Improving the Robustness of Deep Neural Networks via Adversarial Training with Triplet Loss

Pengcheng Li[1] , **Jinfeng Yi**[2] , **Bowen Zhou**[2] and **Lijun Zhang**[1]
[1]National Key Laboratory for Novel Software Technology Nanjing University, Nanjing 210023, China
[2]JD AI Research, China
{lipc, zhanglj}@lamda.nju.edu.cn, {yijinfeng, bowen.zhou}@jd.com

Abstract

Recent studies have highlighted that deep neural networks (DNNs) are vulnerable to adversarial examples. In this paper, we improve the robustness of DNNs by utilizing techniques of Distance Metric Learning. Specifically, we incorporate Triplet Loss, one of the most popular Distance Metric Learning methods, into the framework of adversarial training. Our proposed algorithm, Adversarial Training with Triplet Loss (AT^{2}L), substitutes the adversarial example against the current model for the anchor of triplet loss to effectively smooth the classification boundary. Furthermore, we propose an ensemble version of AT^{2}L, which aggregates different attack methods and model structures for better defense effects. Our empirical studies verify that the proposed approach can significantly improve the robustness of DNNs without sacrificing accuracy. Finally, we demonstrate that our specially designed triplet loss can also be used as a regularization term to enhance other defense methods.

1 Introduction

Deep neural networks (DNNs) have been widely used for security-critical tasks, including but not limited to autonomous driving [Evtimov *et al.*, 2017], surveillance [Ouyang and Wang, 2013], biometric recognition [Xu *et al.*, 2017], and malware detection [Yuan *et al.*, 2014]. However, recent studies have shown that DNNs are vulnerable to adversarial examples [Goodfellow *et al.*, 2014; Papernot *et al.*, 2016; Chen *et al.*, 2017; Li *et al.*, 2018], which are carefully crafted instances that can mislead well-trained DNNs. This raises serious concerns about the security of machine learning models in many real-world applications.

Recently, many efforts have been made to improve the robustness of DNNs, such as (i) using the properties of obfuscated gradients [Athalye *et al.*, 2018] to prevent the attackers from obtaining the true gradient of the model, e.g., mitigating through randomization [Xie *et al.*, 2018], Thermometer encoding [Buckman *et al.*, 2018], and Defense-GAN [Samangouei *et al.*, 2018]; (ii) adding adversarial examples into the training set, e.g., Adversarial Training [Szegedy *et al.*, 2013; Goodfellow *et al.*, 2014], scalable Adversarial Training [Kurakin *et al.*, 2016b], and Ensemble Adversarial Training [Tramèr *et al.*, 2018]. However, it was shown that the first type of defense methods had been broken through by various targeted countermeasures [Carlini and Wagner, 2017a; He *et al.*, 2017; Athalye *et al.*, 2018]. The second type of methods also suffers the distortion of the classification boundary for the reason that they only import adversarial examples against some specific types of attacks.

In this paper, we follow the framework of Adversarial Training and introduce Triplet Loss [Schroff *et al.*, 2015], one of the most popular Distance Metric Learning methods, to improve the robustness by smoothing the classification boundary. Triplet loss is designed to optimize the embedding space such that data points with the same label are closer to each other than those with different labels. The primary challenge of triplet loss is how to select representative triplets, which are made up of three examples from two different classes and jointly constitute a positive pair and a negative pair. Since adversarial examples contain more information about the decision boundary than normal examples, we modify the anchor of triplet loss with adversarial examples to enlarge the distance between adversarial examples and examples with different labels in the embedding space. Then, we add this fine-grained triplet loss to the original adversarial training process and name the new algorithm as Adversarial Training with Triplet Loss (AT^{2}L). We also propose an ensemble algorithm which aggregates different types of attacks and model structures to improve the performance. Furthermore, the proposed triplet loss can be applied to other methods as a regularization term for better robustness.

We summarize our main contributions as follows:

- We introduce triplet loss into the adversarial training framework and modify the anchor of triplet loss with adversarial examples. We also design an ensemble version of our method.
- We propose to take our triplet loss as a regularization term and apply it to existing defense methods for further improvement of robustness.
- We conduct extensive experiments to evaluate our algorithms. The empirical results show that our proposed approach behaves more robust and preserves the accuracy of the model, and the triplet loss can also improve

the performance of other defense methods.

2 Related Work

In this section, we briefly review existing adversarial attack and defense methods.

2.1 Attack Methods

Attack methods can be divided into two main categories: gradient-based attack and optimization-based attack.

The gradient-based attack asks for the structure of the attacked model and requires that the attacked model should be differentiable. Then it generates adversarial examples by adding perturbation along the direction of the gradients. FGSM [Goodfellow *et al.*, 2014], Single-Step Least-Likely (LL) [Kurakin *et al.*, 2016a; Kurakin *et al.*, 2016b] and their iterative versions, i.e., I-FGSM and I-LL, are popular methods in this type of attack.

The optimization-based attack formulates the task of attack as an optimization problem which aims to minimize the norm of perturbation and make the DNN model mis-classify adversarial examples. C&W attack [Carlini and Wagner, 2017b] is by far one of the strongest optimization-based attacks. It can reduce the classifiers' accuracy to almost 0 and has bypassed over 10 different methods designed for detecting adversarial examples [Carlini and Wagner, 2017a]. However, it is more time-consuming than gradient-based algorithms.

2.2 Defense Methods

Many recent defense approaches are based on a technique called obfuscated gradients [Athalye *et al.*, 2018]. It is similar to gradient masking [Papernot *et al.*, 2017] which is a failed defense method that tries to deny the attacker access to a useful gradient, and leads to a false sense of security in defenses against adversarial examples. Typical defense methods using obfuscated gradients are thermometer encoding [Buckman *et al.*, 2018], Stochastic activation pruning [Dhillon *et al.*, 2018], Mitigating through randomization [Xie *et al.*, 2018] and Defense-GAN [Samangouei *et al.*, 2018].

Another common method is adversarial training, which proposes to add adversarial examples to the training set and then retrain the model for better robustness. Szegedy *et al.* (2013) first propose this simple process in which the model is trained on adversarial examples until it learns to classify them correctly. However, this type of methods suffers the distortion of the classification boundary. So in this paper, we introduce Distance Metric Learning to alleviate this distortion.

3 Methodology

In this section, we first introduce the triplet loss. Then we present Adversarial Training with Triplet Loss (AT^{2}L) and an ensemble version of AT^{2}L. Finally, we propose to treat our special triplet loss as a regularization term and combine it with existing defense methods.

3.1 Triplet Loss

A triplet [Schroff *et al.*, 2015] consists of three examples from two different classes, which jointly constitute a positive pair and a negative pair. We denote $(\mathbf{x}_i^a, \mathbf{x}_i^p, \mathbf{x}_i^n)$ as a triplet, where $(\mathbf{x}_i^a, \mathbf{x}_i^p)$ has the same label and $(\mathbf{x}_i^a, \mathbf{x}_i^n)$ has the different. The $\mathbf{x}_i^a$ term is referred to as the anchor of a triplet. The distance between the positive pair is encouraged to be smaller than that of the negative pair, and a soft nearest neighbor classification margin is maximized by optimizing a hinge loss. Specifically, triplet loss forces the network to generate an embedding where the distance between $\mathbf{x}_i^a$ and $\mathbf{x}_i^n$ is larger than the distance between $\mathbf{x}_i^a$ and $\mathbf{x}_i^p$ plus the margin parameter α.

Formally, we define the triplet loss function as follows:

$$\frac{1}{N}\sum_{i=1}^{N}\max\Bigg\{\|f(\mathbf{x}_i^a)-f(\mathbf{x}_i^p)\| - \|f(\mathbf{x}_i^a)-f(\mathbf{x}_i^n)\|+\alpha, 0\Bigg\},$$

where N is the cardinality of the set of triplets used in the training process, $f(\cdot)$ is the output of the last fully connected layer of our neural network, $\|\mathbf{x}_i - \mathbf{x}_j\|$ represents a metric of distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Here we use ℓ_∞ norm in our experiments.

Generating all possible triplets would result in redundant triplets and lead to slow convergence. So in the next sections, we use sampling strategy to generate triplets in our algorithms.

3.2 Adversarial Training With Triplet Loss (AT^{2}L)

The original version of adversarial training is to craft adversarial examples for the entire training set and add them to the training process. Specifically, it generates X_{adv} which contains adversarial examples of instances in training set X. Then it concatenates X_{adv} and X as X' and retrains the model with X'. During each iteration of the original algorithm, it generates the adversarial examples against the current model. The loss function of the original adversarial training is formulated as:

$$\frac{1}{(1+\lambda)k}\left(\sum_{i=1}^{k}\ell(\mathbf{x}_i, y_i)+\lambda\sum_{i=1}^{k}\ell(\mathbf{x}_i^{adv}, y_i)\right), \qquad (1)$$

where λ is the hyper-parameter, k is the size of the mini-batch sampled from X, y is the label of $\mathbf{x}_i$ and $\mathbf{x}_i^{adv}$ is the adversarial example of $\mathbf{x}_i$.

To encourage a larger margin between the positive class and the negative class, we incorporate triplet loss into the loss function. Specifically, for example $\mathbf{x}_i$, we generate adversarial example $\mathbf{x}_i^{adv}$ and sampled an example $\mathbf{x}_i^n$ from the mini-batch which has a different label to construct a new triplet $(\mathbf{x}_i^{adv}, \mathbf{x}_i, \mathbf{x}_i^n)$. The main difference between this triplet and the original triplet is that instead of taking the original example $\mathbf{x}_i$ as the anchor, we choose the adversarial example $\mathbf{x}_i^{adv}$, which contains more information about the decision boundary. Specifically, when dealing with a binary classification problem, we sample $\mathbf{x}_i^n$ which has the opposite label to $\mathbf{x}_i$. For multi-class problems, we sample $\mathbf{x}_i^n$ from the same class as the adversarial example $\mathbf{x}_i^{adv}$, which is an incorrect class from the view of human beings. We apply this new triplet

Algorithm 1 Adversarial Training With Triplet Loss (AT^{2}L)

```
Train $f(\cdot)$ with training data $X$;
repeat
    Construct $X_{adv}$ against $f(\cdot)$ for each instance in $X$;
    $X' = [X, X_{adv}]$;
    Retrain $f(\cdot)$ with $X'$ using Eq. (2);
until Training converged
```

to the triplet loss, and combine it with the loss of adversarial training, so the loss function of our algorithm is formulated as:

$$
\begin{aligned}
\hat{\ell}(\mathbf{x}, y) =& \frac{1}{(1+\lambda_1)k}\left(\sum_{i=1}^{k}\ell(\mathbf{x}_i, y_i) + \lambda_1 \sum_{i=1}^{k}\ell(\mathbf{x}_i^{adv}, y_i)\right) \\
&+ \frac{\lambda_2}{k}\sum_{i=1}^{k}\max\Bigg\{\|f(\mathbf{x}_i^{adv}) - f(\mathbf{x}_i)\| \\
&- \|f(\mathbf{x}_i^{adv}) - f(\mathbf{x}_i^{n})\| + \alpha, 0\Bigg\},
\end{aligned}
\tag{2}
$$

where k is the size of a mini-batch, and λ_1, λ_2 and α are the hyper-parameters. We utilize this new loss function to retrain the model and summarize the proposed algorithm in Algorithm 1.

3.3 Ensemble AT^{2}L

We proceed to improve the robustness of the model against unknown type of attacks for the reason that the originally proposed algorithm can only defend against known type of attacks, where defenders have detailed information about the attacking methods and lack robustness against attacks transferred from unknown models. Our first attempt is to combine different attack methods together to increase the robustness. As shown in Algorithm 2 where A denotes an aggregation of attack methods, we conduct adversarial training on a collection of adversarial examples that are generated by all the attack methods. In this paper, we consider three types of attacks as follows:

- Gradient-based: $A = \{\text{FGSM}, \text{LL}, \text{I-FGSM}, \text{I-LL}\}$.
- Optimization-based: $A = \{\text{C\&W}\}$.
- Mixed: $A = \{\text{FGSM}, \text{LL}, \text{I-FGSM}, \text{I-LL}, \text{C\&W}\}$.

On the other hand, we adopt the idea of Ensemble Adversarial Training [Tramèr *et al.*, 2018], which says that the augmentation of training data with perturbations transferred from other models can improve the robustness not only under a known type of attack, but also under an unknown type of attack. As shown in Algorithm 2, where M is a set of model structures, we extend our training set with adversarial examples against different models in M.

In general, the ensemble version of our algorithm not only considers various types of attacks, but also involves adversarial examples generated against different model structures. Therefore, our algorithm captures more information about the decision boundary, and with our designed triplet loss, it can smooth the classification boundary and learn a better embedding space to alleviate the distortion.

3.4 Triplet Regularization

Our triplet loss can also be regarded as a regularization term:

$$
\begin{aligned}
R(\mathbf{x}, y) =& \frac{\lambda}{k}\sum_{i=1}^{k}\max\Bigg\{\|f(\mathbf{x}_i^{adv}) - f(\mathbf{x}_i)\| \\
&- \|f(\mathbf{x}_i^{adv}) - f(\mathbf{x}_i^{n})\| + \alpha, 0\Bigg\}
\end{aligned}
$$

Thus, it can be incorporated into most of the existing defense methods for better robustness. The defense methods based on obfuscated gradients mostly mask the real gradient by adding non-differentiable preprocessing or random processes, and there is no restriction on the loss function used in the training process. Therefore, we can modify their loss function by adding our triplet regularization term to further increase the robustness.

For example, Buckman *et al.* (2018) propose to encode the input with Thermometer Encoding and retrain the model with the traditional adversarial training. Triplet regularization can be easily applied to this method by changing the loss function of the adversarial training process. Mitigating through randomization [Xie *et al.*, 2018] and Defense-GAN [Samangouei *et al.*, 2018] both perform transformations over original inputs without changing the loss. So we can directly incorporate our triplet regularization into their losses to improve the defense effect.

4 Experiments

In this section, we present experimental results.

4.1 Settings

We conduct experiments over three datasets, i.e., Cats vs. Dogs [Elson *et al.*, 2007], MNIST [LeCun, 1998] and CIFAR10 [Krizhevsky and Hinton, 2009]. Cats vs. Dogs is a large scale image dataset used for binary classification problems. MNIST and CIFAR10 are commonly used datasets for multi-class classification problems.

The attack methods we used in our experiments include FGSM, I-FGSM, LL, I-LL, C&W, LS-PGA and Deepfool [Moosavi-Dezfooli *et al.*, 2016] and the model structures used in the experiments are different for three datasets. The parameters of these methods, detailed model structures and full results of the experiments are described in the full version of our paper[1].

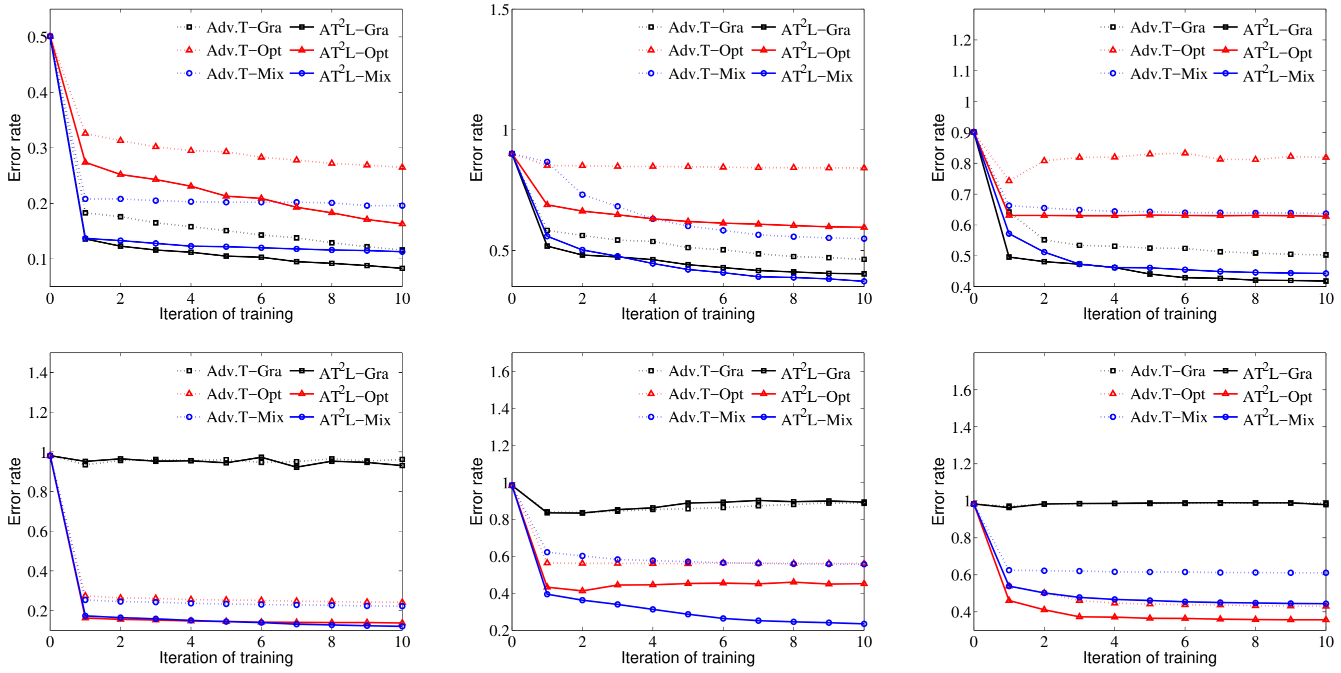
4.2 Adversarial Training With Triplet Loss (AT^{2}L)

To illustrate the advantage of the proposed method, we compare it with adversarial training without triplet loss, whose loss function is Eq. (1). As for the hyper-parameters in Eq. (1) and Eq. (2), we traverse in the appropriate interval and find

[1] https://arxiv.org/abs/1905.11713

Algorithm 2 Ensemble version of AT^{2}L

```
Train $f(\cdot)$ with training data $X$;
repeat
    for $a$ in $A$ do
        for $m$ in $M$ do
            Construct $X^{a,m}$, which is the set of adversarial examples of $X$ against model $m$ under the attack method $a$.
        end for
    end for
    $X_{adv} = \{X^{a,m}\}, a \in A, m \in M$;
    repeat
        Sample $k$ clean examples $B = \{\mathbf{x}_1, ..., \mathbf{x}_k\}$ from training set $X$;
        Sample $k$ adversarial examples $\{\mathbf{x}_1^{adv}, ..., \mathbf{x}_k^{adv}\}$ from $X_{adv}$. Each $\mathbf{x}_i^{adv}$ is the adversarial example of $\mathbf{x}_i$;
        Construct a new training batch $B' = \{\mathbf{x}_1, ..., \mathbf{x}_k, \mathbf{x}_1^{adv}, ..., \mathbf{x}_k^{adv}\}$;
        For each instance of $\{\mathbf{x}_1^{adv}, ..., \mathbf{x}_k^{adv}\}$, take $\mathbf{x}_i$ as $\mathbf{x}_i^p$ in triplet loss and sample an example from $B$ with a different label from $\mathbf{x}_i$ as $\mathbf{x}_i^n$ in triplet loss;
        Perform one training step of network $f(\cdot)$ using the mini-batch $B'$ according to Eq. (2);
    until Training converged
until Training converged
```

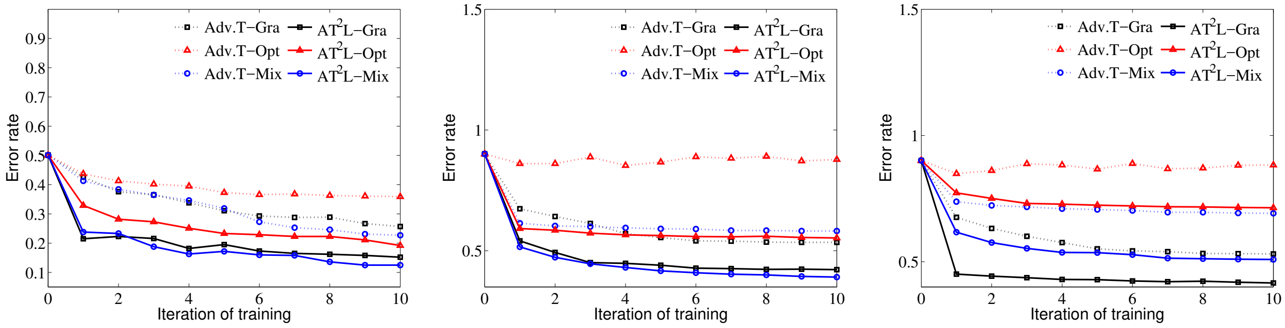


(a) Cats vs. Dogs (b) MNIST (c) CIFAR10

Figure 1: Results on three datasets where attackers perform white-box attacks. ‘Adv.T’ means the traditional adversarial training. ‘-Gra’ means the training process uses the gradient-based attack methods to generate adversarial examples. ‘-Opt’ means using optimization-based attack method, i.e., C&W and ‘-Mix’ means using the mixed version of attack methods. The figures of the upper line are attacked by FGSM and figures of the bottom line are attacked by C&W.

that they have a stable performance in a proper range of values. Each experiment is tested by two types of attack methods, i.e., FGSM and C&W. Due to the limitation of space, we only show results where attackers perform white-box attacks in Fig. 1 and partial results where defenders perform the attack to a network which is not included in the training set M of our algorithm in Fig. 2.

We have the following observations from the results in Fig. 1: (i) when attacked by gradient-based attacks or optimization-based attacks, AT^{2}L trained with adversarial examples generated by corresponding attacks has the best robustness, e.g., when attacked by gradient-based attacks, the model trained with adversarial examples generated by gradient-based attacks exhibits the best robustness; (ii) when

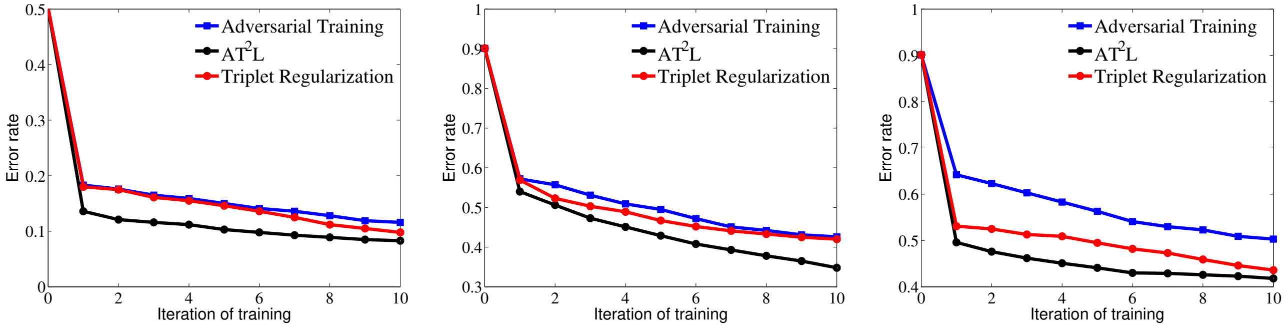


(a) Cats vs. Dogs

(b) MNIST

(c) CIFAR10

Figure 2: Results on three datasets where defenders are under the attack of adversarial examples transferred from unknown models. Notations are the same as Figure 1 and these are attacked by FGSM.



(a) Cats vs. Dogs

(b) MNIST

(c) CIFAR10

Figure 3: Comparison of traditional adversarial training and triplet regularization. The attack method used is FGSM.

trained with gradient-based attacks, AT^{2}L shows almost no robustness against optimization-based attacks, which is shown by the black curves in Fig. 1. However, the robustness of our algorithm trained with optimization-based attacks demonstrates a decent defense effect against gradient-based attacks. This briefly verifies that optimization-based attacks are stronger and contain more information about the decision boundary than gradient-based attacks; (iii) AT^{2}L trained with our mixed version of algorithms shows comparable robustness to the model trained with corresponding attacks. Although the mixed version of AT^{2}L does not perform the best, it provides more reliable robustness when attacked by an unknown type of attacks.

Compared with Fig. 1 (results under adversarial examples transferred from known models), Fig. 2 (results under adversarial examples transferred from unknown models) shows that the results under the attack transferred from unknown models are slightly worse than that under the known type of attacks for the reason that the defenders are lack of precise information of the attack, e.g., the type of the attack method and model structure used for attack. However, the model still shows decent robustness against unknown type of attacks, and this is an advantage of our ensemble AT^{2}L, which aggregates multiple model structures.

We also find that compared with the model trained over clean data, all the models trained with our algorithm, i.e., AT^{2}L, have no loss of accuracy, and more details can be found in the supplemental material.

4.3 Triplet Regularization

To reveal the effect of triplet regularization, we compare it with the original adversarial training. We use FGSM to generate adversarial examples for the training process and test the robustness by the attack of FGSM.

From Fig. 3, we can see that the error rate of the model trained with our triplet regularization (the red curve) keeps decreasing as the number of iterations increases. So the triplet loss itself can increase the robustness of the model, and its effect is no worse than the original adversarial training (the blue curve). This result verifies our hypothesis that enlarging the margin between the adversarial examples and the negative examples and decreasing the margin between examples with the same class can smooth the decision boundary. This also suggests that the designed triplet regularization can work well in most machine learning problems to increase robustness. We can also see form Fig. 3 that AT^{2}L, which integrates both adversarial training and triplet regularization, shows the best performance.

	Ori	Th. En.(1)	Th. En.(7)	Th. En. + TR(1)	Th. En. + TR(7)
Clean	**5.8**	7.6	10.1	6.1	7.2
FGSM	51.5	37.1	20.0	27.6	**15.1**
PGD/LS-PGA	49.5	39.3	20.9	29.7	**13.4**

Table 1: Error rate against known type of attacks on CIFAR10 over Thermometer Models. 'Th. En.' mean Thermometer Encoding. 'TR' means applying our triplet regularization.

Model	Inception-v3			ResNet-v2-101			Inception-ResNet-v2			Ens-adv-Inception-ResNet-v2		
	Ori	Rand	Rand + TR	Ori	Rand	Rand + TR	Ori	Rand	Rand + TR	Ori	Rand	Rand + TR
FGSM	66.8	36.2	**30.5**	73.7	28.2	**21.7**	34.7	19.0	**8.3**	15.6	**4.3**	4.6
Deepfool	100.0	1.7	**1.1**	100.0	2.3	**1.5**	100.0	1.8	**0.8**	99.8	0.9	**0.7**
C&W	100.0	3.1	**2.6**	100.0	2.9	**1.2**	99.7	2.3	**1.3**	99.1	1.2	**0.9**

Table 2: Error rate of different models under the vanilla attack scenario on the ImageNet datasets. 'Ori' means the original model. 'Rand' means adding some randomization layers. 'TR' means applying our triplet regularization.

Model	A			B			C			D		
	Ori	DG	DG + TR	Ori	DG	DG + TR	Ori	DG	DG + TR	Ori	DG	DG + TR
FGSM	88.3	1.2	**1.1**	97.8	4.4	**0.7**	67.9	1.1	**0.8**	96.2	2.0	**1.6**
C&W	85.9	**1.1**	1.4	96.8	8.4	**4.7**	87.4	**1.1**	**1.1**	96.8	1.7	**1.4**

Table 3: Error rates of different models on the MNIST datasets. 'DG' mean Defense-GAN. 'TR' means applying our triplet regularization. A,B,C,D are different model structures, whose details are described in the supplemental material.

4.4 Current Defense Methods With Triplet Regularization

We further explore the effect of the combination of triplet regularization with existing defense methods. We experiment over some representative defense methods and demonstrate that our triplet regularization can be applied to improve their robustness further. Due to the limitation of space, we show partial results in this part, and full results are listed in the supplemental material.

Thermometer Encoding

We follow the setting of the original paper [Buckman *et al.*, 2018] and do experiments over both known and unknown type of attacks. Partial results are shown in Table 1 which indicate employing our triplet regularization indeed improves the robustness based on the effect of the original defense. For example, when attacked by FGSM, the error rate of the model trained using thermometer encoding after 7 iterations is 20.0%. However, combining with our triplet regularization, the model can achieve 15.1% error rate.

Mitigating Through Randomization

We also apply our triplet loss to the work of Xie *et al.* (2018) who proposed to randomly resize or pad the images to a designed size. This defense can be added in front of normal classification process with no additional training or fine-tuning, and can be combined with our triplet regularization directly. We experiment over two settings from the original paper (vanilla attack scenario and ensemble-pattern attack scenario), and examine the performance of our triplet regularization. The result of the vanilla attack scenario is shown in Table 2. For the model of Inception-ResNet-v2, the randomized procedure only achieves 19.0% error rate under FGSM, but with our triplet regularization, the error rate can drop to 9.3%. These results show that our triplet regularization can further improve the robustness of the model based on the original defense method.

Defense-GAN

Defense-GAN is designed to project samples onto the manifold of the generator before classifying them. Our triplet regularization can be easily applied after the projection of Defense-GAN by simply changing the loss function during the training process of the classifier. We follow the setting of models' structures and parameters in the paper of Samangouei *et al.* (2018). As shown in Table 3, when attacked by C&W, model B attains 8.4% error rate using Defense-GAN, while combining with triplet regularization, it achieves 4.7% error rate. Again, this result shows that equipping the original defense method with triplet regularization can make the trained model more robust.

5 Conclusion

In this paper, we propose Adversarial Training with Triplet Loss (AT^{2}L), which incorporates a modified triplet loss in the adversarial training process to alleviate the distortion of the models' classification boundary. We further design an ensemble version of AT^{2}L and propose to use the triple loss as a regularization term. The results of our experiments validate the effectiveness of our algorithms and demonstrate that our triplet regularization can be applied to existing defense methods for further improvement of robustness.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (2018YFB1004300), NSFC-NRF Joint Research Project (61861146001), YESS (2017QNRC001), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[Athalye *et al.*, 2018] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[Buckman *et al.*, 2018] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *Proceedings of the International Conference on Learning Representations*, 2018.

[Carlini and Wagner, 2017a] 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. ACM, 2017.

[Carlini and Wagner, 2017b] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[Chen *et al.*, 2017] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. *arXiv preprint arXiv:1709.04114*, 2017.

[Dhillon *et al.*, 2018] 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.

[Elson *et al.*, 2007] Jeremy Elson, John JD Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. 2007.

[Evtimov *et al.*, 2017] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *arXiv preprint arXiv:1707.08945*, 2017.

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

[He *et al.*, 2017] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defenses: Ensembles of weak defenses are not strong. *arXiv preprint arXiv:1706.04701*, 2017.

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

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

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

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

[Li *et al.*, 2018] Pengcheng Li, Jinfeng Yi, and Lijun Zhang. Query-efficient black-box attack by active learning. In *Proceedings of the 18th IEEE International Conference on Data Mining*, pages 1200–1205, 2018.

[Moosavi-Dezfooli *et al.*, 2016] 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.

[Ouyang and Wang, 2013] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2056–2063, 2013.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy*, pages 372–387. IEEE, 2016.

[Papernot *et al.*, 2017] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

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

[Schroff *et al.*, 2015] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[Szegedy *et al.*, 2013] 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.

[Tramèr *et al.*, 2018] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *Proceedings of the International Conference on Learning Representations*, 2018.

[Xie *et al.*, 2018] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *Proceedings of the International Conference on Learning Representations*, 2018.

[Xu *et al.*, 2017] Xiaojun Xu, Xinyun Chen, Chang Liu, Anna Rohrbach, Trevor Darell, and Dawn Song. Can you fool ai with adversarial examples on a visual turing test? *arXiv preprint arXiv:1709.08693*, 2017.

[Yuan *et al.*, 2014] Zhenlong Yuan, Yongqiang Lu, Zhaoguo Wang, and Yibo Xue. Droid-sec: deep learning in android malware detection. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 371–372. ACM, 2014.