Evaluating the Robustness of Neural Networks Defenses

Nicholas Carlini

Google Brain
Background: Notation

• We have a classification neural network $F(x)$
• Given an input $X$ classified as label $L$
• $F(X)_L$ is the probability of label $L$
• $F(X) = \text{softmax}(F_Z(X))$ [the logits]
• $C(X) = \arg \max_j F(X)_j$
Background: Adversarial Examples

• For a classification neural network $F(x)$
• Given an input $X$ classified as label $L$ ...
• ... it is easy to find an $X'$ close to $X$
• ... so that $F(X') \neq L$
$$x + 0.007 \times \text{sign}(\nabla_x J(\theta, x, y))$$

```
"panda"
57.7% confidence
```

```
"nematode"
8.2% confidence
```

```
\epsilon \text{sign}(\nabla_x J(\theta, x, y))
"gibbon"
99.3% confidence
```
Why should we care about adversarial examples?

Make ML robust

Make ML better
Finding Adversarial Examples

• Formulation: given input x, find $x'$ where
  minimize $d(x,x')$
  such that $F(x') = T$
  $x'$ is "valid"

• Gradient Descent to the rescue?

• Non-linear constraints are hard
Reformulation

• Formulation:
  minimize  \( d(x,x') + g(x') \)
  such that  \( x' \) is "valid"

• Where \( g(x') \) is some kind of loss function on how close \( F(x') \) is to target \( T \)
  • \( g(x') \) is small if \( F(x') = T \)
  • \( g(x') \) is large if \( F(x') \neq T \)
Reformulation

• For example
  • $g(x') = (1-F(x')_T)$
  
  • If $F(x')$ says the probability of $T$ is 1:
    • $g(x') = (1-F(x')_T) = (1-1) = 0$
  
  • $F(x')$ says the probability of $T$ is 0:
    • $g(x') = (1-F(x')_T) = (1-0) = 1$
Does this work?

Problem 1:
Global minimum is not an adversarial example
Does this work?

- Formulation:
  minimize \( \frac{d(x, x')}{5} + g(x') \)
  such that \( x' \) is "valid"

\[
\frac{d(x, x')}{5} + g(x') = \square
\]
Does this work?

• Problem 2: Gradient direction does not point toward the global minimum

\[ \frac{d(x,x')}{5} + g(x') \]

\[
\begin{array}{c}
\text{y} \\
\text{0} & \text{1} & \text{2} \\
\text{x} & \text{0} & \text{1} & \text{2}
\end{array}
\]
Does this work?

Problem 3:
Global minimum is not the minimally perturbed adversarial example

\[
d(x,x')/1e10 + g(x') = 0
\]
This is very hard.

Let's do something simpler.
Fast Gradient Sign

- Unroll the gradient descent step by one

\[ X' = X + \epsilon \ \text{sign}(\nabla_X F(X)_L) \]
How can we stop adversarial examples?
Distillation as a Defense

1. Train a model $F()$ on the training data $X,Y$

2. Generate new training labels $Y'$ by setting $Y' = \{100 \cdot F(x) : x \in X\}$

3. Train a new classifier $G()$ on $X,Y'$
Does it work?
Unfortunately, no.
Constructing a better loss function

1. Global minimum at the decision boundary
2. Gradient points towards the global minimum
Improved Formulation

• Formulation:
  minimize \( d(x, x') + g(x') \)
  such that \( x' \) is "valid"

\[
d(x, x') + g(x')
\]
Visualizations
Case studies on evaluating defenses to adversarial examples
Defense Idea #1: Additional Neural Network Detection

Normal Classifier
Normal Classifier
Detector & Classifier

Detector

Classifier
Detector & Classifier

[Diagram with boxes labeled 'Detector' and 'Classifier', and a red 'no' symbol]
Training an adversarial example detector
Normal Training

(7, 7)
(3, 3)
Detection Training (1)

(3, 3)

(7, n)

(3, n)

(7, 7)

Attack
Detection Training (2)

(7, y)

(8, y)

(7, n)

(8, n)
Sounds great.
Sounds great.

But we already know it's easy to fool neural networks …
... so just construct adversarial examples to

1. be misclassified
2. not be detected
Breaking Detection
Adversarial Training

• minimize \( d(x, x') + g(x') \)
such that \( x' \) is "valid"

• Old: \( g(x') \) measures loss of classifier on \( x' \)
Breaking Detection
Adversarial Training

• minimize $d(x,x') + g(x') + h(x')$
such that $x'$ is "valid"

• Old: $g(x')$ measures loss of classifier on $x'$

• New: $h(x')$ measures loss of detector on $x'$
Original

Adversarial (unsecured)

Adversarial (with detector)
2018
Defense Idea #2:

Thermometer Encoding

Problem:
Neural Networks are "overly linear"
Thermometer Encoding

- Break linearity by changing input representation
- $T(0.13) = 110000000000$
- $T(0.66) = 11111100000$
- $T(0.97) = 111111111111$
Standard Neural Network
With Thermometer Encoding

7  T  F
Claims:

On CIFAR, with distortion 8/255, accuracy of 50%

(compared to 0%)
Unfortunately, thermometer encoding only causes gradient descent to fail
"Fixing" Gradient Descent

[0.1, 0.3, 0.0, 0.2,
Defense Idea #3:

Adversarial Retraining

Adversarial Training

- Given training data \((X, Y)\)
- Sample a minibatch \((x, y)\)
- Generate the adversarial minibatch \((x', y)\)
- Train on \((x', y)\)
- Repeat until convergence
Audio has these same issues, too

"now I would drift gently off to dream land"
[adversarial]
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity
original or adversarial?
original or adversarial?
On audio, traditional ML methods are not vulnerable to adversarial examples.
Questions?

Nicholas Carlini
https://nicholas.carlini.com
nicholas@carlini.com