École Polytechnique Fédérale de Lausanne

# Understanding Catastrophic Overfitting in Adversarial Training

by Kang Peilin

Dr. Seyed-Mohsen Moosavi-Dezfooli
Thesis Supervisor at ETH Zürich

Prof. Martin Jaggi
Thesis Supervisor at EPFL

A thesis submitted in fulfillment of the requirements
for the degree of Master of Computer Science

***At***

Data Analytics Lab
Computer Science Department
ETH Zürich

*And with contributions from*

Machine Learning and Optimization Laboratory
School of Computer and Communication Sciences
EPFL

May 21, 2021

# Acknowledgments

I would first like to thank my master thesis supervisor Dr. Seyed-Mohsen Moosavi-Dezfooli at ETH who provided me this project and guided me all the way through. A heartful thanks for all the discussions, insightful suggestions and encouragements given by you.

I would also like to thank my supervisor Prof. Martin Jaggi at EPFL. I took Machine Learning course offered by Martin in my first master semester, and this course has cultivated my interests of machine learning topic. And I also did a semester project at MLO lab supervised by Martin that further enriched my skills towards machine learning and scientific research. Thank you for being my master thesis supervisor and for all the knowledge, kindness and supports given by you.

I also wish to acknowledge Aaksym Andriushchenko for his discussion and valuable advices. I would also like to express my thanks to the computational resources provided by Data Analytics Lab in the Computer Science Department of ETH Zürich and the help provided by the technical and support staff of cluster.

Last but not least, I wish to extend special thanks to my parents who provided me financial support to make me focus on my master project. And I would also like to show my appreciation to my friends who always encouraged me when I feel upset.

*Zürich, May 21, 2021* Kang Peilin

# Abstract

Recently, FGSM adversarial training is found to be able to train a robust model which is comparable to the one trained by PGD but an order of magnitude faster. However, there is a failure mode called catastrophic overfitting (CO) that the classifier loses its robustness suddenly during the training and hardly recovers by itself. In this paper, we find CO is not only limited to FGSM, but also happens in DF$^\infty$-1 adversarial training. Then, we analyze the geometric properties for both FGSM and DF$^\infty$-1 and find they have totally different decision boundaries after CO. For FGSM, a new decision boundary is generated along the direction of perturbation and makes the small perturbation more effective than the large one. While for DF$^\infty$-1, there is no new decision boundary generated along the direction of perturbation, instead the perturbation generated by DF$^\infty$-1 becomes smaller after CO and thus loses its effectiveness. We also experimentally analyze three hypotheses on potential factors causing CO. And then based on the empirical analysis, we modify the RS-FGSM by not projecting perturbation back to the $l_\infty$ ball. By this small modification, we could achieve $47.56 \pm 0.37\%$ PGD-50-10 accuracy on CIFAR10 with $\epsilon = 8/255$ in contrast to $43.57 \pm 0.30\%$ by RS-FGSM and also further extend the working range of $\epsilon$ from 8/255 to 11/255 on CIFAR10 without CO occurring.

# Contents

# Chapter 1

# Introduction

In recent years, deep learning has achieved great success on many different tasks, such as image classification [11] and language modeling [7]. However, those deep neural networks can be easily attacked by adding small, well-designed, imperceptible perturbations into clean examples [3] [27], which raises security concern when we deploy those models into real-world applications, such as autonomous driving. It would be devastating if a model misclassifies a stop sign to a straight-ahead sign because someone adds well-designed perturbations which are imperceptible by humans. Thus, it's important to train a robust model which is not only accurate on normal clean examples but also on small perturbed adversarial examples.

There are many different types of defense methods, such as preprocessing based methods [10][4] [26][24], regularization based methods [23][18][20], provable defenses on norm-bounded perturbation [30][21][9][5] and adversarial training methods [8][28][14][16][34][25]. Some defenses are found to give a false sense of security because of gradient obfuscation, and they could be defeated by well-designed ad-hoc attacks [2]. Also, a recent paper [6] evaluates top 50 defensive methods and finds most of them either lead to lower robust accuracy or will be broken using a stronger attack. In the end, adversarial training using PGD attack [16] or its variations [34][19][32] leads to the most stable model which has the best empirical robustness when it faces different attacks. Thus, we will focus on adversarial training methods in this paper.

Adversarial training is to train models on adversarial examples constructed by different attack methods. Although we can get a robust model using projected gradient descent (PGD) adversarial training [16], the main drawback of this method is heavy computational overhead resulted from using multi-step gradient descent to generate adversarial examples in each mini-batch weight updates. PGD adversarial training is an order of magnitude slower than standard training, and thus limits scalability to large neural networks, such as ImageNet. Though there are many works trying to accelerate PGD without having to sacrifice its performance, such as [34][25][33], they are still much slower than standard training.

In order to improve computational efficiency, fast gradient sign method (FGSM) adversarial

training [8] uses one gradient step to construct adversarial examples instead of using multi-step gradient descent. This method is computationally efficient, but can be easily defeated by stronger multi-step attacks [28][13]. However recently, [31] claims by simply adding random initialization before FGSM, we can get a model with comparable robustness as the one trained by multi-step methods, such as PGD. But this approach can not always defend against strong multi-step attacks, and this paper[31] presents a failure mode named **catastrophic overfitting (CO)** that the model suddenly losses robustness in a few epochs during training and barely recover by itself. Although the paper [31] finds that we can use a small validation set to evaluate the model during training and stop it before catastrophic overfitting, the model we get is still sub-optimal because of insufficient training. Although there are some works trying to understand why catastrophic overfitting happens and improve adversarial training [1][15][29][12], none of them can fully explain the reason for catastrophic overfitting and the methods proposed in these papers add additional computational overhead to FGSM.

In this project, we work on understanding catastrophic overfitting in adversarial training and try to improve it. First, we find catastrophic overfitting is not only limited to FGSM adversarial training, but also happens on RS-FGSM [31] and DF$^\infty$-1 [17] (DF's superscript of $p$ means using $p$ norm to calculate perturbation at each iteration, and the number after dash means how many iterations to use. If the number of iteration is not specified, it means to stop until finding an adversarial example). Then we analyze the geometric properties of the model before and after catastrophic overfitting. Specifically, we draw the cross-section of the decision boundary spanned by two vectors. One vector is calculated by DeepFool(DF$^2$) [17] which is perpendicular to the decision boundary, and the other one is calculated by the adversarial method used in the training process. We observe that FGSM and DF$^\infty$-1 show different geometric properties after catastrophic overfitting. For FGSM, a new decision boundary is generated along the perturbed direction, and small FGSM perturbation becomes more effective than large FGSM perturbation which causes the perturbation found by full step FGSM to become invalid. While for DF$^\infty$-1, though large perturbation is still more effective than small perturbation, the length of the perturbation generated by DF$^\infty$-1 decreases and becomes invalid. These geometric properties explain why an adversary loses its effectiveness after catastrophic overfitting happens and hardly recovers by itself. Then we focus on analyzing factors that cause catastrophic overfitting. Specifically, we propose experiments to examine three probable hypotheses, two of them put forward by previous works and one of them come up by us. And based on the empirical analysis, we modify the RS-FGSM by not projecting perturbation back to the $l_\infty$ ball, in other words, not clipping the perturbation to the $[-\epsilon, \epsilon]^d$. By this small modification, we could achieve $47.56 \pm 0.37\%$ PGD-50-10 accuracy on CIFAR10 at $\epsilon = 8/255$ in contrast to $43.57 \pm 0.30\%$ by RS-FGSM under the same setting and also further extend the working range of $\epsilon$ (radius of $l_\infty$ ball) from 8/255 to 11/255 on CIFAR10 without suffering from catastrophic overfitting. Our work makes the following **contributions**:

- We show that catastrophic overfitting is a general phenomenon and not only limited to FGSM, but also happens in RS-FGSM and DF$^\infty$-1 adversarial training.

- We analyze the geometric properties of classifiers before and after catastrophic overfitting and demonstrate that FGSM and DF$^\infty$-1 have similar decision boundaries before catastrophic overfitting and become totally different after catastrophic overfitting happens.
- We experimentally analyze three hypotheses on potential factors causing catastrophic overfitting.
- We make a modification to RS-FGSM [31] by not projecting perturbation back to the $l_\infty$ ball. On standard dataset CIFAR10, we show that this modification leads to a better robust accuracy and it permits us to use larger values of $\epsilon$.

In chapter 2, we will overview the CO problem in adversarial training and some related works trying to understand and solve it. In chapter 3, we define the formal definition of CO and show it happens both in FGSM and DF$^\infty$-1 adversarial training. Then we analyze the geometric properties before and after CO in chapter 4 and experimentally analyze three probable hypotheses on factors causing CO in chapter 5. Finally, In chapter 6, we conclude our work and propose some future work.

# Chapter 2

# Problem overview and related works

In this chapter, we will introduce adversarial training in detail, especially one gradient step method FGSM. Then we introduce the catastrophic overfitting phenomenon proposed in [31] which causes previous attempts to train robust models with FGSM and its variants to fail. Finally, we review some related works which try to understand why catastrophic overfitting happens and how to prevent it from happening or recover the model once it happens.

## 2.1 Adversarial Training

Adversarial training is one of the most effective defensive methods to train a robust model which is not only accurate on clean examples, but also on the corresponding adversarially perturbed examples [2]. Adversarial examples are generated by adding small, well-designed perturbation vectors to clean examples. The difference between adversarial and clean examples are imperceptible by humans but they are categorized into different classes by the classifier. Standard training is not enough to train a robust model to resist adversarial examples, this is why adversarial training comes in. The basic idea of adversarial training is to construct adversarial examples and then apply empirical risk minimization on those examples instead of the clean ones. Given a dataset $(x, y) \sim \mathcal{D}$, $\mathcal{D}$ is the underlying data distribution, a model $f$ with parameter $\theta$, a loss function $l$ and a threat model $\Delta$, adversarial training can be formulated as the following min-max problem[16]:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta\in\Delta} l(f_\theta(x+\delta), y)] \tag{2.1}$$

It is well studied on how to minimize outside empirical risk, such as SGD and Adam methods. So the main focus of recent works is on how to find $\delta^* \in \Delta$ of a given data point $x$ that maximizes inner loss function. In this project, we focus on $l_\infty$ bounded threat model $\Delta = \{\delta \mid \|\delta\|_\infty \leq \epsilon, \epsilon > 0\}$. The different adversaries we use to solve the inner maximization problem constitute different adversarial training methods. For simplicity, we will use $l(x+\delta, y; \theta)$ to represent $l(f_\theta(x+\delta), y)$

later in this report.

**Projected Gradient Descent (PGD)** [16] is state-of-the-art defense method which has not been broken by different attack methods [2][6]. As shown in Equation 2.2, PGD first selects a random start point $\delta^0$ inside the $l_\infty$-ball and then uses multi-steps to solve the inner maximization problem with step size $\alpha < \epsilon$ in each step. During each iteration, the perturbation will be projected back to the $l_\infty$-ball. In order to get better inner maximization quality, PGD also adds several random restarts inside the $l_\infty$-ball.

$$\begin{aligned} \delta^0 &\sim \mathcal{U}([-\epsilon, \epsilon]^d) \\ \delta^{t+1} &= \prod_{[-\epsilon,\epsilon]^d}(\delta^t + \alpha \text{sign}(\nabla_x l(x + \delta^t, y; \theta))) \end{aligned} \tag{2.2}$$

The main drawback of PGD is the huge computational overhead, which can be an order of magnitude higher than standard training and limits its scalability to train the large deep neural networks, such as ImageNet.

In order to reduce computational cost, there is another method called **Fast Gradient Sign Method (FGSM)** [8]. FGSM uses one gradient step to approximate the inner maximization problem. As shown on Equation 2.3, FGSM takes the sign of input gradient as the direction where loss increases most rapidly and then multiplies it with step size $\alpha$ (in paper[8] $\alpha = \epsilon$). However, [28][13] claims FGSM is only workable under limited circumstances, for example under small $\epsilon$. When $\epsilon$ is large, the model trained by FGSM can be broken by stronger attacks, such as PGD.

$$\delta_{FGSM} = \alpha \text{sign}(\nabla_x l(x, y; \theta)) \tag{2.3}$$

Recently, the paper [31] revisits FGSM and comes up with a method named **Fast Gradient Sign Method with Random Initialization (RS-FGSM)**:

$$\begin{aligned} \delta &\sim \mathcal{U}([-\epsilon, \epsilon]^d) \\ \delta_{RS-FGSM} &= \prod_{[-\epsilon,\epsilon]^d}(\delta + \alpha \text{sign}(\nabla_x l(x + \delta, y; \theta))) \end{aligned} \tag{2.4}$$

The paper [31] claims that FGSM adversarial training combined with **random initialization** is as effective as PGD-based training but an order of magnitude faster. This paper [31] also demonstrates a failure pattern, named **catastrophic overfitting** (CO) and states this might be the reason why previous attempts at FGSM and its variations fail. Catastrophic overfitting is a phenomenon where robust accuracy with respect to a strong adversary (e.g., PGD) suddenly and drastically drops to 0% after several training epochs and rarely recovers by itself.

This paper [31] does not explain why adding random initialization can make FGSM workable with $\epsilon = 8/255$ on CIFAR10 while vanilla FGSM fails. And it also does not explain why RS-FGSM still suffers from catastrophic overfitting when we further increase the step size $\alpha$ and how to avoid it. In the next subsection, we will introduce some related works trying to understand and solve catastrophic overfitting problems.

## 2.2 Related works

In this subsection, we will review some related works trying to understand catastrophic overfitting problems and improve adversarial training. The paper [29] does not use the term catastrophic overfitting, but points out the same failure mode when training a model using FGSM. The author thinks this is because of overfitting to FGSM perturbations and empirically shows that by adding dropout layer after all non-linear layers and decreasing dropout probability during the training process, we can train a model using FGSM and attain the same robustness as the one trained by stronger attacks, such as PGD. However, the proposed method is evaluated on small $\epsilon$, so it is hard to tell whether this dropout schedule is still effective or not when $\epsilon$ is larger. Another concern is that by using dropout after each non-linear layer, we may face the problem of underfitting. This paper came out almost the same time as [31].

In the paper [1], the author fixes the step size of vanilla FGSM and RS-FGSM to be $\epsilon$ and $1.25\epsilon$ separately, and varied $\epsilon$ of $l_\infty$ threat model from 1/255 to 16/255. The author finds vanilla FGSM experiences catastrophic overfitting when $\epsilon > 6/255$, while RS-FGSM starts to fail when $\epsilon > 9/255$. Then it claims that the main reason why RS-FGSM is more effective than vanilla FGSM under larger $\epsilon$ is that random initialization decreases the expected magnitude of the perturbation. Finally, this paper proposes a regularization method, named GradAlign, which prevents catastrophic overfitting by explicitly maximizing the gradient alignment inside perturbation set to improve the linear approximation quality. By using FGSM+GradAlign, we can prevent catastrophic overfitting but add additional computational cost.

Another paper [15] claims the key success factor of RS-FGSM is the ability to recover from catastrophic overfitting subject to weak attacks. And then they propose a simple fix: switch to multi-step PGD once catastrophic overfitting is detected, and switch back to FGSM after the model recovers to normal. This method is simple but does not explain why multi-step PGD can help the model recover from catastrophic overfitting and also adds additional computational cost.

The paper [12] is a concurrent work of this project. The paper hypothesizes that catastrophic overfitting is caused by **decision boundary distortion**. The author finds when catastrophic overfitting happens, there is a trend of decreasing of expected $l_1$ norm of PGD7 perturbations $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\|\delta_{PGD-7}\|_1]$, increasing of expected squared $l_2$ norm of the input gradients $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\|\nabla_x l(x,y;\theta\|_2]$, and the smaller perturbation can fool the classifier whereas the classifier is robust against larger perturbations. Based on this phenomenon, the author comes up with an idea that the decision boundary is distorted after catastrophic overfitting. And then the paper suggests a simple method that can prevent decision boundary distortion by searching appropriate step size for each input example during adversarial training. But searching minimum scaling

k causes additional computation overhead:

$$\begin{aligned} \delta &= \epsilon \text{sign}(\nabla_x l(x, y; \theta)) \\ k^* &= \min_{k \in [0,1]} [k | y \neq f_\theta(x + k\delta)] \\ \delta &= k^* \delta \end{aligned} \tag{2.5}$$

Actually, the reason why catastrophic overfitting happens remains unclear, regardless of these previous works mentioned. And all previously proposed solutions result in further computational overhead compared to FGSM.

In this chapter, we introduce different adversarial training methods. Though PGD can lead to a robust model, it is computationally expensive. One gradient step methods, such as FGSM and RS-FGSM, are computationally efficient but suffer from catastrophic overfitting which will cause robust accuracy against strong attacks fails to 0. Although there are some works trying to understand the catastrophic overfitting problem, the reason why it happens remains unclear. In this project, we will focus on understanding catastrophic overfitting in adversarial training. In the next chapter, we will introduce the catastrophic overfitting phenomenon in detail.

# Chapter 3

# Catastrophic overfitting in adversarial training

In the last chapter, we introduced a family of defense methods called adversarial training, and show one of the most stable and robust methods in this family is PGD. However PGD has huge computational overhead and hard to be applied to large deep neural networks. But when we turn to computational efficient, one gradient step FGSM methods, we encounter the problem named catastrophic overfitting. In this chapter, we first define the formal definition of catastrophic overfitting and then demonstrate that catastrophic overfitting is a general phenomenon in adversarial training which is not limited to FGSM but also happens when we use $DF^{\infty}$-1 adversarial training.

## 3.1 What is catastrophic overfitting

**Catastrophic overfitting** During the adversarial training, we construct adversarial examples using an adversarial attack method, named method-A, such as FGSM. Then we evaluate the robust accuracy on the test dataset using another stronger adversarial attack method, named method-B, such as PGD. After a certain epoch, there is a sudden increase of method-A accuracy and a sudden decrease of method-B accuracy on the both training and testing dataset. We call this phenomenon catastrophic overfitting. Catastrophic overfitting is not the same as overfitting to training dataset but is the overfitting to the weaker adversarial attack method-A.

As shown on Figure 3.1, the PreAct-ResNet18 is trained by vanilla FGSM on CIFAR10 with $\epsilon = 8/255$. Catastrophic overfitting happens around epoch 13 and results in a sudden decrease of PGD-10 accuracy on the testing dataset and a sudden increase of FGSM accuracy on both training and testing datasets. In the next subsection, we will show that catastrophic overfitting is a general phenomenon for adversarial training and not only limited to vanilla FGSM.

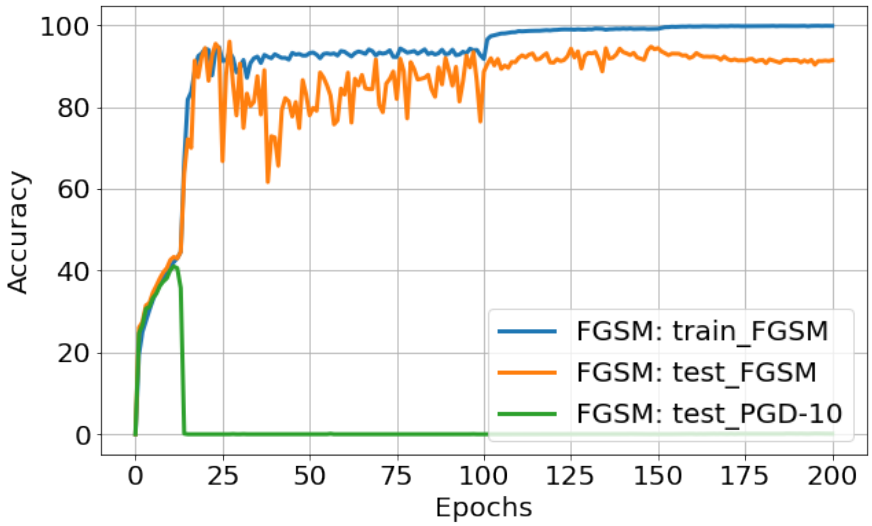



Figure 3.1 – Visualization of the training process regarding FGSM adversarial training using PreAct-ResNet18 on CIFAR10 with $\epsilon = 8/255$. Catastrophic overfitting happens around epoch 13 and is observed by a sudden increase of FGSM accuracy on the both training and testing dataset and a sudden decrease of PGD-10 accuracy on testing dataset.

## 3.2 Catastrophic overfitting: a general phenomenon for adversarial training

We first introduce the detailed experiment settings used in this project before showing more experiment results. Then we demonstrate that catastrophic overfitting still happens in RS-FGSM when $\epsilon$ or step size $\alpha$ is large.

**Experiment settings** We train the PreAct-ResNet18 model from scratch on CIFAR10 with batch size=256, epoch=200 and initial learning rate=0.1. And learning rate will be decayed at epoch 100 and 150 by 10. The paper [22] find adversarial training will overfit to the training set and the best model is not the final model we obtain at the end of the training. In this project, we either evaluate the final model without early stopping to indicate whether catastrophic overfitting happens or not or we will select the best model based on test PGD-10 accuracy during the training process. We use PGD-50-10 (50 iterations and 10 restarts with step size $\alpha = \epsilon/4$) to evaluate the robustness of trained model.

### 3.2.1 RS-FGSM suffers from catastrophic overfitting

In paper [31], the author trains the PreAct-ResNet18 model on CIFAR10 with $\epsilon = 8/255$ using cyclic learning rate, and finds using step size $\alpha = 1.25\epsilon$ leads to the best robust accuracy. And the model starts to suffer from catastrophic overfitting if further increasing the step size $\alpha$. We use the piecewise decay learning rate as suggested by paper[22] that can get the best robust accuracy among other learning rates. Here we run the experiment to check whether large step size $\alpha$ with fixed $\epsilon$ still suffers from catastrophic overfitting using piecewise decay learning rate and check whether this best ratio $(\alpha/\epsilon)$ 1.25 will change according to different $\epsilon$ or not.

**Different step-size $\alpha$ and fixed $\epsilon$.** As shown on the Figure 3.2, we train the model using RS-FGSM adversarial training over different ratio$(\alpha/\epsilon)$ with fixed $\epsilon$ on CIFAR10. We observe that for $\epsilon = 8/255$, catastrophic overfitting happens when $\alpha/\epsilon > 1$, while for $\epsilon = 16/255$, catastrophic overfitting happens when $\alpha/\epsilon > 0.5$. Based on experiment results, we would say catastrophic overfitting happens regardless of the learning rate schedule and the best ratio will change according to different $\epsilon$. However, it's hard to tune this ratio each time for different $\epsilon$ when we use RS-FGSM. We will use $\alpha = \epsilon$ in this project.

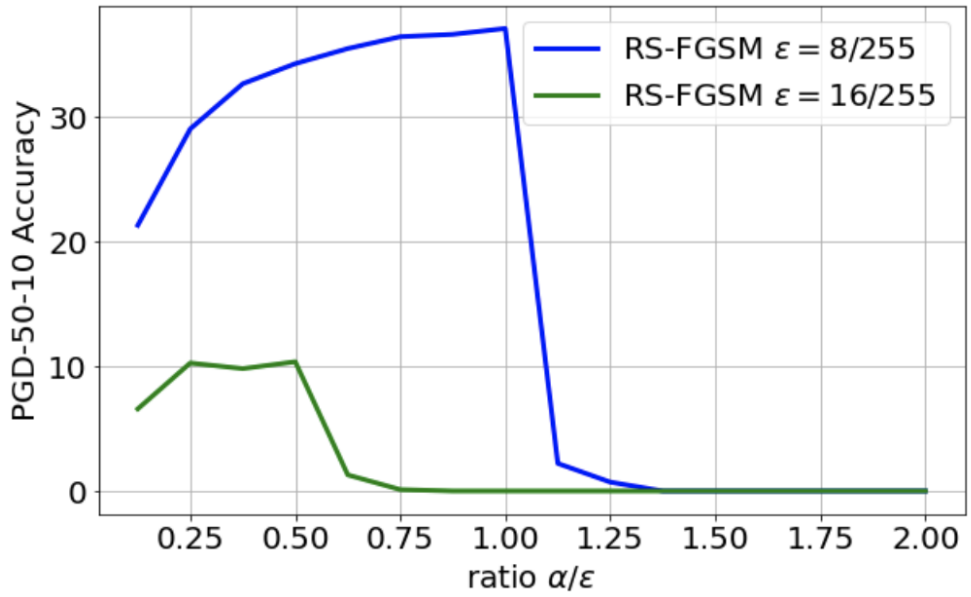
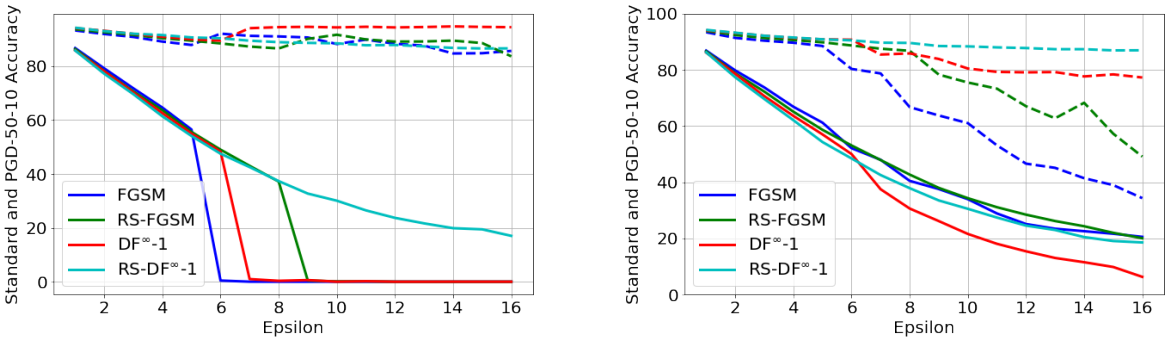



Figure 3.2 – PGD-50-10 accuracy of RS-FGSM adversarial training over different step-size $\alpha =$ ratio$\epsilon$ with fixed $\epsilon = 8/255$ or $\epsilon = 16/255$. We evaluate the PGD-50-10 accuracy of final model without early stopping.

In paper [1], the author finds RS-FGSM still suffer from catastrophic overfitting on CIFAR10 when $\epsilon > 9/255$ with fixed $\alpha = 1.25\epsilon$ and the model is trained by cyclic learning rate. Here we run the experiments over **different $\epsilon$ with fixed step size** $\alpha = \epsilon$ using piecewise decay learning rate. As shown on Figure 3.3, RS-FGSM starts to suffer from catastrophic overfitting when $\epsilon > 8/255$.

This is larger compared to vanilla FGSM which starts to suffer from catastrophic overfitting when $\epsilon > 5/255$. So we would say RS-FGSM does not avoid catastrophic overfitting for large $\epsilon$, but it does mitigate the catastrophic overfitting problem by extending to larger $\epsilon$ compared to vanilla FGSM.




(a) Final model

(b) Best model

Figure 3.3 – Standard accuracy (dashed line) and PGD-50-10 accuracy (solid line) from different adversarial training (AT) methods with different $\epsilon$. For RS-FGSM, the step-size $\alpha = \epsilon$. For DF$^\infty$-1, the overshoot $\eta$ is fixed to 0.02

Based on these two experiments, we could conclude that RS-FGSM still suffers from catastrophic overfitting, but extends the working range of $\epsilon$ compared to vanilla FGSM. And if we want to avoid catastrophic overfitting, we can decrease the step size $\alpha$, but this will leads to a sub-optimal solution, the perturbations we find using a smaller step size $\alpha$ will be in a smaller $l_\infty$-ball than the one used during the evaluation.

### 3.2.2 DF$^\infty$-1 suffers from catastrophic overfitting

In addition to vanilla FGSM and RS-FGSM, we also train the model using 1-iteration $l_\infty$ DeepFool method (DF$^\infty$-1, DF's superscript of $\infty$ means using $l_\infty$ norm to calculate perturbation in each iteration)[17]. The reason we use 1 iteration DeepFool instead of more iterations is because we want to make this method have comparable computational efficiency as one gradient step FGSM. The difference between FGSM and DF$^\infty$-1 is that FGSM always has the fixed step size $\alpha$ for all input examples while DF$^\infty$-1 will adapt the length of perturbation dynamically for each input example without being overly perturbed with decision boundary, as shown on Equation 3.1.

$$\delta_{DF^\infty-1} = (1+\eta)DF(x) \tag{3.1}$$

We can also add random initialization to DF$^\infty$-1 as shown on Equation 3.2.

$$\begin{aligned} \delta &\sim \mathcal{U}([-\epsilon,\epsilon]^d) \\ \delta_{RS-DF^\infty-1} &= \prod_{[-\epsilon,\epsilon]^d}(\delta + (1+\eta)DF(x+\delta)) \end{aligned} \tag{3.2}$$

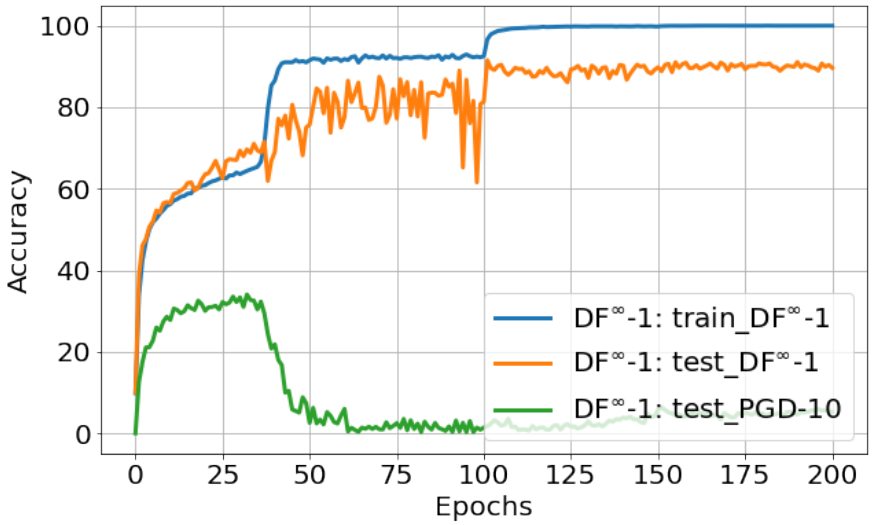



Figure 3.4 – Visualization of the training process regarding DF$^\infty$-1 adversarial training using PreAct-ResNet18 on CIFAR10 with $\eta$ = 0.02. Catastrophic overfitting happens around epoch 38.

As shown on Figure 3.4, the PreAct-ResNet18 model trained by DF$^\infty$-1 on CIFAR10 with $\epsilon = 8/255$ suffers from catastrophic overfitting. There is a decrease of PGD10 accuracy from robust to 0 on testing dataset and an increase of DF$^\infty$-1 accuracy on both training and testing dataset in few epochs. And we observe that the speed of decrease and increase is slower than FGSM. We will discuss the difference between DF$^\infty$-1 and FGSM in next chapter. As shown on Figure 3.3, when we decrease the $\epsilon$ to 6/255, DF$^\infty$-1 can train a robust model without suffering from catastrophic overfitting. When we add random initialization to DF$^\infty$-1, we can extend the $\epsilon$ to 16/255 at which there is no catastrophic overfitting happens but leads to sub-optimal robust accuracy.

In this chapter, we define the formal definition of catastrophic overfitting and empathize that this is overfitting between weak attacks and strong attacks intead of overfitting between training and testing dataset. Besides, we demonstrate that catastrophic overfitting is a general phenomenon in adversarial training and suffers not only by FGSM, RS-FGSM, but also by DF$^\infty$-1. In next chapter we will analysis catastrophic overfitting in geometric way and discuss the difference between FGSM and DF$^\infty$-1.

# Chapter 4

# Geometric analysis of catastrophic overfitting

In the last chapter, we find catastrophic overfitting happens not only in FGSM and but also in DF$^{\infty}$-1 adversarial training methods. And once it happens, robustness of the training model will drop to 0 in a few epochs. Thus it is important to analyze what happens before and after catastrophic overfitting. In this chapter, we first analyze the decision boundaries of the classifiers trained by FGSM or DF$^{\infty}$-1 before and after catastrophic overfitting separately. Then we compare the difference between them. We find although these two methods both cause catastrophic overfitting, they lead to totally different geometric properties after catastrophic overfitting.

## 4.1 Geometric analysis of FGSM adversarial training

We first introduce **DeepFool** [17] before going deeper into the geometric analysis of FGSM adversarial training.

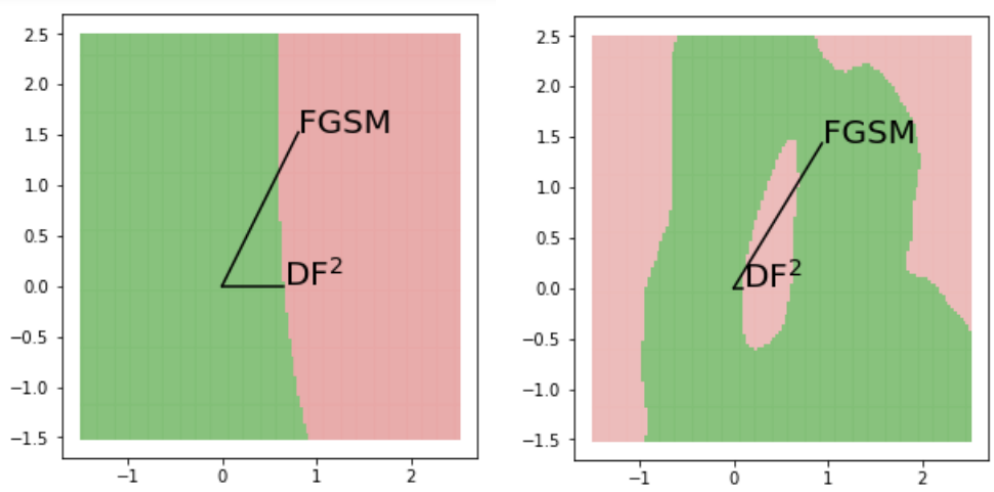
$$\Delta(x;\hat{k}) = \min_{r} \|r\|_2 \text{ subject to } \hat{k}(x+r) \neq \hat{k}(x) \tag{4.1}$$

DF$^2$ (DF's superscript of 2 means using $l_2$ norm to calculate perturbations in each iteration) is an algorithm used to find the smallest perturbations which can fool the deep networks based on iterative linearization of the classifier, as shown in equation 4.1. If the decision boundary is linear, DF$^2$ only needs 1 iteration to find the smallest perturbation, which is perpendicular to the decision boundary. The more complex and non-linear decision boundary is, the more iterations DF$^2$ are needed to find the valid perturbations. In this way, we can use the number of DF$^2$ iterations to estimate the complexity of the decision boundary and use $l_2$ norm of the DF$^2$ perturbations to estimate the robustness of the classifier. During the implementation, we set a maximum of 50 iterations to avoid $\infty$ loop. Based on the empirical results from the paper [17],

$DF^2$ converges in a few iterations (i.e., less than 3), thus 50 iterations should be enough to find the valid perturbation for almost all data points.

The differences between $DF^2$ we use to evaluate the robustness of classifiers here and $DF^\infty$-1 we use in the previous chapter to train the robust model can be summarized into the two following aspects: (1) the number of iterations is different. $DF^2$ either stops looping when finding the valid perturbation that can deceive the classifier or reaches the maximum of 50 iterations, while $DF^\infty$-1 always stops after 1 iteration no matter whether finding the valid perturbation or not; (2) $DF^2$ uses $l_2$ norm to calculate the perturbation in each iteration, while $DF^\infty$-1 uses $l_\infty$ norm to be coincident with the $l_\infty$ threat model we use in this project.

Now, we will introduce the way we visualize the decision boundary. Since it is impossible to visualize the decision boundary of a high-dimensional classifier, we draw the cross-section of the decision boundary spanned by two vectors. One vector is the perturbation vector calculated by $DF^2$, and the other is the perturbation vector calculated by the adversarial method used in the training process (FGSM or $DF^\infty$-1). We use green color to represent the true class and different red color to represent false classes. The origin represents the clean inputs.

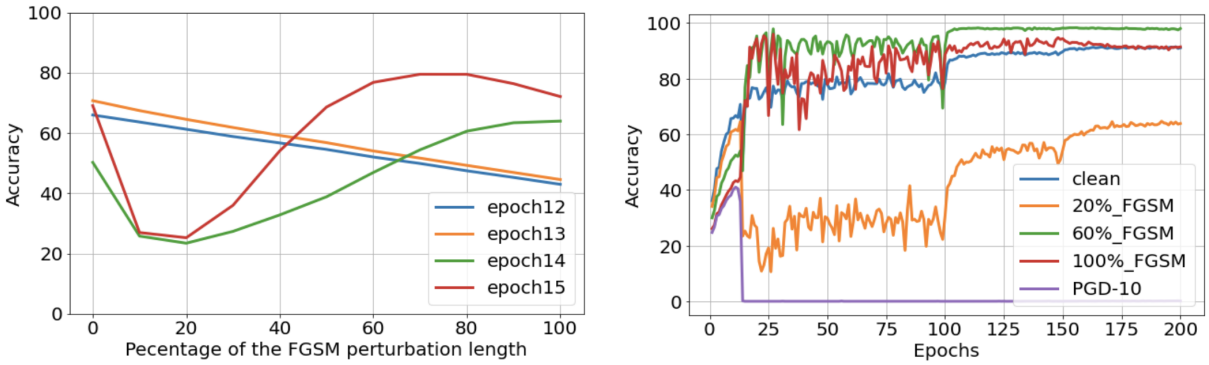



(a) Epoch 10 (b) Epoch 15

Figure 4.1 – Show the cross-section of the decision boundary spanned by two vectors. One is calculated by $DF^2$, and the other one is calculated by the FGSM adversarial method used in the training process. Green color represents the true class and the different red color represents false classes. The original point represents clean input. The classifier is trained by FGSM with $\epsilon = 8/255$ and catastrophic overfitting happens at epoch13. Here epoch10 and epoch15 are the snapshots of the classifier before and after catastrophic overfitting separately.

As shown in Figure 3.1, the PreAct-ResNet18 model trained by vanilla FGSM with $\epsilon = 8/255$ has catastrophic overfitting occurring at epoch 13. We draw the cross-sections of the decision

boundary of a specific input example at epoch10 and epoch15 separately. As shown in Figure 4.1, we observe that (1) small perturbation becomes more effective than the large one along the FGSM direction after catastrophic overfitting. At epoch 10, the small perturbation cannot find the effective adversarial example while the large perturbation can. At epoch15, small perturbation is more effective than large perturbation along the FGSM direction. (2) clean input becomes much close to the decision boundary after catastrophic overfitting. At epoch10, the smallest perturbation found by $DF^2$ that can fool the network is much larger than the one found at epoch15.




(a) (b)

Figure 4.2 – The model is PreAct-ResNet18 trained by vanilla FGSM with $\epsilon = 8/255$ and occurs catastrophic overfitting at epoch13 as shown on Figure 3.1 (a) The test accuracy of perturbed points generated by different length of FGSM perturbations.(b) The test accuracy of perturbed points generated by different length of FGSM perturbations or PGD-10 perturbations during the training process.

Apart from the qualitative analysis, we also add the quantitative analysis to support our observations. In order to verify that small perturbation is more effective than large perturbation after catastrophic overfitting, we do two experiments as shown in Figure 4.2. For a fixed epoch, we calculate the test accuracy of perturbed inputs generated by different lengths of FGSM perturbation. The result shown in Figure 4.2a is coincident with the observation we have by drawing the cross-section of the decision boundary. The curves of epoch12 and epoch13 show that before CO, large perturbation has smaller accuracy than small perturbation and thus it is more effective. And the curves of epoch14 and epoch15 show that after CO, we have the worst accuracy at about 20% of the FGSM perturbation length, and then the accuracy increases as the length of the perturbation increases, which means large perturbation becomes less effective than small perturbation. In order to make sure that small perturbation is more effective than large perturbation holds for all epochs after catastrophic overfitting, we calculate the accuracy of perturbed points generated by 20%, 60%, or 100% FGSM perturbation length. As shown in Figure 4.2b, after CO, the perturbed points generated by 20% FGSM perturbation length always have the smallest accuracy which means these perturbations are the most effective. For a robust

model, small perturbation is weaker than large perturbation, but after catastrophic overfitting, we have the counter-intuitive observations which reflect that new decision boundary may be generated around the perturbed points to make them be classified successfully.

In order to verify that the clean input becomes much near to the decision boundary after CO, we calculate the expected $l_2$ norm of DF$^2$ perturbations. As shown in Figure 4.8d, we observe that after CO happens, the expected $l_2$ norm of DF$^2$ perturbations decreases suddenly.

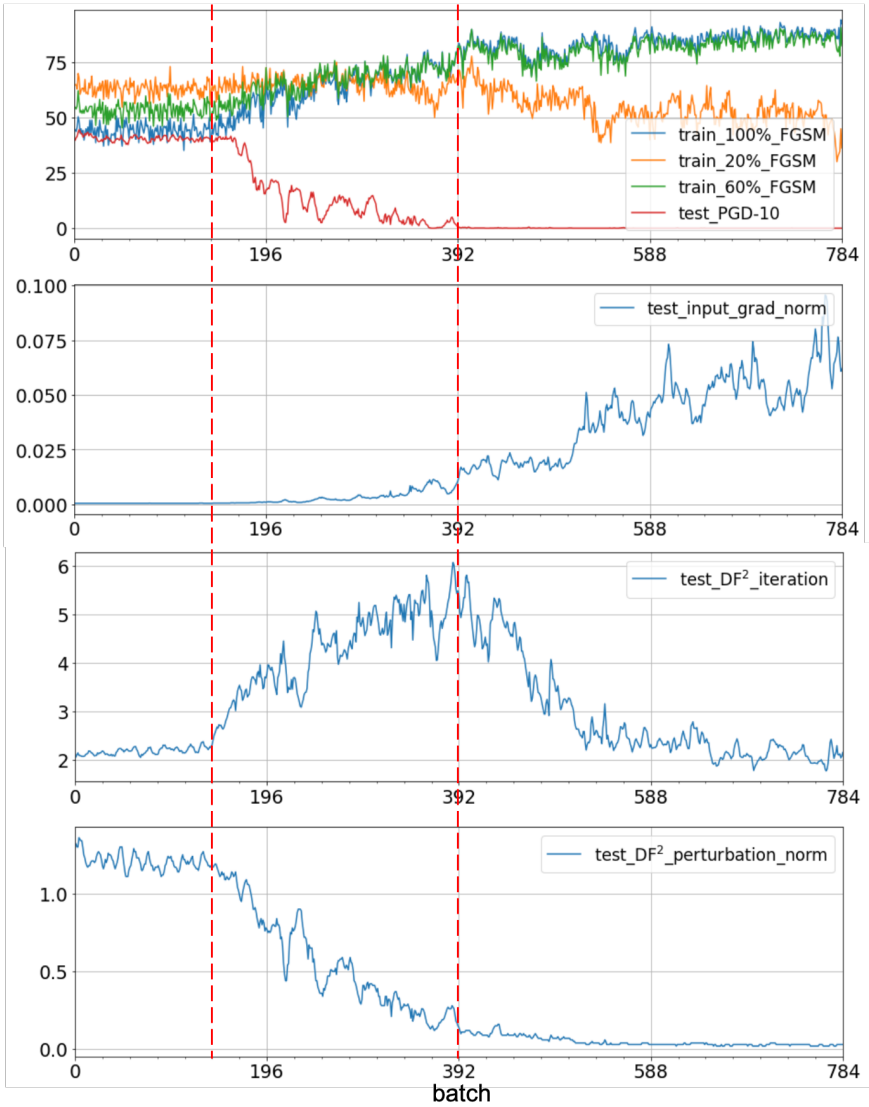
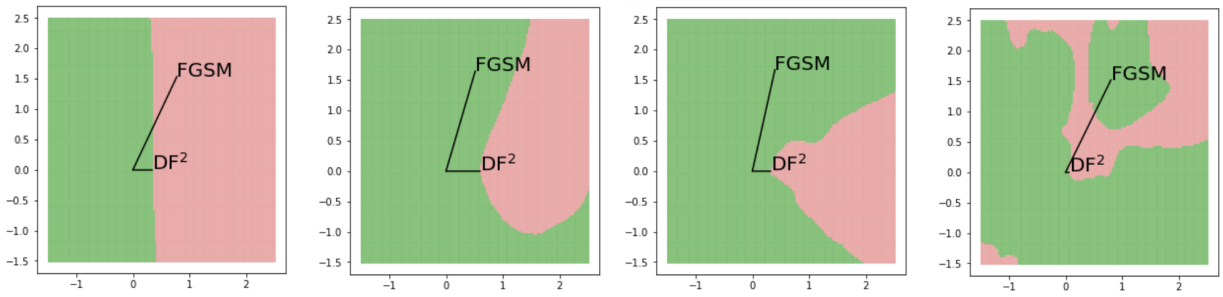



Figure 4.3 – We train PreAct-ResNet18 using vanilla FGSM on CIFAR10 with $\epsilon = 8/255$ and on figure 3.1 we show CO happens at epoch13. Thus, we resume the training from epoch12, train for another 4 epochs and then evaluate the model by batch.

Based on the previous analysis, we clearly demonstrate the geometric difference between

classifiers before and after catastrophic overfitting. For FGSM adversarial training, the catastrophic overfitting happens almost within one epoch. We want to check what happens in these particular epochs around catastrophic overfitting. We train PreAct-ResNet18 using vanilla FGSM on CIFAR10 with $\epsilon = 8/255$ and on Figure 3.1 we show CO happens at epoch13. Thus, we resume the model from epoch12 and train for another 4 epochs (the number of batches in one epoch is 196). Then, we evaluate the accuracy, $l_2$ norm of the input gradients, the number of DF$^2$ iterations, and $l_2$ norm of DF$^2$ perturbations by batch. The result is shown in Figure4.3. We observe that (1) From batch 0 to 140, the model is robust based on the test PGD10 accuracy. At that time, small FGSM perturbations are less effective than large FGSM perturbations. (2) From batch 140 to 392, the number of DF$^2$ iterations increases quickly and the $l_2$ norm of the DF$^2$ perturbations decreases correspondingly. During this stage, the model starts to lose robustness and the test PGD10 accuracy drops to zero. And we also notice that the accuracy of perturbed points constructed by different lengths of FGSM perturbation all increase and have almost the same accuracy. This gives us a clue that FGSM direction becomes ineffective at this stage.(3) From batch 392 to 784, the test PGD10 accuracy remains at 0; $l_2$ norm of the input gradients increases quickly; the accuracy of perturbed points generated by 20% of FGSM perturbation starts to decrease, which means small perturbations become more effective than large perturbations; new decision boundary is generated around the perturbed points.



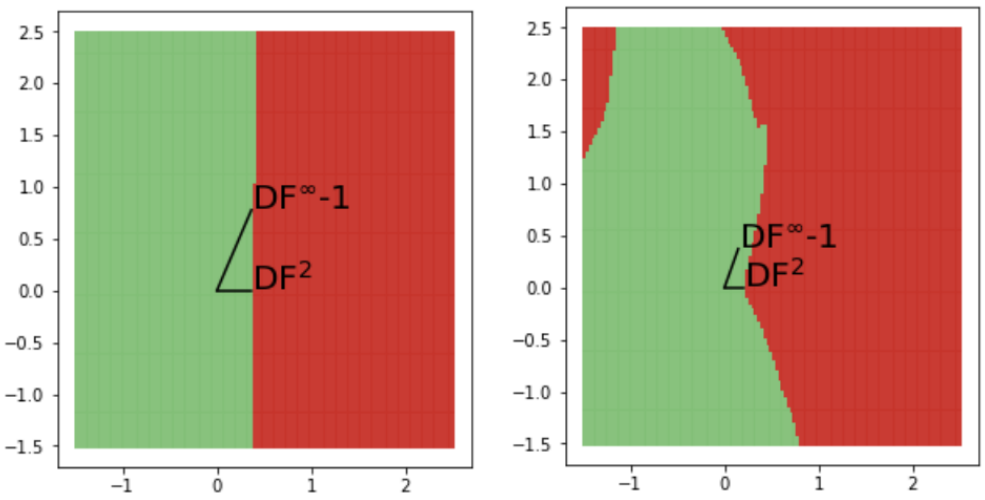

(a) Batch 60 (b) Batch 240 (c) Batch 360 (d) Batch 780

Figure 4.4 – Show the cross-section of the decision boundary spanned by perturbation vectors, which are calculated by DF$^2$ and FGSM adversarial method used in the training process at different batches.

We also draw the cross-sections of the decision boundaries spanned by perturbation vectors, which are calculated by DF$^2$ and FGSM adversarial method used in the training process at different batches as shown on Figure 4.4. The visualization of the decision boundary corresponds to our analysis. From the Figure 4.4b, 4.4c, we observe that at batch 240 and batch 360, all perturbations along the FGSM direction is ineffective. And at batch 60 before CO, large perturbation is more effective, while at batch 780 after CO, small perturbation is more effective.

## 4.2 Geometric analysis of DF$^{\infty}$-1 adversarial training

From the previous chapter we know catastrophic overfitting is not only limited to FGSM but also suffered by DF$^{\infty}$-1. Here we apply the same geometric analysis for DF$^{\infty}$-1 as we do for FGSM in the last section. As shown in Figure 3.4, DF$^{\infty}$-1 with $\epsilon = 8/255$ starts to occur CO at epoch 36, which means the test PGD-10 accuracy starts to decrease. And at around epoch 65, the accuracy decreases to 0. As shown on Figure 4.5, we draw the cross-sections of the decision boundary of a specific input example at epoch 36 and epoch 70 separately. And we also add the quantitative analysis on Figure 4.6 4.7 to support our analysis.




(a) Epoch 36 (b) Epoch 70

Figure 4.5 – Show the cross-section of the decision boundary spanned by two vectors. One is calculated by DF$^2$ and the other one is calculated by DF$^{\infty}$-1 used in the training process. The classifier is trained by DF$^{\infty}$-1 with $\epsilon = 8/255$. Epoch36 and epoch70 are the snapshots of the classifier before and after CO separately.

The expected $l_2$ norm of the perturbation calculated by DF$^{\infty}$-1 during the training process becomes smaller after CO, as shown in Figure 4.6. This is different from FGSM, whose perturbation length will not change during the training process once the step size is fixed.

As shown in Figure 4.7a, 4.7b, large percentage of DF$^{\infty}$-1 perturbation length is always more effective than small percentage of DF$^{\infty}$-1 perturbation length no matter CO happens or not. But after CO happens, perturbations generated by DF$^{\infty}$-1 become smaller, thus still lose their effectiveness. Besides, the clean input becomes much near to the decision boundary after CO. As shown in Figure 4.5, the smallest perturbation found by DF$^2$ which can fool the network at epoch 36 is larger than the one found at epoch 70. This can also be reflected on Figure 4.8d that after CO happens, the expected $l_2$ norm of DF$^2$ perturbations decreases.

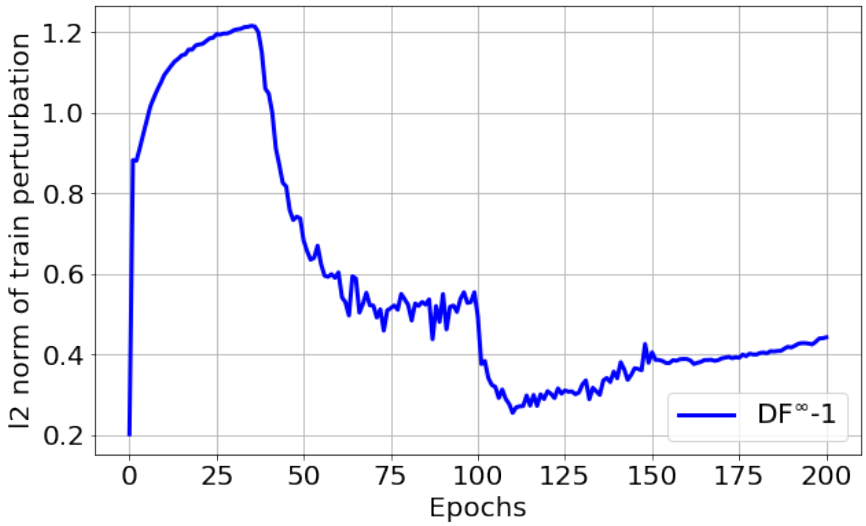
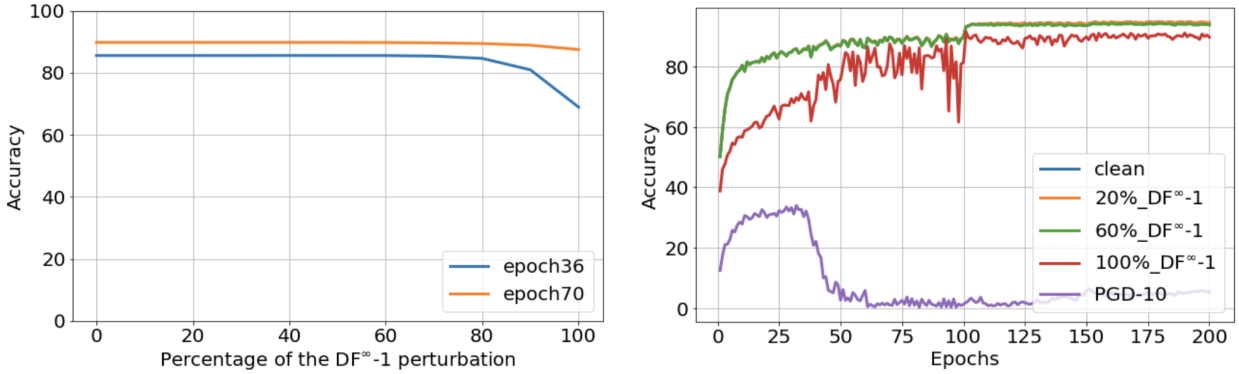


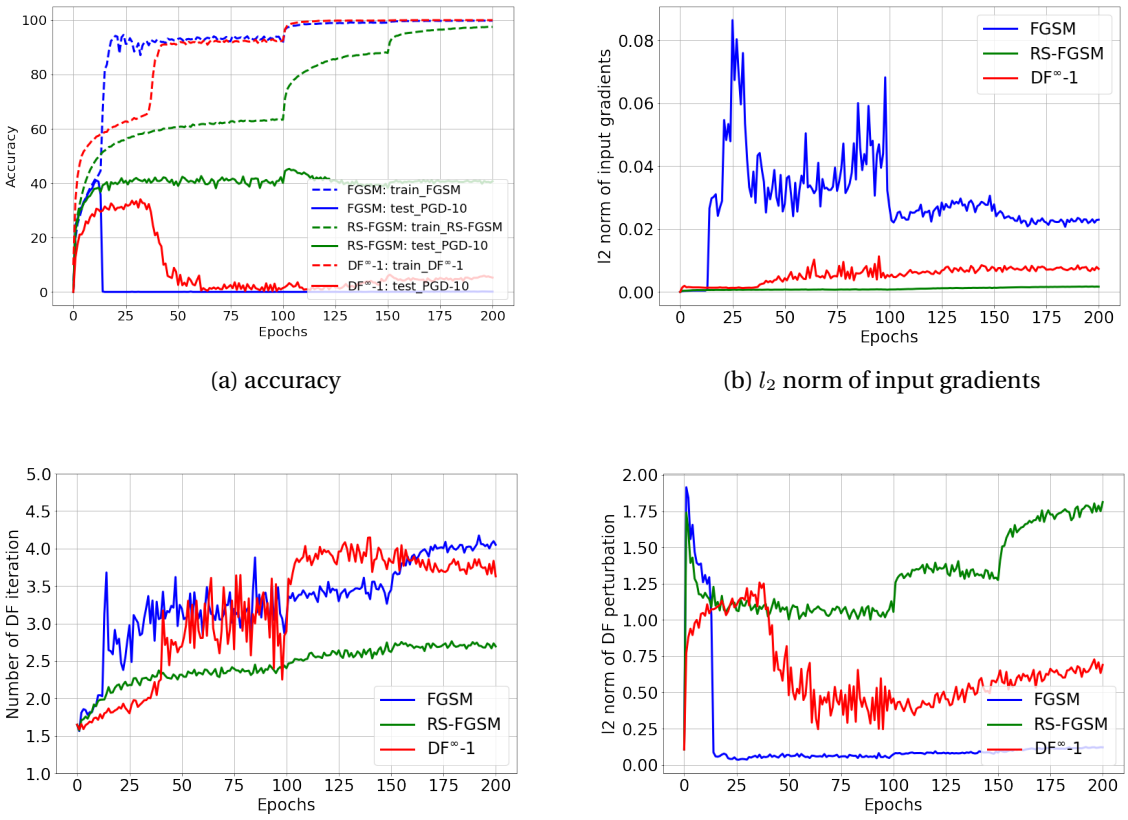

Figure 4.6 – The expected $l_2$ norm of perturbation calculated by DF$^\infty$-1 during the training process




(a) (b)

Figure 4.7 – The model is PreAct-ResNet18 trained by DF$^\infty$-1 with $\epsilon = 8/255$ and starts to suffer from CO at epoch 36 as shown in Figure 3.4. (a) The test accuracy of perturbed points generated by different length of DF$^\infty$-1 perturbations. (b) The test accuracy of perturbed points generated by different length of DF$^\infty$-1 perturbations or PGD-10 perturbations during the training process.

## 4.3 The difference between FGSM and DF$^\infty$-1




(a) accuracy

(b) $l_2$ norm of input gradients

(c) number of DF$^2$ iterations

(d) $l_2$ norm of DF$^2$ perturbations

Figure 4.8 – Different observations during the training process, $\epsilon = 8/255$

Based on previous analysis, we observe that although both FGSM and DF$^\infty$-1 adversarial training suffer from catastrophic overfitting, they have different geometric properties after catastrophic overfitting. We further compare FGSM, DF$^\infty$-1, RS-FGSM (not suffer from CO) under different perspectives. As shown in Figure 4.8a, once CO happens, there is a sudden drop in PGD10 accuracy on testing dataset and a sudden increase in FGSM or DF$^\infty$-1 accuracy on training dataset. But PGD10 accuracy decreases much faster in FGSM compared to DF$^\infty$-1. This might because FGSM perturbation length will not decrease since the step size is fixed, while DF$^\infty$-1 perturbation length will decrease when CO happens as shown in Figure 4.6, and decreasing perturbation length can slow down the process of CO.

As shown on Figure 4.8b, there is a suddenly increase of the expected $l_2$ norm of input gradients for both FGSM and DF$^\infty$-1. However, FGSM is almost an order of magnitude larger than DF$^\infty$-1. As shown on Figure 4.8c 4.8d, The number of DF$^2$ iterations needed to calculate

the smallest perturbations fooling the network also increases while the expected $l_2$ norm of DF$^2$ perturbations decreases. And the $l_2$ norm of DF$^2$ perturbations of the model trained by FGSM is much smaller than the model trained by DF$^\infty$-1 after CO. This might be because FGSM generates a new decision boundary near the clean inputs while DF$^\infty$-1 do not.

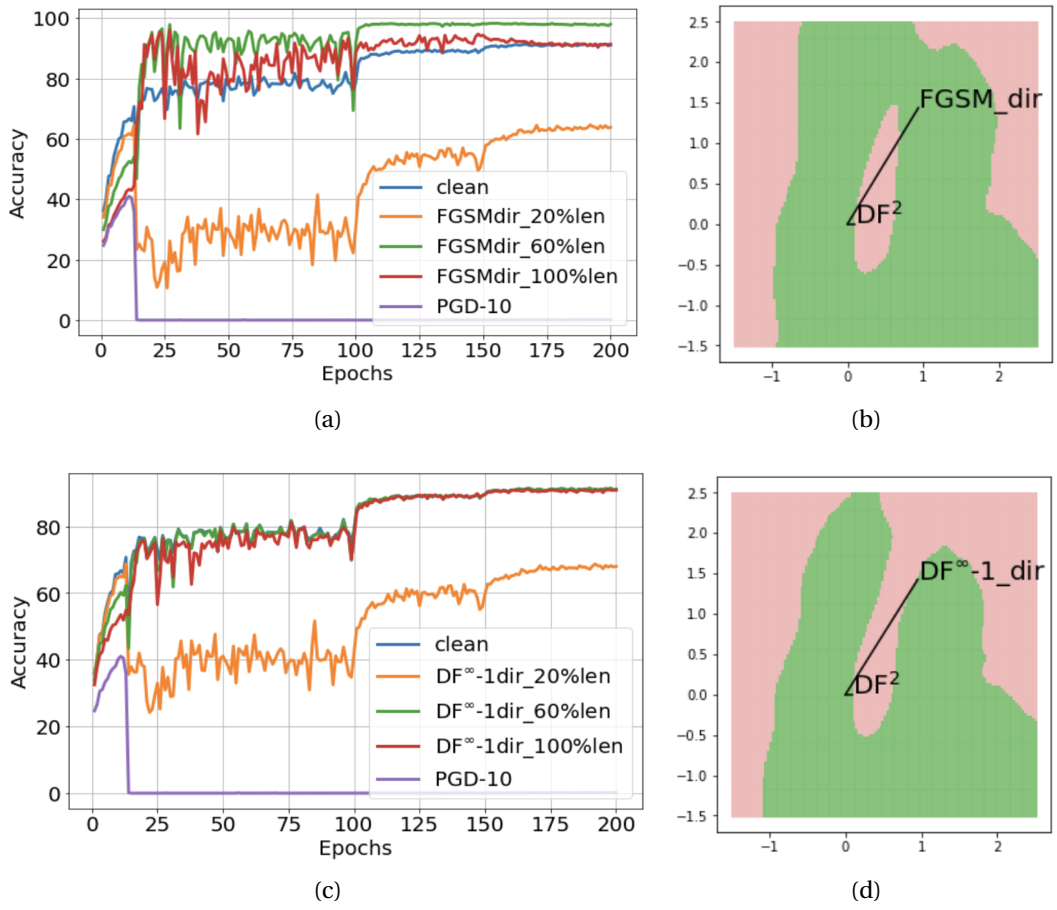



Figure 4.9 – Cross-sections of the decision boundaries and robust accuracy under different percentage of FGSM perturbation length along FGSM or DF$^\infty$-1 direction on the model trained by FGSM. (a) were tested along FGSM direction while (c) were tested along DF$^2$direction.

We draw the cross-sections of the decision boundaries and calculate robust accuracy with different lengths of perturbations along FGSM or DF$^\infty$-1 direction of the model trained by FGSM or DF$^\infty$-1. As shown in Figure 4.9, for **FGSM**, after CO happens, the small perturbation along FGSM and DF$^\infty$-1 direction becomes more effective than the large one, and new decision boundary is generated, which we can observe from Figure 4.9b, 4.9d intuitively. As shown on Figure 4.10, for **DF$^\infty$-1**, after CO happens, small perturbation along FGSM and DF$^\infty$-1 direction is still less effective than the large one.

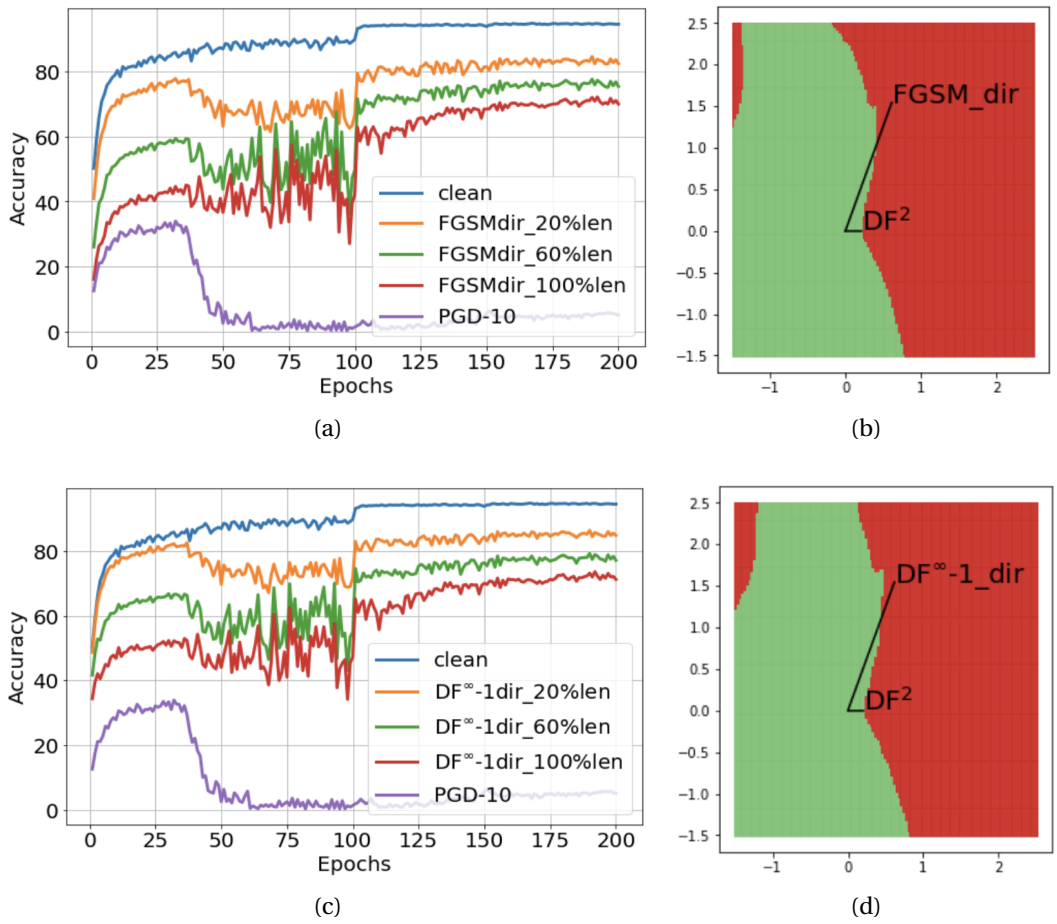



Figure 4.10 – Cross-sections of the decision boundaries and robust accuracy under different percentage of FGSM perturbation length along FGSM or DF$^{\infty}$-1 direction on model trained by DF$^{\infty}$-1. (a) were tested along FGSM direction while (c) were tested along DF$^{2}$direction.

In this chapter, we first analyze the geometric properties of classifiers trained by FGSM and DF$^{\infty}$-1 before and after CO separately. And we find FGSM generates a new decision boundary both along the FGSM direction and DF$^{\infty}$-1 direction, which makes small perturbation become less effective than large perturbation. As for DF$^{\infty}$-1, the marginal between clean inputs and decision boundary becomes smaller but is still larger than FGSM after CO. And DF$^{\infty}$-1 does not generate a new decision boundary and large perturbation is still more effective than small perturbation. But since the perturbation generated by DF$^{\infty}$-1 becomes smaller after CO and thus loses its effectiveness. The geometric analysis demonstrates the decision boundaries after CO and shows why adversary methods used in training lose their effectiveness, but why CO happens still needs more exploration. In the next chapter, we will focus on analyzing factors that cause CO.

# Chapter 5

# Analysis on factors causing catastrophic overfitting

In the last chapter, we find geometric properties of the classifier change after CO and make both FGSM and DF$^\infty$-1 adversarial methods become ineffective. This explains why the model can not recover by itself once CO happens. The difficulty to bring the model back to robust after CO happens increases the importance of preventing the occurrence of CO. In this chapter, we will focus on analyzing factors that cause CO. Specifically, we design experiments to examine three probable hypotheses, two of them come from previous literature and one of them proposed by us. And finally, we make a small modification to RS-FGSM which improve its performance.

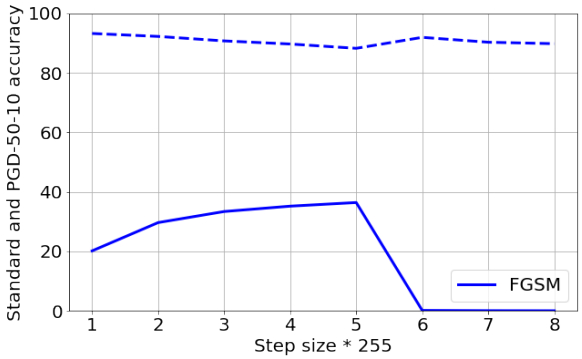
## 5.1 Probable hypotheses on factors causing catastrophic overfitting

### 5.1.1 Hypothesis: large perturbation causes catastrophic overfitting

**Hypothesis** The length of perturbation is evaluated by $l_2$ norm. Large perturbation causes CO while small perturbation can avoid CO.

This hypothesis is first proposed in paper [1]. The author claims that RS-FGSM helps avoid CO by decreasing the expected $l_2$ norm of perturbations. And the paper provides theoretical proof that RS-FGSM always has a smaller expected $l_2$ norm compared to vanilla FGSM. We also find another support phenomenon that for a fixed $\epsilon$, reducing the step size $\alpha$ can avoid CO. As shown in Figure 5.1, we train PreAct-ResNet18 on CIFAR10 using vanilla FGSM. We observe that for the fixed $\epsilon = 8/255$, we can avoid catastrophic overfitting when we reduce the step size to $\alpha \leq 5/255$. And smaller step size leads to smaller perturbations. Though reducing the step size $\alpha$ can avoid CO, it will lead to the sub-optimal solution since using a smaller step size is equivalent to training the model with perturbations from a smaller $l_\infty$ ball than the one used during the

test.




(a)

| step size | final standard / PGD-50-10 | l2 norm |
|---|---|---|
| 1/255 | 93.21 / 20.13 | 0.209 |
| 2/255 | 92.22 / 29.64 | 0.418 |
| 3/255 | 90.71 / 33.36 | 0.626 |
| 4/255 | 89.67 / 35.16 | 0.833 |
| 5/255 | 88.22 / 36.38 | 1.031 |
| 6/255 | 91.91 / 0.11 | 1.251 |
| 7/255 | 90.27 / 0.03 | 1.458 |
| 8/255 | 89.78 / 0 | 1.666 |

(b)

Figure 5.1 – We train PreAct-ResNet18 on CIFAR10 using FGSM with different step size $\alpha$ and fixed $\epsilon = 8/255$.

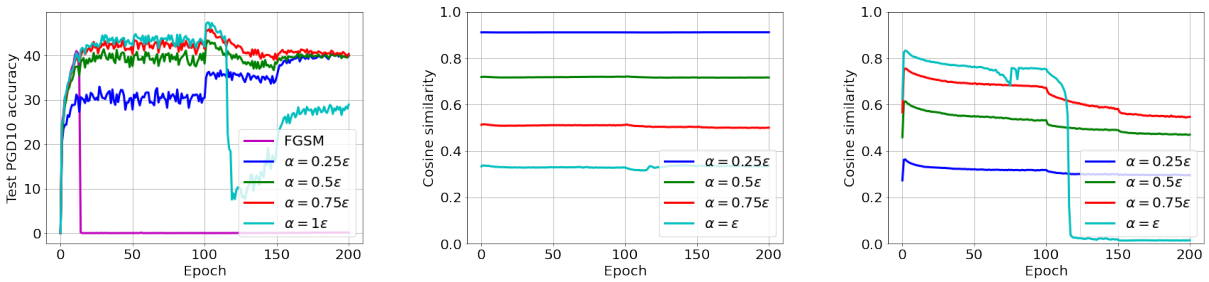
$$
\begin{aligned}
\delta_{FGSM} &= \epsilon \text{sign}(\nabla_x l(x, y; \theta)) \\
\delta &\sim \mathcal{U}([-\epsilon, \epsilon]^d) \\
\delta_{RS-FGSM} &= \prod_{[-\epsilon,\epsilon]^d}(\delta + \alpha \text{sign}(\nabla_x l(x + \delta, y; \theta))) \\
\delta_{magnified} &= \frac{\|\delta_{FGSM}\|_2}{\|\delta_{RS-FGSM}\|_2}\delta_{RS-FGSM}
\end{aligned}
\tag{5.1}
$$

We design the **Magnified-RS-FGSM** method to examine this hypothesis. As shown in Equation 5.1, we first compute $\delta_{FGSM}$ using vanilla FGSM, then we choose a step size $\alpha$ and calculate the perturbation $\delta_{RS-FGSM}$ using RS-FGSM, and finally magnify it by $\frac{\|\delta_{FGSM}\|_2}{\|\delta_{RS-FGSM}\|_2}$ to make the $l_2$ norm of this perturbation same as the perturbation calculated by vanilla FGSM $\delta_{FGSM}$.

As shown in Figure 5.2a, when step size $\alpha \leq 0.75\epsilon$ and $\epsilon = 8/255$, we can magnify the perturbation to the same $l_2$ norm as the perturbation calculated by vanilla FGSM without suffering from CO. This experiment result shows that $l_2$ norm of perturbation is not the only factor that decides whether CO happens or not, but the direction of the perturbation is also important.

Thus, we calculate the cosine similarity between perturbation vector $\delta_{RS-FGSM}$ and random initialized vector $\delta$, and the cosine similarity between perturbation vector $\delta_{RS-FGSM}$ and sign of the input gradient vector $\text{sign}(\nabla_x l(x, y; \theta))$. The results are shown in Figure 5.2b 5.2c, the direction of perturbation vector $\delta_{RS-FGSM}$ computed from smaller step size $\alpha$ is closer to the random initialized direction and farther away from the direction of sign of the input gradient. And we find perturbation computed by smaller step size can be magnified to the same $l_2$ norm as the perturbation calculated by vanilla FGSM while larger step size can not. Based on this

phenomenon, we come up with the hypothesis that perturbation computed from smaller step size is closer to the random initialized direction and can be magnified to a larger perturbation without suffering from CO.




(a) (b) (c)

Figure 5.2 – The PreAct-ResNet18 trained by **Magnified-RS-FGSM** magnified from RS-FGSM with different step size $\alpha$ and fixed $\epsilon = 8/255$. (a) Show the test PGD10 accuracy during the training process with vanilla FGSM and Magnified-RS-FGSM magnified from RS-FGSM with different step size $\alpha$. (b)The cosine similarity between $\delta_{RS-FGSM}$ and random initialized $\delta$. (c) The cosine similarity between $\delta_{RS-FGSM}$ and sign of the input gradient sign$(\nabla_x l(x, y; \theta))$

| | final<br>standard / PGD-50-10 | best<br>standard / PGD-50-10 |
|---|---|---|
| | $\epsilon$ = 8 / 255 | |
| $\alpha = 0.25\epsilon$ | 90.03 / 34.23 | 89.88 / 34.05 |
| $\alpha = 0.5\epsilon$ | 87.06 / 36.36 | 86.66 / 40.94 |
| $\alpha = 0.75\epsilon$ | 85.41 / 36.4 | 85.54 / 43.8 |
| $\alpha = \epsilon$ | 90.17 / 10.09 | 84.42 / 45.35 |
| FGSM | 91.07 / 0 | 66.72 / 40.46 |
| | $\epsilon$ = 12 / 255 | |
| $\alpha = 0.25\epsilon$ | 86.83 / 19.19 | 86.77 / 19.15 |
| $\alpha = 0.5\epsilon$ | 83.18 / 18.96 | 83.11 / 25.02 |
| $\alpha = 0.75\epsilon$ | 87.79 / 0.04 | 70.8 / 30.11 |
| $\alpha = \epsilon$ | 88.55 / 0 | 58.11 / 29.69 |
| FGSM | 88.35 / 0 | 46.59 / 25.09 |

Table 5.1 – Show the standard and PGD-50-10 accuracy of the final and best model trained by Magnified-RS-FGSM magnified from RS-FGSM with different step size $\alpha$ under $\epsilon = 8/255$ or $\epsilon = 12/255$. The gray background of the cell indicates that CO happens under this settings.

In order to verify our new hypothesis, we run the same experiments with $\epsilon = 12/255$ and then evaluate the robustness of the best (selected by PGD-10) and final model using PGD-50-10.

The result is shown on Table 5.1. We observe that when we need to magnify the perturbation to the same as vanilla FGSM with $\epsilon = 12/255$, $\alpha = 0.25\epsilon$ and $\alpha = 0.5\epsilon$ can still avoid catastrophic overfitting while $\alpha = 0.75\epsilon$ suffers from catastrophic overfitting. This supports the conclusion that perturbation computed from smaller step size can be magnified to a larger perturbation without suffering from CO. Another observation is that all step sizes $\alpha \leq 0.75\epsilon$ can avoid CO when $\epsilon = 8/255$. $\alpha = 0.75\epsilon$ has the best robust accuracy of 43.8% which is 9.75% better than $\alpha = 0.25\epsilon$, even if they have the same $l_2$ norm as the perturbation calculated by vanilla FGSM. This indicates that under the same perturbation length if the direction of perturbation is closer to the random initialized vector, the robust accuracy will become smaller than the one farther away from the random initialized vector and closer to the sign of the input gradient vector. This means even if by choosing direction we can get the same perturbation length as vanilla FGSM, we still get the sub-optimal solution.

In this section, we design the experiments to examine the hypothesis that large perturbation can cause CO. And finally we conclude that not only $l_2$ norm of the perturbation matters but also the direction of the perturbation.

### 5.1.2 Hypothesis: perturbation should span the entire model

The paper[31] claims that RS-FGSM helps avoid CO by distributing perturbation features into the entire threat model $[-\epsilon, \epsilon]^d$; in other words, each dimension can take value between $[-\epsilon, \epsilon]$. There are two evidences to support this hypothesis. One is that if we choose the step size $\alpha = 2\epsilon$ to force each dimension to be $\{-\epsilon, \epsilon\}$, RS-FGSM also suffers from catastrophic overfitting when $\epsilon = 8/255$.

$$\begin{aligned} \delta &\sim \{-\frac{\epsilon}{2}, \frac{\epsilon}{2}\}^d \\ \delta_{R+FGSM} &= \textstyle\prod_{[-\epsilon,\epsilon]^d}(\delta + \frac{\epsilon}{2}\text{sign}(\nabla_x l(x+\delta, y; \theta))) \end{aligned} \tag{5.2}$$

The other is that R+FGSM from [28], as shown in Equation 5.2, cannot train the robust model on MNIST dataset as shown on [31] appendix-A while RS-FGSM is capable. The difference between R+FGSM and RS-FGSM is that R+FGSM can only generate perturbation with features on the $l_\infty$ ball $\{-\epsilon, 0, \epsilon\}^d$ while RS-FGSM can take value between $[-\epsilon, \epsilon]$.

$$\begin{aligned} \delta &\sim \{-\epsilon, \epsilon\}^d \\ \delta_{Boundary-RS-FGSM} &= \textstyle\prod_{[-\epsilon,\epsilon]^d}(\delta + \alpha\text{sign}(\nabla_x l(x+\delta, y; \theta))) \end{aligned} \tag{5.3}$$

we design the following **Boundary-RS-FGSM** method to examine the hypothesis that perturbation should span the entire model; in other words, each dimension of the perturbation can take value between $[-\epsilon, \epsilon]$. As shown in Equation 5.3, when we perform the random initialization, instead of taking values uniformly distributed from $[-\epsilon, \epsilon]^d$, we randomly choose either $-\epsilon$ or $\epsilon$ for each dimension of the perturbation. In this way, if the step size $\alpha = \epsilon$, the final perturbation generated will have features on $\{-\epsilon, 0, \epsilon\}$; if the step size $\alpha = 1.5\epsilon$, the final perturbation gener-

ated will have features on $\{-\epsilon, -0.5\epsilon, 0.5\epsilon, \epsilon\}$. They are discrete values and not span inside the $l_\infty$ ball.

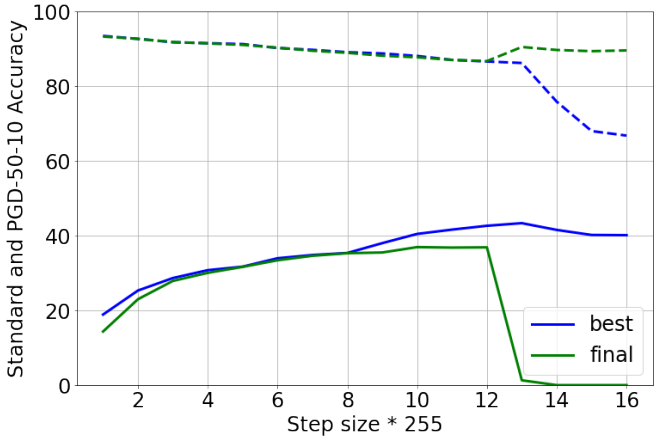



Figure 5.3 – Standard and PGD-50-10 accuracy of **Boundary-RS-FGSM** over different step size $\alpha$ with fixed $\epsilon = 8/255$. We evaluate both the best model (selected by PGD-10 during the training process) and the final model.

| method | best<br>standard / PGD-50-10 |
|---|---|
| FGSM | 66.72 / 40.46 |
| RS-FGSM | 86.77 / 42.69 |
| Boundary-RS-FGSM | 87.03 / 42.72 |

Table 5.2 – Comparison of the standard and robust performance on CIFAR10 with $\epsilon = 8/255$ between vanilla FGSM, RS-FGSM and Boundary-RS-FGSM. We use the step size $\alpha = \epsilon$ for RS-FGSM, and $\alpha = 1.5\epsilon$ for Boundary-RS-FGSM.

As shown on the Figure 5.3, we train the PreAct-ResNet18 model using **Boundary-RS-FGSM** adversarial training with fixed $\epsilon = 8/255$ and different step size $\alpha$ on CIFAR10. We observe that catastrophic overfitting happens when step-size $\alpha \geq 1.5\epsilon$. So when $\alpha = 1.5\epsilon$, we can get the best robust accuracy without suffering from CO.

We also compare the standard and PGD-50-10 performance between RS-FGSM and Boundary-RS-FGSM. As shown in Table 5.2, they can achieve almost the same performance on CIFAR10 with $\epsilon = 8/255$. The experiment results invalidate this hypothesis that each dimension of the perturbation should span in the entire threat model.

We also apply the Boundary-RS-FGSM to train the robust model on MNIST dataset. We use the same setting as the paper [31]. And based on the results on the Table 5.3, Boundary-RS-FGSM

also achieves almost the same robust accuracy as RS-FGSM.

| Method | Step size | PGD-50-10 |
|---|---|---|
| R+FGSM | $0.5\epsilon$ | 31.83±25.36% |
| RS-FGSM | $\epsilon$ | 86.11±1.4% |
| Boundary-RS-FGSM | $\epsilon$ | 80.64±2.22% |
| Boundary-RS-FGSM | $1.5\epsilon$ | 85.44±2.1% |

Table 5.3 – Show the performance of R+FGSM [28], RS-FGSM [31] and Boundary-RS-FGSM on MNIST dataset over 5 random seeds.

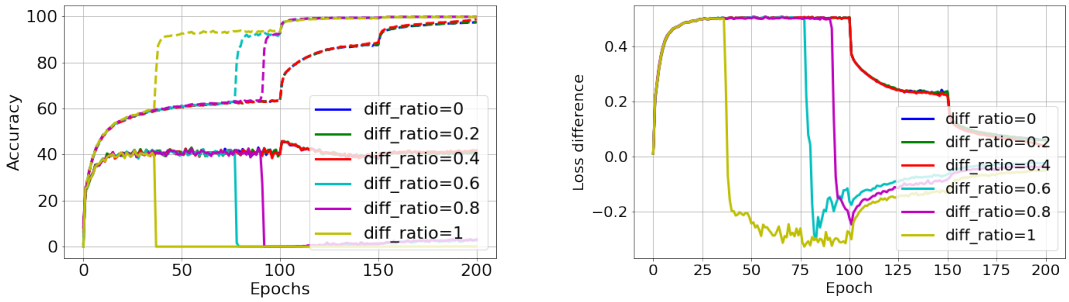
### 5.1.3 Hypothesis: large diversity of perturbations can avoid catastrophic overfitting

$$Diversity = 1 - cos(\delta_a, \delta_b) \tag{5.4}$$

We define the diversity metric of perturbation as Equation 5.4. We compute perturbations twice using the same input and model. $\delta_a$ is the first one and $\delta_b$ is the second one.

$$\begin{aligned} \delta_1, \delta_2 &\sim \mathcal{U}([-\epsilon, \epsilon]^d) \\ \delta &= (1-t)\delta_1 + t\delta_2, \text{ where } t \in [0,1] \\ \delta_{Diff-RS-FGSM} &= \prod_{[-\epsilon,\epsilon]^d}(\delta_1 + \alpha \text{sign}(\nabla_x l(x+\delta, y; \theta))) \end{aligned} \tag{5.5}$$

As for the same input and model, FGSM perturabtions are always the same, it has zero diversity. But for the RS-FGSM, since we have random initialization step, $\delta_a$ and $\delta_b$ will be different and has positive diversity. Thus, we can have the hypothesis that large diversity of the perturbation can avoid CO. In order to examine this hypothesis, we design the following **Diff-RS-FGSM** method, as shown in Equation 5.5.




(a) (b)

Figure 5.4 – Train the PreAct-ResNet18 model using **Diff-RS-FGSM** over different diff-ratio $t$ on CIFAR10 with fixed $\epsilon = 8/255$ and step size $\alpha = \epsilon$. (a) show standard and PGD-50-10 accuracy during the training process. (b) show the loss difference between train_fgsm_loss - train_clean_loss

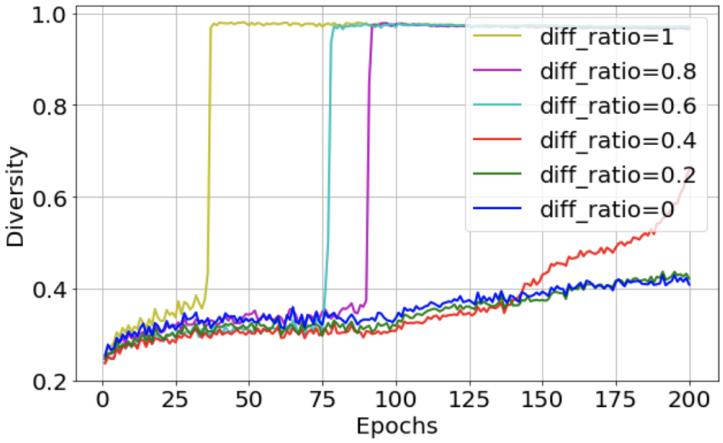


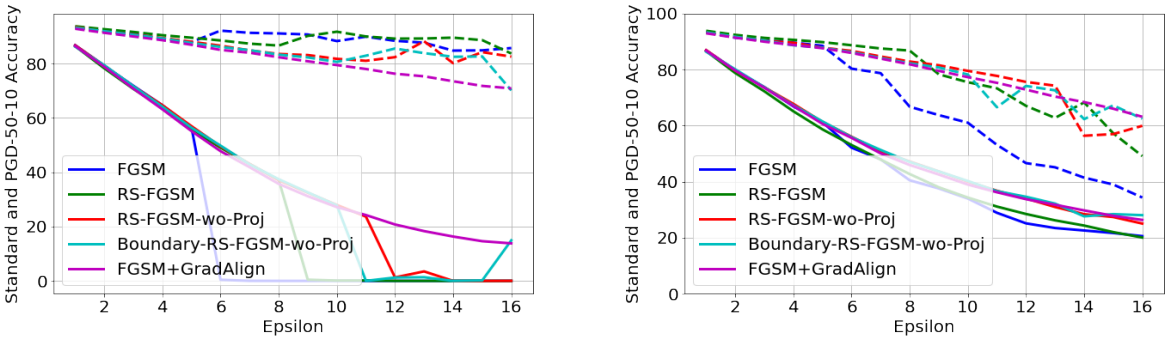

Figure 5.5 – Diversity of the Diff-RS-FGSM under different diff-ratio $t$

When diff-ratio $t = 0$, Diff-RS-FGSM is the same as RS-FGSM and does not suffer from catastrophic overfitting. As shown in Figure 5.4 the larger diff-ratio $t$, the earlier catastrophic overfitting happens. Another phenomenon we notice is that before catastrophic overfitting happens, Diff-RS-FGSM training with diff-ratio $t$ has almost the same accuracy and loss difference between perturbed points and clean points.

As shown in Figure 5.5, Diff-RS-FGSM with different diff-ratio $t$ has almost the same diversity before CO. But the larger the diff-ratio $t$ and the earlier the catastrophic overfitting happens. And once CO happens, the diversity will increased to almost 1. So we reject this hypothesis by saying that large diversity cannot guarantee to avoid CO.

## 5.2 Further improvement on RS-FGSM methods

In the previous section, we empirically analyze three hypotheses on factors causing catastrophic overfitting. For each hypothesis we comp up with counter experiments to show that none of these hypothesis can fully explain why CO happens. But these analyses do shed light on understanding why catastrophic overfitting happens. Based on the Section 5.1.1, not only $l_2$ norm will affect CO, but also the direction of perturbation. If the direction of perturbation is closer to the random initialized vector, the robust accuracy will become smaller than the one far away from the random initialized vector and closer to the sign of the input gradient vector. So we try the RS- FGSM without projecting back to $l_\infty$-ball to follow the sign of the input gradient vector as much as possible. We name the RS-FGSM and Boundary-RS-FGSM without projecting back to **RS-FGSM-wo-Proj** and **Boundary-RS-FGSM-wo-Proj** separately.




(a) Final (b) Best

Figure 5.6 – Standard accuracy (dashed line) and PGD-50-10 accuracy (solid line) from different FGSM methods with different $\epsilon$. (a) Results of the final model showing whether CO happens or not. (b) Results of the best model selected by PGD-10 during the training process.

As shown in Figure 5.6, we observe that RS-FGSM-wo-proj further extends the $\epsilon$ above which CO happens. RS-FGSM extends the working regime of $\epsilon$ from 5/255 to 8/255 compared to vanilla FGSM. And RS-FGSM-wo-Proj further extends the $\epsilon$ to 11/255. Another benefit is that given a fixed $\epsilon$ with which both RS-FGSM and RS-FGSM-wo-Proj do not have catastrophic overfitting, RS-FGSM-wo-Proj still has better robust PGD-50-10 accuracy and even comparable to FGSM+GradAlign[1].

As shown on Table 5.4, when $\epsilon = 8/255$, Both RS-FGSM-wo-Proj and Boundary-RS-FGSM-wo-proj have comparable performance as FGSM+GradAlign with better computational efficiency. When $\epsilon = 16/255$, the average performance of RS-FGSM-wo-Proj and Boundary-RS-FGSM-wo-proj is comparable to FGSM+GradAlign, but has larger standard deviation.

In this chapter, we focus on analyzing factors that cause CO. Specifically, we design experi-

ments to examine three probable hypotheses. Although finally, we find none of them can fully explain why CO happens but the analysis process and results of designed experiments do improve the understanding toward catastrophic overfitting. Then in the second section, we make a small modification to RS-FGSM by not projecting back to $l_\infty$ thread model and gain further improvements on RS-FGSM.

| Method | Accuracy | | Attack |
|---|---|---|---|
| | Standard | Robust | |
| $\epsilon$=8 / 255, step_size = $\epsilon$ | | | |
| FGSM | 69.21±2.18% | 41.19±0.38% | PGD-50-10 |
| RS-FGSM | 86.35±0.34% | 43.57±0.30% | PGD-50-10 |
| RS-FGSM-wo-proj | 82.66±0.56% | 47.56±0.37% | PGD-50-10 |
| Boundary-RS-FGSM-wo-proj | 82.29±0.46% | 47.65±0.52% | PGD-50-10 |
| FGSM + GradAlign | 81.34±0.45% | 46.63±0.52% | PGD-50-10 |
| PGD-10($\alpha = 2\epsilon/10$) | 82.69±0.62% | 50.14±0.64% | PGD-50-10 |
| $\epsilon$=16 / 255, step_size = $\epsilon$ | | | |
| FGSM | 34.42±2.61% | 20.41±0.95% | PGD-50-10 |
| RS-FGSM | 51.87±3.29% | 21.28±0.92% | PGD-50-10 |
| RS-FGSM-wo-proj | 57.89±5.82% | 25.63±0.38% | PGD-50-10 |
| Boundary-RS-FGSM-wo-proj | 57.32±8.06% | 26.32±1.42% | PGD-50-10 |
| FGSM + GradAlign | 63.20±1.03% | 26.04±0.66% | PGD-50-10 |
| PGD-10($\alpha = 2\epsilon/10$) | 65.95±1.40% | 32.86±0.50% | PGD-50-10 |

Table 5.4 – Robustness and accuracy of different robust training methods on CIFAR-10. And the results are shown here with the standard deviation and averaged over 5 random seeds used for training. We report the results by selecting the best test robust accuracy. We reproduce the FGSM+GradAlign[1] and PGD-10 accuracy

# Chapter 6

# Conclusion

We observe that catastrophic overfitting is a general phenomenon in adversarial training, which not only occurs in FGSM and RS-FGSM adversarial training but also occurs in DF$^{\infty}$-1 adversarial training. Thus, it is important to understand catastrophic overfitting and prevent it from happening in order to train the model using weak adversaries and gain robustness against strong adversaries.

We find after catastrophic overfitting happens, the marginal between clean inputs and decision boundary decreases for both FGSM and DF$^{\infty}$-1 trained model. But the mechanisms why FGSM and DF$^{\infty}$-1 adversary loses its effectiveness are different. For FGSM, a new decision boundary is generated along the direction of perturbation and makes the small perturbation more effective than the large perturbation. However, for DF$^{\infty}$-1, there is no decision boundary generated along the direction of perturbation. And large perturbation is still more effective than small perturbation. But the perturbations generated by DF$^{\infty}$-1 becomes smaller after catastrophic overfitting and thus loses their effectiveness.

As for factors that cause catastrophic overfitting, we find not only $l_2$ norm of the perturbation is important but also the direction of the perturbation. Besides, the perturbation is not necessary to span in the entire model and large diversity is not enough to avoid catastrophic overfitting. Finally, we propose a modification to RS-FGSM. After calculating the perturbation, we do not project it back to $l_\infty$-ball to make this perturbation follow the direction of the sign of the input gradient vector as much as possible. This modification permits us to use larger values of $\epsilon$ and improves the robust accuracy compared to RS-FGSM under the same $\epsilon$.

**Future work**

To resolve the catastrophic overfitting problem in adversarial training, there are two main research directions. One is to understand why CO happens and prevent it before it happens, and the other one is to analyze the properties after CO happens and recover the training from CO.

In this project, the geometric properties after catastrophic overfitting happens are well studied for both FGSM and DF$^{\infty}$-1. The remaining question here is why FGSM and DF$^{\infty}$-1 show totally different geometric properties after CO happens. And this question needs further exploration.

As for why catastrophic overfitting happens, we design experiments to analyze three hypotheses on potential factors causing CO, but find that none of them can fully explain why CO happens. So we need to put more efforts to study the main factors that cause CO. Below are some potential research questions:

- Explore the relationship between the direction of the perturbation and the maximum length of the perturbation which does not cause CO. This can be studied both theoretically and empirically.
- In RS-FGSM, we use this equation $\prod_{[-\epsilon,\epsilon]^d}(\delta + \alpha \text{sign}(\nabla_x l(x+\delta, y; \theta)))$ to calculate perturbations. The random initialized $\delta$ shows in two places. One is $\nabla_x l(x + \underline{\boldsymbol{\delta}}, y; \theta)$ to add randomness into the gradient direction inside gradient, and the other one is $\underline{\boldsymbol{\delta}} + \alpha \text{sign}(\nabla_x l(x + \delta, y; \theta))$ to add randomness into the length of each perturbation dimension outside gradient. We find removing either of them $\delta$ will cause the RS-FGSM training to fail. Thus, we can study the usage of $\delta$ in these two places and the roles they play in RS-FGSM to mitigate CO.

# Bibliography


[1] Maksym Andriushchenko and Nicolas Flammarion. *Understanding and Improving Fast Adversarial Training*. 2020. arXiv: 2007.02617 [cs.LG].

[2] Anish Athalye, Nicholas Carlini, and David Wagner. *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*. 2018. arXiv: 1802.00420 [cs.LG].

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. "Evasion Attacks against Machine Learning at Test Time". In: *Lecture Notes in Computer Science* (2013), pp. 387–402. ISSN: 1611-3349. DOI: 10.1007/978-3-642-40994-3_25. URL: http://dx.doi.org/10.1007/978-3-642-40994-3_25.

[4] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. "Thermometer Encoding: One Hot Way To Resist Adversarial Examples". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=S18Su--CW.

[5] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. *Certified Adversarial Robustness via Randomized Smoothing*. 2019. arXiv: 1902.02918 [cs.LG].

[6] Francesco Croce and Matthias Hein. *Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks*. 2020. arXiv: 2003.01690 [cs.LG].

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. In: *Proceedings of the 2019 Conference of the North* (2019). DOI: 10.18653/v1/n19-1423. URL: http://dx.doi.org/10.18653/v1/N19-1423.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2015. arXiv: 1412.6572 [stat.ML].

[9] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. *On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models*. 2018. arXiv: 1810.12715 [cs.LG].

[10] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. *Countering Adversarial Images using Input Transformations*. 2017. arXiv: 1711.00117 [cs.CV].

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). DOI: 10.1109/cvpr.2016.90. URL: http://dx.doi.org/10.1109/cvpr.2016.90.

[12] Hoki Kim, Woojin Lee, and Jaewook Lee. *Understanding Catastrophic Overfitting in Single-step Adversarial Training*. 2020. arXiv: 2010.01799 [cs.LG].

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. *Adversarial Machine Learning at Scale*. 2016. arXiv: 1611.01236 [cs.CV].

[14] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. “Adversarial Examples in the Physical World”. In: *Artificial Intelligence Safety and Security* (July 2018), pp. 99–112. DOI: 10.1201/9781351251389-8. URL: http://dx.doi.org/10.1201/9781351251389-8.

[15] Bai Li, Shiqi Wang, Suman Jana, and Lawrence Carin. *Towards Understanding Fast Adversarial Training*. 2020. arXiv: 2006.03089 [cs.LG].

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. *Towards Deep Learning Models Resistant to Adversarial Attacks*. 2019. arXiv: 1706.06083 [stat.ML].

[17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. *DeepFool: a simple and accurate method to fool deep neural networks*. 2016. arXiv: 1511.04599 [cs.LG].

[18] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. *Robustness via curvature regularization, and vice versa*. 2018. arXiv: 1811.09716 [cs.LG].

[19] Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. *Logit Pairing Methods Can Fool Gradient-Based Attacks*. 2018. arXiv: 1810.12042 [cs.LG].

[20] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. *Adversarial Robustness through Local Linearization*. 2019. arXiv: 1907.02610 [stat.ML].

[21] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. *Certified Defenses against Adversarial Examples*. 2018. arXiv: 1801.09344 [cs.LG].

[22] Leslie Rice, Eric Wong, and J. Zico Kolter. *Overfitting in adversarially robust deep learning*. 2020. arXiv: 2002.11569 [cs.LG].

[23] Andrew Slavin Ross and Finale Doshi-Velez. *Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients*. 2017. arXiv: 1711.09404 [cs.LG].

[24] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*. 2018. arXiv: 1805.06605 [cs.CV].

[25] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. *Adversarial Training for Free!* 2019. arXiv: 1904.12843 [cs.LG].

[26] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. *PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples*. 2017. arXiv: 1710.10766 [cs.LG].

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

[28] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. *Ensemble Adversarial Training: Attacks and Defenses*. 2020. arXiv: 1705.07204 [stat.ML].

[29] B. S. Vivek and R. Venkatesh Babu. "Single-Step Adversarial Training With Dropout Scheduling". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020). DOI: 10.1109/cvpr42600.2020.00103. URL: http://dx.doi.org/10.1109/CVPR42600.2020.00103.

[30] Eric Wong and J. Zico Kolter. *Provable defenses against adversarial examples via the convex outer adversarial polytope*. 2017. arXiv: 1711.00851 [cs.LG].

[31] Eric Wong, Leslie Rice, and J. Zico Kolter. *Fast is better than free: Revisiting adversarial training*. 2020. arXiv: 2001.03994 [cs.LG].

[32] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. "Feature Denoising for Improving Adversarial Robustness". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). DOI: 10.1109/cvpr.2019.00059. URL: http://dx.doi.org/10.1109/CVPR.2019.00059.

[33] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. *You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle*. 2019. arXiv: 1905.00877 [stat.ML].

[34] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. *Theoretically Principled Trade-off between Robustness and Accuracy*. 2019. arXiv: 1901.08573 [cs.LG].