The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Detecting Adversarial Examples through Image Transformation*

Shixin Tian, Guolei Yang, Ying Cai
Department of Computer Science, Iowa State University
{stian,yanggl,yingcai}@iastate.edu

Abstract

Deep Neural Networks (DNNs) have demonstrated remarkable performance in a diverse range of applications. Along with the prevalence of deep learning, it has been revealed that DNNs are vulnerable to attacks. By deliberately crafting *adversarial examples*, an adversary can manipulate a DNN to generate incorrect outputs, which may lead catastrophic consequences in applications such as disease diagnosis and self-driving cars. In this paper, we propose an effective method to detect adversarial examples in image classification. Our key insight is that adversarial examples are usually sensitive to certain image transformation operations such as rotation and shifting. In contrast, a normal image is generally immune to such operations. We implement this idea of image transformation and evaluate its performance in oblivious attacks. Our experiments with two datasets show that our technique can detect nearly 99% of adversarial examples generated by the state-of-the-art algorithm. In addition to oblivious attacks, we consider the case of white-box attacks. We propose to introduce randomness in the process of image transformation, which can achieve a detection ratio of around 70%.

Introduction

The past decade has witnessed the unprecedented thrift of machine learning techniques. Deep Neural Networks (DNNs), at the front of this machine learning trend, has been used to assist decision making in a diverse range of applications from disease diagnosis (Esteva et al. 2017), navigating self-driving cars (Bojarski et al. 2016), to natural language processing (Petrov 2016) and playing the game of Go (Silver et al. 2016). In all these tasks where traditional machine learning techniques found challenging to handle, DNNs have demonstrated superior performance.

Along with the prevalence of deep learning techniques, however, it has been revealed that DNNs are vulnerable to attacks using *adversarial examples* (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015; Kurakin, Goodfellow, and Bengio 2016; Baluja and Fischer 2017; Carlini and Wagner 2017b) in the case of image classification. An adversarial example is generated by adding certain perturbations into a normal image. The perturbed image is specifically generated so that it is visually similar to the original image, but will be misclassified when fed into the DNNs. By deliberately crafting adversarial examples, a malicious attacker can manipulate a well-trained DNN to generate incorrect outputs. This situation may lead to catastrophic consequences and cost human lives in applications such as disease diagnosis and self-driving cars. For example, an attacker could modify a stop sign so that an auto-driving system mistakes it as a speed limit sign and fails to stop. It could result in a severe traffic accident and pose direct threat to the passengers.

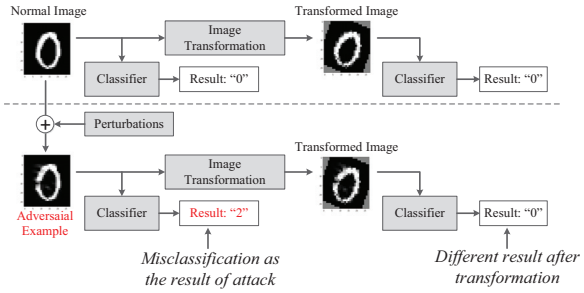

Figure 1: An illustration of image transformation-based adversarial example detection.

Adversarial example attacks raise crucial security concerns in applications where DNNs are used to support decision making. In response to the threat, many defensive techniques have been proposed by the machine learning and security community. A significant number of these techniques attempt to thwart the attack through adversarial examples detection, i.e., distinguishing adversarial examples from normal ones. Unfortunately, the recent research (Carlini and Wagner 2017a) has demonstrated that the adversarial examples generated by the *Carlini-Wagner* (CW) attack (Carlini and Wagner 2017a) can circumvent all existing detection techniques, especially in white-box threat model. This calls for new research on effective methods to detect adversary examples.

In this paper, we propose a novel adversarial example detection method that can effectively thwart the state-of-the-art CW attack. Our key insight is, adversarial examples are

*This research was supported in part by a grant from the Defense Advanced Research Project Agency and a research contract from Kingland Systems Corporation.
Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

usually sensitive to image transformation operations such as rotating and shifting. In other words, the classification results of an image and its transformed version are very likely to be different. In contrast, information contained in a normal image are generally immune to such operations. Figure 1 illustrates this observation with an example using an image from the MNIST dataset. The classification results of the adversarial example are different before and after image transformation, while this phenomenon is not observed on a normal image.

Inspired by this observation, we develop an image transformation-based adversarial example detection method. Our method is to firstly apply certain transformation operations on an image to generate several transformed images. We then use the classification results of these transformed images as features to predict if the original image has been perturbed by an adversary, i.e., whether or not this image is an adversarial example. Our method exploits a set of transformation operations that are effective in distinguishing normal and adversarial examples. By employing image transformations, we could effectively detect adversarial examples of oblivious attacks. To defend against more sophisticated white-box attacks, we propose to introduce randomness in the transformation process. Intensive experiments on several image dataset shows that the proposed method is effective: It can detect 99% of adversarial examples generated in the oblivious attack. For the white-box attacks, it achieves a detection ratio of more than 70%. We summarize our contribution as follows:

- We make the important observation that adversarial example are usually sensitive to image transformation operations. In contrast, normal images are generally immune to such operations. The observation provides a new research direction for adversarial example detection.
- Based on our observation, we propose a novel image transformation-based adversarial example detection method for DNN. To the best of our knowledge, it is the first method that is effective in thwarting the state-of-the-art CW attack in the white-box setting.
- We implement the proposed method and evaluate it on two image datasets. Our method shows superior performance in defending against the CW attacks in both oblivious and white-box threat model.

The rest of the paper is organized as follows. We discuss more related works in Section 2. We present some backgrounds and two threat models in Section 3. In Section 4, we present our technique in detail and show the experimental results of its effectiveness in detecting adversarial examples. Finally, we conclude this paper in Section 5.

Related Work

We briefly summarize attack algorithms to deep neural networks and corresponding defensive techniques.

Attack to Deep Neural Networks

Deep learning is widely adopted to support decision making in many crucial application fields. Hence, understanding the robustness of DNN in adversarial environment is a vital task. Towards this goal, Szegedy et.al (Szegedy et al. 2013) introduced adversarial example attack that targets DNNs in image classification. They found that an adversary can cause the network to misclassify an image by applying a certain hardly perceptible perturbation which maximizes the network's prediction error.

Their work has since inspired several research attempts to further examine the security weakness of DNNs. For example, *the fast gradient sign method* (FGS (Goodfellow, Shlens, and Szegedy 2015)) is a one-step shot attack. It defines a loss function $Loss(x, l)$ which represents the cost of classifying x as label l. Then it tries to maximize $Loss(x, l_x)$ where l_x is the ground-truth label of x. FGS solves this optimization problem by performing one step gradient update from x with volume ϵ. Based on FGS, *the iterative fast gradient sign method* (IFGS (Kurakin, Goodfellow, and Bengio 2016) performs attacks by conducting a series of smaller-step FGS updates iteratively. Other approaches such as DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) and JSMA (Papernot et al. 2016) formalize the attack as different optimization problems.

To our knowledge, the most recent work in this field is due to Carlini et.al (Carlini and Wagner 2017b). The proposed CW attacks is shown to achieve the best attacking results comparing with previous attack algorithms. It can generate adversarial examples that are able to circumvent existing detection methods (Carlini and Wagner 2017a). Due to its significant threat, in this paper, we mainly focus on defending the CW attack. Nevertheless, our method is also effective against the other aforementioned attack algorithms.

Defensive Techniques

In response to the threat, several defensive techniques have been proposed. These techniques generally fall in two categories. The first category of techniques attempts to classify the adversarial examples correctly, which is to construct classifiers that are robust against perturbation added by adversaries. Techniques in this category include (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015; Tramèr et al. 2017; Cao and Gong 2017; Xu, Evans, and Qi 2017). However, most of these techniques are not effective at classifying adversarial examples correctly (Carlini and Wagner 2017a).

The second category is adversarial example detection, where the goal is to build a *detector* that is able to distinguish adversarial examples from the normal ones. Our technique belongs to this category. Inherently, a detector is also a classifier, which is trained on a set of normal and adversarial examples. Detectors learn to capture the difference caused by the added perturbation. Adversarial example detection techniques include (Grosse et al. 2017; Gong, Wang, and Ku 2017; Metzen et al. 2017; Feinman et al. 2017; Meng and Chen 2017). As mentioned above, these existing detection methods can all be bypassed by crafting specific adversarial examples (Carlini and Wagner 2017b). As a result, effective defence against state-of-the-art adversarial example attack remains an open issue, which we aim to address in this paper.

Background

Notations

The notations used in the paper follow the rules in (Carlini and Wagner 2017a). Let $F(\cdot)$ denote a classification model (i.e., a deep neural network). The input to $F(\cdot)$ is the vector representation of an instance (in our case, an image), and the output is the probability distribution over all possible classification labels. More formally, $F(x)_i$ denotes the probability that instance x is classified as label i.

Let $Z(\cdot)$ denote the logits, i.e., the output of the final layer (before the softmax layer) of the DNN. Assuming a softmax activation layer is applied on the logits to compute the label probabilities, the model outputs can be written as:

$$F(x) = softmax(Z(x)) \tag{1}$$

The label with the highest estimated probability is then used as the predicted label of x, denoted by:

$$C(x) = arg\max_i(F(x)_i) \tag{2}$$

Given a model $F(\cdot)$ and a valid input image x, an *adversarial example* generated on x is denoted as x', which is generated by adding certain perturbations into x. As the result of such perturbations, the predicted label of x is changed, i.e., $C(x') \neq C(x)$.

The goal of adversarial example detection is to construct a *detector*, denoted by D, which is also a classifier. Given an input image x, D aims to correctly label the input as an adversarial example or a normal image.

Attack Algorithm

In this paper, we focus on the state-of-the-art adversarial example attack, namely CW attack (Carlini and Wagner 2017a). The CW attacks is shown to achieve the best attacking performance comparing with previous attack algorithms, i.e., it can generate adversarial examples that are able to circumvent most of existing detection methods.

For a given image, the goal of the CW attack is to find a small perturbation that can mislead the model to give the incorrect output that does not match the actual label of the image. The CW attack assumes the adversary wants the image to be classified as a specific (incorrect) label, which is called the *target label*. The attack can be formulated as the following optimization problem:

$$\begin{aligned} &\min ||\delta||_p + c \cdot f(x+\delta) \\ &such\ that\ : x+\delta \in [0,1]^n \end{aligned} \tag{3}$$

where $||\delta||_p$ is a distance metric. Note in this paper, we use L2 norm in the evaluation, i.e., $p = 2$. The function $f(\cdot)$ indicates whether the attack succeed or not, which is defined in the following way:

$$f(x') = max(Z(x')_{l_x} - max\{Z(x')_i : i \neq l_x\}, -\kappa) \tag{4}$$

where κ is a hyperparameter that controls the level of confidence in generating an adversarial example. A larger κ encourages the attacker to generate an adversarial example x' which will be classified as label i by the DNN with higher confidence. Finally, c is a hyperparameter which is used to balance $||\delta||_p$ and $f(\cdot)$. This hyperparameter can be tuned using binary search by the CW attack automatically.

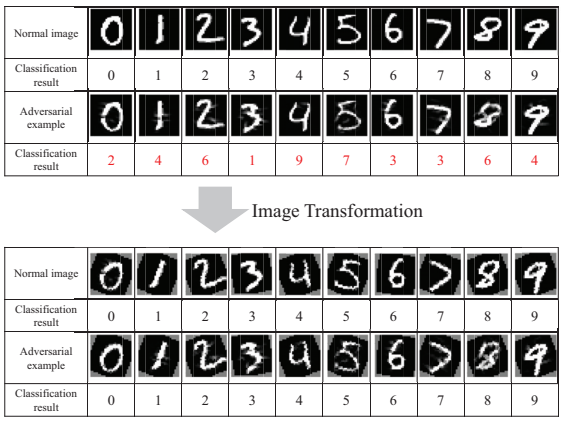

Figure 2: Impact of image transformation on normal and adversarial examples

Threat Models

In this paper, we consider two different threat models.

1. **The Oblivious Model**: the adversary has knowledge of the original classifier F but is not aware of the detector D. Hence, the adversary's goal is only to fool the unsecured model F.
2. **The White-Box Model**: the adversary has knowledge of the model F and is aware of the existence of the detector D. It also has knowledge of the structure and exact parameters of D. In other words, the adversary needs to fool both the classifier F and D simultaneously.

Proposed Methods and Evaluation

There has been a debate on why the adversarial examples exist (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2015; Tramèr et al. 2017; Meng and Chen 2017). But most literate agree the security problem roots from the imperfection of machine learning models when compared with humans in some tasks. When an image appears to be hard to recognize, a human usually would apply certain image transformation operations to help improve its readability. For example, the human may rotate the image, look from a different angle, move it closer/further, etc, in order to discover new features from different prospects.

We find this trick also works for machine learning models when it comes to adversarial examples. In image classification, adversarial examples trick a model by adding small perturbations to specific positions in an image. These positions are selected so that adding perturbations there will have the maximal impact on the model's prediction error, and therefore push the image over the decision boundary. However, image transformation such as rotation and shifting, could change the shape of the decision boundary, and therefore render such perturbations invalid. Figure 2 shows a set of handwriting images and corresponding adversarial examples. Note that transforming an adversarial examples is likely to cause their classification results to change, but have less impact on normal images.

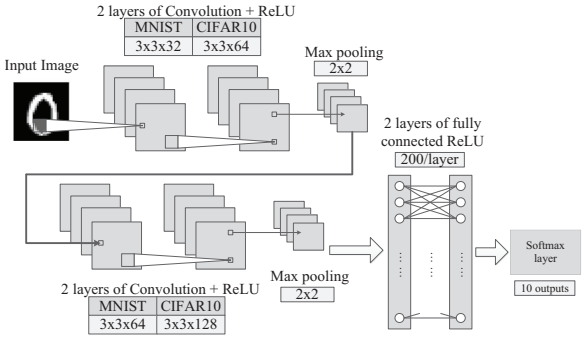

Figure 3: Classifier Architectures

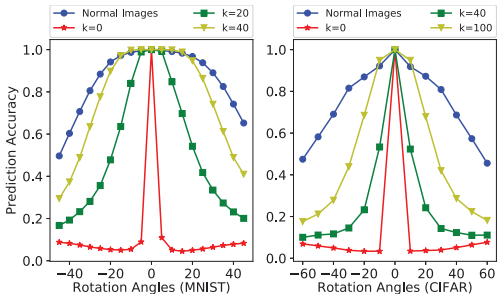

Figure 4: Adversarial examples and normal images demonstrate a significant different level of sensitivity to the rotation operation in terms of classification accuracy.

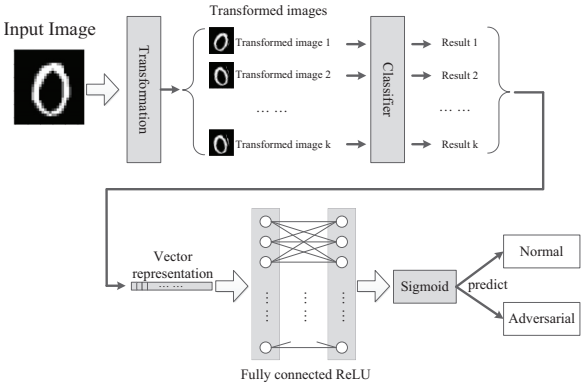

Figure 5: Image transformation-based adversarial example detection

A recent paper (Lu et al. 2017) argues that physical adversarial examples can be correctly classified when viewing the object from different angles and distances. Thus, the adversarial example problem may not be worried too much. However, this paper is not targeting the state-of-the-art attack. Additionally, an opposing opinion is stated in (Athalye and Sutskever 2017) where the authors generated adversarial examples which can yield same prediction labels even when zoomed or rotated. In this paper, we are trying to detect adversarial examples which is easier than to recover their original labels as (Lu et al. 2017). Our detection method would work if there are different patterns between normal images and adversarial examples..

We design the following experiments to validate our observation. First, we implement two Convolutional Neural Networks (CNNs) as the image classifier F, one for each image dataset. The architectures of these CNNs are described in Figure 3. The classifiers are trained on two popular image classification datasets: MNIST (LeCun 1998) and CIFAR10 (Krizhevsky and Hinton 2009). The MNIST dataset has $70,000$ handwritten digits from which $60,000$ are used as the training set and $10,000$ as the testing set. The CIFAR10 dataset consists of $60,000$ colour images in 10 classes. $50,000$ of them are used as the training set and the rest as the testing set. These classifiers are trained at the learning rate of 0.01 with a batch size of 128 and 50 epochs.

For each image in the testing set, we generate three adversarial examples with different confidence levels using the CW attack algorithm. Then, we rotate the original image and its corresponding adversarial examples for a certain angle. We feed the rotated image sets into the classifier and then compute the average prediction accuracy on each set. Note that for an adversarial example, we say a prediction result is accurate if it is classified as the adversary desired. That is too say, the accuracy for adversarial examples is the attack success rate.

We plot the classification accuracy for the four sets of images with respect to different rotation angles in Figure 4. Note that the classifier achieved 100% accuracy for all the images when no rotation is applied because we only care about the images (or adversarial examples) that are classified correctly (attacked successfully). But as the rotation angel increases, it can be seen that it will cause the prediction accuracy (or attack success rate for adversarial examples) to drop for all the image sets. However, the adversarial examples are significantly more sensitive to rotation comparing with the normal image samples. This effect is particularly obvious on adversarial examples with a lower confidence level ($\kappa = 0$).

Our proposed detection method takes advantage of the fact that a minor transformation of an image may result in a significant change of the classification results. Our basic idea is to apply several transformation operations on an image. Then, the transformed images are fed into the model and the corresponding classification results are collected. Finally, using these results as observation, we can train a detector to identify adversarial examples from normal images using supervised learning techniques. The experiments are implemented with Keras using TensorFlow as backend.

In the following subsections, we present the proposed method with respect to the oblivious threat model and the white-box threat model, and evaluate their effectiveness via experiments on the two image datasets. Figure 5 illustrates the proposed method on a high level.

Defense Against Oblivious Attacks

In oblivious attacks, the adversary does not know the existence of such a detector. That means the adversary only generates adversarial examples that aim to maximize the prediction error of the classifier.

In order to train the detector, our first step is to generate a set of training instances. To this end, we generate a set of adversarial examples using the CW attack algorithm for each image in training set of MNIST and CIFAR10 as described above. The target label of each adversarial example is randomly selected (different from the ground truth label of the original image). The adversarial examples and original images are fed into the classifier and we select a subset of images as follows: 1) The image is a normal image and the classifier can classify it correctly. 2) The image is an adversarial example and it can attack the classifier successfully, i.e., its classification result is the target label selected by the adversary. The purpose of this selection process is to filter out two types of images which makes it hard to evaluate the actual performance of the proposed detection method. The first type of images are those hard to classify correctly even without perturbation and the second type of images are those adversarial examples which fail to attack even without any defensive method.

Among the selected images, the original images are labelled *valid* (0) and the adversarial examples are labelled *adversarial* (1). Then, we apply a set of transformation operations on each image. Specifically, an image is firstly shifted to one direction (up, down, left, or right) by 0, 1 or 2 pixels; then, it is rotated by one of the following angles ($[-30, 15, 0, 15, 30]$ for MNIST and $[-50, 25, 0, 25, 50]$ for CIFAR10). As there are 9 different ways to shift and 5 different angles to rotate an image, there are totally $9 \times 5 = 45$ *transformed versions* of each image.

For each image, we feed its 45 transformed versions into the classifier and record the outcomes. More specifically, for each image, we record the logits (output of the DNN before the output layer) of its 45 transformed versions. The length of the logit is denoted by N (in our experiments $N = 10$ as there are 10 classes). By concatenating these 45 logits, we get a vector of length $45 \times N$, which is used as the representation of the image. These vectors are then used as the training instances for the detector.

Table 1: Detector architectures with respect to different dataset

Layer Type	MNIST	CIFAR10
Fully Connected + ReLU	128	128
Fully Connected + ReLU	N/A	32
Fully Connected + Sigmoid	1	1

We use multi-layer perceptrons as the detectors whose architecture is shown in Table 1. The detectors on MNIST have two layers while the detectors on CIFAR10 have 3 layers. On each dataset, we train four detectors on different training sets: D_0 is trained on a dataset in which the adversarial examples have a confidence level of 0 ($\kappa = 0$). Similarly, D_{20} and D_{40} corresponds to adversarial examples with confidence levels of 20 ($\kappa = 20$) and 40 ($\kappa = 40$), respectively. And finally, D_A is trained using adversarial examples from all these confidence levels. The set of normal images used in the three training sets are the same.

For each image in the original testing set (of MNIST and CIFAR10), we generate adversarial examples with different confidence levels. The generated adversarial examples and the original images together are used as the testing set for the detector. We remove all the images which are classified incorrectly by the classifiers due to the same reasons discussed above. The performance of the trained detectors are showed in Figure 6 in terms of detection accuracy. Here, we do not use precision, recall, or F1 score as performance metrics. This is due to the fact that in our experiments, a "positive" case is defined differently for classifying normal images and adversarial examples with different confidence levels and these metrics may vary as the ratio of normal image and adversary examples varies. As such, some common classification performance metrics may be confusing with respect to our results.

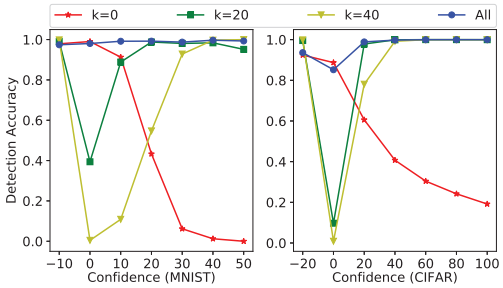

Figure 6: Detection results for oblivious attack. Detectors are evaluated on adversarial examples with different confidence levels. Adversarial examples with negative confidence levels are valid inputs (normal images). Models trained on low-confidence adversarial examples can detect high-confidence adversarial examples.

Except for D_0, we observe that a detector that is trained using only low-confidence adversarial examples is also able to detect high-confidence adversarial examples. This phenomenon is also reported in (Carlini and Wagner 2017a). This result indicates all adversarial examples share certain inherent similarity when put under image transformations. Overall, the detector trained using all adversarial examples achieves the best detection rate. The experiment demonstrates that our idea of image transformation is effective in thwarting the CW attack.

Defense Against White-Box Attacks

In white-box attacks, the adversary has complete knowledge of both the classifier and the detector, making it more challenging to defend. We simulate CW attacks in a white-box threat model with the method presented in (Carlini and Wagner 2017a). The basic idea is to construct a new model which combines the image classifier and the detector. Suppose in the original classification model there are N different labels. Then the new model would have $N+1$ labels from which the last label represents if the input is an adversarial example.

Let G denote the new model which combines the classifier F and the detector D. Denote G's logits as Z_G, F's logits as Z_F and D's output as Z_D. We construct Z_G in the following way:

Table 2: Distortion of adversarial examples

Confidence	Oblivious	White-Box	Increase By(%)
0	1.94	2.55	31.5
10	2.93	4.97	82.3
20	3.29	7.64	132.7
30	4.66	8.75	87.7
40	5.44	10.54	93.8
50	7.72	11.79	52.6

$$Z_G(x)_i = \begin{cases} Z_F(x)_i & \text{if } i \leq N \\ (Z_D(x) \times 2 \times \max Z_F(x) & \text{if } i = N+1 \end{cases}$$

It is easy to see that when an input is detected as an adversarial example by D, it would also be classified as adversarial (the last label) by G. This is because Z_D would be larger than 0.5 when the input is detected as adversarial. Then $Z_G(x)_{N+1}$ would be larger than $Z_G(X)_i$ for $1 \leq i \leq N$. If an input is detected as being valid, G would classify it as the same label as F does. In this way, the new model G combines F and D.

Now we can apply the CW attack on the new model G. For all the adversarial examples generated in the white-box threat model, we check if they can be detected by D. The evaluation shows that D can barely detect any adversarial example, as shown in Table 3. In other words, the CW attack can bypass the detector. Similar results are reported in (Carlini and Wagner 2017a) where the CW attack bypassed *ALL* of the ten detection methods it explored.

Although the CW attack can bypass the detector, the existence of detector will increase the cost of adversarial examples in terms of *distortion*. Distortion is defined as the L2 distance between an original image and its adversarial example. A smaller distortion means that the adversarial examples is closer to the original image and less likely to be distinguished by human eyes. The CW attack can generate adversarial examples with much less distortion than previous attack algorithms. We compare the distortions of the adversarial examples generated in different settings (oblivious attack or white-box attack).

In Table 2, we can see that by introducing the detector, The CW attack must add more distortion to the original image to fool both the image classifier and the detector, especially in generating high confidence level adversarial examples. This is because a higher confidence level means pushing the adversarial examples further beyond the decision boundary.

However, increasing the cost of an attack is not the same as successfully defending against the attack. Taking into consideration the mechanism of the CW attacks, we propose to add randomness into our defensive method. The added random targets the optimization method of The CW attacks. The goal is to make it much harder, if possible, to find the optimal solution to launch an attack.

Specifically, instead of perform image transformation with certain fixed steps, we conduct some randomly selected transformation operations. After an input image is shifted, it is rotated by a random angle selected from certain range. For example, when an image should be rotated by an angle of θ, we rotate it by a random angle θ' where θ' is drawn from a uniform distribution ($\theta' \in Unif(\theta + \epsilon, \theta - \epsilon)$). Besides this simple random rotation, all the other steps in training and testing the detector remains the same. We first examine if the added randomness effect CW attack in the oblivious threat model. The results are shown in Figure 7. Note in our evaluation, δ is set to 5. We can see that the performance of the detectors has a slight drop compared to the determined case showed in Figure 6.

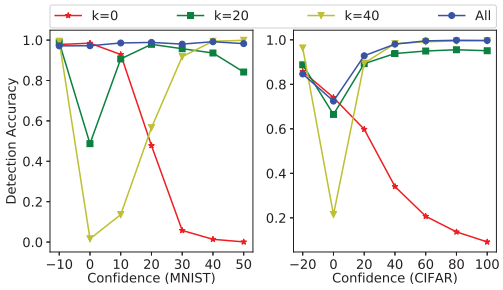

Figure 7: Detection results for the oblivious attack with random rotation angles.

Now we perform the white-box CW attack against the randomized model G. The performance is shown in Figure 8. Even though all the detectors suffer from detection accuracy drops, they are acceptable compared to the existing defensive models. None of then can prevent the CW attack (Carlini and Wagner 2017a) in the white-box setting. Note that these results are not as smooth as others because of lack of samples. In the white-box evaluation, for each detector and each confidence levels, we generate 400 adversarial examples on MNIST dataset and 100 on CIFAR10. Using the CW attack to generate adversarial examples is a time-consuming job. When performing our experiments on a laptop with GPU (NVIDIA GeForce GTX 960M), generating one adversarial example for oblivious threat model takes around 4 seconds while generating one adversarial example for white-box threat model takes around 75 seconds in average. This is because the model G which the CW attack is targeting in the white-box setting is much more complex than the original classifier F which the CW attack is targeting in the oblivious attack. Besides the detector itself, the classifier F is reused for 45 times in G.

An empirical comparison is given in Table 3. In this evaluation, we compare our proposed method with Gong (Gong, Wang, and Ku 2017) and Grosse (Grosse et al. 2017). All the models are trained and tested on adversarial examples with confidence level of 0. The CW attack can always generate successful adversarial examples with high probability, i.e., it achieved high attack success rate. In oblivious threat model, these adversarial examples can be detected by any of the methods. However, we can see that, in white-box threat model, both Gong and Grosse's methods failed to detect the adversarial examples since their detection rate dropped to zero. The distortions increased by less than 10% compared

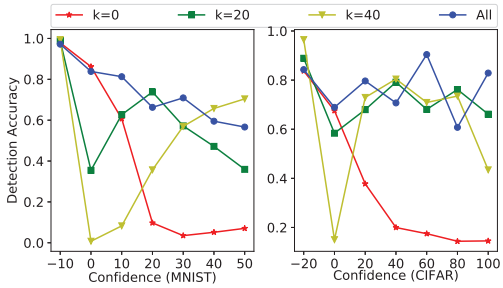

Figure 8: Detection results for the white-box attack with random rotation angles. Adversarial examples in white-box mode are generated and then detection accuracy is calculated. Note adversarial examples with negative confidence levels are valid inputs.

Table 3: Comparison of Different Methods on MNIST

Threat Model	Metric	Gong	Grosse	Proposed
Oblivious	AttackSuc	1	1	0.97
	DetectSuc	0.99	0.99	0.96
	Distortion	1.94	1.94	1.94
White-box	AttackSuc	1	1	0.97
	DetectSuc	0	0	0.85
	Distortion	2.11	2.06	2.55

with those of oblivious model. But our method achieved 85% detect success rate while increased distortion by more than 30%.

Now let us look into the reasons why this trick works in the white-box attack settings. The essential reason is that the detector evaluates an input with different random angles. What happens if we let the attacker use exactly the same rotation angles as the detector? We evaluate detection accuracy in this case and the results showed that all the adversarial examples bypassed our detector (the accuracy dropped to 0, same as Gong and Grosse's techniques). This means that the CW attack believes it has already found adversarial example which can circumvent the detector. If the detector continues using the same angles (as the CW attack does), it would be fooled by the CW attack. So from a general perspective, the CW attack model can still bypass our detector, only if the attack model knows what random angles the detector uses in its future evaluation. In this case, the randomization is eliminated and the detectors are bypassed.

But as a matter of fact, the rotation angles are randomly generated whenever the detector evaluates an input. So in each time when the detector works, new and different angles are used and the adversary can never know. Those adversarial examples that work in the previous configuration (rotation angles), will be highly likely to fail in the new configuration.

Defense Against Other Attacks We show that our detector that targets CW attack is also effective against other adversarial example attacks. We train a detector with adversarial examples generated using the CW attack, and use it to detect adversarial examples generated by two other attack

Table 4: Evaluation on Other Attacks

Attack Method	EPS	AttackSuc	DetectSuc
FGS	0.1	0.175	0.994
FGS	0.3	0.481	0.989
FGS	0.5	0.586	0.967
IFGS	0.1	0.927	0.999
IFGS	0.3	0.999	0.999
IFGS	0.5	1	0.999

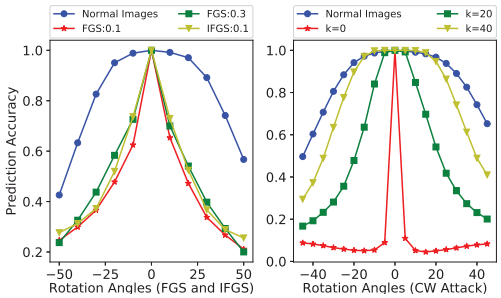

Figure 9: The FGS and IFGS attacks demonstrate similar patterns as the CW attack.

algorithms, namely FGS (Goodfellow, Shlens, and Szegedy 2015) and IFGS (Kurakin, Goodfellow, and Bengio 2016)).

The detector (D_A) is trained with adversarial examples generated by the CW attack with different confidence levels. We can see from Table 4 that the detector is effective against these two attacks eventhough the detector is not trained on adversarial examples generated by them. As such, we can conclude that adversarial examples generated with these different algorithms are all sensitive to image transformations, which can be used by our detector. This is shown in Figure 9, which illustrates the sensitivity of FGS and IFGS adversarial examples to image transformation.

Conclusion

Adversarial example attack can mislead deep neural networks to generate incorrect outputs as the attacker desires. This raises significant security concerns in applications where deep learning is used to support decision making. Several defensive techniques have been proposed but they can all be circumvented by carefully crafting adversarial examples. As such, effective defense against state-of-the-art adversarial example attack remains an open issue. In this paper, we proposed image transformation-based adversarial example detection method. Our insight is that adversarial examples are more sensitive to certain image transformation operation comparing with normal images. Taking advantage of this difference, we trained detectors to identify adversarial examples from normal images. Experimental results showed that our method is effective against the state-of-the-art CW attack in both oblivious attacks and white-box attacks. To our knowledge, this defensive capability is not available from other existing defensive techniques.

Acknowledgement

We would like to thank Professor Neil Zhenqiang Gong from the Department of Electrical and Computer Engineering at the Iowa State University for invaluable discussions and assistances throughout this research.

References

Athalye, A., and Sutskever, I. 2017. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*.

Baluja, S., and Fischer, I. 2017. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*.

Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Cao, X., and Gong, N. Z. 2017. Mitigating evasion attacks to deep neural networks via region-based classification. *Annual Computer Security Applications Conference (ACSAC)*.

Carlini, N., and Wagner, D. 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. *arXiv preprint arXiv:1705.07263*.

Carlini, N., and Wagner, D. 2017b. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, 39–57. IEEE.

Esteva, A.; Kuprel, B.; Novoa, R. A.; Ko, J.; Swetter, S. M.; Blau, H. M.; and Thrun, S. 2017. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542(7639):115–118.

Feinman, R.; Curtin, R. R.; Shintre, S.; and Gardner, A. B. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

Gong, Z.; Wang, W.; and Ku, W.-S. 2017. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. *International Conference on Learning Representations(ICLR)*.

Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; and McDaniel, P. 2017. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

LeCun, Y. 1998. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Lu, J.; Sibai, H.; Fabry, E.; and Forsyth, D. 2017. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*.

Meng, D., and Chen, H. 2017. Magnet: a two-pronged defense against adversarial examples. *ACM Conference on Computer and Communications Security*.

Metzen, J. H.; Genewein, T.; Fischer, V.; and Bischoff, B. 2017. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2574–2582.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, 372–387. IEEE.

Petrov, S. 2016. Announcing syntaxnet: The worlds most accurate parser goes open source. *Google Research Blog*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Xu, W.; Evans, D.; and Qi, Y. 2017. Feature squeezing mitigates and detects carlini/wagner adversarial examples. *arXiv preprint arXiv:1705.10686*.