Adversarially Robust Machine Learning for Critical Applications

By: Ziad Ali
Outline

• Introduction
• Applications of Attacks
• Defenses & their limitations
• Conclusion & Future work
Deep Neural Networks: Feed Forward

Deep neural network

input layer  hidden layer 1  hidden layer 2  hidden layer 3

output layer

Adapted from Nielsen (2015)
Convolutional Neural Network
LeNet 5

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner,
Why Deep Learning Applications are Critical?

- Oil & Gas industry for predicting failure
- Medicine for diagnosis of diseases
- Self-driving cars
- Speech Recognition
- DL based malware detection
Datasets: MNIST & CIFAR-10

- MNIST 28x28
- 60000 Training Images
- 10000 Testing Images

- CIFAR-10 32x32x3
- 50000 Training Images
- 10000 Testing Images

https://www.cs.toronto.edu/~kriz/cifar.html
Attacks on ML:
Adversarial ML: Evasion Attacks

Adapted from Goodfellow (2015)
Adversarial ML: Threat Model

- White-box Attacks: Full access (weights, dataset, learning algorithm)
- Grey-box Attacks: Partial access
- Black-box Attacks: No access
- Adaptive Attacks: attacks targeted to a specific defense
Threat Model: Adversary's Goals

- Confidence Reduction (99% cat to 12% cat)
- Misclassification (cat to any other label)
- Targeted Misclassification (cat to dog)
Threat Model: Adversarial Robustness Metrics

- Classification Error: Number of test samples misclassified
- Robust Classification Error (R): Number of perturbed test samples misclassified
- Robust Accuracy (adversarial robustness): 1-R

Adversarially Robust Generalization Requires More Data (Schmidt et. al 2018)
Attacks: Black-box Attack in Physical World

Adversarial Examples in Physical World (Kurakin et. Al 2015)
Attacks: Segmentation Task

(a)

(b)

Adversarial Attacks and Defenses in Deep Learning (Ren et. al)
Attacks: Speech-to-Text (Audio)

Audio Adversarial Examples: Targeted Attacks on Speech-to-Text (Carlini et. al)
Adversarial Defenses:

- Certified Defenses: give a guarantee of robustness
- Input Pre-processing Defenses: apply transformations to input
- Detection Defenses: detect adversarial behaviour
- Adversarial Retraining: retrain the model on adversarial samples

Adapted from AprilPyone (2020)
Defenses: Input Transformations

- Image Cropping and Rescaling
- Bit-Depth Reduction
- JPEG Compression
- TV minimization
- Image Quilting

- Broken with EOT and BPDA attack by (Athalye et. al)
- Accuracy reduced to 0%!!!
Defense: Key-Based Input Transformation

Block-wise Image Transformation with Secret Key for Adversarially Robust Defense (AprilPyone et al)
Defenses: Detection

Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks (2017)

- He et. al show feature squeezing is vulnerable to adaptive attacks
- Nicholas Carlini bypassed 10 different detection methods to show they are not effective (Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods (2019))
Defenses: Adversarial Retraining

- Proposed by Goodfellow et. al (2015) using FGSM
- ASR fell from 89.4% to 17.8% for FGSM
- Unsuccessful against iterative attacks
- Enhanced by Madry et. al (2017) using PGD
- Defended against majority of strongest attacks (89.3% MNIST, 45.8% CIFAR-10)
- Natural accuracy drops from 95.2% to 87.3%
Adversarial Retraining: Surrogate Losses

- Logit Pairing
- Trades
- MART

<table>
<thead>
<tr>
<th>Defense Method</th>
<th>Loss Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Standard</em></td>
<td>CE(p(\hat{x}', \theta), y)</td>
</tr>
<tr>
<td>ALP</td>
<td>CE(p(\hat{x}', \theta), y) + \lambda \cdot |p(\hat{x}', \theta) - p(x, \theta)|_2^2</td>
</tr>
<tr>
<td>CLP</td>
<td>CE(p(x, \theta), y) + \lambda \cdot |p(\hat{x}', \theta) - p(x, \theta)|_2^2</td>
</tr>
<tr>
<td>TRADES</td>
<td>CE(p(x, \theta), y) + \lambda \cdot KL(p(x, \theta)</td>
</tr>
<tr>
<td>MMA</td>
<td>CE(p(\hat{x}', \theta), y) \cdot 1(h_{\theta}(x) = y) + CE(p(x, \theta), y) \cdot 1(h_{\theta}(x) \neq y)</td>
</tr>
<tr>
<td>MART</td>
<td>BCE(p(\hat{x}', \theta), y) + \lambda \cdot KL(p(x, \theta)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Defense</th>
<th>MNIST</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Natural</td>
<td>FGSM</td>
</tr>
<tr>
<td><em>Standard</em></td>
<td>99.11</td>
<td>97.17</td>
</tr>
<tr>
<td>MMA</td>
<td>98.92</td>
<td>97.25</td>
</tr>
<tr>
<td>Dynamic</td>
<td>98.96</td>
<td>97.34</td>
</tr>
<tr>
<td>TRADES</td>
<td><strong>99.25</strong></td>
<td>96.67</td>
</tr>
<tr>
<td>MART</td>
<td>98.74</td>
<td><strong>97.87</strong></td>
</tr>
</tbody>
</table>

IMPROVING ADVERSARIAL ROBUSTNESS REQUIRES REVISITING MISCLASSIFIED EXAMPLES (Wang et. al)
Defenses: Robust generalization requires more data

- MNIST achieves >90% robustness
- Owing to learning thresholding filters
- CIFAR-10 achieves >45% robustness
- Gap between standard & robust generalization higher on CIFAR-10
- Owing to high dimensions

Adversarially Robust Generalization Requires More Data (Schmidt et. al)
Defenses: Data Augmentation & Unlabeled Extra Data

- Carmon et. al use \textbf{500k} unlabeled extra data
- Using extra data jumps robustness to \textbf{59\%}
- Rebuffi et. Al use data augmentations (CutMix)
- Achieving \textbf{66.56\%} robustness with \textbf{90.51\%} standard accuracy
Defenses: Effect of architecture on robustness

ResNet-18 Architecture

Layer-1
Output = 64 feature maps of size 32x32

Layer-2
Output = 128 feature maps of size 16x16

Layer-3
Output: 256 feature maps of size 8x8

Layer-4
Output: 512 feature maps of size 4x4

avg pool

fc 512
Defenses: Effect of architecture on robustness

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Robust Accuracy (%)</th>
<th>Natural Accuracy (%)</th>
<th>Perturbation Stability (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>width-1</td>
<td>width-5</td>
<td>width-10</td>
</tr>
<tr>
<td>6</td>
<td>47.81±.09</td>
<td>54.45±.16</td>
<td>54.18±.39</td>
</tr>
<tr>
<td>9</td>
<td>48.01±.06</td>
<td>55.34±.17</td>
<td>55.29±.45</td>
</tr>
<tr>
<td>12</td>
<td>47.87±.06</td>
<td>55.61±.04</td>
<td>55.98±.13</td>
</tr>
<tr>
<td>15</td>
<td>47.15±.13</td>
<td>55.49±.15</td>
<td>55.96±.09</td>
</tr>
<tr>
<td>18</td>
<td>47.02±.13</td>
<td>55.43±.12</td>
<td>56.43±.17</td>
</tr>
<tr>
<td>21</td>
<td>46.26±.19</td>
<td>55.31±.20</td>
<td>56.07±.21</td>
</tr>
</tbody>
</table>

**TRADES** Zhang et al. (2019)

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Robust Accuracy (%)</th>
<th>Natural Accuracy (%)</th>
<th>Perturbation Stability (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>width-1</td>
<td>width-5</td>
<td>width-10</td>
</tr>
<tr>
<td>1.00</td>
<td>47.99±.16</td>
<td>50.87±.42</td>
<td>50.12±.13</td>
</tr>
<tr>
<td>1.25</td>
<td>49.24±.12</td>
<td>53.10±.09</td>
<td>51.97±.46</td>
</tr>
<tr>
<td>1.50</td>
<td>49.11±.03</td>
<td>54.15±.03</td>
<td>53.25±.52</td>
</tr>
<tr>
<td>1.75</td>
<td>48.32±.63</td>
<td>54.36±.14</td>
<td>53.65±.80</td>
</tr>
<tr>
<td>2.00</td>
<td>47.44±.06</td>
<td>54.10±.15</td>
<td>55.78±.22</td>
</tr>
</tbody>
</table>

**Adversarial Training** Madry et al. (2018)

Wide residual networks. (Zagoruyko et. al 2017)
**Defenses: RobustBench (CIFAR-10)**

---

**Leaderboard: CIFAR-10, $\ell_\infty = 8/255$, untargeted attack**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Standard accuracy</th>
<th>AutoAttack robust accuracy</th>
<th>Best known robust accuracy</th>
<th>AA eval. potentially unreliable</th>
<th>Extra data</th>
<th>Architecture</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fixing Data Augmentation to Improve Adversarial Robustness</td>
<td>92.23%</td>
<td>66.58%</td>
<td>66.56%</td>
<td>×</td>
<td>✔</td>
<td>WideResNet-70-16</td>
<td>arXiv, Mar 2021</td>
</tr>
<tr>
<td>2</td>
<td>Improving Robustness using Generated Data</td>
<td>88.74%</td>
<td>66.11%</td>
<td>66.10%</td>
<td>×</td>
<td>×</td>
<td>WideResNet-70-16</td>
<td>NeurIPS 2021</td>
</tr>
<tr>
<td>3</td>
<td>Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples</td>
<td>91.10%</td>
<td>65.58%</td>
<td>65.87%</td>
<td>×</td>
<td>✔</td>
<td>WideResNet-70-16</td>
<td>arXiv, Oct 2020</td>
</tr>
</tbody>
</table>

---

https://robustbench.github.io/#leaderboard
### Defenses: RobustBench (ImageNet)

**Leaderboard: ImageNet, \( \ell_\infty = 4/255 \), untargeted attack**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>Standard accuracy</th>
<th>AutoAttack robust accuracy</th>
<th>Best known robust accuracy</th>
<th>AA eval. potentially unreliable</th>
<th>Extra data</th>
<th>Architecture</th>
<th>Venue</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Do Adversarily Robust ImageNet Models Transfer Better?</td>
<td>64.62%</td>
<td>34.96%</td>
<td>34.96%</td>
<td>✗</td>
<td>✗</td>
<td>ResNet-50</td>
<td>NeurIPS 2020</td>
</tr>
<tr>
<td>3</td>
<td>Robustness library</td>
<td>62.56%</td>
<td>29.22%</td>
<td>29.22%</td>
<td>✗</td>
<td>✗</td>
<td>ResNet-50</td>
<td>GitHub, Oct 2019</td>
</tr>
<tr>
<td>4</td>
<td>Fast is better than free: Revisiting adversarial training</td>
<td>55.62%</td>
<td>26.24%</td>
<td>26.24%</td>
<td>✗</td>
<td>✗</td>
<td>ResNet-50</td>
<td>ICLR 2020</td>
</tr>
</tbody>
</table>
Conclusion

• A lot of room for improvement

• Possible future work

• Our current work evaluates secret key based defenses and tries to improve robustness by making changes to the architecture