creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

M.S. THESIS

Semantics-Preserving Adversarial Training

의미 보존 적대적 학습

2021년 2월

서울대학교 대학원

컴퓨터공학부

이 원 석

Semantics-Preserving Adversarial Training

의미 보존 적대적 학습

지도교수 이 상 구

이 논문을 공학석사 학위논문으로 제출함

2020년 12월

서울대학교 대학원

컴퓨터공학부

이 원 석

이원석의 석사 학위논문을 인준함

2020년 12월

위 원 장 김 형 주 (인)

부위원장 이 상 구 (인)

위 원 문 봉 기 (인)

Abstract

Semantics-Preserving Adversarial Training

Lee Wonseok
Department of Computer Science and Engineering
The Graduate School
Seoul National University

Adversarial training is a defense technique that improves adversarial robustness of a deep neural network (DNN) by including adversarial examples in the training data. In this paper, we identify an overlooked problem of adversarial training in that these adversarial examples often have different semantics than the original data, introducing unintended biases into the model. We hypothesize that such non-semantics-preserving (and resultingly ambiguous) adversarial data harm the robustness of the target models. To mitigate such unintended semantic changes of adversarial examples, we propose *semantics-preserving adversarial training* (SPAT) which encourages perturbation on the pixels that are shared among all classes when generating adversarial examples in the training stage. Experiment results show that SPAT improves adversarial robustness and achieves state-of-the-art results in CIFAR-10, CIFAR-100, and STL-10.

Keywords: Adversarial Training, Adversarial Robustness, Image Classification, Data Augmentation

Student Number: 2019-23759

Contents

Abstract **i**

Contents **ii**

List of Figures **iv**

List of Tables **vi**

Chapter 1 Introduction **1**

Chapter 2 Preliminaries **5**

Chapter 3 Related Works **9**

Chapter 4 Semantics-Preserving Adversarial Training **11**

4.1 Problem of PGD-training 11

4.2 Semantics-Preserving Adversarial Training 13

4.3 Combining with Adversarial Training Variants 14

Chapter 5 Analysis of Adversarial Examples **16**

5.1 Visualizing Various Adversarial Examples 16

5.2 Comparing the Attack Success Rate 17

Chapter 6 Experiments & Results **22**

6.1 Evaluating Robustness 22

6.1.1 CIFAR-10 & CIFAR-100 22

6.1.2 CIFAR-10 with 500K Unlabeled Data 24

6.1.3 STL-10 25

6.2 Effect of Label Smoothing Hyperparameter α 25

Chapter 7 Conclusion & Future Work **29**

List of Figures

Figure 2.1 An example of finding an adversarial example by PGD in an ϵ-ball [1]. 7

Figure 4.1 (Left) natural image (Middle) gradient w.r.t. x of standard model (Right) gradient w.r.t. x of adversarially trained model. 12

Figure 5.1 Adversarial examples of adversarially trained model generated on various perturbation limits. All adversarial images are generated by C&W$_\infty$ attack. From left to right: original image, $\epsilon = 32/255$, $\epsilon = 16/255$, $\epsilon = 8/255$. From top to bottom: ship to airplane, horse to frog, bird to frog, horse to frog, automobile to ship. 19

Figure 5.2 Adversarial examples generated on adversarially trained model with various attacks. All adversarial images are generated on $\epsilon = 32/255$. From left to right: original image, PGD-CE, PGD-LS ($\alpha = 0.2$), PGD-LS ($\alpha = 0.8$). 20

Figure 5.3 Robust accuracy under various attacks on a standard model and an adversarially trained model. 21

Figure 6.1 Clean and Robust Accuracy (%) under various α and perturbation limits on CIFAR-10 dataset. 26

Figure 6.2 Clean and Robust Accuracy (%) under various α and perturbation limits on CIFAR-100 dataset. 27

List of Tables

Table 5.1 Robust accuracy (%) of standard model under PGD-KL and PGD-LS attack with different α values. 18

Table 5.2 Robust accuracy (%) of adversarially trained model under PGD-KL and PGD-LS attack with different α values. . . 19

Table 6.1 Natural and Robust accuracy (%) of WRN-34-10 trained on CIFAR-10 dataset. 23

Table 6.2 Natural and Robust accuracy (%) of WRN-34-10 trained on CIFAR-100 dataset. 24

Table 6.3 Natural and Robust accuracy (%) of WRN-28-10 trained on CIFAR-10 with 500k unlabeled dataset. 25

Table 6.4 Natural and Robust accuracy (%) of ResNet-18 trained on STL-10. 26

Chapter 1

Introduction

Recent successes in many deep learning applications such as computer vision [2], speech recognition [3], game playing [4], and natural language processing [5] raised expectations for AI applications in real life. However, as Deep Neural Networks (DNNs) turn out to be too brittle and susceptible to small perturbations known as adversarial examples [6, 7], serious concerns are being raised on applying DNNs to safety-critical real life tasks such as face recognition [8], autonomous driving [9], and medical applications [10].

A broad definition of an adversarial example is an input to a machine learning model that is intentionally designed by an attacker to fool the model into producing an incorrect output [11]. In the image classification domain, although unrestricted attacks such as adversarial rotations, and translations [12] exist, typically, adversarial examples are crafted by adding some small perturbations to examples to change model outputs, where perturbation size is restricted by an L_p norm ϵ-ball constraint. These are called sensitivity-based adversarial examples [13]. Underlying assumption here is that every data point inside an ϵ-ball is semantically identical. Extensive studies were made to effectively find adversarial examples inside an ϵ-ball and to make classifiers empirically or provably robust to such L_p norm bounded adversarial attacks [14, 15, 16, 17, 18, 19, 20]. However, many defense methods including even the recent researches were later shown to be ineffective [21, 22, 23, 24].

In spite of such bitter failures, adversarial training, which incorporates adversarial examples into the training data, remains as one of the best defense methods. Projected gradient descent (PGD) is typically utilized to find adversarial examples used in training stage. PGD finds a data point x' which is most likely to be adversarial inside an ϵ-ball centered at the original data x by maximizing the loss function and the x' is used as training data in place of the original data x. Therefore, adversarial training can be thought of as an online data augmentation technique.

Recently, authors in [13] exposed a problem of adversarial training. This failure mode motivates us to rethink the adversarial training from the beginning. The problem of adversarial training is that actually, data points in ϵ-ball are not always semantically identical. There are perturbations that change oracle (human) label inside ϵ-ball in MNIST dataset [13]. Additionally, adversarial examples of adversarially trained models are often perceived as samples from different classes [25]. Even if the label does not change, at least the semantics can become mixed or ambiguous. In the perspective of the data augmentation, such a data is undesirable because it makes data noisy and disrupts model from learning intended semantics. Instead of learning the intended task-relevant information, a model learns unintended features and wrong type of invariances. We hypothesize that such non-semantics-preserving (and resultingly ambiguous) adversarial data harm the robustness of the target model.

To mitigate such unintended semantic changes of adversarial examples, we propose *semantics-preserving adversarial training* (SPAT) which encourages perturbation on the pixels that are shared among all classes when generating adversarial examples in the training stage. We show in Section 5 that perturbing on the pixels that are shared among all classes is more effective in preserving original semantics than perturbing on the pixels that are only influential to the true class. Our aim is to train a model with more semantics-preserving adversarial examples.

By proposing SPAT, we are arguing for the necessity to separate adversarial examples for training and adversarial examples for evaluating the robustness. When evaluating the adversarial robustness, even if the semantics is mixed or ambiguous, it is plausible to decide the label of the data based on the dominant semantics of the data as long as the label of the data is same. However, when training, such semantically ambiguous data disturbs a model from learning intended semantics.

SPAT is a simple yet effective method. It is worth noting that SPAT is orthogonal to existing adversarial training variants in that SPAT suggests a new method for generating adversarial examples used in training stage which remains relatively unexplored. We show in Section 6 that when combined with TRADES and MART, SPAT achieves state-of-the-art results in CIFAR-10, CIFAR-100, and STL-10 and is further improved with additional unlabeled data in CIFAR-10.

Our contributions are summarized as:

- We analyze and visualize adversarial examples on various settings in a complex dataset.

- We identify an overlooked problem of adversarial training in that these adversarial examples often have different semantics than the original data, introducing unintended biases into the model. To mitigate such unintended semantic changes of adversarial examples, we propose *semantics-preserving adversarial training* (SPAT).

- We experimentally show that SPAT can improve adversarial robustness and achieve state-of-the-art results in CIFAR-10, CIFAR-100, and STL-10.

In Chapter 2, we introduce some preliminaries. In Chapter 3, we introduce some related works. In Chapter 4, we analyze the current problem of PGD-based adversarial training and propose SPAT as a solution to mitigate such problems. In Chapter 5, we visualize various attacks and empirically show that PGD-LS attack is more effective at preserving semantics than other attacks. In Chapter 6, we show that SPAT with proper choice of α which is dependent on perturbation limit ϵ improves robustness and present state-of-the-art robustness on CIFAR-10, CIFAR-100, and STL-10.

Chapter 2

Preliminaries

Consider a standard image classification task. Given a classfier which is parametrized by θ and data $(x, y) \sim D$ where x is a image and $y \in \{0, 1\}^K$ is a one-hot encoded class label:

Standard Training In standard training, we minimize the loss with training data. Formally, the goal of Empirical Risk Minimization (ERM) is to find parameter θ that minimizes the risk:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, E_{(x,y)\sim D}[L(\theta, x, y)] \tag{2.1}$$

DNNs trained with standard training usually have good test set performance, but are highly vulnerable to adversarial examples.

Adversarial Example An adversarial example is an input to a machine learning model that is intentionally designed by an attacker to fool the model into producing an incorrect output [11]. In the image classification domain, although unrestricted attacks such as adversarial rotations, and translations [12] exist, typically, adversarial examples are crafted by adding some small perturbations to examples to change model outputs, where perturbation size is restricted by an L_p norm ϵ-ball constraint.

Adversarial Training In adversarial training, we minimize the loss with training data or their adversarial examples. Sometimes both are used and sometimes only adversarial examples are used. We explain the latter here. Then, the former is obvious.

When we minimize the loss with adversarial examples, we allow some perturbations for each data point x. The perturbation set S is chosen to capture semantic similarity of images. Usually, l_p ball centered at x is used:

$$B_\epsilon^p(x) = \{x' : ||x - x'||_p \leq \epsilon\} \tag{2.2}$$

In this paper, we use $p = \infty$. Then, Adversarial Risk Minimization (ARM) is formulated as a saddle point problem:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, E_{(x,y)\sim D}[\max_{x' \in B_\epsilon^\infty(x)} L(\theta, x', y)] \tag{2.3}$$

which can be rewritten as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, E_{(x,y)\sim D}[L(\theta, \hat{x}', y)] \tag{2.4}$$

where

$$\hat{x}' = \underset{x' \in B_\epsilon^\infty(x)}{\operatorname{argmax}}\, L(\theta, x', y) \tag{2.5}$$

It is alternating between inner maximization problem and outer minimization problem and rewritten formulation can be thought of as just a modification of standard training where adversarial data is used instead of natural data. In its variants, loss function L in equation 2.4 and equation 2.5 are not essentially same. Compared with DNNs trained with standard training, DNNs trained with adversarial training are more robust to adversarial examples and have many benefits such as human-aligned gradient, and shape bias while having slightly lower test set performance. Although many defense methods have

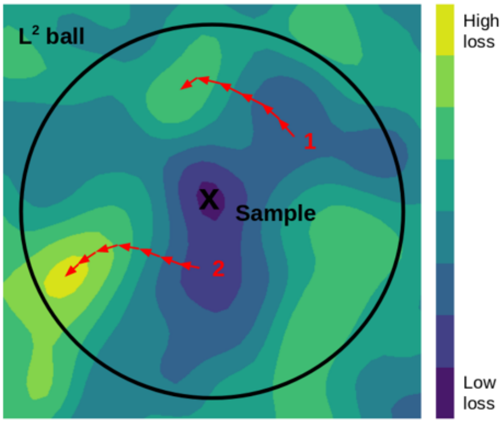


Figure 2.1: An example of finding an adversarial example by PGD in an ϵ-ball [1].

failed, adversarial training remains as a one of the best defense methods against adversarial examples.

Projected Gradient Descent To find adversarial examples (= approximately solve inner maximization problem), adversarial training usually utilizes Projected Gradient Descent (PGD) [15]. First, we introduce Fast Gradient Sign Method (FGSM) [7] and then explain PGD. Both find a point that maximizes the loss within a ϵ-ball because that point would be likely to be misclassified by the model. FGSM finds an adversarial example as

$$x' = x + \epsilon sgn(\nabla_x L(\theta, x, y)) \tag{2.6}$$

which is a one-step first-order method.

The multi-step variant of FGSM is called PGD:

$$x^{(t+1)} = \Pi_{B_\epsilon^\infty(x)}(x^{(t)} + \alpha sgn(\nabla_{x^{(t)}} L(\theta, x^{(t)}, y))) \tag{2.7}$$

where $\alpha > 0$ is a step size and Π is a projection operator that projects adversarial example onto ϵ-ball centered at original data point x and $x^{(t)}$ is a adversarial example at step t. The differences with regular gradient descent are (1) the result is projected onto ϵ-ball and (2) the gradient is computed with respect to the input instead of the parameter. For a visual explanation, refer to Figure 2.1. For both methods, at the beginning($x^{(0)}$), small Gaussian or uniform noise may be added to x and that is called a random start. PGD is performed for fixed iteration T and is called PGD-T algorithm. Most commonly used function for surrogate loss of inner maximization is standard cross entropy loss [15, 17, 26]. KL-divergence function and other methods have been used as well [16, 27, 28]. We emphasize that $\hat{x}'$ found by PGD is not always adversarial. Instead, PGD finds a point that is most likely to be misclassified by the model.

Chapter 3

Related Works

Since its discovery [6], adversarial noise was investigated by many researchers on adversarial attacks, defenses and its characteristics. Started by the work of [7] which proposed first attack Fast Gradient Sign Method (FGSM) and first defense based on FGSM, many effective attacks including DeepFool [29], JSMA [30], BIM [31, 32], PGD [15], C&W [28], and spatially transformed attack [33] have been proposed and defenses including distillation [34], thermometer encoding [35], stochastic gradients [36], exploding/vanishing gradients [37], and adversarial training [15], followed. However, many defenses including methods that utilize gradient obfuscation were later turned out to be ineffective [22, 24, 21, 23].

Adversarial training is a defense technique that incorporates adversarial examples into training data. Due to its efficacy, it has been studied widely and many variants exist [15, 16, 17, 38, 26, 39], which are orthogonal to our work. However, ALP was controversial for its effectiveness [40, 41, 42] and drawbacks of adversarial training was recently discovered [13]. The work of [43, 44] investigated to improve accuracy while preserving robustness.

Researches on cause and characteristics of adversarial examples were performed. The authors of [45] argued that adversarial examples can be directly attributed to the presence of non-robust features. The authors of [46] showed that excessive invariance can cause adversarial vulnerability and [13] claimed that there are fundamental tradeoffs between invariance and sensitivity. The authors of [47] argued that learning shortcut can cause adversarial vulnerability. The authors of [25] showed that robustness may be at odds with accuracy. In contrast, the authors of [48] showed that adversarial examples can improve accuracy. The authors of [39, 49] showed that unlabeled data can improve robustness. The authors of [50] analyzed adversarially trained CNNs (AT-CNNs) and showed that adversarial training alleviates the texture bias of standard CNNs and helps CNNs become more shape-biased.

Chapter 4

Semantics-Preserving Adversarial Training

In this chapter, we analyze current problem of PGD-based adversarial training and propose *semantics-preserving adversarial training* (SPAT) algorithm, which encourages perturbation on the pixels that are shared among all classes when generating adversarial examples in the training stage.

4.1 Problem of PGD-training

Several researches have been conducted on how the survived adversarially trained models differ from the standard models [50, 25]. Specifically, authors in [25] have shown that the gradients of adversarially trained models align well with perceptually relevant features of the input image while the gradients of standard models seem as mere noises to humans. See Figure 4.1 for examples.

Since PGD finds adversarial examples based on the gradients of the models, the distinct aspects of gradients induce significant disparity between the PGD-generated examples from the adversarially trained models and the standard models. Adversarial examples of standard models seem as noisy version of the

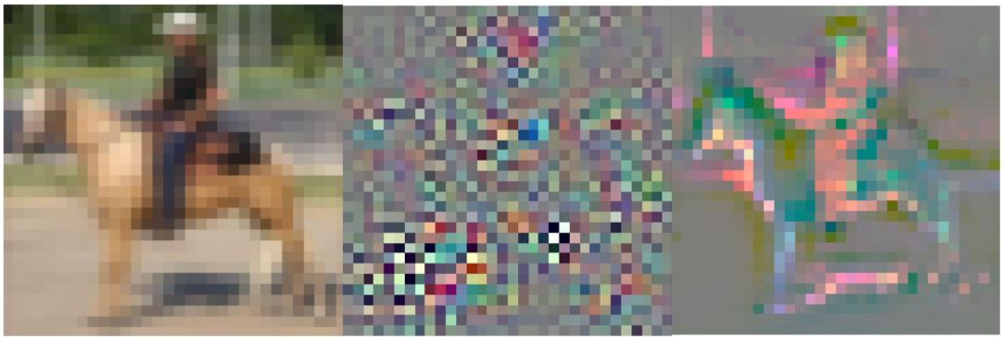


Figure 4.1: (Left) natural image (Middle) gradient w.r.t. x of standard model (Right) gradient w.r.t. x of adversarially trained model.

original images. In contrast, adversarial examples of adversarially trained models look semantically different from the original images and they often belong to different classes. See Figure 5.1 for examples.

Moreover, there are perturbations that change oracle (human) label inside ϵ-ball in MNIST dataset [13]. This may apply to other datasets and other size of ϵ-balls as well. Thus it can not be guaranteed that adversarial examples have same semantics as original images. If not totally change labels, adversarial perturbations may make images ambiguous by adding semantics of different classes or by erasing semantics of the original classes.

In the perspective of an adversarial attack whose goal is to generate images that the model misclassifies, this is a very interesting phenomenon and it is a evidence that adversarial training can teach the semantics to the models to some degree. However, in the perspective of a data augmentation in adversarial training, training with such an attack is harmful because it prevents the model from learning intended semantics in that it mixes up the task-relevant and task-irrelevant information.

We hypothesize that such non-semantics-preserving (and resultingly ambiguous) adversarial data harm the robustness of the target model and this may be one of the cause of phenomenon that defenses against sensitivity-based attacks harm a model's accuracy on invariance-based attacks [13]. That is, making the model robust in ϵ-balls actually gives the model invariance in wrong direction so that the model becomes invariant to semantics.

4.2 Semantics-Preserving Adversarial Training

To solve this problem, we propose *semantics-preserving adversarial training* (SPAT), where we use label smoothed cross entropy loss (LSCE) [51] instead of standard cross entropy loss (CE) for surrogate loss of inner maximization problem. That is, we use

$$LSCE(p, y) = \sum_{k=1}^{K} -y_k^{LS} \log(p_k) \tag{4.1}$$

where

$$y_k^{LS} = \begin{cases} (1-\alpha) & \text{if } y_k = 1 \\ \alpha/(K-1) & \text{if } y_k = 0 \end{cases} \tag{4.2}$$

for surrogate loss of inner maximization where $\alpha \in [0, 1]$ is a label smoothing hyperparameter and p_k is k-th element of softmax layer output. For full formulation, refer to Equation 4.3. Note that LSCE is equivalent to CE when $\alpha = 0$. Since PGD with cross entropy loss perturbs towards increasing loss only with original class, it encourages erasing semantics of true class and adding semantics of other classes. As a result, PGD with cross entropy loss changes the original semantics of the images.

$$\hat{x}' = \underset{x' \in B_\epsilon^\infty(x)}{\operatorname{argmax}} \, LSCE(p(x', \theta), y) \tag{4.3}$$

In contrast, as SPAT encourages to perturb on the pixels that are shared among all classes, it mitigates two causes of semantic changes of PGD-generated adversarial examples: adding semantics of other classes and erasing semantics of the original class. Such a semantics-preserving effect increases as label smoothing hyperparameter α gets bigger. As α gets bigger, PGD will perturb more on parts that are common across all other classes, therefore lesser erasing semantics of the true class. However, as α gets bigger, it provides less invariance to the model since evenly distributed loss prevents the sample from diverging from the original data point. Therefore, there is a tradeoff. When using LSCE loss for PGD, we call it PGD-LS for convenience and same go for PGD-CE and PGD-KL.

$$CE(p, q) = Entropy(p) + D_{KL}(p||q) \tag{4.4}$$

Since CE loss function is equivalent to KL divergence except for the entropy (which is the constant part), if the softmax probability is same, LSCE loss is equivalent to KL divergence (refer to Equation 4.4). However, we claim that LSCE has advantage over KL divergence in that we are able to control how much semantics to preserve with label smoothing hyperparameter α. Since KL divergence highly depends on the sample prediction computed by trained models, it varies from model to model and from example to example. In contrast, with LSCE, we are able to control the ratio between the true class and other classes. Overall, PGD-LS can be thought of as a generalization of PGD-CE and PGD-KL.

4.3 Combining with Adversarial Training Variants

Since our method is changing the surrogate loss for inner maximization problem, it is orthogonal to various existing adversarial training methods. Therefore, we combine our method with Madry [15], TRADES [16], and MART [17].

Madry + SPAT Loss function is formulated as $CE(p(\hat{x}', \theta), y)$.

TRADES + SPAT Loss function is formulated as
$CE(p(x, \theta), y) + KL(p(x, \theta)||p(\hat{x}', \theta))$.

MART + SPAT Loss function is formulated as
$BCE(p(\hat{x}', \theta), y) + KL(p(x, \theta)||p(\hat{x}', \theta))(1 - p_y(x, \theta))$.

In SPAT, all the adversarial examples $\hat{x}'$ are generated by PGD-LS (Equation 4.3).

Chapter 5

Analysis of Adversarial Examples

In this chapter, we compare our proposed PGD-LS attack with various PGD-based attacks. First, we show that semantic changes occur in ϵ-balls and such semantic changes can be mitigated with PGD-LS. Next, to show the effect of hyperparameter α and compare with other PGD-based attacks numerically, we plot attack success rate curves on a standard model and an adversarially trained model.

5.1 Visualizing Various Adversarial Examples

Here, we visualize adversarial examples generated by various PGD-based attacks and various perturbation limits. First, to test the effect of perturbation limit on adversarial examples, we generate adversarial examples with C&W$_\infty$ attack on various perturbation limits on CIFAR-10 dataset. Figure 5.1 shows the result. With $\epsilon = 32/255$, labels of the images completely change. For example, images in first row show a ship turning into a airplane. On $\epsilon = 16/255$, semantics of the images change to some degree and labels often become ambiguous and mixed. For instance, images in first row show a ship becoming ambiguous between a ship and a airplane and images in second row show that the shape of a horse is deformed. On $\epsilon = 8/255$, which is the most commonly used perturbation limit on CIFAR-10 dataset, labels of the images are preserved but some images show mixed semantics. For example, semantics of adversarial image in third row is mixed but the label is preserved. However, since we cannot inspect every image in every used dataset, we cannot assure that there is no label-changing or ambiguous adversarial examples in defined epsilon balls.

Secondly, to confirm that PGD-LS attack is more effective at preserving semantics than PGD-CE attack, we visualize adversarial examples generated by PGD-CE and PGD-LS attack on perturbation limit of $\epsilon = 32/255$. Figure 5.2 shows that adversarial examples generated by PGD-LS attack preserve more semantics than adversarial examples generated by PGD-CE attack and semantics-preserving effect is greater with larger α. Therefore, by using larger α on larger perturbation limit, adversarial training can become more stable by a larger semantics-preserving effect.

5.2 Comparing the Attack Success Rate

Here, we analyze the effect of label smoothing hyperparameter α on attack success rate of PGD-LS against a standard model and an adversarially trained model (Madry). We vary α from 0 to 1 with stride 0.1. Note that when $\alpha = 0$, it is equivalent to PGD-CE. We also plot PGD-KL [16] for comparison. Note that robust accuracy is equal to $1-$ attack success rate. Figure 5.3 shows the result.

As expected, we observe that bigger α leads to lower attack success rate ($=$ higher robust accuracy) due to its larger semantics-preserving effect. It is worth noting that for both model, when α is 1.0, accuracy under attacks get higher than accuracy for clean examples. This is because of closed set nature of classification problem. In closed set classification, moving away from every class except true class results in moving towards true class.

Attack success rates in a standard model and an adversarially trained model show quite different aspect. On a standard model, $\alpha = 0$ shows huge difference from other PGD-LS. In contrast, on adversarially trained model, PGD-LS shows gradual changes.

Relation of PGD-KL vs PGD-LS vs PGD-CE We also notice that PGD-KL shows difference in a standard model and an adversarially trained model. Therefore, we investigate what α value leads PGD-LS to have similar attack success rate with PGD-KL. We show results in Table 5.1 and 5.2. Results show that for a standard model, PGD-LS with α between 1e-4 and 5e-5 is similar with PGD-KL and for an adversarially trained model, PGD-LS with α between 0.4 and 0.5 is similar with PGD-KL. Probably standard training makes a highly confident classifier whereas adversarial training yields less confident classifier. The disadvantage of PGD-KL is that since its power of attack (or attack success rate) is determined by the model's sample prediction which is uncontrollable, it is undependable.

Attack	PGD-KL	PGD-LS				
		1e-2	1e-3	1e-4	5e-5	1e-5
PGD-20 acc	47.51	80.04	69.23	51.28	44.16	25.3

Table 5.1: Robust accuracy (%) of standard model under PGD-KL and PGD-LS attack with different α values.

Attack	PGD-KL	PGD-LS				
		0.1	0.2	0.3	0.4	0.5
PGD-20 acc	71.58	57.7	62.93	67.7	72.14	75.57

Table 5.2: Robust accuracy (%) of adversarially trained model under PGD-KL and PGD-LS attack with different α values.

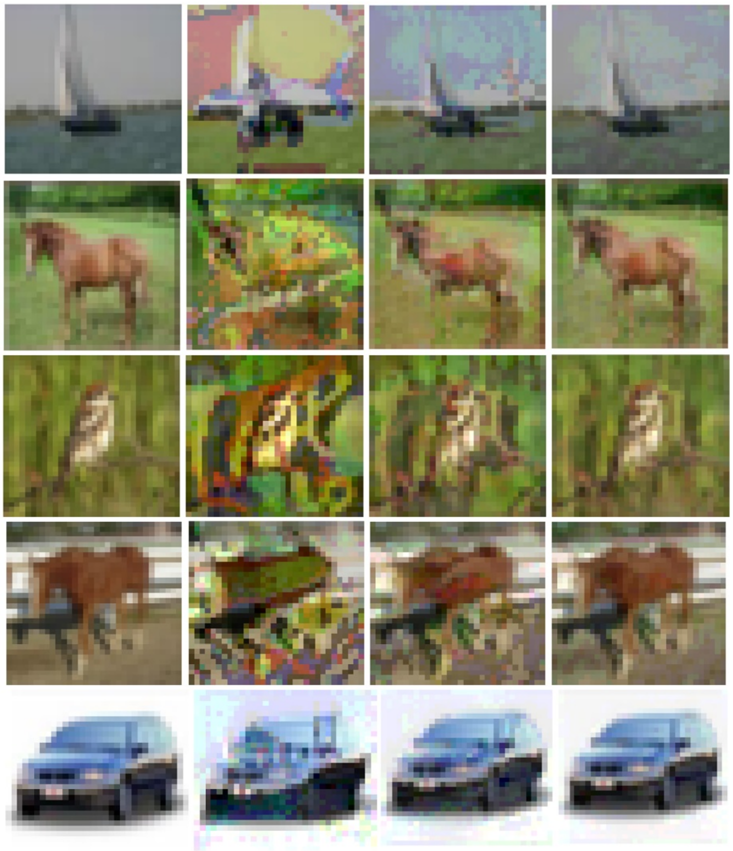


Figure 5.1: Adversarial examples of adversarially trained model generated on various perturbation limits. All adversarial images are generated by C&W$_\infty$ attack. From left to right: original image, $\epsilon = 32/255$, $\epsilon = 16/255$, $\epsilon = 8/255$. From top to bottom: ship to airplane, horse to frog, bird to frog, horse to frog, automobile to ship.

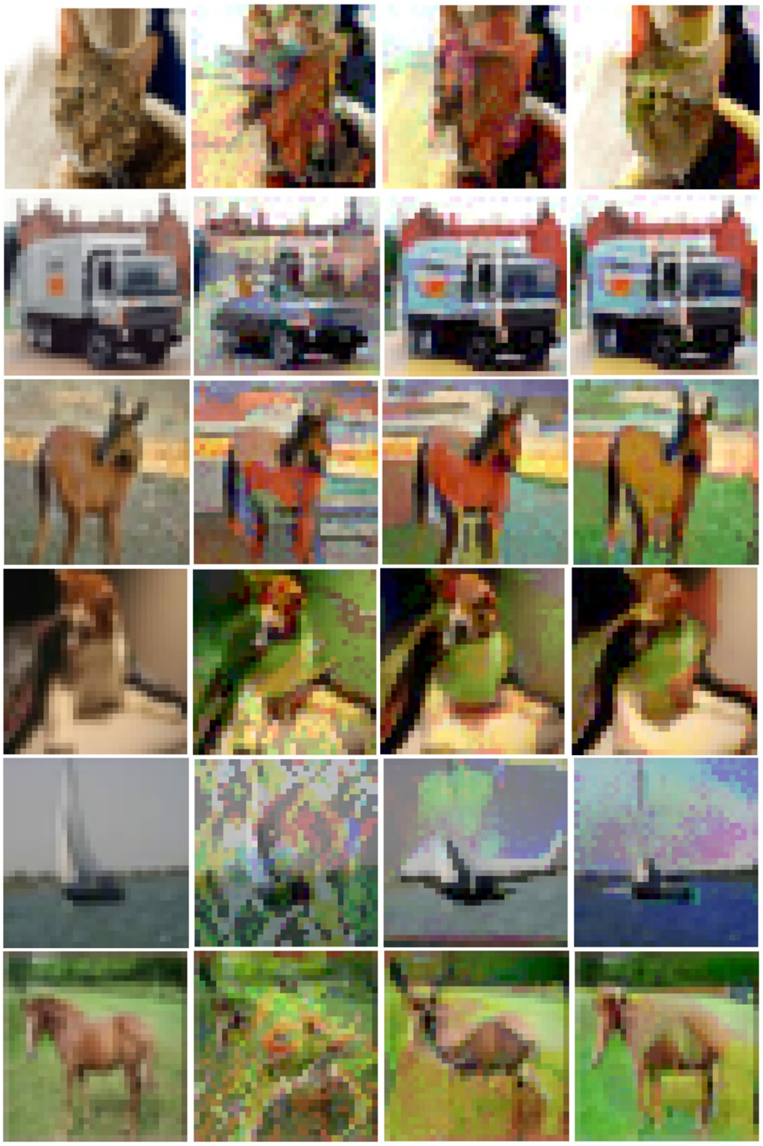
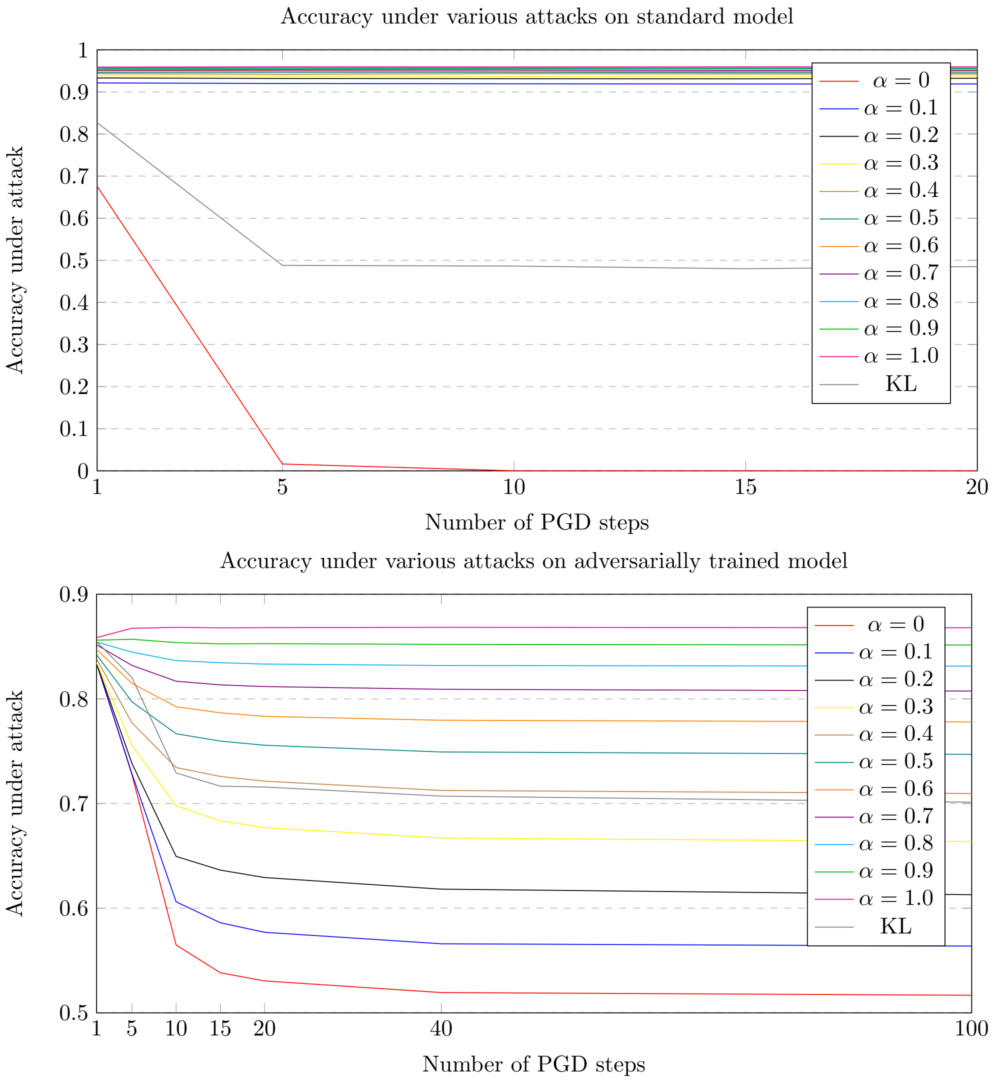


Figure 5.2: Adversarial examples generated on adversarially trained model with various attacks. All adversarial images are generated on $\epsilon = 32/255$. From left to right: original image, PGD-CE, PGD-LS ($\alpha = 0.2$), PGD-LS ($\alpha = 0.8$).



Accuracy under various attacks on standard model

Accuracy under various attacks on adversarially trained model

Figure 5.3: Robust accuracy under various attacks on a standard model and an adversarially trained model.

Chapter 6

Experiments & Results

In this chapter, we first verify the efficacy of SPAT empirically by several experiments and then check how the label smoothing parameter α of the SPAT affects the accuracy of a classifier on various perturbation limits.

6.1 Evaluating Robustness

We train WideResNet-34-10 [52] on CIFAR-10 and CIFAR-100 dataset [53] to benchmark state-of-the-art robustness and train with 500k unlabeled data on CIFAR-10 to achieve further improvements. Then, we train ResNet-18 [2] on STL-10 [54] to test our method on larger images.

6.1.1 CIFAR-10 & CIFAR-100

We compare our method with adversarial training variants: 1) Madry [15], 2) TRADES [16], and 3) MART [17].

Training Details For CIFAR-10, we follow all the settings in MART. Models are trained with SGD with momentum 0.9, weight decay 7e-4 and initial learning rate is 0.1 and divided by 0.1 at 75-th and 90-th epoch. All images are

Defense	Natural	FGSM	PGD-20	CW_∞
Madry	84.39	59.93	56.37	54.14
TRADES	**85.97**	**62.29**	57.32	54.38
MART	83.7	61.93	58.46	53.28
TRADES + SPAT ($\alpha = 0.1$)	84.6	61.46	58.22	**54.97**
MART + SPAT ($\alpha = 0.3$)	81.93	61.87	**59.59**	51.57

Table 6.1: Natural and Robust accuracy (%) of WRN-34-10 trained on CIFAR-10 dataset.

normalized into [0, 1] and when training, data augmentation such as random horizontal flipping and random crop with 4 pixel padding is performed. The perturbation limit is $\epsilon = 8/255$ and for training attack, we use PGD-10 with random start and step size is $\epsilon/4$. For all hyperparameters, we use $\lambda = 6$ for TRADES and TRADES + SPAT, $\lambda = 5$ for MART and MART + SPAT. For CIFAR-100, we use same settings except for weight decay which follow their original implementations.

We test all models against FGSM(w/o random start), PGD-20, and C&W$_\infty$ (optimized by PGD for 30 steps) [28] attacks.

Results & Discussion Table 6.1, 6.2 shows the result. In CIFAR-10, our proposed method MART + SPAT outperforms all other methods in terms of PGD-20 accuracy, which is the most common comparison setting. Also, TRADES + SPAT outperforms all other methods in terms of CW_∞ accuracy. Compared with its original algorithm TRADES, TRADES + SPAT improves on PGD-20 and CW_∞ accuracy. MART + SPAT improves on PGD-20 over MART but worsens on CW_∞ accuracy. This phenomenon is similar to relation of MART and TRADES. MART has higher PGD-20 accuracy than TRADES but shows lower CW_∞ accuracy. We presume that this is because BCE loss used in MART

Defense	Natural	FGSM	PGD-20	CW_∞
Madry	**61.77**	34.42	31.51	30.1
TRADES	57.13	32.9	31.02	28.07
MART	58.56	36.46	34.12	30.16
TRADES + SPAT ($\alpha = 0.1$)	55.54	33.35	31.20	28.14
MART + SPAT ($\alpha = 0.2$)	60.28	**36.91**	**34.66**	**30.93**

Table 6.2: Natural and Robust accuracy (%) of WRN-34-10 trained on CIFAR-100 dataset.

(instead of CE in TRADES) sometimes cause mismatch between robustness against PGD attack and C&W attack. All our method worsens in natural accuracy, which conforms with claim that robustness may be inherently at odds with natural accuracy [25, 16].

In CIFAR-100, our proposed method MART + SPAT outperforms all other methods in all robust accuracy and also improves natural accuracy over its original algorithm, MART. TRADES + SPAT improves on all robust accuracy over TRADES but worsens natural accuracy.

Overall, experiment results show that our proposed method SPAT consistently improves robust accuracy. It was generally thought that increasing the power of attack by increasing the number of attack iterations can create more robust model [15, 55, 42]. Our results show that stronger attack is not the only way to creating more robust models. This conforms with our intuition that semantics-preserving data augmentation is important.

6.1.2 CIFAR-10 with 500K Unlabeled Data

Here, we investigate the additional benefit of unlabeled data with SPAT. We follow exact same settings in RST [39]. Specifically, we train RST + SPAT and MART + SPAT on WideResNet-28-10 and compare them with RST and MART

Defense	Natural	PGD-20
RST	89.65	63.00
MART	**89.81**	63.06
RST + SPAT ($\alpha = 0.1$)	89.52	**63.47**
MART + SPAT ($\alpha = 0.1$)	89.44	63.37

Table 6.3: Natural and Robust accuracy (%) of WRN-28-10 trained on CIFAR-10 with 500k unlabeled dataset.

on natural accuracy and PGD-20 (settings in [39]) accuracy. Evaluation results are shown in Table 6.3. Results show that SPAT improves PGD-20 accuracy on both RST and MART. We again confirm that SPAT consistently improves robust accuracy.

6.1.3 STL-10

Here, we train our methods on STL-10 to test versatility of our methods on larger images. Since STL-10 has larger images, their semantics are more distinct. We follow similar settings as in previous experiments except that we train our models for 200 epochs and batch size of 32. Evaluation results are shown in Table 6.4. Results show that SPAT again improves PGD-20 accuracy on both TRADES and MART and similar to previous experiments, natural accuracy decreases.

6.2 Effect of Label Smoothing Hyperparameter α

To test the effect of label smoothing hyperparameter α on SPAT on various perturbation limits, we train ResNet-50 [2] on CIFAR-10 dataset. We apply SPAT on the standard adversarial training method, Madry [15]. We vary α from 0 to 1 with stride 0.2. Note that when $\alpha = 0$, it is equivalent to Madry.

Defense	Natural	PGD-20
TRADES	**70.15**	42.16
MART	67.89	44.24
TRADES + SPAT ($\alpha = 0.1$)	69.24	43.33
MART + SPAT ($\alpha = 0.2$)	67.85	**45.19**

Table 6.4: Natural and Robust accuracy (%) of ResNet-18 trained on STL-10.

Adversarial Setting We train models on various perturbation limits to see the effect of SPAT on various amount of possible semantics change. The perturbation limits for training are $\epsilon = 4/255, 8/255, 16/255$ and $\epsilon = 8/255$ for evaluation. For training, we use PGD-10 with random start and step size is $\epsilon/4$. For evaluation, we use PGD-20 with random start and step size is $\epsilon/10$.

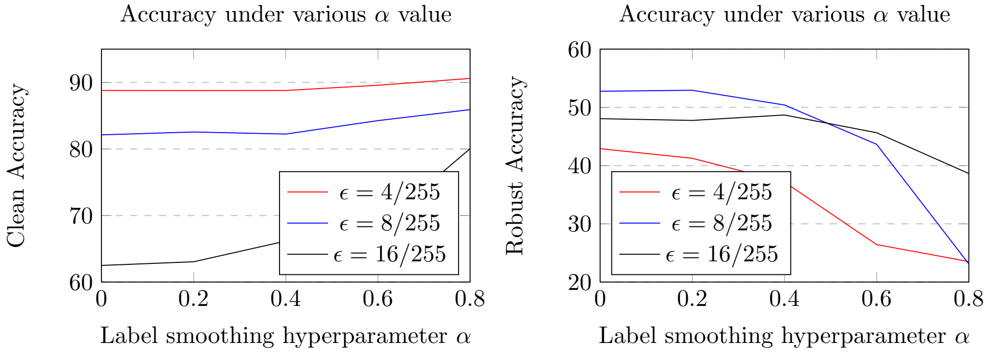


Figure 6.1: Clean and Robust Accuracy (%) under various α and perturbation limits on CIFAR-10 dataset.

Evaluation Results In Figure 6.1, we show the performance of Madry and SPAT w.r.t. α on various perturbation limits. Clean accuracy refers to the accuracy of a classifier evaluated on natural images and robust accuracy refers to the accuracy of a classifier evaluated on adversarial examples generated by PGD-20. All DNNs trained by SPAT show higher clean accuracy compared

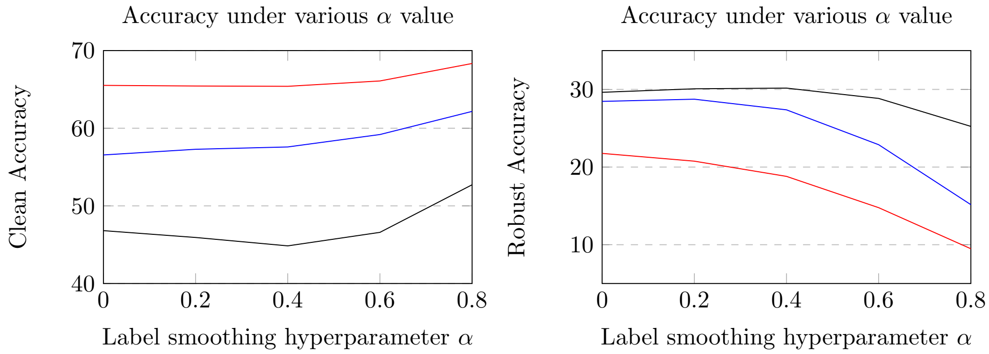


Figure 6.2: Clean and Robust Accuracy (%) under various α and perturbation limits on CIFAR-100 dataset.

to the models that are trained by Madry. The models trained by SPAT with higher α have higher clean accuracy, but when $\alpha = 1$, training gets broken. The model trained by SPAT with $\alpha = 1$ has clean accuracy of 32.44% and robust accuracy of 0% when $\epsilon = 8/255$. We presume that training with data that is more 'friendly' than clean data (or flattering) kills training.

For robust accuracy, a DNN trained with $\alpha = 0$ get highest robust accuracy when $\epsilon = 4/255$. When $\epsilon = 8/255$, SPAT with $\alpha = 0.2$ get highest robust accuracy. When $\epsilon = 16/255$, SPAT with $\alpha = 0.4$ get highest robust accuracy. In contrast to clean accuracy, robust accuracy peaks at certain α value and decreases as it gets farther away from the peak α value. In addition, the peak α value is higher on bigger ϵ-ball.

Higher clean accuracy is achieved with smaller ϵ-ball. In contrast, higher robust accuracy is achieved with middle sized ϵ-ball and some semantics preservation. Overall, this conforms with our intuition that although some degree of invariances are essential to achieve robustness, too much invariance (or unintended bias) caused by non-semantics-preserving data hinders adversarial training and that can be mitigated with semantics-preserving adversarial training.

We confirm that semantics-preserving adversarial training with proper choice of α helps to increase robustness of the model even with large perturbation limits. The optimal amount of semantics to be preserved which is controlled by α is dependent on the radius of the ϵ-ball since larger perturbation limit allows for more semantic changes. When α is too big compared to ϵ, SPAT makes adversarial data too close to original data so that it does not provide enough invariances and the model becomes less robust. SPAT could serve as a guide to finding the appropriate size of ϵ-ball.

Chapter 7

Conclusion & Future Work

In this paper, motivated by recently discovered vulnerability of adversarially trained DNNs, we investigate the effect of semantics of adversarial data on adversarial robustness. We show that adversarial data often change original semantics and such semantic changes are bigger in larger ϵ-balls.

To mitigate such semantic changes of adversarial data for adversarial training, we propose a PGD-LS and a semantics-preserving adversarial training (SPAT) algorithm. Our empirical analysis show that PGD-LS preserve more semantics than other PGD-based attacks. Experiment results show that SPAT with proper choice of α which is dependent on the perturbation limit improves robustness. We conclude that not only insufficient invariance but also too much invariance (= semantics-changing adversarial data) impairs robustness.

Efficiently finding the optimal combination of ϵ and α remains uninvestigated. Also, there could be other heuristic techniques to preserve semantics of adversarial data. We developed our method from intuition that perturbations on the pixels that are shared among all classes would preserve more semantics and showed its effectivness empircally but theoretical analysis is rather weak. We leave them as future works.

Fundamentally, learning the intention and defining the semantics without human supervision would be the goal. Also, accounting for the amount of semantics-change of data could be a future research direction.

References

[1] Oscar Knagg. Know your enemy: How you can create and defend against adversarial attacks, Jan 2019.

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

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[4] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] 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.

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

[8] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[9] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[10] Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

[11] I. Goodfellow and N. Papernot. Is attacking machine learning easier than defending it?, Feb 2017.

[12] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811, 2019.

[13] Florian Tramèr, Jens Behrmann, Nicholas Carlini, Nicolas Papernot, and Jörn-Henrik Jacobsen. Fundamental tradeoffs between invariance and sensitivity to adversarial perturbations. *arXiv preprint arXiv:2002.04599*, 2020.

[14] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to

adversarial attacks. In *International Conference on Learning Representations*, 2018.

[16] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

[17] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

[18] 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, 2018.

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

[20] Mislav Balunovic and Martin Vechev. Adversarial training and provable defenses: Bridging the gap. In *International Conference on Learning Representations*, 2019.

[21] 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.

[22] 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, 2018.

[23] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

[24] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

[25] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2018.

[26] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 480–491, 2019.

[27] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[28] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[30] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learn-

ing in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

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

[32] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[33] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

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

[35] 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.

[36] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[37] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

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

[39] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11192–11203, 2019.

[40] Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

[41] Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks. *arXiv preprint arXiv:1810.12042*, 2018.

[42] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. In *International Conference on Learning Representations*, 2019.

[43] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint arXiv:1910.08051*, 2019.

[44] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. *arXiv preprint arXiv:2002.11242*, 2020.

[45] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[46] Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*, 2018.

[47] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020.

[48] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

[49] Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems*, pages 12214–12223, 2019.

[50] Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *International Conference on Machine Learning*, pages 7502–7511, 2019.

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

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

[53] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[54] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[55] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

초 록

적대적 학습은 적대적 예제를 학습 데이터에 포함시킴으로써 심층 신경망의 적대적 강건성을 개선하는 방어 방법이다. 이 논문에서는 적대적 예제들이 원본 데이터와는 때때로 다른 의미를 가지며, 모델에 의도하지 않은 편향을 집어 넣는다는 기존에는 간과되어왔던 적대적 학습의 문제를 밝힌다. 우리는 이러한 의미를 보존하지 않는, 그리고 결과적으로 애매모호한 적대적 데이터가 목표 모델의 강건성을 해친다고 가설을 세웠다. 우리는 이러한 적대적 예제들의 의도하지 않은 의미적 변화를 완화하기 위해, 학습 단계에서 적대적 예제들을 생성할 때 모든 클래스들에게서 공유되는 픽셀에 교란하도록 권장하는, 의미 보존 적대적 학습을 제안한다. 실험 결과는 의미 보존 적대적 학습이 적대적 강건성을 개선하며, CIFAR-10과 CIFAR-100과 STL-10에서 최고의 성능을 달성함을 보인다.

주요어: 적대적 학습, 적대적 강건성, 이미지 분류, 데이터 증대

학번: 2019-23759