Journal on Big Data
DOI:10.32604/jbd.2020.012294
Article

Tech Science Press

A Survey on Adversarial Examples in Deep Learning

Kai Chen[1,*], Haoqi Zhu[2], Leiming Yan[1] and Jinwei Wang[1]

[1]School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[2]School of Atmospheric Sciences, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[*]Corresponding Author: Kai Chen. Email: 20178314013@nuist.edu.cn

Received: 15 March 2020; Accepted: 15 July 2020

Abstract: Adversarial examples are hot topics in the field of security in deep learning. The feature, generation methods, attack and defense methods of the adversarial examples are focuses of the current research on adversarial examples. This article explains the key technologies and theories of adversarial examples from the concept of adversarial examples, the occurrences of the adversarial examples, the attacking methods of adversarial examples. This article lists the possible reasons for the adversarial examples. This article also analyzes several typical generation methods of adversarial examples in detail: Limited-memory BFGS (L-BFGS), Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Iterative Least-likely Class Method (LLC), etc. Furthermore, in the perspective of the attack methods and reasons of the adversarial examples, the main defense techniques for the adversarial examples are listed: preprocessing, regularization and adversarial training method, distillation method, etc., which application scenarios and deficiencies of different defense measures are pointed out. This article further discusses the application of adversarial examples which currently is mainly used in adversarial evaluation and adversarial training. Finally, the overall research direction of the adversarial examples is prospected to completely solve the adversarial attack problem. There are still a lot of practical and theoretical problems that need to be solved. Finding out the characteristics of the adversarial examples, giving a mathematical description of its practical application prospects, exploring the universal method of adversarial example generation and the generation mechanism of the adversarial examples are the main research directions of the adversarial examples in the future.

Keywords: Adversarial examples; generation methods; defense methods

1 Introduction

In 2012, in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), deep learning began to emerge [1]. In recent years, deep learning has developed rapidly, and its application scope has been further expanded [1–2], the network structure is more complicated [2–3]. The training method has been improved, and the application of some important techniques has further improved the classification performance and reduced the training time [2–5]. For example, in the field of image recognition, experimental results on some standard test sets indicate that the recognition capabilities of deep learning models can already reach the level of human intelligence. However, while deep learning brings great convenience to people, it also has some security problems. Hidden security issues have gradually attracted the attention of security experts. Therefore, many scholars have begun to pay attention to the anti-interference ability of deep learning models which the research on deep learning adversarial example problems.

Early in the security field that applied deep learning algorithms such as spam detection systems and intrusion detection systems, the problem of evading detection based on the characteristics of the system model was discovered, which brought great challenges to the security detection in deep learning. Up to today, more and more problems that threaten the security of deep learning have been discovered. There are illegal authentication hazards that mimic the identity of victims against the defects of the Face Recognition System (FRS), there are also privacy theft hazards involving medical data and people's picture data, and malicious control hazards against autonomous vehicles and voice control systems [6].

Therefore, the problem of adversarial examples in deep learning is more and more worthy of attention. The causes and methods of adversarial examples are the key issues in the study of adversarial examples. Using adversarial examples for adversarial training and improving the robustness of the system and the security of deep learning are imminent. Although it has been speculated that the reason for the adversarial examples is the highly non-linear features of the deep neural network and the insufficient average model and insufficient regularization in supervised learning, Goodfellow pointed out that the linear rather than nonlinear high-dimensional space is the real reason for the adversarial example [7]. At present, the main generation methods of adversarial examples are: L-BGFS, FGSM, BIM, LLC and other methods. Finding the attack features and attack methods of the adversarial examples is the core of our problem solving. From the application scenarios, the attack methods are mainly divided into two types, one is the black box attack, and the other is the white box attack [2]. The ability of adversarial examples to have black box attacks is due to the transferability of adversarial examples [5]. Goodfellow proposed that the generalization ability of adversarial examples in different models is caused by the high degree of consistency between anti-interference and model weights. Therefore, when training the same task, the adversarial examples can learn similar functions on different models [8]. Exploring different defense algorithms for adversarial example attacks is the main goal of studying adversarial examples. Currently, the defense techniques for adversarial examples can be divided into four categories: regularization method, adversarial training method, distillation method, and rejection option method. Although these methods can resist adversarial examples attacks to a certain extent, these methods cannot be applied to all models, so researching more powerful algorithms to defend adversarial example attacks is the main research direction in the future.

2 Basic Issues to Adversarial Examples

2.1 Deep Learning

Deep learning [1] is a branch of machine learning whose main purpose is to automatically learn effective feature representations from data. The deep learning model learns different neural networks through training, and uses the feature conversion between the internal levels of the neural network to abstract the original data into a higher-level feature representation. Figure 1 shows the data processing flow of deep learning. The existing deep neural networks mainly include the following types: Deep neural network (DNN), convolutional neural network (CNN), generative adversarial network (GAN), recurrent neural network (RNN), auto encoder (AE), etc.

2.2 The Concept of Adversarial Example

By deliberately adding imperceptible perturbations to the input examples in the data set, the model gives a wrong output with high confidence. Only a small disturbance on a picture needed, the classifier misclassifies the picture with high confidence, and even classifies it into a specified label (not the label to which the picture belongs correctly) [5].

The example usually contains a pair of label-example. Since the adversarial example has predicted its class label at the time of generation, the adversarial example is as if it contains implicit class labels. Therefore, the example is called the adversarial example in this article to emphasize the adversarial features.

To put it simply, there is a learning system *M* and clean input example (no noise added into the example) *C*. We assume that example *C* is correctly classified by the learning system, which *M(C)* is true. Then we build an example A that is almost the same as example *C* but noise is added into it. The example A will be misclassified, which makes *M(A)* is not true. The example *A* is called an adversarial example.

Although the probability of adversarial interference is much smaller than that of noise interference, the probability of being misclassified by the classifier is much higher than that of noise interference. In addition, models with different structures trained on different subsets of the training set will all misclassify the same adversarial example. Adversarial example has become a blind spot for the training algorithm. Biggio elaborated the related concepts of the adversarial example [9], and proposed an adversary model, an adversary's goal, adversary's knowledge and adversary's capability.

In order to stimulate the optimal attack strategy of adversarial example, we need to know the knowledge required for the attack and the ability of the opponent to manipulate the data, and finally use the adversarial model to modify the potential data distribution and use the anti-attack ability to generate the optimal attack strategy. We use the general model of the adversary to clarify some concepts of adversarial examples.

2.3 Cause of Adversarial Example

Attackers always add as little perturbations as possible to in the original image to generate adversarial examples. On the one hand, too much perturbation to the original image may be easier to detect; on the other hand, the image with too much perturbation may no longer be an adversarial example because its essence will change. Goodfellow believe that the classifier is too linear to lead to the existence of adversarial examples [7], and the area between the real boundary and the classifier boundary is called the adversarial regions (Adversarial Regions) [10].The schematic diagram of the adversarial example is shown in Fig. 1, where the straight line represents the classification boundary of the model and the curve represents the true boundary. If the image b slightly crosses the classification boundary of the model and is in the adversarial area, it is an adversarial example; conversely, if the image is added too much perturbation and crosses the adversarial area, it is not an adversarial example. However, the true boundary is often difficult to determine accurately.

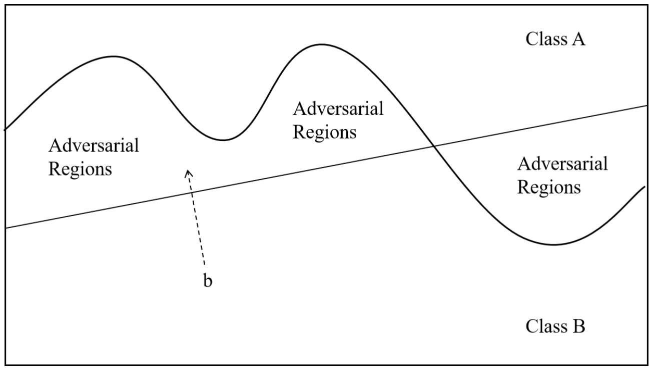

Figure 1: Adversarial examples

Neural networks are susceptible to be fooled. When the adversarial examples appears, the neural network will treat an unrecognizable image as an image of a recognized class, which exposes the blind spots of the machine learning algorithm, indicating that there are hidden features and blind spots in the process of backpropagation, showing the output of the neural network is associated with the data distribution in an unobvious way. The cause of the adversarial example is a mystery. People have speculated that the reason for the adversarial example is the highly non-linear network structure of the deep neural networks, and the insufficient average and regularization of the model in supervised learning. Later, Goodfellow proposed that the linear feature of the high-dimensional space rather than the non-linear features are the real reasons for the existence of the adversarial examples. Szegedy proposed that the high-level neural network space, rather than a single neuron, contains higher-level abstract semantic information of the neural network. At the same time, Szegedy found that the input and output mapping of the deep learning network is discontinuous. The reason why the neural network misclassifies the adversarial examples may be that the probability that the neural network will correctly classify the adversarial examples is extremely low, so it is difficult to observe the adversarial examples in the test set [5]. Therefore, it is necessary to further explore how to improve the probability that the neural network correctly classifies the adversarial examples.

2.4 Adversarial Example Attack

Wang Haibing, director of the laboratory focused on Internet security, said: "Attacking artificial intelligence with adversarial examples is actually to attacking it from the core algorithm level." The adversarial example attack method is mainly divided into two types from the application scenario. One is white box attack; the other is black box attack [2]. The white box attack refers that the attacker fully understands the target classifier. The attacker knows the target of the classifier, the type of classifier, the algorithm that the training model learned and the parameters used by the algorithm of classifier. What's more, the attacker can interact with the machine learning system in the process of generating adversarial examples. Black box attack means that the attacker knows the target's features and type of classifier, but does not know the classifier form or training data. Even though, the attacker can still interact with the machine learning system. For example, you can observe and judge the output of the classifier by inputting data.

2.5 Transferable Features of Adversarial Examples

The adversarial example has transferability is first proposed by Szegedy [5]. The adversarial example's transferability means that the adversarial example is misclassified by model M1, and can also be misclassified by model *M2.* The transferability of the adversarial example means that the attacker can choose to attack a machine learning model without directly touching the basic model, so that the example is misclassified. Szegedy studied the transferability of different models on the same data set. In addition, they also trained the same or different models on disjoint subsets of the data and studied the transferability between them. But the disadvantage is that the experimental results of Szegedy and others are all realized on the MNIST dataset. Goodfellow proposed that the generalization of adversarial examples between different models is due to the fact that the adversarial interference is highly consistent with the vector of the model, so when training the same task, the opponent can learn similar functions on different models. This generalization feature means that if the attacker wants to attack the model, there is no need to have access to the target model, just sending the adversarial examples generated by the local model to the target model.

3 Generation Methods

Adversarial examples are the key to evaluate and improve the robustness of machine learning. Therefore, studying how to generate adversarial examples is a necessary step for studying adversarial examples. There are many ways to generate adversarial samples. At present, the main generation methods include L-BFGS, FGSM, BIM, etc.

3.1 L-BFGS (Limited-Memory Broyden-Fletcher-Goldfarb-Shanno)

There are many ways to generate adversarial example, including calculating the gradient on the image pixels or directly solving the optimization problem of an objective function on the image pixels. Szegedy proposed the L-BFGS method in literature [5]. In the optimization process, an input image that can be correctly classified is slightly disturbed so that it is no longer correctly classified. In a sense, this method is to optimize the traversal of the manifold network representation and find the adversarial example in the input space. The adversarial example exists in a low probability area in the manifold space, so it is difficult to obtain by simple random sampling near the input point.

At present, various latest computer vision models use input transformations during training to improve the robustness and convergence speed of the model. However, for a given example, these transformations do not affect the statistical results. The example transformations are highly correlated and are transformed from the same probability distribution of the entire model training data. Therefore, Szegedy proposed that the L-BFGS method revolves around the training data, using the model and its shortcomings to establish local Spatial.

Table 1: Typical adversarial example construction methods

	Derived Features	Attack Target	Iteration	Attack Mode	Scope of Application
L-BFGS	Optimized search	Targeted	Multiple	White box	Special
Deep Fool	Optimized search	Non-targeted	Multiple	White box	Special
UAP	Optimized search	Non-targeted	Multiple	White box	Universal
FGSM	Feature construction	Non-targeted	Single	White box	Special
BIM	Feature construction	Non-targeted	Multiple	White box	Special
LLC	Feature construction	Non-targeted	Multiple	White box	Special
JSMA	Feature construction	Targeted	Multiple	White box	Special
PBA	Feature construction	Targeted&Non-target	Multiple	Black box	Special
ATN	generative model	Targeted&Non-target	Multiple	Black&White box	Special

3.2 FGSM (Fast Gradient Sign Method)

Goodfellow proposed a fast gradient sign method which is one of the simplest methods to generate adversarial examples [7]. Its core is to move the input image toward the direction of reduced category confidence. $x \in R^m$ is input image, y is the class label corresponding to input x , θ is the model parameter, η is the step size, ε is the selected hyperparameter, $J(\theta, x, y)$ is the loss function of the training neural network, $\nabla_x J(\theta, x, y)$ is the partial derivative of the loss function. We can linearize the loss function around the daily current value to obtain the maximum norm limit of interference:

$$\eta = \varepsilon\ sign(\nabla_x J(\theta, x, y)) \tag{1}$$

Then the adversarial example is obtained by solving the following formula:

$$\widetilde{x} = x + \eta \tag{2}$$

The adversarial perturbation superimposed on the input of a typical picture will cause the illusion of the classifier and mistakenly identify the panda as a gibbon. In computation, this method has a huge advantage, because only one forward and one backward gradient calculation can produce adversarial examples. The FGS method is very simple and easy to implement with any framework, so it is usually used in this way to generate adversarial examples.

3.3 BIM (Basic Iterative Method)

In fact, in most cases, the adversarial example is generated by the FGS method alone is invalid, because the difference between the two categories is too large. The most extreme case is that a category may be in a “dead zone” where the Rectified Linear Unit (ReLU) is less than zero. If you consider the above two situations, you need to study a better and more practical method than the FGS method. If the FGS method adopts a direct direction toward the direction of reduced confidence in the category, the step may be wrong, so it is more appropriate to refer to the idea of gradient descent and step-by-step selection and advancement. Although this iterative process cannot achieve linear iteration in the gradient direction and requires multiple calculations, it is still simpler and better than the L-BFGS method.

Based on this, Kurakin improved the FGS method and proposed a more direct method: Basic Iterative Method (BIM). This method uses the idea of gradient descent and advances iteratively step by step. The fast gradient method is used multiple times with a small step size, and the pixel values of the intermediate results of each step are trimmed to ensure that they are in a neighborhood of the original image [7]. This method is further subdivided into two types: (1) Reduce the confidence that the classifier predicts that the example once belonged to the category; (2) Increase the confidence that the example was once predicted as the smallest possible category:

$$\begin{cases} X_0^{adv} = X \\ X_{N+1}^{adv} = Clip_{X,\varepsilon}\{X_{N+1}^{adv} + \alpha sign(\nabla_X J(X_N^{adv}, y_{true}))\} \end{cases} \tag{3}$$

In the above formula, X is a 3-D input image. Assuming that each pixel intensity value is an integer value between [0,255]. y_{true} is the real class label of image X, and $J(X, y)$ is the cross-entropy cost function of the neural network on the given image X and label y. α is the step size which usually equals one, and the number of iterations is selected as $\min(\varepsilon + 4, 1.25\varepsilon)$. For the output layer of the neural network, the cross-entropy cost function applied to integer class labels is equivalent to solving the negative logarithmic conditional probability on the real class labels of a given image: $J(X, y) = -\log p(y \mid X)$; $Clip_{X,\varepsilon}\{X'\}$ is the cropping function of the X' pixels of the image which ensures that the cropped image is still in the neighborhood of the original image.

However, the above method only attempts to increase the loss value of the correct classification. It does not clearly indicate which wrong class label should be selected by the model. These methods are only suitable for data sets with few types and different types from each other (such as MNIST and CIFAR-10).

3.4 LLC (iterative Least-likely Class Method)

The data set ImageNet contains 3000 different classes, and there are different degrees of differences between the classes. The above methods are only suitable for the case where the number of classifications is relatively small, such as mistakenly dividing a kind of dog into another. In order to generate more error classes, Kurakin introduced the Iterative least-likely class method [7]. This iterative method attempts to divide the adversarial examples into a specific target class. Based on the training depth on image X, the neural network obtains the predicted result and selects the smallest possible class.

$$y_{LL} = \arg_y \min\{p(y \mid X)\} \tag{4}$$

For a trained classifier, the adversarial examples generated by the smallest possible class method are usually completely different from the real class, so this attack method will lead to misclassification, such as mistaken dogs for airplanes.

The iterative least-like class method is an improvement of the iterative method, which maximizes $sign\{-\nabla_x \log p(y_{LL} \mid X)\}$ by iterating in the direction $sign\{-\nabla_x \log p(y_{LL} \mid X)\}$, so that the adversarial image is classified as y_1. For the cross-entropy cost function of the neural network, the expression in the last step is equivalent to $sign\{-\nabla_X J(X, y_{LL})\}$. Therefore, the iterative generation process of the adversarial example has the following formulation:

$$\begin{cases} X_0^{adv} = X \\ X_{N+1}^{adv} = Clip_{X,\varepsilon}\{X_{N+1}^{adv} - \alpha sign(\nabla_X J(X_N^{adv}, y_{LL}))\} \end{cases} \tag{5}$$

3.5 Deep Fool Method

Moosavi-Dezfooli proposed an untargeted Deep Fool method [11]. They believed that the deep model has hyperplanes that can segment different classes of data. Deep Fool calculates the minimum perturbation from the binary classification model. The minimum perturbation is the shortest distance from the current input point to the segmented hyperplane, thereby they can derive the perturbation generation method under the binary classification task and extend it from binary classification to multi-classification. As shown in Fig. 2, in the linear binary classification model $f(x)$, the minimum perturbation of changing the classifier decision is the orthogonal projection of the example point x_0 to the cut hyperplane

$F=\{x:w^{T}\cdot x+b=0\}$.

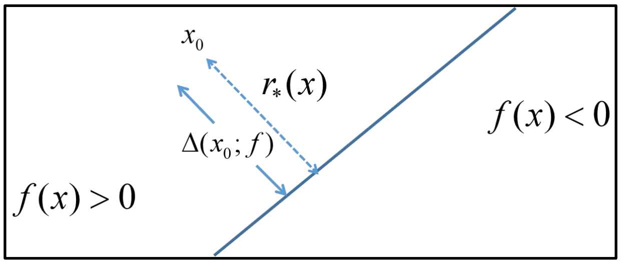

Figure 2: Minimum disturbance distance in a binary classification model

The specific analytical formula is

$$signf(x_0+r)\neq signf(x_0) \tag{6}$$

The constraint that the equation must satisfy is:

$$r_*(x_0)=\arg\min\|r\|_2=-\frac{f(x_0)}{\|w\|_2^2}w \tag{7}$$

$r_*(x_0)$ is the distance from the current point x_0 to the dividing hyperplane. The Deep Fool method uses an iterative process to solve for the minimum disturbance and promotes it to a more general binary classification. On this basis, the algorithm is further extended to a multi-classification model. The input is mapped to a hyperplane P surrounded by multiple decision surfaces. Similar to the case of the binary classification, they choose the hyperplane closest to P and project on its surface to solve the minimum disturbance. The Deep Fool method has relatively few changes to the original input. At the same time, the generated adversarial examples have a better attack effect and the calculation amount is relatively high.

3.6 UAP (Universal Adversarial Perturbations)

Moosavi-Dezfooli further proved the existence of universal adversarial perturbations across data and network architecture [12]. Such perturbations can lead to misclassification of different pictures and the generalization characteristics across models. This method performs an iterative version of Deep Fool attack on all pictures in the training set until it finds a disturbance that can deceive most of the training set. Formally, for a input x satisfying the distribution μ , the algorithm searches for the universal perturbations η with an upper limit of ξ as shown in Eq. (8).

$$\eta:\|\eta_P\|\leq\xi \tag{8}$$

The constraint that the equation must satisfy is:

$$\underset{x-\mu}{\overset{p}{}}(f(x+\eta)\neq f(x))\geq 1-\delta \tag{9}$$

By traversing the training data set multiple times, the algorithm can find a variety of universal perturbations and can deceive the deep neural network with high precision.

3.7 JSMA (Jacobian-Based Saliency Map Attack)

Papernot et al. proposed that the JSMA method uses a Jacobian matrix to evaluate the sensitivity of the model to each input feature [13], and finds the significant pixels in the whole picture that are conducive to the realization of attack targets. This method finds the salient points by calculating the positive derivative (Jacobi matrix) to find the input characteristics that cause a significant change in the DNN output. In contrast to the FGSM calculation of the reverse gradient, this algorithm modifies one original image pixel at a time and monitors the impact of changes on the classification results. The saliency list is calculated by using the gradient output of the network layer. Once the saliency list is calculated, the algorithm will select the most effective pixel to deceive the network. The JSMA method modifies the original input. At the same time, the calculation process is relatively simple because the JSMA method uses forward propagation to calculate the salient points.

3.8 PBA (Practical Black-box Attacks)

Papernot et al. attacked a remotely hosted deep neural network model for the first time in the case of black boxes [13,14]. The adversary knowledge of the neural network model is limited to the input and output of the deep learning model. The adversary views the target neural network structure as an oracle machine. In order to train an alternative model, a batch of random data is first constructed, and a synthetic example is constructed by querying the oracle machine. Using synthetic examples, the adversary trains an alternative model to simulate the input and output of the original neural network, and to imitate the decision boundaries of the original model [15]. After training the substitute model with synthetic data, based on the existing substitute model, the FGSM method was used to generate adversarial examples, which attacked the deep learning models hosted by MetaMind, Amazon, and Google, and generated more than 80% of the misclassifications. The construction of the PBA method is relatively complex, but it is more difficult to defend in compare with other construction methods. In 2018, Ilyas proposed that adversarial attacks in the real world have more restrictions than black-box attacks [16]. He defined three threat models that are closer to real-world scenarios, overcome the status of the number of queries, and successfully attacked the deep learning API hosted by Google.

3.9 ATN (Adversarial Transformation Network)

Baluja proposed a feed-forward neural network that trains an adversarial transformation network by self-supervised learning, and converts the input into an adversarial example [17]. ATN minimizes the output of the classifier when it gave the original input, while constraining the new classification to match against the target class. Reference shows the application of ATN in white box and black box scenarios, and analyzes its effectiveness for various classifiers.

4 Defense Methods

Traditional techniques that make machine learning models more robust, such as weight decay, usually cannot effectively defend adversarial examples. Machine learning models are often interfered by adversarial examples, which are maliciously interfered with input in order to mislead the model during testing. Adversarial examples are a security threat to the actual deployment of machine learning systems. It is worth noting that these inputs are transformed between models, thereby implementing black box attacks on the deployed models. The defense techniques against adversarial samples mainly include: Preprocessing, regularization and adversarial training, distillation method, rejection option, etc., as shown in Fig. 3.

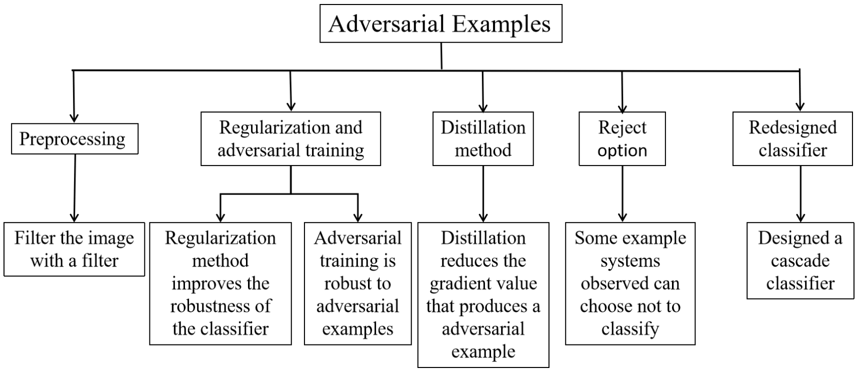
Adversarial Examples
Preprocessing
Regularization and adversarial training
Distillation method
Reject option
Redesigned classifier
Filter the image with a filter
Regularization method improves the robustness of the classifier
Adversarial training is robust to adversarial examples
Distillation reduces the gradient value that produces a adversarial example
Some example systems observed can choose not to classify
Designed a cascade classifier

Figure 3: The defense methods of adversarial example

4.1 Preprocessing

Natural images have some special properties, such as high correlation between adjacent pixels and low energy in the high-frequency domain. Assuming that this antagonistic change does not exist in the same space as the natural image, some work considers using filtering. The filter filters the image to remove the interference of adversarial examples. Although preprocessing the input makes the attack more challenging, it does not eliminate the possibility of being attacked. In addition, these filters usually reduce the classification accuracy of the data that is not subject to interference.

4.2 Regularization and Adversarial Training

Some studies have proposed regularization methods [18,19] and adversarial training methods [20-21] to improve the robustness of the classifier. Not only can the adversarial examples be continuously updated to attack the current model, but also the model with adversarial examples can be used to train the neural network to reduce the error rate. That is to say, this model can resist the adversarial examples to some extent, so through adversarial training can improve the deep learning to some extent Anti-jamming ability against adversarial examples. The adversarial examples were originally a system vulnerability, but we can use it to become a means for humans to confront the adversarial examples. The adversarial example is not limited to a specific neural network, so there is no need to obtain the source code of the model to make the adversarial examples. As long as the model is trained to perform the same task, they will be "spoofed" by the same adversarial examples while these models have different structures or use different training examples. Therefore, as long as people design a model and generate corresponding adversarial example, they can use them to attack those artificial intelligence algorithms for similar tasks. It is difficult to use conventional methods to solve adversarial example problem. Some researchers have tried a variety of traditional methods, including averaging multiple models, averaging multiple judgments of the same image, noise-aware training and constructing a generative model, which cannot solve the problem of adversarial example. Specialized training can make the model more resistant, but it can't really eliminate blind spots. Although it is difficult for people to be deceived by these examples, sometimes we can be deceived in unexpected places. Psychology has provided a vast array of examples of visual illusions. These illusions can be considered as "adversarial example" for humans. The adversarial examples facing neural networks and the adversarial examples facing humans do not completely overlap. Both humans and machines make mistakes, but the mistakes are different.

Kurakin further explored adversarial learning [22], showing that adversarial training is robust to adversarial examples, and made suggestions for adversarial training of large models and data sets. In addition, he proposed that when performing adversarial example training, improving the performance of the model can increase the robustness of the adversarial example. Different degrees of interference of adversarial examples will have different adversarial strength to resist the attack. Based on this phenomenon, Song combined the adversarial training examples and different adversarial strength to propose Multi-strength Adversarial training method (MAT) to mitigate the attack of adversarial examples [16]. Tramer pointed out that the model after adversarial training is still vulnerable to attack [17], because the discriminant hyperplane of the model changes obviously near the data points, which hinders the first-order approximate attack based on model loss. But it cannot resist black box attacks from adversarial example migration. Song further promoted adversarial training and proposed ensemble adversarial training. This method improves the training effect by adding the interference input of certain fixed pre-training models to the training data.

However, experiments have shown that the classification error rate based on adversarial example training is 1% lower than the classification error rate only on undisturbed data [23], which means that when these learning systems are applied in practice, they cannot completely rely on the defense mechanism. The experiments show that when the model is unknown, we can also successfully attack the robust learning system. Therefore, these methods cannot effectively resist the adversarial example attack.

4.3 Distillation Method

The distillation method is a method proposed by Hinton to imitate a large model with a small model. The basic idea is training the one-hot vector output by the classification model which called a hard target. After training a model with a hard target, not only does maximum value is retained, but also the entire probability vector is used as a target (called a soft target). In this way, not only each input example has a one-hot vector with less information, but also a vector with a certain probability for each category. The training network will get some additional information in this way. If a picture is difficult to distinguish between two categories, there will be a higher probability. Such a label actually comes with the information obtained by training the large model, so it can improve the the performance of the small model.

Papernot proposed a distillation defense mechanism against adversarial examples for deep neural networks, and verified the effectiveness of their defense mechanisms on two types of deep neural networks [19]. This is because distillation reduces the gradient value when generating adversarial examples and increases the minimum average value of the modified feature example required to generate adversarial example [13]. Carlini pointed out that the adversary can recover the effect of distillation by accessing the classifier parameters, and studied the application of distillation in the black box and unknown models [24]. Carlini verified that the method cannot improve the robustness of the classifier and points out that the distillation method is not very effective [17]. By slightly modifying the distillation mechanism on the standard attack, it is not effective enough to resist adversarial example attack. The distillation method can significantly reduce the gradient value of the loss function, but in the case of black box attack and unknown model function, the change of the eigenvalues cannot effectively resist the opponent’s attack [25].

4.4 Reject Option

Some scholars have studied reject option method [26–29]. They can choose not to classify certain example systems observed. For example, when the class condition posterior probability is close, the rejection option is selected. In the literature [29], the author pointed out that the correct classification examples tend to have larger maximum class condition posterior probabilities than examples that are misclassified and not in the probability distribution. Therefore, rejecting input examples by checking the corresponding class condition posterior probabilities is invalid for detecting adversarial examples. A similar method is to classify the invalid data as “garbage” [29–31]. In these methods, the classifier is trained to evenly distribute the invalid data, and then the examples are discarded with low confidence during the test. Although this this method improves the robustness of the adversarial example classifier, but it does not consider the probability of detecting example discard. In the literature [32], the author proposes a binary classifier to augment the classifier and use it distinguishes the adversarial example from the clean example, but the author does not give the error rate of the method. In addition, it is worth noting that the adversarial example not only “spoofing” classifier, but also can “sproof” detector, so this method is not very effective.

4.5 Redesigned Classifier

Unlike directly training deep neural networks to test adversarial examples, Li proposed a simpler scheme based on the output analysis of the convolutional layer, and designed a cascade classifier through a special adversarial generation mechanism [33]. The classifier can effectively detect adversarial examples at the same time. Experiments show that the adversarial examples can be reconstructed on the image through small average filtering. These findings also prompt us to think more about the classification mechanism of deep convolutional neural networks.

Fawzi focused on exploring the robustness of the classifier [12], which believed that there should be a trade-off between the performance and robustness of the classifier. Huang further explored this trade-off in practical applications [34], which regards the learning process as a min-max problem, and considers the neural network learning problem in the worst case, which allows the opponent to make different interferences at each data point. The learning process is in the expected interference minimize the loss error, and call this learning process “adversarial learning”.

5 Adversarial Example Application

Up to now, the application of adversarial samples is mainly used in adversarial evaluation and adversarial training.

5.1 Adversarial Evaluation

Adversarial evaluation helps the model analyze errors early and judge whether the model is successful. Smith and others mentioned that the central idea of adversarial evalution is to study different adversarial roles through different scholars, so that the division of labor for evaluating different roles is clear and maximizes the contribution [20]. Based on different scholars and their models, Smith proposed a new natural language evaluation model on different adversarial roles. Jia [35] mentioned that reading comprehension is a very attractive platform. Computer vision works by adding subtle anti-interference to the input image, but this subtle interference cannot change the real label of an image. However, changing a word in a sentence can completely change the meaning of the sentence. By adding a scattered sentence to the input picture, rather than adding a semantic retention to generate an adversarial example, the generated sentence can make the picture “confuse” model, but it is not consistent with the correct classification result Contradictory and cannot “confuse” humans. Jia mentioned that although the reading comprehension system is very successful in the standard evaluation system, but it performs poorly in adversarial evaluation. The standard evaluation is too tolerant of the literal prompt model. On the contrary, adversarial evaluation reveals that existing models are too stable for changing semantics. Feng pointed out recursion neural network encoder-decoder models have made significant progress in data-driven dialogue systems [36]. However, the evaluation of dialogue results is still a challenging problem. Adversarial loss is a direct evaluation of whether the generated dialogue results are more like human expressions. This method will reduce the need for humans to participate in dialog evaluation, and train recursive neural networks to distinguish between dialog model examples and human-generated examples. Although this method can be proved to be feasible, there are still many problems in practical applications.

5.2 Adversarial Training

Park improves the performance of the backpropagation algorithm by adding adversarial gradients to the training process [37]. Unlike changing the input without improving versatility, Park got a minimum set of discards by maximizing the difference between the output of the discarded network and the output of the supervised network. These confirmed discarded sets are used to retrain the neural network. Experiments have shown that training on the configured subnetwork can improve the generalization ability of the supervised and semi-supervised back learning models on the data sets MNIST and CIFAR-10 [38].

In actual classification tasks, it is difficult to collect training examples from all possible environmental categories. Therefore, when an example of an invisible class appears, a good classifier should be able to judge that it is an unknown class, rather than classify it as any known category. Yu used the idea of adversarial training to propose an adversarial example generation (ASG) framework for classifying uncertain classes. The strategy generates positive and negative examples of known categories in an unsupervised manner. For the generated examples, ASG distinguishes the classes they belong to in a supervised manner [39].

6 The Future Development of Adversarial Example

In summary, the study of adversarial examples is already a hot issue in the field of machine learning security, and finding the characteristics of adversarial examples, the generation mechanism of adversarial examples, and the attack methods of the adversarial examples are the key issues in studying the adversarial examples. Defensive algorithms that explore different adversarial example attacks are the main targets. Combining these two parts to solve the attack of the adversarial example is the main research direction of the adversarial example in the future.

(1) How to define the transferable of adversarial examples, how to measure the degree of transferable, and how to determine the upper and lower bounds of the transfer, so as to use the transfer to effectively generate adversarial examples, effectively detect the adversarial examples, and defend against the attacks of the adversarial examples.

(2) The reason and principle of the generation of adversarial examples on general data or specific data make the classifier unable to correctly identify the adversarial examples. Establish a complete mathematical and unified theory of adversarial example generation, and then realize a high probability guaranteed deep neural network implementation theory and form of defense adversarial example attacks, laying a theoretical foundation for the actual deployment of deep neural network machine learning algorithms.

(3) Construct a universal benchmark software platform for generating adversarial examples, so that the current research on adversarial examples can evaluate the experimental results on a unified standard data set [40].

7 Conclusion

The problem of adversarial examples is getting more and more attention. Discussing the reasons for the occurrence of adversarial examples and how to generate them are the key issues in the study of adversarial examples. This article first summarizes the causes of the adversarial examples and the latest research progress and points out that the current conjecture is not convincing. Further researches on the reasons for adversarial example deserve us to find out. Secondly, the main generation methods of the adversarial examples are the F-BFGS method, the FGSM, the base iterative method and iterative least-likely class method. The review pointed out their advantages and disadvantages and applicable scenarios. The purpose of studying the reasons for the occurrence of adversarial examples and the generation method is to protect the machine learning system from the attacks of adversarial example. At the end of this article, the main popular defense technologies that are currently based are preprocessing method, regularization method, adversarial training method, distillation method, reject option method, and other methods reviewed. This paper points out the application scenarios and deficiencies in different defense measures, explaining that none of the above defense measures can completely avoid adversarial example attacks. In summary, to further study the characteristics of the adversarial example, give a mathematical description of practical application prospect, explore universal adversarial example generation method, completely solve the adversarial attack problem, there are still a large number of theoretical and practical problems to be solved.

Funding Statement: This work is supported by the NSFC [Grant Nos. 61772281, 61703212]; the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

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

References

[1] S. loffe, C. Szegedy, "Batch normalization, accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, Lille, France, pp. 448–456, 2015.

[2] V. Mnih, K. Kavukcuoglu and D. Silver, "Human-level control through deep reinforcement learning," *Journal of Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] N. Srivastava, G. Hinton and A. Krizhevsky, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[4] K. He, X. Zhang and S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, USA, pp. 770–778, 2016.

[5] C. Szegedy, W. Zaremba and 1. Sutskever, "Intriguing properties of neural networks," in *Proc. ICLR*, pp. 1312–1320, 2014.

[6] P. Li, W. T. Zhao and Q. Liu and C. J. Wu, “Review of machine learning security and its defense technology,” *Computer Science and Exploration*, vol. 12, no. 2, pp. 171–184, 2018.

[7] J. Goodfellow, J. Shlens and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” in *Proc. ICML,* Beijing, China, pp.278–293, 2014.

[8] A. Kurakin, 1. Goodfellow and S. Bengio, “Adversarial examples in the physical world,” in *Proc. ICLR*, Toulon, France, pp. 1726–1738, 2016.

[9] B. Biggio, I. Corona and D. Maiorca, “Evasion attacks against machine learning at test time,” in *Proc. MLKD Databases*, Springer Berlin Heidelberg, pp. 387–402, 2017.

[10] P. McDaniel, N. Papernot and Z. B. Celik, “Machine Learning in Adversarial Settings,” *IEEE Security & Privacy*, vol. 14, no. 3, pp. 68–72, 2016.

[11] S. M. Moosavi-dezfooli, A. Fawzi and P. Frossard, “Deepfool: A simple and accurate method to fool deep neural networks” in *Proc. CVPR*, pp. 282–297, 2016.

[12] S. M. Moosavi-dezfooli, A. Fawzi and O. Fawzi. “Universal adversarial perturbations,” in *Proc. CVPR*, pp. 1765－1773, 2017

[13] N. Papernot, P. Mcdaniel and I. J. Goodfellow, “Practical black-box attacks against machine learning,” in *Proc. of the IEEE European Sym. on Security and Privacy*, Saarbricken, Germany, pp. 506－519, 2016.

[14] N. Papernot, P. Mcdaniel and I. J. Goodfellow, “Practical black-box attacks against deep learning systems using adversarial examples,” *Cryptography and Security*, 2016.

[15] Y. Senzaki, S. Ohata and K. Matsuura, “Simple black-box adversarial examples generation with very few queries,” *The Institute of Electronics, Information and Communication Engineers*, vol. E103–D, no. 2, 2020.

[16] A. Ilyas, L. Engstrom, A. Athalye and J. Lin, “Black-box adversarial attacks with limited queries and information,” in *Proc. ICML*, Stockholm, Sweden, pp. 237－245, 2018.

[17] S. Baluja and I. Fischer, “Adversarial transformation networks: Learning to generate adversarial examples,” in *Proc. CVPR*, HI, USA, pp. 2300–2309, 2017.

[18] Y. Liu, X. Chen and C. Liu, “Delving into transferable adversarial examples and black-box attacks,” in *Proc. ICLR*, pp. 1–14, 2017.

[19] N. Papernot, P. Mcdaniel and S. Jha, “The limitations of deep learning in adversarial settings,” in *Proc. of the IEEE European Sym. on Security and Privacy*, Saarbrucken, Germany, pp. 372–387, 2016.

[20] D. F. Smith, A. Wiliem and B. C. Lovell, “Face recognition on consumer devices: Reflections on replay attacks,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 4, pp. 736–745, 2015.

[21] B. Zhang, B. Tondi and M. Barni, “Adversarial examples for replay attacks against CNN-based face recognition with anti-spoofing capability,” *Computer Vision & Image Understanding*, vol. 197, no. 2, pp. 33–44, 2020.

[22] A. Kurakin, I. J. Goodfellow and S. Bengio, “Adversarial examples in the physical world,” in *Proc. ICLR*, pp. 1607–1617, 2016.

[23] Xiao, B. Li and J. Y. Zhu, “Generating adversarial examples with adversarial networks” in *Proc. IJCAL*, Macao, China, pp. 3805–3911, 2019.

[24] N. Carlini, P. Mishra and T. Vaidya, “Hidden voice commands,” in *Proc. USENIX*, Austin, USA, pp, 513–530, 2016.

[25] Xie, J. Wang and Z. Zhang, “Adversarial examples for semantic segmentation and object detection,” in *Proc. ICCV*, Venice, Italy, pp. 1378–1387, 2017.

[26] G. Fumera, F. Roli and G. Giacinto, “Reject option with multiple thresholds,” *Pattern Recognition*, vol. 33, no. 12, pp. 2099–2101, 2000.

[27] R. Herbei and M. H. Wegkamp, “Classification with reject option.” *Canadian Journal of Statistics*, vol. 34, no. 4, pp. 709–721, 2010.

[28] P. L. Bartlett and M. H. Wegkamp, “Classification with a reject option using a hinge loss,” *Journal of Machine Learning Research*, vol. 9, pp. 1823–1840, 2008.

[29] Cortes, G. Desalvo and M. Mohri, “Learning with rejection,” in *Proc. ICML*, Bari, ltaly, pp. 67–82, 2016.

[30] Bromley, J. Denker, "Improving rejection performance on handwritten digits by training with "Rubbish"," *Journal of Neural Computation*, vol. 5, no. 3, pp. 367–370, 1993.

[31] Y. Lecun, L. Bottou and Y. Bengio, "Gradient-based learning applied to document recognition," in *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[32] B. Yadav and V. S. Devi, "Novelty detection applied to the classification problem using probabilistic neural network," in *Proc. CIDM*, Orlando, USA, pp. 265–272, 2014.

[33] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proc. ICCV*, Venice, Italy, pp. 5775–5783, 2017.

[34] S. Aditya and S. Gandharba, "Reversible image steganography using dual-layer LSB matching," *Sensing and Imaging*, vol. 21, no. 1, 2020.

[35] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proc. EMNLP*, Copenhagen, Denmark, pp. 2021–2031, 2017.

[36] G. Feng, Q. J. Zhao, X. Li, X. H. Kuang, J. W. Zhang *et al.,* "Detecting adversarial examples via prediction difference for deep neural networks," *Information Science*, vol. 1, no. 1, pp. 501, 2019.

[37] S. Park, J. K. Park and S. J. Shin, "Adversarial dropout for supervised and semi-supervised learning," in *Proc. AAAI*, New Orleans, USA, pp. 219–231, 2018.

[38] Y. Yu, W. Y. Qu and N. Li, "Open-category classification by adversarial sample generation," in *Proc. IJCAI*, Melbourne, Australia, pp. 3357–3363, 2017.

[39] Russakovsky, J. Deng and H. Su, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[40] Z. Gong, W. Wang and W. S. Ku, "Adversarial and clean data are not twins," arXiv:1704.04960, 2017.