Proceedings of the 4th International Conference on Computing and Data Science (CONF-CDS 2022)
DOI: 10.54254/2755-2721/2/20220672

Improving transfer ability of adversarial sample based on adversarial transformation

Changlin Song[1,2]

[1]Graduation, School of Cyber Science and Engineering, Southeast University, Nanjing 210089, China

[2]cscl_song@qq.com

Abstract. In recent years, Deep neural networks (DNN) have caused a huge sensation, showing outstanding capabilities in the fields of computer vision tasks and natural language processing. It has been gradually discovered that DNN are easily disturbed by adversarial samples, which are formed by superimposing the original samples and tiny perturbations. Although these tiny perturbations are imperceptible to the naked eye, they can significantly interfere with the model output. In security-related fields, adversarial examples bring huge hidden dangers to the deployment of DNN systems. When testing and evaluating DNN systems, researchers usually study the robustness of deep neural networks using transfer-based attacks, which are black-box attacks using carefully crafted adversarial examples from the source model. Adversarial samples with strong transfer ability are more aggressive to black-box models, so how to improve the transfer ability of adversarial samples has attracted the attention of many scholars. Since the existing methods based on adversarial transformation to improve the transfer ability of adversarial samples are more complicated to train and cost to attack, this paper uses a cycle-consistent generative adversarial network to implement adversarial transformation, which reduces the attack cost and training cost. At the same time, our extensive experiments on the CIFAR series datasets verify the superiority of this generation method in improving the transfer ability.

Keywords: adversarial samples, adversarial transformation, cycle-consistent generative adversarial networks.

1. Introduction

Deep learning is an extension and continuation of machine learning. The birth of machine learning has enabled various classification and regression tasks to be better solved. With the introduction of deep learning, this field has been developing unprecedentedly. Well-known complex tasks, such as computer vision [1], speech recognition, natural language processing have achieved the best-known results with the aid of deep learning. At the same time, deep learning tasks have also landed in areas such as autonomous driving and face recognition.

DOI: 10.54254/2755-2721/2/20220672

Deep learning is a widely used multidisciplinary tool. Like many practical technologies, it also faces security challenges, such as adversarial sample attacks. As early as 2014, Szegedy[2] proposed the concept of adversarial examples for images. Adversarial examples are generated by adding computationally perturbed noise to the original clean image, and they cause the image classifier to misclassify the perturbed image. These disturbance noises are so tiny that they are imperceptible to the naked eye. The existence of adversarial examples exposes the major flaws of deep learning technology in the security field, forcing people to look at deep learning and its related applications more cautiously.

Reviewing the attack measures of multi-generational adversarial examples [3]: white-box and black-box attacks dominate. A white-box attack is an attack in which the attacker knows all the information about the target model. Unlike white-box attacks, which need to know all the information about the model, in black-box attacks, the attacker knows part of the information. The paper [4] is the first to demonstrate that black-box attacks on DNN classifiers are more convenient and practical for attackers who do not know the model. Therefore, improving the transfer ability of adversarial examples will improve the generality of adversarial examples, and finally make them applicable to attacks against more models.

Adversarial transformation is an idea to improve the transfer ability of adversarial samples. Image transformation is an emerging way to defend against adversarial examples, and the goal of adversarial transformation is for adversarial examples to learn how to defend against image transformations. A feasible way to implement adversarial transformation is to use Convolutional Neural Networks (CNN) [5] for data augmentation. The way it works is to conduct a tentative attack on the target network, making the adversarial samples robust against different image changes, so as to remove part of the adversarial noise with poor transfer ability while preserving the semantic information of the image.

At present, the transfer ability of adversarial sample attacks needs to be improved, and the evaluation system of adversarial samples needs to be improved. It is a feasible research scheme to improve the transfer ability of adversarial samples based on adversarial transformation. Iterative adversarial training is required to fully utilize the convolutional neural network to realize the adversarial transformation, and the cascaded model for attack requires the adversarial samples prepared in advance as input, which makes the network training process more complicated and the attack cost relatively high. This paper explores an adversarial transformation network reconstructed by generative adversarial networks and uses it as an attack module of adversarial samples to cascade before the victim model to complete the entire attack chain. Compared with the adversarial transformation that fully utilizes the convolutional neural network, the adversarial transformation network proposed in this paper can directly use the original clean image to generate adversarial samples, which reduces the attack cost and network training difficulty. In summary, the contributions of this paper are as follow：

- An improved method of adversarial transformation network is proposed, which regards the problem of adversarial sample generation as a case of style transfer. This method avoids the tedious iteration of the gradient notation method in the training process and saves the training cost.
- Evaluate our proposed attack method (CGATA) with many adversarial example attacks based on the CIFAR dataset. Experimental results show that our method has good transferability under black-box attacks.
- A feasible transferability score (ATS) is proposed, and the ATS calculated from the experimental data further proves that our method has better transferability under black-box attacks. This score can be used as a concise and powerful reference for the transferability of adversarial examples.

2. Related work

2.1. Generation of adversarial examples

There are two ways to generate adversarial examples. The first way is the white-box model scenario, in which the target model acts as the source model and the attacker fully knows everything about the target model, such as model structure or parameter settings. The second way is the black-box model scenario, where the attacker does not know the specific information of the target model. In general, the scenario

DOI: 10.54254/2755-2721/2/20220672

of the black-box model is more realistic for a system based on a deep neural network, and the research in this paper will also focus on the scenario of the black-box model.

There are two adversarial attacks built for black-box attacks: query-based attacks and transfer-based attacks. Inquiry-based attacks need to actively interrogate the target model for instances of interest and use the feedback it gives to construct adversarial examples. However, query-based attacks often require a large number of queries to determine suitable adversarial examples, which not only brings unimaginably high costs but also makes the attack easier to detect. In contrast, transfer -based attacks are more practical and applicable. Transfer -based attacks use local existing models to generate adversarial samples, and directly use these adversarial samples to attack remote target models. It is worth noting that the transfer-based attack method relies more on the transfer ability of adversarial samples, which means that if the transfer ability of the adversarial samples generated by the source model is stronger, the threat to other different models will be stronger.

Unfortunately, transfer-based attacks often exhibit limited success rates, especially when attacking some complex models, because of the potential overfitting of the source model. To improve the transferability of adversarial examples, a common solution is to treat the process of generating adversarial examples as an optimization problem. From this perspective, previous researchers have attempted to transfer traditional optimization algorithms and use them to generate transferable adversarial examples.

2.2. Ways of improving transfer ability

Methods to improve transferability can be mainly divided into two categories. The first is to include as many advanced optimization algorithms as possible, like momentum methods [6] and gradient acceleration algorithms. The second is data augmentation. Utilizing this method requires the generated adversarial samples to be robust to specific image transformations [7], while also protecting the integrity of the image content. Specific image transformations are image resizing, image transformation, and image scaling. However, an image transformation or a combination of multiple simple transformations will make the constructed adversarial samples overfit to the applied image transformation, and it is difficult to resist unknown image transformations, resulting in poor transfer ability.

Therefore, an immediate remedy is to identify image transforms that preserve picture information and fine-tune the combination of image transforms appropriate for each picture. However, such a process incurs incalculable computational costs. A feasible strategy is to train an adversarial transformation network to automatically adjust the optimal image transformation. After such an image transformation, the adversarial samples will improve the transfer ability while preserving the image information.

2.3. Adversarial sample attack

The attack methods of adversarial samples are mainly divided into two categories: attack algorithms based on gradient optimization and attack algorithms based on constraint optimization.

2.3.1. Fast gradient sign method (FGSM). It is the first method to attack by computing adversarial perturbations on loss gradients. Its attack update equation is as follows:

$$x^{adv} = x + \epsilon \times sign\left(\nabla x L(x, y^{true})\right) \tag{1}$$

The parameter ϵ adjusts the specific degree of the small disturbance, $\nabla x L(x, y^{true})$ is the first derivative of the loss function with respect to the input$x$, y^{true} is the correct label, and $sign$ is the sign function. The way FGSM works is: First, the loss function of the attack model is obtained and its gradient value relative to the input is calculated.

DOI: 10.54254/2755-2721/2/20220672

*2.3.2. BIM.*BIM is an iterative version of traditional FGSM, and it is one of the many variants of FGSM. The improvement method of BIM is to use FGSM to iteratively add adversarial disturbances on the original clean samples for a limited number of times, to generate more attack capabilities within the scope of disturbance constraints. Powerful adversarial examples. Its attack method can be expressed as follows:

$$x_0^{adv} = x \tag{2}$$

$$x_{n+1}^{adv} = Clip_x^{\epsilon}\left\{x_n^{adv} + \alpha \times sign\left(\nabla xL\left(x_n^{adv}, y^{true}\right)\right)\right\} \tag{3}$$

Where α represents the step size of each iteration, and $Clip_x^{\epsilon}$ represents the cropping operation on the image, which ensures that the processed pixels are in the range of the original image.

2.3.3. Momentum iterative fast gradient sign method (MI-FGSM). The core idea of this method [6] is similar to the above-mentioned basic iterative fast gradient notation method. The specific algorithm is described as follows:

$$g_{i+1} = \mu g_i + \frac{\nabla xJ(x_i^A, y)}{\left|\left|\nabla xJ(x_i^A, y)\right|\right|_1} \tag{4}$$

$$x_{i+1}^A = x_i^A + \alpha sign(g_{i+1}) \tag{5}$$

g_{i+1} represents the gradient momentum accumulated after i iterations, and μ is the decay factor of the momentum term; when μ is 0, the above form is equivalent to the I-FGSM algorithm. $\left|\left|\nabla xJ\left(x_i^A, y\right)\right|\right|_1$ means that the current gradient obtained in each iteration is normalized by its own l_1 distance. The algorithm introduces momentum technology in the iterative process of the I-FGSM algorithm, to accumulate the velocity vector in the direction of the loss gradient change to stabilize the update direction of the gradient, so that the optimization process is not easy to fall into the local optimum.

3. Generate adversarial examples

This chapter will elaborate on the research method of this paper. First, the construction principle of adversarial samples is introduced, followed by a detailed description of the idea of adversarial transformation and the generation algorithm of adversarial samples.

3.1. Problem definition

Let x be the unprocessed image and y be its corresponding label. We can think of a deep neural network image classifier as a function $f(x)$ that returns a probability vector that represents the probability of the class the input image belongs to. Given the target model f and the unprocessed image x, the attacker's goal is to find an adversarial image x^{adv}. Where x^{adv} should satisfy the following two conditions:

$$\arg\max f\left(x^{adv}\right) \neq y, \tag{6}$$

$$\left|x^{adv} - x\right|_p \leq \epsilon \tag{7}$$

Equation (3.1) reflects that the attacker's goal is to mislead the target model into making wrong predictions. The condition of Equation (3.2) limits the adversary's acceptable interference expectation. Usually, the interference expectation ϵ is a very small value, which ensures that the change is unobservable to the human eye. In the research of this paper, we use the symbol l_∞ to define the visibility of anti-jamming, which is also the most widely used method.

We denote the training loss function of the deep neural network classifier f by $J(f(X), y)$, so that the problem of generating adversarial samples x^{adv} can be defined as the following optimization problem:

$$\max_{x^{adv}} \; J\left(f\left(x^{adv}\right), y\right) \tag{8}$$

$$s.t. \quad \left|x^{adv} - x\right|_\infty \leq \epsilon. \tag{9}$$

In this optimization problem, the attacker's goal is to maximize the training loss of the deep neural network under the premise that the adversarial examples meet the perturbation constraints.

DOI: 10.54254/2755-2721/2/20220672

3.2. Adversarial transform network

Image transformation on adversarial samples is a means of resisting adversarial attacks. If the idea of adversarial learning is used to let the adversarial samples learn how to deal with image transformation in advance, it can strengthen its attack ability, and then improve the transfer ability of the adversarial samples. Adversarial samples with strong transfer ability should add some suitable adversarial noises based on retaining the information of the original image. These noises do not affect the original information of the picture but can interfere with the deep neural network-based discriminator, making it make wrong judgments. Obtaining adversarial examples with strong transfer ability requires the adversarial example generator to learn the optimal adversarial perturbation. A feasible method is to train an adversarial transformation network to strengthen the adversarial examples. Here, we consider a simple attack mode—undirected adversarial example attack.

The implementation method of this adversarial transformation network [5] is to use the convolutional neural network to query the adversarial samples and use rich image transformation methods or sets to optimize the adversarial samples, and finally improve their transfer ability. The workflow is as follows: firstly, the adversarial sample dataset is input, and the adversarial transformation network is trained to reduce the destructiveness of the adversarial samples to the target discriminator without destroying the original information of the image. Next, the trained adversarial transformation network is cascaded in front of the target model as a pre-training module. Here, the adversarial transformation network can be regarded as an attack reinforcement module of adversarial samples. This method needs to iterate the training process and each input must be a pre-generated adversarial sample, so using this method for adversarial sample attack will bring high attack cost.

We explore an improved adversarial transformation network. The advantage of this network is that it can directly generate adversarial samples with better transfer ability without iterative training and using the original clean samples. The network construction process is as shown in Figure 1:

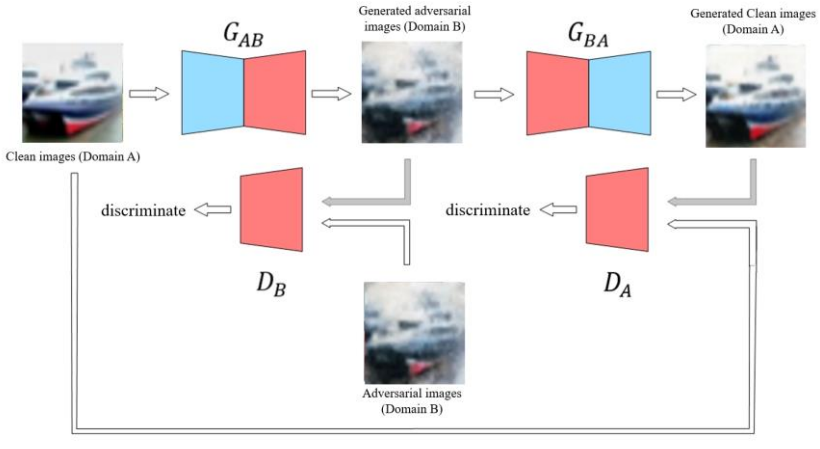

Figure1. The network construction process.

We add a new generator G_{BA} to convert b_i back to a_i. At the same time, to identify this conversion process, an additional discriminator D_A needs to be added to determine whether a_i retains the image information of the original image. For example, the regenerated image a_i of adversarial examples of landscape should still be the landscape of the original image. The improved new network architecture follows the idea of CycleGAN [8]. CycleGAN realizes the conversion of images between different domains based on preserving image information through style transfer.

The adversarial transformation network architecture based on CycleGAN is shown in Figure 1. We build two generators G_{AB} and G_{BA}, and two discriminators D_B and D_A. Now, the input clean samples follow two paths: first, the input image enters the discriminator, which judges whether it is real; second, the input image enters the generator, which is converted into an adversarial sample, which is then evaluated by the discriminator, evaluating the goal is to detect whether it is real or not, and finally

DOI: 10.54254/2755-2721/2/20220672

measure the loss between the input image and the transformed image. To reflect the whole network more clearly, the information of the network is summarized in the Table 1.

Table 1. Specific functions of each part of the network.

	Input	Output	target
Generator G_{AB}	clean samples transformed by adversarial samples or clean samples	adversarial samples	Generate adversarial examples
Generator G_{BA}	adversarial samples transformed by clean samples or adversarial samples	clean samples	Generate clean examples
Discriminator D_A	clean samples or transformation of adversarial samples	The probability of adversarial examples	don't be fooled by Generator G_{BA}
Discriminator D_B	adversarial samples or transformation of clean samples	The probability of adversarial examples	don't be fooled by Generator G_{AB}

The loss function of CycleGAN is described below.

3.2.1. Cycle consistency loss. The core idea of CycleGAN for image transformation is to complete a cycle: transform from one domain to another and back again, so theoretically the original image $\boldsymbol{a}$ and the reconstructed image $\hat{\boldsymbol{a}}$ are the same. If they are different, you can measure their pixel-level loss, which is the cycle-consistency loss. To define the cycle consistency loss, two generators need to be constructed: one generator $\boldsymbol{G_{AB}}$ transforms from A domain to B domain; the other generator $\boldsymbol{G_{BA}}$ transforms from B domain to A domain. Its loss function is as follows:

$$Loss_{cycle}(G_{AB}, G_{BA}) = E_{y\sim p_{data(y)}}\left[\left|G_{AB}\left(G_{BA}(y)\right) - y\right|\right] + E_{x\sim p_{data(x)}}\left[\left|G_{BA}\left(G_{AB}(y)\right) - x\right|\right] \quad (10)$$

3.2.2. Adversarial loss. In addition to the cycle consistency loss, there is also an adversarial loss. Each generator has a corresponding discriminator, $\boldsymbol{G_{AB}}$ corresponds to discriminator $\boldsymbol{D_B}$, and $\boldsymbol{G_{BA}}$ corresponds to discriminator $\boldsymbol{D_A}$. This loss includes two aspects: one is the probability that the given image is a real image rather than a transformed image; the other is whether the input image fools the discriminator. The adversarial loss ensures that the transformed image looks real, clear, and indistinguishable from the real image. For the mapping function $\boldsymbol{G:\ a \to b}$ and the corresponding discriminator $\boldsymbol{D_B}$, its loss function should be as follows:

$$Loss_{GAN}(G_{AB}, D_B, X, Y) = E_{y\sim p_{data(y)}}[logD_B(y)] + E_{x\sim p_{data(x)}}[log(1 - D_B(G_{AB}(x))] \quad (11)$$

3.2.3. Total loss. The two losses mentioned above are integrated and calculated, and finally trained as the statistical loss of the network. The overall loss describes that the entire transformation is realistic and meaningful. It should be two adversarial losses (since two generators are built), plus the cycle consistency loss (where $\boldsymbol{\lambda}$ is a parameter that adjusts the cycle consistency loss), then the expression for the overall loss is as follows:

$$L = Loss_{GAN}(G_{AB}, D_B, X, Y) + Loss_{GAN}(G_{BA}, D_A, Y, X) + \lambda\, Loss_{cycle}(G_{AB}, G_{BA}) \quad (12)$$

In terms of network architecture, this paper adapts the generative network architecture of Zhu et al. [8], who achieved impressive results on neural style transfer. The generator network consists of two convolutional neural networks with a stride of 2 and two deconvolutional networks with a stride of 2. At the same time, the residual structure of ResNet is used, so that low-level information can cross the network through shortcuts. The implementation of the discriminator is slightly simpler, as it itself is like a basic convolutional neural network classifier. Therefore, this paper uses three convolutional networks

DOI: 10.54254/2755-2721/2/20220672

with a stride of 2. The last layer of the network is fed into a fully connected linear layer to generate a one-dimensional output, and finally the sigmod function is used to output the discriminated probability distribution.

4. Experiment

4.1. Experimental setup

In this article, we will use the TensorFlow-based backend Keras-GAN implementation with keras_contrib installed. The computing platform used in the experiment is NVIDIA GeForce RTX 3060 with Deep Learning Framework: Keras2.2.4, Tensorflow1.12, Pytorch1.11.0

4.1.1. Datasets. There are many types of data sets used for deep learning, and this paper selects representative data sets for experimental testing. The first is the CIFAR series of commonly used object recognition small datasets, including CIFAR-10 and CIFAR-100. The CIFAR series dataset was collected by Alex Krizhevsky, Vinod Nair and Geoffrey Hinton. The specific information of the data set used in the experiment is shown in Table 2.

Table 2. Information of the dataset.

Datasets	categories	Quantity of data	Size of pictures
CIFAR-10	10	60000	32*32
CIFAR-100	100	60000	32*32

After the clean samples are prepared, we will use the BIM method to process the clean samples to generate adversarial samples for subsequent training.

4.1.2. Source model. This paper selects the open-source neural network TensorFlow-Slim image classification model library (TF-Slim) [9] as the source of the source model. Please refer to Appendix A for the specific model structure. The models used are listed in Table 3, along with their classification accuracy on benign input tests.

Table 3. Classification accuracy of each model.

Source model	ResNet-V1-50	ResNet-V2-50	Inception V3	Inception V4	Inception-ResNet-v2
MNIST	99.7	99.8	99.7	99.8	99.8
CIFAR-10	93.4	95.0	96.3	97.6	97.7
CIFAR-100	72.0	74.4	78.0	80.2	80.4

4.2. White-box based adversarial transformation attack

The process of implementing a white-box attack is relatively simple. The main idea is to attack the target model after determining its parameter settings. Firstly, a specific data set is constructed using the adversarial sample generation algorithm (BIM) for the target model, and the adversarial transformation network is trained with this data set and used to attack the target model; then the attack using traditional methods (FGSM, BIM, MI-FGSM) to generate adversarial samples is tested. performance. The results of the attack experiment are shown in Table 4.

Table 4. Classification accuracy after white-box based attack.

.Attacking Method	ResNet-V1-50	Inception V3	Inception-ResNet-v2
FGSM	12.4	27.2	57.2
BIM	6.9	2.3	57.5
MI-FGSM	1.8	0	3.2
$CGATA^1$	1.9	1.4	2.1

The first row in Table 4 shows the target model under attack. In addition to the comparison group method mentioned above, CycleGAN-Adversarial-Transformation Attack (CGATA) represents the

DOI: 10.54254/2755-2721/2/20220672

attack method proposed in this paper. It can be understood that the adversarial transformation network is cascaded at the front end of the target network, and the target network is attacked by inputting clean samples. In the white-box case, $CGATA^1$ represents the attack with the target model as the source model, and all datasets are trained with CIFAR-10 in the experiments of the white-box attack.

The experimental results show the attack success rate of different attack methods on the three target models, which is reflected in the classification accuracy of the target model. The data source in the table is the top-1 classification accuracy obtained by inputting 1000 adversarial sample test data into the target model discrimination network (the statistical principles of the experimental data are the same below and will not be repeated). According to the classification accuracy of the target model, we can agree on an evaluation standard: the lower the classification accuracy of the target model, the stronger the impact of the attack on the model's misjudgment, and vice versa.

4.3. Black-box based adversarial transformation attack

In this subsection, we evaluate the attack capability of the adversarial examples generated by the adversarial transformation network to analyze their transferability effectively and objectively. To deal with the attack environment under the black-box model, a feasible method is to train the adversarial transformation network with datasets generated by different source models and optimize its internal generator. Then the adversarial transformation network is cascaded before other target models to complete the construction of the transfer attack. In the actual construction stage, this experimental process is equivalent to using the source model for combined attack. Due to the variety of combinations, here we select a typical network as the source model to create a dataset for training the adversarial transformation network and select the commonly used network model as the target model for the attack. . The experimental results are shown in Table 5.

Table 5. Classification accuracy after black-box based attack.

Source model	Attacking method	ResNetV1-50	ResNetV2-50	Inception V3	Inception V4	Inception-ResNet-v2
ResV1	FGSM	-	48.2	51.3	56.8	61.8
	BIM	-	40.7	46.2	54.0	57.8
	MI-FGSM	-	33.6	41.1	42.1	45.6
	$CGATA$	-	25.1	30.7	33.2	40.8
IncV3	FGSM	55.7	60.7	-	55.2	68.9
	BIM	53.8	61.8	-	60.5	68.2
	MI-FGSM	56.8	59.8	-	51.6	52.3
	$CGATA$	48.2	51.2	-	43.1	46.8
IncResV2	FGSM	60.7	68.7	64.0	74.5	-
	BIM	52.4	56.4	40.5	74.5	-
	MI-FGSM	51.1	56.1	36.2	48.4	-
	$CGATA$	46.2	50.2	28.1	31.8	-

The first row in Table 5 shows the target model under attack, and the first column is the source model required to build CGATA training. In order to better compare and quantify the data, we use CIFAR-100 with more target classifications in the black-box experiment, because the more discriminative probability vector classifications of the target model, the easier it is to respond to small perturbations; at the same time, since we want to analyze the transfer ability of the attack, we choose more target models as reference. In fact, the above attack combination still includes non-migrating white-box attacks, which we omit here for the sake of data neatness.

In terms of result analysis, the attack results in the black box mode are far inferior to those in the white box mode, but it does show a certain transfer ability. Different from the attack in the white-box mode, CGATA beats all experimentally selected attack methods, showing strong attack performance. Among them, since the generation method of training data is BIM algorithm, we will focus on comparing the attack performance of CGATA and BIM. For example, when using ResNetV1 as the source model,

DOI: 10.54254/2755-2721/2/20220672

CGATA reduces the classification accuracy of Inceptionv3 by 15.5%. It is further demonstrated that the adversarial transformation network with CycleGAN as the idea can improve the transfer ability of adversarial samples.

4.4. Adversarial sample transfer ability analysis

Since the use of generated adversarial samples to attack other models is a transfer -based attack method, the evaluation criteria involved can only be relatively objective. The analysis and evaluation indicators proposed in this paper are mainly aimed at the experimental model system involved in this experiment. When using adversarial samples to attack models with deeper network levels, the expected attack strength calculated by the evaluation indicators in this paper can only be used as a reference.

Use the results of 4.3 for evaluation. Because it involves the adaptive evaluation of adversarial samples in the transfer process, to quantify this indicator, we first set the importance weight ρ for each target model to represent the reference value of the model. The frequency of scientific research use and the performance of the network are scored based on the AI 2021 annual data report [10] and the DAWNBench [11] platform developed by Stanford University.

As shown in Table 4-5, it can be seen from the above experiments that the vulnerability of the target model decreases from left to right, and the ability of ResNetv2 to resist adversarial sample attacks is better than that of the v1 version; while the network after inceptionv3 draws on the residual error The module has been optimized, and the model robustness has also been improved relative to ResNet. Therefore, in the definition of ρ, the model that is easier to break will have a relatively small proportion of the score, and vice versa.

For the neural network model used in this experiment, the corresponding weight ρ is shown in Table 6.

Table 6. Weight ρ of each model.

Target model	ResNetV1-50	ResNetV2-50	Inception V3	Inception V4	Inception-ResNet-v2
ρ	10	15	20	25	30

Combined with the evaluation criteria mentioned above, the specific Adversarial-Samples-Transformation-Score hereinafter referred to as ATS is defined as follows:

$$ATS(x) = \sum_i^N [\rho_i \times (P_{clean_i} - P_{black_i})] \quad (13)$$

Among them, ρ_i represents the evaluation weight index of the corresponding model, P_{black_i} represents the classification accuracy when the adversarial samples generated by the attack method are input into the target model in the black-box scenario, and P_{clean_i} represents the benign input accuracy of the target model. The higher the ATS score, the stronger the transfer ability. For a clearer comparison, this paper normalizes the ATS, and the closer it is to 1, the stronger the transfer ability. The evaluation indicators obtained in the experiment are shown in Table7.

Table 7. ATS of each attacking method.

Source model	FGSM	BIM	MI_FGSM	CGATA
ResNetv1	0.374	0.473	0.605	0.732
Inceptionv3	0.245	0.226	0.354	0.462
Inception-ResNet-v2	0.114	0.251	0.392	0.523

DOI: 10.54254/2755-2721/2/20220672

Figure 2 is the visualization result of ATS. Due to the normalization operation, the value range of ATS is determined between 0 and 1. The closer it is to 1, the stronger the transfer of the attack, and vice versa. ATS indirectly represents the transferability of generated adversarial examples. Under the generation of different source models and different methods, similar trends are basically shown: CGATA exhibits the strongest transfer ability, which is in line with the expectations of this paper.

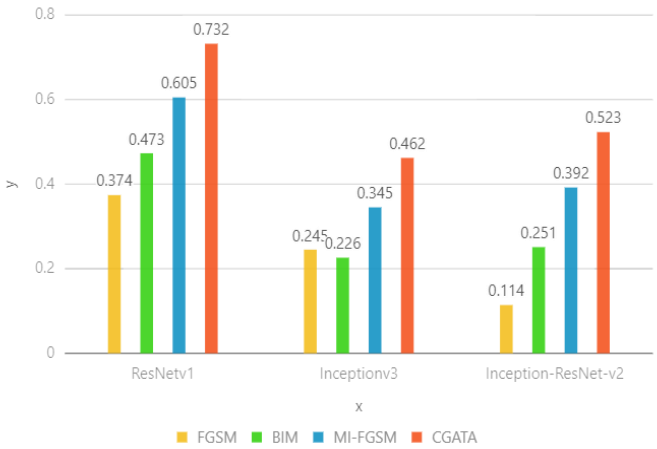

Figure 2. The visualization result of ATS (after normalized).

4.5. Adversarial transform network performance analysis

Finally, the performance of the adversarial transformation network constructed in this paper is tested. The test content mainly focuses on the efficiency of generating adversarial samples. The specific evaluation criterion is the time required to generate a unit number of adversarial samples. The shorter the time required, the lower the attack cost, and vice versa.

Table 8. Attack cost under time measure.

Attacking method	FGSM	BIM	CGATA
Time	0.265	2.043	0.067

It can be seen from the experimental results in the Table 8 that our method spends less time on generating adversarial examples. Therefore, the attack cost of adversarial attacks using CGATA is greatly reduced.

5. Conclusion

Systems based on deep neural networks are active in various fields, which solve many problems related to vision, speech and video, and bring convenience to people's production and life. But at the same time, attacks against deep neural networks have also emerged quietly. A typical method is the transfer attack using adversarial examples. For this kind of attack, weak transfer ability will seriously weaken the strength of the attack, so improving the transfer ability of adversarial samples becomes the main means to strengthen this kind of attack. At present, the method of improving the transfer ability of adversarial samples based on adversarial transformation requires complex training and high cost. This paper uses the cycle-consistent generative adversarial network to improve the adversarial transformation network and proposes a new attack method (CGATA) based on the network. We demonstrate the superiority of the improved method in improving the transfer ability through many experiments.

DOI: 10.54254/2755-2721/2/20220672

Reference

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep C onvolutional Neural Networks[C/OL]//Advances in Neural Information Processing Syst ems: 25. Curran Associates, Inc., 2012. https://proceedings.neurips.cc/paper/2012/hash/c 399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[2] GOODFELLOW I, SZEGEDY C. Explaining and Harnessing Adversarial Examples[C/O L] arXiv:1412.6572 [stat.ML]. https://doi.org/10.48550/arXiv.1412.6572.

[3] AKHTAR N, MIAN A, KARDAN N, et al. Advances in Adversarial Attacks and Defen ses in Computer Vision: A Survey[J/OL]. IEEE Access, 2021, 9: 155161-155196. http s://doi.org/10.1109/ACCESS.2021.3127960.

[4] PAPERNOT N, MCDANIEL P, GOODFELLOW I, et al. Practical Black-Box Attacks a gainst Machine Learning[C/OL]//Proceedings of the 2017 ACM on Asia Conference o n Computer and Communications Security. Abu Dhabi United Arab Emirates: ACM, 2017: 506-519. https://dl.acm.org/doi/10.1145/3052973.3053009.

[5] WU W, SU Y, LYU M R, et al. Improving the Transferability of Adversarial Samples with Adversarial Transformations[C/OL]//2021 IEEE/CVF Conference on Computer Vi sion and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, 2021: 9020-9029. https://ieeexplore.ieee.org/document/9577804/.

[6] DONG Y, LIAO F, PANG T, et al. Boosting Adversarial Attacks With Momentum[C/O L]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 9185-9193. https://openaccess.thecvf.com/content_cvpr_2018/html/Dong_Boosting _Adversarial_Attacks_CVPR_2018_paper.html.

[7] XIE C, ZHANG Z, ZHOU Y, et al. Improving Transferability of Adversarial Examples With Input Diversity[C/OL]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 2730-2739. https://openaccess.thecvf.com/content _CVPR_2019/html/Xie_Improving_Transferability_of_Adversarial_Examples_With_Input_ Diversity_CVPR_2019_paper.html.

[8] ZHU J Y, PARK T, ISOLA P, et al. Unpaired Image-To-Image Translation Using Cycl e-Consistent Adversarial Networks[C/OL]//Proceedings of the IEEE International Confe rence on Computer Vision. 2017: 2223-2232. https://openaccess.thecvf.com/content_iccv _2017/html/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.html.

[9] SILBERMAN N, GUADARRAMA S, "TensorFlow-Slim image classification model libra ry" 2016[DB]. https://github.com/tensorflow/models/tree/master/research/slim.

[10] ZHANG D, MISHRA S, BRYNJOLFSSON E, et al. “The AI Index 2021 Annual Repor t,” AI Index Steering Committee, Human-Centered Artificial Intelligence Institute, Sta nford University, Stanford, CA, March 2021[R], Chinese translation by Synced.

[11] COLEMAN C, NARAYANAN D, KANG D, et al. Stanford DAWN Project [DB] http:/ /dawn.cs.stanford.edu/benchmark.