A Closer Look at Accuracy vs. Robustness

Chao Chen

LEHIGH UNIVERSITY

Notations

Sample $x \in \mathcal{X} \subset \mathbb{R}^d$ with ground truth y.

A model $f: \mathcal{X} \to \mathbb{R}^C$ predicts the probability of x in C classes.

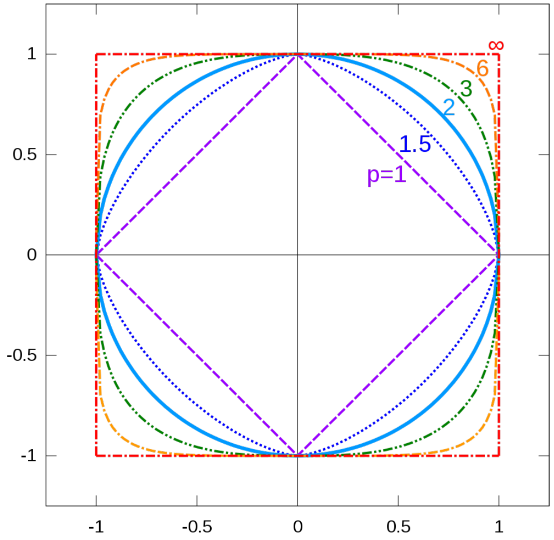
$[C] = \{1,2,\ldots,C\}$

$f(x)_i$ is the i-th element of the vector $f(x)$.

$dist$ is a metric (distance function).

$\mathbb{B}(x,\epsilon)$ is a ball of radius $\epsilon > 0$ around x in the metric space $dist$.

$\mathbb{B}_\infty$ denotes a ℓ_∞ ball.



Definitions

Robustness: (prediction is unchanged)

A classifier is robust at x with radius $\epsilon > 0$ if for all $x' \in \mathbb{B}(x, \epsilon)$, $f(x') = f(x)$

Astuteness: (prediction is correct)

A classifier is astute at (x, y) if for all $x' \in \mathbb{B}(x, \epsilon)$, $f(x') = y$

The astuteness (of f at radius $\epsilon > 0$ under a distribution μ):

$$\Pr_{(x,y)\sim\mu}[g(x') = y \text{ for all } x' \in \mathbb{B}(x, \epsilon)]$$

The goal of robust classification is to find f with highest astuteness (robust accuracy).

Definitions

Local Lipschitzness:

f is L-locally Lipschitz at radius r if for each $i \in [C]$, we have

$$|f(x)_i - f(x')_i| \le L \cdot dist(x, x'), \qquad \forall x' \text{ with } dist(x, x') \le r$$

Separation:

$\mathcal{X}$ contain C disjoint classes $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(C)}$, where all points in $\mathcal{X}^{(i)}$ have label $i \in [C]$.

r-separation:

$$dist\left(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}\right) \ge 2r, \qquad \text{for all } i \neq j$$

where $dist\left(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}\right) = \min_{x \in \mathcal{X}^{(i)}, x' \in \mathcal{X}^{(j)}} dist(x, x')$

Are real image datasets r-separation?

Experiments on four datasets: MNIST, CIFAR-10, SVHN, ResImageNet.

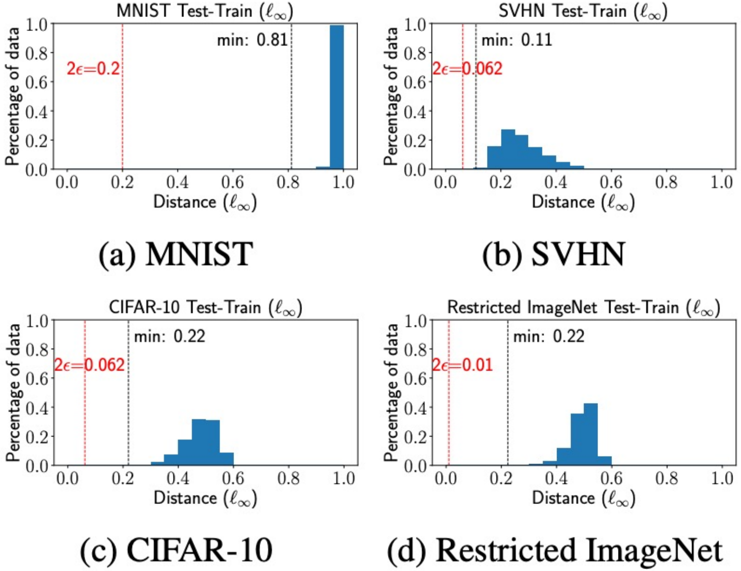
Train-Train: ℓ_∞ distance between the training sample and its closest neighbor with different label in the training set.

Test-Train: ℓ_∞ distance between the test sample and its closest neighbor with different label in the training set.

ϵ: the typical adversarial attack radius for the datasets.

[Baring a handful of highly noisy examples.]

	adversarial perturbation ε	minimum Train-Train separation	minimum Test-Train separation
MNIST	0.1	0.737	0.812
CIFAR-10	0.031	0.212	0.220
SVHN	0.031	0.094	0.110
ResImageNet	0.005	0.180	0.224



(a) MNIST (b) SVHN (c) CIFAR-10 (d) Restricted ImageNet

When the dataset is r-separation…

It is possible to find both robust and accurate model for r-separated data.

Consider function $f: \mathcal{X} \rightarrow \mathbb{R}^C$ and $x \in \mathcal{X}$ with true label $y \in [C]$, if

I. f is $\frac{1}{r}$-Locally Lipschitz in radius r around x, and

II. $f(x)_j - f(x)_y \geq 2$ for all $j \neq y$

Then $g(x) = \arg\min_i f(x)_i$ is astute at x with radius r.

Intuitively,

Condition-I indicates that the changes of prediction in $\mathbb{B}(x, r)$ are slow.

Condition-II indicates that the function has relatively high "confidence" for x's ground truth.

When the dataset is r-separation…

For a r-separated dataset. Consider the function

I. f is $\frac{1}{r}$-Locally Lipschitz in radius r around x, and

II. $f(x)_j - f(x)_y \geq 2$ for all $j \neq y$.

Then $g(x) = \arg\min_i f(x)_i$ is astute at x with radius r.

Proof:

$$\left|f(x)_j - f(x')_j\right| \leq \frac{1}{r} \cdot dist(x, x') \leq \frac{1}{r} \cdot r = 1, \qquad \forall x' \text{ with } dist(x, x') \leq r$$

$$f(x')_j \geq f(x)_j - 1 \geq f(x)_y + 1 \geq f(x')_y$$

Thus, for all $j \neq y$: $\arg\min_i f(x')_i = \arg\min_i f(x)_i = y$.

When the distribution is r-separated, there exists an astute classifier $g(x) = \arg\min_i f(x)_i$ classifies x correctly and is astute with radius r.

When the dataset is r-separation…

When $\mathcal{X}$ is r-separated, denoting C classes $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(C)}$. There exists a function f such that:

I. f is $\frac{1}{r}$ -locally Lipschitz in a ball of radius r around each $x \in \cup_{i \in [C]} \mathcal{X}$, and

II. the classifier $g(x) = \arg\min_i f(x)_i$ has astuteness 1 with radius r.

Intuitively,

Condition-I indicates that f doesn't change a lot near each data x.

Condition-II means that the classifier based on f is astute.

When the dataset is r-separation…

When $\mathcal{X}$ is r-separated, denoting C classes $\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(C)}$. There exists a function f such that:

I. f is $\frac{1}{r}$ -locally Lipschitz in a ball of radius r around each $x \in \cup_{i \in [C]} \mathcal{X}$, and

II. the classifier $g(x) = \arg\min_i f(x)_i$ has astuteness 1 with radius r.

Proof:

Consider a vector-valued function $f(x)\colon \mathcal{X} \to \mathbb{R}^C$ and $dist\left(x, \mathcal{X}^{(i)}\right) = \min_{z \in \mathcal{X}^{(i)}} dist(x, z)$

$$f(x) = \frac{1}{r} \cdot \left(dist\left(x, \mathcal{X}^{(1)}\right), \dots, dist\left(x, \mathcal{X}^{(C)}\right)\right)$$

Then for any x, we have

$$f(x)_i - f(x')_i = \frac{dist\left(x, \mathcal{X}^{(i)}\right) - dist\left(x', \mathcal{X}^{(i)}\right)}{r} \leq \frac{dist(x, x')}{r}$$

When the dataset is r-separation…

When $\mathcal{X}$ is r-separated, denoting C classes $\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(C)}$. There exists a function f such that:

I. f is $\frac{1}{r}$ -locally Lipschitz in a ball of radius r around each $x \in \cup_{i \in [C]} \mathcal{X}$, and

II. the classifier $g(x) = \arg\min_i f(x)_i$ has astuteness 1 with radius r.

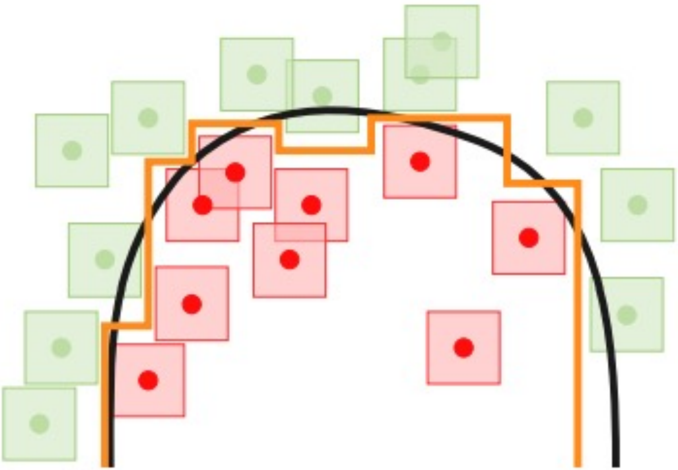
Proof: as for condition-II, we proof "$\forall x \in \mathcal{X}^{(y)}$, $f(x)_j - f(x)_y \geq 2$ for all $j \neq y$" instead.

Since $x \in \mathcal{X}^{(y)}$, $f(x)_y = dist\left(x, \mathcal{X}^{(y)}\right) = 0$

$$f(x)_j - f(x)_y = \frac{dist\left(x, \mathcal{X}^{(j)}\right)}{r} \geq \frac{dist\left(\mathcal{X}^{(y)}, \mathcal{X}^{(j)}\right)}{r} \geq \frac{2r}{r} = 2$$

When the dataset is r-separation…

An example to show robust model with small Lipschitzness (orange curve), and vulnerable model to adversarial attacks (black curve).



Experiments

Explore two questions (why existing works trade robustness off for accuracy):

- How locally Lipschitz are the classifiers produced by existing training methods?
- How well do classifiers produced by existing training methods generalize?

Training methods:

- Natural training (Natural);
- Gradient Regularization (GR);
- Locally Linear Regularization (LLR);
- Adversarial Training (AT);
- TRADES [Higher β means higher weight given to enforcing local Lipschitzness];
- Robust Self Training (RST) [Higher λ in RST means higher weight is given to robust accuracy].

Adversarial attacks: PGD

Experiments

- How locally Lipschitz are the classifiers produced by existing training methods?

[CNN1: smaller network, CNN2: larger network in MNIST dataset.]

“test Lipschitz” is quantified by

$$\frac{1}{n}\sum_{i=1}^{n} \max_{x_i' \in \mathbb{B}_\infty(x_i,\epsilon)} \frac{|f(x_i) - f(x_i')|_1}{\left|x_i - x_i'\right|_\infty}$$

and is evaluated by a PGD-like procedure (take steps towards the gradient in multiple steps.)

architecture	CNN1						CNN2					
	train acc.	test acc.	adv test acc.	test lipschitz	gap	adv gap	train acc.	test acc.	adv test acc.	test lipschitz	gap	adv gap
Natural	100.00	99.20	59.83	67.25	0.80	0.45	100.00	99.51	86.01	23.06	0.49	-0.28
GR	99.99	99.29	91.03	26.05	0.70	3.49	99.99	99.55	93.71	20.26	0.44	2.55
LLR	100.00	99.43	92.14	30.44	0.57	4.42	100.00	99.57	95.13	9.75	0.43	2.28
AT	99.98	99.31	97.21	8.84	0.67	2.67	99.98	99.48	98.03	6.09	0.50	1.92
RST(λ=.5)	100.00	99.34	96.53	11.09	0.66	3.16	100.00	99.53	97.72	8.27	0.47	2.27
RST(λ=1)	100.00	99.31	96.96	11.31	0.69	2.95	100.00	99.55	98.27	6.26	0.45	1.73
RST(λ=2)	100.00	99.31	97.09	12.39	0.69	2.87	100.00	99.56	98.48	4.55	0.44	1.52
TRADES(β=1)	99.81	99.26	96.60	9.69	0.55	2.10	99.96	99.58	98.10	4.74	0.38	1.70
TRADES(β=3)	99.21	98.96	96.66	7.83	0.25	1.33	99.80	99.57	98.54	2.14	0.23	1.18
TRADES(β=6)	97.50	97.54	93.68	2.87	-0.04	0.37	99.61	99.59	98.73	1.36	0.02	0.80

Experiments

- How locally Lipschitz are the classifiers produced by existing training methods?

1. Models trained by Natural, GR, LLR have significantly worse Lipschitzness than others.
2. Models trained by TRADE are the most locally Lipschitz overall.
3. Local Lipschitzness is correlated with adversarial attacks.
4. There are diminishing returns in the correlation between robustness and local Lipschitzness.

architecture	CNN1				CNN2			
	train acc.	test acc.	adv test acc.	test lipschitz	train acc.	test acc.	adv test acc.	test lipschitz
Natural	100.00	99.20	59.83	67.25	100.00	99.51	86.01	23.06
GR	99.99	99.29	91.03	26.05	99.99	99.55	93.71	20.26
LLR	100.00	99.43	92.14	30.44	100.00	99.57	95.13	9.75
AT	99.98	99.31	97.21	8.84	99.98	99.48	98.03	6.09
RST(λ=.5)	100.00	99.34	96.53	11.09	100.00	99.53	97.72	8.27
RST(λ=1)	100.00	99.31	96.96	11.31	100.00	99.55	98.27	6.26
RST(λ=2)	100.00	99.31	97.09	12.39	100.00	99.56	98.48	4.55
TRADES(β=1)	99.81	99.26	96.60	9.69	99.96	99.58	98.10	4.74
TRADES(β=3)	99.21	98.96	96.66	7.83	99.80	99.57	98.54	2.14
TRADES(β=6)	97.50	97.54	93.68	2.87	99.61	99.59	98.73	1.36

	CIFAR-10				Restricted ImageNet			
	train acc.	test acc.	adv test acc.	test lipschitz	train acc.	test acc.	adv test acc.	test lipschitz
Natural	100.00	93.81	0.00	425.71	97.72	93.47	7.89	32228.51
GR	94.90	80.74	21.32	28.53	91.12	88.51	62.14	886.75
LLR	100.00	91.44	22.05	94.68	98.76	93.44	52.62	4795.66
RST(λ=.5)	99.90	85.11	39.58	20.67	96.08	92.02	79.24	451.57
RST(λ=1)	99.86	84.61	40.89	23.15	95.66	92.06	79.69	355.43
RST(λ=2)	99.73	83.87	41.75	23.80	96.02	91.14	81.41	394.40
AT	99.84	83.51	43.51	26.23	96.22	90.33	82.25	287.97
TRADES(β=1)	99.76	84.96	43.66	28.01	97.39	92.27	79.90	2144.66
TRADES(β=3)	99.78	85.55	46.63	22.42	95.74	90.75	82.28	396.67
TRADES(β=6)	98.93	84.46	48.58	13.05	93.34	88.92	82.13	200.90

Experiments

- How well do classifiers produced by existing training methods generalize?

1. Locally Lipschitz classifiers, AT, TRADES and RST, also have large generalization gaps.
2. RST has better test accuracy than AT, it continues to have a large generalization gap. This generalization behavior is unlike linear classification, where imposing local Lipschitzness leads to higher margin and better generalization.
3. Imposing local Lipschitzness in these methods, appears to hurt generalization instead of helping. This suggests that these robust training methods may not be generalizing properly.

architecture	CNN1					CNN2				
	train acc.	test acc.	adv test acc.	gap	adv gap	train acc.	test acc.	adv test acc.	gap	adv gap
Natural	100.00	99.20	59.83	0.80	0.45	100.00	99.51	86.01	0.49	-0.28
GR	99.99	99.29	91.03	0.70	3.49	99.99	99.55	93.71	0.44	2.55
LLR	100.00	99.43	92.14	0.57	4.42	100.00	99.57	95.13	0.43	2.28
AT	99.98	99.31	97.21	0.67	2.67	99.98	99.48	98.03	0.50	1.92
RST(λ=.5)	100.00	99.34	96.53	0.66	3.16	100.00	99.53	97.72	0.47	2.27
RST(λ=1)	100.00	99.31	96.96	0.69	2.95	100.00	99.55	98.27	0.45	1.73
RST(λ=2)	100.00	99.31	97.09	0.69	2.87	100.00	99.56	98.48	0.44	1.52
TRADES(β=1)	99.81	99.26	96.60	0.55	2.10	99.96	99.58	98.10	0.38	1.70
TRADES(β=3)	99.21	98.96	96.66	0.25	1.33	99.80	99.57	98.54	0.23	1.18
TRADES(β=6)	97.50	97.54	93.68	-0.04	0.37	99.61	99.59	98.73	0.02	0.80

	CIFAR-10					Restricted ImageNet				
	train acc.	test acc.	adv test acc.	gap	adv gap	train acc.	test acc.	adv test acc.	gap	adv gap
Natural	100.00	93.81	0.00	6.19	0.00	97.72	93.47	7.89	4.25	-0.46
GR	94.90	80.74	21.32	14.16	3.94	91.12	88.51	62.14	2.61	0.19
LLR	100.00	91.44	22.05	8.56	4.50	98.76	93.44	52.62	5.32	0.22
RST(λ=.5)	99.90	85.11	39.58	14.79	36.26	96.08	92.02	79.24	4.06	4.57
RST(λ=1)	99.86	84.61	40.89	15.25	41.31	95.66	92.06	79.69	3.61	4.67
RST(λ=2)	99.73	83.87	41.75	15.86	43.54	96.02	91.14	81.41	4.87	6.19
AT	99.84	83.51	43.51	16.33	49.94	96.22	90.33	82.25	5.90	8.23
TRADES(β=1)	99.76	84.96	43.66	14.80	44.60	97.39	92.27	79.90	5.13	6.66
TRADES(β=3)	99.78	85.55	46.63	14.23	47.67	95.74	90.75	82.28	5.00	6.41
TRADES(β=6)	98.93	84.46	48.58	14.47	42.65	93.34	88.92	82.13	4.42	5.31

Experiments

- Can we improve the generalization gap of these models?

1. Dropout helps to narrow the generalization gap between training and test acc.
2. Dropout makes the model smoother (smaller Lipschitzness).
3. Combining dropout with the robust methods may be a good strategy for generalization.

		SVHN					CIFAR-10				
	dropout	test acc.	adv test acc.	test lipschitz	gap	adv gap	test acc.	adv test acc.	test lipschitz	gap	adv gap
Natural	False	95.85	2.66	149.82	4.15	0.87	93.81	0.00	425.71	6.19	0.00
Natural	True	96.66	1.52	152.38	3.34	1.22	93.87	0.00	384.48	6.13	0.00
AT	False	91.68	54.17	16.51	5.11	25.74	83.51	43.51	26.23	16.33	49.94
AT	True	93.05	57.90	11.68	-0.14	6.48	85.20	43.07	31.59	14.51	44.05
RST(λ=2)	False	92.39	51.39	23.17	6.86	36.02	83.87	41.75	23.80	15.86	43.54
RST(λ=2)	True	95.19	55.22	17.59	1.90	11.30	85.49	40.24	34.45	14.00	33.07
TRADES(β=3)	False	91.85	54.37	10.15	7.48	33.33	85.55	46.63	22.42	14.23	47.67
TRADES(β=3)	True	94.00	62.41	4.99	0.48	7.91	86.43	49.01	14.69	12.59	35.03
TRADES(β=6)	False	91.83	58.12	5.20	5.35	23.88	84.46	48.58	13.05	14.47	42.65
TRADES(β=6)	True	93.46	63.24	3.30	0.45	5.97	84.69	52.32	8.13	11.91	26.49

Experiments – Robust models

- Gradient Regularization (GR): ($d = \frac{\nabla f(x)}{|\nabla f(x)|_2}$)

$$\min_f \mathbb{E}\Big\{\mathcal{L}(f(\mathbf{X}), Y) + \beta\|\nabla_{\mathbf{X}}\mathcal{L}(f(\mathbf{X}), Y)\|_2^2\Big\}.$$

$$\|\nabla f(\mathbf{X})\|_2^2 \approx \left(\frac{\mathcal{L}(f(\mathbf{X}+hd), Y) - \mathcal{L}(f(\mathbf{X}), Y)}{h}\right)$$

- Locally-Linear Regularization model (LLR):

$$g(f, \delta, \mathbf{X}) = |\mathcal{L}(f(\mathbf{X}+\delta), Y) - \mathcal{L}(f(\mathbf{X}), Y) - \delta^T\nabla_{\mathbf{X}}\mathcal{L}(f(\mathbf{X}), Y)|$$

Define $\gamma(\varepsilon, \mathbf{X}) = \mathbb{E}\Big\{\max_{\delta\in B(\mathbf{X},\varepsilon)} g(f, \delta, \mathbf{X})\Big\}$ and also $\delta_{LLR} = \mathbb{E}\Big\{\operatorname{argmax}_{\delta\in B(\mathbf{X},\varepsilon)} g(f, \delta, \mathbf{X})\}$.
The loss function for Locally-Linear Regularization (LLR) model is

$$\mathbb{E}\Big\{\mathcal{L}(f(\mathbf{X}), Y) + \lambda\gamma(\varepsilon, \mathbf{X}) + \mu\|\delta_{LLR}^T\nabla_{\mathbf{X}}\mathcal{L}(f(\mathbf{X}), Y)\|\Big\}$$

Experiments – Robust models

LEHIGH UNIVERSITY

- Adversarial training (AT):

$$\min_f \mathbb{E}\Big\{\max_{\mathbf{X}'\in\mathbb{B}(\mathbf{X},\varepsilon)} \mathcal{L}(f(\mathbf{X}'),Y)\Big\}.$$

- Robust self-training (RST):

$$\min_f \mathbb{E}\Big\{\mathcal{L}(f(\mathbf{X}),Y) + \beta \max_{\mathbf{X}'\in\mathbb{B}(\mathbf{X},\varepsilon)} \mathcal{L}(f(\mathbf{X}'),Y)\Big\}.$$

- Locally-Lipschitz models (TRADES):

$$\min_f \mathbb{E}\Big\{\mathcal{L}(f(\mathbf{X}),Y) + \beta \max_{\mathbf{X}'\in\mathbb{B}(\mathbf{X},\varepsilon)} \mathcal{L}(f(\mathbf{X}),f(\mathbf{X}'))\Big\},$$

Thank you

LEHIGH
UNIVERSITY