CS8695 Research in Computer Science

Chen Liu

Understanding and Building Reliable Deep Neural Networks

Semester A, 2023-2024

Outline

Introduction

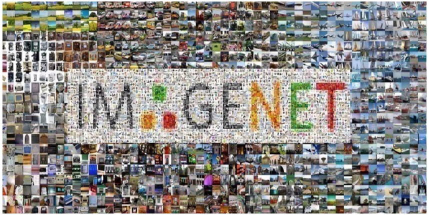
Verified Robustness

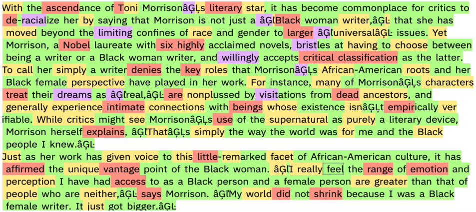
Empirical Robustness

Efficiency

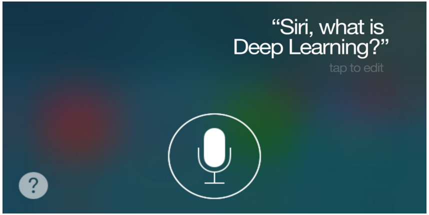
Benefits of Robustness

Broad Applications of AI





With the ascendance of Toni MorrisonâĢĻs literary star, it has become commonplace for critics to de-racialize her by saying that Morrison is not just a âĢľBlack woman writer,âĢĿ that she has moved beyond the limiting confines of race and gender to larger âĢľuniversalâĢĿ issues. Yet Morrison, a Nobel laureate with six highly acclaimed novels, bristles at having to choose between being a writer or a Black woman writer, and willingly accepts critical classification as the latter.
To call her simply a writer denies the key roles that MorrisonâĢĻs African-American roots and her Black female perspective have played in her work. For instance, many of MorrisonâĢĻs characters treat their dreams as âĢľreal,âĢĿ are nonplussed by visitations from dead ancestors, and generally experience intimate connections with beings whose existence isnâĢĻt empirically ver ifiable. While critics might see MorrisonâĢĻs use of the supernatural as purely a literary device, Morrison herself explains, âĢľThatâĢĻs simply the way the world was for me and the Black people I knew.âĢĿ
Just as her work has given voice to this little-remarked facet of African-American culture, it has affirmed the unique vantage point of the Black woman. âĢľI really feel the range of emotion and perception I have had access to as a Black person and a female person are greater than that of people who are neither,âĢĿ says Morrison. âĢľMy world did not shrink because I was a Black female writer. It just got bigger.âĢĿ



AI Comes with Risks

Including but not limited to:

- Wrong predictions with malicious input.
- Sensitive data or information leakage.
- Ethics violation.

AI Comes with Risks

Including but not limited to:

- Wrong predictions with malicious input.
- Sensitive data or information leakage.
- Ethics violation.

Especially when modern AI systems broadly deploy deep neural networks, which are hard to interpret and like black boxes.

AI Comes with Risks

Including but not limited to:

- Wrong predictions with malicious input.
- Sensitive data or information leakage.
- Ethics violation.

Especially when modern AI systems broadly deploy deep neural networks, which are hard to interpret and like black boxes.

Artificial intelligence is NOT human intelligence!

Failure Cases of AI

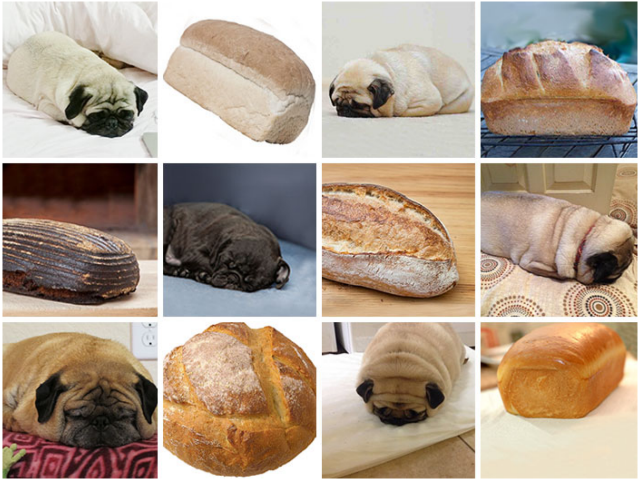
Failure Cases of AI

Original Text Prediction = **Negative**. (Confidence = 78.0%)
*This movie had **terrible** acting, **terrible** plot, and **terrible** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **considered** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **kids** they didn't understand that theme.*
Adversarial Text Prediction = **Positive**. (Confidence = 59.8%)
*This movie had **horrific** acting, **horrific** plot, and **horrifying** choice of actors. (Leslie Nielsen ...come on!!!) the one part I **regarded** slightly funny was the battling FBI/CIA agents, but because the audience was mainly **youngsters** they didn't understand that theme.*

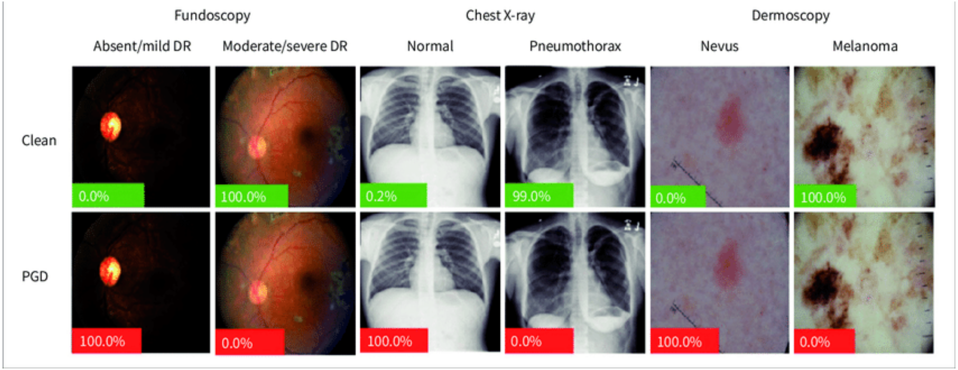
Table 1: Example of attack results for the sentiment analysis task. Modified words are highlighted in green and red for the original and adversarial texts, respectively.

Original Text Prediction: **Entailment** (Confidence = 86%)
Premise: *A runner wearing purple strives for the finish line.* **Hypothesis:** *A **runner** wants to head for the finish line.*
Adversarial Text Prediction: **Contradiction** (Confidence = 43%)
Premise: *A runner wearing purple strives for the finish line.* **Hypothesis:** *A **racer** wants to head for the finish line.*

Failure Cases of AI



Failure Cases of AI



Failure Cases of AI

Failure Cases of AI

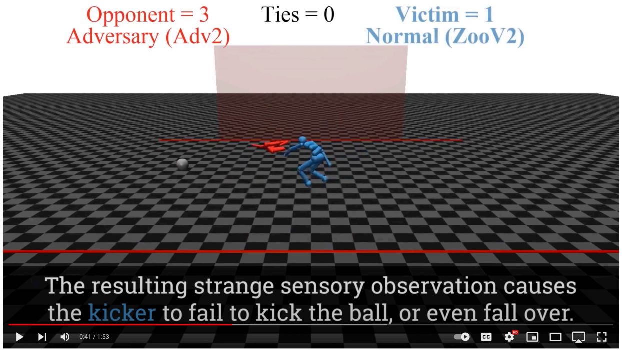


Figure: https://www.youtube.com/watch?v=XPFQ9TBvtCE

[1]

[1] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, S. Russell. "Adversarial Policies: Attacking Deep Reinforcement Learning". ICLR 2020.

Failure Cases of AI

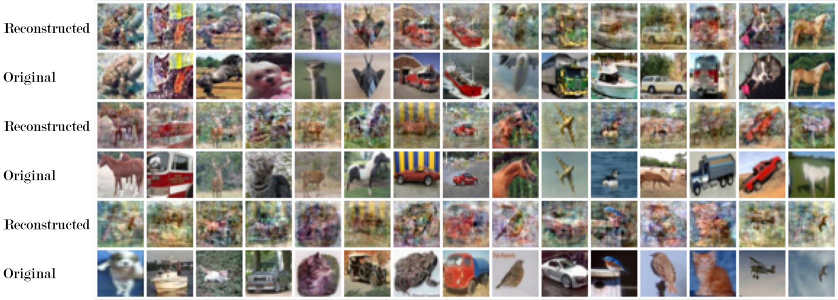
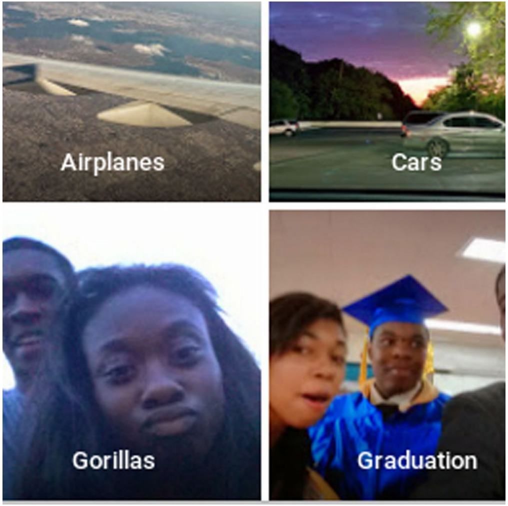


Figure: Dataset reconstruction. [1].

[1] N. Haim, G. Vardi, G. Yehudai, O. Shamir, M. Irani"Reconstructing Training Data from Trained Neural Networks". NeurIPS 2022.

Failure Cases of AI



Adversarial Examples

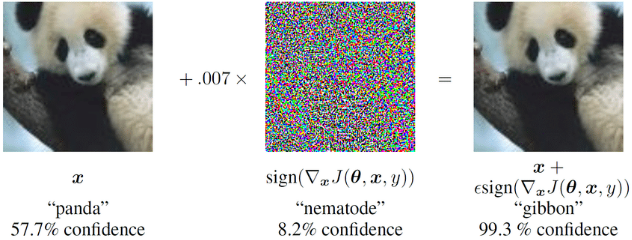
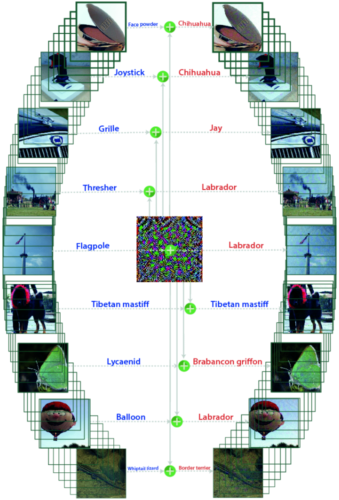


Figure: Image from pytorch.org.

For an AI model $f : \mathbb{R}^M \to \mathbb{R}^C$ which maps the M-dimensional input $\boldsymbol{x}$ to C categories, adversarial examples $\boldsymbol{x}'$ are the perturbed input that looks almost the same as $\boldsymbol{x}$, but $f(\boldsymbol{x})$ is quite different from $f(\boldsymbol{x}')$. Undefended neural network models can be easily broken by adversarial perturbations!

Adversarial Examples



- Adversarial perturbations can be universal!

Robust Learning Problem

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boxed{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Robust Learning Problem

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boxed{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta, \theta))}$$

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta, \theta))}$$

Robust Learning Problem

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boxed{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Adversarial attacks.
Adversarial training.

Robust Learning Problem

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boxed{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Adversarial attacks.
Adversarial training.

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \overline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Robust Learning Problem

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boxed{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Adversarial attacks.
Adversarial training.

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \overline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Robustness verification.
Training provably networks.

Robust Learning Problem

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boxed{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Empirical robustness.

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Adversarial attacks.
Adversarial training.

Verified robustness.

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \overline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Robustness verification.
Training provably networks.

Robust Learning Problem

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \boxed{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Empirical robustness.

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Adversarial attacks.
Adversarial training.

Verified robustness.

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \overline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Robustness verification.
Training provably networks.

Verified robust accuracy $\leq$ "True" robust accuracy $\leq$ Empirical robust accuracy

Outline

Introduction

Verified Robustness

Empirical Robustness

Efficiency

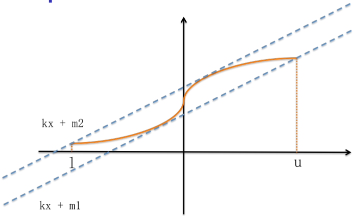
Benefits of Robustness

Linear Approximation of Deep Neural Networks

Motivation:

1. The decision boundary of deep neural network is complex and nonlinear.
2. The nonlinearity arises from the activation function.
3. Estimating the nonlinear activation function by linear functions can derive the lower and the upper bound of the network outputs.

Linear Approximation of Deep Neural Networks



- Given any nonlinear function $\sigma(\boldsymbol{x})$ with bounded input $\mathbf{l} \leq \boldsymbol{x} \leq \mathbf{u}$, we can introduce one diagonal matrix $\mathbf{D}$ and two vectors $\mathbf{m}_1$, $\mathbf{m}_2$:

$$\mathbf{D}\boldsymbol{x} + \mathbf{m}_1 \leq \sigma(\boldsymbol{x}) \leq \mathbf{D}\boldsymbol{x} + \mathbf{m}_2$$

- Equivalently, $\forall \boldsymbol{x} : \mathbf{l} \leq \boldsymbol{x} \leq \mathbf{u}$, we have $\mathbf{D}, \mathbf{m}_1, \mathbf{m}_2$ and $\exists \mathbf{m} : \mathbf{m}_1 \leq \mathbf{m} \leq \mathbf{m}_2$, such that

$$\sigma(\boldsymbol{x}) = \mathbf{D}\boldsymbol{x} + \mathbf{m}$$

C. Liu, R. Tomioka, V. Cevher. "On Certifying Non-uniform Bounds against Adversarial Attacks.". ICML 2019.

Linear Approximation of Deep Neural Networks

- Recall the N-layer neural network.

$$
\begin{aligned}
\boldsymbol{z}^{(i+1)} &= \mathbf{W}^{(i)}\widehat{\boldsymbol{z}}^{(i)} + \boldsymbol{b}^{(i)} \qquad i = 1, 2, ..., N-1 \\
\widehat{\boldsymbol{z}}^{(i)} &= \sigma(\boldsymbol{z}^{(i)}) \qquad\qquad\quad\;\; i = 2, 3, ..., N-1
\end{aligned}
\tag{1}
$$

Linear Approximation of Deep Neural Networks

- Recall the N-layer neural network.

$$\begin{aligned} \boldsymbol{z}^{(i+1)} &= \mathbf{W}^{(i)}\widehat{\boldsymbol{z}}^{(i)} + \boldsymbol{b}^{(i)} \qquad i = 1, 2, ..., N-1 \\ \widehat{\boldsymbol{z}}^{(i)} &= \sigma(\boldsymbol{z}^{(i)}) \qquad\qquad\quad\; i = 2, 3, ..., N-1 \end{aligned} \tag{1}$$

- We can linearize the output of each layer.

$$\begin{aligned} \boldsymbol{z}^{(i)} &= \mathbf{W}^{(i-1)}(\sigma(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x} + \mathbf{m}^{(1)}) + \boldsymbol{b}^{(1)})...) + \boldsymbol{b}^{i-2})) + \boldsymbol{b}^{(i-1)} \\ &= \mathbf{W}^{(i-1)}(\mathbf{D}^{(i-1)}(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x} + \mathbf{m}^{(1)}) + \boldsymbol{b}^{(1)})...) + \boldsymbol{b}^{(i-2)}) + \mathbf{m}^{(i-1)}) + \boldsymbol{b}^{(i-1)} \\ &= \left(\Pi_{j=2}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(1)}\mathbf{x} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\boldsymbol{b}^{(h)} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(h)}\mathbf{m}^{(h)} \end{aligned} \tag{2}$$

Linear Approximation of Deep Neural Networks

- Recall the N-layer neural network.

$$\begin{aligned} \boldsymbol{z}^{(i+1)} &= \mathbf{W}^{(i)}\widehat{\boldsymbol{z}}^{(i)} + \boldsymbol{b}^{(i)} && i = 1, 2, ..., N-1 \\ \widehat{\boldsymbol{z}}^{(i)} &= \sigma(\boldsymbol{z}^{(i)}) && i = 2, 3, ..., N-1 \end{aligned} \tag{1}$$

- We can linearize the output of each layer.

$$\begin{aligned} \boldsymbol{z}^{(i)} &= \mathbf{W}^{(i-1)}(\sigma(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x}+\mathbf{m}^{(1)}) + \boldsymbol{b}^{(1)})...) + \boldsymbol{b}^{i-2})) + \boldsymbol{b}^{(i-1)} \\ &= \mathbf{W}^{(i-1)}(\mathbf{D}^{(i-1)}(\mathbf{W}^{(i-2)}(...(\mathbf{W}^{(1)}(\mathbf{x}+\mathbf{m}^{(1)}) + \boldsymbol{b}^{(1)})...) + \boldsymbol{b}^{(i-2)}) + \mathbf{m}^{(i-1)}) + \boldsymbol{b}^{(i-1)} \\ &= \left(\Pi_{j=2}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(1)}\mathbf{x} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\boldsymbol{b}^{(h)} + \sum_{h=1}^{i-1}\left(\Pi_{j=h+1}^{i-1}\mathbf{W}^{(j)}\mathbf{D}^{(j)}\right)\mathbf{W}^{(h)}\mathbf{m}^{(h)} \end{aligned} \tag{2}$$

- Bound for $\{\mathbf{m}^{(h)}\}_{h=1}^{i-1}$ $\rightarrow$ bounds for $\boldsymbol{z}^{(i)}$ $\rightarrow$ bound for $\mathbf{m}^{(i)}$
- Iteratively estimate the bounds for $\{\boldsymbol{z}^{(i)}\}_{i=2}^{N}$

Linear Approximation of Deep Neural Networks

Corollary (Model Linearization)

Given a classification model $f(\boldsymbol{x}, \theta) : \mathbb{R}^H \times \Theta \to \mathbb{R}^K$ *parameterized by* θ*, a data point* $(\boldsymbol{x}, y)$ *and a pre-defined adversarial budget* $\mathcal{S}_\epsilon(\boldsymbol{x})$*,* $\exists \mathbf{W} \in \mathbb{R}^{H \times K}$*,* $\boldsymbol{b} \in \mathbb{R}^K$ *such that*

$$\forall \Delta \in \mathcal{S}_\epsilon, f(\boldsymbol{x} + \Delta, \theta) - f(\boldsymbol{x} + \Delta, \theta,)_y \leq \mathbf{W}\Delta + \boldsymbol{b} \tag{3}$$

Linear Approximation of Deep Neural Networks

Corollary (Model Linearization)

Given a classification model $f(\boldsymbol{x}, \theta) : \mathbb{R}^H \times \Theta \rightarrow \mathbb{R}^K$ parameterized by θ, a data point $(\boldsymbol{x}, y)$ and a pre-defined adversarial budget $\mathcal{S}_\epsilon(\boldsymbol{x})$, $\exists \mathbf{W} \in \mathbb{R}^{H \times K}$, $\boldsymbol{b} \in \mathbb{R}^K$ such that

$$\forall \Delta \in \mathcal{S}_\epsilon, f(\boldsymbol{x} + \Delta, \theta) - f(\boldsymbol{x} + \Delta, \theta,)_y \leq \mathbf{W}\Delta + \boldsymbol{b} \tag{3}$$

- If $\forall \Delta \in \mathcal{S}_\epsilon$, $\mathbf{W}\Delta + \boldsymbol{b} \leq 0$, then $f(\boldsymbol{x} + \Delta, \theta) - f(\boldsymbol{x} + \Delta, \theta,)_y \leq 0$, the model is guaranteed robust.

Geometric Intepretation of Model Linearization

- $\{\Delta | \mathbf{W}\Delta + \boldsymbol{b} \leq 0\}$ forms a polyhedron in $\mathbb{R}^H$ space and is an envelope of the model's decision boundary.

C. Liu, M. Salzmann, S. Süsstrunk. "Training Provably Robust Models by Polyhedral Envelope Regularization". TNNLS 2021.

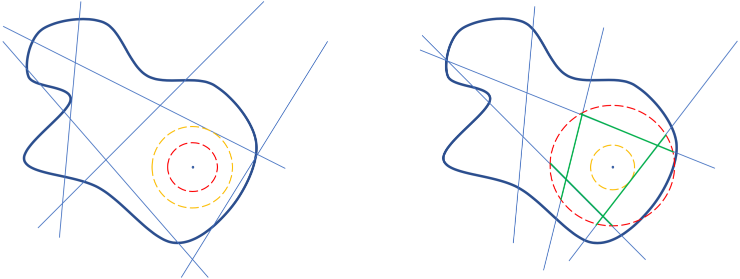
Geometric Intepretation of Model Linearization

- $\{\Delta | \mathbf{W}\Delta + \boldsymbol{b} \leq 0\}$ forms a polyhedron in $\mathbb{R}^H$ space and is an envelope of the model's decision boundary.
- If $\Delta \in \mathcal{S}_\epsilon \cap \{\Delta | \mathbf{W}\Delta + \boldsymbol{b} \leq 0\}$, then $\boldsymbol{x} + \Delta$ is guaranteed to have the same prediction as $\boldsymbol{x}$.

C. Liu, M. Salzmann, S. Süsstrunk. "Training Provably Robust Models by Polyhedral Envelope Regularization". TNNLS 2021.

Geometric Intepretation of Model Linearization

- $\{\Delta | \mathbf{W}\Delta + \boldsymbol{b} \leq 0\}$ forms a polyhedron in $\mathbb{R}^H$ space and is an envelope of the model's decision boundary.
- If $\Delta \in \mathcal{S}_\epsilon \cap \{\Delta | \mathbf{W}\Delta + \boldsymbol{b} \leq 0\}$, then $\boldsymbol{x} + \Delta$ is guaranteed to have the same prediction as $\boldsymbol{x}$.
- Geometric interpretation: when ϵ is too big or too small.



C. Liu, M. Salzmann, S. Süsstrunk. "Training Provably Robust Models by Polyhedral Envelope Regularization". TNNLS 2021.

Rethinking Linear Approximation

Limitations:

- Computational complexity.
- Degraded bounds when ϵ is big or model is deep.

Rethinking Linear Approximation

Limitations:

- Computational complexity.
- Degraded bounds when ϵ is big or model is deep.

It is difficult to apply linear approximation to complex models.

Randomized Smoothing

Definition (Randomized Smoothing)

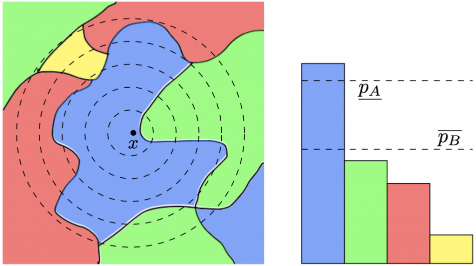
Consider a classification model $f(\boldsymbol{x}, \theta) : \mathbb{R}^H \times \Theta \rightarrow \mathcal{K}$ mapping the input to a category, its smoothed model g by a random distribution $\mathcal{D}$ is defined by $g(\boldsymbol{x}, \theta) := \mathbb{E}_{\delta \in \mathcal{D}} f(\boldsymbol{x} + \delta, \theta)$

J. Cohen, E. Rosenfeld, Z. Kolter. "Certified Adversarial Robustness via Randomized Smoothing". ICML 2019.

Randomized Smoothing

Definition (Randomized Smoothing)

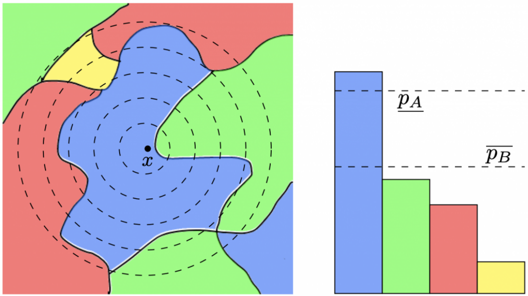
Consider a classification model $f(\boldsymbol{x}, \theta) : \mathbb{R}^H \times \Theta \to \mathcal{K}$ mapping the input to a category, its smoothed model g by a random distribution $\mathcal{D}$ is defined by $g(\boldsymbol{x}, \theta) := \mathbb{E}_{\delta \in \mathcal{D}} f(\boldsymbol{x} + \delta, \theta)$



J. Cohen, E. Rosenfeld, Z. Kolter. "Certified Adversarial Robustness via Randomized Smoothing". ICML 2019.

Randomized Smoothing

- Adversarial examples are usually "in the corner" of the decision boundary.
- An adversarial example δ for f may be surrounded by non-adversarial examples, so it will not be an adversarial example for g.
- Randomized smoothing effectively smooth the decision boundary of f.



Randomized Smoothing

We use p to represent the PDF of the distribution $\mathcal{D}$ and consider a perturbation Δ, then

$$
\begin{aligned}
g(\boldsymbol{x}, \theta) &= \int_{\mathbb{R}^H} p(\delta) f(\boldsymbol{x} + \delta, \theta) d_\delta \\
g(\boldsymbol{x} + \Delta, \theta) &= \int_{\mathbb{R}^H} p(\delta) f(\boldsymbol{x} + \Delta + \delta, \theta) d_\delta = \int_{\mathbb{R}^H} p(\delta - \Delta) f(\boldsymbol{x} + \delta, \theta) d_\delta
\end{aligned} \tag{4}
$$

Randomized Smoothing

We use p to represent the PDF of the distribution $\mathcal{D}$ and consider a perturbation Δ, then

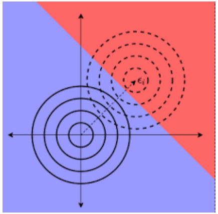
$$\begin{aligned} g(\boldsymbol{x}, \theta) &= \int_{\mathbb{R}^H} p(\delta) f(\boldsymbol{x} + \delta, \theta) d_\delta \\ g(\boldsymbol{x} + \Delta, \theta) &= \int_{\mathbb{R}^H} p(\delta) f(\boldsymbol{x} + \Delta + \delta, \theta) d_\delta = \int_{\mathbb{R}^H} p(\delta - \Delta) f(\boldsymbol{x} + \delta, \theta) d_\delta \end{aligned} \tag{4}$$

By Neyman-Pearson lemma, we can bound the lower bound of $g(\boldsymbol{x} + \Delta, \theta)$ if we bound the magnitude of Δ and the lower bound of $g(\boldsymbol{x}, \theta)$.

Randomized Smoothing

Theorem

Let f be a classifier and g is defined as $g(\boldsymbol{x}, \theta) := \mathbb{E}_{\delta\sim\mathcal{D}} f(\boldsymbol{x}+\delta, \theta)$ where $\mathcal{D}$ is a Gaussian distribution $\mathcal{N}(0, \sigma^2\mathbf{I})$, we assume c_A is one output label and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy $\mathbb{P}_{\delta\sim\mathcal{D}}(f(\boldsymbol{x}+\delta,\theta) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c\neq c_A} \mathbb{P}_{\delta\sim\mathcal{D}}(f(\boldsymbol{x}+\delta,\theta) = c)$, then we have $g(\boldsymbol{x}+\Delta, \theta)$ for all $\|\Delta\|_2 \leq \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right)$ where Φ is the cumulative distribution function of standard Gaussian.



Rethinking Randomized Smoothing

Pros:

- Scalable to any model architecture.

Cons:

- Slow inference because of Monte Carlo sampling.
- Probability guarantee.

Outline

Introduction

Verified Robustness

Empirical Robustness

Efficiency

Benefits of Robustness

Adversarial Training

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta \in \mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta, \theta))}$$

Adversarial Training

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

- Generate adversarial examples.
  - Run iteratively
    $\Delta \leftarrow \prod_{\mathcal{S}_\epsilon}(\Delta + \alpha\nabla_\Delta \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta)))$
- Training using adversarial examples.

Adversarial Training

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

- Generate adversarial examples.
  - Run iteratively $\Delta \leftarrow \prod_{\mathcal{S}_\epsilon} (\Delta + \alpha\nabla_\Delta \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta)))$
- Training using adversarial examples.

Vanilla training v.s. adversarial training.

$$\mathcal{L}_0(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \mathcal{L}(f(\boldsymbol{x},\theta))$$
$$\mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

Adversarial Training

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

- Generate adversarial examples.
  - Run iteratively
    $\Delta \leftarrow \prod_{\mathcal{S}_\epsilon}(\Delta + \alpha\nabla_\Delta \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta)))$
- Training using adversarial examples.

Vanilla training v.s. adversarial training.

$$\mathcal{L}_0(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \mathcal{L}(f(\boldsymbol{x},\theta))$$
$$\mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

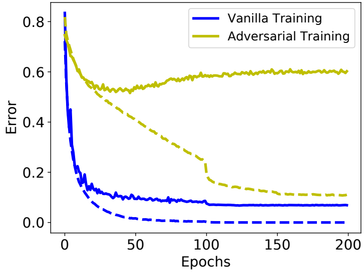


Figure: Learning curves of vanilla training (clean error) and adversarial training (robust error). Dashed and solid lines are for the training and test sets.

Adversarial Training

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

- Generate adversarial examples.
  - Run iteratively $\Delta \leftarrow \prod_{\mathcal{S}_\epsilon} (\Delta + \alpha\nabla_\Delta \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta)))$
- Training using adversarial examples.

Vanilla training v.s. adversarial training.

$$\mathcal{L}_0(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \mathcal{L}(f(\boldsymbol{x},\theta))$$
$$\mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \underline{\max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))}$$

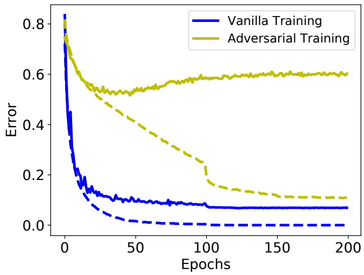


Figure: Learning curves of vanilla training (clean error) and adversarial training (robust error). Dashed and solid lines are for the training and test sets.

Convergence $\downarrow$. Generalization gap $\uparrow$.

Non-smooth Nature of Adversarial Loss Landscapes

$$g(\boldsymbol{x}, \theta) = \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| \leq L_{\theta\mathbf{x}} \|\mathbf{x}_1 - \mathbf{x}_2\|$$

C. Liu, M. Salzmann, T. Lin, R. Tomioka, S. Süsstrunk. "On the Loss Landscape of Adversarial Training: Identifying Challenges and How to Overcome Them". NeurIPS 2020.

Non-smooth Nature of Adversarial Loss Landscapes

$$g(\boldsymbol{x}, \theta) = \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| \leq L_{\theta\mathbf{x}} \|\mathbf{x}_1 - \mathbf{x}_2\|$$

$$\|\mathcal{L}_\epsilon(\theta_1) - \mathcal{L}_\epsilon(\theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta \mathcal{L}_\epsilon(\theta_1) - \nabla_\theta \mathcal{L}_\epsilon(\theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\| + 2\epsilon L_{\theta\mathbf{x}}$$

C. Liu, M. Salzmann, T. Lin, R. Tomioka, S. Süsstrunk. "On the Loss Landscape of Adversarial Training: Identifying Challenges and How to Overcome Them". NeurIPS 2020.

Non-smooth Nature of Adversarial Loss Landscapes

$$g(\boldsymbol{x}, \theta) = \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}} \max_{\Delta \in \mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\begin{aligned}
\|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| &\leq L_\theta \|\theta_1 - \theta_2\| \\
\|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| &\leq L_{\theta\theta} \|\theta_1 - \theta_2\| \\
\|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| &\leq L_{\theta\mathbf{x}} \|\mathbf{x}_1 - \mathbf{x}_2\|
\end{aligned}$$

$$\begin{aligned}
\|\mathcal{L}_\epsilon(\theta_1) - \mathcal{L}_\epsilon(\theta_2)\| &\leq L_\theta \|\theta_1 - \theta_2\| \\
\|\nabla_\theta \mathcal{L}_\epsilon(\theta_1) - \nabla_\theta \mathcal{L}_\epsilon(\theta_2)\| &\leq L_{\theta\theta} \|\theta_1 - \theta_2\| + 2\epsilon L_{\theta\mathbf{x}}
\end{aligned}$$

Adversarial perturbations depends on model parameters $\Rightarrow$ **Non-smoothness**.

C. Liu, M. Salzmann, T. Lin, R. Tomioka, S. Süsstrunk. "On the Loss Landscape of Adversarial Training: Identifying Challenges and How to Overcome Them". NeurIPS 2020.

Non-smooth Nature of Adversarial Loss Landscapes

$$g(\boldsymbol{x}, \theta) = \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta)) \qquad \mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \max_{\Delta \in \mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\begin{aligned} \|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| &\le L_\theta \|\theta_1 - \theta_2\| \\ \|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| &\le L_{\theta\theta} \|\theta_1 - \theta_2\| \\ \|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| &\le L_{\theta\mathbf{x}} \|\mathbf{x}_1 - \mathbf{x}_2\| \end{aligned}$$

$$\begin{aligned} \|\mathcal{L}_\epsilon(\theta_1) - \mathcal{L}_\epsilon(\theta_2)\| &\le L_\theta \|\theta_1 - \theta_2\| \\ \|\nabla_\theta \mathcal{L}_\epsilon(\theta_1) - \nabla_\theta \mathcal{L}_\epsilon(\theta_2)\| &\le L_{\theta\theta} \|\theta_1 - \theta_2\| + 2\epsilon L_{\theta\mathbf{x}} \end{aligned}$$

Adversarial perturbations depends on model parameters $\Rightarrow$ **Non-smoothness**.

Abrupt changes in the optimal adversarial perturbations $\Rightarrow$ **Non-smooth points** in the loss landscape.

C. Liu, M. Salzmann, T. Lin, R. Tomioka, S. Süsstrunk. "On the Loss Landscape of Adversarial Training: Identifying Challenges and How to Overcome Them". NeurIPS 2020.

Non-smooth Nature of Adversarial Loss Landscape

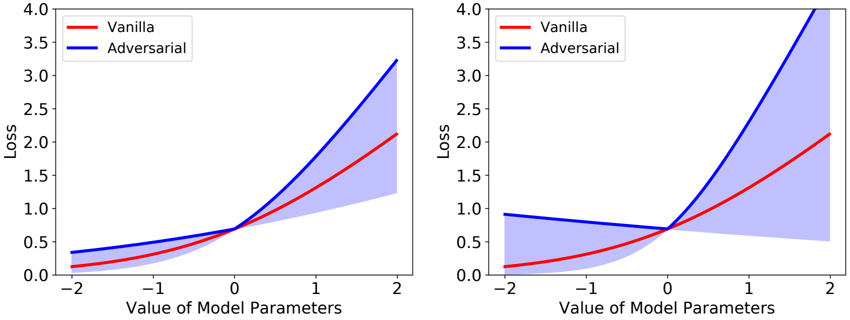


Figure: $\mathcal{L}_\epsilon(\theta) = \max_{\|\Delta\| \le \epsilon} \log\left(1 + e^{\theta\Delta}\right)$ with $\epsilon = 0.6$ (left) and $\epsilon = 1.2$ (right).

$\Delta = \epsilon$ when $\theta > 0$ and $\Delta = -\epsilon$ when $\theta \le 0$.

Non-smooth Nature of Adversarial Loss Landscape

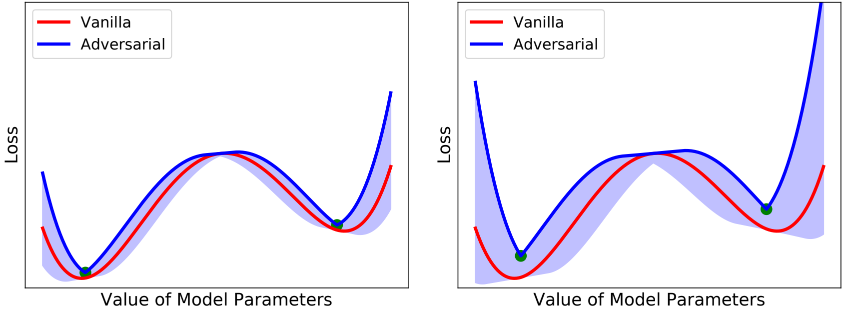


Figure: Polynomial loss function with small ϵ (left) and big ϵ (right).

Non-smoothness and Convergence Property

$$g(\boldsymbol{x}, \theta) = \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| \leq L_{\theta\mathbf{x}} \|\mathbf{x}_1 - \mathbf{x}_2\|$$

Non-smoothness and Convergence Property

$$g(\boldsymbol{x}, \theta) = \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| \leq L_{\theta\mathbf{x}} \|\mathbf{x}_1 - \mathbf{x}_2\|$$

$$\|\mathcal{L}_\epsilon(\theta_1) - \mathcal{L}_\epsilon(\theta_2)\| \leq L_\theta \|\theta_1 - \theta_2\|$$
$$\|\nabla_\theta \mathcal{L}_\epsilon(\theta_1) - \nabla_\theta \mathcal{L}_\epsilon(\theta_2)\| \leq L_{\theta\theta} \|\theta_1 - \theta_2\| + 2\epsilon L_{\theta\mathbf{x}}$$

Non-smoothness and Convergence Property

$$g(\boldsymbol{x}, \theta) = \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta)) \qquad \mathcal{L}_\epsilon(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x} + \Delta, \theta))$$

$$\begin{aligned}
\|g(\mathbf{x}, \theta_1) - g(\mathbf{x}, \theta_2)\| &\leq L_\theta\|\theta_1 - \theta_2\| \\
\|\nabla_\theta g(\mathbf{x}, \theta_1) - \nabla_\theta g(\mathbf{x}, \theta_2)\| &\leq L_{\theta\theta}\|\theta_1 - \theta_2\| \\
\|\nabla_\theta g(\mathbf{x}_1, \theta) - \nabla_\theta g(\mathbf{x}_2, \theta)\| &\leq L_{\theta\mathbf{x}}\|\mathbf{x}_1 - \mathbf{x}_2\|
\end{aligned}$$

$$\begin{aligned}
\|\mathcal{L}_\epsilon(\theta_1) - \mathcal{L}_\epsilon(\theta_2)\| &\leq L_\theta\|\theta_1 - \theta_2\| \\
\|\nabla_\theta\mathcal{L}_\epsilon(\theta_1) - \nabla_\theta\mathcal{L}_\epsilon(\theta_2)\| &\leq L_{\theta\theta}\|\theta_1 - \theta_2\| + 2\epsilon L_{\theta\mathbf{x}}
\end{aligned}$$

Theorem (Convergence Property of Adversarial Training)

Using the SGD update $\theta_{t+1} = \theta_t - \alpha_t\nabla_\theta\widehat{\mathcal{L}}_\epsilon(\theta_t)$ with unbiased, variance-bounded stochastic gradient $\nabla_\theta\widehat{\mathcal{L}}_\epsilon(\theta_t)$ and $\alpha_t = \frac{1}{L_{\theta\theta}\sqrt{T}}$ for T iterations, then:

$$\forall\gamma > 2,\ P(\|\nabla_\theta\mathcal{L}_\epsilon(\theta_T)\| \geq \gamma\epsilon L_{\theta\boldsymbol{x}}) < \frac{4}{\gamma^2 - 2\gamma + 4} \tag{5}$$

Overfitting in Adversarial Training

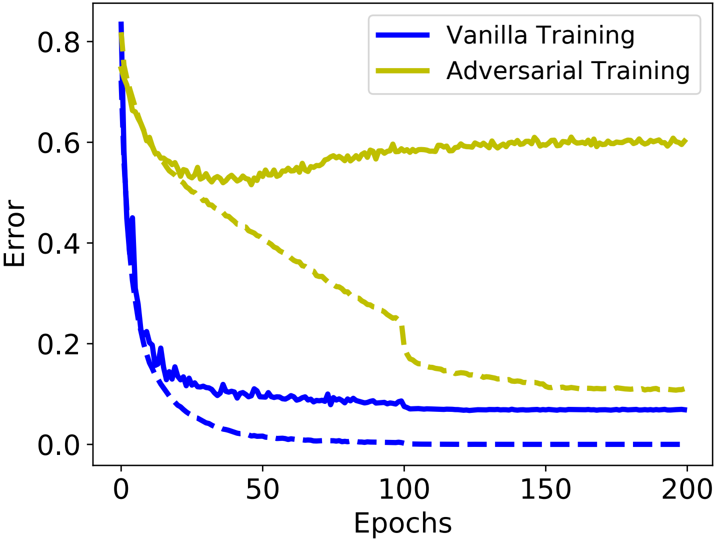


Figure: Learning curves of vanilla training (clean error) and adversarial training (robust error). Dashed and solid

Overfitting in Adversarial Training

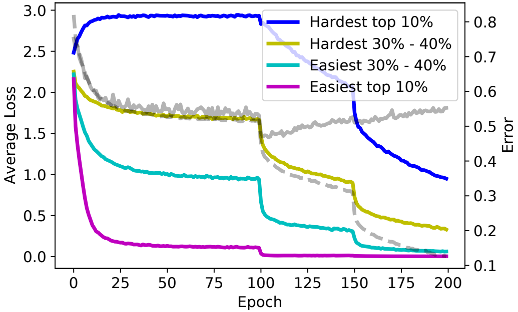


Figure: The loss values of the groups of instances of different difficulty levels.

Overfitting in Adversarial Training

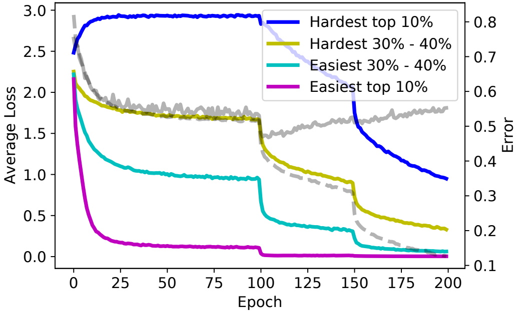


Figure: The loss values of the groups of instances of different difficulty levels.

Adversarial overfitting arises from **hard adversarial instances**.

Training Instances of Different Difficulty Levels

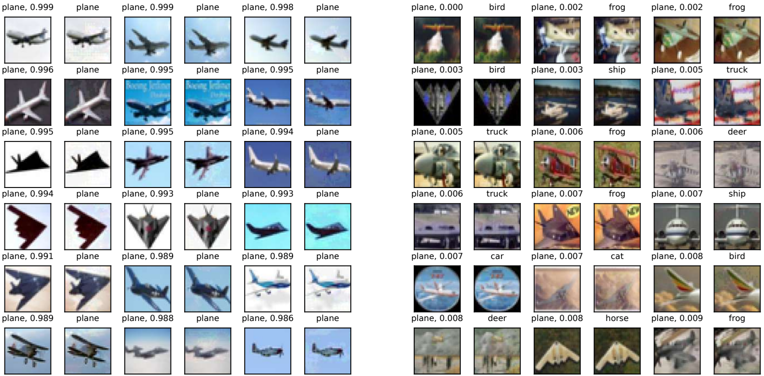


Figure: (Left) easy examples. (Right) hard examples.

Training Instances of Different Difficulty Levels

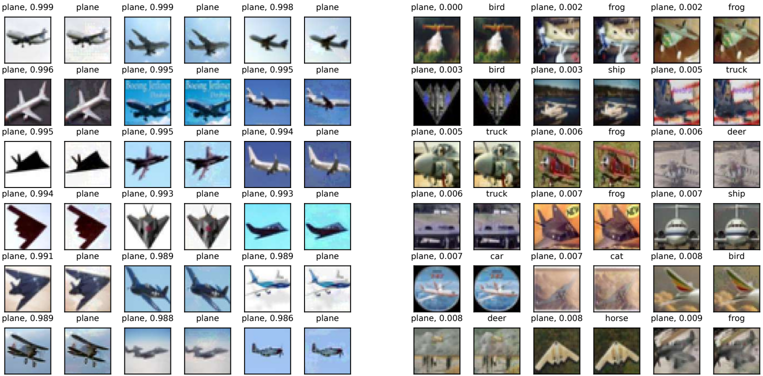


Figure: (Left) easy examples. (Right) hard examples.

How to quantitatively measure the difficulty?

Training Instances of Different Difficulty Levels

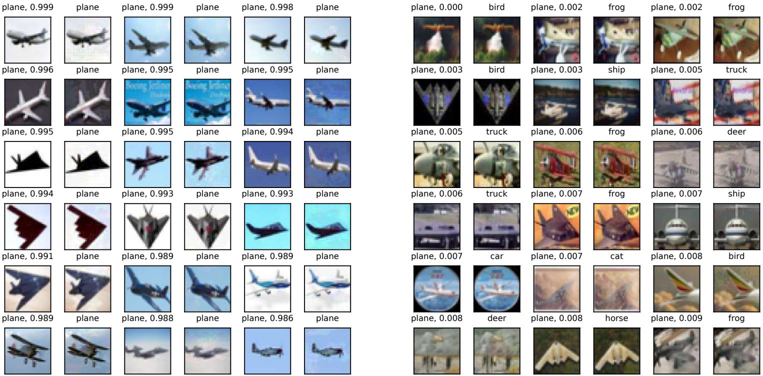


Figure: (Left) easy examples. (Right) hard examples.

How to quantitatively measure the difficulty?

Conditional variance: $\mathbb{E}[Var(y|\boldsymbol{x})]$.

Overfitting in Adversarial Training: Why?

Data The data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is binary, i.e., $\boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \{-1, +1\}$. It is sub-Gaussian with positive conditional variance $\sigma^2 = \mathbb{E}[Var[y|\boldsymbol{x}]] = \sigma^2 > 0$.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Data The data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ is binary, i.e., $\boldsymbol{x}_i \in \mathbb{R}^m, y_i \in \{-1, +1\}$. It is sub-Gaussian with positive conditional variance $\sigma^2 = \mathbb{E}[Var[y|\boldsymbol{x}]] = \sigma^2 > 0$.

Lipschitz constant $Lip(f(\cdot, \theta)) = sup_{\boldsymbol{x}_1, \boldsymbol{x}_2} \frac{\|f(\boldsymbol{x}_1, \theta) - f(\boldsymbol{x}_2, \theta)\|}{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|}$ is a good indicator of the adversarial vulnerability.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Theorem (Informal and Simplified)

Given training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and a model parameterized by bounded parameters θ, we conduct adversarial training and let $\boldsymbol{x}'$ to the adversarial examples of $\boldsymbol{x}$. If the training loss $C = \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}'_i, \theta) - y_i)^2$ is sufficiently small, then the Lipschitz constant of the model is lower bounded by the following equation almost surely.

$$Lip(f(\cdot, \theta)) \geq \beta(\sigma^2 - C + h(\epsilon, C)) \tag{6}$$

where β is a constant, $h(\epsilon, C)$ decreases with C and increases with ϵ.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Theorem (Informal and Simplified)

Given training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and a model parameterized by bounded parameters θ, we conduct adversarial training and let $\boldsymbol{x}'$ to the adversarial examples of $\boldsymbol{x}$. If the training loss $C = \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}'_i, \theta) - y_i)^2$ is sufficiently small, then the Lipschitz constant of the model is lower bounded by the following equation almost surely.

$$Lip(f(\cdot, \theta)) \geq H(\sigma^2, \epsilon, C) \tag{6}$$

$\sigma \uparrow$, $H \uparrow$; $\epsilon \uparrow$, $H \uparrow$; $C \downarrow$, $H \uparrow$.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Theorem (Informal and Simplified)

Given training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and a model parameterized by bounded parameters θ, we conduct adversarial training and let $\boldsymbol{x}'$ to the adversarial examples of $\boldsymbol{x}$. If the training loss $C = \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}'_i, \theta) - y_i)^2$ is sufficiently small, then the Lipschitz constant of the model is lower bounded by the following equation almost surely.

$$Lip(f(\cdot, \theta)) \geq H(\sigma^2, \epsilon, C) \tag{6}$$

- Lipschitz constant indicates adversarial vulnerability.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Theorem (Informal and Simplified)

Given training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and a model parameterized by bounded parameters θ, we conduct adversarial training and let $\boldsymbol{x}'$ to the adversarial examples of $\boldsymbol{x}$. If the training loss $C = \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}'_i, \theta) - y_i)^2$ is sufficiently small, then the Lipschitz constant of the model is lower bounded by the following equation almost surely.

$$Lip(f(\cdot, \theta)) \geq H(\sigma^2, \epsilon, C) \tag{6}$$

- Lipschitz constant indicates adversarial vulnerability.
  - C is sufficiently small $\Longrightarrow$ Lipschitz constant indicates generalization gap.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Theorem (Informal and Simplified)

Given training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, and a model parameterized by bounded parameters θ, we conduct adversarial training and let $\boldsymbol{x}'$ to the adversarial examples of $\boldsymbol{x}$. If the training loss $C = \frac{1}{n}\sum_{i=1}^n (f(\boldsymbol{x}'_i, \theta) - y_i)^2$ is sufficiently small, then the Lipschitz constant of the model is lower bounded by the following equation almost surely.

$$Lip(f(\cdot, \theta)) \geq H(\sigma^2, \epsilon, C) \tag{6}$$

- Lipschitz constant indicates adversarial vulnerability.
  - C is sufficiently small $\Longrightarrow$ Lipschitz constant indicates generalization gap.
  - $C\downarrow$: training processes $\Longrightarrow$ $H\uparrow$: overfitting.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Theorem (Informal and Simplified)

Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, and a model parameterized by bounded parameters θ, we conduct adversarial training and let $\mathbf{x}'$ to the adversarial examples of $\mathbf{x}$. If the training loss $C = \frac{1}{n}\sum_{i=1}^n (f(\mathbf{x}'_i, \theta) - y_i)^2$ is sufficiently small, then the Lipschitz constant of the model is lower bounded by the following equation almost surely.

$$Lip(f(\cdot, \theta)) \geq H(\sigma^2, \epsilon, C) \tag{6}$$

- Lipschitz constant indicates adversarial vulnerability.
  - C is sufficiently small $\Longrightarrow$ Lipschitz constant indicates generalization gap.
  - $C \downarrow$: training processes $\Longrightarrow$ $H \uparrow$: overfitting.
  - $\sigma \uparrow$: harder instances $\Longrightarrow$ $H \uparrow$: overfitting.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: Why?

Theorem (Informal and Simplified)

Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, and a model parameterized by bounded parameters θ, we conduct adversarial training and let $\mathbf{x}'$ to the adversarial examples of $\mathbf{x}$. If the training loss $C = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}'_i, \theta) - y_i)^2$ is sufficiently small, then the Lipschitz constant of the model is lower bounded by the following equation almost surely.

$$Lip(f(\cdot, \theta)) \geq H(\sigma^2, \epsilon, C) \tag{6}$$

- Lipschitz constant indicates adversarial vulnerability.
  - C is sufficiently small $\Longrightarrow$ Lipschitz constant indicates generalization gap.
  - $C \downarrow$: training processes $\Longrightarrow$ $H \uparrow$: overfitting.
  - $\sigma \uparrow$: harder instances $\Longrightarrow$ $H \uparrow$: overfitting.
  - $\epsilon \uparrow$: larger adversarial budget $\Longrightarrow$ $H \uparrow$: overfitting.

C. Liu, Z. Huang, M. Salzmann, T. Zhang, S. Süsstrunk. "On the Impact of Hard Adversarial Instances on Overfitting in Adversarial Training". 2022.

Overfitting in Adversarial Training: How?

Methods mitigating adversarial overfitting implicitly downplay hard instances.

- Weaker perturbation.
- Adaptive and easier targets.
- Smaller weights when calculating the loss objective.

Outline

Introduction

Verified Robustness

Empirical Robustness

Efficiency

Benefits of Robustness

Adversarial Training is Expensive

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta, \theta))$$

- If we run projected gradient descent (PGD) for M iterations, then the complexity of adversarial training will be $(M+1)$ times that of training on clean inputs.

Adversarial Training is Expensive

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))$$

- If we run projected gradient descent (PGD) for M iterations, then the complexity of adversarial training will be $(M+1)$ times that of training on clean inputs.
- We can decrease the value of M to decrease the complexity.

Adversarial Training is Expensive

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}} \max_{\Delta\in\mathcal{S}_\epsilon} \mathcal{L}(f(\boldsymbol{x}+\Delta,\theta))$$

- If we run projected gradient descent (PGD) for M iterations, then the complexity of adversarial training will be $(M+1)$ times that of training on clean inputs.
- We can decrease the value of M to decrease the complexity.
- But at the cost of performance and stability.

Catastrophic Overfitting

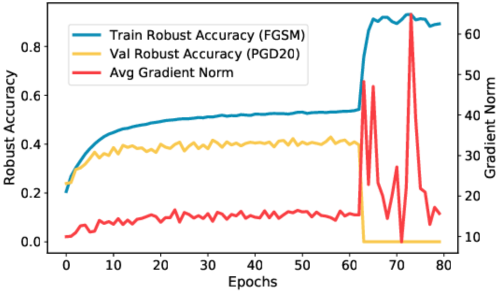


Figure: Catastrophic Overfitting.

Catastrophic Overfitting

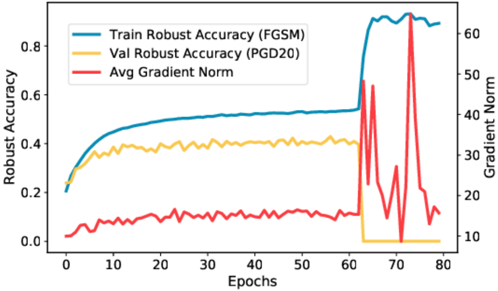


Figure: Catastrophic Overfitting.

- Small M typically means large step sizes.

Catastrophic Overfitting

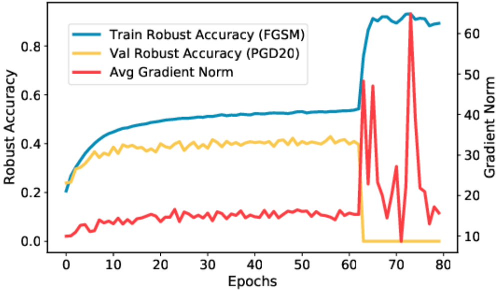


Figure: Catastrophic Overfitting.

- Small M typically means large step sizes.
- Large gradient norm $\nabla_{\Delta}\mathcal{L}$ indicates distorted loss landscape.

Catastrophic Overfitting

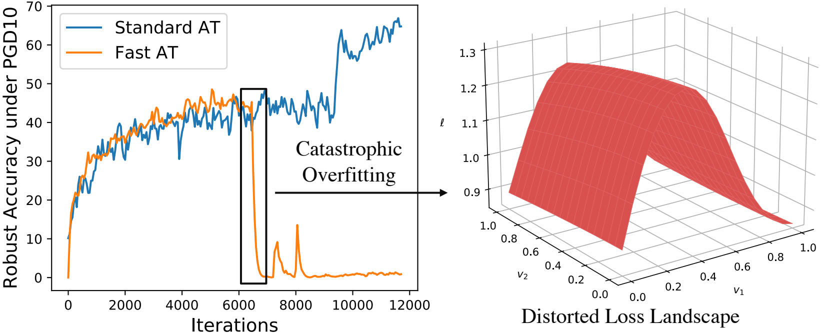


Figure: Loss landscape distortion when catastrophic overfitting happens.

Solutions for Catastrophic Overfitting

Inspired by pre-conditioned optimizers.

- Large gradients $\rightarrow$ hard examples $\rightarrow$ smaller step size.
- Small gradients $\rightarrow$ easy examples $\rightarrow$ larger step size.

Z. Huang, Y. Fan, C. Liu, W. Zhang, Y. Zhang, M. Salzmann, S. Süsstrunk, J. Wang. "Fast Adversarial Training with Adaptive Steps". TIP 2023.

Y. Jiang, C. Liu, Z. Huang, M. Salzmann, S. Süsstrunk. "Towards Stable and Efficient Adversarial Training against l_1 Bounded Adversarial Attacks". ICML 2023.

Solutions for Catastrophic Overfitting

Inspired by pre-conditioned optimizers.

- Large gradients $\rightarrow$ hard examples $\rightarrow$ smaller step size.
- Small gradients $\rightarrow$ easy examples $\rightarrow$ larger step size.
- We use exponential moving average to calculate the expected gradient magnitude $m \leftarrow \beta m + (1 - \beta)\|\nabla_{\Delta}\mathcal{L}\|_2$ for each training instance.

Z. Huang, Y. Fan, C. Liu, W. Zhang, Y. Zhang, M. Salzmann, S. Süsstrunk, J. Wang. "Fast Adversarial Training with Adaptive Steps". TIP 2023.

Y. Jiang, C. Liu, Z. Huang, M. Salzmann, S. Süsstrunk. "Towards Stable and Efficient Adversarial Training against l_1 Bounded Adversarial Attacks". ICML 2023.

Solutions for Catastrophic Overfitting

Inspired by pre-conditioned optimizers.

- Large gradients $\rightarrow$ hard examples $\rightarrow$ smaller step size.
- Small gradients $\rightarrow$ easy examples $\rightarrow$ larger step size.
- We use exponential moving average to calculate the expected gradient magnitude $m \leftarrow \beta m + (1 - \beta)\|\nabla_{\Delta}\mathcal{L}\|_2$ for each training instance.
- The actual step size is $\frac{\alpha}{m}$ for each training instance.

Z. Huang, Y. Fan, C. Liu, W. Zhang, Y. Zhang, M. Salzmann, S. Süsstrunk, J. Wang. "Fast Adversarial Training with Adaptive Steps". TIP 2023.

Y. Jiang, C. Liu, Z. Huang, M. Salzmann, S. Süsstrunk. "Towards Stable and Efficient Adversarial Training against l_1 Bounded Adversarial Attacks". ICML 2023.

Other Solutions for Catastrophic Overfitting

- Smaller step size but memroize the perturbations in the last epoch.
- Gradient regularization to make the loss landscape more smooth.

H. Zheng, Z. Zhang, J. Gu, H. Lee, A. Prakash. "Efficient adversarial training with transferable adversarial examples". CVPR 2020.

M. Andriushchenko, N. Flammarion. "Understanding and improving fast adversarial training". NeurIPS 2020.

Other Ways to Improve Effectiveness

- Pruning network to make it more sparse can help robustness.
  - We can even prune the network with their initialized parameters unchanged. (strong lottery ticket hypothesis)
- Proper quantization can help robustness.

C. Liu, Z. Zhao, S. Süsstrunk, M. Salzmann. "Robust Binary Models by Pruning Randomly-initialized Networks". NeurIPS 2022.

Outline

Introduction

Verified Robustness

Empirical Robustness

Efficiency

Benefits of Robustness

Challenges of Obtaining Robustness

- Larger models.
- Larger datasets.
- Higher complexity.
- Poor transferability between different types of perturbations.
- ...

Additional Benefits of Obtaining Robustness

Adversarial perturbations can be considered as a strong data augmentation.

C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille ,Q. Le. "Adversarial Examples Improve Image Recognition". CVPR 2020.

Additional Benefits of Obtaining Robustness

Adversarial perturbations can be considered as a strong data augmentation.

- Use clean inputs to train convolutional layers + normalization layers A.
- Use adversarial inputs to train convolutional layers + normalization layers B.

C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille ,Q. Le. "Adversarial Examples Improve Image Recognition". CVPR 2020.

Additional Benefits of Obtaining Robustness

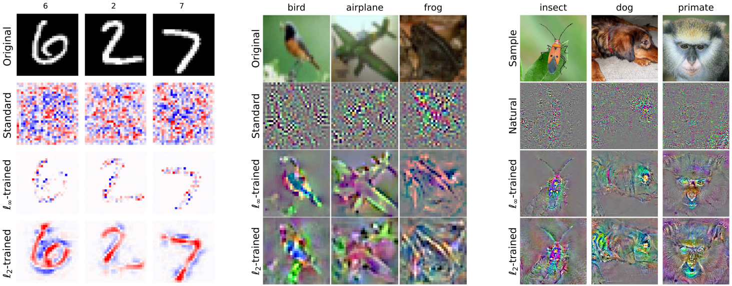
Adversarial perturbations can be considered as a strong data augmentation.

- Use clean inputs to train convolutional layers + normalization layers A.
- Use adversarial inputs to train convolutional layers + normalization layers B.
- Then we will get two models with shared convolutional layers. Both has good performance, since the shared layers are trained by more data.

C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille ,Q. Le. "Adversarial Examples Improve Image Recognition". CVPR 2020.

Additional Benefits of Obtaining Robustness

Adversarial perturbations destroy the non-robust features of the input and force the model to learn robust features, which is aligned with human perception.



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

Figure: The visualization of $\nabla_{\Delta}\mathcal{L}$.

A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry. "Adversarial Examples are not Bugs, They are Features". NeurIPS 2019.

Additional Benefits of Obtaining Robustness

Robust features are usually more general features.

- Pretrained models by adversarial training can achieve better performance after fine-tuning on a related task.

H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, A. Madry. "Do adversarially robust imagenet models transfer better?". NeurIPS 2020.

Remaining Challenges

- A better trade-off between clean accuracy and robust accuracy.
- Robustness against multiple types of adversarial perturbations.
- Narrow the gap between empirical robustness and verified robustness.
- ...