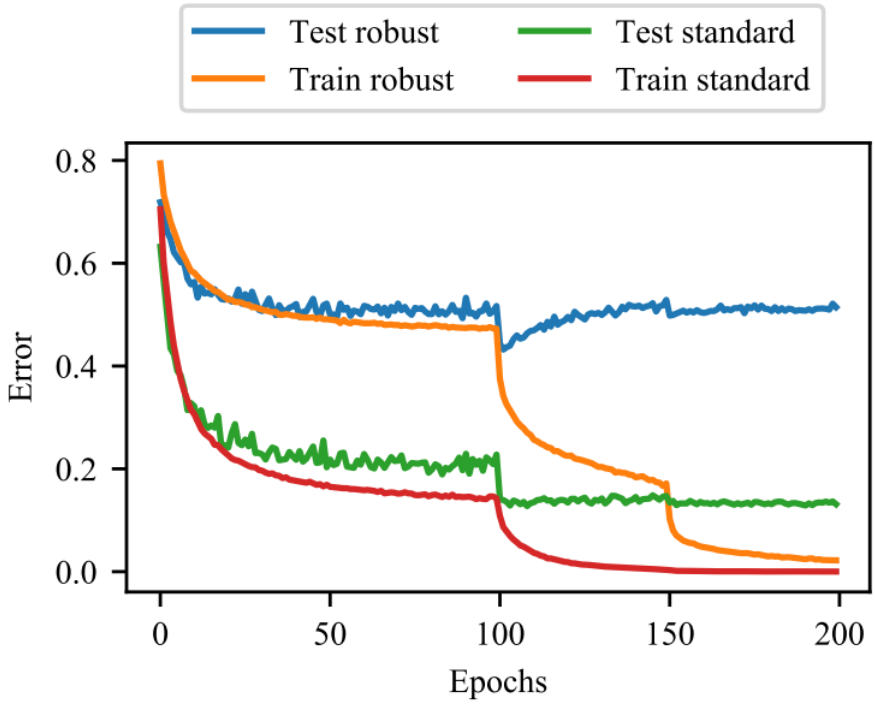


Figure 1. The learning curves for a robustly trained model replicating the experiment done by Madry et al. (2017) on CIFAR-10. The curves demonstrate "robust overfitting"; shortly after the first learning rate decay the model momentarily attains 43.2% robust error, and is actually more robust than the model at the end of training, which only attains 51.4% robust test error against a 10-step PGD adversary for ℓ_∞ radius of $\epsilon = 8/255$. The learning rate is decayed at 100 and 150 epochs.

- 在 l_2 和 l_∞ 对抗攻击下的多个数据集中：

Table 1. Robust performance showing the occurrence of robust overfitting across datasets and perturbation threat models. The "best" robust test error is the lowest test error observed during training. The final robust test error is averaged over the last five epochs. The difference between final and best robust test error indicates the degradation in robust performance during training.

			Robust test error (%)		
Dataset	Norm	Radius	Final	Best	Diff
SVHN	ℓ_∞	$8/255$	45.6 ± 0.40	39.0	6.6
	ℓ_2	$128/255$	26.4 ± 0.27	25.2	1.2
CIFAR-10	ℓ_∞	$8/255$	51.4 ± 0.41	43.2	8.2
	ℓ_2	$128/255$	31.1 ± 0.46	28.4	2.7
CIFAR-100	ℓ_∞	$8/255$	78.6 ± 0.39	71.9	6.7
	ℓ_2	$128/255$	62.5 ± 0.09	56.8	5.7
ImageNet	ℓ_∞	$4/255$	85.5 ± 8.87	62.7	22.8
	ℓ_2	$76/255$	94.8 ± 1.16	63.0	31.8

2. 实验 2：不同的学习率退化算法对过拟合现象的影响；（都会出现过拟合现象）

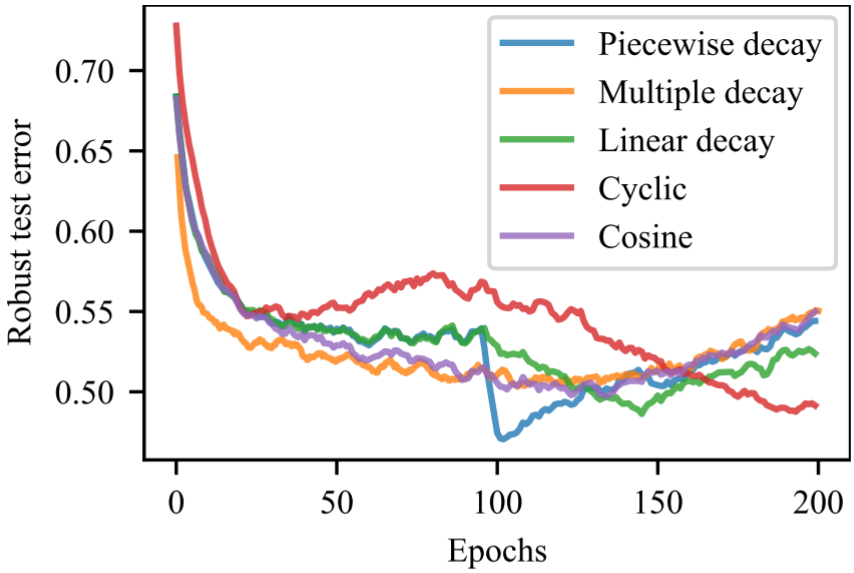


Figure 2. Robust test error over training epochs for various learning rate schedules on CIFAR-10. None of the alternative smoother learning rate schedules can achieve a peak performance competitive with the standard piecewise decay learning rate, indicating that the peak performance is obtained by having a single discrete jump. Note that the multiple decay schedule is actually run for 500 epochs, but compressed into this plot for a clear comparison.

3. 实验 3：可以在实验中设置一个验证集来判断对抗训练是否已经过拟合；

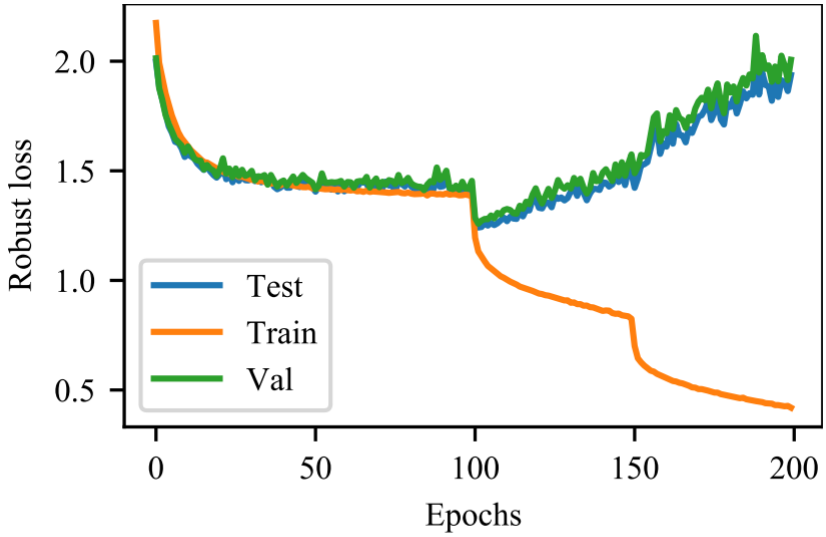


Figure 4. Learning curves for a CIFAR-10 pre-activation ResNet18 model trained with a hold-out validation set of 1,000 examples. We find that the hold-out validation set is enough to reflect the test set performance, and stopping based on the validation set is able to prevent overfitting and recover 46.9% robust test error, in comparison to 46.7% achieved by the best-performing model checkpoint.

4. 实验 4：不同的模型大小对过拟合的影响；（都会出现过拟合现象，但是复杂网络能够达到最优鲁棒性）

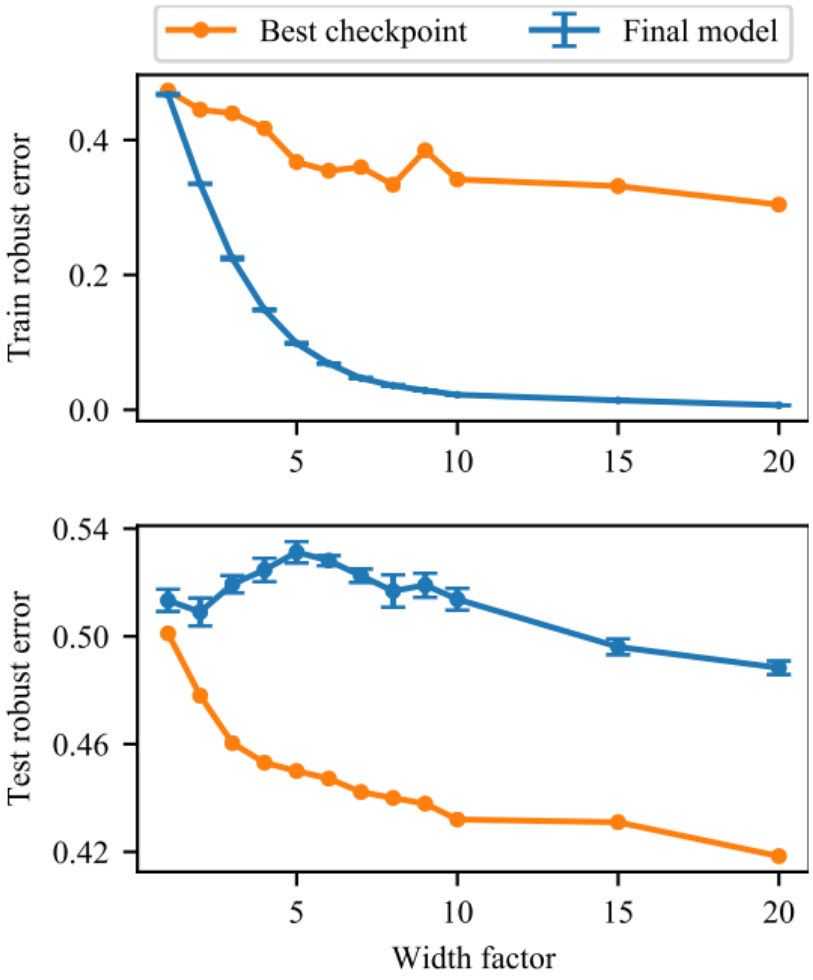


Figure 5. Generalization curves depicting double descent for adversarially robust generalization, where hypothesis class complexity is controlled by varying the width factor for a wide residual network. Each final model point represents the average performance over the last 5 epochs with the corresponding width factor from training until convergence. The best checkpoint refers to the lowest robust test error achieved by a model checkpoint during training, and illustrates the significant gap in performance between the best and final models resulting from robust overfitting.

5. 实验 5：现有的抑制过拟合的方法对过拟合的影响；（都会出现过拟合现象，并且除了 `Semi-Supervised` 方法，其他方法效果都和 `early stop` 相似）

Table 2. Robust performance of PGD-based adversarial training with different regularization methods on CIFAR-10 using a PreActResNet18 for ℓ_∞ with radius $8/255$. The "best" robust test error is the lowest test error achieved during training whereas the final robust test error is averaged over the last five epochs. Each of the regularization methods listed is trained using the optimally chosen hyperparameter. Pure early stopping is done with a validation set.

	Robust Test Error (%)		
Reg Method	Final	Best	Diff
Early stopping w/ val	**46.9**	46.7	0.2
ℓ_1 regularization	53.0 ± 0.39	48.6	4.4
ℓ_2 regularization	55.2 ± 0.4	46.4	~~55.2~~
Cutout	48.8 ± 0.79	46.7	2.1
Mixup	49.1 ± 1.32	46.3	2.8
Semi-supervised	47.1 ± 4.32	40.2	6.9

Links

- 论文链接：Rice L, Wong E, Kolter Z. Overfitting in adversarially robust deep learning[C]//International Conference on Machine Learning. PMLR, 2020: 8093-8104.
- 论文代码：https://github.com/locuslab/robust_overfitting