Adaptive and Diverse Techniques for Generating Adversarial Examples

by

Warren He

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Dawn Song, Chair
Professor David Wagner
Professor Steven Weber
Professor Raluca Popa

Fall 2018

Adaptive and Diverse Techniques for Generating Adversarial Examples

Abstract

Adaptive and Diverse Techniques for Generating Adversarial Examples

by

Warren He

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Dawn Song, Chair

Deep neural networks (DNNs) have rapidly advanced the state of the art in many important, difficult problems. However, recent research has shown that they are vulnerable to adversarial examples. Small worst-case perturbations to a DNN model's input can cause it to be processed incorrectly. Subsequent work has proposed a variety of ways to defend DNN models from adversarial examples, but many defenses are not adequately evaluated on general adversaries.

In this dissertation, we present techniques for generating adversarial examples in order to evaluate defenses under a threat model with an adaptive adversary, with a focus on the task of image classification. We demonstrate our techniques on four proposed defenses and identify new limitations in them.

Next, in order to assess the generality of a promising class of defenses based on adversarial training, we exercise defenses on a diverse set of points near benign examples, other than adversarial examples generated by well known attack methods. First, we analyze a neighborhood of examples in a large sample of directions. Second, we experiment with three new attack methods that differ from previous additive gradient based methods in important ways. We find that these defenses are less robust to these new attacks.

Overall, our results show that current defenses perform better on existing well known attacks, which suggests that we have yet to see a defense that can stand up to a general adversary. We hope that this work sheds light for future work on more general defenses.

Contents

Acknowledgments

For all the support I received in writing this dissertation, I would like to thank my advisor, Dawn Song; the members of my dissertation committee, David Wagner, Steven Weber, and Raluca Popa; and my colleagues Chaowei Xiao, Arjun Bhagoji, James Wei, Xinyun Chen, Nicholas Carlini, Jun-Yan Zhu, and Bo Li.

Chapter 1

Introduction

Deep neural networks (DNNs) are vulnerable to *adversarial examples*, which are slightly perturbed inputs that cause prediction errors. Recent research on adversarial examples has proposed techniques to defend DNN models from the effects of adversarial examples. These defense proposals come in several categories, including input pre-processing, changes to the training method, changes to the network architecture, adversarial retraining, the addition of non-deterministic steps, and the addition of a secondary classifier (for detection approaches).

In this dissertation, we provide techniques for evaluating defenses which complicate the task of formulating a loss function for use with existing gradient based attacks. To do so, we perform an in-depth evaluation of three proposed defenses that were demonstrated to be effective against a variety of attacks and describe new weaknesses that were not previously known.

- We show that feature squeezing [Xu et al., 2017a], an ensemble defense that combines two input pre-processing techniques, can be evaded by an optimization based attack using a surrogate loss function that imitates the pre-processing but in a differentiable way.
- We show that ensemble of specialists [Abbasi and Gagné, 2017], an ensemble defense that combines classifiers trained to be more robust at simpler tasks, can be evaded by an optimization attack using a loss function that considers each constituent classifier.
- We show that an ensemble of detectors from different defenses [Gong et al., 2017, Metzen et al., 2017, Feinman et al., 2017] can be evaded by optimizing a loss function that favors misclassification and
- We show that region classification [Cao and Gong, 2017], a non-deterministic defense that samples classification results from nearby inputs, can be evaded by finding adversarial examples that are consistently misclassified when perturbed in a few random directions.

We analyze the common themes of these weaknesses and propose stronger criteria for guiding future research in adversarial examples defenses.

In order to better assess the generality of current defenses, we then preemptively study a collection of new attack techniques that differ from existing attacks in important ways.

- We study Bhagoji et al.'s black-box attacks [2018], where the attacker can query the model.
- We study AdvGAN [Xiao et al., 2018a], which trains a neural network to create perturbations rather than using the gradients of the model.
- We study stAdv [Xiao et al., 2018b], which perturbs an image by spatially shifting its pixels rather than adding to the pixels' values.

While previous attacks have been sufficient in evading some of the defenses we study, these additional results demonstrate the broadness of possible future attacks.

Although deep learning has rapidly advanced state of the art performance in important and difficult problems, we have a limited understanding of the resulting neural networks. In order to improve our confidence in deploying these models in the real world, we should be aware of not only their successes but also their limitations.

1.1 Background

In this section, we introduce the topics of deep learning and adversarial examples, and we provide an overview of previous defenses against adversarial examples and related work.

Deep learning A class of functions $F_\theta(x)$ called *neural networks* applies a sequence of linear combination operations, often a matrix multiply or a convolution, and nonlinear operations, such as a rectified linear unit ($\text{ReLU}(x) = \max(x, 0)$). Neural networks can approximate different functions by using different weights θ in the linear combination steps. Deep neural networks (DNNs), which have many layers of linear combinations and nonlinearities, are expressive. Convolutional neural networks (CNNs) are a class of neural networks in which some of the linear combination operations are convolutions, which for a given intermediate value limits the number of dependencies on intermediate values from the previous layer (as opposed to a "fully connected" matrix multiply). Additionally, neural networks are differentiable, which makes them suitable for use in machine learning. In deep learning, a system trains a neural network model for a given task by adjusting the model's weights based on the derivative of an objective function of the neural network's inputs or intermediate values.

Advances in deep learning have greatly improved the state of the art performance in a variety of difficult problems, such as image recognition [Krizhevsky et al., 2012, He et al., 2016], text analysis [Collobert and Weston, 2008], and speech recognition [Hinton et al., 2012a]. New systems that use deep neural networks [Watson Visual Recognition, Google Vision API, Clarifai] reduce the amount of human attention needed in important processes such as online content moderation.

In this dissertation, we focus on classification models, where the task is to assign an input to a class $c \in C$. To adapt a neural network, with real-valued output, to perform classification, we use an architecture that has an output dimensionality of $|C|$, where each dimension corresponds to a possible class. The output for each dimension represents a confidence level that the input belongs

to the corresponding class. In our experiments, we use image classification models, where the input, $w \times h$ pixels and c channels, is a vector $x \in \mathbb{R}^{w \times h \times c}$.

Adversarial examples While deep neural networks appear to be robust to random noise, recent work has pointed out that they are strongly affected by small worst-case perturbations. These perturbations applied to an input that is normally correctly classified can cause the model to classify it incorrectly. These are called *adversarial examples* [Szegedy et al., 2014a, Goodfellow et al., 2015, Nguyen et al., 2015, Papernot et al., 2016b].

Specifically, suppose we have a classifier F_θ with model parameters θ (we may omit θ for brevity when the context is clear). Let x be an input to the classifier with corresponding ground truth label y. An adversarial example x^* is some instance in the input space that is close to x by some distance metric $d(x, x^*)$ which causes F_θ to produce an incorrect output. In order to isolate the effects of an attack from model inaccuracy, we only consider those x originally satisfying $F_\theta(x) = y$.

Prior work considers two classes of adversarial examples. First, an *untargeted* adversarial example is an instance x^* that causes the classifier to produce any incorrect output: $F_\theta(x^*) \neq y$. Second, a *targeted* adversarial example is an x^* that causes the classifier to produce a specific incorrect output y^*: $F_\theta(x^*) = y^*$ where $y \neq y^*$.

Defenses To improve the robustness of models against adversarial examples, prior work investigates in two directions. The first direction attempts to produce correct predictions on adversarial examples, while not compromising the accuracy on legitimate inputs [Papernot et al., 2016c, Goodfellow et al., 2015, Gu and Rigazio, 2014, Mądry et al., 2017, Cao and Gong, 2017]. The other direction instead attempts to *detect* adversarial examples, without introducing too many false positives. In this case, the model can reject an instance and refuse to classify those that it detects as adversarial [Metzen et al., 2017, Grosse et al., 2017, Xu et al., 2017a, Abbasi and Gagné, 2017]. Many defenses that have been proposed have later been shown to be ineffective in settings where an attacker is aware of the defense in use [Carlini and Wagner, 2017a,b, Athalye and Carlini, 2018].

Threat models Research on defenses has considered different threat models, which we distinguish with two properties: (i) how much information the attacker has about the model and (ii) how much information the attacker has about any defenses in use.

The level of knowledge an attacker has about the model divides attacks into white-box and black-box attacks. In a *white-box* attack, the attacker has full knowledge of the model, including the model architecture, training data, and parameters. Prior work has shown that attacks can also be performed with less information about the model, in *black-box* attacks. One technique for using white-box attack methods in a black-box setting is *transfer*—adversarial examples generated for one model can successfully fool other models, even models of different architectures and models trained on different data [Goodfellow et al., 2015, Papernot et al., 2016a]. An attacker can thus train a model of its own and generate adversarial examples to fool a black-box model [Papernot et al., 2017, Liu et al., 2017a].

We additionally consider static and adaptive adversaries. A *static adversary* is not aware of any defenses that may be in place to protect the model against adversarial examples. A static adversary can generate adversarial examples using existing methods but does not tailor attacks to any specific defense. An *adaptive adversary* is aware of the defense methods used in the model and can adapt attacks accordingly. This is a strictly more powerful adversary than a static adversary. In this dissertation, we focus on adaptive attackers because it is hard to generalize when it is appropriate to assume that adversaries will all be static attackers.

Common experimental setup

In this dissertation, we use the following common data, models, attack methods, and performance metrics.

Datasets and models. To evaluate the effectiveness of the different defense strategies, we use two standard datasets, MNIST [LeCun, 1998] and CIFAR-10 [Krizhevsky and Hinton, 2009] datasets. MNIST has 28×28 pixel black-and-white images (784 dimensions) of handwritten digits. CIFAR-10 has 32×32 pixel RGB natural images (3,072 dimensions) of ten categories of objects: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

We use a collection of small CNNs for MNIST. For CIFAR-10, we use residual networks [He et al., 2016] and wide residual networks [Zagoruyko and Komodakis, 2016]. In our experiments, we use a ResNet32 and a wide ResNet34, with a widening factor of 10.

Adversarial example generation methods. Previous work describes methods to generate adversarial examples from given benign images. We use include the following well known attacks in our experiments.

The Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2015] takes a fixed-size step in the direction of a misclassification. This generates images at a fixed L_∞ distance from the original image (modulo image box constraints).

Carlini and Wagner's approach, which is shown to be effective on finding adversarial examples with small distortions, uses an optimizer to minimize a loss function [2017c]:

$$\text{loss}(x') = \|x' - x\|_2^2 + c \cdot J(F_\theta(x'), y)$$

Here, F_θ is a part of the trained classifier that outputs a vector of logits, and J computes some penalty based on the logits and some label y, either a ground truth label for non-targeted attacks or a target label for targeted attacks. A constant c is a hyperparameter that adjusts the relative weighting between distortion and misclassification. We omit details of the design choice and refer the reader to the original paper.

Performance measurements. We measure an attack's success rate on a model as the fraction of benign inputs for which the attack can generate an adversarial example that the model misclassifies (or classifies as the target class for targeted attacks), among benign examples that were originally

correctly classified. For examples where an attack method successfully generates an adversarial example, we measure the *distortion* between an adversarial example and the original input. The metrics we use for distortion include the root-mean-squared (RMS), the L_2-norm, and L_∞-norm of their distance. When we evaluate defenses, we measure the accuracy of the system on adversarial examples.

1.2 Related work

We give an overview of other work related to this dissertation.

Adaptive attack evaluation. Previous work, notably by Carlini and Wagner [2017a], has evaluated earlier defenses including several that were initially developed for static attackers. They found that adaptive attackers can effectively evade these defenses. In Chapter 2, we focus on newer defenses that have undergone some testing on strong adaptive adversaries already.

Adversarial examples in feature space. Previous has examined local neighborhoods around adversarial examples in the feature space of deep learning models. Liu et al. [2017b] and Tramèr et al. [2017b] examine limited regions around benign samples to study why some adversarial examples transfer across different models. Madry et al. [2017] explore regions around benign samples to validate the robustness of an adversarially trained model. Tabacof and Valle [2016] examine regions around adversarial examples to estimate the examples' robustness to random noise. Cao and Gong [2017] determine that considering the region around an input instance produces a more robust classification than looking at the input instance alone as a single point. In Section 3.1, we examine larger neighborhoods: in many directions and at greater distances.

Query based adversarial examples. In concurrent work, Chen et al. [2017] propose to use finite differences to replace back-propagated gradients in existing optimization based methods for generating adversarial examples. In Section 3.2, we consider a technique that also uses finite differences [Bhagoji et al., 2018], but more closely related to the fast gradient sign method (FGSM) [Goodfellow et al., 2015] and iterative FGSM. Brendel et al. [2018] propose a different method for generating adversarial examples when an attacker can query a model, but where the confidence level is not available in the query output. Their method searches for the model's decision boundaries. We also propose to query a model to find decision boundaries in Section 3.1, but with the goal of characterizing adversarial and benign examples.

Using generative adversarial networks. Zhao et al. [2018] propose to use generative adversarial networks (GANs) to generate especially realistic adversarial examples. In Section 3.2, we evaluate an attack that also uses GANs to generate realistic adversarial examples [Xiao et al., 2018a], but which additionally keeps the generated examples very close to the original images.

Chapter 2

Evaluating defenses under adaptive adversaries

In this section, we perform in-depth evaluation of four defenses against adaptive adversaries. Previously, Carlini and Wagner [2017a] have demonstrated adaptive attack methods that evade proposed defenses, several of which they claim simply arose from insufficient testing when the defenses were validated. With a variety of defenses that involve input pre-processing, secondary classifiers, and distributional detection shown to have weaknesses, we turn our attention to different approaches. We study representative examples of (i) defenses that use ensembles of detection methods and (ii) defenses that incorporate behaviors that the adversary can't predict.

In these cases, it is less clear how to apply existing attack methods, so it is difficult to determine how robust these defenses are against a general adaptive adversary. We present some new techniques to experiment with for evaluating robustness by walking through our own attacks on these defenses.

2.1 Ensemble defenses

We consider defenses that attempt to combine multiple (somewhat weaker) defenses to construct a larger strong defense. In particular, we look at three instances of ensemble defense strategies. First and second are feature squeezing [Xu et al., 2017a] and the *specialists+1* ensemble method [Abbasi and Gagné, 2017], both of which take this approach by construction. These defenses are constructed from components that are intended to be useful together. Their authors have shown that these defenses effectively detect low-perturbation adversarial examples generated by a static adversary. Third, to study the effectiveness of ensembling defenses more broadly, we merge together many detectors that were not designed to be used in conjunction with any other detector. In particular, as an example demonstration, we ensemble three independent detection mechanisms [Gong et al., 2017, Metzen et al., 2017, Feinman et al., 2017] to build one detection mechanism.

For each of these defense strategies, we propose attack methods to generate adversarial examples as an adaptive adversary against the individual component defense (when applicable) as

well as the composite defense strategy. We use these attack methods to evaluate each component defense and composite defense: if our method succeeds at generating adversarial examples, this means that an adaptive adversary can defeat the defense. To gauge how strong the combined defense is compared to the components, we compare the level of distortion needed to fool each (using the same optimization method).

Experimental setup

For the MNIST and CIFAR-10 datasets, we randomly sample 100 images in the test set, filter out examples that are not correctly classified, and generate adversarial examples based on the correctly classified images. When evaluating each defense strategy, we use the same model architectures described in their papers respectively [Xu et al., 2017a, Abbasi and Gagné, 2017, Gong et al., 2017, Metzen et al., 2017, Feinman et al., 2017].

Our experiments took up to three minutes to generate each adversarial example. The attacks we use can scale up to larger models, which require more computation per optimization step. On the other hand, prior work has shown that larger models are actually easier to fool, with lower-distortion adversarial examples or better success at a fixed level of distortion [Goodfellow et al., 2015, Moosavi-Dezfooli et al., 2016, Tabacof and Valle, 2016, Carlini and Wagner, 2017c]. Our own results agree, with adversarial examples on a ResNet32 for CIFAR-10 having significantly lower distortion than adversarial examples on a smaller CNN for MNIST (a much smaller dataset). We expect even larger datasets would be even easier to attack.

Adaptive attacks on feature squeezing

In this and next section, we investigate ensemble defense strategies that are intentionally constructed to have component defenses which work together to detect adversarial examples. The first defense we study is *feature squeezing*, proposed by Xu et al. [2017a,b].

Background: feature squeezing defense. To perform feature squeezing, one generates a lower fidelity version of the input image through a process known as "squeezing" before passing it into the classifier. Xu et al. proposed two methods of squeezing: reducing the color depth to fewer bits, and spatially smoothing the pixels with a median filter. According to their paper, the two methods of squeezing work well together because they address two major kinds of perturbation used in adversarial examples: color depth reduction eliminates small changes to many pixels, while spatial smoothing eliminates large changes to a few pixels.

In order to detect adversarial examples, Xu et al. propose a system combining the two squeezing methods. First, the system runs the classifier on three different versions of the image: the original image, the reduced-color-depth version and the spatially smoothed version of the original image. Then, it compares the softmax probability vectors across these three classifier outputs. The L_1 *score* of the input is the highest L_1 distance between any pair of softmax probability vectors among the three. It flags inputs where the L_1 score exceeds a threshold as adversarial.

In their experiments, Xu et al. show that MNIST and CIFAR-10 classifiers are accurate on squeezed inputs. On adversarial examples generated by a static adversary using FGSM [Goodfellow et al., 2015] and JSMA [Papernot et al., 2016b], they show that their detector achieves 99.74% accuracy on a test set with equal portions benign and adversarial examples. They also show that squeezing the input alone prevents 84 – 100% of the adversarial examples (correctly classifying them). Recently, Xu et al. showed that a simplified detector that uses the original version of the input and the spatially-smoothed version (excluding the color-depth-reduced version) achieves a 98.80% overall detection accuracy on MNIST and 87.50% on CIFAR-10 against a static adversary using a variety of Carlini and Wagner's attacks [Xu et al., 2017b].

Summary of our approach and results. We demonstrate that feature squeezing is not an effective defense in two stages. First, we show that an adaptive attacker can construct an adversarial example that remains adversarial after it is squeezed by each method (color depth reduction and spatial smoothing). Then, we use this approach to construct adversarial examples that are classified the same way both with and without squeezing, causing the L_1 score to be smaller than a given fixed threshold. Our results show that the combined detection method is not effective against an adaptive attacker.

Evading individual feature squeezing defense components

In these experiments, we evaluate whether adversarial examples are robust to each individual feature squeezing defense component, i.e., whether adversarial examples remain adversarial after each individual feature squeezing process (color depth reduction and spatial smoothing) separately. These experiments attack the components of the combined feature squeezing detection scheme. Performing this attack is necessary for defeating the combined detection scheme, wherein the predicted label probabilities of squeezed images are compared against each other.

Evading color-depth-reduction defense

The first method of squeezing an image that Xu et al. propose is color depth reduction. This method rounds each value in the input to 2^b evenly spaced values spanning the same range, which we refer to as reducing to b bits.

Attack Approach. We use Carlini and Wagner's method described in Section 1.1 to generate adversarial examples that are robust to color depth reduction. After each step of the optimization procedure, an intermediate image (perturbed from the original image) is available from the optimizer. We check if a reduced-color-depth version of this intermediate image is adversarial. We run the optimization multiple times, initializing the optimization with random perturbations of the original image each time so that it explores different optimization paths. For each original image, we keep the successful adversarial example that has the lowest L_2 distance to the original image among all the generated successful adversarial examples for this original image.

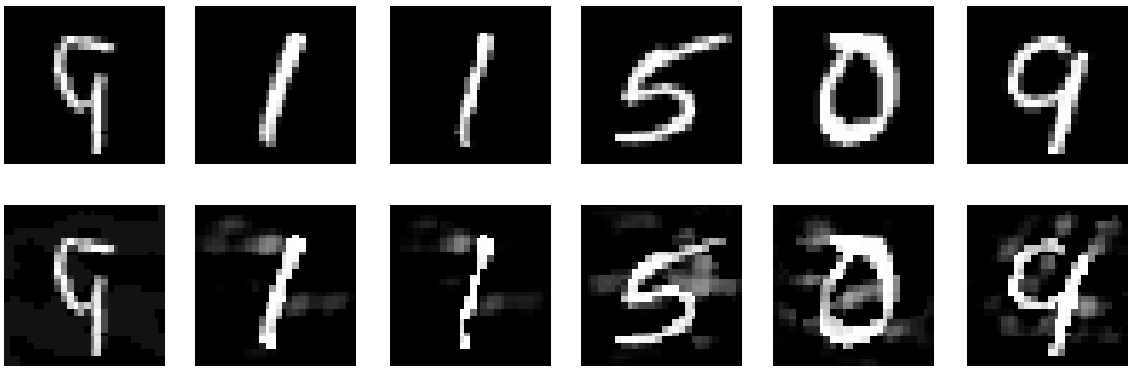

Figure 2.1: Adversarial examples for color depth reduction (to 1 bit) on MNIST. First row: original images. Second row: adversarially perturbed. L_2 distortions, from left to right: 1.49, 2.61, 2.63, 3.83, 3.89, 3.90.

Bit depth	Adv success	Avg L_2
1	100%	3.86
2	99%	1.69
3	100%	1.43
4	100%	1.39
5	100%	1.44
6	100%	1.33
7	100%	1.33
8	100%	1.38

Table 2.1: Summary of MNIST adversarial examples that are misclassified when reduced to different color depths. "Adv success" measures the fraction of original images for which we successfully found an adversarial example. "Avg L_2" measures the average L_2 distortion of the successful adversarial examples.

Attack results on MNIST. We evaluate color depth reduction to 1 – 7 bits. On the strongest defense evaluated by Xu et al., which reduces color depth to 1 bit, we successfully generated adversarial examples for *all* original images, with an average L_2 distortion of 3.86. Figure 2.1 shows a sample of these adversarial examples.

Table 2.1 summarizes our results for other bit depths. Notice that for a system without any color depth reduction (retaining the original 8 bits of depth), we find adversarial examples with an average L_2 distortion of 1.38. Reducing color depth to fewer bits makes the system less sensitive to small changes, which requires larger distortions; however, the distortions are still very small.

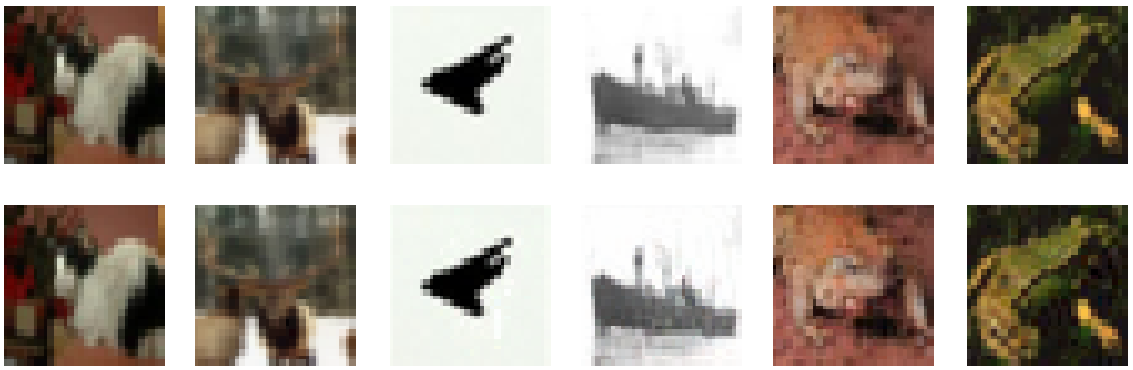

Figure 2.2: Adversarial examples for color depth reduction (to 3 bits) on CIFAR-10. Distortions, from left to right: 0.0194, 0.0954, 0.322, 0.942, 0.948, 0.948. Layout is the same as Figure 2.1.

Attack results on CIFAR-10. We evaluate color depth reduction to 3 bits, which Xu et al. recommend as a good balance between the accuracy on adversarial inputs and accuracy on benign images for CIFAR-10. We succeeded at generating adversarial examples for *all* original images, with an average L_2 distortion of 0.945. Figure 2.2 shows a sample of these adversarial examples. For comparison, adversarial examples for a classifier without color depth reduction have an average L_2 distortion of 0.214. Although this method of squeezing increases the distortion needed for successfully generating non-targeted adversarial examples using the same optimization method, again, such a distortion is still small and imperceptible.

Summary. An adaptive attacker can successfully generate adversarial examples with small distortions for a system that applies color depth reduction to the input image before classifying it.

Evading spatial smoothing

Xu et al. propose a second method for feature squeezing, which applies a median filter to the input, which replaces each pixel with the median value of a neighborhood around the pixel.

To generate adversarial examples that are misclassified after spatial smoothing, we use Carlini and Wagner's method from Section 1.1 with the addition of a median filter as part of the classification model.

A median filter for TensorFlow was not available, so we implemented our own.

Attack results on MNIST. We evaluate a range of median filter sizes, ranging from 1×2 to 5×5. For a 3×3 filter, with which Xu et al. achieved the best accuracy, we successfully generated adversarial examples for *all* original images, with an average distortion of 1.29. Figure 2.3 shows a sample of these adversarial examples. Table 2.2 summarizes our results for other filter sizes. Larger

Filter size	Adv success	Avg L_2
3×3	100%	1.29
2×2	100%	1.57
5×5	100%	0.612
3×1	100%	1.33
1×3	100%	1.29
2×1	100%	1.52
1×2	100%	1.51
5×1	100%	0.943
1×5	100%	0.931

Table 2.2: Summary of MNIST adversarial examples that are misclassified when spatially smoothed with varying sizes of median filters. Columns have the same meaning as in Table 2.1. Some filters make adversarial examples *easier* to find.

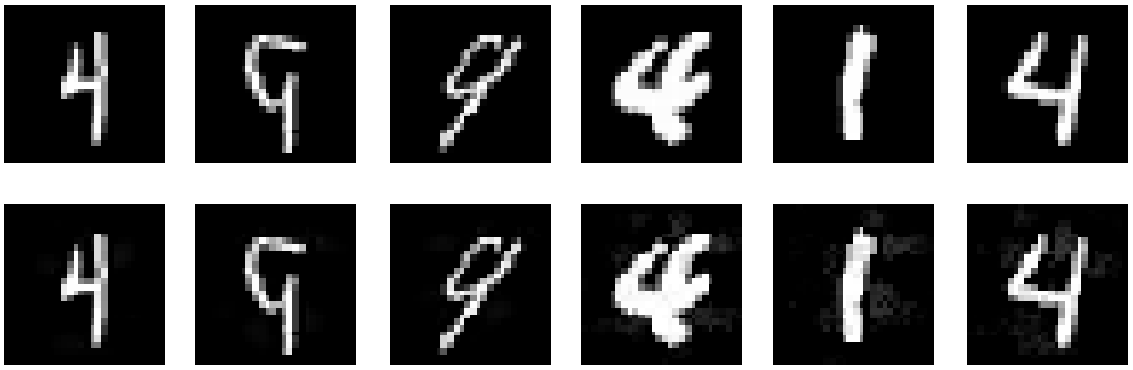

Figure 2.3: Adversarial examples for spatial smoothing (with 3×3 filter) on MNIST. Distortions, from left to right: 0.236, 0.241, 0.282, 1.27, 1,31, 1.31. Layout is the same as Figure 2.1.

median filters did not require greater distortion. Compared to adversarial examples generated for a system without any spatial smoothing (average distortion of 1.38), the average distortion is not increased.

Attack results on CIFAR-10 We evaluate a 2×2 median filter, which Xu et al. identify as achieving a good rejection rate of adversarial examples and accuracy on benign images on CIFAR-10. We successfully generated adversarial examples for *all* original images, which have an average distortion of 0.205. Figure 2.4 shows a sample of these adversarial examples. The average distortion is not higher than for a system without spatial smoothing (0.214).

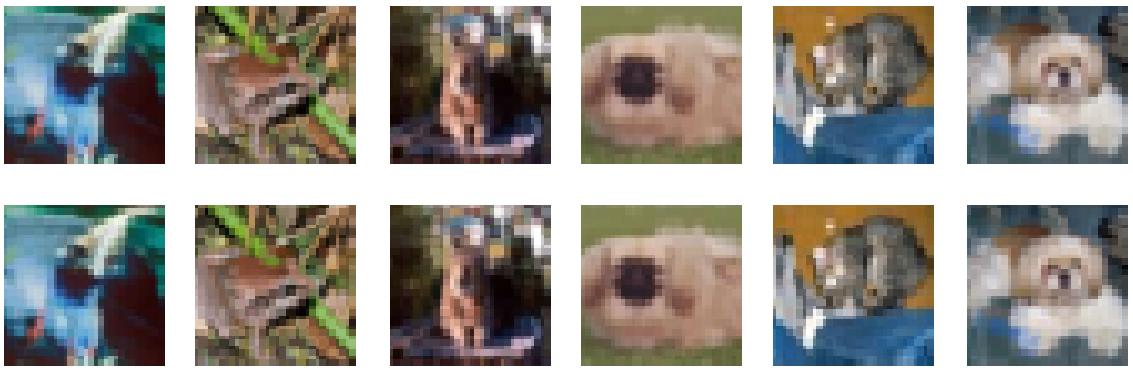

Figure 2.4: Adversarial examples for spatial smoothing (with 2×2 filter) on CIFAR-10. Distortions, from left to right: 0.0273, 0.0537, 0.0584, 0.198, 0.211, 0.212. Layout is the same as Figure 2.1.

Summary Spatial smoothing alone is not an effective defense against an adaptive attacker. We have shown that an adaptive adversary can create adversarial examples for a system that applies spatial smoothing which are not more distorted than adversarial examples for a baseline system that does not apply spatial smoothing.

Evading combination of multiple squeezing techniques

While the individual feature squeezing techniques are weak against an adaptive attacker, we investigate whether the detection scheme that combines them is stronger. In this case, we find that this detection scheme is not much stronger than the strongest component defense, color depth reduction.

Background: Composite feature squeezing defense The detection scheme combines both methods of squeezing. In particular, the detection system has three "branches," where each one runs the classifier on a different version of the input, the original input, a reduced-color-depth version and a spatially-smoothed version of the original input. These three branches output different softmax probability vectors, and the scheme compares the L_1 distance between these vectors against a threshold to determine whether the input is adversarial.

In this experiment, we evaluate the effectiveness of the combined defense against an adaptive attacker, i.e., whether adversarial examples can be misclassified and bypass this detection scheme.

Attack Approach. We use the procedure from the color-depth-reduction evasion attack with modifications to support the detection mechanism. In this attack, we run two instances of the

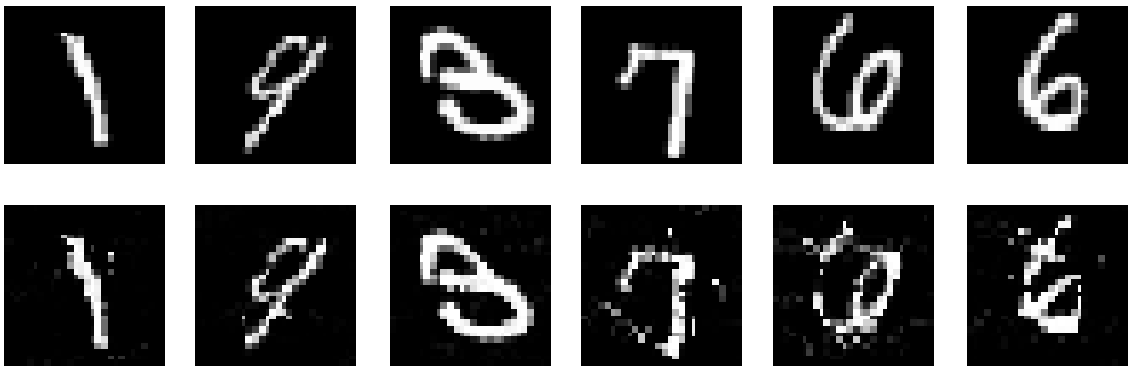

Figure 2.5: Adversarial examples for combined feature squeezing detection on MNIST. Distortions, from left to right: 2.00, 2.04, 2.39, 4.66, 4.77, 4.79. Layout is the same as Figure 2.1.

model in parallel during optimization, representing the detector branches that operate on the input and the spatially smoothed version of the input. The optimizer receives gradients from both models, equally. We do not include the reduced-color-depth branch in the gradient calculation, because the color depth reduction removes gradient information; it is, however, included when we compute the L_1 score. We collect only adversarial examples that have an L_1 score below a threshold of 0.3076, a level at which Xu et al. achieved the best accuracy in their experiments on MNIST.

Attack results on MNIST We evaluate a combination of color depth reduction to 1 bit and smoothing with a 2×2 median filter, which Xu et al. found to be accurate on adversarial examples generated by a static adversary [Xu et al., 2017b]. We successfully generated adversarial examples for *all* original images, with an average distortion of 4.76 and L_1 score of 0.209. Figure 2.5 shows a sample of these adversarial examples. These examples are misclassified and successfully evade detection. This distortion is 23.3% larger than for color depth reduction alone, but still very small.

Attack results on CIFAR-10. We evaluate a combination of color depth reduction to 3 bits and smoothing with a 2×2 median filter, a combination of settings that perform well in Xu et al.'s experiments. We successfully generated adversarial examples for *all* original images, with an average distortion of 0.601 and L_1 score of 0.168. Figure 2.6 shows a sample of these adversarial examples. These examples are misclassified and successfully evade detection.

This distortion is even lower than that of the color depth reduction defense alone. Although Xu et al. do not prescribe a threshold specific to CIFAR-10, the average L_1 score for these examples is lower (i.e., detected as less adversarial) than the average L_1 score for the original images, which is 0.225.

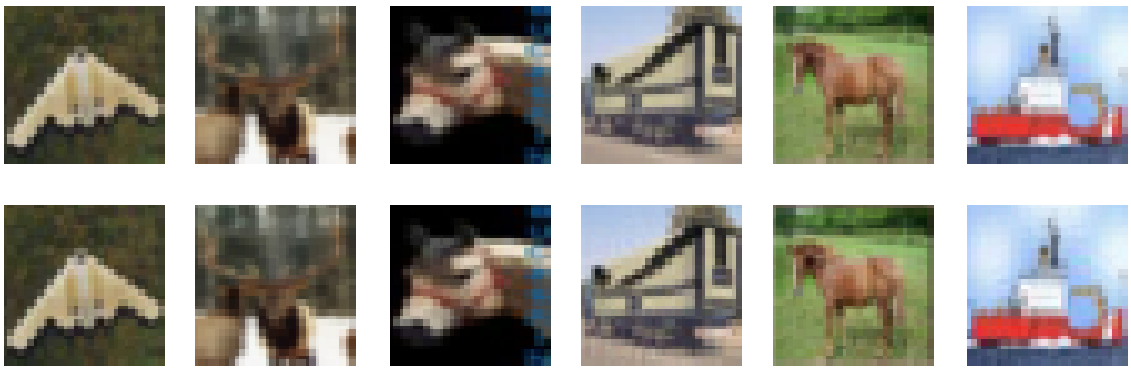

Figure 2.6: Adversarial examples for combined feature squeezing detection on CIFAR-10. Distortions, from left to right: 0.117, 0.120, 0.130, 0.604, 0.614, 0.617. Layout is the same as Figure 2.1.

Summary. The detection scheme that combines two methods of squeezing is not always stronger than the strongest component, color depth reduction. The improvement is low even on MNIST, which is particularly well suited for feature squeezing, with images being black and white (little change from color depth reduction) and having large, contiguous areas of the same color (little change from spatial smoothing). On CIFAR-10, the combined attack requires less distortion than the color depth reduction defense alone.

Evading ensemble of specialists

We study a second defense that combines multiple component defenses, an ensemble of specialists, proposed by Abbasi and Gagné [2017].

Background: ensemble of specialist defense. The defense consists of a generalist classifier (which classifies among all classes) and a collection of specialists (which classify among subsets of the classes). The specialists classify subsets of the classes as follows. Where C is the set of all classes in the task, for each class i, let U_i be the set of classes with which i is most often confused in adversarial examples. To compute U_i, Abbasi and Gagné select the top 80% of misclassifications caused by non-targeted FGSM attacks for each class i. Further, $K = |C|$ additional subsets are defined: $U_{K+i} = C \setminus U_i$ to be the complement set of U_i. For each $j = 1, ..., 2K$, a specialist classifier F_j is trained on a subset of the dataset containing images belonging to the classes in U_j to classify input images into the classes in U_j only. In addition, a generalist classifier F_{2K+1} is trained to classify input images into classes in $U_{2K+1} = C$. Each classifier in the ensemble may be susceptible to basic adversarial examples, but the proposed defense assumes that each specialist can detect a few specific attacks, thus the attacker cannot fool all specialists and the generalist at the same time. The defense combines them to jointly detect general adversarial examples.

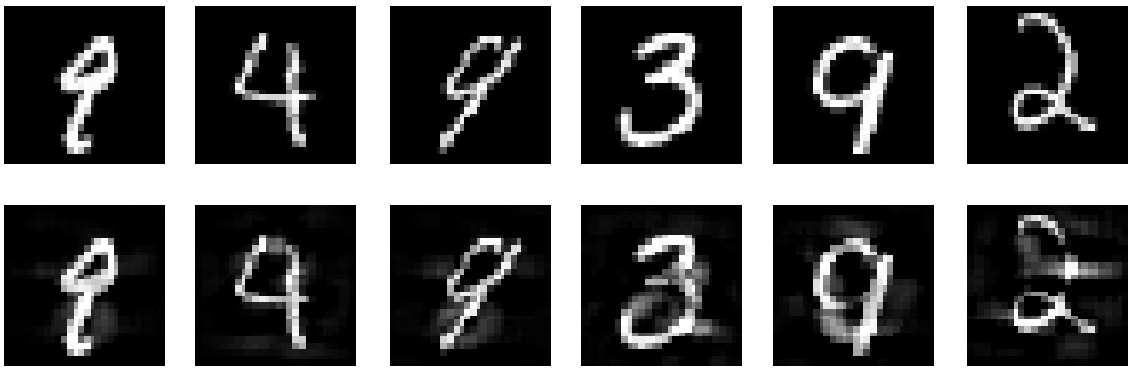

Figure 2.7: Adversarial examples for specialists+1 on MNIST. Distortions, from left to right: 1.55, 1.76, 1.83, 3.77, 3.90, 3.93. Layout is the same as Figure 2.1.

In order to classify an input, the system first checks if, for any class i, the generalist classifier and all specialists that can classify i agree that the input belongs to class i. If such a class i exists, note that at most one class can get the generalist's vote, it must be unique. In this case, the system takes the mean of the outputs of the generalist and the specialists that can classify i. Otherwise, at least one classifier has misclassified the input, and the system takes the mean of the outputs of all classifiers in the ensemble.

Abbasi and Gagné [2017] find that using an ensemble constructed this way successfully reduces the system's confidence (mean confidence among classifiers activated by the voting scheme) on adversarial examples generated by a static attacker using FGSM [Goodfellow et al., 2015], DeepFool [Moosavi-Dezfooli et al., 2016], and Szegedy et al.'s approach [2014b]. They conclude that a classification system can use an ensemble of diverse specialists this way and detect low-confidence examples as adversarial.

Attack approach. In this experiment, we evaluate the effectiveness of Abbasi and Gagné's specialists+1 ensemble against an adaptive attacker. We considered a scenario where a user provides an image to a system, and the system uses a specialists+1 ensemble to classify the image or reject it as adversarial.

We attempt to create targeted adversarial examples, where we chose target classes randomly. For each original image, then our goal is to create an adversarial example that is classified as the target class by the generalist classifier and all applicable specialists at the same time, and with high confidence from those classifiers. We adapt Carlini and Wagner's method [2017c] to generate adversarial examples. In this experiment, we kept only adversarial examples that were misclassified with confidence greater than the average confidence on a sample of benign images, 0.999708. We modified the loss function to support multiple classifiers:

$$\text{loss}(x') = \|x' - x\|_2^2 + c \sum_{j \in \{1,\ldots,2K+1\}; y^* \in U_j} J(F_j(x'), y^*)$$

We evaluate this defense on MNIST only. While Abbasi and Gagné also propose the defense for CIFAR-10, the architecture described in their experiments has low accuracy on CIFAR-10, resulting in low confidence even for benign images.

Attack results on MNIST. We successfully generated adversarial examples for *all* original images, which have an average L_2 distortion of 3.87. Figure 2.7 shows a sample of these adversarial examples in the second row. These adversarial examples are classified as the target label by the generalist and all applicable specialists. For comparison, the average confidence of a single generalist classifier on correctly classified benign images is 0.998951, and a batch of targeted adversarial examples with at least that confidence has average distortion 3.65. The distortion needed for high-confidence adversarial examples on specialists+1 is 6.03% higher than for a non-ensemble MNIST classifier.

Although this defense defines the specialists to focus on common misclassifications caused by non-targeted adversarial examples, it is still weaker at detecting the common misclassifications. Among the examples, 33 targeted a class that the original image's ground truth class was commonly confused with. The average distortion for these images is 3.06, below the average of the entire set.

Summary. The specialists+1 ensemble does not effectively ensure low confidence on adversarial examples generated by an adaptive attacker. An adaptive attacker can successfully generate adversarial examples with small distortions, which are unanimously classified as a target class, and thus evade the detection of the specialist+1 ensemble defense.

Evading ensemble of detectors

In the previous sections, we have investigated ensembles of defenses that are intentionally constructed to be useful together. In Xu et al.'s work, the color depth reduction is intended to remove small changes to many pixels, and the median smoothing to remove large changes to a few pixels. Similarly, Abbasi and Gagné propose using an ensemble of generalist and specialist classifiers together; without the others, this approach would not be useful.

To study the effectiveness of ensembling defenses more broadly, we merge together three recently proposed detectors that were not designed to be used in conjunction with any other detector. We consider only detectors that are applied to a fixed classification network for simplicity, and therefore study the following schemes:

- Gong et al. [2017] propose using adversarial training to detect adversarial examples. Given the original model, generate adversarial examples on the training data. Then, train a new classifier that distinguishes the original training data from the adversarial data.
- Metzen et al. [2017] construct a similar scheme, however instead of using the original images as the input to the detector, they train on the inner convolutional layers of the network.

		Source Defense		
		Gong	Metzen	Feinman
Target	Gong	100%	51%	21%
	Metzen	43%	100%	18%
	Feinman	96%	92%	100%

Table 2.3: Probability that adversarial examples constructed for a given source defense also fool the given target defense on CIFAR-10. Defenses generated against Metzen et al. transfer to the others with the highest probability, and Feinman et al. with the lowest.

- Feinman et al. [2017] examine the final hidden layer of a neural network and find that adversarial examples are separable from the original images by training a density estimate using Gaussian kernels.

When using Carlini and Wagner's attack, these approaches are known to provide only slight increases in robustness, i.e., only increase the required distortion slightly when generating the adversarial examples with the detector vs. without the detector [2017a]. Given this, we now examine if constructing an ensemble of these defenses provides additional robustness. To ensemble these defenses, we run each detection method and report the input as adversarial if any of the three detectors do.

Attack approach. We perform this experiment on CIFAR-10 exclusively, as Metzen et al.'s defense is intended for a ResNet applied to CIFAR-10. We are able to construct adversarial examples for all defenses independently. To defeat all three defenses together, we construct a new classifier $G(\cdot)$ so that using the loss function from Section 1.1 directly can construct adversarial examples.

We use the same notation as Carlini and Wagner [2017a]. Let $F(\cdot)$ be a classifier on N classes, and $\text{softmax}(F(\cdot))_i$ be the probability of class i (so that $F(\cdot)_i$ are the logits). Let $\{D_j(x)\}_{j=1}^{J}$ be one of J different detectors so that the probability that detector D_j reports object x as adversarial is $\text{sigmoid}(D_j(x))$ (that is, D_j returns the logits). We report that an instance is adversarial if the probability of any detector is greater than one half. That is, if for any j, $\text{sigmoid}(D_j(x)) > \frac{1}{2}$, or, alternatively, $D_j(x) > 0$.

When we ensemble the three defenses, we set $J = 3$ and define $D(x) = \max_j D_j(x)$, so that $D(x)$ reports adversarial (i.e., $D(x) > 0$) if any of the three detectors do.

Given this, we use the same $G(\cdot)$ construction as Carlini and Wagner's previous work on these defenses [Carlini and Wagner, 2017a]. This function $G(\cdot)$ returns $N + 1$ classes (with the new class reserved for adversarial examples) so that $\arg\max_i G(x)_i = \arg\max_i F(x)_i$ when x is not adversarial, and $\arg\max_i G(x) = N + 1$ when x is adversarial. To do this, Carlini and Wagner

[2017a] specifically defines

$$G(x)_i = \begin{cases} F(x)_i & \text{if } i \leq N \\ (D(x)+1) \cdot \max_j F(x)_j & \text{if } i = N+1 \end{cases}$$

If for a given instance x, $D_j(x) > 0$ (for any classifier j) then we will have $\arg\max_i G(x)_i = N+1$ since we multiply a value greater than one by the largest of the other output logits. Conversely, if $\arg\max_i G(x)_i \neq N+1$ then we must have $D(x) < 0$ implying that all detectors report the instance is benign.

Therefore, by constructing adversarial examples on G so that the target class is not $N+1$, we can construct adversarial examples on F that are not detected by any detector.

Attack results on CIFAR-10. The L_2 distortion required to construct adversarial examples on an unsecured network is 0.11. To construct adversarial examples on this network $G(\cdot)$ with the three defenses increases the distortion to 0.18, an increase of 60%. However, this distortion is still imperceptible.

Transferability of adversarial examples across different detectors. In order to understand the reason that these defenses do not significantly increase robustness when combined together, we hypothesize that the transferability property Szegedy et al. [2014a], Goodfellow et al. [2015], Papernot et al. [2016a], Liu et al. [2017a] of adversarial examples is simplifying the attacker's task. To verify this, we construct adversarial examples on each of the three defenses in isolation and check the probability that these examples also fool the other two defenses. Table 2.3 contains this data. Feinman's defense is the weakest of the three, and so transfers least often (and adversarial examples transfer to it most often). The other two defenses are approximately equally effective. From this, we can see one possible reason why constructing an ensemble of these weak defenses is not significantly more secure than each independently: the adversarial examples that fool one detector may also fool the other detectors. We conclude that one must be careful when ensembling defenses to build them to cover the weaknesses of the others, and not simply assemble them blindly.

Conclusion

In this section we explore techniques for evaluating ensemble defenses in under an adaptive adversary. We demonstrate our proposed techniques, based on optimization, in examining whether multiple (possibly weak) defenses can be combined to create a strong defense. We studied three such defenses that combined multiple components: two defenses designed with a rationale of why their components should work well together and one that combined unrelated recently proposed detectors.

We showed that an adaptive adversary can generate adversarial examples with low distortion that fool all of the defenses that we evaluate. The feature squeezing detection scheme, which combines two methods of squeezing an input image, is at best marginally stronger than color

depth reduction alone. The specialists+1 ensemble, which combines several specialist classifiers, increases the required distortion slightly, but again, distortion is still small. We also showed that combining a collection of recently proposed detection mechanisms is also ineffective. In particular, our results show that adversarial examples transfer across the individual detectors.

This work sheds light on a few important lessons when evaluating defenses against adversarial examples: (i) one should evaluate defenses using strong attacks. For example, FGSM can quickly generate adversarial examples, but may fail to generate successful attacks when other iterative optimization based methods can succeed; and (ii) one should evaluate defenses using adaptive adversaries. It is important to develop defenses that are secure against attackers who know the defense mechanisms being used.

Our results indicate that combining weak defenses does not automatically improve the robustness of these systems.

2.2 Non-deterministic defenses

Next, we consider defenses that incorporate behaviors that an attacker cannot predict.

Defenses

In this section, we discuss a non-deterministic defense (from among many), region classification. We also discuss a defense that combines region classification with adversarial training.

Region classification

Cao and Gong [2017] propose *region classification*, a defense against adversarial examples that takes the majority prediction on several slightly perturbed versions of an input, uniformly sampled from a hypercube around it. This approximates computing the majority prediction across the neighborhood around an input as a region. In contrast, the usual method of classifying only the input instance can be referred to as *point classification*.

Cao and Gong show that region classification approach successfully defends against low-distortion adversarial examples generated by existing attacks, and they suggest that adversarial examples robust to region classification, such as Carlini and Wagner's high-confidence attack, have higher distortion and can be detected by other means.

Adversarial training

Adversarial training modifies the training procedure, substituting a portion of the training examples with adversarial examples. We experiment with Madry et al.'s defense, which performs adversarial training using PGD, an attack that follows the gradient of the model's loss function for multiple steps to generate an adversarial example.

Background and experimental setup

Datasets. We use two popular academic image classification datasets for our experiments: MNIST and CIFAR-10. In these experiments, the MNIST images' pixel values are in the range $[0, 1]$; in CIFAR-10, they are in $[0, 255]$.

Adversarial examples. For simplicity, we focus our analysis on untargeted attacks. We quantify the distortion using the root-mean-square (RMS) distance metric between the original input instance and the adversarial example. This is similar to the L_2-norm, but the RMS normalizes for different image sizes.

Models. For each dataset, we perform experiments on two models trained from one architecture. For MNIST, the architecture is a convolutional neural network;[1] for CIFAR-10, a wide ResNet34.[2] In order to study the effect of PGD adversarial training on a model's decision regions, from each dataset, we use a defended model trained with the PGD adversarial training defense and an undefended model trained with normal examples. The PGD adversarial training on MNIST used an L_∞ perturbation limit of 0.3; on CIFAR-10, 8.

OPTMARGIN attack on region classification

In this section, we develop a concrete example where limiting the analysis of a neighborhood to a small ball leads to evasion attacks on an adversarial example defense.

Proposed OPTMARGIN attack

We introduce an attack, OPTMARGIN, which can generate low-distortion adversarial examples that are robust to small perturbations, like those used in region classification.

In our OPTMARGIN attack, we create a surrogate model of the region classifier, which classifies a smaller number of perturbed input points. This is equivalent to an ensemble of models $f_i(x) = f(x+v_i)$, where f is the point classifier used in the region classifier and v_i are perturbations applied to the input x. Our attack uses existing optimization attack techniques to generate an example that fools the entire ensemble while minimizing its distortion [Liu et al., 2017b, He et al., 2017].

Let $Z(x)$ refer to the $|C|$-dimensional vector of class weights, in logits, that f internally uses to classify image x. For each model in our ensemble, we define a loss term based on the objective function in Carlini and Wagner's L_2 attack [2017c]:

$$\ell_i(x') = \ell(x' + v_i) = \max\left(-\kappa,\ Z(x' + v_i)_y - \max\{Z(x' + v_i)_j : j \neq y\}\right)$$

This loss term increases when model f_i predicts the correct class y over the next most likely class. When the prediction is incorrect, the value bottoms out at $-\kappa$ logits, with κ referred to

[1] https://github.com/MadryLab/mnist_challenge

[2] https://github.com/MadryLab/cifar10_challenge

	MNIST				CIFAR-10			
Examples	Normal		Adv tr.		Normal		Adv tr.	
OPTBRITTLE	100%	0.0732	100%	0.0879	100%	0.824	100%	3.83
OPTMARGIN (ours)	100%	0.158	100%	0.168	100%	1.13	100%	4.08
OPTSTRONG	100%	0.214	28%	0.391	100%	2.86	73%	37.4
FGSM	91%	0.219	6%	0.221	82%	8.00	36%	8.00

Table 2.4: Average distortion (RMS) of adversarial examples generated by different attacks, along with and attack success rate (%) under point classification. On MNIST, the level of distortion in OPTMARGIN examples is visible to humans, but the original class is still distinctly visible (see Figure 2.8 for sample images).

as the confidence margin. In OPTMARGIN, we use $\kappa = 0$, meaning it is acceptable that the model just barely misclassifies its input. With these loss terms, we extend Carlini and Wagner's L_2 attack [2017c] to use an objective function that uses the sum of these terms. Whereas Carlini and Wagner would have one $\ell(x')$ in the minimization problem below, we have:

$$\text{minimize} \quad ||x' - x||_2^2 + c \cdot (\ell_1(x') + ... + \ell_n(x')) \tag{2.1}$$

We use 20 classifiers in the attacker's ensemble, where we choose $v_1, ..., v_{19}$ to be random orthogonal vectors of uniform magnitude ϵ, and $v_{20} = 0$. This choice is meant to make it likely for a random perturbation to lie in the region between the v_is. Adding $f_{20}(x) = f(x)$ to the ensemble causes the attack to generate examples that are also adversarial under point classification.

For stability in optimization, we used fixed values of v_i throughout the optimization of the attack. This idea is similar to Carlini & Wagner's attack [2017a] on Feinman et al.'s stochastic dropout defense [2017].

Distortion evaluation

We compare the results of our OPTMARGIN attack with Carlini and Wagner's L_2 attack [2017c] with low confidence $\kappa = 0$, which we denote OPTBRITTLE, and with high confidence $\kappa = 40$, which we denote OPTSTRONG, as well as FGSM [Goodfellow et al., 2015] with $\epsilon = 0.3$ (in L_∞ distance) for MNIST and 8 for CIFAR-10. In our OPTMARGIN attacks, we use $\epsilon = 0.3$ (in RMS distance) for MNIST and $\epsilon = 8$ for CIFAR-10. Figure 2.8 shows a sample of images generated by each method. Table 2.4 shows the average distortion (amount of perturbation used) across a random sample of adversarial examples.

On average, the OPTMARGIN examples have higher distortion than OPTBRITTLE examples (which are easily corrected by region classification) but much lower distortion than OPTSTRONG examples.

The OPTSTRONG attack produces examples with higher distortion, which Cao and Gong discount; they suggest that these are easier to detect through other means. Additionally, the OPT-

	MNIST				CIFAR-10			
	Region cls.		Point cls.		Region cls.		Point cls.	
Examples	Normal	Adv. tr.	Normal	Adv. tr.	Normal	Adv. tr.	Normal	Adv. tr.
Benign	99%	100%	99%	100%	93%	86%	96%	86%
FGSM	16%	54%	9%	94%	16%	55%	17%	55%
OPTBRITTLE	95%	89%	**0%**	**0%**	71%	79%	**0%**	**0%**
OPTMARGIN (ours)	**1%**	**10%**	**0%**	**0%**	**5%**	**5%**	**0%**	6%

Table 2.5: Accuracy of region classification and point classification on examples from different attacks. More effective attacks result in lower accuracy. The attacks that achieve the lowest accuracy for each configuration of defenses are shown in bold. We omit comparison with OPTSTRONG due to its disproportionately high distortion and low attack success rate.

STRONG attack does not succeed in finding adversarial examples with satisfactory confidence margins for all images on PGD adversarially trained models.[3] The FGSM samples are also less successful on the PGD adversarially trained models. The average distortion reported in Table 2.4 is averaged over only the successful adversarial examples in these two cases. The distortion and success rate can be improved by using intermediate confidence values, at the cost of lower robustness. Due to the low success rate and high distortion, we do not consider OPTSTRONG attacks in the rest of our experiments.

Evading region classification

We evaluate the effectiveness of our OPTMARGIN attack by testing the generated examples on Cao and Gong's region classification defense.

We use a region classifier that takes 100 samples from a hypercube around the input. Cao and Gong determined reasonable hypercube radii for similar models by increasing the radius until the region classifier's accuracy on benign data would fall below the accuracy of a point classifier. We use their reported values in our own experiments: 0.3 for a CNN MNIST classifier and 5.1 (0.02 of 255) for a ResNet CIFAR-10 classifier.

In the following experiments, we test with a sample of 100 images from the test set of MNIST and CIFAR-10.

Table 2.5 shows the accuracy of four different configurations of defenses for each task: no defense (point classification with normal training), region classification (with normal training), PGD adversarial training (with point classification), and region classification with PGD adversarial training.

Cao and Gong develop their own attacks against region classification, CW-L_0-A, CW-L_2-A, and CW-L_∞-A. These start with Carlini & Wagner's low-confidence L_0, L_2, and L_∞ attacks, respectively, and amplify the generated perturbation by some multiplicative factor. They evaluate

[3]We use the official implementation of Carlini and Wagner's high confidence attack, which does not output a lower-confidence adversarial example even if it encounters one.

these in a targeted attack setting. Their best result on MNIST is with CW-L_2-A with a $2\times$ amplification, resulting in 63% attack success rate. Their best result on CIFAR-10 is with CW-L_∞-A with a $2.8\times$ amplification, resulting in 85% attack success rate. In our experiments with OPTMARGIN in an untargeted attack setting, we observe high attack success rates at similar increases in distortion.

These results show that our OPTMARGIN attack successfully evades region classification and point classification.

Performance

Using multiple models in an ensemble increases the computational cost of optimizing adversarial examples, proportional to the number of models in the ensemble. Our optimization code, based on Carlini & Wagner's, uses 4 binary search steps with up to 1,000 optimization iterations each. In our slowest attack, on the PGD adversarially trained CIFAR-10 model, our attack takes around 8 minutes per image on a GeForce GTX 1080.

Although this is computationally expensive, an attacker can generate successful adversarial examples with a small ensemble (20 models) compared to the large number of samples used in region classification (100)—the slowdown factor is less for the attacker than for the defender.

Conclusion

In this section, we explore a technique for evaluating non-deterministic defenses under an adaptive adversary. We demonstrate this technique on region classification, where we show that an attacker can adapt an existing attack to generate adversarial examples that are consistently misclassified within a region. We find that the increase in computational cost for the adapted attack is even smaller than the increase in computational cost to estimate region classification by sampling points.

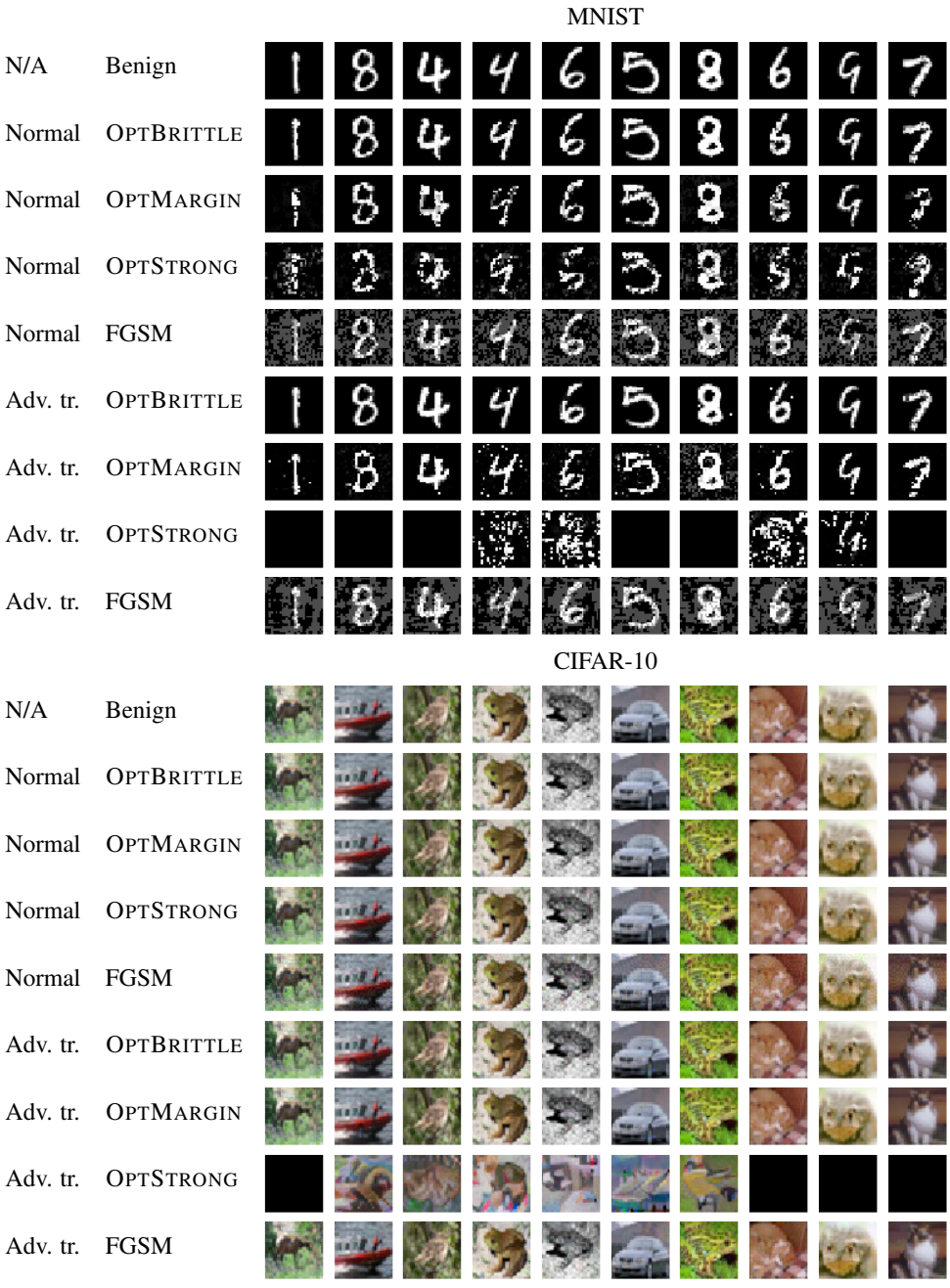

Figure 2.8: Adversarially perturbed images generated by different attack methods, for differently trained models, and their corresponding original images. Instances where the attack does not produce an example are shown as black squares.

Chapter 3

Exercising defenses on diverse attack methods

So far, we have examined cases where a defense that is effective against previously known attacks is much weaker against a new attack from an adaptive adversary. These suggest that some defenses may be over specialized for certain attacks. In this chapter, we explore this idea further by comparing defenses against newer attack methods that demonstrate important departures from previous methods.

We focus on one class of defenses, adversarial training, which has shown promising results on a broad category of attacks based on gradient information. First, we conduct a brute-force examination in a large sample of directions in feature space around natural inputs. Second, we evaluate three new attack methods:

- An attack that uses finite differences to estimate the worst case perturbation rather than computing gradients [Bhagoji et al., 2018].
- AdvGAN [Xiao et al., 2018a], which uses a generative adversarial network (GAN) to generate a perturbation.
- stAdv[Xiao et al., 2018b], which perturbs an input image by spatially shifting its pixels, rather than additively perturbing them.

3.1 Decision boundaries

As a first step in exploring images near natural examples that a given model classifies differently, we study the *decision boundaries* of a model—the surfaces in the model's input space where the output prediction changes between classes. A nearby decision boundary indicates that adversarial examples exist on the other side, while finding boundaries to be far away from benign examples indicates robustness to perturbations. For comparison, we also analyze the decision boundaries around adversarial examples.

Analysis of surrounding decision boundaries

In addition to the goal of finding nearby adversarial examples, we want to characterize the decision boundaries both near and far. We have shown in Section 2.2 that examining a small ball around a given input instance may not adequately distinguish OPTMARGIN adversarial examples, as there exist adversarial examples that are also consistently (mis-)classified in the surrounding region. In this section, we introduce a more comprehensive analysis of the neighborhood around an input instance.

Specifically, we consider the distance to the nearest boundary in many directions and adjacent decision regions' classes.

Decision boundary distance

To gather information on the sizes and shapes of a model's decision regions, we estimate the distance to a decision boundary in a sample of random directions in the model's input space, starting from a given input point. In each direction, we estimate the distance to a decision boundary by computing the model's prediction on perturbed inputs at points along the direction. In our experiments, we check every 0.02 units (in RMS distance) for MNIST (data is in the scale of $[0, 1]$) and every 2 units for CIFAR-10 (data is in the scale of $[0, 255]$). When the model's prediction on the perturbed image changes from the prediction on the original image (at the center), we use that distance as the estimate of how far the decision boundary is in that direction.

When the search encounters a boundary this way, we also record the predicted class of the adjacent region.

For CIFAR-10, we perform this search over a set of 1,000 random orthogonal directions (for comparison, the input space is 3,072-dimensional). For MNIST, we search over 784 random orthogonal directions (spanning the entire input space) in both positive and negative directions, for a total of 1,568 directions.

Individual instances. Figure 3.1 shows the decision boundary distances for a typical set of a benign example and adversarial examples generated as described in Section 2.2 (OPTBRITTLE is an easily mitigated C&W low-confidence L_2 attack; OPTMARGIN is our method for generating robust examples; FGSM is the fast gradient sign method from Goodfellow et al. [2015]). It shows these attacks applied to models trained normally and models trained with PGD adversarial examples. See Figure 3.3 for a copy of this data plotted in L_∞ distance.

The boundary distance plots for examples generated by the basic optimization attack are strikingly different from those for benign examples. As one would expect from the optimization criteria, they are as close to the boundary adjacent to the original class as possible, in a majority of the directions. These plots depict why region classification works well on these examples: a small perturbation in nearly every direction crosses the boundary to the original class.

For our OPTMARGIN attack, the plots lie higher, indicating that the approach successfully creates a margin of robustness in many random directions. Additionally, in the MNIST examples, the original class is not as prominent in the adjacent classes. Thus, these examples are challenging

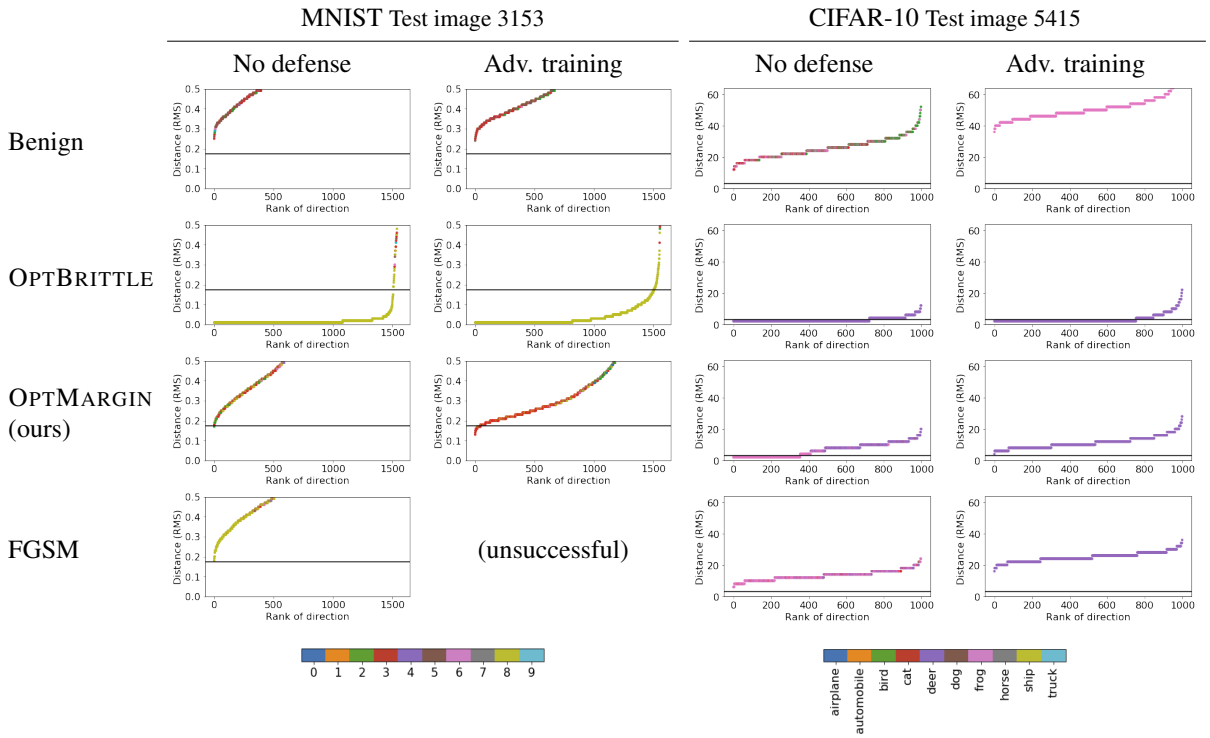

Figure 3.1: Decision boundary distances (RMS) from single sample images, plotted in ascending order. Colors represent the adjacent class to an encountered boundary. A black line is drawn at the expected distance of an image sampled during region classification. Results are shown for models with normal training and models with PGD adversarial training. For MNIST, original example correctly classified 8 (yellow); OPTBRITTLE and OPTMARGIN examples misclassified as 5 (brown); FGSM example misclassified as 2 (green). For CIFAR-10, original example correctly classified as DEER (purple); OPTBRITTLE, OPTMARGIN, and FGSM examples misclassified as HORSE (gray).

for region classification both due to robustness to perturbation and due to the neighboring incorrect decision regions.

Summary statistics. We summarize the decision boundary distances of each image by looking at the minimum and median distances across the random directions. Figure 3.2 shows these representative distances for a sample of correctly classified benign examples and successful adversarial examples. See Figure 3.4 for a copy of this data plotted in L_∞ distance.

These plots visualize why OPTMARGIN and FGSM examples, in aggregate, are more robust to random perturbations than the OPTBRITTLE attack. The black line, which represents the expected distance that region classification will check, lies below the green OPTMARGIN line in the median distance plots, indicating that region classification often samples points that match the adversarial example's incorrect class. OPTMARGIN and FGSM examples, however, are still less robust than

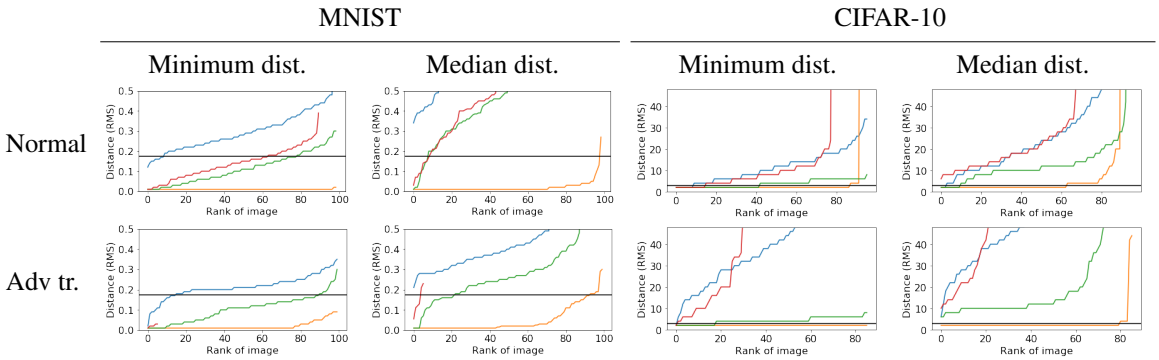

Figure 3.2: Minimum and median decision boundary distances across random directions, for a sample of images. **Blue**: Benign. **Red**: FGSM. **Green**: OPTMARGIN (ours). **Orange**: OPTBRITTLE. Each statistic is plotted in ascending order. A black line is drawn at the expected distance of images sampled by region classification.

benign examples to random noise.

Unfortunately, on MNIST, no simple threshold on any one of these statistics accurately separates benign examples (blue) from OPTMARGIN examples (green). At any candidate threshold (a horizontal line), there is either too much of the blue line below it (false positives) or too much of the green line above it (false negatives).

PGD adversarial training on the MNIST architecture results in decision boundaries closer to the benign examples, reducing the robustness to random perturbations. In CIFAR-10, however, the opposite is observed, with boundaries farther from benign examples in the PGD adversarially trained model. The effect of PGD adversarial training on the robustness of benign examples to random perturbations is not universally beneficial nor harmful.

Adjacent class purity

Another observation we made from plots like those in Figure 3.1 is that adversarial examples tend to have most directions lead to a boundary adjacent to a single class. We compute the *purity of the top k classes* around an input image as the largest cumulative fraction of random directions that encounter a boundary adjacent to one of k classes.

Figure 3.5 shows the purity of the top k classes averaged across different samples of images, for varying values of k. These purity scores are especially high for OPTBRITTLE adversarial examples compared to the benign examples. The difference is smaller in CIFAR-10, with the purity of benign examples being higher.

Region classification takes advantage of cases where the purity of the top 1 class is high, *and* the one class is the correct class, *and* random samples from the region are likely to be past those boundaries.

Adversarial examples generated by OPTMARGIN and FGSM are much harder to distinguish from benign examples in this metric.

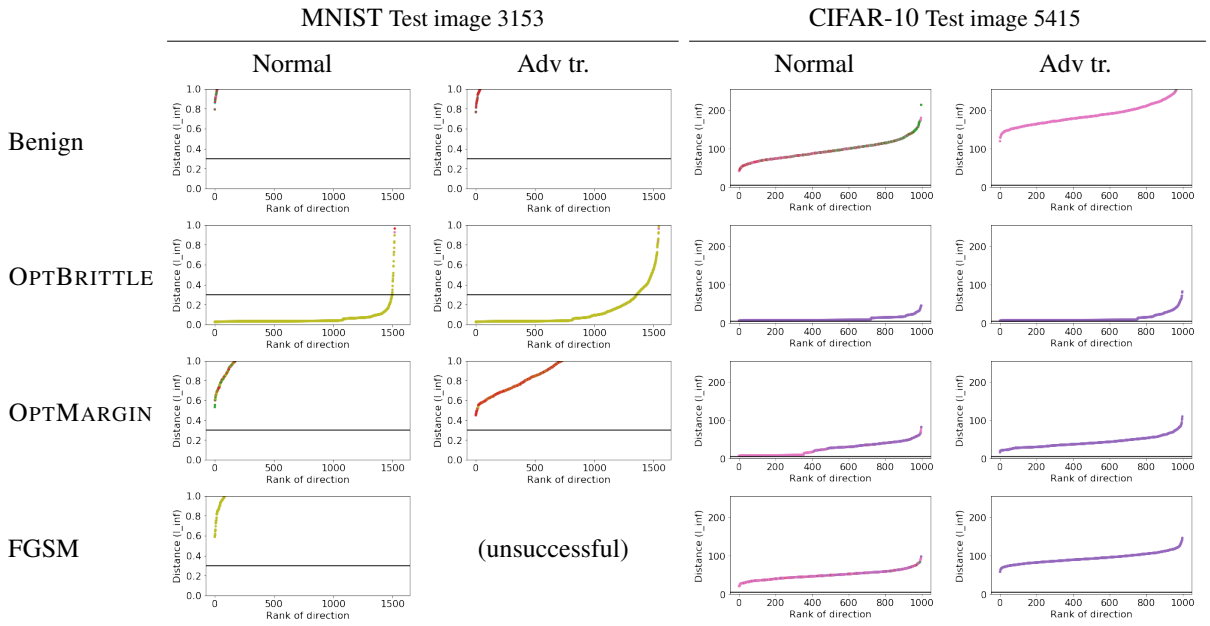

Figure 3.3: Equivalent of Figure 3.1, decision boundary distances from sample images, plotted in L_∞ distance. A black line is drawn at the radius of the region used in region classification.

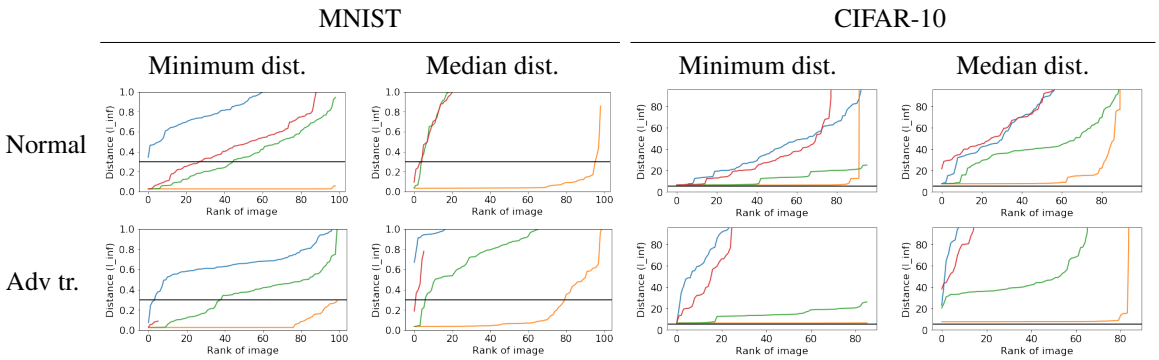

Figure 3.4: Equivalent of Figure 3.2, minimum and median decision boundary distances across random directions, plotted in L_∞ distance. **Blue**: Benign. **Red**: FGSM. **Green**: OptMargin (ours). **Orange**: OptBrittle. A black line is drawn at the radius of the region used in region classification.

Classification using surrounding decision boundaries

Cao and Gong's region classification defense is limited in its consideration of a hypercube region of a fixed radius, the same in all directions. We successfully bypassed this defense with our OptMargin attack, which created adversarial examples that were robust to small perturbations in many directions. However, the surrounding decision boundaries of these adversarial examples and benign examples are still different, in ways that sampling a hypercube would not reveal.

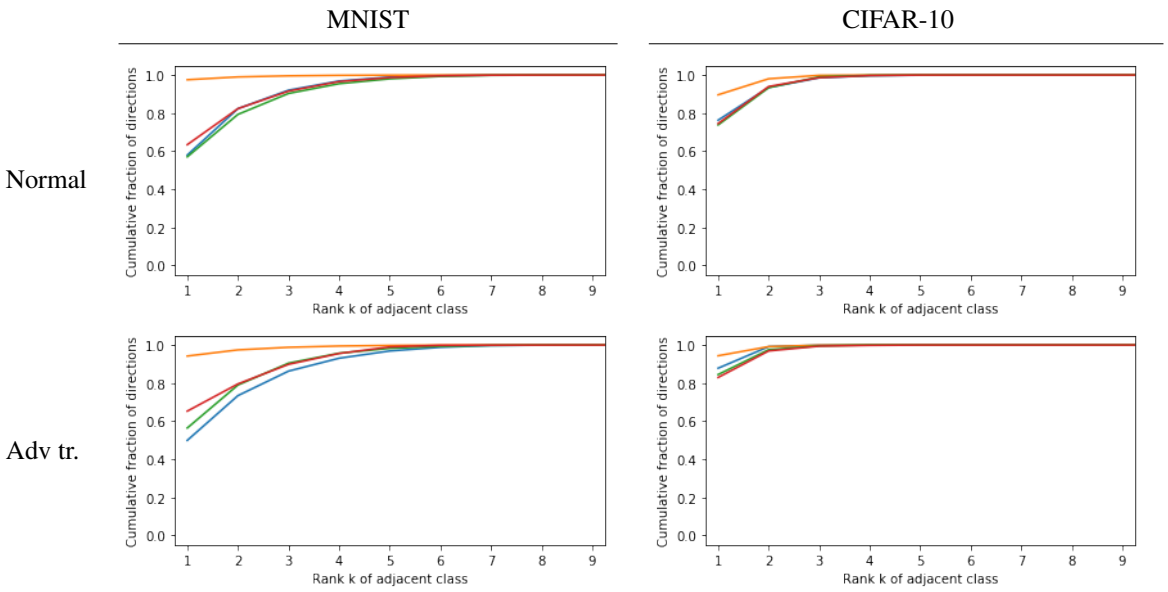

Figure 3.5: Average purity of adjacent classes around benign and adversarial examples. **Orange**: OPTBRITTLE. **Red**: FGSM. **Green**: OPTMARGIN (ours). **Blue**: Benign. Curves that are lower on the left indicate images surrounded by decision regions of multiple classes. Curves that near the top at rank 1 indicate images surrounded almost entirely by a single class.

In this section, we propose a more general system for utilizing the neighborhood of an input to determine whether the input is adversarial. Our design considers the distribution of distances to a decision boundary in a set of randomly chosen directions and the distribution of adjacent classes—much more information than Cao and Gong's approach.

Design

We ask the following question: Can information about the decision boundaries around an input be used to differentiate the adversarial examples generated using the current attack methods and benign examples? These adversarial examples are surrounded by distinctive boundaries on some models, such as the PGD adversarially trained CIFAR-10 model (seen in Figure 3.2). However, this is not the case for either MNIST model, where no simple threshold can accurately differentiate OPTMARGIN adversarial examples from benign examples. In order to support both models, we design a classifier that uses comprehensive boundary information from many random directions.

We construct a neural network to classify decision boundary information, which we show in Figure 3.6. The network processes the distribution of boundary distances by applying two 1-D convolutional layers to a sorted array of distances. Then, it flattens the result, appends the first three purity scores, and applies two fully connected layers, resulting in a binary classification. We use rectified linear units for activation in internal layers. During training, we use dropout [Hinton et al., 2012b] with probability 0.5 in internal layers.

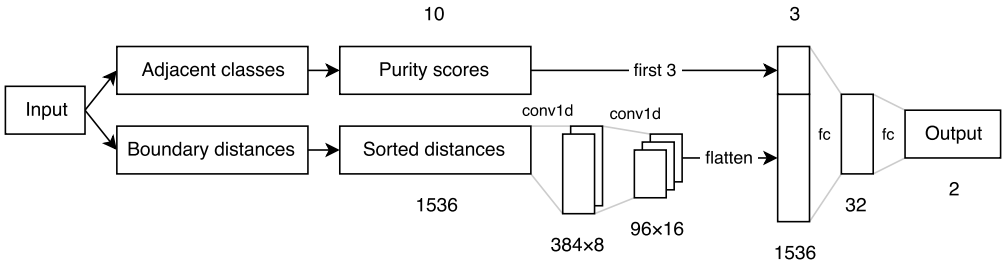

Figure 3.6: Architecture of our decision boundary classifier. Sizes are shown for our MNIST experiments.

Experimental Results

We train with an Adam optimizer with a batch size of 128 and a learning rate of 0.001. For MNIST, we train on 8,000 examples (each *example* here contains both a benign image and an adversarial image generated by a training attack) for 32 epochs, and we test on 2,000 other examples. For CIFAR-10, where it was more costly to examine the decision boundaries on the larger models, we train on 350 examples for 1,462 epochs, and we test on 100 other examples.

We filtered these sets only to train on correctly classified benign examples and successful adversarial examples.

To arrive at the final binary decision of whether an input is adversarial or not, we choose whichever class's (adversarial or benign) output confidence value is higher. Table 3.1 shows the false positive and false negative rates of the model.

FGSM creates fewer successful adversarial examples, especially for adversarially trained models. The examples from our experiments ($\epsilon = 0.3$ for MNIST and 8 for CIFAR-10) have higher distortion than the OPTMARGIN examples and are farther away from decision boundaries. We trained a classifier on successful FGSM adversarial examples for normal models (without adversarial training). Table 3.2 shows the accuracy of these classifiers. PGD adversarial training is effective enough that we did not have many successful adversarial examples to train the classifier.

This classifier achieves high accuracy on the attacks we study in this section. These results suggest that our current best attack, OPTMARGIN, does not accurately mimic the distribution of decision boundary distances and adjacent classes. On MNIST, the model with normal training had better accuracy, while the model with PGD adversarial training had better accuracy on CIFAR-10. We do not have a conclusive explanation for this, but we do note that these were the models with decision boundaries being farther from benign examples (Figure 3.2). It remains an open question, however, whether adversaries can adapt their attacks to generate examples with surrounding decision boundaries that more closely match benign data.

Performance

Assuming one already has a base model for classifying input data, the performance characteristics of this experiment are dominated by two parts: (i) collecting decision boundary information around

Training attack	False pos. Benign	False neg. OPTBRITTLE	False neg. OPTMARGIN	Accuracy Our approach	Accuracy Cao and Gong
MNIST, normal training					
OPTBRITTLE	1.0%	1.0%	74.1%	90.4%	10%
OPTMARGIN	**9.6%**	0.6%	7.2%		
MNIST, PGD adversarial training					
OPTBRITTLE	2.6%	2.0%	39.8%		
OPTMARGIN	10.3%	0.4%	14.5%		
CIFAR-10, normal training					
OPTBRITTLE	5.3%	3.2%	56.8%	96.4%	5%
OPTMARGIN	8.4%	7.4%	5.3%		
CIFAR-10, PGD adversarial training					
OPTBRITTLE	0.0%	2.4%	51.8%		
OPTMARGIN	**3.6%**	0.0%	1.2%		

Table 3.1: False positive and false negative rates for the decision boundary classifier, trained on examples from one attack and evaluated on examples generated by the same or a different attack. We consider the accuracy under the worst-case benign/adversarial data split (all-benign if the false positive rate is higher; all-adversarial if the false negative rate is higher), and we select the best choice of base model and training set. These best-of-worst-case numbers are shown in bold and compared with Cao & Gong's approach from Table 2.5.

Dataset	Normal training False pos.	Normal training False neg.
MNIST	7.0%	12.8%
CIFAR-10	20.0%	32.9%

Table 3.2: False positive and false negative rates for the decision boundary classifier, trained and evaluated on FGSM examples.

given inputs and (ii) training a model for classifying the decision boundary information.

Our iterative approach to part (i) is expensive, involving many forward invocations of the base model. In our slowest experiment, with benign images on the PGD adversarially trained wide ResNet34 CIFAR-10 model, it took around 70 seconds per image to compute decision boundary information for 1,000 directions on a GeForce GTX 1080. This time varies from image to image because our algorithm stops searching in a direction when it encounters a boundary. Collecting decision boundary information for OPTBRITTLE examples was much faster, for instance. Collecting information in fewer directions can save time, and should perform well as long as the samples adequately capture the distribution of distances and adjacent classes.

Part (ii) depends only on the number of directions, and the performance is independent of the base model's complexity. In our experiments, this training phase took about 1 minute for each

model and training set configuration.

Running the decision boundary classifier on the decision boundary information is fast compared to the training and boundary collection.

Conclusion

We analyze the neighborhood of adversarial examples from our OPTMARGIN attack by looking at the decision boundaries around them, as well as the boundaries around benign examples and less robust adversarial examples. Our experiments showed that with adversarial training, while it may make models more robust to existing attacks, can decrease the distances from benign examples to decision boundaries. We find that the comprehensive information about surrounding decision boundaries reveals there are still differences between our robust adversarial examples and benign examples. It remains to be seen how attackers might generate adversarial examples that better mimic benign examples' surrounding decision boundaries.

3.2 New attack methods

The attacks we presented so far rely on a few common techniques, sharing in common that they use gradient descent algorithms to alter pixel values. While these techniques are effective, focusing on them exclusively would limit our understanding of the full attack space. In this section, we investigate three new attack methods that represent major departures from this paradigm. In collaboration with Bhagoji et al., we evaluate (i) an attack that uses finite differences to generate perturbations, and in collaboration with Xiao et al., we evaluate (ii) AdvGAN, an attack that uses a generative adversarial network (GAN) to synthesize perturbations and (iii) stAdv, an attack that moves pixels spatially rather than altering their values. We perform a comparative evaluation of these attacks with previous attacks on defended models.

Defenses

We focus on defenses that use adversarial training. Recently, this category of defenses been shown to hold up to an especially broad range of adaptive attacks [Mądry et al., 2017]. Across the experiments in this section, we test on up to three variants of each model architecture for each dataset, using different adversarial training defenses.

1. FGSM adversarial training [Goodfellow et al., 2015], which trains models on FGSM adversarial examples
2. Ensemble adversarial training [Tramèr et al., 2017a], which trains models on a combination of benign examples and FGSM adversarial examples taken from other models
3. PGD adversarial training [Mądry et al., 2017], which trains models on PGD adversarial examples

For the adversarial examples used in training, we limit the perturbation to an L_∞ norm of 0.3 for MNIST and 8 for CIFAR-10.

Replacing gradients with finite differences

Bhagoji et al. [2018] propose a collection of black-box attacks that use finite differences, for use in scenarios where the attacker can query the model. They first describe attacks based on FGSM—in Bhagoji et al.'s attacks, the gradient of the loss function is replaced with a finite difference. This way, the attacker does not need to know the model's weights. Instead, they only need to be able to observe the model's output confidence on provided inputs. The authors show how to compute the cross-entropy loss used in FGSM from a model's confidence outputs, as well as Carlini and Wagner's logit based loss. Estimating one of these gradients using symmetric finite differences requires twice as many queries as input dimensions. Bhagoji et al. go on to demonstrate ways to approximate the gradient with fewer queries, using methods that group dimensions together. With these approximate gradients, they propose Single-step attacks and Iterative attacks: Single-step attacks take one fixed-sized step according to the approximate gradient, similar to FGSM; Iterative attacks take a sequence of smaller fixed sized steps and re-approximate the gradient at each step. They show that their attacks approach white-box level attack success rate, outperforming transfer based black-box attacks especially in generating targeted adversarial examples.

In this section, we evaluate the adversarial training defenses on Bhagoji et al.'s attacks.

Experimental setup

Data. For MNIST, Single-step attacks are carried out on the test set of 10,000 samples, while Iterative attacks are carried out on 1,000 randomly chosen samples from the test set. For the CIFAR-10, we choose 1,000 random samples from the test set for both Single-step and Iterative attacks. In our evaluation of targeted attacks, we choose target y^* for each sample uniformly at random from the set of classification outputs, except the true class y of that sample.

Models. On MNIST, we trained two different CNNs, denoted Model A and Model B, with the architectures taken from Tramèr et al. [2017a]. Model A has 2 convolutional layers followed by a fully connected layer while Model B has only 3 convolutional layers. Both models have an accuracy of 99.2% on the test set. For CIFAR-10, we use ResNet32 and wide ResNet34. In the ensemble adversarial training for the MNIST models, we include adversarial examples from two additional CNNs also from Tramèr et al. [2017a]. We denote adversarially trained models with subscripts: adv-ϵ for FGSM adversarial training, adv-ens-ϵ for ensemble adversarial training, and adv-iter-ϵ for PGD adversarial training.

Results

In this section, we focus on *untargeted* attacks on adversarially trained models. We find that Single-step Gradient Estimation attacks match the success rate of their white-box counterparts even with

query reduction.

Adversarially trained models are not robust to Gradient Estimation attacks. Our experiments show that Iterative black-box attacks continue to work well even against adversarially trained networks as seen in Table 3.3. For example, the Iterative Gradient Estimation attack using Finite Differences with a logit loss (IFD-logit) achieves an attack success rate of 76.5% against Model $A_{\text{adv-0.3}}$ and 96.4% against Model $A_{\text{adv-ens-0.3}}$. This attack works well for CIFAR-10 models as well, achieving attack success rates of 100% against both ResNet32 $_{\text{adv-8}}$ and ResNet32 $_{\text{adv-ens-8}}$. This reduces slightly to 98% and 91% respectively when query reduction using random grouping is used. For both datasets, IFD-logit matches white-box attack performance. For MNIST, using PCA for query reduction to just 8000 queries per sample, a 51% attack success rate is achieved for both Model $A_{\text{adv-0.3}}$ and Model $A_{\text{adv-ens-0.3}}$.

Dataset	**White-box**		**Gradient Estimation, FD**		**Gradient Estimation, Query Reduction**			
MNIST	Single-step	Iterative	Single-step [1568]	Iterative [62720]	Single-step [$\sim$ 200]		Iterative [8000]	
Models	FGS (logit)	IFGS (logit)	FD-logit	IFD-logit	PCA-100	RG-8	PCA-100	RG-8
$A_{\text{adv-0.3}}$	2.9 (6.0)	78.5 (3.1)	2.8 (5.9)	76.5 (3.1)	4.1 (5.8)	2.0 (5.3)	50.7 (4.2)	27.5 (2.4)
$A_{\text{adv-ens-0.3}}$	6.2 (6.2)	96.2 (2.7)	6.2 (6.3)	**96.4 (2.7)**	5.4 (6.2)	3.7 (6.4)	51.0 (3.9)	32.0 (2.1)
$A_{\text{adv-iter-0.3}}$	7.3 (7.5)	11.0 (3.6)	7.5 (7.2)	11.6 (3.5)	3.5 (4.0)	1.6 (4.2)	9.0 (2.8)	3.0 (1.4)
CIFAR-10	Single-step	Iterative	Single-step [6144]	Iterative [61440]	Single-step [$\sim$ 800]		Iterative [$\sim$ 8000]	
Models	FGS (logit)	IFGS (logit)	FD-logit	IFD-logit	PCA-400	RG-8	PCA-400	RG-8
ResNet32 $_{\text{adv-8}}$	8.9 (438.8)	100.0 (73.7)	8.5 (401.9)	**100.0 (73.8)**	8.0 (402.1)	7.7 (401.8)	97.0 (151.3)	98.0 (92.9)
ResNet32 $_{\text{adv-ens-8}}$	13.3 (437.9)	100.0 (85.3)	12.2 (399.8)	**100.0 (85.2)**	15.4 (396.1)	13.8 (395.9)	82.7 (178.7)	90.8 (106.6)
ResNet32 $_{\text{adv-iter-8}}$	50.4 (346.6)	57.3 (252.4)	47.5 (331.1)	54.6 (196.3)	47.5 (344.1)	38.4 (341.4)	51.3 (256.6)	42.4 (153.3)

Table 3.3: **Untargeted black-box attacks** for models with **adversarial training**: attack success rates and average L_2-squared distortion in parentheses. For Gradient Estimation attacks, the number of queries is shown in brackets. **Top**: MNIST, $\epsilon = 0.3$. **Bottom**: CIFAR-10, $\epsilon = 8$.

Model $A_{\text{adv-iter-0.3}}$ is robust even against iterative attacks, with the highest black-box attack success rate achieved being 11.6%—marginally higher than the white-box attack success rate. On CIFAR-10, the iteratively trained model has poor performance on both benign and adversarial examples. The IFD-logit attack achieves an untargeted attack success rate of 55% on this model, which is lower than on the other adversarially trained models, but still significant. This is in line with Mądry et al.'s observation [2017] that iterative adversarial training needs models with large capacity for it to be effective. This highlights a limitation of this defense, since it is not clear what model capacity is needed, and the models we use already have a large number of parameters.

Perturbations from generative adversarial networks

Xiao et al. [2018a] propose AdvGAN, an attack that uses a generative adversarial network to generate the perturbation that would be added to an input image. They train a generator $\mathcal{G}$ that takes a benign example x that outputs a perturbation $\mathcal{G}(x)$ and a discriminator $\mathcal{D}$ that tries to determine whether x or the perturbed image $x^* = x + \mathcal{G}(x)$ is the original example. The discriminator makes the generator favor perturbations that keep the perturbed image looking realistic, and they add two

additional terms to the GAN loss specific to the problem of generating adversarial examples: (i) a cross-entropy loss on the target model's classification of the perturbed image, in order to favor misclassification; and (ii) a hinge loss on the distortion, in order to ensure small perturbations. Xiao et al. show that the perturbed images are adversarial and are perceptually realistic. Given the fact that AdvGAN strives to generate adversarial instances from the underlying true data distribution, it can essentially produce more photo-realistic adversarial perturbations compared with other attack strategies. Thus, AdvGAN could have a higher chance to produce adversarial examples that are resilient under different defense methods. In this section, we quantitatively evaluate this property for AdvGAN on CIFAR-10.

Threat model. As shown in the literature, most of the current defense strategies are not robust when attacking against them [Carlini and Wagner, 2017a]. Here we consider a weaker threat model, where the adversary is not aware of the defenses and directly tries to attack the original learning model, which is also the first threat model analyzed in Carlini and Wagner [2017a]. In this case, if an adversary can transfer the attack to the adversarially trained model, it implies the robustness of the attack strategy. Under this setting, we first apply different attack methods to generate adversarial examples based on the original model without being aware of any defense. Then we apply different defenses to directly defend against these adversarial instances.

Semi-whitebox attack. Xiao et al. describe AdvGAN operating in a *semi-whitebox* setting, where the adversary at first can access the model architecture and parameters (in these experiments, the original non-adversarially trained model), during which they propose to train the GAN; later the adversary must generate adversarial examples *without* access to the model's information. We first consider this attack setting, in comparison to white-box methods which continuously have access to the model's information (again, the non-adversarially trained model). We evaluate the effectiveness of these transferred attacks against the adversarially trained models. We compute the attack success rate under a fixed distortion budget of an L_∞-norm of 8 ($[0, 255]$ scale). In Table 3.4, we show that the attack success rate of adversarial examples generated by AdvGAN on different models is higher than those of the fast gradient sign method (FGSM) and an optimization method (Opt.) [Carlini and Wagner, 2017c].

Opt. in our experiments uses the low-confidence L_∞ attack. Carlini and Wagner note that, in a slightly more adaptive approach, the attacker can use a higher confidence parameter to improve transfer attack success rates. An attacker may be able to similarly adjust AdvGAN's training favor high confidence adversarial examples for better transferability as well.

Black-box attack. Xiao et al. provide a black-box adaptation of their attack, based on distilling a substitute model from the target model's outputs on chosen inputs [Hinton et al., 2015]. They describe a dynamic distillation procedure, where the substitute model is trained along with the generator and discriminator, using the target model's outputs on images perturbed by the generator's output. For AdvGAN, we use ResNet32 as the black-box model and train a distilled model on a disjoint set of training data. We report the attack success rate in Table 3.5. For the black-box at-

Model	Defense	FGSM	Opt.	AdvGAN
ResNet32	Adv.	13.10%	11.90%	**16.03%**
	Ensemble.	10.00%	10.30%	**14.32%**
	Iter. Adv	22.80%	21.40%	**29.47%**
Wide ResNet34	Adv.	5.04%	7.61%	**14.26%**
	Ensemble	4.65%	8.43%	**13.94%**
	Iter. Adv.	14.90%	13.90%	**20.75%**

Table 3.4: Attack success rate of transferred adversarial examples generated by AdvGAN in semi-whitebox setting, and other transferred attacks under defenses on CIFAR-10.

tack comparison purpose, transferability based attack is applied for FGSM and optimization based methods (Opt.), using examples generated for wide ResNet34. Again, we report attack success rate at a fixed distortion budget of an L_∞-norm of 8. We can see that the adversarial examples generated by the black-box AdvGAN consistently achieve higher attack success rate compared with other attack methods.

	Transfer		
Defense	FGSM	Opt.	AdvGAN
Adv.	13.58%	10.80%	**15.96%**
Ensemble	10.49%	9.60%	**12.47%**
Iter. Adv.	22.96%	21.70%	**24.28%**

Table 3.5: Attack success rate of transferred adversarial examples generated by different black-box adversarial strategies under defenses on CIFAR-10.

Spatial perturbations

Xiao et al. [2018b] demonstrate an attack that generates adversarial examples by spatially transforming the input image. In their design, they use a displacement map that specifies, for each pixel in the output, where in the input image to sample for the color. With a differentiable sampling operation, where a weighted average of four surrounding pixels is used for floating point coordinates, they adapt existing gradient based attack techniques to find a displacement map that results in a misclassification. They use additional loss terms to favor small, locally smooth displacements. Xiao et al. show that the perturbed images are adversarial and difficult for humans to distinguish from the original images. We experiment with the same static adversary threat model we used with AdvGAN, where the attacker tries to transfer adversarial examples generated for the non-adversarially trained model.

Model	Def.	FGSM	Opt.	stAdv
ResNet32	Adv.	13.10%	11.90%	**43.36%**
	Ens.	10.00%	10.30%	**36.89%**
	PGD	22.80%	21.40%	**49.19%**
Wide ResNet34	Adv.	5.04%	7.61%	**31.66%**
	Ens.	4.65%	8.43%	**29.56%**
	PGD	14.90%	13.90%	**31.60%**

Table 3.6: Attack success rates of adversarial examples generated by stAdv against ResNet and wide ResNet on CIFAR-10, under defenses.

We compare with the same well known attacks as we used in the AdvGAN experiments, FGSM and Opt. under an L_∞ distortion budget of 8. The distortion of adversarial examples generated by stAdv isn't well measured by the L_∞-norm because displacing a high contrast edge makes a large difference in the values of the displaced pixels. However, Xiao et al. [2018b] confirmed in a human perceptual study that stAdv's adversarial examples are rated equally realistic to benign images. The results are shown in Table 3.6. We observe that the three defense strategies can achieve high performance (less than 10% attack success rate) against FGSM and Opt. attacks.

These defense methods only achieve low defense performance on stAdv, which improves the attack success rate to more than 29% among all defense strategies. These results indicate that new types of adversarial strategies, such as Xiao et al.'s spatial transformation based attack, may open new directions for developing better defense systems.

Mean blur defense. We also test the adversarial examples against the 3×3 average pooling restoration mechanism [Li and Li, 2016]. Table 3.7 shows the classification accuracy of recovered images after performing 3×3 average filter on different models (without adversarial training). We find that the simple 3×3 average pooling restoration mechanism can recover the original class from FGSM examples and improve the classification accuracy up to around 70% under a static adversary. Carlini and Wagner have also shown that such mean blur defense strategy can defend against adversarial examples generated by their attack and improve the model accuracy to around 80% [2017a]. From Table 3.7, we can see that the mean blur defense method can only improve the model accuracy to around 50% on stAdv examples, which means adversarial examples generated by stAdv are more robust compared to other attacks.

Filter	ResNet32	Wide ResNet34
3×3 Average	45.12%	50.12%

Table 3.7: Performance of blurring on AdvGAN adversarial examples on CIFAR-10: model accuracy on recovered images.

We also perform a perfect knowledge adaptive attack against the mean blur defense following the same attack strategy suggested in Carlini and Wagner [2017a], where we add the 3×3 average pooling layer into the original network and apply stAdv to attack the new network again. We observe that the success rate of an adaptive attack is nearly 100%, which is consistent with Carlini and Wagner's findings [2017a] with their attack.

Conclusion

We study the effectiveness of promising adversarial training defenses under new attacks that have important differences from existing additive, gradient based approaches. Our results consistently showed that the newer attacks outperform common gradient based attacks. These attacks are not tailored for bypassing any specific defense. The results from these experiments suggest that previous work on adversarial examples defenses are not robust to a wide range of possible attacks.

Chapter 4

Summary and conclusion

We explore techniques for evaluating defenses against adversarial examples under an adaptive adversary, focusing on cases where a mechanism complicates the formulation of a loss function for adapting existing attacks. We demonstrate these techniques on a collection of defenses, including representative examples of ensemble detectors and a non-deterministic recovery defense. Our experiments with our adaptive attacks show four examples of defenses could be bypassed effectively.

Next, we study inputs that are close to benign examples and are misclassified, other than adversarial examples generated by well known methods. We perform a brute-force analysis of the decision boundaries in a large sample of directions around different kinds of examples, and we experiment with three new attacks that have important differences from previous gradient based attack methods. We observed that a promising category of defense methods, adversarial training, performs worse on these new attacks and can reduce robustness to random noise.

The results from our experiments on these attacks and defenses suggest that our current best defenses, particularly in image classification tasks, against adversarial examples are overly specialized for the well studied attack methods. We demonstrated that defenses which can prevent a single attack, or prevent a common technique used in attacks, or prevent an entire domain of possible perturbations, all can be weakened and bypassed by novel, practical attacks.

Developing effective defenses against adversarial examples is an important step towards being able to deploy deep learning systems in more real-world use cases. We hope this dissertation sheds light for future work in more general defenses.

Bibliography

Mahdieh Abbasi and Christian Gagné. Robustness to adversarial examples through an ensemble of specialists. *5th International Conference on Learning Representations (ICLR) Workshop*, 2017.

Anish Athalye and Nicholas Carlini. On the robustness of the CVPR 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018.

Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *European Conference on Computer Vision*, pages 158–174. Springer, Cham, 2018.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SyZI0GWCZ`.

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

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *ACM Workshop on Artificial Intelligence and Security (AISEC)*, 2017a.

Nicholas Carlini and David Wagner. MagNet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv preprint arXiv:1711.08478*, 2017b.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017c.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. *arXiv preprint arXiv:1708.03999*, 2017.

Clarifai. Clarifai | image & video recognition API. `https://clarifai.com`. Accessed: 2017-08-22.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *3rd International Conference on Learning Representations (ICLR)*, 2015.

Google Vision API. Vision API - image content analysis | google cloud platform. `https://cloud.google.com/vision/`. Accessed: 2017-08-22.

Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, Vancouver, BC, 2017. USENIX Association. URL `https://www.usenix.org/conference/woot17/workshop-program/presentation/he`.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012a.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012b.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Yann LeCun. The MNIST database of handwritten digits. 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. *arXiv preprint arXiv:1612.07767*, 2016.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *5th International Conference on Learning Representations (ICLR)*, 2017a.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *5th International Conference on Learning Representations (ICLR)*, 2017b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *5th International Conference on Learning Representations (ICLR)*, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582. IEEE, 2016.

Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv:1706.06083 [cs, stat]*, June 2017.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436. IEEE, 2015.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016a.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016b.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 582–597. IEEE, 2016c.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014a.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014b.

Pedro Tabacof and Eduardo Valle. Exploring the space of adversarial images. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 426–433. IEEE, 2016.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017a.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017b.

Watson Visual Recognition. Watson visual recognition. `https://www.ibm.com/watson/services/visual-recognition/`. Accessed: 2017-10-27.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018a.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *International Conference on Learning Representations (ICLR)*, 2018b.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017a.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing mitigates and detects Carlini/Wagner adversarial examples. *arXiv preprint arXiv:1705.10686*, 2017b.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H1BLjgZCb`.