Evaluating Robustness of Neural Nets

Presented by: Zacchaeus Scheffer and Akash Kumar
Summary

- Survey existing attacks
- Approach
- Three Attacks
- Attack Evaluation
- Defensive Distillation
Summary: Terminology

Neural Network:
\[ F(x) = \text{softmax}(Z(x)) \]
Z(x) is the network excluding the softmax at the end.

L_p norm gives us a distance
- L_0 -> number of pixels changed
- L_\infty -> max amount change in a pixel
- L_2 \rightarrow \quad d(x, x_i') = \sum_p (\delta_i)^2_p

\[
\text{softmax}(x)_k = \frac{e^{x_k}}{\sum_i^n e^{x_i}}
\]

- x: clean image
- x_i': adversarial image that gets mapped to incorrect class i
- x_i' = x + \delta_i
Summary: Terminology (cont.)

Targeted Attack Metrics

- **Average Case:**
  - Mean distance to the adversarial image of each incorrect class
  - \[ \frac{1}{n-1} \sum_{i \neq t} d(x, x'_i) \]
  - \( n = \) number of classes, \( t = \) true class

- **Best Case:**
  - The smallest distance to an adversarial image of all classes
  - \( \min_i d(x, x'_i) \)

- **Worst Case:**
  - The largest distance to an adversarial image of all classes
  - \( \max_i d(x, x'_i) \)
Existing Attacks:

- Box-constrained L-BFGS (Szegedy et al.)
- Fast Gradient Sign (FGS) (Goodfellow et al.)
- Jacobian-based Saliency Map Attack (JSMA) (Papernot et al.)
- Deepfool (M. Dezfooli et al.)
Box-constrained L-BFGS

- White-box, slow, targeted
- Distance metric: $L_2$
- Optimization function:

$$\text{minimize} \quad c \cdot \|x - x'\|_2^2 + \text{loss}_{F,l}(x')$$

such that $x' \in [0, 1]^n$

- $x_i$: clean image
- $x_i'$: adversarial image
- n: number of images
Fast Gradient Sign (FGS)

- White-box, fast, targeted
- Distance metric: $L_{\infty}$

$$x' = x - \epsilon \cdot \text{sign}(\nabla \text{loss}_{F,t}(x))$$

- Iterative Gradient Sign
  - Multiple smaller steps

$$x'_i = x'_{i-1} - \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla \text{loss}_{F,t}(x'_{i-1})))$$

- $x_i$: clean image
- $x'_i$: adversarial image
- $t$: target image
- $\epsilon$: constant
Jacobian-based Saliency Map Attack (JSMA)

- White-box, slow, targeted
- Greedy algorithm
- Saliency map Generation
- Two variants:
  - JSMA-F
  - JSMA-Z
- Saliency map:

\[
\alpha_{pq} = \sum_{i \in \{p, q\}} \frac{\partial Z(x)_t}{\partial x_i}
\]

\[
\beta_{pq} = \left( \sum_{i \in \{p, q\}} \sum_j \frac{\partial Z(x)_j}{\partial x_i} \right) - \alpha_{pq}
\]

- \(\alpha_{pq}\): change in pixels p and q
- \(\beta_{pq}\): change in all other outputs
DeepFool

- White-box, fast, untargeted
- Distance metric: $L_2$
- Optimization function:

$$\Delta(x; \hat{k}) := \min_r \|r\|_2 \text{ subject to } \hat{k}(x + r) \neq \hat{k}(x)$$

- $x$: clean image
- $k(x)$: estimated label
- $r$: perturbation required
New approach

Start with standard formulation:

\[
\begin{align*}
\text{minimize} & \quad \mathcal{D}(x, x + \delta) \\
\text{such that} & \quad f(x + \delta) \leq 0 \\
& \quad x + \delta \in [0, 1]^n
\end{align*}
\]

Reformulate as:

\[
\begin{align*}
\text{minimize} & \quad \|\delta\|_p + c \cdot f(x + \delta) \\
\text{such that} & \quad x + \delta \in [0, 1]^n
\end{align*}
\]

Where we minimize “c” s.t. resulting solution yields \( f(x+\delta) \leq 0 \)
Objective Functions Explored

\[ f_1(x') = -\text{loss}_{F,t}(x') + 1 \]
\[ f_2(x') = (\max_{i \neq t} F(x')_i - F(x')_t)^+ \]
\[ f_3(x') = \text{softplus}(\max_{i \neq t} F(x')_i - F(x')_t) - \log(2) \]
\[ f_4(x') = (0.5 - F(x')_t)^+ \]
\[ f_5(x') = -\log(2F(x')_t - 2) \]
\[ f_6(x') = (\max_{i \neq t} Z(x')_i - Z(x')_t)^+ \]
\[ f_7(x') = \text{softplus}(\max_{i \neq t} Z(x')_i - Z(x')_t) - \log(2) \]

Notations:
- \((e)^+ : \max(e, 0)\)
- \(\text{softplus}(x) : \log(1 + \exp(x))\)
- \(\text{loss}_{F,s}(x) : \text{cross-entropy loss}\)

\[ f(x + \delta) \leq 0 \]
\[ \text{Bes} F(x) = \text{softmax}(Z(x)) \]
Dealing with Box Constraints: $x + \delta \in [0, 1]$

- Projected Gradient Descent
  - Clips all the coordinates
    \[ y_{k+1} = x_k - t_k \nabla f(x_k) \]
    \[ x_{k+1} = \arg \min_{x \in C} \| y_{k+1} - x \| \]
  - Con: Alters the input

- Clipped Gradient Descent
  - Clips objective function
    \[ f(\min(\max(x + \delta, 0), 1)) \]
  - Con: Gradient becomes zero
Dealing with Box Constraints

- Change of variables:
  - tanh function

\[
\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x_i \quad \text{such that} \quad x_i + \delta_i \in [0, 1]
\]
Parameters Space:

• Objective Function

• Box Constraints

• Constant $c$

minimize $\|\delta\|_p + c \cdot f(x + \delta)$

such that $x + \delta \in [0, 1]^n$
Finding Best Combination

- Brute force: Permute(box constraint, objective function($f_n$))
- Minimize $c$ with binary search
- Report mean $L_2$ norm of (successful) attack, and probability of successful attack.
- Dataset: MNIST

<table>
<thead>
<tr>
<th></th>
<th>Worst Case</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Change of Variable mean</td>
<td>prob</td>
<td>Clipped Descent mean</td>
<td>prob</td>
<td>Projected Descent mean</td>
</tr>
<tr>
<td>$f_1$</td>
<td>7.76</td>
<td>100%</td>
<td>9.48</td>
<td>100%</td>
<td>7.37</td>
</tr>
<tr>
<td>$f_2$</td>
<td>2.93</td>
<td>18%</td>
<td>10.22</td>
<td>40%</td>
<td>18.90</td>
</tr>
<tr>
<td>$f_3$</td>
<td>3.09</td>
<td>17%</td>
<td>11.91</td>
<td>41%</td>
<td>24.01</td>
</tr>
<tr>
<td>$f_4$</td>
<td>3.55</td>
<td>24%</td>
<td>4.25</td>
<td>35%</td>
<td>4.10</td>
</tr>
<tr>
<td>$f_5$</td>
<td>6.42</td>
<td>100%</td>
<td>7.86</td>
<td>100%</td>
<td>6.12</td>
</tr>
<tr>
<td>$f_6$</td>
<td>6.03</td>
<td>100%</td>
<td>7.50</td>
<td>100%</td>
<td>5.89</td>
</tr>
<tr>
<td>$f_7$</td>
<td>6.20</td>
<td>100%</td>
<td>7.57</td>
<td>100%</td>
<td>5.94</td>
</tr>
</tbody>
</table>
Different Attacks

- L$_2$ attack:
  - Objective function:

    minimize $\frac{1}{2}(\tanh(w) + 1) - x + c \cdot f(\frac{1}{2}(\tanh(w) + 1))$

  - Multiple starts

    minimize $\|\delta\|_p + c \cdot f(x + \delta)$
    such that $x + \delta \in [0, 1]^n$
Different Attacks (Cont.)

- $L_0$ attack:
  - Non-differentiable metric
  - Iterative approach
  - Use of $L_2$ attack
  - Eliminate pixels

Notes:
- JSMA: Include pixels
Different Attacks (Cont.)

- $L^\infty$ attack:
  - Not-fully differentiable

\[
\text{minimize } c \cdot f(x + \delta) + \|\delta\|_\infty
\]
  - Stuck in oscillation
  - Again iterative attack

\[
\text{minimize } c \cdot f(x + \delta) + \sum_i [(\delta_i - \tau)^+]
\]
Attack Evaluation

- $L_0$ and $L_2$ attacks:
  - 2-10 times lower distortions
  - Success rate: 100%

- $L_\infty$ attack:
  - Similar perturbation, but higher success rate
  - Can change any image to any label only changing last bit of each pixel

- Previous attacks:
  - Model complexity $\uparrow = $ effectiveness $\downarrow$

- This attack:
  - Model complexity $\uparrow = $ effectiveness $\uparrow$
## Attacks on ImageNet

<table>
<thead>
<tr>
<th></th>
<th>Untargeted</th>
<th></th>
<th>Average Case</th>
<th></th>
<th>Least Likely</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>prob</td>
<td>mean</td>
<td>prob</td>
<td>mean</td>
<td>prob</td>
</tr>
<tr>
<td><strong>Our (L_0)</strong></td>
<td>48</td>
<td>100%</td>
<td>410</td>
<td>100%</td>
<td>5200</td>
<td>100%</td>
</tr>
<tr>
<td>JSMA-Z</td>
<td>-</td>
<td>0%</td>
<td>-</td>
<td>0%</td>
<td>-</td>
<td>0%</td>
</tr>
<tr>
<td>JSMA-F</td>
<td>-</td>
<td>0%</td>
<td>-</td>
<td>0%</td>
<td>-</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Our (L_2)</strong></td>
<td>0.32</td>
<td>100%</td>
<td>0.96</td>
<td>100%</td>
<td>2.22</td>
<td>100%</td>
</tr>
<tr>
<td>Deepfool</td>
<td>0.91</td>
<td>100%</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Our (L_\infty)</strong></td>
<td>0.004</td>
<td>100%</td>
<td>0.006</td>
<td>100%</td>
<td>0.01</td>
<td>100%</td>
</tr>
<tr>
<td>FGS</td>
<td>0.004</td>
<td>100%</td>
<td>0.064</td>
<td>2%</td>
<td>-</td>
<td>0%</td>
</tr>
<tr>
<td>IGS</td>
<td>0.004</td>
<td>100%</td>
<td>0.01</td>
<td>99%</td>
<td>0.03</td>
<td>98%</td>
</tr>
</tbody>
</table>
Defensive Distillation

- State of the art method to increase robustness
- Reduces attack success probability from 95% to 0.5% (for old attack methods)
- Almost no effect on this new method:

<table>
<thead>
<tr>
<th>Worst Case</th>
<th>MNIST</th>
<th>CIFAR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>prob</td>
</tr>
<tr>
<td>Our $L_0$</td>
<td>36</td>
<td>100%</td>
</tr>
<tr>
<td>Our $L_2$</td>
<td>2.9</td>
<td>100%</td>
</tr>
<tr>
<td>Our $L_\infty$</td>
<td>0.25</td>
<td>100%</td>
</tr>
</tbody>
</table>
Pros

• Achieved 100% attack success rate for all three metrics ($L_{0,2,\infty}$)
• 100% success on all three datasets: MNIST, CIFAR and ImageNet
• Simultaneously had smaller average perturbation
• Additionally worked on “robust” networks trained using the state of the art robustness training: defensive distillation.
• Perturbed images always look like the original class
• Junk images perturbed to a class still look like junk
• L0 attack in this paper -> First targeted attack on ImageNet
Cons

- Did not perform speed tests, so we don’t know quickly it can be done.
  - They state "no attack takes longer than a few minutes on any instance"
- L0 attack is dependent upon L2 attack.
Summary

- The paper introduces three new attacks or all distance metrics. They also show that their attacks are more effective than previous attacks.
- The attack proposed by this paper defeat the strongest defense: Defensive distillation
- The authors discussed about various parameters that can impact the efficacy of an attack
- They propose high-confidence adversarial examples for transferability test, and show that its break defensive distillation.