Postprint: Algorithm for Defending Against Adversarial Examples Based on Adaptive Noise Addition

Authors: Liu Ye, Huang Xianying, Liu Wenxing, Zhu Xiaofei, Li Zhaoping

Date: 2020-09-28T00:00:00+00:00

Abstract

In recent years, image classification technology based on deep neural networks has achieved tremendous success; however, recent studies have demonstrated that deep neural networks are susceptible to adversarial example attacks. To address this issue, some approaches have trained networks by adding Gaussian noise to images, thereby enhancing the network' s capability to defend against adversarial examples. Nevertheless, this method fails to consider that neural networks exhibit varying sensitivities to different regions of an image when introducing noise. To tackle this problem, we propose an adversarial training algorithm with gradient-guided noise addition. During network training, this algorithm adds adaptive noise to images based on the sensitivity of different regions: larger noise is added to regions of higher sensitivity to suppress the network' s sensitivity to image variations, while smaller noise is added to regions of lower sensitivity to improve classification accuracy. Comparative experiments on the CIFAR-10 dataset with existing algorithms demonstrate that the proposed method effectively enhances the accuracy of neural networks in classifying adversarial examples.

Full Text

Preamble

Algorithm for Defending Against Adversarial Examples Based on Adaptive Noise Addition

Liu Ye, Huang Xianying†, Liu Wenxing, Zhu Xiaofei, Li Zhaoping

(School of Computer Science & Engineering, Chongqing University of Technology, Chongqing 400054, China)

Abstract: In recent years, image classification techniques based on deep neural networks have achieved remarkable success. However, recent studies have shown that deep neural networks are vulnerable to adversarial attacks. To address this issue, some approaches train networks by adding Gaussian noise to images to improve their defense capabilities against adversarial examples. However, these methods do not consider that neural networks exhibit varying sensitivity to different regions of an image. To overcome this limitation, this paper proposes an adversarial training algorithm based on gradient-guided noise addition. During network training, this algorithm adds adaptive noise to different regions according to their sensitivity—larger noise is added to more sensitive regions to suppress the network' s sensitivity to image variations, while smaller noise is added to less sensitive regions to improve classification accuracy. Experimental results on the CIFAR-10 dataset demonstrate that the proposed method effectively improves the accuracy of neural networks when classifying adversarial examples compared to existing algorithms.

Keywords: deep neural networks; image classification; adversarial examples; adaptive noise

0 Introduction

In recent years, deep neural networks (DNNs) have achieved tremendous success across various applications, including image classification [1], speech recognition [2], machine translation [3], autonomous driving [4], image captioning [5], and object recognition [6]. However, in 2014, Christian Szegedy et al. discovered that DNNs are highly susceptible to adversarial attacks. In image classification tasks, given a correctly classified image, adding carefully crafted tiny perturbations can cause deep neural networks to misclassify with high confidence. Such perturbed images are called adversarial examples [7]. Beyond image classification, other DNN applications have also been targeted by adversarial examples, including visual question answering [8, 9], image captioning [10], semantic segmentation [11], and others [26, 27], posing significant threats to the deployment of deep neural networks.

Image classification serves as a fundamental task for DNNs in computer vision applications, with extremely broad utility, yet it remains one of the domains most severely affected by adversarial attacks. To enhance the ability of neural networks to correctly classify adversarial examples (i.e., to defend against them), recent work [12, 13] has approached the problem from the perspective of model regularization. The core idea is to train neural networks by adding Gaussian noise to input images during the training phase, thereby achieving network regularization and improving classification accuracy on adversarial examples. However, these methods sample noise from the same Gaussian distribution (identical mean and standard deviation) for all images, failing to consider that networks exhibit different sensitivity to different pixels in an image [14] —that is, changing different pixels in the input image has varying impacts on the classification result. Other approaches have proposed adversarial training

methods [15, 16], which generate adversarial examples using projected gradient descent [15] during training and incorporate these examples into the training set to improve defense capabilities. However, due to the label leaking problem [17], achieving robust defense requires multiple gradient computations for each input image during adversarial example generation, resulting in training time costs exceeding ten times that of normal training.

This paper combines the ideas of adversarial training and training with noisy images to propose a Gradient-Guided Noise Addition (GGNA) adversarial training algorithm. The fundamental concept is that since adversarial examples alter the pixels of the original image, to reduce the network' s sensitivity to pixel changes in the input image, training with noisy images should add more noise to pixels with higher sensitivity (gradient values) to more effectively reduce the network' s sensitivity to changes in those pixels, thereby improving defense capabilities. Conversely, less noise should be added to pixels with lower sensitivity to maintain classification accuracy. Specifically, during training, the algorithm first computes the gradient values for each pixel in the input image, then normalizes these gradient magnitudes to obtain the standard deviation of a Gaussian distribution (with fixed mean 0). Noise added to different pixels is independently sampled from Gaussian distributions with corresponding standard deviations. These noisy images are then added to the training set to train the network. During adversarial example testing, noise is also added to the test examples.

Figure 1 shows the difference between an image and its corresponding adaptive noise map versus a standard noise map. Figure 1(a) displays an image from the CIFAR-10 dataset, Figure 1(b) shows the image with adaptive noise added according to the model' s sensitivity to different pixels, and Figure 1(c) shows the image with standard Gaussian noise.

The main contributions of this paper are: a) To improve the ability of models to correctly classify adversarial examples, we combine noise-based training with adversarial training to propose a novel defense algorithm. b) The proposed GGNA method considers network sensitivity to different pixels during noisy image training, achieving adaptive noise addition. c) The GGNA method employs adversarial training principles but requires only a single gradient computation per image during training, significantly reducing computational overhead compared to standard adversarial training.

1.1 Adversarial Attacks

Recent research has produced numerous results in adversarial example generation (i.e., adversarial attacks). Generally, based on the amount of information exposed to the attacker, adversarial attacks can be categorized as white-box attacks (PGD [15], FGSM [7], and DeepFool [18]) or black-box attacks. In white-box attacks, the attacker has complete access to the neural network, including its architecture and weights, enabling gradient computation via backpropagation. Gradients are valuable to attackers as they represent the sensitivity of the

output to the input image; attackers can modify pixels according to the gradient direction to generate adversarial examples. In black-box attacks [19], the attacker only knows external information (e.g., inputs and outputs) and relies on the transferability of adversarial examples. Since white-box attacks provide richer information, they typically achieve higher attack success rates [20].

Common attack methods include:

The Fast Gradient Sign Method (FGSM) [7] is an effective single-step adversarial attack. Its basic idea is: given an input vector and a target, FGSM changes each element along the gradient direction of the test loss with respect to each element. Adversarial example generation can be described as follows:

where represents the total perturbation constraint determining attack strength, denotes the output of a DNN with parameters on input , is the sign function. Note that if , the generated adversarial sample requires clipping to . The iterative process can be described as:

where is the projection space with upper and lower bounds, is the total perturbation constraint, is the step size per iteration, and is the number of iterations. PGD is a very powerful attack method in white-box scenarios. Compared to FGSM, PGD-generated adversarial examples are more likely to cause model misclassification.

The DeepFool attack method [18] was proposed by Moosavi-Dezfooli et al. to improve FGSM' s requirement for manual perturbation size selection. This method solves the following optimization problem through multiple iterations until obtaining a perturbation that satisfies the condition:

where is the classifier, is the clean image, and is the generated adversarial example. DeepFool attacks the decision boundary and produces smaller perturbations compared to FGSM.

1.2 Adversarial Defense

In response to adversarial attacks, numerous defense methods have been proposed for image classification, including adversarial example detection, adversarial training, and regularization-based methods. Pang et al. [22] proposed a detection method that encourages neural networks to learn latent representations of images by minimizing reverse cross-entropy, thereby separating adversarial examples from clean samples. Although detection algorithms are easy to implement and achieve high detection rates, they only identify whether an input is adversarial; correct classification of detected adversarial examples still requires combination with other defense methods.

Madry et al. [15] improved defense through adversarial training, which theoretically solves the following min-max problem:

where represents adversarial examples, is the space bounded by and , is the corresponding label, is the loss function, is the clean sample, is the neural network

classifier, and represents classifier parameters. The inner maximization problem is approximated by generating adversarial examples via PGD attacks, while the outer minimization problem updates network parameters to minimize adversarial loss caused by inner adversarial examples. The recent TRADES method [16] improves adversarial training by framing it as approximately solving:

where is a regularization term. This method balances accuracy on clean and adversarial examples, achieving better defense effects. For the inner maximization problem, TRADES also uses PGD with multiple iterations to generate adversarial examples.

Zantedeschi et al. [12] trained networks with Gaussian noise added to input images for data augmentation, achieving network regularization to reduce sensitivity to input variations. Noise is also added during adversarial example testing. However, this method uses standard Gaussian noise without considering gradient information. Liu et al. [13] proposed adding Gaussian noise to inputs and networks for defense. Wang et al. [25] proposed MART, which distinguishes misclassified and correctly classified samples during training and applies different maximization methods to train robust models.

2 Gradient-Guided Noise Addition (GGNA) Adversarial Training Algorithm

Projected Gradient Descent (PGD) [15] is a multi-step variant of FGSM. Its basic idea is to initialize with and iteratively compute gradients of the input to update adversarial examples.

2.1 Relationship Analysis Between Gradient and Noise Addition

In neural networks for image classification, the gradient values of the output with respect to each pixel in the input image vary, meaning each pixel has a different impact on the output. For pixels with large gradient values, even small changes can easily cause a correctly classified image to be misclassified. For pixels with gradient values near zero, large perturbations have minimal impact on classification results. Figure 2 [Figure 2: see original paper] shows the absolute gradient values in an 8$×$8 region of a single channel of an input image for a neural network classifier, demonstrating significant variation across pixels.

Since adversarial examples are images with altered pixels, to more effectively reduce the impact of pixel changes on the output, noise-based training should add more noise to pixels with large gradients to suppress network sensitivity, while adding less noise to pixels with near-zero gradients to maintain classification accuracy. As added noise is independently sampled from Gaussian distributions, when the mean is set to 0, larger standard deviations yield higher probabilities of sampling larger noise values. Based on this analysis, during noisy image train-

ing, input image gradients can be converted to standard deviations of Gaussian distributions, with noise for each pixel sampled independently from a Gaussian distribution with that pixel' s gradient-derived standard deviation. Pixels with large gradients receive noise from distributions with large standard deviations, while pixels with small gradients receive noise from distributions with small standard deviations.

2.2 Algorithm Theoretical Analysis

In image classification tasks with neural networks, training a standard classification model requires minimizing the loss to achieve high accuracy. However, due to adversarial examples, we want models to maintain high accuracy on both clean and adversarial samples, minimizing the loss:

This is typically approximated by generating adversarial examples that maximize the loss, commonly implemented as:

where represents sampling from a Gaussian distribution with mean 0 and standard deviation , i.e., Equation (8) adds identically distributed Gaussian noise to images. Equation (9) modifies images along the gradient direction through multiple iterations to generate adversarial examples, where represents the input image gradient. Combining these formulas with the gradient-noise relationship analysis from Section 2.1 yields the following adversarial example generation method:

where represents the number of iterations and the modification magnitude per iteration. Larger gradients should correspond to larger Gaussian standard deviations . The standard deviation is obtained by first taking the absolute value of the gradient, i.e., , then applying min-max normalization: , where 0.0001 is added to the denominator to prevent division by zero, normalizing to [0,1].

However, after normalization, pixel gradient magnitudes may still differ by over 1000×. To stabilize the standard deviations derived from gradients, the normalized gradients are divided by their mean and clipped to [0,1]. To control the maximum standard deviation of the Gaussian distribution, multiply by a hyperparameter in [0,1], finally using the computed gradient as the Gaussian standard deviation .

2.3 Algorithm Description

This section details the proposed gradient-guided noise addition adversarial training algorithm. During network training, the algorithm first computes gradients for each pixel in the original input image, then converts these gradients to Gaussian standard deviations. Noise added to each pixel is independently sampled from Gaussian distributions with corresponding standard deviations, and these noisy images are added to the training set until convergence. During adversarial example testing, noise is also added to test examples, with multiple

predictions made and final results determined by voting. The detailed steps are as follows:

2.3.1 Training Phase

a) Compute pixel gradients in image : Using adversarial training principles, compute the gradient of input image according to Equation (5). The loss consists of two parts: the first is the cross-entropy between the output and label to maintain accuracy on clean samples. The second is the relative entropy between the noisy image and clean image , which minimizes the difference so the network classifies noisy and clean samples identically. Parameters are then updated as , where represents network parameters and the learning rate, until convergence.

b) Convert gradient to Gaussian standard deviation : Since larger gradient values should correspond to larger standard deviations , first take the absolute value . To prevent division by zero, add 0.0001 to the denominator, normalize to [0,1], then divide by the mean and clip to [0,1]. Multiply by hyperparameter to control the maximum standard deviation, yielding .

c) Add adaptive noise to obtain noisy image : Since different pixels in input image have different gradients, noise for each pixel is sampled from different Gaussian distributions (with different standard deviations), i.e., .

d) Compute loss and update network parameters : Calculate loss using Equation (5), then update parameters until convergence.

2.3.2 Testing Adversarial Examples Phase

During testing, since gradient information is unavailable, only standard Gaussian noise is added to adversarial examples. When the perturbation magnitude of adversarial examples is known, the Gaussian noise magnitude can be adjusted accordingly. For simplicity, the standard deviation of test noise can be set to the mean of the actual noise standard deviations added during training. Multiple test results are integrated via voting to obtain the final prediction. No noise is added when testing clean samples.

The pseudo-code for the Gradient-Guided Noise Addition (GGNA) adversarial training method is shown in Algorithm 1. Figure 3 illustrates the difference between GGNA and previous noise addition methods when the maximum Gaussian standard deviation is 0.15. Figure 3(a) shows a clean image from CIFAR-10, (b) shows the adaptive noise map generated by GGNA during training, (c) shows the noise map from standard Gaussian noise addition, (d) shows the gradient map of one channel of image (a), (e) shows the actual noise added by GGNA, (f) shows the actual noise added by previous methods, (g) shows the noise standard deviation for GGNA, and (h) shows the standard deviation for standard Gaussian noise. The figure demonstrates that GGNA achieves adaptive noise

addition during training—adding larger noise to pixels with large gradients and smaller noise to pixels with small gradients—whereas previous methods sample noise from identical distributions without considering gradient magnitude.

Algorithm 1: Gradient-Guided Noise Addition Adversarial Training Algorithm

Input: Gaussian noise standard deviation , learning rate , batch size , network parameters .

Output: Trained network .

a) Read samples from dataset
b) for do /* Compute gradient of input image / / Convert gradient to standard deviation / / Add noise and clip to [0,1] / / Compute loss / / Update parameters using computed loss */
c) Repeat steps a) through i) until convergence

3 Experiments

3.2 Differences Between GGNA and Previous Methods in Noise Addition

Figure 3 shows the differences between the proposed GGNA method and previous noise-based training methods during the training phase when the maximum Gaussian standard deviation is 0.15. Figure 3(a) is a clean image from CIFAR-10, (b) is the adaptive noise map produced by GGNA, (c) is the noise map from standard Gaussian noise addition, (d) is the gradient map of one channel of image (a), (e) is the actual noise added by GGNA, (f) is the actual noise added by previous methods, (g) shows the noise standard deviation for GGNA, and (h) shows the standard deviation for Gaussian noise. The figure demonstrates that to reduce network sensitivity to input variations, GGNA achieves adaptive noise addition during training—adding larger noise to high-gradient pixels and smaller noise to low-gradient pixels—while previous methods sample noise from identical distributions without considering gradient magnitude.

3.3 Effect of Noise Standard Deviation During GGNA Training on Accuracy

Table 1 presents the actual noise standard deviation added, accuracy on clean samples, and accuracy on adversarial examples when GGNA' s maximum noise standard deviation is set to [0.25, 0.35]. Adversarial examples are generated using PGD attack with perturbation size 8/255, iteration count 8, and step size 1/255. For simplicity, the test noise standard deviation is set to the mean of the actual training noise standard deviations, and no Gaussian noise is added when testing clean samples. Table 1 shows that within a certain range, as the training noise standard deviation increases, adversarial example accuracy improves, while clean sample accuracy decreases. To balance these trade-offs, subsequent

experiments use a maximum standard deviation of 0.30, corresponding to an actual added noise standard deviation of 0.18.

Table 1: Effect of Adding Noise with Different Standard Deviations During Training on Accuracy

Maximum Std Dev	Actual Std Dev	Adversarial Accuracy	Clean Accuracy
...	...	...	...

3.4 Training Time Comparison Between GGNA and Standard Adversarial Training

Table 2 compares the training time per epoch between GGNA and TRADES(10) on CIFAR-10. TRADES(10) denotes adversarial training with 10 iterations of gradient computation for adversarial example generation. Since GGNA requires only a single gradient computation during adversarial training, its training time is only 30% of TRADES(10), effectively reducing training time.

Table 2: Training Time Comparison Between GGNA and TRADES Method

Method	Training Time per Epoch
TRADES(10)	···
GGNA	···

3.5 Accuracy Comparison of Different Defense Methods on Clean and Adversarial Examples

Since defense methods exhibit trade-offs between adversarial and clean accuracy [24], this experiment compares these accuracies under identical conditions using PGD attack with perturbation size 8/255, iteration count 8, and step size 3/255. The compared methods include Normal, MART, Gaussian noise, TRADES(10), and the proposed GGNA. For GGNA with maximum noise standard deviation 0.30, the actual noise standard deviation is 0.18. For fair comparison, the standard Gaussian noise method uses a standard deviation of 0.18 (labeled Gaussian noise). Table 3 shows that GGNA achieves better accuracy on both adversarial and clean examples.

Table 3: Accuracy of Different Methods on Normal and Adversarial Examples

Method	Adversarial Accuracy	Clean Accuracy
Normal	···	···

Method	Adversarial Accuracy	Clean Accuracy
MART	···	···
Gaussian noise	···	···
TRADES(10)	···	···
GGNA	···	···

3.6.1 PGD Attack

For comprehensive comparison, extensive testing was conducted under PGD attack, varying step size , iteration count , and maximum perturbation . Based on perturbation calculation methods, attacks are categorized as and types. In attacks, perturbation size is calculated as the maximum absolute difference: . In attacks, it' s calculated as the Euclidean norm: , where is the number of pixels per channel (32$×32*inthisexperiment*).*Forattackswithmaximumperturbation*8/255,*thecorrespondingattackperturbatio*
For GGNA, training uses maximum noise standard deviation 0.3, and testing adds Gaussian noise with standard deviation 0.18. Gaussian noise methods use standard deviation 0.18 for both training and testing.

a) **Effect of attack step size on adversarial accuracy:** Figure 4 [Figure 4: see original paper] shows adversarial accuracy of four defense methods under PGD attack with iteration count 8 and increasing step size . Figures 4(a) and 4(b) show accuracy under and attacks, respectively. As step size increases, all methods' accuracy decreases, but GGNA maintains higher accuracy than the other three methods under both attack types.

b) **Effect of attack iteration count on adversarial accuracy:** Figure 5 [Figure 5: see original paper] compares adversarial accuracy as PGD iteration count increases, with step size set to . Figures 5(a) and 5(b) show results under and attacks. As iterations increase, GGNA achieves comparable accuracy to MART under attacks and outperforms the other two methods. Under attacks, GGNA demonstrates superior accuracy compared to all three comparison methods.

c) **Effect of maximum perturbation on adversarial accuracy:** Figure 6 [Figure 6: see original paper] shows adversarial accuracy under PGD attack with increasing maximum perturbation . For attacks, ranges from [8/255, 26/255]; for attacks, ranges from [3/255, 15/255]$×$32, with iteration count 8 and step size . Under attacks, GGNA achieves competitive accuracy with MART and outperforms the other two methods. Under attacks, GGNA maintains higher accuracy than all comparison methods as perturbation increases.

3.6.2 FGSM and DeepFool Attacks

This section presents adversarial accuracy under DeepFool and FGSM attacks. For DeepFool, maximum iterations are 50 and maximum perturbation is

$(4/255)\$\times 32. For FGSM, maximum perturbation is (4/255)\times\$32$ for attacks and 8/255 for attacks. Due to minimal perturbations in DeepFool attacks, GGNA and Gaussian noise methods use test noise standard deviation 0.1. Table 4 shows that GGNA achieves higher adversarial accuracy than the other three methods against both FGSM and DeepFool attacks.

Table 4: Adversarial Examples Accuracy Under FGSM and DeepFool Attacks

Attack Type	TRADES(10)	Gaussian noise	MART	GGNA
DeepFool	···	···	···	···
FGSM	···	···	···	···

3.7 Discussion

Experiments on CIFAR-10 using multiple attack methods (PGD, FGSM, DeepFool) compared GGNA against MART, TRADES, and Gaussian noise. Since PGD-generated adversarial examples are more effective at causing misclassification, comprehensive tests were conducted varying step size, iteration count, and maximum perturbation. Results show that under attacks, GGNA outperforms the other three methods in adversarial accuracy. Under attacks, GGNA achieves accuracy comparable to the state-of-the-art MART method and higher than TRADES and Gaussian noise. Overall, GGNA provides better or competitive defense performance while significantly reducing training time compared to standard adversarial training.

4 Conclusion

To improve the defense capabilities of deep neural network-based image classification models against adversarial examples, this paper proposes the Gradient-Guided Noise Addition (GGNA) adversarial training method. Extensive experimental results demonstrate that GGNA achieves comparable or superior performance to TRADES, MART, and Gaussian noise methods across multiple adversarial attack scenarios, effectively enhancing the ability of image classification models to correctly classify adversarial examples.

References

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C]// Proc of the 26th Annual Conference on Neural Information Processing Systems, Nevada: MIT Press, 2012: 1097–1105.

[2] Hinton G, Deng Li, Yu Dong, et al. Deep neural networks for acoustic modeling in speech recognition [J]. IEEE Signal Processing Magazine, 2012, 29 (6): 1-29.

[3] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate [C]// Proc of the 3rd International Conference on Learning Representations, San Diego: IEEE Press, 2015: 1-10.

[4] Chen Chenyi, Seff Air, Kornhauser Alain, et al. DeepDriving: Learning affordance for direct perception in autonomous driving [C]// Proc of IEEE International Conference on Computer Vision, Santiago: IEEE Press, 2015: 2722–2730.

[5] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention [C]// Proc of the 32nd International Conference on Machine Learning, Lille: ACM Press, 2015: 2048–2057.

[6] Szegedy C, Liu Wei, Jia Yangqing, et al. Going deeper with convolutions [C]// Proc of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston: IEEE Press, 2015: 1–9.

[7] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks [C]// Proc of the 2nd International Conference on Learning Representations, Banff National Park: IEEE Press, 2014: 1-10.

[8] Xu Xiaojun, Chen Xinyun, Liu Chang, et al. Can you fool ai with adversarial examples on a visual turing test [J]. arXiv preprint arXiv: 1709. 08 693, 2017.

[9] Akhtar N, Mian A. Threat of Adversarial attacks on deep learning in computer Vision: a survey [J]. IEEE Access, 2018, 6 (4): 14410–14430.

[10] Chen Hongge, Zhang Huan, Chen Pinyu, et al. Show-and-fool: Crafting adversarial examples for neural image captioning [J]. arXiv preprint arXiv: 1712. 02051, 2017.

[11] Metzen J H, Kumar M C, Brox T, et al. Universal adversarial perturbations against semantic image segmentation [C]// Proc of the IEEE International Conference on Computer Vision, Venice: IEEE Press, 2017: 2774–2783.

[12] Zantedeschi V, Nicolae M I, Rawat A. Efficient defenses against adversarial atacks [C]// Proc of the 10th ACM Workshop on Artificial Intelligence and Security, colocated with CCS, 2017: 39–49.

[13] Liu Xuanqing, Cheng Minhao, Zhang Huan, et al. Towards robust neural networks via random self-ensemble [J]. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, 11211 LNCS: 381–397.

[14] Papernot N, McDaniel P, Jha S, et al. The limitations of deep learning in adversarial settings [C]// IEEE European Symposium on Security and Privacy, Saarbrucken: IEEE Press, 2016: 372–387.

[15] Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks [C]// Proc of the 6th International Conference on Learning Representations, Vancouver: IEEE Press, 2018: 1-10.

[16] Zhang Hongyang, Yu Yaodong, Jiao Jiantao, et al. Theoretically principled trade-off between robustness and accuracy [C]// Proc of the 36th International Conference on Machine Learning, California: ACM Press, 2019: 7472-7482.

[17] Kurakin A, Goodfellow I, Bengio S. Adversarial machine learning at scale [C]// Proc of the 5th International Conference on Learning Representations, Toulon: IEEE Press, 2017: 1-17.

[18] Moosavi-Dezfooli S M, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks [J]. Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas: IEEE Press, 2016: 2574-2582.

[19] Chen Pinyu, Zhang Huan, Sharma Yash, et al. ZOO: Zeroth order optimization based black-box atacks to deep neural networks without training substitute models [C]// Proc of the 10th ACM Workshop on Artificial Intelligence and Security, colocated with CCS, 2017: 15-26.

[20] Athalye A, Carlini N, Wagner D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples [C]// Proc of the 35th International Conference on Machine Learning, Stockholm: ACM Press, 2018: 436-448.

[21] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks [C]// Proc of the 2nd International Conference on Learning Representations, Banff National Park: IEEE Press, 2014: 82-95.

[22] Pang Tianyu, Du Chao, Dong Yinpeng, et al. Towards robust detection of adversarial examples [C]// Proc of the 32nd Annual Conference on Neural Information Processing Systems, Montreal: MIT Press, 2018: 1-10.

[23] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images [R]. Citeseer, 2009.

[24] Tsipras D, Santurkar S, Engstrom L, et al. Robustness may be at odds with accuracy [C]// Proc of the 7th International Conference on Learning Representations, New Orleans: IEEE Press, 2019: 108-132.

[25] Wang Yisen, Zou Difan, Yi Jinfeng, et al. Improving Adversarial Robustness Requires Revisiting Misclassified Examples [C]// Proc of the 8th International Conference on Learning Representations, IEEE Press, 2020: 1-15.

[26] 马玉琨, 毋立芳, 简萌, 等. 一种面向人脸活体检测的对抗样本生成算法 [J]. 软件学报, 2019, 30 (02): 469-480. (Ma Yukun, Wu Lifang, Jian Meng, et al. Algorithm to generate adversarial examples for face-spoofing detection. [J]. Journal of Software, 2019, 30 (2): 469—480.)

[27] 王文琦, 汪润, 王丽娜, 等. 面向中文文本倾向性分类的对抗样本生成方法 [J]. 软件学报, 2019, 30 (08): 2415-2427. (Wang Wenqi, Wang Run, Wang Lina, et al. Adversarial examples generation approach for tendency classification on chinese texts [J]. Journal of Software, 2019, 30 (8): 2415-2427.)