A Survey of Adversarial Attacks and Defenses for image data on Deep Learning

Huayu Li, Dmitry Namiot

***Abstract*—This article provides a detailed survey of the so-called adversarial attacks and defenses. These are special modifications to the input data of machine learning systems that are designed to cause machine learning systems to work incorrectly. The article discusses traditional approaches when the problem of constructing adversarial examples is considered as an optimization problem - the search for the minimum possible modifications of correlative data that "deceive" the machine learning system. As tasks (goals) for adversarial attacks, classification systems are almost always considered. This corresponds, in practice, to the so-called critical systems (driverless vehicles, avionics, special applications, etc.). Attacks on such systems are obviously the most dangerous. In general, sensitivity to attacks means the lack of robustness of the machine (deep) learning system. It is robustness problems that are the main obstacle to the introduction of machine learning in the management of critical systems.**

***Ключевые слова*—adversarial machine learning, deep learning, security**

I. Introduction

In recent years, "artificial intelligence" has entered a period of rapid development, and various countries, as well as IT giants (such as IBM, Google, Apple, Amazon, Huawei, etc.) have increased their research and investment in the field of "artificial intelligence". Among them, "machine learning" is a popular research topic.

Currently, machine learning has achieved recognized results in complex tasks such as computer vision [1], speech recognition [2, 3] and natural language processing [4], and has been widely used in cutting-edge fields such as self-driving [5] and face recognition [6].

With the gradual maturity of "machine learning" technology, it has been applied in many fields that have strict requirements on security, such as military [7], finance [8], and healthcare [9], which directly affects the safety of people's personal, property, and privacy. While enjoying the convenience brought by "machine learning", people tend to ignore the "Threat". These so-called "Threats" pose a threat to people's safety and property safety.

Like many computer application techniques, machine learning, as a complex computer system, has been found to have security issues [10, 11]. C Szegedy et al. [12] found that during the training phase, an attacker can cause a machine learning model to give a false prediction with high confidence by adding some perturbations to clean samples and that these perturbations cannot be recognized by the human visual system.

Deep learning, the paradigm of machine learning, has shown great promise in recent years [13]. However, the security risks of deep learning techniques being spoofed by adversarial samples are exacerbated by the security concerns of deep learning frameworks.

Adversarial data undoubtedly restricts the further application of machine learning technology. Therefore, it is very important to improve the adversarial robustness of neural networks (the ability to resist adversarial samples).

This research has been supported by the Interdisciplinary Scientific, Educational School of Moscow University "Brain, Cognitive Systems, Artificial Intelligence" and the Chinese Scholarship Council.

II. Adversarial Attacks

A. Adversarial Samples

The adversarial samples are artificially and maliciously constructed. By adding an imperceptible perturbation η to a clean sample x, the model $\mathcal{F}$ causes the sample x' to be misclassified, and this output has a high confidence level for the model, where $x' = x + \eta$. This process is expressed in mathematical language as:

$$\mathcal{F}(x) \neq \mathcal{F}(x')$$
$$||x' - x||_p \leq \epsilon, \text{ where } \epsilon \to 0.$$

B. Classification of Adversarial Attack Methods

Adversarial attacks can be divided into two categories depending on the extent to which the attacker has information about the machine learning model:

- White-Box Attack: The attacker has full knowledge of his target model, including the structure of the model, the parameter values, the means of training, and also information about the set used for training.
- Black-Box Attack: In contrast to a white-box attack, the attacker does not know the internal structure of the attacking model, the training parameters, etc. The attacker can pass in data to the model and develop an attack strategy by observing and judging the output and interacting with the model. This approach is more consistent with the real situation.

And according to the purpose of the attack, the attacks can be divided into the following two categories.:

- Non-Target Attack: The attacker does not predetermine the outcome of the target model's misprediction, i.e. the

Received Feb, 1, 2022.
Huayu Li, Lomonosov Moscow State University (email: leesir1017@gmail.com)
Dmitry Namiot, Lomonosov Moscow State University (email: dnamiot@gmail.com).

output can be arbitrary, and simply allows the target model to misclassify the adversarial sample.
- Targeted Attack: An attacker needs to consider misclassifying an adversarial sample into a specified classification category when formulating an attack strategy, such as constructing an adversarial sample. This attack technique is mostly used when the model needs to classify multiple features, i.e. a multi-classification problem.

C. Transferability of Adversarial samples

Transferability [14] is a property that adversarial samples have. The adversarial sample generated by model A can attack model B.

D. Methods for Generating Adversarial Samples

1) Method L-BFGS: The first adversarial attack algorithm aimed at attacking deep neural network models was proposed by szegedy et al. [12]. Its ultimate goal is to find an imperceptible minimum input perturbation $\arg\min_r ||r||_2$ in the constraint space of the input, i.e., $r = x' - x$, and to make the model classification wrong, i.e., $F(x) \neq F(x') = y'$. Since this optimization problem is not easy to solve. So authors used the L-BFGS method to convert this difficult-to-solve optimization problem into a box-constrained form with the goal of finding an adversarial sample x'-minimization formula:

$$c||r|| + \text{loss}_F(x', y'), \text{there exists } x' \in [0, 1],$$

The above optimization process is performed in an iterative form and the parameter c is gradually made larger by linear lookup until the adversarial sample is found.

L-BFGS attacked the best image classification models of the time, AlexNet [1] and QuocNet [15], successfully, causing the model to misclassify a large number of images.

Also, the authors argue that the semantic information in deep neural networks is based on the whole network and not on the neurons of a particular layer. The following two conclusions were also drawn: Adversarial samples can generalize across models: a large portion of the adversarial samples generated on model A are also valid on model B. (Which has the same structure as model A with different Hyperparameters); D_1, D_2 are different subsets of Dataset D, each trained with a different model, and the adversarial samples generated on D_1 are also valid on D_1.

2) Method FGSM and its extensions: Goodfellow et al. [16] proposed an effective untargeted l_∞-based attack method. (As shown in Figure 1).

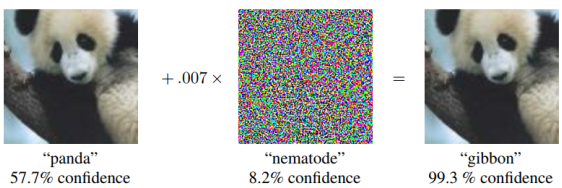


Figure 1: An very famous example of an adversarial sample. And Perturbation misleading model GoogleNet [17] caused by FGSM method identifies panda as a gibbon [16].

FGSM adds noise to the input samples in the direction of the gradient of the loss function to obtain an adversarial perturbation, which is then added to the input samples to form an adversarial sample, and this process is as follows:

$$x^* = x + \epsilon \cdot \text{sign}(\nabla_x J_{loss}(x, y)),$$

where ϵ is the size of perturbation; sign$(\cdot)$ - sign function; $J_{loss}(x, y)$ - loss function. The R+FGSM method [18], unlike FSGM, adds perturbations in the opposite direction of the gradient, and the process is as follows:

$$x^* = x - \epsilon \cdot \text{sign}(\nabla_x J_{loss}(x, y')),$$

where y' - target class. This method allows for FGSM targeted attacks. FGSM generates adversarial samples faster than L-BFGS [12] because it only needs to calculate the gradient of the loss function. However, there is a high probability that adversarial samples with small perturbations will not be obtained.

Furthermore, by adding random perturbations to the input samples before the execution of the FGSM, the diversity of the FGSM adversarial samples can be improved by this step [18].

The authors experimented with Softmax and Maxout networks on the MNIST dataset and the parameters $\epsilon = 0.25$ and $\epsilon = 0.1$, respectively, and the error rates for the adversarial samples were as follows [16]:

	Softmax	Maxout
$\epsilon = 0.25$	99.9%	89.4%
$\epsilon = 0.1$	-	87.15%

and the confidence level of the model for both error samples is extremely high.

In 2018, DONG et al. [19] proposed an optimized Momentum Iterative FGSM(MI-FGSM) method based on momentum iteration. The gradients are calculated by following equation:

$$g_{t+1} = \mu \cdot g_t \cdot + \frac{\nabla_x J(\theta, x'_t, y)}{||\nabla_x J(\theta, x'_t, y)||_1}$$

then apply the signed gradient in the equation:

$$x'_{i+1} = x_i + \alpha \cdot \text{sign}(g_{t+1})$$

to update x'_{i+1}, and finally generate perturbations.

The use of momentum facilitates faster convergence ,and smoother update directions and also helps to get rid of local maxima, thus increasing the success of the attack, but this method is less transferable.

BIM is an extension of FGSM and was proposed by KURAKIN et al. [20]. BIM perturbs in multiple small steps along the direction of increasing the gradient by iteration and recalculates the gradient direction after each small step, and the iterative process is as follows:

$$x_{t+1} = \text{Clip}_\epsilon(x'_t + \alpha \cdot \text{sign}(\nabla_x J_\theta(\theta, x'_t, y)))$$
$$\text{for } t = 0 \text{ to } T,\ x'_0 = x.$$

where Clip$(\cdot)$ constrains each input feature of the coordinates, such as pixels, to be restricted to the perturbed neighborhood of input x and the feasible input space.

BIM can construct more accurate perturbations than FGSM, however, this method requires significantly more computational effort and time.

Finally, it was verified that in a real physical environment, adversarial samples obtained through image capture can still cause the classifier to misclassify. The experiments demonstrate the possibility of adversarial samples in the physical world for machine learning systems by using photographs taken by mobile phones as input to the Inception v3 [21] network.

3) JSMA method: papernot et al. [22] propose a white-box targeted attack algorithm based on norm l_0, which requires that the number of pixels to be modified be minimized (this can be achieved by modifying the values of only a few pixels in the image). This method directly calculates the gradient of the loss function corresponding to the input samples X:

$$\nabla F(X) = \frac{\partial F(X)}{\partial X}$$

and then calculates its adversarial saliency plot, by which the input samples with the greatest influence on the particular output $F(X)$ of the classifier can be obtained, and the most influential samples will be used as perturbations. This method achieves good results on white-box target-specific misclassification attacks.

4) Deepfool Method: Moosavi-Dezfooli et al. [23] proposed a l_2-based untargeted attack method, which is called Deepfool. $f(x) = \omega^T x + b$ - affine classifier , The minimum perturbation that causes a clean sample x_0 to be misclassified is the distance from x_0 to the hyperplane of the classifier $\mathcal{F} = \{x : f(x) = 0\}$ (as shown in fig 2), the distance $\Delta(x_0, f(x))$ is $-\frac{f(x_0)}{||\omega||_2^2}$.

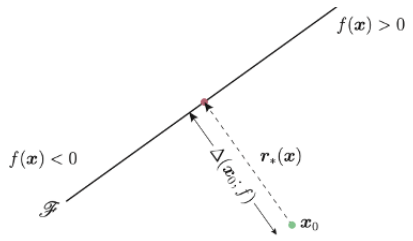


Figure 2: Adversarial examples for a linear binary classifier [23].

- (For binary classifiers) Provided that the classifier f is linear, the iterative formula for computing the perturbation is :

$$\underset{r_t}{\text{argmin}}\, ||r_t||_2, \text{subject to:}$$
$$f(x_t') + \nabla f(x_t')^T r_t = 0$$

  and the iteration stops when $\text{sign}(f(x_t)) \neq \text{sign}(f(x_0))$, meaning that x_0 has been updated to the other side of the hyperplane, resulting in a misclassification. Then, the minimum perturbation r^* is obtained, where r^* is the sum of r_t in the iterative equation.
- (For multiclass classifiers) As there are multiple classification hyperplanes, the distance from x_0 to each classification hyperplane boundary needs to be found, and then the vector with the smallest norm among them is chosen as the final perturbation.

The experiments were conducted on three datasets and eight classifiers. Experiments show that Deepfool produces less perturbation than FGSM on some benchmark datasets. Moreover, DeepFool calculates faster and can generate more accurate perturbations.

Classifier	DeepFool	Fast gradient sign	Clean
LeNet (MNIST)	0.8%	4.4%	1%
FC500-150-10 (MNIST)	1.5%	4.9%	1.7%
NIN (CIFAR-10)	11.2%	21.2%	11.5%
LeNet (CIFAR-10)	20.0%	28.6%	22.6%

Figure 3: Test error rates for the following 4 classifiers after fine-tuning using 3 different "adversarial samples" (containing a column of clean samples) [23].

Thus, this method can be used as a reliable adversarial robust tool to accurately estimate subtle perturbations and build more robust classifiers.

5) Optimization-Based Methods: Carlini and Wagner [24] propose three adversarial attack methods based on three metrics: "L_0 - The number of pixels in the image that have been modified"; "L_2 - Euclidean distances for adversarial and clean samples"; "L_∞ - The maximum changing value of pixels in the sample", which are used to find perturbations that minimise various similarity metrics. The perturbations are made approximately undetectable by limiting the L_0 (Adversarial simple: CW_0), L_2 (Adversarial simple: CW_2), and L_∞ (Adversarial simple: CW_∞) norms. This approach can be formulated as a constraint minimisation problem:

$$\text{minimize } ||\delta||_p + c \cdot f_{pre}(x + \delta)$$
$$\text{subject to: } x + \delta \in [0, 1], p \in \{0, 2, \infty\}$$

where δ is the adversarial perturbation, $\delta = x' - x$; $f_{pre}(x+\delta)$ - loss function which reflects the outcome of the adversarial attack, when the function is less than or equal to 0, it indicates that the network is predicted to be the target of the attack; conversely, it is not.

To ensure that a valid picture is generated, i.e. for the perturbation δ: $x + \delta \in [0, 1]$. The above problem of optimising δ is transformed into optimising ω by introducing a new variable ω, which can be described as follows:

$$\delta = \frac{1}{2}(\tanh(\omega) + 1) - x$$

In this way, Since $-1 \leq \tanh(\omega) \leq 1$, $+\delta = \frac{1}{2}(\tanh(\omega) + 1)$ is always located in $[0, 1]$ during optimization.

In addition to obtaining 100% success rate of the attack on the normally trained DNN models of MNIST, CIFAR10 (as shown in Figure 4), IMAGENET (as shown in Figure 5, and the C&W attack can also disrupt defensive distillation models. (as shown in Figure 6).

	Best Case				**Average Case**				**Worst Case**			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	8.5	100%	5.9	100%	16	100%	13	100%	33	100%	24	100%
JSMA-Z	20	100%	20	100%	56	100%	58	100%	180	98%	150	100%
JSMA-F	17	100%	25	100%	45	100%	110	100%	100	100%	240	100%
Our L_2	1.36	100%	0.17	100%	1.76	100%	0.33	100%	2.60	100%	0.51	100%
Deepfool	2.11	100%	0.85	100%	–	-	–	-	–	-	–	-
Our L_∞	0.13	100%	0.0092	100%	0.16	100%	0.013	100%	0.23	100%	0.019	100%
Fast Gradient Sign	0.22	100%	0.015	99%	0.26	42%	0.029	51%	–	0%	0.34	1%
Iterative Gradient Sign	0.14	100%	0.0078	100%	0.19	100%	0.014	100%	0.26	100%	0.023	100%

Figure 4: Comparison of C&W Attacks for MNIST AND CIFAR models [24].

	Untargeted		Average Case		Least Likely	
	mean	prob	mean	prob	mean	prob
Our L_0	48	100%	410	100%	5200	100%
JSMA-Z	-	0%	-	0%	-	0%
JSMA-F	-	0%	-	0%	-	0%
Our L_2	0.32	100%	0.96	100%	2.22	100%
Deepfool	0.91	100%	-	-	-	-
Our L_∞	0.004	100%	0.006	100%	0.01	100%
FGS	0.004	100%	0.064	2%	-	0%
IGS	0.004	100%	0.01	99%	0.03	98%

Figure 5: Comparison of C&W Attacks for INCEPTION V3 model ON IMAGENET [24].

	Best Case				Average Case				Worst Case			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	10	100%	7.4	100%	19	100%	15	100%	36	100%	29	100%
Our L_2	1.7	100%	0.36	100%	2.2	100%	0.60	100%	2.9	100%	0.92	100%
Our L_∞	0.14	100%	0.002	100%	0.18	100%	0.023	100%	0.25	100%	0.038	100%

Figure 6: Comparison of C&W Attacks when applied to defensively distilled networks [24].

6) Universal Adversarial Perturbation: Universal Adversarial Perturbation (UAP) [25] is, as the name implies, a universal perturbation computation method for different network models. This is the first systematic study of UAP. It works by searching for perturbations on a series of training datasets and adding the resulting perturbations δ' to each data sample by aggregating the original perturbations in such a way as to drive them closer to the bounds of the classifier and repeating the process until the sample is misclassified. Experiments show that this algorithm can effectively attack deep neural networks such as VGG [26](VGG-16, VGG-19), and ResNet [27](ResNet-152). This perturbation, which can span different samples, can be applied simultaneously to other different models.

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4 %
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Figure 7: Generalizability of the universal perturbations across above networks, where the percentages indicate the fooling rates [25].

Since then, in 2017, Mopuri et al. [28] proposed the Fast Feature Fool method. This method allows the training of the target model without obtaining information about the dataset and the internal structure of the model itself and has a much shorter convergence time than UAP. The ”data independent” nature of the method (no access to the target dataset) allows the resulting perturbations to exhibit greater cross-network transferability when trained on the same dataset.

Based on the FFF method [28], in 2018, Mopuri et al. [29] proposed the Generalizable Data-free UAP(GD-UAP) method and demonstrate that the new approach is not only more transferable but also proves generality and effectiveness in different computer vision tasks.

7) GAN-based attacks: XIAO et al. [30] proposed an adversarial generative method based on an adversarial network (GAN) [31]. This adversarial generative network consists of three main components, namely a generator $\mathcal{G}$, a discriminator $\mathcal{D}$ and a target neural network $\mathcal{L}$. The method maps the original sample x to an adversarial perturbation $\mathcal{G}(x)$ via a GAN generator $\mathcal{G}$. Once training is complete, the network is able to generate a new adversarial sample $x + \mathcal{G}(x)$. This sample is sent to the discriminator $\mathcal{D}$, which discriminates whether the input sample is adversarial or not, while spoofing the target neural network $\mathcal{L}_{GAN}$ with the generated adversarial sample.

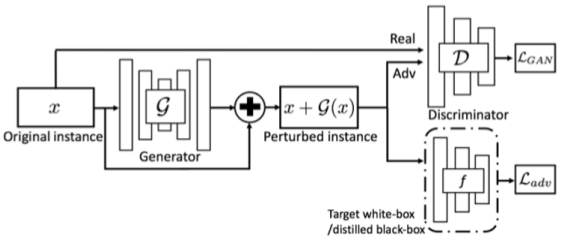


Figure 8: Overview of GanAdv [30].

In the experimental section, the effectiveness of this method [30] is verified against the model after adversarial training. It shows that AdvGAN outperforms FGSM [16] and optimization-based methods [24] in adversarial training (as shown in Figure 10), producing adversarial samples that are visually indistinguishable from real samples. (as shown in Figure 9)

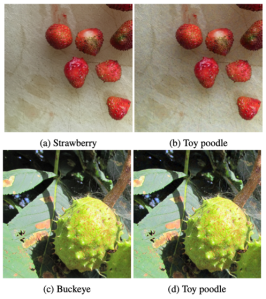


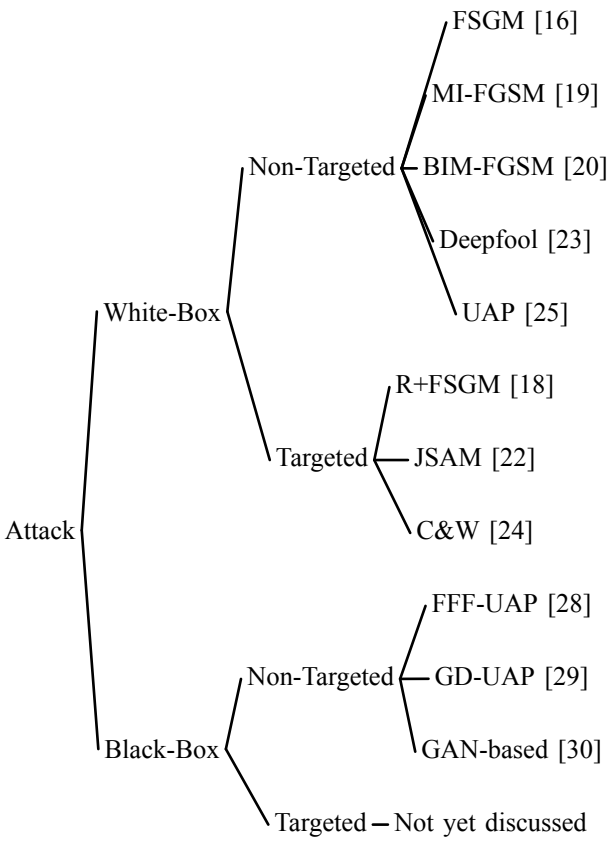
Figure 9: Left: input clean images; right: adversarial images [30].

	MNIST			CIFAR-10		
Defense	FGSM	Opt.	**AdvGAN**	FGSM	Opt.	**AdvGAN**
Adv.	3.1%	3.5%	**11.5%**	13.58%	10.8%	**15.96%**
Ens.	2.5%	3.4%	**10.3%**	10.49%	9.6%	**12.47%**
Iter.Adv.	2.4%	2.5%	**12.2%**	22.96%	21.70%	**24.28%**

Figure 10: Attack success rates of adversarial examples generated by FSGM [16], C&W [24], AdvGAN [30] on MNIST and CIFAR-10 under defensive measures (By different Black-box settings) [30].

E. A short summary of adversarial attacks

According to the subsection II-B and according to the optimal application scenario, the above attack methods can be grouped as follows:



III. Defense Against Adversarial Samples

A. Defensive methods against adversarial samples

1) Detection method based on adversarial sample distributions: The authors argue [32] that the adversarial samples can be explained by the concept of "Manifold". Many training data, such as images, actually exist in a low-dimensional manifold region in a high-dimensional space. The adversarial perturbation does not change the true label of the original data, it simply "pushes" the samples out of the manifold region. The authors' hypothesis is based on this, i.e. that the adversarial perturbation samples are outside the data manifold. Experiments show that adversarial samples crafted to fool DNNs can be effectively detected using two new features: kernel density estimation, and Bayesian neural network.

2) Adding detection sub-network: METZEN et al. [33] proposed the Adversary Detector Network (ADN), which is a detection method that expands a pre-trained neural network with a binary detector network that is trained to distinguish between normal and adversarial samples. The output of the detector represents the probability that the data is an adversarial sample or not. The design of the detector is related to the particular data set and the architecture used is typically a convolutional neural network(CNN).

The ADN method is effective in detecting FGSM [16], Deepfool [23] and BIM [20] attacks. Experiments show that when the FGSM method is used to generate adversarial samples, the detection network can detect 97% of the adversarial samples; when the DeepFool method is used to generate adversarial samples, the detection network can detect 82% of the adversarial samples; And for BIM the detection network can detect 82% of the adversarial samples. The detection network can detect 89%, 87% (Deepfool l_∞), 90% (Deepfool l_2), 85% (BIM l_2), 91% (BIM l_∞) of adversarial samples when the attacker knows the gradients of the classification network and the detection network and performs dynamic adversarial training using the FGSM, Deepfool, BIM methods respectively.

3) Adding binary classifiers: Gong [34] et al. distinguish between adversarial and clean samples in deep neural networks by constructing a binary classifier. This method performs well on MNIST, CIFAR, and SVHN datasets and is robust to second-round adversarial attacks, while it acts as a pre-processing step without imposing any assumptions on the model it protects. However, as this method has limited generalization capability and is sensitive to different adversarial sample generation algorithms, i.e. a binary classifier trained on the $\mathcal{A}_{\text{adv}}$ adversarial dataset does not achieve the same good accuracy in the $\mathcal{B}_{\text{adv}}$ adversarial dataset.

4) Feature Squeezing: XU et al. [35] argue that the dimensionality of the input features is positively related to the size of the attack surface. Based on this principle, they proposed a feature compression-based detection method to compare the prediction results between compressed and uncompressed inputs. The authors first used two compression methods, Squeezing Color Bits and Spatial Smoothing, to reduce degrees of freedom and eliminate adversarial perturbations.

An adversarial sample is detected by first calculating the maximum distance $||d_1-d_2||_1$ between the prediction a of the input image d_1 and the prediction a of the compressed image d_2. If the output produced by the original and compressed inputs differs significantly from the model (the l_1 norm difference between the results is greater than some threshold H, then there is $||d_1 - d_2||_1 > H$), then the original input may be an adversarial sample. (as shown in Figure 11)

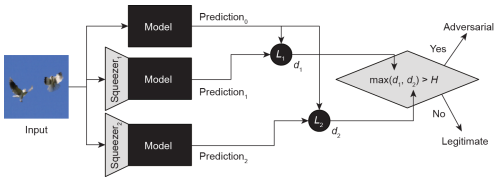


Figure 11: Flowchart of feature squeezing method [35].

5) Defensive distillation: The concept of Distillation was first introduced by Hinton [36] and refers to the transfer of knowledge from a complex network to a simple network. This knowledge is extracted in the form of a class probability vector of training data and fed back to the original model. Papernot [37] proposed defensive distillation, which is an extension of the distillation algorithm. As shown in Figure 12, a distillation model is trained for the original model using the distillation algorithm. When training the distillation model, the input is the set of samples needed to train the original model.

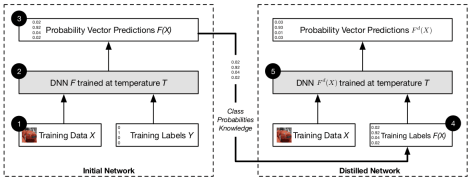


Figure 12: Schematic of Defensive Distillation [37].

The authors tested the spoofing rate of the adversarial samples before and after applying the defensive distillation technique on both MNIST and CIFAR-10 datasets and obtained that the successful spoofing rate of the adversarial samples decreased from 95.86% to 0.45% on the MNIST dataset and from 87.89% to 5.11% on the CIFAR-10 dataset.

Distillation Temperature	MNIST Adversarial Samples Success Rate (%)	CIFAR10 Adversarial Samples Success Rate (%)
1	91	92.78
2	82.23	87.67
5	24.67	67
10	6.78	47.56
20	1.34	18.23
30	1.44	13.23
40	0.45	9.34
50	1.45	6.23
100	0.45	5.11
No distillation	95.89	87.89

Figure 13: Defensive distillation on MNIST and CIFAR10 models [37].

6) Regularization: Ross et al. [38] used input gradient regularization to improve the robustness against attacks by penalizing the degree of output variation concerning the input on the trained objective function, resulting in small adversarial perturbations that do not significantly affect the prediction results of the model.

7) Deep contractive network: Gu et al. [39] introduced the Deep Contractive Network Network (DCN) method based on the idea of CAE [6]. This method uses the smoothing penalty term of the Compressed Auto-Encoder (CAE) in the training process, which serves to make the output of the model smoother by making small changes in the input not changing the hidden layer activation values too much. It is demonstrated that the proposed new method improves the robustness of the neural network to adversarial samples.

8) Defensive approach based on RealWorld observations: Zantedeschi et al. [40] propose two defense strategies, Bounded ReLU and Gaussian data augmentation, by using the Bounded ReLU activation layer and adding a certain amount of noise to the original input data, allowing for enhanced generalization of the model while gaining some robustness to adversarial samples. The experiments also show that the combination of the two strategies (the former as a constraint and the latter as a training data type) enables the model to resist various adversarial attacks.

9) Magnet: Meng et al. [41] proposed a defense framework that does not rely on adversarial samples using only the features of the data itself, and introduced an autoencoder to make the framework more generalizable. This framework consists of several detector modules and a reformer module. First, the framework models the clean sample, like a curve in the two-dimensional case, and the detector determines whether the sample is adversarial or not based on the distance between the manifold boundary of the adversarial sample and the clean sample. In the test phase, the detector determines whether the sample is adversarial or not and if the result is ”yes”, then the sample is thrown out. For samples where the detector cannot detect the adversarial nature (small perturbation and the detector’s decision is ”no”), the reformer will reconstruct the input sample and find a sample close to the original sample to generalize the input sample and input it into the classifier. This framework is not ideal for white-box attacks, but it is effective against black-box attacks and performs well in the case of gray-box attacks using different autoencoders.

10) Defensive based on GAN: This defense method is based on GAN [31]. A generative adversarial net (GAN) is a powerful tool that can be used to learn data distributions and form generators. A large body of work has attempted to use GAN to learn clean data distributions to generate clean predictions with adversarial inputs. Defense-GAN (D-GAN) [42] is representative of such work.

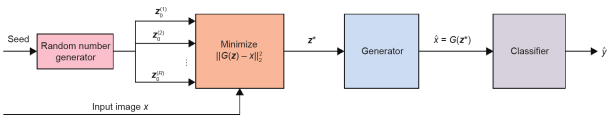


Figure 14: Flowchart of Defense-GAN [42].

The generators of Defence-GAN are trained unsupervised on a clean sample training set, and during training, no adversarial samples are used. At the same time, the classifier is trained on the same real dataset. If the training is good and representative of the clean image distribution, then a minimum distance between the input image and the adversarial samples can be derived by l_2, thus distinguishing the clean samples, which are then sent to the classifier. The success rate of the Defence-GAN method is highly correlated with the performance of the GAN network.

ϵ	Defense-GAN-Rec MNIST	Defense-GAN-Rec F-MNIST
0.10	0.9864 ± 0.0011	0.8844 ± 0.0017
0.15	0.9836 ± 0.0026	0.8267 ± 0.0065
0.20	0.9772 ± 0.0019	0.7492 ± 0.0170
0.25	0.9641 ± 0.0001	0.6384 ± 0.0159
0.30	0.9307 ± 0.0034	0.5126 ± 0.0096

Figure 15: Accuracy of classification of model using Defense-GAN [42].

11) Defense based on Adversarial Training: Adversarial training is often considered to be the most effective method for improving the robustness of deep neural networks [43]. During adversarial training, small perturbations are added to the sample, usually generated by the model itself, and the deep neural network is then adapted to be robust to unknown (non-modeled) adversarial samples.

Goodfellow et al. [16] first added adversarial samples to the model training phase, based on the MINST dataset.

The experiments showed that the model after training using adversarial training was more robust against the adversarial samples generated by FGSM.

Adversarial training is good for improving the robustness of the model, but it is time-consuming and computationally expensive during training. In these two works [44, 45] such problems are reduced and methods to improve adversarial training are proposed in this work [46].

IV. Summary and Future Works

This work introduces approaches to adversarial attacks and defenses in deep learning for “image data”, discusses definitions, and classifies them. This gives a clear picture of the progress and research thinking in this area, but at the same time, it is undeniable that the security challenges in the field of deep learning, and indeed in machine learning, are still enormous:

1) The lack of a strong universal defense system. Often, a particular defense strategy is only available against a limited number of attacks. When faced with new ”threats”, defensive countermeasures are often ineffective.
2) To date, the problem of deep learning “robustness” is still in its “infancy”. Some of the existing work proposes evaluation methods, but does not provide clear estimates of accuracy and extensibility, and often does not provide complete and reliable analysis data.
3) Most of the adversarial research has been on image data (in the field of computer vision). As other fields such as natural language processing, voice recognition, and machine learning for cyberspace security development, are increasingly confronted the security problem is also becoming increasingly serious.

Through research and analysis of previous works and inspired by the above challenges, I believe that future research on the ”adversarial” aspects of deep learning can be carried out from the following perspectives:

- (for challenges 1, 3) To establish a cross-domain security system for deep learning models. This system can be applied to different domains, and it contains a sound robust evaluation system and information on many different types of adversarial samples. For different adversarial attacks, this system can evaluate the robustness of the target model against the attack promptly and provide a defense strategy with a high degree of confidence.
- (for challenge 2) The establishment of such systems is premised on the need to further improve robustness evaluation methods.

The design and development of such systems, while taking into account the cost of deployment and the efficiency of the system itself, is an issue worth exploring and studying.

The existence of adversarial samples is a “double-edged sword” that reveals the vulnerability of deep neural networks. However, it is undeniable that its use as a criterion for evaluating the robustness of neural networks can lead to more excellent researchers joining the field of adversarial machine learning and obtaining more results on defensive strategies, thus making deep learning models more robust. It can be seen that adversarial attacks and defenses are complementary to each other. Thereby, further building safe deep learning models is a topic well worth investigating.

References

[1] Krizhevsky Alex, Sutskever Ilya, Hinton Geoffrey E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. — 2012. — Vol. 25.

[2] Rabiner Lawrence R. A tutorial on hidden markov models and selected applications in speech recognition // Proceedings of the IEEE. — 1989. — Vol. 77, no. 2. — P. 257–286.

[3] Graves Alex, Mohamed Abdel-rahman, Hinton Geoffrey. Speech recognition with deep recurrent neural networks // 2013 IEEE international conference on acoustics, speech and signal processing / Ieee. — 2013. — P. 6645–6649.

[4] Efficient estimation of word representations in vector space / Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean // arXiv preprint arXiv:1301.3781. — 2013.

[5] End to end learning for self-driving cars / Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski et al. // arXiv preprint arXiv:1604.07316. — 2016.

[6] Deepface: Closing the gap to human-level performance in face verification / Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, Lior Wolf // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. — 2014. — 09.

[7] Namiot Dmitry, Ilyushin Eugene, Chizhov Ivan. Military applications of machine learning // International Journal of Open Information Technologies. — 2021. — Vol. 10, no. 1. — P. 69–76.

[8] Dixon Matthew F, Halperin Igor, Bilokon Paul. Machine learning in Finance. — Springer, 2020. — Vol. 1170.

[9] Bhardwaj Rohan, Nambiar Ankita R., Dutta Debojyoti. A study of machine learning in healthcare. — 2017. — Vol. 2. — P. 236–241.

[10] Machine learning security: Threats, countermeasures, and evaluations / Mingfu Xue, Chengxiang Yuan, Heyi Wu et al. // IEEE Access. — 2020. — Vol. 8. — P. 74720–74742.

[11] Safety verification of deep neural networks / Xiaowei Huang, Marta Kwiatkowska, Sen Wang, Min Wu // International conference on computer aided verification / Springer. — 2017. — P. 3–29.

[12] Intriguing properties of neural networks / Christian Szegedy, Wojciech Zaremba, Ilya Sutskever et al. // arXiv preprint arXiv:1312.6199. — 2013.

[13] GUO Lili, DING Shifei. Research progress on deep learning // Computer Science. — 2015. — Vol. 42(5). — P. 28–33.

[14] Wu Lei, Zhu Zhanxing. Towards understanding and improving the transferability of adversarial examples in deep neural networks // Proceedings of The 12th Asian Conference on Machine Learning / Ed. by Sinno Jialin Pan, Masashi Sugiyama. — Vol. 129 of Proceedings of Machine Learning Research. — PMLR, 2020. — 18–20 Nov. — P. 837–850. — URL: https://proceedings.mlr.press/v129/wu20a.html.

[15] Le Quoc V. Building high-level features using large scale unsupervised learning // 2013 IEEE international conference on acoustics, speech and signal processing / IEEE. — 2013. — P. 8595–8598.

[16] Goodfellow Ian J, Shlens Jonathon, Szegedy Christian. Explaining and harnessing adversarial examples // arXiv preprint arXiv:1412.6572. — 2014.
[17] Going deeper with convolutions / Christian Szegedy, Wei Liu, Yangqing Jia et al. // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2015. — P. 1–9.
[18] Ensemble adversarial training: Attacks and defenses / Florian Tramèr, Alexey Kurakin, Nicolas Papernot et al. // arXiv preprint arXiv:1705.07204. — 2017.
[19] Boosting adversarial attacks with momentum / Yinpeng Dong, Fangzhou Liao, Tianyu Pang et al. // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2018. — P. 9185–9193.
[20] Kurakin Alexey, Goodfellow Ian, Bengio Samy et al. Adversarial examples in the physical world. — 2016.
[21] Rethinking the inception architecture for computer vision / Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe et al. // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 2818–2826.
[22] The limitations of deep learning in adversarial settings / Nicolas Papernot, Patrick McDaniel, Somesh Jha et al. // 2016 IEEE European symposium on security and privacy (EuroS&P) / IEEE. — 2016. — P. 372–387.
[23] Moosavi-Dezfooli Seyed-Mohsen, Fawzi Alhussein, Frossard Pascal. Deepfool: a simple and accurate method to fool deep neural networks // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 2574–2582.
[24] Carlini Nicholas, Wagner David. Towards evaluating the robustness of neural networks // 2017 ieee symposium on security and privacy (sp) / IEEE. — 2017. — P. 39–57.
[25] Universal adversarial perturbations / Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2017. — P. 1765–1773.
[26] Simonyan Karen, Zisserman Andrew. Very deep convolutional networks for large-scale image recognition // arXiv preprint arXiv:1409.1556. — 2014.
[27] Deep residual learning for image recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 770–778.
[28] Mopuri Konda Reddy, Garg Utsav, Babu R Venkatesh. Fast feature fool: A data independent approach to universal adversarial perturbations // arXiv preprint arXiv:1707.05572. — 2017.
[29] Mopuri Konda Reddy, Ganeshan Aditya, Babu R Venkatesh. Generalizable data-free objective for crafting universal adversarial perturbations // IEEE transactions on pattern analysis and machine intelligence. — 2018. — Vol. 41, no. 10. — P. 2452–2465.
[30] Generating adversarial examples with adversarial networks / Chaowei Xiao, Bo Li, Jun-Yan Zhu et al. // arXiv preprint arXiv:1801.02610. — 2018.
[31] Generative adversarial nets / Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza et al. // Advances in neural information processing systems. — 2014. — Vol. 27.
[32] Detecting adversarial samples from artifacts / Reuben Feinman, Ryan R Curtin, Saurabh Shintre, Andrew B Gardner // arXiv preprint arXiv:1703.00410. — 2017.
[33] On detecting adversarial perturbations / Jan Hendrik Metzen, Tim Genewein, Volker Fischer, Bastian Bischoff // arXiv preprint arXiv:1702.04267. — 2017.
[34] Gong Zhitao, Wang Wenlu, Ku Wei-Shinn. Adversarial and clean data are not twins // arXiv preprint arXiv:1704.04960. — 2017.
[35] Xu Weilin, Evans David, Qi Yanjun. Feature squeezing: Detecting adversarial examples in deep neural networks // arXiv preprint arXiv:1704.01155. — 2017.
[36] Hinton Geoffrey, Vinyals Oriol, Dean Jeff. Distilling the knowledge in a neural network // arXiv preprint arXiv:1503.02531. — 2015.
[37] Distillation as a defense to adversarial perturbations against deep neural networks / Nicolas Papernot, Patrick McDaniel, Xi Wu et al. // 2016 IEEE symposium on security and privacy (SP) / IEEE. — 2016. — P. 582–597.
[38] Ross Andrew Slavin, Doshi-Velez Finale. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients // Thirty-second AAAI conference on artificial intelligence. — 2018.
[39] Gu Shixiang, Rigazio Luca. Towards deep neural network architectures robust to adversarial examples // arXiv preprint arXiv:1412.5068. — 2014.
[40] Zantedeschi Valentina, Nicolae Maria-Irina, Rawat Ambrish. Efficient defenses against adversarial attacks // Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. — 2017. — P. 39–49.
[41] Meng Dongyu, Chen Hao. Magnet: a two-pronged defense against adversarial examples // Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. — 2017. — P. 135–147.
[42] Samangouei Pouya, Kabkab Maya, Chellappa Rama. Defense-gan: Protecting classifiers against adversarial attacks using generative models // arXiv preprint arXiv:1805.06605. — 2018.
[43] Improving the robustness of deep neural networks via stability training / Stephan Zheng, Yang Song, Thomas Leung, Ian Goodfellow // Proceedings of the ieee conference on computer vision and pattern recognition. — 2016. — P. 4480–4488.
[44] Adversarial training for free! / Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi et al. // Advances in Neural Information Processing Systems. — 2019. — Vol. 32.
[45] Wong Eric, Rice Leslie, Kolter J Zico. Fast is better than free: Revisiting adversarial training // arXiv preprint arXiv:2001.03994. — 2020.
[46] Bag of tricks for adversarial training / Tianyu Pang, Xiao Yang, Yinpeng Dong et al. // arXiv preprint arXiv:2010.00467. — 2020.