Rethinking the Min-max Problem for Adversarial Robustness

Yisen Wang
Peking University

yisen.wang@pku.edu.cn
https://yisenwang.github.io/

Guest Lecture for CS498@UIUC
Mar 17, 2021
ML is Everywhere

- Medical diagnosis
- Image classification
- Speech recognition
- Object detection
- Machine Learning
- Playing games
- Autonomous driving
However

Dog, 82% confidence
Ostrich, 98% confidence
Are we doomed? 
(Is ML inherently not reliable?)

**NO!** But we need to re-think how we do ML 
(adversarial aspects = stress-testing our solutions)
Adversarial Example

Model training:
\[
\min_{\theta} \sum_{(x_i, y_i) \in D_{train}} L(f_\theta(x_i), y_i)
\]

Adversarial attack:
\[
\max_{x'} L(f_\theta(x'), y) \quad \text{st. } \|x' - x\|_p \leq \epsilon \text{ for } x \in D_{test}
\]

- increase error
- small change
- test time attack

\[
\|x' - x\|_\infty \leq \epsilon = \frac{8}{255} \approx 0.031
\]

• Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014):
\[
x' = x + \epsilon \cdot \text{sign} \nabla_x L(f_\theta(x), y)
\]

• Projected Gradient Descent (PGD) is an iterative version of FGSM (Madry et al., 2018)
\[
x'(k+1) = \Pi_\epsilon (x'(k) + \alpha \cdot \text{sign} \nabla_x L(f_\theta(x'(k)), y))
\]

\(D_{train}\): training data
\(x_i\): training sample
\(y_i\): class label
\(L\): loss function
\(f_\theta\): model
How to obtain adversarially **robust** models?
Adversarial training is a **min-max optimization** process:

\[
\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max \left( \left\| x'_i - x_i \right\|_p \leq \epsilon \right) L(f_\theta(x'_i), y_i)
\]

\[L: \text{loss}, \quad f_\theta: \text{model}, \quad x_i: \text{clean example}, \quad y_i: \text{class}, \quad x'_i: \text{adversarial example.}\]

1. **Inner Maximization:**
   - This is to **generate adversarial examples**, by maximizing the loss \(L\).
   - It is a constrained optimization problem: \(\|x'_i - x_i\|_p \leq \epsilon\).

2. **Outer Minimization:**
   - A typical process to **train a model**, but on adversarial examples \(x'_i\) generated by the inner maximization.
Convergence Score of the Maximization

**Question:** How well the inner maximization is solved?

**Definition (First-Order Stationary Condition (FOSC))**

Given a data sample \( x^0 \in X \), let \( x^k \) be an intermediate example found at the \( k^{th} \) step of the inner maximization. The First-Order Stationary Condition of \( x^k \) is

\[
c(x^k) = \max_{x \in \chi} \langle x - x^k, \nabla_x f(\theta, x^k) \rangle,
\]

where \( \chi = \{ x \mid \|x - x^0\|_\infty \leq \epsilon \} \) is the input domain of the \( \epsilon \)-ball around normal example \( x^0 \), \( f(\theta, x^k) = \ell(h_\theta(x^k), y) \), and \( \langle \cdot \rangle \) is the inner product.

**FOSC:**

- A smaller value of \( c(x^k) \) indicates a better solution of the inner maximization, or equivalently, better convergence quality of the adversarial example \( x^k \).
- To help Danskin’s Theorem hold.

Convergence Theorem

Theorem 1

Under certain assumptions, let \( \Delta = L_S(\theta^0) - \min_{\theta} L_S(\theta) \). If the step size of the outer minimization is set to \( \eta_t = \min \left( \frac{1}{L}, \sqrt{\frac{\Delta}{L\sigma^2 T}} \right) \). Then the output of Adversarial Training satisfies

\[
\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla L_S(\theta^t)\|^2_2] \leq 4\sigma \sqrt{\frac{L\Delta}{T}} + \frac{5L^2_{\theta x}\delta}{\mu},
\]

where \( L = \left( \frac{L_{\theta x}L_{\theta x}}{\mu} + L_{\theta \theta} \right) \).

- Inner maximization: FOSC \( \leq \delta \), adversarial training can converge to a first-order stationary point up to a precision of \( \frac{5L^2_{\theta x}\delta}{\mu} \).

- If \( \delta \) is sufficiently small such that \( \frac{5L^2_{\theta x}\delta}{\mu} \) small enough, adversarial training can find a robust model \( \theta^T \).
Why do we need FOSC?

Adversarial Training with different settings for PGD-based inner maximization.

- **PGD step size**: PGD-$\frac{\epsilon}{2}$ / PGD-$\frac{\epsilon}{4}$ produces the best robustness, their FOSC values are also concentrated around 0.

- **PGD step number**: similar robustness, with PGD-20/30 are slightly better, reflected by the distribution of FOSC.

- **Loss distributions** are similar for different robustness.

FOSC is a good and reliable indicator of the final robustness.
FOSC View of Adversarial Training

- Standard adversarial training **overfits** to strong PGD adversarial examples at the **early stage**.

- Simply use **weak attack FGSM** at the **early stage** can improve robustness.

- Improvement in robustness is also reflected in FOSC distribution.

The principle behind warm-up techniques
Warm-up is a method to solve max better, is there other options?
Rethinking the Robust Generalization Gap

Adversarial training is a \textit{min-max optimization} process:

\[
\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max \max_{x'_i \in B(\theta, \epsilon)} L(f_{\theta}(x'_i), y_i)
\]

where

\[
\rho(w) = \frac{1}{n} \sum_{i=1}^{n} \max_{x'_i \in B(w, \epsilon)} \ell(f_w(x'_i), y_i)
\]

Standard training

Adversarial training

View from weight loss landscape

- Inspiring from standard Training:
  - flatter weight loss landscape, smaller standard generalization gap

Is this conclusion still existing in adversarial training?

Adapted Visualization Method

• Inspiring from standard Training:
  • flatter weight loss landscape, smaller standard generalization gap
• Is this conclusion still existing in adversarial training?

Visualization method in Hao Li et al. NeurIPS2018

Standard training
$g(\alpha) = L(f_{w+a \text{d}}(x_i), y_i)$

Adversarial training
$g(\alpha) = L(f_{w+a \text{d}}(x_i'), y_i)$ ?
$x_i'$ is from pre-generated adversarial examples\[^{[1,2]}\]

The correct way:
$g(\alpha) = \rho(w + a \text{d}) = \frac{1}{n} \sum_{i=1}^{n} \max_{\|x_i' - x_i\|_p \leq \alpha} \ell(f_{w + a \text{d}}(x_i'), y_i),$

Generating adversarial examples on-the-fly

\[^{[1]}\] Understanding adversarial robustness through loss landscape geometries, arxiv 2019.
\[^{[2]}\] Interpreting adversarial robustness: A view from decision surface in input space, arxiv 2018.
Weight loss landscape has a strong correlation with robust generalization gap.
Weight loss landscape has a strong correlation with robust generalization gap.
Theoretical view

- Informally from PAC-Bayesian bound

\[
\mathbb{E}_{\{x_i, y_i\}_{i=1}^n, u}[\rho(w + u)] \leq \rho(w) + \{\mathbb{E}_{u}[\rho(w + u)] - \rho(w)\} + 4\sqrt{\frac{1}{n}KL(w + u\|P) + \ln \frac{2n}{\delta}}.
\]

flatness of weight loss landscape

- Explicitly flattening the weight loss landscape via replacing expectation by maximization

\[
\min_\theta \frac{1}{n} \sum_{i=1}^n \max_{\|x'_i - x_i\|_p \leq \epsilon} L(f_\theta(x'_i), y_i) \quad \rightarrow \quad \min_\theta \max_{\|v\|_p \leq \|\theta\|_p} \frac{1}{n} \sum_{i=1}^n \max_{\|x'_i - x_i\|_p \leq \epsilon} L(f_\theta + v(x'_i), y_i)
\]

- Two max makes the maximization \((\text{min-max})\) solve better

- How to intuitively understand these two perturbations?
  - Input perturbation is \textit{local} worst for each example
  - Weight perturbation is \textit{global} worst for multiple examples
Implementation

AWP-based Adversarial training (AT-AWP)

\[
\min_{\theta} \max_{\|v\|_p \leq \gamma \|\theta\|_p} \frac{1}{n} \sum_{i=1}^{n} \max_{\|x'_i - x_i\|_p \leq \epsilon} L(f_{\theta+v}(x'_i), y_i)
\]

• An empirical implementation:
  1. craft adversarial examples \(x'_i\);
  2. calculate AWP based on \(x'_i\) using one extra forward and backward propagation;
  3. update the parameter using the gradient based on \(x'_i\) and AWP.

• Only \(\sim 8\%\) time overhead in our implementation of AT-AWP.

• AWP is easily extended to other methods, such as TRADES, MART and RST.
Real robustness improvement

- AWP indeed flattens weight loss landscape, and reduces the robust generalization gap.

- AWP really improves both the last and best robustness during training.
AWP vs. Random WP

- AWP easily finds the worst-case perturbation, while RWP needs a relatively large perturbation;
- AWP obtain a flatter weight loss landscape using smaller perturbations;
- AWP balances the training robustness and robust gap well.

(a) Loss curve  (b) Weight loss landscapes  (c) Robustness
Universal robustness improvement

Table 2: Test robustness (%) on CIFAR-10 using WideResNet under $L_\infty$ threat model. We omit the standard deviations of 5 runs as they are very small (< 0.40%), which hardly effect the results.

<table>
<thead>
<tr>
<th>Defense</th>
<th>Natural</th>
<th>FGSM</th>
<th>PGD-20</th>
<th>PGD-100</th>
<th>CW_\infty</th>
<th>SPSA</th>
<th>AA</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>86.07</td>
<td>61.76</td>
<td>56.10</td>
<td>55.79</td>
<td>54.19</td>
<td>61.40</td>
<td>52.60\textsuperscript{4}</td>
</tr>
<tr>
<td>AT-AWP</td>
<td>85.57</td>
<td>62.90</td>
<td>58.14</td>
<td>57.94</td>
<td>55.96</td>
<td>62.65</td>
<td>54.04</td>
</tr>
<tr>
<td>TRADES</td>
<td>84.65</td>
<td>61.32</td>
<td>56.33</td>
<td>56.07</td>
<td>54.20</td>
<td>61.10</td>
<td>53.18</td>
</tr>
<tr>
<td>TRADES-AWP</td>
<td>85.36</td>
<td>63.49</td>
<td>59.27</td>
<td>59.12</td>
<td>57.07</td>
<td>63.85</td>
<td>56.17</td>
</tr>
<tr>
<td>MART</td>
<td>84.17</td>
<td>61.61</td>
<td>58.56</td>
<td>57.88</td>
<td>54.58</td>
<td>58.90</td>
<td>51.10</td>
</tr>
<tr>
<td>MART-AWP</td>
<td>84.43</td>
<td>63.98</td>
<td>60.68</td>
<td>59.32</td>
<td>56.37</td>
<td>62.75</td>
<td>54.23</td>
</tr>
<tr>
<td>Pre-training</td>
<td>87.89</td>
<td>63.27</td>
<td>57.37</td>
<td>56.80</td>
<td>55.95</td>
<td>62.55</td>
<td>54.99</td>
</tr>
<tr>
<td>Pre-training-AWP</td>
<td>88.33</td>
<td>66.34</td>
<td>61.40</td>
<td>61.21</td>
<td>59.28</td>
<td>65.55</td>
<td>57.39</td>
</tr>
<tr>
<td>RST</td>
<td>89.69</td>
<td>67.94</td>
<td>62.60</td>
<td>62.22</td>
<td>60.47</td>
<td>67.60</td>
<td>59.65</td>
</tr>
<tr>
<td>RST-AWP</td>
<td>88.25</td>
<td>69.60</td>
<td>63.73</td>
<td>63.58</td>
<td>61.62</td>
<td>68.72</td>
<td>61.10</td>
</tr>
</tbody>
</table>

Table 3: Test robustness (%) on CIFAR-10 using WideResNet under $L_\infty$ threat model. In brackets, + indicates improvements over Pre-training.

<table>
<thead>
<tr>
<th>Defense</th>
<th>PGD-20</th>
<th>CW_\infty</th>
<th>AA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-training</td>
<td>57.37</td>
<td>55.95</td>
<td>54.92</td>
</tr>
<tr>
<td>TRADES-AWP</td>
<td>59.27\textsuperscript{(+1.90)}</td>
<td>57.07\textsuperscript{(+1.12)}</td>
<td>56.17\textsuperscript{(+1.25)}</td>
</tr>
<tr>
<td>Pre-training-AWP</td>
<td>61.40\textsuperscript{(+4.03)}</td>
<td>59.28\textsuperscript{(+3.33)}</td>
<td>57.39\textsuperscript{(+2.47)}</td>
</tr>
</tbody>
</table>
Except max process, how about min process?
Revisiting the Input Examples

Adversarial training is a **min-max optimization** process:

\[
\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \max_{\|x'_i - x_i\|_p \leq \epsilon} L(f_\theta(x'_i), y_i)
\]

$L$: loss, $f_\theta$: model, $x_i$: clean example, $y_i$: class, $x'_i$: adversarial example.

Adversarial examples are only defined on **correctly classified examples**

**How about misclassified examples?**

Misclassified vs. correctly classified examples

- A pre-trained network to select the **same size (13%)**
  - Subset of misclassified examples $S^-$
  - Subset of correctly classified examples $S^+$

**Misclassified examples have a significant impact on the final robustness**
Delving into the max and min processes

- For inner maximization process:
  - Weak attack on misclassified examples $S^-$
  - Weak attack on correctly classified examples $S^+$

- For outer minimization process:
  - Regularization on misclassified examples $S^-$
  - Regularization on correctly classified examples $S^+$

Different maximization techniques have **negligible** effect

Different minimization techniques have **significant** effect
Misclassification aware adversarial risk

- Adversarial risk:
  \[ \mathcal{R}(h_\theta) = \frac{1}{n} \sum_{i=1}^{n} \max_{x'_i \in B_\epsilon(x_i)} \mathbf{1}(h_\theta(x'_i) \neq y_i), \]

- Correctly classified and misclassified examples:
  \[ S_{h_\theta}^+ = \{ i : i \in [n], h_\theta(x_i) = y_i \} \quad \text{and} \quad S_{h_\theta}^- = \{ i : i \in [n], h_\theta(x_i) \neq y_i \} \]

- Misclassification aware adversarial risk:
  \[
  \min_\theta \mathcal{R}_{\text{misc}}(h_\theta) := \frac{1}{n} \left( \sum_{i \in S_{h_\theta}^+} \mathcal{R}^+(h_\theta, x_i) + \sum_{i \in S_{h_\theta}^-} \mathcal{R}^-(h_\theta, x_i) \right)
  \]
  \[= \frac{1}{n} \sum_{i=1}^{n} \left( \mathbf{1}(h_\theta(x'_i) \neq y_i) + \mathbf{1}(h_\theta(x_i) \neq h_\theta(x'_i)) \cdot \mathbf{1}(h_\theta(x_i) \neq y_i) \right) \]
Misclassification Aware adveRsarial Training (MART)

- Surrogate loss functions (existing methods and MART):

<table>
<thead>
<tr>
<th>Defense Method</th>
<th>Loss Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>$\text{CE}(\mathbf{p}(\hat{x}', \theta), y)$</td>
</tr>
<tr>
<td>ALP</td>
<td>$\text{CE}(\mathbf{p}(\hat{x}', \theta), y) + \lambda \cdot | \mathbf{p}(\hat{x}', \theta) - \mathbf{p}(x, \theta) |^2$</td>
</tr>
<tr>
<td>CLP</td>
<td>$\text{CE}(\mathbf{p}(x, \theta), y) + \lambda \cdot | \mathbf{p}(\hat{x}', \theta) - \mathbf{p}(x, \theta) |^2$</td>
</tr>
<tr>
<td>TRADES</td>
<td>$\text{CE}(\mathbf{p}(x, \theta), y) + \lambda \cdot \text{KL}(\mathbf{p}(x, \theta) | \mathbf{p}(\hat{x}', \theta))$</td>
</tr>
<tr>
<td>MMA</td>
<td>$\text{CE}(\mathbf{p}(\hat{x}', \theta), y) \cdot 1(h_\theta(x) = y) + \text{CE}(\mathbf{p}(x, \theta), y) \cdot 1(h_\theta(x) \neq y)$</td>
</tr>
<tr>
<td>MART</td>
<td>$\text{BCE}(\mathbf{p}(\hat{x}', \theta), y) + \lambda \cdot \text{KL}(\mathbf{p}(x, \theta) | \mathbf{p}(\hat{x}', \theta)) \cdot (1 - \mathbf{p}_y(x, \theta))$</td>
</tr>
</tbody>
</table>

$$\text{BCE}(\mathbf{p}(\hat{x}', \theta), y_i) = -\log \left( \mathbf{p}_{y_i}(\hat{x}', \theta) \right) - \log \left( 1 - \max_{k \neq y_i} \mathbf{p}_k(\hat{x}', \theta) \right)$$

![Graphs](image1.png)  
(a) Removing  
(b) Replacing
Beyond training objective, is model architecture related to robustness?
Skip connection matters

• Neural network architectures:
  • Skip connection, activation, batch normalization, …

• Skip connection

white-box / black-box

100% / 52.52%
100% / 55.24%
100% / 62.10%
99.86% / 47.70%

Skip connections expose more transferable information!

Skip Gradient Method (SGM)

\[ \nabla_x \ell = \frac{\partial \ell}{\partial z_L} \prod_{i=0}^{L-1} \left( \gamma \frac{\partial f_{i+1}}{\partial z_i} + 1 \right) \frac{\partial z_0}{\partial x} \]

Target model: IncV3

- PGD
- DI
- SGM (ours)
- MI
- TI

Target model: IncV3

Source Model

Parameter \( \gamma \) for SGM
Takehome Message

• For the min-max problem, the following aspects are essential:
  – how to make max solves better
  – How to make min process easily
• Model architecture is also important for adversarial research
Related Papers

- Dongxian Wu, Shu-Tao Xia, Yisen Wang, “Adversarial Weight Perturbation Helps Robust Generalization”, NeurIPS 2020
- Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, Quanquan Gu, “Improving Adversarial Robustness Requires Revisiting Misclassified Examples”, ICLR 2020
- Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, Yisen Wang, “Improving Adversarial Robustness via Channel-wise Activation Suppressing”, ICLR 2021 Spotlight

Building ML one can truly rely on
Thanks!

yisen.wang@pku.edu.cn
https://yisenwang.github.io/