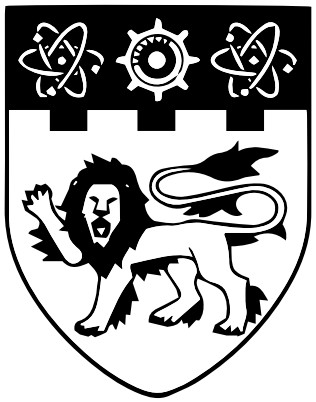


NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

Trustworthy And Certified Robustness For Deep learning

Xia Song

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2022

Trustworthy And Certified Robustness For Deep learning

XIA SONG

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

**A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN SIGNAL PROCESSING**

2022

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

. . . . . .
Date

. . . . . .
Your Name

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice

....

Date Supervisor's Name

Authorship Attribution Statement

This thesis does not contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

....

Date Your Name

Table of Contents

Abstract

Currently deep learning becomes the main technology for solving most computer vision and pattern recognition tasks, such as classification, objection detection and recognition. However, researchers found out that deep learning based systems were vulnerable to adversarial examples, which by adding small and human imperceptible corruptions, could mislead AI system with a high successful rate. Randomized smoothing is a promising technology to enhance the trustworthy and provide the certified robustness for AI system, which gives the worst-case decision boundary towards all adversarial examples by statistic estimation. However, the robustness optimization object in this technology was non-differentiable, due to Monte Carlo sampling and the useful information from inputs data was corrupted, due to high Gaussian variance. To solve this problem, this dissertation gives in-depth analysis towards current robustness estimation optimization methods and proposed a new generalized consistency optimization, consisting of a looser accuracy item which is more flexible to train, and a more accurate robustness item. Meanwhile, this dissertation first utilized the linear decomposition method to select the discriminant information from the high variance-level noise corruption. Experiment results show that the proposed the generalized optimization with linear decomposition randomized smoothing gained are more effective than state-of-the-art methods.

Keywords: Deep learning, Certified Defense, Generalized optimization, Adversarial examples.

Acknowledgements

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its IAF-ICP Programme ICP1900093 and the Schaeffler Hub for Advanced Research at NTU.

Acronyms

NN	Neural Network
ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural Network
RS	Randomized Smoothing
PCA	Principal Component Analysis
ZOO	Zero-Order Optimization
FGSM	Fast Gradient Sign Method
NES	Natural Evolution Strategy
DFO	Derivative Free Optimization
PRN	Perturbation Rectifying Network
GAN	Generative Adversarial Network

Symbols

Π	An Pi Symbol
σ	The standard deviation
Φ	Cumulative Density Function
δ	Adversarial Perturbation
f	Base Classifier
g	Smoothed Classifier
P	The Probability
N	Normal Distribution

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Modern deep neural network has made super-human performance in a wide range of applications such as computer vision and nature language processing. However, several researches [2, 8] have indicated that those well-trained models are extremely vulnerable to adversarial attack, which, by adding small and human imperceptible corruption, could seriously mislead neural network's prediction results. This greatly restricts their application towards some safety-critical domains.

In response, many researchers started to find the countermeasures to resist adversarial attack thus improving deep learning's robustness. Recently, two mainstreams of defense methods are well developed: the heuristic based defense and certified based defense. The heuristic based defense such as adversarial training, mainly enhances model's robustness by repeatedly training models with new generated adversarial examples [2, 9]. Though effective towards existing adversarial attacks, those methods are not fully reliable and trustworthy: most well performed heuristic defense methods were later broken by unseen and stronger adversaries [10, 11]. Meanwhile, the iterative update of new adversaries usually requires huge computation resource.

Alternatively, a growing party of researches started to pay attention to certified defense methods, by which the classifier is guarantee for a constant prediction result for any input within certain set around it [12,13],. Randomized smoothing is a promising certified defense method which can be easily implemented for large-scale recognition task such as ImageNet [14] and also provide the tightest theoretical boundary under Gaussian noise [15]. With Gaussian corruption, randomized smoothing turns any base classifier to a smoothed version by changing its prediction on one point to a certain Gaussian distribution. In other word, the base classifier computed the prediction over a gaussian distribution with mean as the clean input by Monte Carlo sampling and the smoothed classifier returns the class label with highest average probability.

However, there are two deficiencies in randomized smoothing. First, under high levels of noise, some classes discriminant information was heavily perturbed thus decreasing the final prediction accuracy. Second, in [15], due to utilizing of 0-1 hard mapping and Monte Carlo sampling, the robustness item was non differentiable towards the model's output, thus making maximize robustness inapplicable for directly gradient descent methods. Recently, many researchers proposed different methods for training a base classifier with good robustness, e.g., Cohen et al. [15] enhance the training process with Gaussian augmentation. Later, [16] proposed a more accurate robustness optimization and further enhanced model with adversarial training. [17] replaced the hard 0-1 mapping with a soft mapping function, thus optimizing an approximate robustness by a directly gradient descent method. Most recently, [18, 19] proposed a regularization item and ensemble methods to further enhance the performance.

1.2 Major contributions of the Dissertation

1. This dissertation first proposes a generalized consistency optimization method which combines with a looser accuracy item and more accurate robustness

regularization.

2. This dissertation first analyzes the perturbation problems when the variance of noise greatly surpasses the variance of discriminant. The principal component analysis is utilized to compute the variance of each dimension and further select the useful information. The detailed experiments are conducted to prove the effectiveness of this transformation.

3. This dissertation in-depth analyzes the uneven gradient problem in the approximate robustness optimization proposed by [17] and add a gradient threshold, which greatly accelerate its optimization process.

4. The extensive experiment results shows that our proposed generalized consistency optimization with linear decomposition randomized smoothing outperforms current state-of-the-art methods.

1.3 Organisation of the Dissertation

1. **Introduction**

   This chapter briefly describes the vulnerability of current deep learning models and the merits of certified defense methods over heuristic defense methods. Furthermore, two deficiencies of current randomized smoothing are indicated and contributions of this dissertation are summarized.

2. **Related work and preliminaries**

   This chapter briefly explains the architecture of CNNs and its gradient backward mechanism Then, two mainstreams of attack methods: white-box and black-box attacks and two representative defense methods: adversarial training and randomized smoothing are introduced, along with the related work on corresponding areas.

3. **Certified defense with randomized smoothing**

   This chapter first introduces the mathematical support of randomized smoothing and then gives in-depth mathematical analysis on problem of robustness gradient. Three proposed robustness optimization methods, Gaussian augmentation, Maximize certified robustness [17], and consistency regularization [18], are analyzed and their potential drawbacks are indicated. process.

4. **Generalized optimization and linear decomposition**

   This chapter first proposes the generalized optimization, consisting of a loose accuracy item and an accurate robustness item, and gives detailed explanation on its advantages and work mechanism. Later, the class discriminant information loss problem in randomized smoothing is analyzed and the method of selecting useful information from high variance noise by linear decomposition is proposed.

5. **Experiment results**

   This chapter first briefly describes the training environment, evaluation met-

rics and dataset. And then, comprehensive experiments results are given. The comparison of our proposed methods with the state-of-the-art optimization methods are given, which shows our proposed method outperforms previous methods on accuracy, robustness and training efficiency.

6. **Conclusion and future work**

   This chapter gives the conclusion of our work on certified robustness area, which summarize the motivations, contribution and future work.

Chapter 2

Related Work and Preliminaries

2.1 Related Work

Recently, many works have proposed different methods to enhance the classifier's robustness for adversarial examples. Generally, those methods can be divided into empirical defenses, which seem robust to known attacks but along with potential weakness, and certified defense which can guarantee a provable robustness to certain kinds of adversarial perturbations.

2.1.1 Empirical defense

Adversarial training [2,8,9],is to date the most powerful defense method in empirical defense, which repeatedly generates new adversarial examples during the iteration of classifier and add them to the training set. In this kind of training, the classifier is trained to minimize the worst-case loss over a neighborhood around the input. Though, this defense seems powerful, it cannot guarantee whether the prediction by the empirical robust classifier is truly robust to adversarial examples. In fact, many well performed adversarial training methods are later broken by a stronger or more delicately designed adversaries [20]. To end this arms race between the defenders and attackers, many researchers start

to find the defense with formal robustness guarantee

2.1.2 Certified defense

In certified defense, any input is guaranteed with a constant prediction result with a certain neighborhood, often a l_2 and l_∞ ball. Those methods typically are either exact (”complete”) or conservative (”sound but incomplete”). Exact methods will report whether there exists or not an adversary near the data point to mislead the classifier, usually by mixed integer linear programming [21,22] or Satisfiability Modulo Theories [23, 24] (refers to sodm). However, those methods take large amount of computational resource thus being hard to transfer to large-scale neural network (more than 100,000 activations) [22]. Conservative methods will also certify there is or not an adversary existing, but they might decline to make a certification when the data x have a safe neighbor. The advantage is that those methods are more flexible and consume less computing resource, thus making them more efficient to build certified defense [12, 13, 25]. However, those methods either require specific network architecture (e.g. ReLU activation or layered feedforward structure) or need extensive customization for new architecture. Furthermore, none of those methods are shown to be feasible to provide defense for the modern machine learning tasks such as ImageNet classification.

2.1.3 Randomized Smoothing

In randomized smoothing, it does not directly provide certified robustness to the original classifier. Instead, it generates a new smoothed classifier by noise corruption and provide certified robustness for smoothed classifier. Lecuyer et al. [26] first prove that by utilizing inequalities from the differential privacy lit-

erature, it can provide a strong l_2 and l_1 robustness guarantee for smoothed classifier. Later, Li et al. [27], give a stronger robustness guarantee for l_2 norm adversary based on information theory. In[3], Cohen et al. first provides a tight robustness guarantee for l_2 norm adversary and greatly enlarges the average certified robustness radius. Based on this,Salam et al. [16] uses adversarial training and Zhai et al. [17] uses scalable robustness training to maximize the certified radius and achieve the new state-of-the art result.

2.2 Preliminaries

2.2.1 Convolutional neural networks

Convolutional Neural Networks (CNNs) are the generalized version of multilayer perceptron. Based on its large-scale trainable parameters and shared-weight architecture of convolution kernels, it achieves a great performance improvement compared with traditional machine learning methods in some complex recognition tasks. Currently, CNNs are the most widely used deep learning models in solving complex scenarios and multi-modal problems, such as image and video recognition, medical image analysis, brain computer interface [28] and so on.

The basic structures in CNNs including the following four blocks: convolutional layer, activation layer, pooling layer and fully connected layer. Most modern researchers design a better performed CNN model based on different combination of above four base blocks.

Convolutional layer: The convolutional layer consists of several learnable kernels or filters. Those kernels are usually in small size such as 3*3 or 5*5. When calculating, each input block which has the same size of kernel is taken

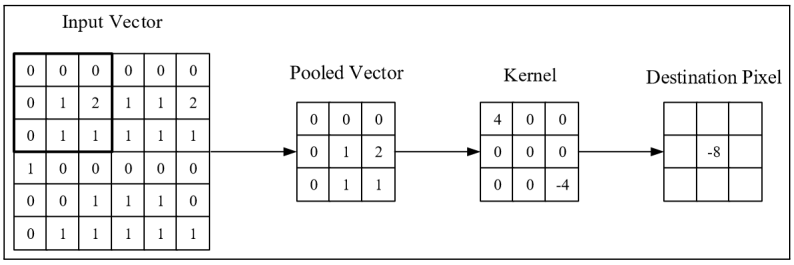


Figure 2.1: The calculation process of convolutional layer [1]

out to do the convolutional operation with the trained kernel, shown in Figure 2.1. Then by a shifting window, the kernel filter across the whole inputs and produce a 2D activation map.

Activation layer: Because convolution calculation is linear operation, CNNs introduce activation layer which can help achieve non-linear mapping. Currently, the most widely used activation function is ReLu activation which is defined as:

$$f(x) = \max(0, x) \tag{2.1}$$

The advantages of this activation over Sigmoid or tanh are:

1. More efficient gradient descent and backward, which avoid gradient exploration and disappear.
2. Simple calculation process, which avoiding complex function such as exponential.

Pooling layer: The pooling operation usually after convolutional layer and activation layer. The mechanism of this operation is to down-sampling the output feature map and most widely used pooling method is Max Pooling shown in Figure 2.2 . The function of pooling can be summarized as: keeping the main feature and reduce the computational complexity, thus further avoiding overfitting.

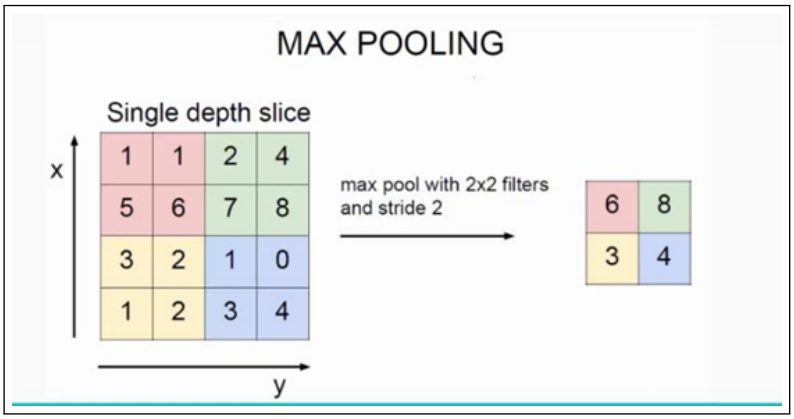


Figure 2.2: The calculation process of pooling layer.

Fully connected layer: The fully connect layer actual is a linear mapping function, shown in Equation 2.2, which maps the learned feature to the data label space. Usually, there will be a SoftMax regression to help do the final classification.

$$f(x) = W \cdot x + b \tag{2.2}$$

2.2.2 Adversarial attack

Though deep learning achieve great performance and are well generalized in many computer vision areas, research has shown that AI based systems are extremely vulnerable to adversarial attack or adversarial examples. Those adversarial, though being small and even imperceptible to human eyes, could mislead the AI system with a very high successful rate. In Figure 2.3, the adversarial examples successfully misleading a deep learning classifier to misclassify pandas as gibbons with high confidence after adding small adversarial perturbation to the original input. Current, the generation of adversarial attack methods can be roughly divided into two parts: white-box attack and black-box attack, which will be detailed introduced in the following parts.

White-box attack: The first white-box attack method was proposed by Szegedy

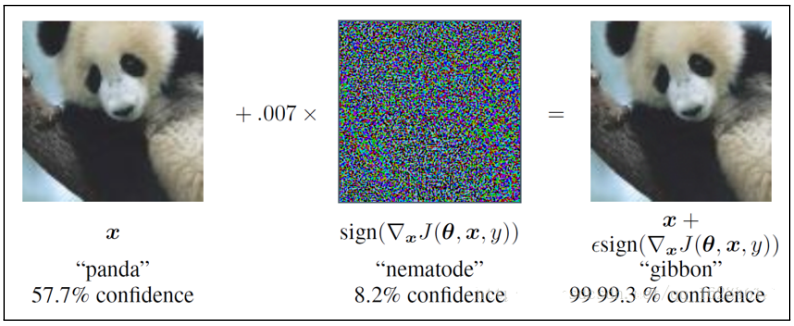


Figure 2.3: An example of adversarial attack [2].

[2], named Box-constrained L-BFGS attack. In his work, the adversarial attack was generated by solving Equation 2.3, where I_c represents the clean input, ρ is the adversarial perturbation, L is the label of input I_c and C is the deep learning classifier. However, the calculation complexity of this equation is much too high. In his paper, this target was transformed as finding the minimal perturbation to maximize the loss function, which turns the problem a Convex Optimization problem.

$$\min_{\rho} \|\rho\| \, s.t. C(I_c + \rho) \neq L(I_c + \rho) \in [0,1]^m, I_c \in R^m \tag{2.3}$$

The beginning explanation of the existing of adversarial examples is in the high dimensional space, the semantic information is not represented by one unit but the whole space expression. Meanwhile, the mapping of deep neural networks from input to output is almost discontinuous. Later, in [8], GoodFellow et al. propose a more efficient adversarial generation method named Fast Gradient Sign Method (FGSM), which can greatly accelerate the adversarial examples generation speed. In 2017, Nicholas Carlini and David Wagner [3] proposed a customized white-box attack method: adversarial examples are generated separately for each neural network, and the precondition is to know all information of the neural network to be attacked, including the input and output of the network, loss function, etc. This attack is hard to be noticed by human eyes by restricting the l_2,l_0 and l_∞ norm of adversarial perturbation, shown in Figure 2.4 . Experiment result showed that all those three norm-based attacks were pretty effective, which achieve 100% successfully attack rate in the best

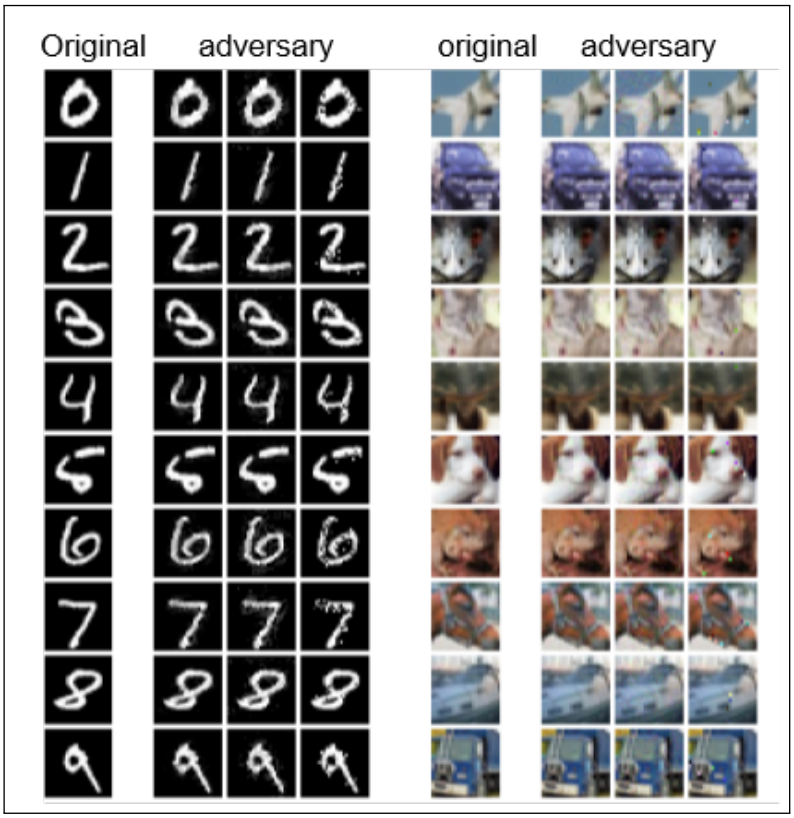


Figure 2.4: The adversarial examples generated by CW attack [3].

neural network model. Meanwhile, the generated adversarial examples can be transferred to attack those unknown neural networks, thus achieving black-box attack. Later, Jiawei Su [29] proposed a pixel based attack named one pixel attack, which by only modifying one pixel of the input, can successfully mislead neural network. Experiment result showed that 67.97% of Cifar-10 data and 16.04% of ImageNet data can be corrupted by one pixel attack with average confidence decreasing 74.03% and 22.91%.

Black-box attack: The current mainstream black-box attack methods are mainly divided into the following three types: Substitute Model, Gradient estimation and heuristic search [30].The substitute model mainly based on the transferability of adversarial examples, which means that the adversarial examples generated in one specific model are probably effective in other models [31]. Papernot [32] first utilized this feature to generate adversarial examples, where they first generated adversarial examples in local models and then transferred them

to other models. Liu [31] first test the transferability of the adversarial examples in a large-scale dataset, such as ImageNet and also indicated that the non-targeted adversarial showed a better transferability. Final, it indicating that the transferability of adversarial examples is due to the redundancy after the decision boundary is mapped to low dimension.

While gradient based adversarial attack methods are widely used in white-box situation, some research showed that the gradient based black-box attack method also achieved good performance in generating adversarial examples. Chen [33] et al. proposed a black-box attack method based on zero-order optimization (ZOO). This scheme achieved a good attack success rate, but its generation speed was not high. Later, Tu [34] et al. combined the method of self-encoding and proposed a method called AutoZoom, which effectively improved its generation speed. Meanwhile, referring to typical white-box gradient search methods such as FGSM and BIM, Bhagoji [35] et al. proposed a black-box attack method named FD based on gradient estimation. In addition, Ilyas [36] et al. combined natural evolution strategy (NES, natural evolution strategy) with PGD for gradient estimation to generate adversarial examples.

The heuristic search method generates adversarial examples through derivative-free optimization (DFO). Brendel et al. [37] proposed a decision-based adversarial example generation method using only input labels, which by iteratively access the neural networks' output to judge whether the added perturbation makes the input close to decision boundary. In addition, Bhagoji [35] et al. combined DFO with particle swarm optimization and proposed a black-box attack method called FD-POS. Particle swarm optimization methods have also been used to fool face recognition systems. However, due to the fact that most black-box methods generate adversary by estimating, the perturbation caused by black box methods is usually larger than white-box methods to gain a good performance. Figure 2.5 and Figure 2.6 show the two examples of heuristic search based adversarial examples.

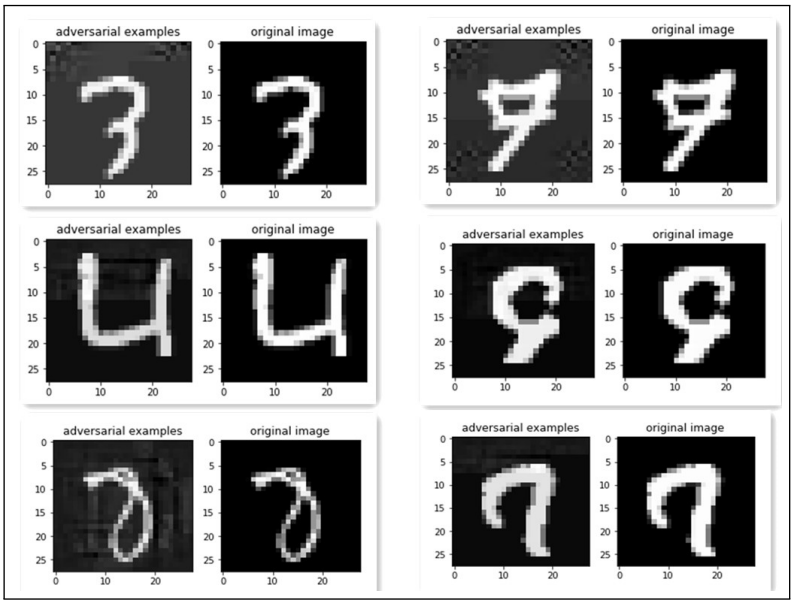


Figure 2.5: The black-box attack adversarial examples generated in MNIST [4]

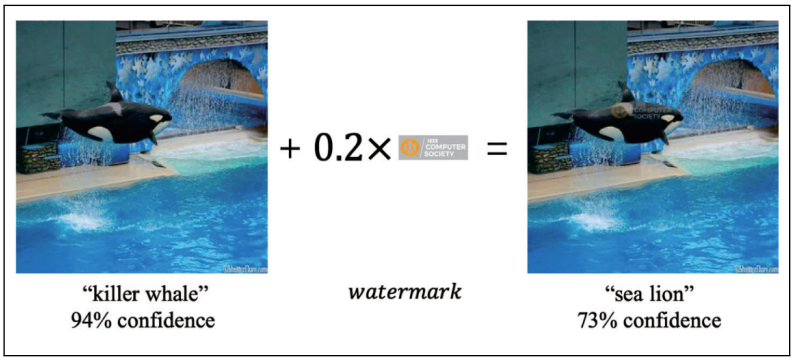


Figure 2.6: The black-box watermark attack [5]

2.2.3 Heuristic based adversarial attack defense methods

At present, the mainstream Heuristic based adversarial attack defense methods can be roughly divided into three categories: defense based on modified neural networks, defense by adding additional networks, defense based on adversarial training. This section will further describe the working mechanism of each of the above three categories.

Defense based on modified neural network: Defensive distillation network (defensive distillation) was first proposed by Hinton et al. [6], who believes that a

robust deep learning network must satisfy that there is a relatively smooth classification function, which can intuitively classify the input relatively consistently in a given field. Moreover, when encountering adversarial attacks, modifying the network structure will waste too much computing resources and time. To solve the above problems, Hinton et al. [6] proposed an adversarial method based on distillation network, the principle of which is shown in Figure 2.7.

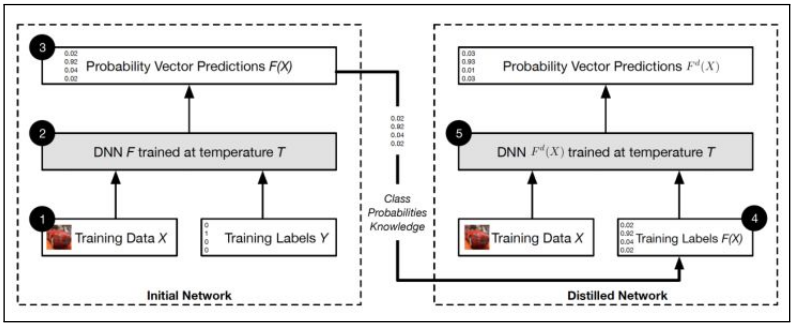


Figure 2.7: The architecture of distillation network [6]

The model first trains a deep neural network classifier through the input samples and labels, and then estimates its probability distribution $F(x)$. Then, use the sample x and the estimated result $F(x)$ of the first step to train a distillation network with the same distillation temperature T and architecture to obtain a new probability distribution $F^d(x)$. Finally, it combines the entire model for classification, so as to achieve effective defense against sample attacks.

Defense based on adding additional network: Adding additional network is a method of adding an additional defense network to the model through training, which can defend against adversarial examples without adjusting the parameters and architecture of the model. Naveed et al. [7] proposed a Perturbation Rectifying Network (PRN, Perturbation Rectifying Network) to resist Universal Adversarial Perturbations. A separate detector is trained by adding a preprocessing perturbation correction network layer to the initial model and extracting features from the input and output differences of the PRN of the training images. The test image is first passed through the PRN, and then its features are used to detect

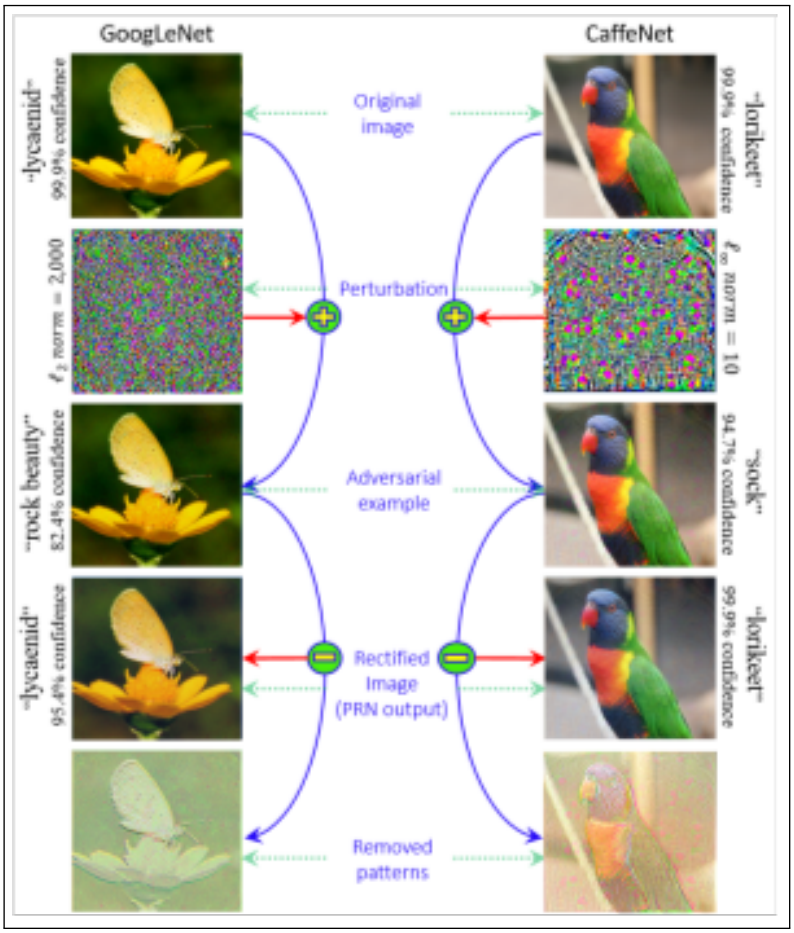


Figure 2.8: The defense based on perturbation rectifying network [7].

perturbations. If an adversarial perturbation is detected, the output of the PRN is used to classify the test image. Figure 2.8 shows the correction process performed by the PRN, and the removed patterns are analyzed separately by the detector. This method has been tested on VGG-F, CaffeNet and GoogLeNet models, and has a certain transfer ability.

Defense based on adversarial training: The adversarial training method starts from the input data set, which adds the adversarial perturbation to the training data set, and continuously updates and generates the new adversarial examples in the training step. Goodfellow [8] et al. used the adversarial training method to train the model on the MNIST dataset, where experiments showed that the training model after incorporating the adversarial examples had stronger robustness. Kuraki et al. [38] used this scheme to conduct robustness tests on the

ImageNet dataset. By continuously adding adversarial example in the training phase, the experiment results show that the network trained by this scheme has better performance against single-step attacks, but being ineffective against adaptive iterative attacks. Harini [39] et al. improved the adversarial training process and used the logit pairing method to optimize adversarial training. It has been tested on the ImageNet dataset, and the experiment results showed that the model trained by the logit pairing method is better than vanilla. This scheme increases the failure probability of the top white-box PGD attack from 1.5% to 27.9%, and reduces the success rate of the top black-box attack from 66.6% to 47.1%.

Additionally, Lee et al. [40] proposed an adversarial example defense method based on generative adversarial network (GAN, generative adversarial network). By training the classifier and the adversarial generative network at the same time, the robustness of the classifier to adversarial examples is finally enhanced after continuous and repeated adversarial attack interactions. In addition, this method can effectively reduce the overfitting of the neural network and improve the regularization ability of the model. Subsequently, Samangoue et al. [41] proposed an adversarial example protection network named Defense-GAN. By properly training the additive network, Defense-GAN can learn the distribution of the original samples and find the closest clean sample input classifier while under adversarial attack. Experiments show that this scheme has good portability and can effectively resist many different attack strategies.

However, adversarial training can only add one or several specific adversarial perturbations in the training process, so it has poor generalization ability for other unknown or stronger attacks. In addition, during adversarial training, the network needs to be continuously updated and by interaction new adversarial examples are generated, which requires a lot of time and resources. Models using adversarial training can achieve a good result in the short term, but it is difficult to guarantee good robustness in the long run.

2.2.4 Randomized Smoothing

Randomized smoothing is a certified defense methods which can guarantee every input a certified radius R, within which no adversary exists. Assume a base classifier $f_\theta : x \to U^k$ and a mapping function (usually SoftMax) $F : U^k \to P^k$ where x is the n dimensional inputs, $U^k \sim \{u_1, ..., u_k\}$ is the corresponding output value of k predicted classes and $P^k \sim \{p_1, ..., p_k\}$ is the probability of k predicted classes. With Gaussian corruption, randomized smoothing would turn any base classifier f_θ to a smoothed version g_θ by changing its prediction on one point x to a certain Gaussian distribution $x + \varepsilon \sim N\left(x, \sigma^2 I\right)$, where ε is the n dimensional isotropic Gaussian noise with mean 0 and variance σ^2. The predicted probability $\widehat{P_c}$ and the classifying result of smoothed classifier g_θ is defined in Equation 2.4 and Equation 2.5.

$$\widehat{P_c} = \underset{x+\varepsilon \sim N\left(x,\sigma^2 I\right)}{E} (M(f_\theta(x+\varepsilon)_c)) \tag{2.4}$$

$$g(x) = \arg\max_{c \in y}(\widehat{P_c}) \tag{2.5}$$

Since the value of $\underset{x+\varepsilon \sim N\left(x,\sigma^2 I\right)}{E} (M(f_\theta(x+\varepsilon)_c))$ is not possible to be exactly evaluated, in [cohen], it is first estimated by Monte Carlo sampling that succeed with arbitrarily high probability. In other word, the base classifier computed the prediction over a gaussian distribution with mean as the clean input by Monte Carlo sampling and the smoothed classifier returns the class label with highest average probability. By smoothing the original classifier, for each input with label c_A and also being correctly classified by smoothed classifier $g(x)$, the input is guaranteed with a norm based (most l_2 norm based and the following dissertation only considers l_2 norm based situation for simplicity) certified robustness radius R. The certified robustness radius R is calculated by Equation 2.6, where Φ^{-1} is the inverse of standard Gaussian Cumulative Distribution Function (CDF)

and $\underline{P_A}$ is the lower bound probability estimation of the most probable class c_A and $\overline{P_B}$ is the upper bound probability estimation of the "runner-up" class c_B.

$$R = \frac{\sigma}{2} * (\Phi^{-1}(\underline{P_A}) - \Phi^{-1}(\overline{P_B})) \tag{2.6}$$

Chapter 3

Certified defense with randomized smoothing

3.1 Mathematical support for randomized smoothing

Theorem 1 For any adversarial perturbation δ with its l_2 norm $\|\delta\|_2 \leq$, the classification result for adversarial example $x' = x + \delta$ by smoothed classifier $g(x)$ defined as in Equation 2.5 is guaranteed to be the same as the correct result c_A.

The proof of Theorem 1 is conducted by [16, 17] and this dissertation just shows some key steps to help understand the mechanism. For more details, please refer to Lemma 1 and Lemma 2 in [16]. The proof is based on the following lemma:

Lemma 1 *For any measurable function* $f : X \to [0,1]$, *let* $\Phi(a) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{a} \exp(-\frac{1}{2}s^2)ds$ *and* $\widehat{f} = \underset{x+\varepsilon \sim N(x,\sigma^2 I)}{E}(f_\theta(x+\varepsilon)_c)$, *the mapping* $x \to \Phi^{-1}(\widehat{f})$ *is* $1/\sigma$ *-Lipschitz.*

This Lemma is generalized from the Lemma 2 in [salman]. Assume $y' =$

$\arg\max_{y' \neq y}(\underset{\varepsilon \sim N(0,\sigma^2 I)}{E}(f_\theta(x+\varepsilon)_c))$. For any class $c \in Y$, define $\widehat{f_c}$ as:

$$\widehat{f_c}(x) = \underset{\varepsilon \sim N(0,\sigma^2 I)}{E}(f_\theta(x+\varepsilon)_c) \tag{3.1}$$

For $f : X \to [0,1]$, by lemma 1, we have $x \to \Phi^{-1}(\widehat{f_c}(x))$ is $1/\sigma$ -Lipschitz. So, for any $y' \neq y$, if the perturbation δ, which fulfills that $\|\delta\|_2 \leq \frac{\sigma}{2} * (\Phi^{-1}(\widehat{f_y}(x)) - \Phi^{-1}(\max_{y' \neq y} \widehat{f_{y'}}(x)))$, we have:

$$\begin{aligned}
&\Phi^{-1}(\widehat{f_y}(x+\delta)) \geq \Phi^{-1}(\widehat{f_y}(x)) - \tfrac{1}{2} * [\Phi^{-1}(\widehat{f_y}(x)) - \Phi^{-1}(\widehat{f_{y'}}(x))] \\
&\geq \tfrac{1}{2} * [\Phi^{-1}(\widehat{f_y}(x)) + \Phi^{-1}(\widehat{f_{y'}}(x))] \\
&\Phi^{-1}(\widehat{f_{y'}}(x+\delta)) \leq \Phi^{-1}(\widehat{f_{y'}}(x)) + \tfrac{1}{2} * [\Phi^{-1}(\widehat{f_y}(x)) - \Phi^{-1}(\widehat{f_{y'}}(x))] \\
&\leq \tfrac{1}{2} * [\Phi^{-1}(\widehat{f_y}(x)) + \Phi^{-1}(\widehat{f_{y'}}(x))]
\end{aligned} \tag{3.2}$$

Thus,$\Phi^{-1}(\widehat{f_y}(x+\delta)) \geq \Phi^{-1}(\widehat{f_{y'}}(x+\delta))$ and due to the monotonicity of Φ^{-1}, we have $\widehat{f_y}(x+\delta) \geq \widehat{f_{y'}}(x+\delta)$, which guarantee that $g(x+\delta) = y$.

3.2 Existing problem in robustness gradient

In randomized smoothing, the mapping function F is the bridge connecting the neural network's output U^k and and the probability P^k, which is utilized in calculating robustness.However, in [15] work, they took 0-1 hard mapping as the intermediary, where the class with largest neural network's is assigned with probability 1 and others 0. This consequently makes the gradient between robustness and neural network's outputs disappear, thus making gradient descent optimization inapplicable. So in [33]'s work, they only utilized Gaussian augmentation when training, conducing to a poor robustness performance.

Later, [16] and [17] proved Lemma 1, which supports that Theorem 1 still establish with the soft mapping function, such as SoftMax or Sigmoid. So,

in [17], it first proposes using a soft mapping function to keep the robustness's gradient thus solving the robustness optimization with gradient descent methods. In his work, the optimization is divided into two parts: accuracy item and robustness item, which is shown in Equation 3.3:

$$\begin{aligned} &\max \underset{(x,y\sim D)}{E}(ACC+\beta * Robustness) \\ &ACC=\log(g_\theta(x)_y) \\ &Robustness=\Phi^{-1}(g_\theta(x)_y) \end{aligned} \tag{3.3}$$

where $g_\theta(x)_y = \underset{x+\varepsilon\sim N\left(x,\sigma^2 I\right)}{E}(F(f_\theta(x+\varepsilon)_y))$ is the output probability of smoothed classifier. The reason why it only uses correct class probability to calculate the robustness is that for multi-classes classification the probability $1-P_A$ can always be the upper bound probability of the runner-up class. However, in MACER [17], it only indicates the robustness optimization item could be solved by gradient descent methods, but regarding two potential problems:

1. The gradient of robustness is extremely uneven and becomes infinite when $g_\theta(x)_y$ nears to 1, which cause slowly converging speed and gradient exploration when our target is make $g_\theta(x)_y$ to 1.
2. There would be a larger estimation error when we only sample a small number of points from the Gaussian distribution $x+\varepsilon\sim N\left(x,\sigma^2 I\right)$.

Those two drawbacks make it hard to train a well performed MACER model, where the training process is time-consuming and pretty sensitive to the selection of hyper-parameters. To eliminate the first drawback, we first calculate the gradient of robustness item towards the neural network's output, shown in the following Equation 3.4:

$$\begin{aligned} &\nabla\Phi^{-1}(g_\theta(x)_y)=\frac{\nabla g_\theta(x)_y}{\Phi'(\Phi^{-1}(g_\theta(x)_y)} \\ &=\nabla g_\theta(x)_y * \sqrt{2\pi} * \exp(\frac{\Phi^{-1}(g_\theta(x)_y)^2}{2\sigma^2}) \end{aligned} \tag{3.4}$$

Where $g_\theta(x)_y \approx \frac{1}{n}\sum_{i=1}^{n}[F(f_\theta(x+\varepsilon_i)_y)]$ is the estimated probability by Monte Carlo sampling. It is easy to find out that when $g_\theta(x)_y$ is near to 1, the gradient of robustness will be infinite. This is easily to cause gradient exploration while our target is to make $g_\theta(x)_y$ as near as 1. To solve this problem, we suggest two solutions:

1. The soft mapping function should choose those functions which has a zero gradient when $g_\theta(x)_y$ nears to 1, such as Sigmoid or SoftMax.
2. Considering the estimation error and gradient exploration problem, there must be a threshold on the estimated probability $g_\theta(x)_y$.

Figure 3.1 gives the comparison between modified robustness gradient and original robustness gradient, which shows that with threshold 0.99 and SoftMax mapping, the robustness gradient has a much smoother trend and will not explore.

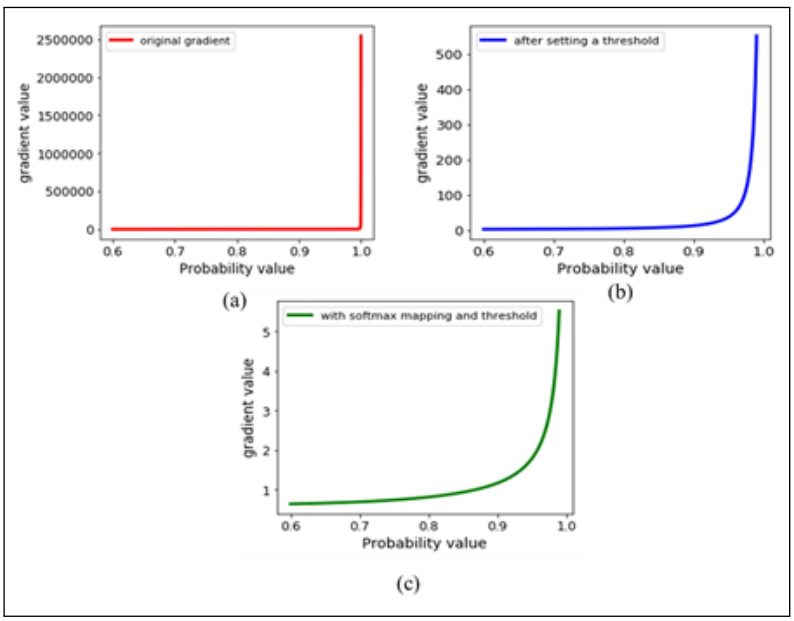


Figure 3.1: the gradient of the Robustness among the domain of , (a) the gradient without any modification, (b) the gradient with a threshold 0.99, (c) the gradient with softmax mapping and a threshold.

Later, [18] propose an original robustness optimization method named Consis-

tency Regularization, which tries to make the base classifier give a constant prediction result for all the points belongs to one specific distribution $x+\varepsilon \sim N\left(x,\sigma^2 I\right)$. The training loss in[Jpngheon] is shown as follows:

$$\begin{aligned} L &= L^{nature} + L^{consistency} \\ &= \underset{x+\varepsilon\sim N\left(x,\sigma^2 I\right)}{E}[L(F(x+\varepsilon),y) + \lambda * KL(\widehat{F}(x) \parallel F(x+\varepsilon)) + \eta H(\widehat{F}(x))] \\ &\approx \tfrac{1}{m}\sum_i (L(F(x+\varepsilon_i),y) + \lambda * KL(\widehat{F}(x) \parallel F(x+\varepsilon_i))) + \eta H(\widehat{F}(x)) \end{aligned} \tag{3.5}$$

Where $KL(\cdot \parallel \cdot)$ and $H(\cdot)$ denote the Kullback-Leibler (KL) divergence and the entropy respectively. In the above formula, the $l^{nature} = l(F(x+\varepsilon_i),y)$ tries to make the base classifier correctly classifies as many data as possible and the $l^{consistency} = l(\lambda * KL(\widehat{F}(x) \parallel F(x+\varepsilon)) + \eta H(\widehat{F}(x))$ tries to make the model gives the same prediction result for all points belonging to the same distribution $x+\varepsilon \sim N\left(x,\sigma^2 I\right)$. Though this consistency loss not directly optimizes the robustness item, with the 0-1 hard mapping, making all the points in one distribution get the same largest output is approximately making P_A to 1, which maximize the robustness. Meanwhile, this loss perfectly avoids the gradient exploration problem and not so heavily relies on the Monte Carlo estimation result $\widehat{F}(x)$. Thus, the consistency optimization outperforms MACER in both training efficiency and model's performance.

However, in consistency optimization, giving the constant prediction results for every sampled point belonging to one distribution is a hard condition. Meanwhile, for those misclassified data, making all the sampled points get the same wrong prediction not only waste the computation resource but also reduce their chances to be correctly classified. In the next chapter, this dissertation first proposes a modified optimization based on consistency named generalized consistency optimization, consisting of a loose accuracy item and a tighter robustness item. Meanwhile, to solve the class discriminant information loss problem caused by high level noise variance, the linear decomposition method is utilized to select the useful information.

Chapter 4

Generalized Optimization and Linear Decomposition

This chapter will discuss two proposed methods in this dissertation. The generalized consistency optimization is a generalized version of [], consisting of a looser accuracy item and a tighter robustness item. Meanwhile, this dissertation first discusses the discriminant information loss problem due to high variance level noise and utilizes the linear decomposition method to select useful information.

4.1 Generalized Consistency Optimization

Follows [18], our generalized consistency also includes two parts: nature accuracy loss and robustness consistency loss. The original consistency aims to make all inputs, including those being misclassified, to get the constant prediction within one specific distribution. However, achieving the above target might have two potential problems:

1. With high variance level noise, the sampled points $x+\varepsilon \sim N\left(x, \sigma^2 I\right)$ will distribute in a very large l_2 -ball area, thus making it difficult to keep all points be classified the same.

2. For those misclassified distribution $N\left(x,\sigma^2 I\right)$, making all points $x+\varepsilon \sim N\left(x,\sigma^2 I\right)$ get the same wrong prediction not only waste the computation resource but also reduce their chances to be correctly classified.

So, first we constrain the consistency loss, which only requires points, belonging to the corrected classified distribution $N\left(x,\sigma^2 I\right)$, to get a constant prediction result. The new consistency loss is shown as follows:

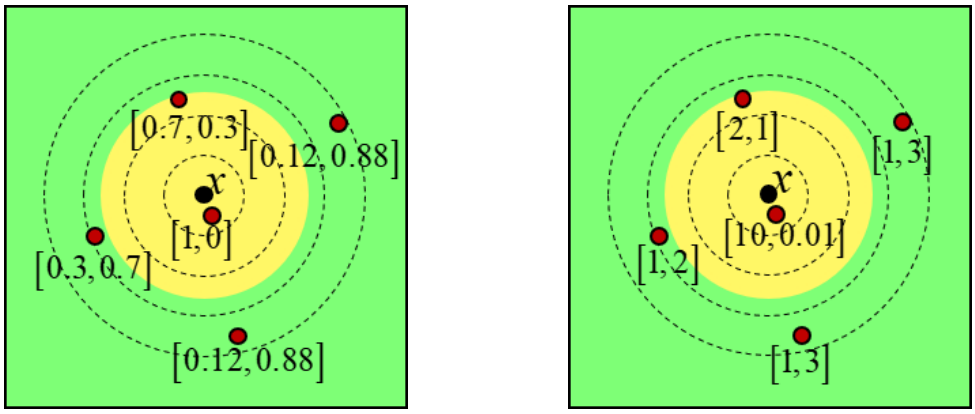
$$L^{\text{consistency}} = \underset{x+\varepsilon\sim N\left(x,\sigma^2 I\right)}{E}[\lambda * KL(\widehat{F}(x) \parallel F(x+\varepsilon)_{\left\{\arg\max \widehat{F}(x)=c\right\}}) + \eta H(\widehat{F}(x)) \quad (4.1)$$

However, only regularizing those correctly classified distributions will restrict the perception area of this loss. In other word, this loss will not consider those distributions which are promising to be correctly classified, thus retarding robustness converge speed. To solve this, we loose the accuracy condition, aiming to find more potential promising distribution which could be correctly classified. Meanwhile, due to the fact that within a Gaussian distribution, those points with a smaller l_2 distance from the mean value is more similar to clean image thus usually easier to be classified. So, when deciding the final prediction result, it is reasonable to assign those points with a smaller l_2 from mean value a higher weight. In past research, the accuracy item is defined in Equation 4.2 , where due to the fact $F : X \to [0,1]$, the maximal weight of each points in deciding the final classification is 1. To make the voting classification more flexible, we loose the accuracy item, defined as Equation 4.3, where the weight of some points in final decision can be large enough if they are really important point.

$$g(x) = \underset{c\in y}{\arg\max}[\underset{x+\varepsilon\sim N\left(x,\sigma^2 I\right)}{E}(F(f_\theta(x+\varepsilon)_c))] \quad (4.2)$$

$$g(x) = \underset{c\in y}{\arg\max}[F(\underset{x+\varepsilon\sim N\left(x,\sigma^2 I\right)}{E}(f_\theta(x+\varepsilon)_c)] \quad (4.3)$$

To help easier understand the difference between those two accuracy item, in



(a) (b)

Figure 4.1: Weight of each point for deciding the final decision

Figure 4.1, we give a 2-dimensional situation to illustrate. In f(a), the weight of each point for deciding the final prediction probability is up to 1, while in f(b), those points near to the original point can be assigned with a larger weight and lay a greater influence on the final decision. In chapter 5, detailed experiment results are given to support that our proposed loose accuracy item not only have a faster converge speed, but also can achieve a higher accuracy.

4.2 Information selection with Linear Decomposition

While lots of research concern on how to design a more accurate optimization to make model performance better, seldom people consider the noise corruption problem. However, instead of the estimation error in optimization object, the information corruption caused by noise is the crucial factor leading to a low accuracy and robustness. Equation 2.6 indicates that the certified robustness radius is directly proportional to the standard deviation of Gaussian. Thus, to get a large robustness, high variance noise is required when classifying.

In practice, the most widely used noise standard deviation levels are 0.25 and 0.5 where the corresponding variance levels are 0.0625 and 0.25. To find out

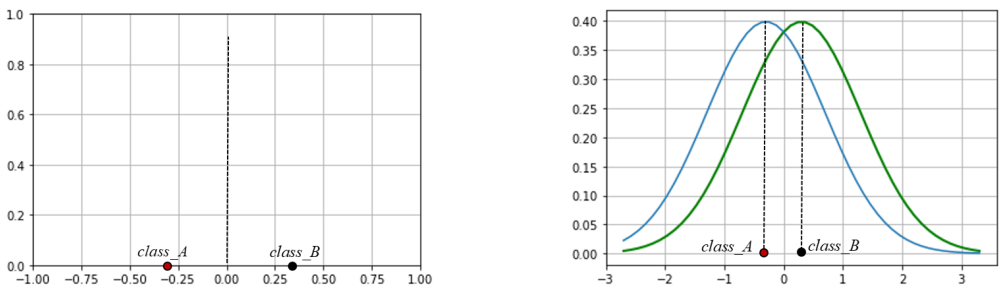
the concrete influence for adding those noise, we utilize Principal Component Analysis (PCA), which is a linear decomposition method and the transformation is shown in Equation 4.4, where X is a $m \times n$ and W is the transformation matrix with dimension $n \times n$.

$$X' = X \cdot W \tag{4.4}$$

PCA will transform the input from its original space to a new latent space where each normalized basis is sorted according to its the covariance and mutually orthogonal. Those two intriguing properties make PCA a suitable method to select the information from high variance level noise.

1. The fact that each normalized basis in the latent space is mutually orthogonal can guarantee that the noise variance level of each dimension in the latent space equals to the variance level of each dimension in the original space. In other word, adding the noise directly in X' is the same as adding the noise in X.

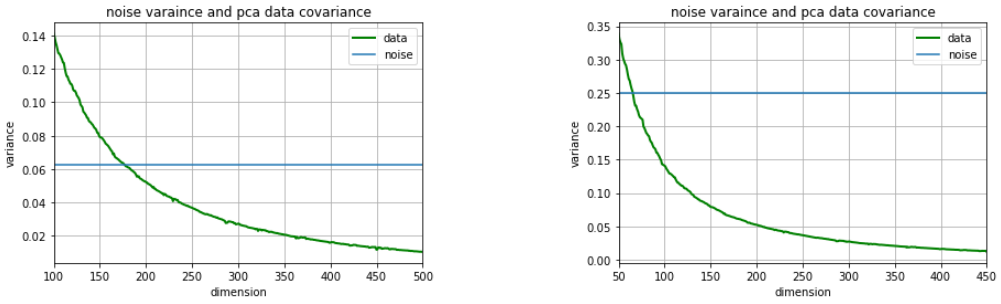
2. Though the covariance of each dimension is irrelevant to the class information. However, if the noise variance is far larger than the covariance, the information of this dimension could be thoroughly helpless in classifying.



(a) (b)

Figure 4.2: best-case covariance situation when variance of noise is 10-fold larger than the dimension's covariance:(a) original data distribution, where the intra-class variance is 0, (b) data distribution after Gaussian noise with 10-folder variance.

To support above second statement, in Figure 4.2, we give the best-case covariance situation (the intra-class variance is 0) when the variance of noise is 5-fold larger than the dimension's covariance. Obviously, the original useful class discriminant information after noise corrupting becomes helpless in classifying. In



(a) (b)

Figure 4.3: The relationship of the covariance of each dimension of Cifar10 dataset with different noise variance:(a) Noise variance 0.0625 and PCA dimension 100 to 500, (b) Noise variance 0.25 and PCA dimension 50 to 450.

Figure 4.3, we give the relationship of the covariance of each dimension after PCA with the noise variance level, which shows that, the information of most dimensions after PCA is heavily corrupted by Gaussian noise. So, to select the useful information from noise, we only save those dimension whose covariance is larger than 1/5 noise's variance.

Chapter 5

Experiment result and discussion

We validate the effectiveness of our proposed generalized consistency optimization and linear decomposition methods mainly in CIFAR-10 dataset and we widely compared our methods with Gaussian augmentation [15], MACE [17], consistency [18] and Ensemble [19], which is current state-of-the art method. Overall, the experiment results demonstrate that our method greatly outperform Gaussian augmentation and MACER in both accuracy and robustness. When compared with the original consistency optimization, our method gain around 0.34 improvement in average certified radius as long as 6.6% improvement in clean accuracy both in $\sigma = 0.25$ and 0.5. When compare with [19], which combined consistency optimization and ensembles, our proposed generalized consistency optimization with linear decomposition method still slightly outperform it in both accuracy and average robustness. Though, the total training time, robustness calculating time and model size of [19] are 2 times greater than ours.

5.1 Setups

Evaluation metrics: we evaluate our proposed method mainly based on two metrics:

Table 5.1: Approximated certified test accuracy and ACR on Cifar-10

σ	models	ACR	ACC	0.25	0.50	0.75	1.00	1.25	1.50	1.75
0.25	Ours	**0.542**	75.8	**67.4**	54.1	**45.7**	0	0	0	0
	Ensemble(m=3) [19]	0.535	75.1	66.1	**55.3**	43.5	0	0	0	0
	Consistency [18]	0.508	69.2	61.2	51.3	42.1	0	0	0	0
	MACER-150 [17]	0.449	71.4	59.0	44.6	32.8	0	0	0	0
	Gaussian+LD	0.451	**78.2**	62.5	44.2	29.2	0	0	0	0
	Gaussian [15]	0.428	74.8	60.0	42.8	26.6	0	0	0	0
0.50	Ours	**0.731**	62.0	**55.1**	**49.5**	**41.3**	35.4	29.8	**26.5**	**23.5**
	Ensemble(m=3) [19]	0.722	59.5	52.0	47.1	41.3	**37.1**	**31.8**	26.3	23.3
	Consistency [18]	0.703	56.3	52.1	45.5	39.8	33.6	29.7	25.7	24.3
	MACER-150 [17]	0.658	59.3	54.2	46.1	38.0	31.6	24.3	19.5	17.1
	Gaussian+LD	0.60	65.0	58.2	46.8	36.4	28.7	19.1	12.9	9.9
	Gaussian [15]	0.537	**65.2**	54.6	41.4	32.0	24.3	15.2	9.4	5.2

1. The certified accuracy at predetermined radius r, which means the percentage of data which can be guaranteed to be correctly classified under any adversarial perturbation within a l_2 ball with radius r. The certified accuracy equals to the clean accuracy when we set r as 0.

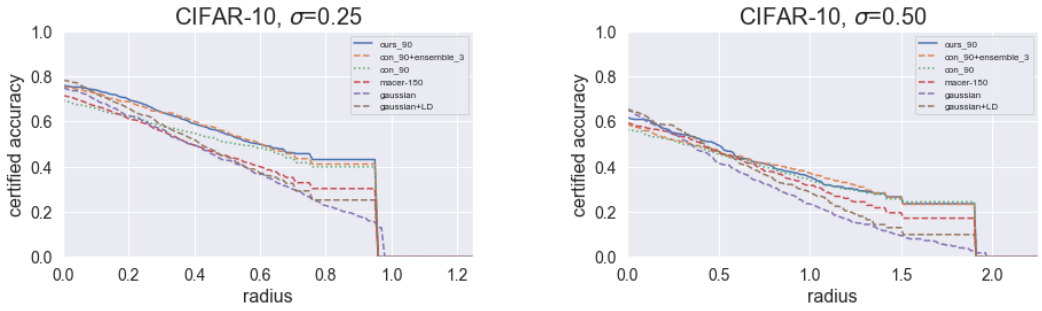
2. The average certified radius, which is the average of certified robustness radius of all test data. Large average certified radius means model's better certified robustness.

In our experiment, we followed previous work[], which use the given CERTIFY for evaluation, with $n = 100000$, $n_0 = 100$ and $\alpha = 0.001$, meaning that with probability of 0.1%, the smoothed classifier does not return the most probable class or falsely certified a non-robust input.

Training environments: We use the same model as piror work [18, 19, 33], named ResNet-110[resnet] in CIFAR-10 dataset. For a fair comparison, we use the same training scheme as consistency [18]. On CIFAR-10, the total epoches for training [19], [18], [15] and ours are all 90 epoches with SGD optimizer and initialized learning rate 0.01. we decay the learning rate at 30^{th} and 60^{th} epoch and we set the number of sampling n is 4. For ensemble, we set the

number of ensemble is 3, which includes three ResNet-110 for training and evaluation. For MACER [17], we train it for 150 epoches, with learning rate decays in 50^{th} and 100^{th} and we set the number of sampling n is 8. Usually with a larger n, the robustness estimation will be more accurate. However the total training also increasing proportional.

5.2 Result



(a) (b)

Figure 5.1: The certified radius-accuracy curve of different models,(a)$\sigma = 0.25$, (b)$\sigma = 0.50$.

We main compared our work with consistency, which is the original optimization version and ensemble, which is current state-of-the-art method. Additional, we also give the performance of [15], which is the baseline model trained with Gaussian augmentation and MACER [17], which utilizes gradient based method to maximize certified robustness. we give the performance of different models in tab1 and in Figure 5.1, we show the certified radius-accuracy curve, under which the area is ACR. In Table 5.1, it shows that our propose generalized consistency optimization with linear decomposition outperforms the original consistency optimization and slightly outperforms the current state-of-the-art method named ensemble, whose total training time, robustness calculating time and model size are 2 times greater than ours.

Table 5.2: Comparison of training efficiency on CIFAR-10 with $\sigma = 0.25$

Model	ACR	ACC	Memeories	Runtime(h)
Gaussian [15]	0.428	74.8	2.9G	0.5
Consistency [18]	0.508	69.2	2.9G	2.1
Ours	**0.542**	**75.8**	3.1G	2.2
Ensemble [19]	0.532	75.1	8.7G	5.8
MACER [17]	0.449	71.4	9.4G	7.6

5.3 Runtime analysis

With better performance on robustness, the generalized optimization with linear decomposition also shows great advantage in terms of training efficiency and memories. In this experiment, every method is training on CIFAR-10 with one GPU of NVIDIA 3080TI. We use $\sigma = 0.25$ and for ensemble, we set number of models is 3.

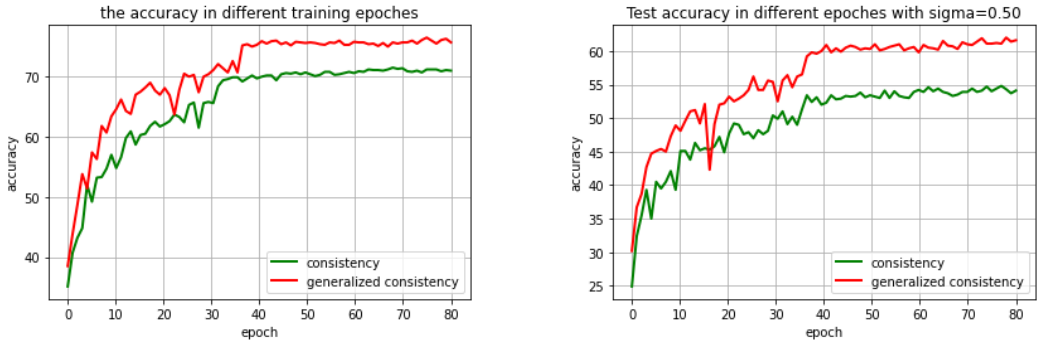
The result is shown in Table 5.2. Though our proposed method indeed cost 4 times of training time than Gaussian augmentation, when compared with MACER and ensemble, the resource needed for us is much less. Further more, our method achieves better accuracy and ACR, which support the efficiency of our proposed method.

5.4 Ablation study

To further verify the function of each part in our method: the generalized consistency optimization and the liner decomposition. We perform each part separately on CIFAR-10 dataset.

Effect of generalized consistency optimization: to verify the effectiveness of generalized consistency optimization, we compare it with the original consis-

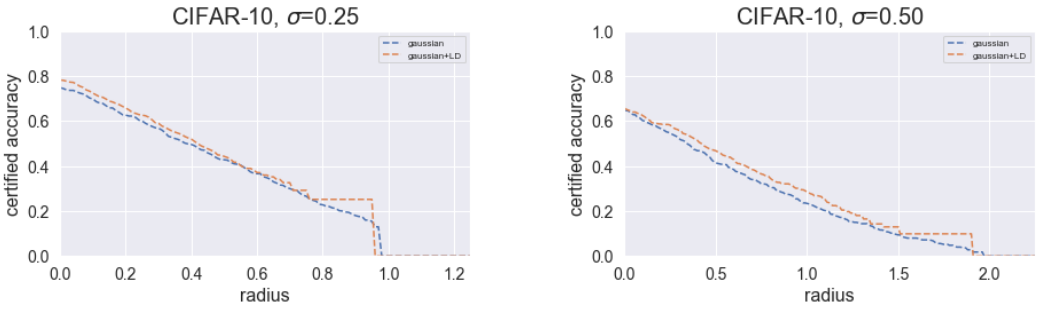
tency optimization. As our proposed generalized consistency optimization consists of a looser accuracy item, which contains a more flexible accuracy voting mechanism, and a more accurate robustness item, which will not restrict the consistency of those misclassified inputs, our accuracy more flexible to train and can consider more potential points when training. We give the accuracy in each



(a) (b)

Figure 5.2: The accuracy in different training epoch,(a)trained with $\sigma = 0.25$, (b)trained with $\sigma = 0.50$.

training epoch of the model trained by generalized consistency and consistency optimization in Figure 5.2. The result shows that, with our generalized consistency optimization, the accuracy converge much faster and achieve a higher value, which verify the effectiveness of our method. **Effect of linear decom-**



(a) (b)

Figure 5.3: The certified radius-accuracy of models with or without linear decomposed data,(a)trained with $\sigma = 0.25$, (b)trained with $\sigma = 0.50$.

position: to verify the effectiveness of linear decomposition in selecting useful information from noise data, we experiment two methods: original data trained

with Gaussian augmentation, data after PCA, which only keep those dimensions with a co-variance large than $\frac{\sigma^2}{10}$. We give the certified radius-accuracy curve of those two models in Figure 5.3, which indicates that, after information selection with linear decomposition, the model achieves better accuracy and robustness.

Chapter 6

Conclusion and future work

6.1 Conclusion

Randomized smoothing is current most effective certified defense method to defend adversarial examples. However, there are two deficiencies in randomized smoothing. First, due to utilizing of 0-1 hard mapping and Monte Carlo sampling, the robustness item is non differentiable towards the model's output, thus making maximize robustness inapplicable for directly gradient descent methods. Second, under high levels of noise, some classes discriminant information was heavily perturbed thus decreasing the final prediction accuracy.

Aiming to solve the above two problems, this dissertation proposed a generalized consistency optimization with linear decomposition method. The generalized consistency optimization provides a looser accuracy constrain and a more accurate robustness constrain when compared with original consistency optimization. Additionally, with linear decomposition, the information is selected according to the ratio of covariance to noise variance, thus discarding heavily corrupted information. With those two modifications, experiment results show the proposed method greatly outperform previous methods, including original consistency optimization. Furthermore, when compared with the state-of-the art method based on model ensembles, our method slightly outperforms it, while only using 33%

training time and storage space.

6.2 Future work

In randomized smoothing, there are still many problems unsolved, such as how to make a more accurate estimation towards robustness. Meanwhile, the linear decomposition method to select information from data is quite simply, thus being not extremely effective. Future working will further explore a more accurate robustness estimation item as well as a better information selection method.

References

[1] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[2] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[3] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 3–14, 2017.

[4] Wenli Xiao, Hao Jiang, and Song Xia. A new black box attack generating adversarial examples based on reinforcement learning. In *2020 Information Communication Technologies Conference (ICTC)*, pages 141–146. IEEE, 2020.

[5] Hao Jiang, Jintao Yang, Guang Hua, Lixia Li, Ying Wang, Shenghui Tu, and Song Xia. Fawa: Fast adversarial watermark attack. *IEEE Transactions on Computers*, 2021.

[6] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[7] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3389–3398, 2018.

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[9] 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.

[10] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[11] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33:1633–1645, 2020.

[12] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[13] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[15] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[16] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

[18] Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570, 2020.

[19] Miklós Z Horváth, Mark Niklas Mueller, Marc Fischer, and Martin Vechev. Boosting randomized smoothing with variance reduced classifiers. In *International Conference on Learning Representations*, 2021.

[20] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

[21] Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint arXiv:1706.07351*, 2017.

[22] Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

[23] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer, 2017.

[24] Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 269–286. Springer, 2017.

[25] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In *the 22nd*

International Conference on Artificial Intelligence and Statistics, pages 2057–2066. PMLR, 2019.

[26] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019.

[27] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness. 2018.

[28] Oleksii Avilov, Sébastien Rimbert, Anton Popov, and Laurent Bougrain. Deep learning techniques to improve intraoperative awareness detection from electroencephalographic signals. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 142–145. IEEE, 2020.

[29] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[30] Lei Bu, Zhe Zhao, Yuchao Duan, and Fu Song. Taking care of the discretization problem: A comprehensive study of the discretization problem and a black-box adversarial attack in discrete integer domain. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[31] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[32] 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*, 2016.

[33] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural net-

works without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[34] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

[35] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–169, 2018.

[36] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2137–2146. PMLR, 2018.

[37] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[38] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[39] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[40] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, 2017.

[41] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.