Journal of Information Hiding and Privacy Protection
DOI: 10.32604/jihpp.2022.029006
Article

Tech Science Press

An Overview of Adversarial Attacks and Defenses

Kai Chen*, Jinwei Wang and Jiawei Zhang

Nanjing University of Information Science and Technology, Nanjing, 210044, China
*Corresponding Author: Kai Chen. Email: 20201220003@nuist.edu.cn
Received: 23 February 2022; Accepted: 11 May 2022

Abstract: In recent years, machine learning has become more and more popular, especially the continuous development of deep learning technology, which has brought great revolutions to many fields. In tasks such as image classification, natural language processing, information hiding, multimedia synthesis, and so on, the performance of deep learning has far exceeded the traditional algorithms. However, researchers found that although deep learning can train an accurate model through a large amount of data to complete various tasks, the model is vulnerable to the example which is modified artificially. This technology is called adversarial attacks, while the examples are called adversarial examples. The existence of adversarial attacks poses a great threat to the security of the neural network. Based on the brief introduction of the concept and causes of adversarial example, this paper analyzes the main ideas of adversarial attacks, studies the representative classical adversarial attack methods and the detection and defense methods.

Keywords: Deep learning; adversarial example; adversarial attacks; adversarial defenses

1 Introduction

The improvement of computer performance and the ever-increasing mass of data have stimulated the rapid development of artificial intelligence. Artificial intelligence technology, mainly represented by deep learning, has achieved remarkable results in image classification, object detection, speech recognition and other fields, and has penetrated into other fields. Artificial intelligence has gradually entered people's lives and brought convenience to people's lives. With the wide application of deep learning, people put forward higher requirements for the robustness of neural networks. As one of the main threats affecting the security of neural networks, adversarial examples have received extensive attention.

Adversarial examples refer to a class of examples in which imperceptible perturbations are added to the examples, causing the classifier to misclassify. Generally speaking, there are usually two requirements for the added perturbation: one is to ensure its weakness. It is invisible to the naked eye or visible to the naked eye but does not affect the overall visual effect; the other is to ensure its

confrontation. After the perturbation is added to the original image, the generated image can make the neural network misclassify.

From the perspective of whether the attacker knows the model parameters, adversarial attacks can be classified into white-box attacks and black-box attacks. White-box attack refers to the attack when the internal structure and parameters of the model are known, and black-box attack refers to the attack when the internal structure and parameters of the model are unknown. Under the white-box condition, although there are many ways to generate adversarial examples, the basic operation process can be completed through the following steps: ① Input the normal examples to the classifier; ② Obtain the loss of the classifier to the normal examples, and reverse it ③ Calculate the gradient of the input example; ④ Calculate the perturbation according to the obtained gradient; ⑤ Superimpose the perturbation and the input example to get the adversarial example. Adversarial examples have good transferability. Adversarial examples generated for a certain network can still maintain their adversarial properties when attacking other models. Under black-box conditions, attackers can often exploit the mobility of adversarial examples to attack.

The defense of adversarial examples is mainly divided into two aspects: one is to detect or restore the examples before they enter the target classifier. It is aim to reject the adversarial examples into the classifier or restore the adversarial examples to normal examples and then enter the classifier. This method has low cost and strong versatility, but is easily broken by attackers using targeted attack methods. The second is to improve the robustness of the classifier, which generally requires modifying the structure of the model or using a specific training method. This method has high cost and low generality, but it is not easy to be broken. In practical applications, it is often possible to combine these two methods to obtain better defense effects.

This paper will start from the attack and defense of adversarial examples, and sort out the representative classical adversarial example generation methods. The defense of adversarial examples starts from two perspectives of active defense and passive defense, and introduces several mainstream detection and defense methods of adversarial examples.

2 Adversarial Attack

2.1 L-BFGS

Szegedy et al. [1] first proposed the concept of adversarial examples in 2014, which was later called L-BFGS. They found that only a small perturbation of the image required a huge perturbation to the neural network, causing the image to be misclassified with high confidence. Szegedy et al. used an optimization-based approach to calculate the adversarial perturbation, optimizing the objective function as in Eq. (1):

$$\min c\|\eta\| + J_\theta(x', y) \quad s.t\ x' \in [0, 1] \tag{1}$$

To find a suitable constant c, L-BFGS computes an approximation of adversarial examples by performing a linear search for $c > 0$. They believe that adversarial examples are extremely rare in the normal data space and are artificially specially designed examples.

2.2 FGSM

Goodfellow et al. [2] explained the reason for the emergence of adversarial examples in 2014. The remarkable effect of adversarial examples on attack classifiers is not the high-dimensional nonlinearity of neural networks, but precisely the high-dimensional linearity of networks. Assume that the original input image is X, and the corresponding adversarial example is, set the relevant weight of the classifier

as, where, limit, to ensure that the perturbation added to the image is small and invisible to the naked eye, then the classification process of the adversarial example after entering the classifier is shown in Eq. (2):

$$\omega^T X' = \omega^T X + \omega^T \eta \tag{2}$$

When the network has high dimensional attributes, the perturbation will increase sharply because of linear expansion, which will lead to the misclassification of the classifier. On this basis, an adversarial example generation method based on the principle of gradient descent is proposed in the literature, which is called FGSM later. FGSM allows the neural network to misclassify the generated image by adding increments in the direction of the gradient, which can be calculated by the back propagation algorithm. The perturbation calculation method of FGSM is shown in Eq. (3):

$$\rho = \varepsilon \text{sign}\left(\nabla \text{J}\left(\theta, x, y\right)\right) \tag{3}$$

where, is the perturbation to be added, is the parameter of the classification model, x is the input of the model, y is the correct label of the input, and $J\left(\theta, x, y\right)$is the loss function during the training of the neural network. FGSM can generate adversarial examples quickly, and its generation speed is much higher than L-BFGS.

2.3 BIM

The Basic Iterative Method (BIM), proposed by Kurakin et al. [3], is an extension of FGSM. In this method, multiple perturbations are carried out along the gradient direction through iteration, each perturbation step is small, and the gradient direction is recalculated after each perturbation. The iteration process is shown in Eq. (4):

$$x'_{i+1} = Clip_{\varepsilon}\{x' + \alpha sign(\nabla_x L\left(x', y\right)\} \tag{4}$$

where, is a constraint function, restricts the perturbation to the neighborhood of input x, and α is expressed as step size. Compared with FGSM, BIM can construct more accurate perturbation and achieve better attack effect. It has been widely used in many adversarial attack and defense methods, but the cost is increased calculation amount and calculation time.

2.4 MI-FGSM

In 2018, Yinpeng et al. [4] proposed another FSGM optimization Method, the Momentum Iterative Fast Gradient Sign Method (MI-FGSM). The use of momentum helps to stabilize the updating direction of perturbation, and also helps to make the loss function jump out of the local extreme value, so as to further improve the resistance and transferability of examples. The method generates a perturbation by incorporating momentum into the basic iterative algorithm, first feeding it into the classifier to get the gradient, then updating it by accumulating the velocity vector in the direction of the gradient in Eq. (5), then applying the symbolic gradient in Eq. (6) to update it, and finally generating the perturbation:

$$g_{t+1} = \mu \times \text{g}_t + \frac{\nabla_x J\left(x'_t, y\right)}{\|\nabla_x J\left(x'_t, y\right)\|_1} \tag{5}$$

$$x'_{i+1} = x' + \alpha \times \text{sign}\left(g_{t+1}\right) \tag{6}$$

2.5 PGD

Project Gradient Descent (PGD) [5] was proposed by Madry et al. PGD is an iterative attack. Compared with the ordinary FGSM, which only performs one iteration, PGD performs several iterations, each iteration step is small and the perturbation is projected to a specified range, as shown in Eq. (7):

$$X_{t+1} = \prod_{X+S}(X_t + \alpha \cdot sign\left(\nabla_X J\left(X_t, y\right)\right) \tag{7}$$

Because the step size of each iteration is small, the optimal solution can be found after several steps. Madry et al. demonstrated that PGD is the strongest first-order attack method. If the neural network is robust to the examples generated by PGD, it is robust to other first-order antagonistic examples. The experiment also proves that the model using PGD algorithm for adversarial training has good robustness.

2.6 ILLC

The Iterative least-likely Class Method (ILLC) [3] was proposed by Goodfellow et al. ILLC completes the transition from target-free attack to target attack. By selecting the category with the lowest confidence of the original image classification as the target category, the attack process is shown in Eq. (8):

$$y_{LL} = argmax\{p(y|X)\} \tag{8}$$

To make the adversarial examples classified as, iterate along the direction of and maximize, as shown in Eq. (9):

$$\mathrm{x}_0' = x, x_{N+1}' = Clip_{x,\varepsilon}\left\{x_N' - \alpha sign\left(\nabla_x J\left(x_N', y_{\mathrm{LL}}\right)\right)\right\} \tag{9}$$

2.7 JSMA

Papernot et al. [6] calculated the Jacobian matrix of original example X with the following calculation method:

$$J_F(x) = \frac{\partial F(x)}{\partial x} = \left[\frac{\partial F_j(x)}{\partial x_i}\right]_{i*j} \tag{10}$$

F represents the neural network from layer 2 to the last layer (logit was used at the beginning of the last layer, later modified to Softmax Layer).

In this way, they found that the input characteristics of example X had the most significant effect on the output. A small perturbation is designed to cause a large change in output, so that a small change in feature can fool the neural network.

Then, we define two opposing saliency maps to select the feature pixels to be created during each iteration. With a 97% success rate of attack, only 4.02% of features were changed per example. However, due to the complexity of calculating Jacobian matrix, the generation of adversarial examples by this method is slow.

2.8 C&W

Carlini et al. [7] propose three adversarial attack methods for finding perturbations that minimize various similarity measures against defensive distillation. According to their further research [8], C&W

attacks are effective against most existing adversarial detection defenses. By limiting the L_0, L_2, L_∞ norm, the perturbation is almost imperceptible to the human eye.

Experimental results show that all the three attacks can achieve 100% attack success rate under defense distillation, and have strong transferability of adversarial examples. When evaluated on MNIST, CIFAR10 and ImageNet, the C&W method performed significantly better than I-FGSM and PGD attacks.

2.9 DeepFool

Moosavi-Dezfooli et al. [9] proposed an adversarial attack method based on decision boundary. Deepfool can generate smaller perturbations than a fast gradient attack. The method assumes that there is a decision boundary for each classification result, and there are different classification results inside and outside the decision boundary. Through several iterations, the image is disturbed along the direction of the decision boundary, and the position of the image is gradually pushed to the outside of the decision boundary, which makes the classifier classification error. This method is faster than a fast gradient attack and can generate more precise perturbations that are less noticeable to the human.

2.10 AdvGAN

After GAN [10] is put forward, countermeasures against attacks using GAN network emerge at the right moment, among which the most classic is AdvGAN proposed by Xiao et al. [11]. The core idea of AdvGAN is to map the natural example to the adversarial perturbation through the generator of GAN, and then add it to the corresponding natural example. The discriminator is responsible for determining whether the input example is adversarial example.

The Fig. 1 shows the overall structure of AdvGAN, which consists of three parts: generator G, discriminant D, and target neural network C. Input natural example x into G to generate adversarial perturbation G(x). x + G(x) is then sent to discriminator D, which is used to distinguish the generated example from the original natural example. Discriminator D Narrows the characteristic differences between the generated instance and the data in the original class by calculating losses. In order to resist attacks, the data is input into target classification model C to obtain losses. By optimizing the loss function, the model converges, and finally the adversarial example generator G is obtained.

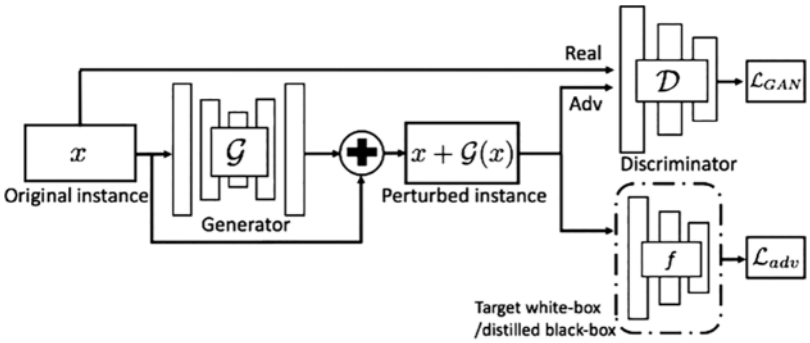

Figure 1: Overall structure of AdvGAN

2.11 Universal Perturbation Attack

Moosavi-Dezfooli et al. [12] proposed a general method to adversarial perturbation. The perturbation superimposed on different images can generate adversarial examples. This method constantly

looks for the minimum perturbation that can push the example to the decision boundary at the current number of iterations. The final universal perturbation is found through continuous accumulation. The advantage of this method is that the adversarial perturbation calculated for a subset of data can be used to attack non-subset images and make these images misclassified with high confidence. In addition, they also prove that the universality of the perturbation not only exists between images, but also between different models.

2.12 Perception Perturbation Attack

Traditional anti-perturbation algorithms usually constrain the anti-perturbation with the L_p norm under RGB. Zhao et al. [13] proposed a new perceptual color distance to constrain the generation of anti-perturbation examples. Adversarial examples generated at perceived color distance allow greater perturbation to the image in RGB space without sacrificing the visual quality of the example. Such adversarial examples have higher confidence, better portability and robustness. The authors show that this method can be combined with the structural information of the image to further improve the invisibility against perturbation.

3 Adversarial Defense

There are two main ways to adversarial defense. One is active defense, which improves the robustness of neural network models. This section mainly discusses network distillation and adversarial training. The other is passive defense, and this section mainly discusses preprocessing and adversarial example detection. Adversarial attack detection is independent of active defense methods and can be used alone or in combination with active defense methods.

3.1 Network Distillation

Distillation network [14] was proposed by G. Hinton and was originally used to reduce the number of parameters of neural network models. By transferring the knowledge of the large network into the small network, and then operating with less computational cost in the deployment phase, while retaining a certain performance, the softmax layer of the large network can be described as:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{11}$$

T is a temporary variable that controls the distillation level of the network. When the temporary variable T is set to 1, the output is the output of the normal softmax layer. When T becomes larger, the variance of the probabilities of the output classes of the Softmax layer decreases. When T becomes smaller, the difference in the probabilities of the output classes of the Softmax layer will increase.

In [15], Papernot et al. proposed that network distillation can not only compress the network model, but also make the distilled model obtain a certain robustness. Papernot et al. found that neural networks are sensitive to the input of adversarial examples, and network distillation can reduce the sensitivity of the model. Based on this discovery, N. Papernot et al. used a higher T for network distillation, which improved the robustness of the neural network while retaining the original accuracy of the model. At the same time, the generalization of the neural network was improved.

3.2 Adversarial Training

The core idea of adversarial training is to add adversarial examples to the model training process to increase the robustness of the neural network model. Goodfellow et al. [2] and Huang et al. [16] introduced adversarial examples in the training phase, so that training with adversarial examples can

make the model more robust. Goodfellow et al. [2] found that introducing adversarial examples into the training phase has a certain regularization effect, and even the regularization effect is better than dropout.

After the concept of adversarial examples was put forward, although many attack algorithms against adversarial examples emerged, there was little work on in-depth analysis of how to improve the robustness of the model. In [5], Madry et al. summarized the attack of adversarial examples and the robustness of the model into a common theoretical system, and summarized the attack and defense of adversarial examples into a saddle-point optimization formula, which can describe for:

$$\min_{\theta} \ \rho(\theta), \ where \rho(\theta) = \mathbb{E}_{(x,y)\sim D}\left[\max_{\delta \in S} \ L(\theta, x+\delta, y)\right] \tag{12}$$

where x is the natural example, δ is the perturbation information, S is the set of perturbation information, y is the label of the natural example, D is the distribution satisfied by the natural data (x, y), and θ is the parameter of the neural network. This formulation can be viewed as a saddle point problem, i.e., the inner maximization and outer minimization. The purpose of internal maximization is to find adversarial examples that can maximize the loss of the model, while the purpose of external minimization is to find appropriate network parameters to minimize the loss obtained from the output of adversarial examples into the model. Eq. (12) enables us to place attack and defense in a common theoretical framework, and naturally encapsulates most of the previous work on adversarial examples, so Eq. (12) is supported by a large number of subsequent works on adversarial defense.

In order to solve the min-max problem proposed by Eq. (12), Madry et al. proposed to use the adversarial examples generated by PGD attack for training. The experimental results show that as the number of iterations increases, the loss of the model on adversarial examples becomes smaller and smaller, and a more robust model is finally obtained.

Madry et al. proposed an adversarial training method capable of training robust models, breaking the dilemma that adversarial training can only defend against single-step attacks. However, due to the need to construct adversarial examples through the PGD attack method during the training process, it usually takes longer than traditional training. Shortening the cost of adversarial training becomes particularly important, especially for some tasks that are originally expensive to train.

Based on this consideration, Wong et al. [17] proposed fast adversarial training, which is an improved version of fast gradient symbolic method adversarial training. Wong et al. in [17] demonstrated that adversarial training combining fast gradient notation with random initialization can significantly reduce the computational cost and training time, while the training effect is comparable to PGD-based adversarial training.

To trade off adversarial robustness with accuracy, Zhang H et al. proposed a defense method TRADES [18]. Zhang et al. added KL divergence as a regular term to adversarial training, emphasizing the consistency of neural network output on natural examples and adversarial examples, so that adversarial training does not only rely on predicted labels. Zhang et al. used this method as the basis for participating in the NeurIPS 2018 Adversarial Vision Challenge, and out of about 2000 entries, won the first place, outperforming the runner-up method by 11.41% in terms of average L_2 norm perturbation.

3.3 Preprocessing

Adversarial examples have different intrinsic characteristics from natural examples. Compared with the characteristics of natural examples, the interference features of adversarial examples are

less robust, so these adversarial perturbations can be eliminated by preprocessing. The preprocessing methods mainly include: using traditional filtering or discretization methods for input examples, or using deep learning denoising methods; reconstructing the input and mapping the adversarial examples into the manifold of natural examples.

Xie et al. [19] visually compared the feature maps obtained from adversarial examples and natural examples. Xie et al. found that there is a lot of noise in the feature maps obtained from adversarial examples, even in areas without relevant semantic information in the image. Activation information is found in the feature map. Based on this finding, Xie et al. proposed a feature denoising method, they designed a feature denoising module, added it to the neural network, and then performed end-to-end training.

Goodfellow et al. [2] proposed that the reason for adversarial examples is the high linearity of neural networks in high-dimensional space, and Buckman et al. [20] proposed a thermometer encoding defense based on this assumption. Thermometer-encoded defenses quantify and discretize input examples, which can effectively remove adversarial perturbations.

Adversarial examples can be converted into natural examples by reconstruction, and the reconstructed examples are no longer adversarial and will not affect the classification performance of the neural network. Gu and Rigazio found that adding noise can destroy adversarial examples, and denoising encoders can remove adversarial noises, but stacking denoising encoders with the original neural network is still vulnerable to adversarial examples. Gu et al. proposed a variant of the penalized autoencoder network called deep contractive autoencoder [21]. The denoising autoencoder network can learn the invariant features of each layer of the neural network and encode the adversarial examples as raw examples to remove their adversarial perturbations.

The basic idea of preprocessing defense methods is to preprocess input examples to remove adversarial noise in the input examples, but because this idea often relies on gradient masking, it can be bypassed by BPDA attacks [22]. The advantage of this type of approach is that it can be combined with adversarial training to achieve better results than adversarial training alone.

3.4 Adversarial Detecting

Although methods for improving model robustness have been widely recognized, such defense methods require retraining the network, face huge computational overhead, and are difficult to apply in practical applications. The adversarial attack detection method selects whether to send the input example to the target model by classifying whether it is an adversarial example. This method is computationally inexpensive and does not require changing the structure of the network.

Xu et al. [23] believe that the dimension of input features is too large, making it easy for adversaries to attack successfully. Xu et al. proposed an adversarial example detection method based on feature compression, which reduces the search space for adversarial examples to be generated by compressing unnecessary features. The detection idea is to compare the predicted probabilities of the model before and after compression of the input examples. If the difference between the predicted probability of the input example before and after compression exceeds a certain threshold, the example is considered to be an adversarial example, and the input of the example is rejected.

Pang et al. [24] proposed the use of Reverse Cross-Entropy (RCE) loss function for adversarial example detection, which is easy to implement and hardly increases the training cost. The inverse cross-entropy loss function proposed by PANG et al. encourages the neural network to learn the latent features of the examples, so that it can better distinguish between natural examples and adversarial examples. PANG et al. adopted a threshold strategy for detecting adversarial examples, which can

filter out adversarial examples to improve the robustness of the neural network. PANG et al. tested on the MNIST and CIFAR10 datasets, and the robustness of the neural network in the face of various adversarial attack methods has been significantly improved.

4 Conclusion

This paper introduces the current mainstream adversarial attacks and adversarial defense methods. We sort out the representative and classic adversarial example generation methods, and classify and analyze the defense methods of adversarial examples. We analyze the characteristics of classical adversarial example generation methods, and point out the advantages and disadvantages of different adversarial example defense methods. The related research on adversarial attacks and defenses methods provides a new perspective for improving the interpretability of neural networks, and can promote the further development of artificial intelligence security field.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Szegedy, W. Zaremba and I. Sutskever, "Intriguing properties of neural networks," arXiv preprint arXiv:1312.6199, 2013.
[2] I. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014.
[3] A. Kurakin, I. Goodfellow and S. Bengio, "Adversarial examples in the physical world," arXiv preprint arXiv:1607.02533, 2016.
[4] D. Yinpeng, "Boosting adversarial attacks with momentum," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 9185–9193, 2018.
[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017.
[6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik *et al.,* "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symp. on Security and Privacy (EuroS&P)*, Saarbruecken, Germany, pp. 372–387, 2016.
[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symp. on Security and Privacy (sp)*, San Jose, CA, USA, pp. 39–57, 2017.
[8] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. of the 10th ACM Workshop on Artificial Intelligence and Security*, Dallas, TX, USA, 2017.
[9] S. -M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, 2016.
[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley *et al.,* "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.
[11] C. Xiao, B. Li and J. Zhu, "Generating adversarial examples with adversarial networks," arXiv preprint arXiv:1801.02610, 2018.
[12] S. -M. Moosavi-Dezfooli, A. Fawzi and O. Fawzi, "Universal adversarial perturbations," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Venice, Italy, 2017.
[13] Z. Zhao, Z. Liu and M. Larson, "Towards large yet imperceptible adversarial image perturbations with perceptual color distance," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.

[14] G. Hinton, O. Vinyals and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531 2.7, 2015.
[15] N. Papernot, P. McDaniel, X. Wu, S. Jha and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symp. on Security and Privacy (SP)*, IEEE, San Jose, CA, USA, pp. 582–597, 2016.
[16] R. Huang, B. Xu, D. Schuurmans and C. Szepesvári, "Learning with a strong adversary," arXiv preprint arXiv:1511.03034, 2015.
[17] E. Wong, L. Rice and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," arXiv preprint arXiv:2001.03994, 2020.
[18] H. Zhang, Y. Yu, J. Jiao and E. Xing, "Theoretically principled trade-off between robustness and accuracy," in *Int. Conf. on Machine Learning*, PMLR, Long Beach, California, 2019.
[19] C. Xie, Y. Wu and L. Maaten, "Feature denoising for improving adversarial robustness," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, California, 2019.
[20] J. Buckman, A. Roy and C. Raffel, "Thermometer encoding: One hot way to resist adversarial examples," in *Int. Conf. on Learning Representations*, Vancouver, Canada, 2018.
[21] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," arXiv preprint arXiv:1412.5068, 2015.
[22] A. Athalye, N. Carlini and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Int. Conf. on Machine Learning*, PMLR, Stockholm, Sweden, 2018.
[23] W. Xu, D. Evans and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," arXiv preprint arXiv:1704.01155, 2017.
[24] T. Pang, C. Du and J. Zhu, "Robust deep learning via reverse cross-entropy training and thresholding test," arXiv preprint arXiv:1706.00633, 2017.