Efficient Defenses Against Adversarial Examples for Deep Neural Networks

Valentina Zantedeschi @vzantedesc
Jean Monnet University

Irina Nicolae @ririnicolae
IBM Research AI

Ambrish Rawat @ambrishrawat

GreHack #5
November 17, 2017
So far...

- Machine learning for security
  - Intrusion detection\(^1\)
  - Malware analysis\(^2\)

This talk is about

- Security for machine learning

---


Machine Learning and Adversarial Examples
Machine Learning

Training

Inputs

- e.g. picture

Expected Outputs

- e.g. class id

Prediction

Prediction Model

Training

Prediction Model

Bird
Adversarial Examples

- Perturb model inputs with crafted noise
- Model fails to recognize input correctly
- Attack undetectable by humans
- Random noise does not work.
Practical Examples of Attacks
Attack noise hides pedestrians from the detection system.

Car ends up ignoring the stop sign.

True image  Adversarial image

---

Executing Voice Commands

Okay Google, text John!\textsuperscript{5}

- Stealthy voice commands recognized by devices
- Humans cannot detect it.

\textsuperscript{5}Zhang et al., \textit{DolphinAttack: Inaudible Voice Commands}, ACM CCS 2017.
Deep Learning and Adversarial Samples
Deep Neural Networks

Deep Magic Box

Input
e.g picture

Deep Magic Box

Output
e.g. class id
Deep Neural Networks

- Interconnected layers propagate the information forward.
- Model learns weights for each neuron.
Deep Neural Networks

- Specific neurons light-up depending on the input.
- Cumulative effect of activation moves forward in the layers.
Small variations in the input → important changes in the output.

+ Enhanced discriminative capacities
- Opens the door to adversarial examples
The **learned model** slightly differs from the **true** data distribution...
... which makes room for adversarial examples.
• Most attacks try to move inputs across the boundary.
• Attacking with a random distortion doesn’t work well in practice.
Finding Adversarial Examples

Given $x$, find $x'$ where
- $x$ and $x'$ are close
- $\text{output}(x) \neq \text{output}(x')$

<table>
<thead>
<tr>
<th>Approximations of the original problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGSM [1]</td>
</tr>
<tr>
<td>Random + FGSM [2]</td>
</tr>
<tr>
<td>DeepFool [3]</td>
</tr>
<tr>
<td>JSMA [4]</td>
</tr>
<tr>
<td>C&amp;W [5]</td>
</tr>
<tr>
<td>quick, rough, fixed budget</td>
</tr>
<tr>
<td>random step, then FGSM</td>
</tr>
<tr>
<td>find minimal perturbations</td>
</tr>
<tr>
<td>modify most salient pixels</td>
</tr>
<tr>
<td>strongest to date</td>
</tr>
</tbody>
</table>
• Adapt the classifier to attack directions by including adversarial data at training.
Defense: Adversarial Training

- Adapt the classifier to attack directions by including adversarial data at training.
- But there are always new adversarial samples to be crafted.
### Defenses

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>data augmentation train also with adv. examples</td>
</tr>
<tr>
<td>VAT</td>
<td>data augmentation train also with virtual adv. examples</td>
</tr>
<tr>
<td>FS</td>
<td>preprocessing squeeze input domain</td>
</tr>
<tr>
<td>LS</td>
<td>preprocessing smooth target outputs</td>
</tr>
</tbody>
</table>

- Adversarial Training (AT) [1]
- Virtual Adversarial Training (VAT) [6]
- Feature Squeezing (FS) [7]
- Label Smoothing (LS) [8]
Contribution: Effective Defenses Against Adversarial Samples
Gaussian Data Augmentation (GDA)

Gaussian noise does not work for attacks, but does it work as a defense?

- Reinforce neighborhoods around points using random noise.
- For each input image, generate $N$ versions by adding Gaussian noise to the pixels.
- Train the model on the original data and the noisy inputs.
**Objective** Limit the cumulative effect of errors in the layers.

\[
f(x) = \begin{cases} 
0, & x < 0 \\
x, & x \geq 0.
\end{cases}
\]
**Objective** Limit the cumulative effect of errors in the layers.

**ReLU**

\[
f(x) = \begin{cases} 
0, & x < 0 \\
x, & x \geq 0.
\end{cases}
\]

**Bounded RELU**

\[
f_t(x) = \begin{cases} 
0, & x < 0 \\
x, & 0 \leq x < t \\
t, & x \geq t.
\end{cases}
\]
Comparison with Other Defenses

<table>
<thead>
<tr>
<th>Defense</th>
<th>Training</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Squeezing</td>
<td>preproc. input</td>
<td>preproc. input, perf. loss</td>
</tr>
<tr>
<td>Label Smoothing</td>
<td>preproc. output</td>
<td>-</td>
</tr>
<tr>
<td>Adversarial Training</td>
<td>train + attack + retrain</td>
<td>-</td>
</tr>
<tr>
<td>GDA + BRELU</td>
<td>add noise</td>
<td>-</td>
</tr>
</tbody>
</table>

**Advantages of GDA + BRELU**

- Defense agnostic to attack strategy
- Model performance for original inputs is conserved
- Performs better than other defenses on adversarial samples
- Almost no overhead for training and prediction.
Experiments
Setup

- MNIST dataset of handwritten digits
  - 60,000 training + 10,000 test images
- CIFAR-10 dataset of $32 \times 32$ RGB images
  - 50,000 training + 10,000 test images
  - 10 categories
- Convolutional neural net (CNN) architecture
Setup

Threat model

- **Black-box**: attacker has access to inputs and outputs
- **White-box**: attacker also has access to model parameters

Steps

- Train model with different defenses
- Generate attack images
- Compute defense performance on attack images
<table>
<thead>
<tr>
<th>FGSM</th>
<th>DeepFool</th>
<th>JSMA</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

With GDA + BRELU, the perturbation necessary for an attack becomes **visually detectable**.
Comparison of different defenses against white-box attacks

(a) FGSM attack

(b) Random + FGSM attack

CIFAR-10

Accuracy = % of correct predictions = TP + TN
## Black-Box Attacks

### Comparison of different defenses against black-box attacks

<table>
<thead>
<tr>
<th>Defense</th>
<th>Attack</th>
<th>FGSM</th>
<th>Rand+FGSM</th>
<th>DeepFool</th>
<th>JSMA</th>
<th>C&amp;W</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>FGSM</td>
<td>94.46</td>
<td>40.70</td>
<td>92.95</td>
<td>97.95</td>
<td>93.10</td>
</tr>
<tr>
<td></td>
<td>Feature squeezing</td>
<td>96.31</td>
<td>91.09</td>
<td>96.68</td>
<td>97.48</td>
<td>96.75</td>
</tr>
<tr>
<td></td>
<td>Label smoothing</td>
<td>86.79</td>
<td>20.28</td>
<td>84.58</td>
<td>95.86</td>
<td>84.81</td>
</tr>
<tr>
<td></td>
<td>FGSM adv. training</td>
<td>91.86</td>
<td>49.77</td>
<td>85.91</td>
<td>98.62</td>
<td>97.71</td>
</tr>
<tr>
<td>VAT</td>
<td></td>
<td>97.53</td>
<td>74.35</td>
<td>96.03</td>
<td>98.26</td>
<td>96.11</td>
</tr>
<tr>
<td>GDA + RELU</td>
<td></td>
<td>98.47</td>
<td>80.25</td>
<td>97.84</td>
<td>98.96</td>
<td>97.87</td>
</tr>
<tr>
<td>GDA + BRELU</td>
<td></td>
<td>98.08</td>
<td>75.50</td>
<td>98.00</td>
<td>98.88</td>
<td>98.03</td>
</tr>
</tbody>
</table>

Attacks transferred from ResNet to CNN on MNIST

Accuracy = % of correct predictions = TP + TN
Adversarial Attacks and Defenses

Fast-Gradient Sign Method

- Choose an image
- Fix the perturbation size
- Submit!

choose model

UPLOAD IMAGE

5

SUBMIT
Conclusion
Conclusion

Our contribution

- Improved defense against multiple types of attacks
- Model performance for clean inputs is preserved
- No retraining, no overhead for prediction
- Easy to integrate into models.

Takeaway

- The problem of adversarial examples needs to be solved before applying machine learning.

nemesis

- Our library of attacks and defenses
- Soon to be open source.


