On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training

Chen Liu[1], Zhichao Huang[2], Mathieu Salzmann[1], Tong Zhang[2], Sabine Süsstrunk[1]

[1] École Polytechnique Fédérale de Lausanne
[2] Hong Kong University of Science and Technology

December 20, 2021

Overview

1. Introduction
2. Empirical Investigation
3. Theoretical Analysis
4. Case Study
5. Summary

Content

1. Introduction
2. Empirical Investigation
3. Theoretical Analysis
4. Case Study
5. Summary

Definition (Robustness Problem)

Given a classification model $f(\theta, \mathbf{x}) : \Theta \times \mathbb{R}^H \rightarrow \mathbb{R}^K$ parameterized by θ, data points drawn from the distribution $(\mathbf{x}, y) \sim \mathcal{D}$ and loss function $\mathcal{L}$, robustness problem is formulation as follows:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \max_{\mathbf{x}' \in \mathcal{S}_\epsilon(\mathbf{x})} \mathcal{L}(f(\theta, \mathbf{x}'), y) \tag{1}$$

where $\mathcal{S}_\epsilon(\mathbf{x})$ is called the adversarial budget: $\mathcal{S}_\epsilon(\mathbf{x}) = \{\mathbf{x}' | \|\mathbf{x} - \mathbf{x}'\|_\infty \leq \epsilon\}$.

Adversarial Training: generate optimal x' and then optimize θ on x'.

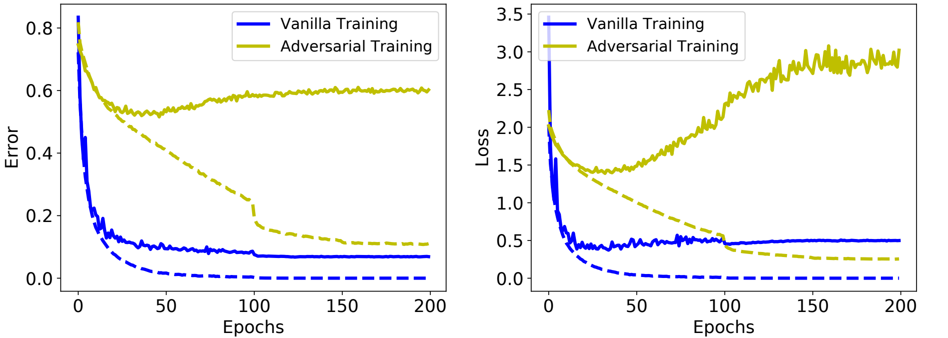


Figure: The training (dashed line) and test (solid line) curve in error (left) and loss (right). The model is ResNet18; the dataset is CIFAR10.

Introduction

- Adversarial overfitting is **universal**: it happens in all kinds of adversarial budgets, datasets and model architectures!

Introduction

- Adversarial overfitting is **universal**: it happens in all kinds of adversarial budgets, datasets and model architectures!
- There are several works mitigating adversarial overfitting: some are still valid, some are proven invalid by adaptive attacks.

Introduction

- Adversarial overfitting is **universal**: it happens in all kinds of adversarial budgets, datasets and model architectures!
- There are several works mitigating adversarial overfitting: some are still valid, some are proven invalid by adaptive attacks.
- The reason behind adversarial overfitting is still poorly understood.

Introduction

- Adversarial overfitting is **universal**: it happens in all kinds of adversarial budgets, datasets and model architectures!
- There are several works mitigating adversarial overfitting: some are still valid, some are proven invalid by adaptive attacks.
- The reason behind adversarial overfitting is still poorly understood.
- We mainly study this phenomenon from the aspect of **training instances.**

Content

1 Introduction

2 Empirical Investigation

3 Theoretical Analysis

4 Case Study

5 Summary

- Use the average loss as the basis of the metric.
- Given the dataset $\mathcal{D}$, the instance $\boldsymbol{x}$ and its average loss $\bar{\mathcal{L}}(\boldsymbol{x})$, the difficulty metric is defined as:

$$d(\boldsymbol{x}) = \mathbb{P}(\overline{\mathcal{L}}(\boldsymbol{x}) < \overline{\mathcal{L}}(\widetilde{\boldsymbol{x}}) | \widetilde{\boldsymbol{x}} \sim U(\mathcal{D})) + \frac{1}{2}\mathbb{P}(\overline{\mathcal{L}}(\boldsymbol{x}) = \overline{\mathcal{L}}(\widetilde{\boldsymbol{x}}) | \widetilde{\boldsymbol{x}} \sim U(\mathcal{D})) \ , \tag{2}$$

- Use the average loss as the basis of the metric.
- Given the dataset $\mathcal{D}$, the instance $\boldsymbol{x}$ and its average loss $\bar{\mathcal{L}}(\boldsymbol{x})$, the difficulty metric is defined as:

$$d(\boldsymbol{x}) = \mathbb{P}(\overline{\mathcal{L}}(\boldsymbol{x}) < \overline{\mathcal{L}}(\widetilde{\boldsymbol{x}}) | \widetilde{\boldsymbol{x}} \sim U(\mathcal{D})) + \frac{1}{2}\mathbb{P}(\overline{\mathcal{L}}(\boldsymbol{x}) = \overline{\mathcal{L}}(\widetilde{\boldsymbol{x}}) | \widetilde{\boldsymbol{x}} \sim U(\mathcal{D})) \ , \tag{2}$$

- It is a normalized metric: 0 for the hardest and 1 for the easiest.
- Empirically, it mainly depends on the data itself and the perturbation applied. Training algorithm and model architecture hardly change the difficulty value.

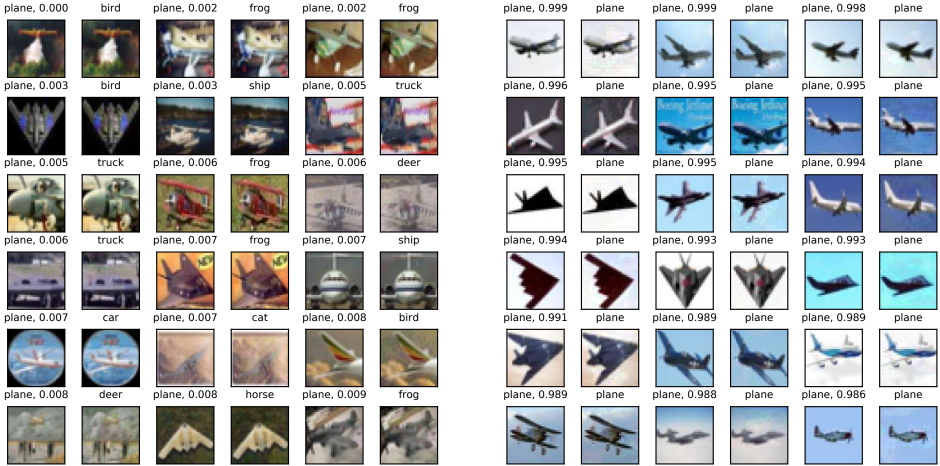


Figure: Hard (left) and easy (right) examples in CIFAR10.

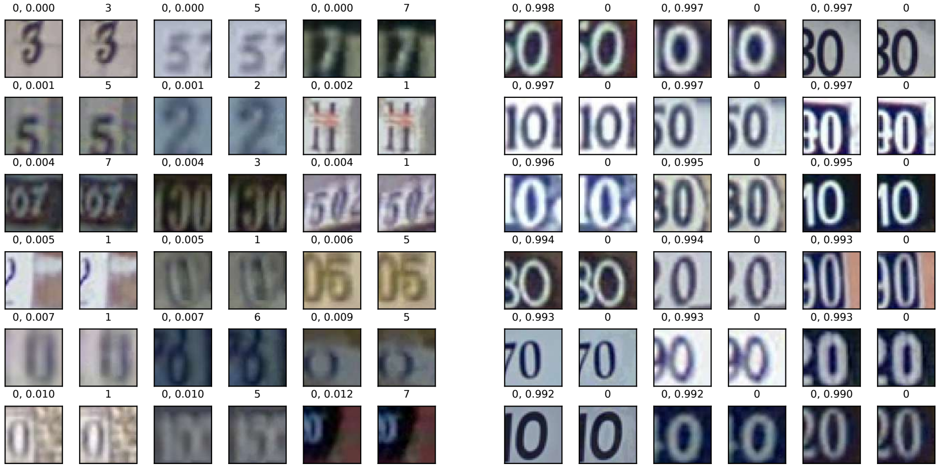


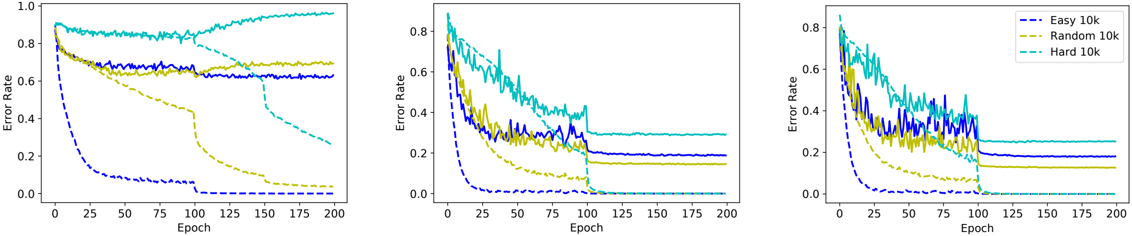
Figure: Hard (left) and easy (right) examples in SVHN.

We divide the training set into ten non-overlapping groups: $\{\mathcal{G}_i\}_{i=0}^{9}$ where $\mathcal{G}_i = \{\boldsymbol{x} \in \mathcal{D} | 0.1 \times i \leq d(\boldsymbol{x}) < 0.1 \times (i+1)\}$.

Empirical Investigation

Training on a Subset

We divide the training set into ten non-overlapping groups: $\{\mathcal{G}_i\}_{i=0}^{9}$ where $\mathcal{G}_i = \{\boldsymbol{x} \in \mathcal{D} | 0.1 \times i \leq d(\boldsymbol{x}) < 0.1 \times (i+1)\}$.



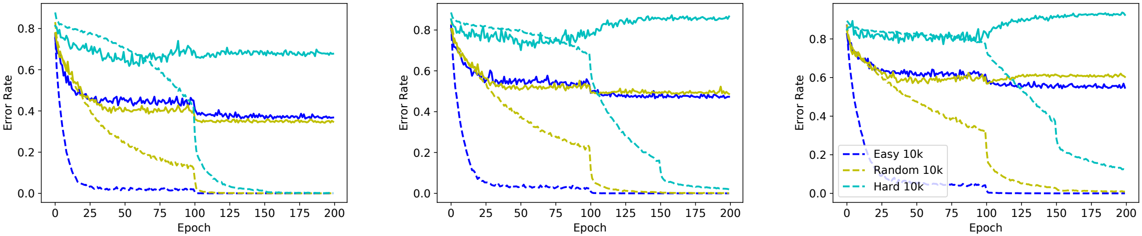
(a) PGD Adversarial Training (b) FGSM Adversarial Training (c) Vanilla Training

Figure: Learning curves obtained by training on the 10000 easiest, random and hardest instances of CIFAR10 under different scenarios. The training error (dashed lines) is the error on the selected instances, and the test error (solid lines) is the error on the whole test set.

Empirical Investigation

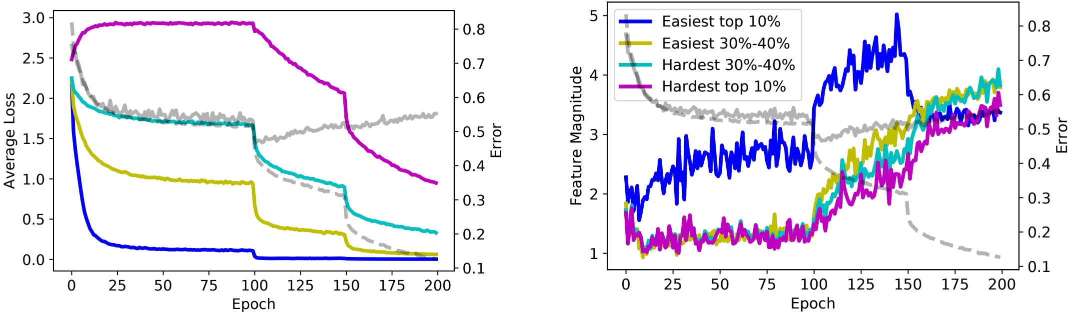
Training on a Subset

We divide the training set into ten non-overlapping groups: $\{\mathcal{G}_i\}_{i=0}^{9}$ where $\mathcal{G}_i = \{\boldsymbol{x} \in \mathcal{D} | 0.1 \times i \leq d(\boldsymbol{x}) < 0.1 \times (i+1)\}$.



(a) $\epsilon = 2/255$ (b) $\epsilon = 4/255$ (c) $\epsilon = 6/255$

Figure: Learning curves of training on PGD-perturbed inputs against different sizes of l_∞ norm based adversarial budgets using the easiest, the random and the hardest 10000 training instances. The instance difficulty is determined by the corresponding adversarial budget and is thus different under different adversarial budgets. The dashed lines are robust training error on the selected training set, the solid lines are robust test error on the entire test set.



(a) Average loss.

(b) Average l_2 norm of extracted features.

Figure: Analysis on the groups $\mathcal{G}_0$, $\mathcal{G}_3$, $\mathcal{G}_6$ and $\mathcal{G}_9$ in the training set. The right vertical axis corresponds to the training (dashed grey line) and test (solid grey line) error under adversarial attacks for both plots. **Left plot:** The left vertical axis represents the average loss of different groups. **Right plot:** The left vertical axis represents the average l_2 norm of features extracted during training for different groups.

- Harder the training data is, larger the generalization gap is.
- The gap between models trained by easy and hard data increases with the adversarial budget.
- In the early phase of training, the model tends to fit easy training instances; in the late phase of training, the model fits harder and harder training instances, when adversarial overfitting happens.

The above phenomenon always happens for different dataset, adversarial budget (l_∞, l_2) and model architectures.

Content

1. Introduction
2. Empirical Investigation
3. Theoretical Analysis
4. Case Study
5. Summary

We use $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ to represent the m-dimensional training data, and $(\mathbf{X}, \boldsymbol{y})$ as its matrix form.

$\{\boldsymbol{x}_i', y_i\}_{i=1}^n$ and $(\mathbf{X}', \boldsymbol{y})$ are their adversarial counterparts.

Here, $\boldsymbol{x}_i \in \mathbb{R}^m$, $y_i \in \{-1, +1\}$, $\mathbf{X} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{y} \in \{-1, +1\}^n$.

Model: A linear model parameterized by $\boldsymbol{w} \in \mathbb{R}^m$, it outputs $sign(\boldsymbol{w}^T \boldsymbol{x})$ given the input $\boldsymbol{x}$.

Model: A linear model parameterized by $\boldsymbol{w} \in \mathbb{R}^m$, it outputs $sign(\boldsymbol{w}^T \boldsymbol{x})$ given the input $\boldsymbol{x}$.

Data: A Gaussian mixture model with K-mode components. Specifically, the k-th component has a probability p_k of being sampled and is formulated as follows:

$$\text{if } y_i = +1,\ \boldsymbol{x}_i \sim \mathcal{N}(r_k \boldsymbol{\eta}, \mathbf{I});\ \text{if } y_i = -1,\ \boldsymbol{x}_i \sim \mathcal{N}(-r_k \boldsymbol{\eta}, \mathbf{I}). \tag{3}$$

Without the loss of generality, $r_1 < r_2 < ... < r_{K-1} < r_K$. Therefore, the first component is the hardest one while the $K - th$ one is the easiest one.

Theorem

If a logistic regression model is adversarially trained on n separable training instances sampled from the l-th component of the GMM model described in (3). If $\frac{m}{n \log n}$ is sufficiently large[a], then with probability $1 - O(\frac{1}{n})$, the expected adversarial test error $\mathcal{R}$ under the adversarial budget $\mathcal{S}^{(2)}(\epsilon)$, which is a function of r_l and ϵ, on the whole GMM model described in (3) is given by

$$\mathcal{R}(r_l, \epsilon) = \sum_{k=1}^{K} p_k \Phi\left(r_k g(r_l) - \epsilon\right), \; g(r_l) = (C_1 - \frac{1}{C_2 r_l^2 + o(r_l^2)})^{\frac{1}{2}}, \; C_1, C_2 \geq 0. \tag{4}$$

C_1, C_2 are independent of ϵ and r_l. The function Φ is defined as $\Phi(x) = \mathbb{P}(Z > x)$, $Z \sim \mathcal{N}(0, 1)$.

[a] Specifically, m and n need to satisfy $m > 10n \log n + n - 1$ and $m > Cnr_l\sqrt{\log 2n}\|\boldsymbol{\eta}\|$. The constant C is derived in the proof of Theorem 1 in [3].

Theoretical Analysis

A Toy Example: Logistic Regression fit Gaussian Mixture Model

$$\mathcal{R}(r_I, \epsilon) = \sum_{k=1}^{K} p_k \Phi\left(r_k g(r_I) - \epsilon\right), \; g(r_I) = \left(C_1 - \frac{1}{C_2 r_I^2 + o(r_I^2)}\right)^{\frac{1}{2}}, \; C_1, C_2 \geq 0.$$

$\mathcal{R}(r_I, \epsilon)$ increases with the decrease of r_I, indicating hard adversarial training instances lead to larger generalization gap.

$$\mathcal{R}(r_l, \epsilon) = \sum_{k=1}^{K} p_k \Phi\left(r_k g(r_l) - \epsilon\right), \; g(r_l) = (C_1 - \frac{1}{C_2 r_l^2 + o(r_l^2)})^{\frac{1}{2}}, \; C_1, C_2 \geq 0.$$

$\mathcal{R}(r_l, \epsilon)$ increases with the decrease of r_l, indicating hard adversarial training instances lead to larger generalization gap.

Corollary

Under the conditions of the previous theorem and the definition of $\mathcal{R}$ in Equation (4), if $\epsilon_1 < \epsilon_2$, then we have $\forall\ 0 \leq i < j \leq K, \mathcal{R}(r_i, \epsilon_1) - \mathcal{R}(r_j, \epsilon_1) < \mathcal{R}(r_i, \epsilon_2) - \mathcal{R}(r_j, \epsilon_2)$.

The gap in performance between the models trained by the easy and hard instances increases with the size of the adversarial budget ϵ. **Adversarial training is more sensitive to the instance difficulty.**

Model: A general nonlinear model parameterized by $\boldsymbol{w} \in \mathbb{R}^b$, it outputs $sign(f(\boldsymbol{w}, \boldsymbol{x}))$ where f represents a neural network.

Theoretical Analysis

General Nonlinear Model

Model: A general nonlinear model parameterized by $\boldsymbol{w} \in \mathbb{R}^b$, it outputs $sign(f(\boldsymbol{w}, \boldsymbol{x}))$ where f represents a neural network.

Data: The dataa distribution is a mixture of K c-isoperimetric components.

Assumption

The data distribution μ is a composition of K c-isoperimetric distributions on $\mathbb{R}^m$, each of which has a positive conditional variance. That is, $\mu = \sum_{k=1}^{K} \alpha_k \mu_k$, where $\alpha_k > 0$ and $\sum_{k=1}^{K} \alpha_k = 1$. We define $\sigma_k^2 = \mathbb{E}_{\mu_k}[Var[y|\boldsymbol{x}]]$, and without loss of generality assume that $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_K > 0$. Furthermore, given any L-Lipschitz function $f_{\boldsymbol{w}}$, i.e., $\forall \boldsymbol{x}_1, \boldsymbol{x}_2, \|f_{\boldsymbol{w}}(\boldsymbol{x}_1) - f_{\boldsymbol{w}}(\boldsymbol{x}_2)\| \leq L\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$, we have

$$\forall k \in \{1, 2, ..., K\} \; \mathbb{P}(\boldsymbol{x} \sim \mu_k, \|f_{\boldsymbol{w}}(\boldsymbol{x}) - \mathbb{E}_{\mu_k}(f_{\boldsymbol{w}})\| \geq t) \leq 2e^{-\frac{mt^2}{2cL^2}} \; . \tag{5}$$

Definition

Given the dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, the model $f_{\boldsymbol{w}}$, the adversarial budget $\mathcal{S}^{(p)}(\epsilon)$ and a positive constant C, we define the function $h(C, \epsilon)$ as

$$h(C,\epsilon) = \min_{\boldsymbol{w}\in\mathcal{T}(C,\epsilon)} \min_i h_{i,\boldsymbol{w}}(\epsilon) \ \ s.t.\ \mathcal{T}(C,\epsilon) = \left\{ \boldsymbol{w} \middle| \frac{1}{n}\sum_{i=1}^{n}(f_{\boldsymbol{w}}(\boldsymbol{x}_i') - y_i)^2 \le C \right\}, \qquad (6)$$
$$\text{where } h_{i,\boldsymbol{w}}(\epsilon) = \max \zeta,\ \ s.t.\ [f_{\boldsymbol{w}}(\boldsymbol{x}_i) - \zeta, f_{\boldsymbol{w}}(\boldsymbol{x}_i) + \zeta] \subset \left\{ f_{\boldsymbol{w}}(\boldsymbol{x}_i + \Delta) \middle| \Delta \in \mathcal{S}^{(p)}(\epsilon) \right\}.$$

Here, $\boldsymbol{x}_i'$ is the adversarial example of $\boldsymbol{x}$. We omit the superscript (p) for notation simplicity.

Definition

Given the dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, the model $f_{\boldsymbol{w}}$, the adversarial budget $\mathcal{S}^{(p)}(\epsilon)$ and a positive constant C, we define the function $h(C,\epsilon)$ as

$$h(C,\epsilon) = \min_{\boldsymbol{w}\in\mathcal{T}(C,\epsilon)} \min_i h_{i,\boldsymbol{w}}(\epsilon) \;\; s.t. \; \mathcal{T}(C,\epsilon) = \left\{ \boldsymbol{w} \Big| \frac{1}{n}\sum_{i=1}^n (f_{\boldsymbol{w}}(\boldsymbol{x}_i') - y_i)^2 \leq C \right\}, \qquad (6)$$

$$\text{where } h_{i,\boldsymbol{w}}(\epsilon) = \max \zeta, \;\; s.t. \; [f_{\boldsymbol{w}}(\boldsymbol{x}_i) - \zeta, f_{\boldsymbol{w}}(\boldsymbol{x}_i) + \zeta] \subset \left\{ f_{\boldsymbol{w}}(\boldsymbol{x}_i + \Delta) | \Delta \in \mathcal{S}^{(p)}(\epsilon) \right\}.$$

Here, $\boldsymbol{x}_i'$ is the adversarial example of $\boldsymbol{x}$. We omit the superscript (p) for notation simplicity.

$h(C,\epsilon)$ monotonically increases with the increase of ϵ but with the decrease of C.

We use the **Lipschitz constant** as the proxy to measure the generalization performance.

$$\forall \boldsymbol{x}_1, \boldsymbol{x}_2, \|f(\boldsymbol{w}, \boldsymbol{x}_1) - f(\boldsymbol{w}, \boldsymbol{x}_2)\| \leq L\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$$

Theorem

Given n training pairs $\{\mathbf{x}_i, y_i\}_{i=1}^n$ *sampled from the l-th component* μ_l *of the distribution in Assumption, the parametric model* $f_{\mathbf{w}}$*, the adversarial budget* $\mathcal{S}^{(p)}(\epsilon)$ *and the corresponding function h defined in Definition, we assume that the model* $f_{\mathbf{w}}$ *is in the function space* $\mathcal{F} = \{f_{\mathbf{w}}, \mathbf{w} \in \mathcal{W}\}$ *with* $\mathcal{W} \subset \mathbb{R}^b$ *having a finite diameter* $diam(\mathcal{W}) \leq W$ *and,* $\forall \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{W}, \|f_{\mathbf{w}_1} - f_{\mathbf{w}_2}\|_\infty \leq J\|\mathbf{w}_1 - \mathbf{w}_2\|_\infty$. *We train the model* $f_{\mathbf{w}}$ *adversarially using these n data points. Let* $\mathbf{x}'$ *be the adversarial example of the data point* $\mathbf{x}$ *and* $\delta \in (0,1)$. *If we have* $\frac{1}{n}\sum_{i=1}^n (f_{\mathbf{w}}(\mathbf{x}_i') - y_i)^2 = C$ *and* $\gamma := \sigma_l^2 + h^2(C, \epsilon) - C \geq 0$, *then with probability at least* $1 - \delta$, *the Lipschitz constant of* $f_{\mathbf{w}}$ *is lower bounded as*

$$Lip(f_{\mathbf{w}}) \geq \frac{\gamma}{2^7}\sqrt{\frac{nm}{c\left(b\log(4WJ\gamma^{-1}) - \log\left(\delta/2 - 2e^{-2^{-11}n\gamma^2}\right)\right)}}, \tag{7}$$

where $Lip(f_{\mathbf{w}})$ *is the Lipschitz constant of* $f_{\mathbf{w}}$: $\forall \mathbf{x}_1, \mathbf{x}_2, \|f_{\mathbf{w}}(\mathbf{x}_1) - f_{\mathbf{w}}(\mathbf{x}_2)\| \leq Lip(f_{\mathbf{w}})\|\mathbf{x}_1 - \mathbf{x}_2\|$.

Theorem (Informal)

... If we have $\frac{1}{n}\sum_{i=1}^{n}(f_{\mathbf{w}}(\boldsymbol{x}_i') - y_i)^2 = C$ *and* $\gamma := \sigma_l^2 + h^2(C, \epsilon) - C \geq 0$, *then with high probability, the Lipschitz constant of* $f_{\mathbf{w}}$ *is lower bounded as*

$$Lip(f_{\mathbf{w}}) \gtrsim \frac{\gamma}{2^7}\sqrt{\frac{nm}{bc\log(4WJ\gamma^{-1})}} \tag{8}$$

Theoretical Analysis

General Nonlinear Model

Theorem (Informal)

... If we have $\frac{1}{n}\sum_{i=1}^{n}(f_{\mathbf{w}}(\mathbf{x}_i') - y_i)^2 = C$ *and* $\gamma := \sigma_l^2 + h^2(C, \epsilon) - C \geq 0$*, then with high probability, the Lipschitz constant of* $f_{\mathbf{w}}$ *is lower bounded as*

$$Lip(f_{\mathbf{w}}) \gtrsim \frac{\gamma}{2^7}\sqrt{\frac{nm}{bc\log(4WJ\gamma^{-1})}} \tag{8}$$

- Training progresses: $C \downarrow$, then $\gamma \uparrow$, then $Lip(f_{\mathbf{w}}) \uparrow$, generalization gap $\uparrow$.
- Training with hard instances: $\sigma_l \uparrow$, then $\gamma \uparrow$, then $Lip(f_{\mathbf{w}}) \uparrow$, generalization gap $\uparrow$.
- Training with larger adversarial budget $\epsilon \uparrow$, then $\gamma \uparrow$, then $Lip(f_{\mathbf{w}}) \uparrow$, generalization gap $\uparrow$.

Theoretical Analysis

General Nonlinear Model

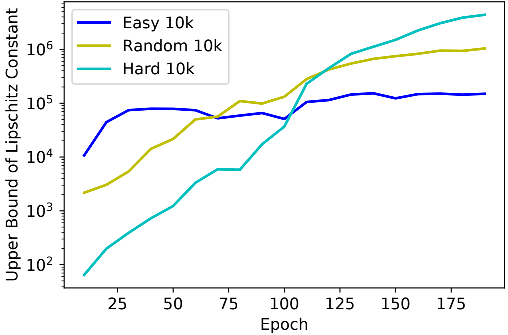


Figure: The curves of the Lipschitz upper bound when the model is adversarially trained by the easiest, the random and the hardest 10000 instances. The y-axis is log-scale.

Content

1. Introduction
2. Empirical Investigation
3. Theoretical Analysis
4. Case Study
5. Summary

Existing methods successfully mitigating adversarial overfitting all avoid fitting hard input-pairs.

Existing methods successfully mitigating adversarial overfitting all avoid fitting hard input-pairs.

- Instance-Adaptive Training [1]: assign different ϵ values to different training instances.
    - Assign smaller ϵ to training instances $\boldsymbol{x}$ whose $d(\boldsymbol{x})$ values are small.

Existing methods successfully mitigating adversarial overfitting all avoid fitting hard input-pairs.

- Instance-Adaptive Training [1]: assign different ϵ values to different training instances.
  - Assign smaller ϵ to training instances $\boldsymbol{x}$ whose $d(\boldsymbol{x})$ values are small.
- Self-Adaptive Training [2]: generate adaptive target instead of using one-hot label.
  - For easy instnaces, the adaptive target is very close to one-hot target; for hard instances, the difference between these two values is huge.

Existing methods successfully mitigating adversarial overfitting all avoid fitting hard input-pairs.

- Instance-Adaptive Training [1]: assign different ϵ values to different training instances.
  - Assign smaller ϵ to training instances $\boldsymbol{x}$ whose $d(\boldsymbol{x})$ values are small.
- Self-Adaptive Training [2]: generate adaptive target instead of using one-hot label.
  - For easy instnaces, the adaptive target is very close to one-hot target; for hard instances, the difference between these two values is huge.

Existing methods highlighting hard training adversarial instances are found invalid.

- Geometry-Aware Adversarial Training [4]: assign larger weights to training instances close to the decision boundary.
  - Proven invalid by adaptive attacks.

Content

1 Introduction

2 Empirical Investigation

3 Theoretical Analysis

4 Case Study

5 Summary

Summary

Take-aways:

- Hard instances leads to overfitting in adversarial training.
- Compared with vanilla training, adversarial training is more sensitive to hard instances.
- Methods mitigating adversarial overfitting avoids fitting adversarial input-target pairs. By contrast, methods highlighting hard instances may not achieve true robustness.

Summary

Some questions I am working / supervising on

- Training provably robust models.
- Robust compressed model.
- Robustness against multiple l_p norm based attacks.
- Robustness on deep equilibrium models, such as Neural ODE.

Summary

Some questions I am working / supervising on

- Training provably robust models.
- Robust compressed model.
- Robustness against multiple l_p norm based attacks.
- Robustness on deep equilibrium models, such as Neural ODE.

Some open questions I am interested in.

- Adversarial training with semi-supervised training.
- Optimization properties of training provably robust models.
- Fundamental reasons why adversarial examples exists for deep nonlinear models.

Thank You!

Yogesh Balaji, Tom Goldstein, and Judy Hoffman.
Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets.
arXiv preprint arXiv:1910.08051, 2019.

Lang Huang, Chao Zhang, and Hongyang Zhang.
Self-adaptive training: beyond empirical risk minimization.
Advances in Neural Information Processing Systems, 33, 2020.

Ke Wang and Christos Thrampoulidis.
Benign overfitting in binary classification of gaussian mixtures.
arXiv preprint arXiv:2011.09148, 2020.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli.
Geometry-aware instance-reweighted adversarial training.
In *International Conference on Learning Representations*, 2021.