Provably Minimally-Distorted Adversarial Examples

N. Carlini\textsuperscript{1}, G. Katz\textsuperscript{2}, C. Barrett\textsuperscript{3}, D. Dill\textsuperscript{4}

\textsuperscript{1}University of California, Berkeley \textsuperscript{2}Stanford University

arXiv: 1709.10207
Reviewed by: Bill Zhang
University of Virginia
https://qdata.github.io/deep2Read/
Outline

Introduction

Background and Notation

Model Setup

Evaluation

Conclusion

References
Introduction
Basic Premise and Motivation

- Over half of proposed defenses against adversarial examples for ICLR 2018 have already been broken
- In recent years, people have proposed methods to formally verify neural networks; take an network and formally prove that it satisfies a certain property (or provide a counterexample)
- Propose a method to formally verify effectiveness of adversarial attacks and defenses; apply verification to construct provably minimally-distorted examples
Introduction

Types of Evaluation

- Attack evaluation: Use provably minimally-distorted examples and compare to an attack’s example to evaluate efficacy of an attack
- Defense evaluation: Observe how applying a certain defense affects how distorted minimally-distorted example is; proof vs empirical observations
Neural networks: Multilayer network $F = F_n \circ F_{n-1} \circ ... \circ F_1 \circ F_0$ where $F_n$, the final layer, is a softmax activation; output of second to last layer is logits $Z = F_{n-1} \circ ... \circ F_1 \circ F_0$

$l_F(x, y)$ is cross-entropy loss of $F$ on input $x$ with label $y$

Focus on greyscale MNIST, which have inputs of form $[0, 1]^{WH}$

Adversarial examples: Given $x$ classified as $t$, find $x'$ which produces target $t'$ where $x$ is close to $x'$ using some distance measurement: for consistency, use $L_1$ and $L_\infty$
Background and Notation

Example Generation

- Fast Sign Method (FSM): one-step algorithm, 
  \[ x' = FGM(x) = \text{clip}_{[0,1]}(x + \epsilon \text{sign}(\nabla l_F(x, y))) \]

- Basic Iterative Method (BIM) or PGD: iterative application of 
  FGM, 
  \[ x_{i+1}' = \text{clip}_{[x-\alpha,x+\alpha]}(FGM(x_i')) \]

- Carlini and Wagner Method (CW): iterative attack which 
  constructs examples by approximately solving \( \min d(x, x') \) 
  such that \( F(x') = t' \) where \( d \) is the distance metric; to make 
  easier, instead use \( \min d(x, x') + cg(x') \) where \( g(x') \) encodes 
  how close to adversarial \( x' \) is

  \[ g(x') = \max (\max \{ Z(x')_i : i \neq t \} - Z(x')_t, 0) \]
Focus on recently proposed Reluplex algorithm (Katz et al., 2017b)

Simplex-based approach that effectively tackles networks with piecewise-linear activation functions (ReLUs) or max-pooling layers

Reluplex can be used to determine whether there exists an adversarial example within $\delta$ of $x$; done by encoding neural network and constraints regarding $\delta$ as a set of linear equations and ReLU constraints

By using Reluplex iteratively like binary search, can approximate optimal $\delta$
Background and Notation

Current Focus

- Current work is focused on adversarial training and provable (certified) defenses
- Downside to certified defenses is that it only works for small networks with small datasets
- This work can take an arbitrary defense and prove properties about it on a small dataset
- Also has limitation of not scaling to large datasets
Model Setup

- Neural network verification is NP-complete; only networks with a few hundred nodes can be soundly verified
- Use fully-connected, 3-layer network with only 20k weights and 100 hidden neurons for MNIST
- Use proof-of-concept implementation of Reluplex online; only non-linear function it can support is ReLU function
- Modify to support max operators; allows for support of max-pooling layers

\[
\max(x, y) = \text{ReLU}(x - y) + y
\]

- Also, modify to support absolute values to compute distances for \( L_1 \) and \( L_\infty \)

\[
|x| = \max(x, -x) = \text{ReLU}(2x) - x
\]

- Increase in ReLU constraints slowed performance
Each experiment included network $F$, distance metric $d \in \{L_1, L_\infty\}$, input $x$, target label $l' \neq F(x)$, and initial adv. input $x'_{\text{init}}$ where $F(x'_{\text{init}}) = l'$

Use ReLU search to find bounds $\delta_{\text{min}}$ and $\delta_{\text{max}}$ on optimal $\delta$; initialize $\delta_{\text{min}} = 0$ and $\delta_{\text{max}} = x'_{\text{init}}$

For $x'_{\text{init}}$, use example generated using CW method

$L_1$ initial distances typically much larger, which made Reluplex slower
Evaluation

- Arbitrarily pick 10 source images with known labels from MNIST test set
- Consider two networks: one as described previously, $N$, another with adversarial training, $\tilde{N}$
- Also consider both $L_1$ and $L_\infty$
- For every combination of network, distance metric, and source image $x$, consider each of other 9 labels for $x$; use CW to make targeted attack and produce initial example, then use Reluplex to generate minimally-distorted example
Evaluation

- First sub-row: successfully terminated Reluplex, Second sub-row: all experiments (incl. timeouts); distances are averages
- Naturally, results only hold for the specific networks and inputs, but can be used to provide intuition on performance

<table>
<thead>
<tr>
<th></th>
<th>Number of Points</th>
<th>Carlini-Wagner</th>
<th>Minimally Distorted Adversarial Example</th>
<th>Percent Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N, L_\infty )</td>
<td>38/90</td>
<td>0.042</td>
<td>0.038</td>
<td>11.632</td>
</tr>
<tr>
<td></td>
<td>90/90</td>
<td>0.063</td>
<td>0.061</td>
<td>6.027</td>
</tr>
<tr>
<td>( N, L_1 )</td>
<td>6/90</td>
<td>1.94</td>
<td>1.731</td>
<td>34.909</td>
</tr>
<tr>
<td></td>
<td>90/90</td>
<td>7.551</td>
<td>7.492</td>
<td>3.297</td>
</tr>
<tr>
<td>( \bar{N}, L_\infty )</td>
<td>81/90</td>
<td>0.211</td>
<td>0.193</td>
<td>11.637</td>
</tr>
<tr>
<td></td>
<td>90/90</td>
<td>0.219</td>
<td>0.203</td>
<td>10.568</td>
</tr>
<tr>
<td>( \bar{N}, L_1 )</td>
<td>64/90</td>
<td>6.44</td>
<td>6.36</td>
<td>6.285</td>
</tr>
<tr>
<td></td>
<td>90/90</td>
<td>8.187</td>
<td>8.128</td>
<td>4.486</td>
</tr>
</tbody>
</table>
Evaluation
Evaluating Attacks

- Iterative attacks like CW produce near-optimal examples
- There is, however, still room to improve iterative attacks: ground-truth adversarial examples frequently had 30-40% less distortion than best iterative example; happens because PGD finds local, not global minimum
- If iterative attack performs poorly on one target label, it will tend to perform poorly on others too; frequently, gradient descent leads away from target towards inferior local minimum
Evaluation
Evaluating Defenses

▶ To evaluate Madry et al., only consider $L_\infty$ cases because too few $L_1$ Reluplex searches terminated; only consider subset of 35 cases which converged for both $N$ and $\tilde{N}$

<table>
<thead>
<tr>
<th></th>
<th>Number of Points</th>
<th>CW</th>
<th>Minimally Distorted</th>
<th>Percent Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N, L_\infty$</td>
<td>35/35</td>
<td>0.042</td>
<td>0.039</td>
<td>12.319</td>
</tr>
<tr>
<td>$\tilde{N}, L_\infty$</td>
<td>35/35</td>
<td>0.18</td>
<td>0.165</td>
<td>11.153</td>
</tr>
</tbody>
</table>
Evaluation
Evaluating Defenses

- Adversarial training from Madry et al. is effective; increases minimally-distorted distance from average of 0.039 to 0.165 (423% increase)
- 7 out of 35 experiments, however, actually had smaller minimal distances after adversarial training compared to original network (average 12.8% decrease)
- Highlights necessity to evaluate defenses against large sets of data
Evaluation
Evaluating Defenses

- Training on iterative attacks does not overfit
- Easier to formally analyze Madry et al.: Reluplex terminated on significantly more experiments after adversarial training
- Unsure as to why; not because adversarially trained network makes use of less ReLU units since there is no statistical difference in use of ReLU units
Conclusion

- Neural networks have great potential for safety-critical systems, but susceptibility to adversarial examples is a great hindrance
- Introduce provably minimally-distorted examples and show how to construct with formal verification approaches
- Showed that Carlini and Wagner produced examples very close to minimally-distorted and that Madry et. al. provably increased robustness of network; to their knowledge, first proof of robustness for a defense not designed to be proven secure
- Current verification techniques are limited to small networks; limitation expected to be lifted in the future
- Also, networks can be designed to be more amenable to verification
References