Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods

Nicholas Carlini and David Wagner

Presented by Kevin Lee, Shrijesh Siwakoti and Ananth Chillarige

Introduction

Current research has aimed to properly classify adversarial examples in machine learning, but could not achieve that mark. Most research shifted to trying to detect adversarial examples from natural images in their work, but have found little success in building a robust defense model.

This paper is proposing a **new guideline for adversarial machine learning model defenses** because the current evaluations are not as robust as some of these papers claim them to be:

- Most models are specific to a certain dataset (MNIST)
- Most defense models claim to be robust but only train on two different weak attacks
- Most models cannot withstand full black box attacks, none can withstand a white box attack and most fail on semi-black box attacks
- Different defense detection models focus on rudimentary concepts that do not establish robustness in proper detection

Notation

F (·) = Neural network used for classification

All networks are feed-forward neural networks of layers $\mathbf{F^i}$

$\mathbf{F(x)_i}$ = The probability that object **x** is labeled with class **i**

All outputs are known as logits, **Z(·)**

Some model layers use ReLU activation: $\mathbf{F_i\ (x) = ReLU(A_i \cdot F_{i-1}\ (x) + b_i\ )}$

F(x) = softmax(Z(x))

C(x), classification of **F(·)** on **x** is represented by:

$$\mathbf{C(x) = arg\ max_i\ (F(x)_i)}$$

Notation (cont.)

D is noted to be the Detection Neural Network for adversarial examples

G(·) is denoted as the function for creating the adversarial examples using C&W L2 attack

Carlini & Wagner Attack:

The attack uses gradient descent to solve minimize $\boldsymbol{// x' - x //_2^2 + c \cdot \ell(x')}$

x’ = adversarial example, x = natural image

c = constant determined by binary search

ℓ(x') is the loss function described:

$\boldsymbol{\ell(x') = \max(\max\{Z(x')_i : i , t \} - Z(x')_t , -\kappa)}$

k = confidence of adversarial example

Datasets

CIFAR-10:

- Computer vision dataset used for object recognition
- Contains 60,000 32x32 color images containing one of 10 object classes
- 6000 images per class
- Images range from airplanes, automobiles to animals such as dogs, bird, etc.

MNIST

- Database of handwritten digits
- Training set of 60,000 examples
- Test set of 10,000 examples
- The digits have been size-normalized and centered in a fixed-size image

Generating Adversarial Examples: L2 - C&W

L2 attack algorithm from Carlini & Wagner

High level: Iterative attack done while constructing adversarial examples

Uses gradient descent to solve: $\| x' - x \|_2^2 + c \cdot \ell(x')$

Generating Adversarial Examples: L2 - C&W (cont.)

Intuition on attack algorithm: $\| \mathbf{x}' - \mathbf{x} \|_2^2 + \mathbf{c} \cdot \boldsymbol{\ell}(\mathbf{x}')$

constant c: chosen by binary search

- Ex: c is too small: distance function ($\| \mathbf{x}' - \mathbf{x} \|_2^2$) dominates and the optimal solution will not have a different label
- c is too large: the objective term ($\mathbf{c} \cdot \boldsymbol{\ell}(\mathbf{x}')$) dominates and the adversarial example will not be nearby

Generating Adversarial Examples: L2 - C&W (cont.)

Loss function: $\boldsymbol{\ell(x') = \max(\max\{Z(x')_i : i \mathrel{!}= t\} - Z(x')_t\ , -\kappa)}$

Intuition behind loss function:

$\boldsymbol{\max(\max\{Z(x')_i : i \mathrel{!}= t\}}$ is used to compare target class t with next-most-likely class

- This gets minimized when the target class is significantly more likely than the second most likely class (unwanted behavior in the model)
- This is fixed by taking the maximum between this and -κ which controls the confidence of the adversarial example
  - Ex: κ = 0 is low confidence as is only classified as the target class
  - As κ increases, the model will classify the adversarial example as more likely (more confident)

Generating Adversarial Examples: L2 - C&W (cont.)

The loss function operates of logits **Z(·)**, not over probabilities F

- constant c ~ $1/|\nabla \ell|$
- If F were used instead of Z, no “good” constant c would exist since f varies by many different contributing factors and Z usually only varies by one

Threat Model

Zero-Knowledge

Perfect-Knowledge

Limited-Knowledge

Threat Model: Zero-Knowledge

Adversary generates adversarial examples on the unsecured model F, and is not aware that the detector D is in place. The detector is successful if it can detect these adversarial examples.

- Weakest threat model
- The attacker is not even aware that a defense is in place
  - Generate examples with C&W L2, and check if defense can detect
- **Failing this test should imply that the other two tests will fail**

Threat Model: Perfect-Knowledge

Adversary is aware the neural network is being secured with a given detection scheme D, knows the model parameters used by D, and can use these to evade both original network F and the detector simultaneously

- Direct attack on the defended model
- Most powerful threat model
- Construct a new loss function, and generate adversarial examples that can fool the classifier and evade the detector
    - Choosing a good loss function is the hardest part
- **Current defenses are basically defenseless against this**

Threat Model: Limited-Knowledge

Adversary is aware the neural network is being secured with a given detection scheme, knows how it was trained, but does not have access to the trained detector D (or the exact training data)

- Most difficult attack for adversary
- Evaluation is only interesting if zero-knowledge attack fails and perfect-knowledge attack succeeded
- Attack relies on transferability property
    - Attacker trains a separate model in the same was as the original model (similar size and quality), and perform a white box attack on the substitute model
    - Evaluate whether these adversarial examples transfer to the original model

Defenses

4 different types of defenses:

- Three of the models use a second neural network to classify images as natural or adversarial.
- Three use PCA to detect statistical properties
- Two perform input-normalization with randomization and blurring
- Two use other statistical tests: maximum mean and kernel density estimation

Defenses: Secondary Classification Based Detection

Building a second classifier to detect adversarial example:

Adversarial Retraining: Two models (*Grosse et al.* and *Gong et. al*) propose very similar defense techniques, where they introduce a new neural layer for creating adversarial examples and train the network to detect these adversarial examples instead of attempting to classify adversarial example correctly.

Defenses: Secondary Classification Based Detection (cont.)

Grosse et al. implementation:

(1) Train a model F_{base} on the training data $X_0 = X$.

(2) Generate adversarial examples on model F_{base} for each $(x_i, y_i) \in X$. Call these examples x'_i .

(3) Let $X_1 = X_0 \cup \{(x'_i, N + 1) : i \in |X|\}$ where $N + 1$ is the new label for adversarial examples.

(4) Train a model $F_{secured}$ on the training data X_1.

Gong et al. implementation:

$$X_1 = \{(x_i, 1) : i \in |X|\} \cup \{(x'_i, 0) : i \in |X|\}.$$

Defenses: Secondary Classification Based Detection (cont.)

Zero-Knowledge Attack Evaluation:

- Used C&W attack on MNIST training set
  - Grosse et al. detects 98.5% attacks as adversarial and classifies the remaining 1.5% correctly.
  - Gong et al. detects 98% correctly
- Both fail against CIFAR training set

Perfect Knowledge Attack Evaluation:

- Defenses were ineffective and added almost no increase in robustness with MNIST
  - C&W attack on Grosse et al. as well as Gong et al. to create adversarial examples succeeded 100% of the time
  - Mean L2 distortion increased by 10% with Grosse et al. and 2.31% with Gong et al.
- The attacks work on similar manner on CIFAR
  - Similar results
  - Distortion increased by less than 5%

Defenses: Secondary Classification Based Detection (cont.)

Limited Knowledge Attack Evaluation:

- Transferable adversarial models created with datasets split into half
    - One of the models known by the defender’s model and one by the attacker
- C&W attack on Grosse et al. created on the defender’s model
    - Attack succeeded 98% of the time on both F_{base} and $F_{secured}$
    - Mean distortion of 5.3
- Gong et al. produces identical results

Defenses: Secondary Classification Based Detection (cont.)

Examining Convolutional Layers: Metzen et al. detect adversarial examples by looking at the inner convolutional layers of the network. They augment the classification with a detection network that takes its input from various intermediate layers of the classification network. Since the implementation looks at the inner layers of a ResNet, the defense only argues robustness against CIFAR.

Defenses: Secondary Classification Based Detection (cont.)

Zero-Knowledge Attack Evaluation:

- Best detector has a 81% true positive rate with the C&W attack
  - 28% false positive rate

Perfect Knowledge Attack Evaluation:

- C&W's attack is able to produce adversarial examples that simultaneously are misclassified by the original network and evades the detector

Defenses: Secondary Classification Based Detection (cont.)

Limited-Knowledge Attack Evaluation:

- Split training data into half; train two detector models; second detector is attacked given only white box access to the first detector
- 84% success rate when using low-confidence adversarial examples
- Attack success rate can be increased to 98% with high confidence examples at the expense of increasing mean distortion.

Defenses: Principal Component Analysis Detection

Principal Component Analysis (PCA) transforms a set of points from n-dimensional space to a k-dimensional space (reducing the number of vectors in different components that captures similarity) by linear transformation

Defenses: Principal Component Analysis Detection (cont.)

Input Image PCA

Hendrycks & Gimbel propose that adversarial example places higher weighting on larger principle components than natural images

Dimensionality Reduction

Bhagoji et al. propose that reducing the W · H · C = N-dimensional input to a much smaller k-dimension dataset for the model to train on will defend against adversarial attacks. They hypothesize that adv. examples rely on the last principle components where this defense only allows manipulation on the first k-inputs

Hidden Layer PCA

Li et al. proposed to apply PCA to the values after each inner convolutional layer and use a cascade classifier to detect adversarial examples. It classifies an image as natural if all ci layers accepts it as input but rejects it if any do not.

Defenses: Principal Component Analysis Detection (cont.)

Hendrycks & Gimbel

Zero-Knowledge Attack Evaluation:

$$S(j) = \frac{1}{N}\sum_{i=1}^{N} |PCA(X_i)_j|.$$

- Reproduce using MNIST and create a scoring function for the classification network
- Train and compute scoring for the network and then find the next closes adversarial example
- Find that adversarial examples and natural images are same in first principal components but differ at the larger ones

Looking Deeper:

- MNIST natural images have some pixels set to zero, specifically border pixels
- Detected difference between the natural and adversarial examples is because the border pixels are nearly always zero for natural MNIST instances and typical adversarial examples have non-zero values on the border pixels
- Evaluations on CIFAR-10 results in no detectable difference between adversarial examples and natural images
- This is specific to only MNIST

Perfect Knowledge Attack will be explained with *Bhagoji et al.*

Defenses: Principal Component Analysis Detection (cont.)

Bhagoji et al.

After reimplementing 26 model with varying k (principal components) parameters, models with fewer than 25 dimensions had lower accuracies while models with more than 25 dimensions reported 97% accuracy of higher

Perfect-Knowledge Attack Evaluation:

- Most difficult model to attack uses only 25 dimensions and it is nearly 3x more robust than the model with 784 dimensions
  - This model is less robust than any unsecured convolution neural network (which achieve 99.5% accuracy) and better robustness to adversarial examples

Defenses: Principal Component Analysis Detection (cont.)

Bhagoji et al.

Second model presented but was not used was a model that combines PCA with a convolutional neural network where the network architecture is held fixed and the only changing input is k

- Model achieves 99% accuracy when using at least 25 dimensions
- No more robust than an unsecure neural network
- Not accurate when measuring only on the first principal components

Defenses: Principal Component Analysis Detection (cont.)

Li et al.

Each classifier C_i uses a linear SVM that acts on the PCA of the ith convolutional layer of the network. This scheme was evaluated on ImageNet

Zero-Knowledge Attack Evaluation:

- Original Model generated adversarial examples with Szegedy et. al's L-BFGS algorithm and found that first linear SVM achieved 80% true positive rate at 0% false positive rate
- Using C&W's attack on first layer of cascade classifier
    - MNIST: Achieves a 62% true positive rate at a 37% false positive rate
    - CIFAR: Achieves a 56% true positive rate at a 40% false positive rate
- The results on C&W's attack on the cascade classifiers was a 92% and 98% false positive rate on MNIST/CIFAR

Defenses: Distributional Detection

Use classic statistical methods to detect adversarial examples by comparing the distribution of natural images to the distribution of adversarial images

Maximum Mean Discrepancy (MMD): Grosse et al. propose a statistical hypothesis that checks if two sets of images are from the same underlying distribution.

To test whether X_1 and X_2 are drawn from the same distribution:

1. Find $a = MMD(X_1,X_2)$
2. Shuffle X_1 and X_2 into Y_1 and Y_2 and find $b = MMD(Y_1, Y_2)$
3. If $a<b$ then reject null hypothesis, they are from different distribution
4. Repeat this many times to find the p-value as the fraction of times null hypothesis was rejected

Defenses: Distributional Detection

Zero-Knowledge Attack Evaluation:

- Fails to reject hypothesis with C&W’s attack with a set of 100 images.
- Fails to detect attacks on CIFAR even with a set as large as 10,000 images
- Since MMD is not effective, statistical tests will not be effective to detect adversarial examples
    - MMD is one of the most powerful multidimensional statistical tests.

Defenses: Distributional Detection

Kernel Density Estimation: FEinman et al. use a Gaussian Mixture Model o model outputs from the final hidden layer of a network and argue that adversarial examples belong to a different distribution than that of natural images.

Kernel Density estimation estimates the likelihood of an instance as:

$$KDE(x) = \frac{1}{|X_t|} \sum_{s \in X_t} \exp(\frac{|F^{n-1}(x) - F^{n-1}(s)|^2}{\sigma^2})$$

The detector is constructed by selecting a threshold (τ) and *x* is reported as adversarial if *KDE(x)* < τ, otherwise *x* is reported as natural.

Defenses: Distributional Detection

Zero-Knowledge Attack Evaluation:

- Defense can detect C&W examples on MNIST but not on CIFAR
- The adversarial example has a higher likelihood score than a natural image 80% of the time with CIFAR

Perfect -Knowledge Attack Evaluation:

- C&W attack using MNIST are not detected by the defense
- This attack is more effective with MNIST than CIFAR because the defense is effectively a K nearest-neighbor classifies
  - Prior work has shown that using an RBF kernel SVM is effective at defending MNIST from adversarial example

Defenses: Distributional Detection

Limited-Knowledge Attack Evaluation:

- With use of 5% of the training instances, the attacks succeeds with MNIST
    - When the defense uses 95% data and the attacks uses 5%, the attacks succeeds with equal distortion

Defenses: Normalization Detection

- Feinman et al. Proposes *Bayesian neural network uncertainty* which measures uncertainty of neural network on the given input
- Add dropout randomization to the network hoping that natural images will have same (correct) label, while adversarial won’t always be predicted with same label

Defenses: Normalization Detection (cont.)

Apply randomized network F_r to input L times, and define uncertainty as:

$$U(x) = \left(\frac{1}{L}\sum_{i=1}^{L}\|F_r(x)\|\right) - \left\|\frac{1}{L}\sum_{i=1}^{L}F_r(x)\right\|$$

- ||y|| represents L2 norm.
- If predictions are same all the time, then uncertainty is 0
- Uncertainty is measured by sum of variances of each component of the output
    - Computed over L outputs of randomized network

Defenses: Normalization Detection (cont.)

$$U(x) = \left(\frac{1}{L} \sum_{i=1}^{L} \|F_r(x)\| \right) - \left\| \frac{1}{L} \sum_{i=1}^{L} F_r(x) \right\|$$

- Pick a threshold T, L, and compare U(x) to T (Natural if U(x) <= T)
- C&W chose T = 0.02, L = 30 (algo not sensitive to L > 20)
- When ran with FGS and JSMA on MNIST:
    - detects 96% of adversarial examples with false positive rate < 1%

Defenses: Normalization Detection (cont.)

Zero-Knowledge Attack Evaluation:

- 75% of adversarial examples detected on MNIST
- **Surprising:** 95% adversarial examples detected on CIFAR

Defenses: Normalization Detection (cont.)

Zero-Knowledge Attack Evaluation:

- 75% of adversarial examples detected on MNIST
- **Surprising:** 95% adversarial examples detected on CIFAR

Defenses: Normalization Detection (cont.)

Perfect-Knowledge Attack Evaluation:

- Gradient descent does not perform well when attacking this defense because calculation for U(.) uses F(.) and not Z(.)
- Construct a loss function by sampling K different deterministic networks each using different randomness, and produce a loss function that averages the loss function on each fixed model

$$\ell'(x') = \sum_{j=1}^{K} \ell_{Z_j}(x')$$

- If they were able to get the same result t with K different random models, it would likely produce label t for any model (given a large enough K)
- Generated adversarial examples with this approach that fooled the dropout defense with 98% success
- Largest distortion required on MNIST (3.68, baseline 2.05), 5x larger distortion factor for CIFAR (1.1)

Defenses: Normalization Detection (cont.)

Limited-Knowledge Attack Evaluation:

- Construct two models F(.) and G(.) on different subsets of training data
- Provide adversary access to parameters of F and use white box approach to generate adversarial examples, and test on G
- On MNIST, adversarial examples transfer to G with 90% success
    - Transfer rate increases to 98% by increasing mean distortion 15% (4.23)

Defenses: Normalization Detection (cont.)

- Li al. Proposes applying 3x3 average filter to blur images before applying classifier
- Author admits defense is “overly simplistic”, but argues it is still effective
- Can remove adversarial examples generated with FGS

Defenses: Normalization Detection (cont.)

Zero-Knowledge Attack Evaluation:

- 80% of low confidence adversarial examples at mean L2 distortion level of 2.05 are no longer classified incorrectly
- Even for high confidence examples, distortion must be increased by a factor of 3x

Defenses: Normalization Detection (cont.)

Perfect-Knowledge Attack Evaluation:

- Taking a mean over every 3x3 region on image is the same as adding convolutional layer to the beginning of the neural network
- Given network F, define F’(x) = F(blur(x)) and apply C&W attack against F’
    - Mean distance to adversarial examples does not increase, so blurring is not an effective defense

Lessons: Properties of adversarial examples

- Randomization can increase required distortion
- MNIST properties may not hold on CIFAR
- Detection neural networks/ using more layers can be bypassed
- Operating on raw pixel values is ineffective for detection

Lessons: Recommendations for Defenses

- Evaluate on strong attacks (not just fgs/jsma)
- Demonstrate white-box attacks fail
- Report false positive and true positive rates
- Evaluate on more than MNIST
- Release source code

Conclusion

It is important to think from the perspective from an attacker that knows how the defense works

After Studying ten defenses:

- Existing defenses lack thorough security evaluations
- Adversarial examples are much more difficult to detect than previously recognized