Research and Implementation of Deep Learning Counter Attack Defense Strategy

Zhuonan Chang
Internet of Things Engineering major of International School in Beijing University of Posts and Telecommunications China

changzhuonan@bupt.edu.cn

Abstract. In recent years, deep learning technology has made excellent achievements in many fields, such as computer vision, natural language processing, and speech processing. More and more related applications have appeared, which has brought a lot of convenience to people's lives. However, while deep learning performs well, it also has the flaws in being vulnerable. The attack method can make the deep learning model make wrong judgments by adding some small disturbances to the input samples. This brings huge security issues to deep learn applications At present, there have been many research results from defense against attacks. This article first analyzes a variety of classic adversarial attack methods in detail, and classifies these attack methods according to their attack scope. The differences and common points are abstract from the classification, and it is found that there are sensitive points in the counter disturbance, and the fluctuation of the sensitive points affects the classification of the deep learning model. Inspired by this, we propose a defense strategy that only filters sensitive points in the adversarial sample, avoiding the processing of non-sensitive points, reducing calculations, and improving efficiency.

Keywords: Deep learning; adversarial attack; defense strategy; sensitive points.

1. Deep learning
Deep learning is a kind of deep artificial neural network. Deep learning realizes the classification of data by learning the characteristics of the data. It usually consists of three parts: input layer, hidden layer and output layer, as shown in Figure 1. In the figure, each circle represents a neuron, each neuron is a perceptron, the connections between neurons represent model parameters, the activation function in the neuron is responsible for activating the input value, and the activated value is again The input of a layer of neurons until it reaches the output layer to achieve classification. The hidden layer usually has multiple layers, which can be divided into convolutional layer and pooling layer according to function. Convolutional layer is mainly used to extract features and share parameters, and pool The layer is mainly used to reduce parameters and avoid overfitting [1].

The excellent feature extraction ability of deep learning has made it a great success in many fields such as computer vision. Many complex problems solved by deep learning have reached or even surpassed human level. However, studies have shown that deep learning technology also has serious security problems, such as being easily attacked by adversarial samples to generate wrong categories.
1.1. Introduction to Convolutional Neural Networks.

Convolutional neural network (CNN) is a type of deep learning. It is inspired by the study of the principles of human brain cognition [2]. Human cognition of objects is completed by layer-by-layer grading. The original signal is taken into the cerebral cortex, and then the relevant cells of the cerebral cortex find the direction and edge. Then the brain abstracts the shape of the object of the edge and direction, and finally the brain further abstracts the type of the object. Convolutional neural network is to imitate this way of human cognition of objects, constructing multi-layer neural network, the lower layer network is used to extract image features, and then the lower layer features constitute the upper layer features, and the upper layer neural network further extracts the features, And finally make the classification in the top-level neural network [3].

Convolutional neural network was first proposed by Yann Lecun in 1994. It was used for the classification of MNIST data sets. It achieved extremely good results at that time. Since then, researchers have opened the door to convolutional deep learning research [4]. Various convolutions Deep learning models and various optimization methods are emerging one after another. Many deep learning models existed before the advent of convolutional deep learning, but these models are often fully connected networks, with many model parameters, and the cost of training the model is very high. Under the conditions of hardware development at that time, training the model is almost difficult to achieve one thing. Convolutional deep learning is different from previous deep learning [5]. It uses a combination of convolution kernel and pooling layer to greatly reduce model parameters, and not only does not reduce the accuracy of the model, it even improves a level.

A typical convolutional neural network usually consists of a convolutional layer, a pooling layer and a fully connected layer. Among them, the pooling layer forms multiple convolutional groups immediately after the convolutional layer, proposes features layer by layer and transfers the features, and finally completes the classification through the fully connected layer [5]. We use \( \{W^l, B^l\} \) to represent the parameters of the convolutional deep learning model, where \( W^l \) and \( B^l \) represent the weight and deviation of the convolution kernel on layer \( l \), then the feedforward operation of the model is as follows:

\[
A^l(X) = f_p \left( f_a \left( W^l \ast A^{l-1}(X) + B^l \right) \right) \tag{1}
\]

Among them, the symbol \( \ast \) represents the convolution operation, and \( f_a \) represents the activation function, mainly ReLu, PRelu, etc. \( f_p \) means pooling operation, usually there are maximum pooling and
average pooling. Pooling operation can be used to reduce the dimension, it not only reduces the computational complexity, but also improves the generalization ability of the model. AL represents the feature map after the execution of the convolutional layer and the pooling layer. When the feature map is passed into the fully connected layer, all the feature maps need to be flattened and converted into vectors. Finally, the prediction is made by the fully connected layer. When the prediction is incorrect, we use the cross-direction loss function to calculate the distance between the predicted category and the actual label, and then update the connection weight of the model through the back propagation algorithm [6]. The back propagation algorithm helps the model reduce the value of the loss function and speed up the convergence of the model. The optimization of the model is shown in the following formula:

$$\Theta = \arg \min(\Theta, \ D) \tag{2}$$

1.2. Convolutional layer
The convolutional layer is the core part of the convolutional neural network. Each convolutional layer is composed of a set of convolution kernels. The calculation process of convolution can be regarded as a two-dimensional sliding window. Each convolution kernel slide from left to right and from top to bottom, and each sliding calculate the sum of the product of the value of the corresponding position of the image and the value of the convolution kernel. As the new value of the corresponding position of the feature map, the step length of each sliding is a hyper parameter, which is manually specified. Each convolution kernel will calculate a corresponding feature map[7].

In the convolution process, there are two hyperparameters, strides and zero padding, which determine the result of each convolution. When the step length is 1, the distance is one pixel at a time; when the step length is 2, the distance is 2 pixels at a time, and so on. Usually the step size is 1 is the most common, and the size of the feature map after convolution will be different if the moving step size is different. Zero padding is mainly used to control the output result of the convolution kernel. Zero padding refers to padding a specified number of zero values on the boundary of the original image, which can be used to control the size of the convolution result [8].

1.3. Pooling layer
In convolutional deep learning, usually the pooling layer and the convolutional layer appear at the same time, and the pooling layer is used to pool the feature maps obtained by the convolutional layer. The main function of the pooling layer are to reduce the dimension, that is, to reduce the number of parameters, which will greatly reduce the computational cost of training the network. At the same time, the introduction to pooling adds random operations to avoid overfitting [6]. The most common pooling currently includes maximum pooling and average pooling. Maximum pooling means that the maximum value in the coverage of the pooling filter are selected as the result value every time a sliding window; average pooling means that the average value of all pixel value "in the coverage of the filter is calculated as the result every time the window is sliding. value It can be found that the pooling operation is actually down-sampling, and each pooling will greatly reduce the number of parameters.

The pooling layer also has two hyper-parameters: step length and zero padding. The step size of the pooling layer has the same effect as the step size of the convolutional layer, and it is also used to control the distance moved during each pooling. Zero padding is to pad zero at the boundary of the feature map obtained by the pooling layer. Since the zero value has little effect on the maximum pooling, zero padding is usually not used in the pooling operation.

2. Defense strategy based on adversarial attacks
So far, researchers have proposed a variety of counter-attack methods, with different attack principles. The advent of new counter-attack methods will promote the emergence of new defense methods; the emergence of new defense methods will promote the emergence of new defense methods. Attack and defense oppose each other and promote each other. As the saying goes, knowing oneself and one another can survive a hundred battles. We start with the analysis of the attack principle in order to derive the
defense strategy from the attack method. According to the different attack scope, we divide the attack methods of two categories, one is global attack, and the other is local attack.

2.1. Global attack
Global attack refers to the attack method adding disturbance to the entire input sample. The difference is that some pixels have a larger disturbance range, and some pixels have a smaller disturbance range, as shown in Figure 2.

![Figure 2 Global disturbance](image1)

![Figure 3 Local disturbance](image2)

2.2. Local attack
Local attack means that the attack method adds perturbation pixels to one or more pixels in the entire input sample. The number or percentage of disturbed pixels is often a hyperparameter and can be specified manually. As shown in Figure 3.

This type of attack method often uses evolutionary algorithms to find the key components that have a greater impact on the output of the deep learning model, and then adds small disturbances to the key components to generate adversarial samples. The advantage of this attack method is that it does not need to know the parameters of the deep learning model, and it can successfully attack some non-differentiable network models. The main counter attack methods based on evolutionary algorithms are One-Pixel Attack and LocSearchAdv [9].

2.3. Defense Strategy
Start with the principle of counter-attack methods and compare global attacks. Although there are many ways to counterattack attacks, they can be divided into two categories in terms of the scope of the attack. The success rate of these two types of attacks is quite high. It's just that the global attack takes all pixels into consideration by calculating the gradient, while the local attack only find some sensitive points to add disturbance.

Combining the abstract formulas of the two types of attack methods together, it can be found that if only certain components have relatively large numerical changes during a global attack, and other points have relatively small changes that can be ignored, then the constraints of the global attack are approximately local. The constraints of the attack. This tells us that the two types of adversarial sample generation methods are essentially looking for some key components, and then add some disturbances on these key components, and the disturbances added on the key components are usually more obvious, the purpose is to achieve deception. The purpose of the deep learning model [10].
3. Design and implementation of defense strategies
Since this topic requires experimental defense methods of the defense effects of ResNet, AlexNet and LeNet, we need to design these three classic deep learning models. The data set for the deep learning model we implemented is Cifar-10 and MNIST. The Cifar-10 data set categories are airplanes, cars, trucks, boats, frogs, birds, horses, cats, dogs, and deer. The size of each image is 32x32, the RGB color channel is 3, the MNIST category is 0 to 9, the size of each picture is 28x28, and the black and white channel is 1. That is, both data sets are 10 classification problems [11]. Therefore, the deep learning model structure required for the classification of these two data sets is basically the same in the experiment of this subject. The difference is that the size of the input layer is different. For the convenience of the experiment and inspired by transfer learning, we can first design the Cifar-10 data set. The above deep learning model, its input layer is 32x32x3, reuse its network structure and then add another layer before the input layer to become a new input layer size of 8x28x1, and make the output of this layer 32x32x3. In this way, there is no need to redesign the deep learning model for MNIST classification.

When using the FGSM attack method to generate adversarial samples, the variable $\mathcal{E}$ are used to control the magnitude of the perturbation. When the value of the variable is small, the generated adversarial samples are difficult to detect with the human eye to be different from the original samples [12]. This is a sample that is not easily perceptible. Generation is particularly difficult. Experiments have found that when $\mathcal{E} = 0.01$, the ratio of the validation set that can generate adversarial examples is 1.300, and only 140 images in the verification set of about 10,000 can generate adversarial examples. However, as the value of the value continues to increase, it can be generated. The proportion of adversarial samples will be significantly improved. For example, when $\mathcal{E} = 0.55$, the image of adversarial samples can be generated in the verification set to 99.565. However, as the value increases, the difference between the generated sample and the original sample will be particularly obvious, and even the human eye can hardly recognize the image category. We do not consider this sample to be an adversarial sample, because this sample is not only deceiving Deep learning also to deceive human eyes. Therefore, we choose between the two and choose an $\mathcal{E}$ value to ensure the attack success ratio and prevent the disturbance added to the original sample of being too obvious. The experiment found that it is more appropriate to take 0.5 to 5. At this time, the ratio of the verification set that can generate adversarial samples is $82.79070^{[13]}$.

4. Conclusion
Aiming at the shortcomings and deficiencies in current defense methods, this article attempts to propose an efficient and universal defense method. This article first analyzes a variety of classic counter attack methods in detail, and classifies these attack methods according to the attack range, abstracts the common points and differences from the classification, and finds that there are sensitive points in the confrontation sample. Inspired by the second, a search and filter are proposed. Defense methods of sensitive points of the sample.

Improve the universality of defense methods. Universality includes two aspects. One is the generalization of attack methods. Yes, some defense methods are only effective against specific attack methods. The defense method we designed is black box defense. The essence of all attack methods is to add disturbance to the original sample. We only care about the magnitude and location of the disturbance when defending. It doesn't care what kind of attack method is used to generate the disturbance, which makes this method applicable to multiple attack methods and solves the shortcomings of white box defense. The second is the universality of network models. For some defense methods to be effective, they involve operations such as loss function differentiation, which leads to a limited scope of application. We use differential evolution algorithm to find sensitive points. This algorithm does not involve the internal structure and loss functioned as the network model. And so on, you only need to know the input and output of the model, which expands the scope of application.

Improve the execution efficiency of defense methods. This defense method mainly improves defense efficiency from two aspects. The first are to use local filtering instead of global filtering. Many current
defense methods deal with all disturbances to achieve the purpose of defense, such as data compression. However, we found that there are sensitive points in the adversarial sample, which can be achieved by only defending a few sensitive points. The defensive effect reduces the amount of calculation and at the same time reduces the damage to the sample. The second is that our proposed strategy for filtering sensitive points takes the average of the pixel values of neighboring non-sensitive points around the sensitive point as the new value of the sensitive point, which does not involve operations such as differentiation, and the calculation is simpler.

References
[1] The Pascal Visual Object Classes Challenge: A Retrospective[J]. Mark Everingham, S. M. Ali Eslami, Luc Gool, Christopher K. I. Williams, John Winn, Andrew Zisserman. International Journal of Computer Vision. 2015 (1)
[2] Deep learning in neural networks: An overview[J]. Jürgen Schmidhuber. Neural Networks. 2014
[3] A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms[J]. Pinar Civicioglu, Erkan Besdok. Artificial Intelligence Review. 2013 (4)
[4] Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces.[J]. Rainer Storn, Kenneth Price. Journal of Global Optimization. 1997
[5] Deep learning. LeCun Y, Bengio Y, Hinton G. Nature . 2015
[6] The limitations of deep learning in adversarial settings. Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, Ananthram Swami. Security and Privacy (Euro S&P), 2016 IEEE European Symposium on. 2016
[7] Towards deep learning models resistant to adversarial attacks. MADRY A, MAKELOV A, SCHMIDT L, et al. 6th International Conference on Learning Representations. 2018
[8] ImageNet:Large scale visual recognition challenge. Russakovsky O, Deng Jia, Su Hao, et al. International Journal of Computer Vision. 2015
[9] Ensemble adversarial training:attacks and defenses. TRAMER F, KURAKIN A, PAPERNOT N, et al. 6th International Conference on Learning Representations. 2018
[10] Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. ROSS A S, DOSHI-VELEZ F. AAAI Conference on Artificial Intelligence. 2018
[11] Explaining and Harnessing Adversarial Examples. GOODFELLOW I J, SHLENS J, SZEGEDY C. International Conference on Learning Representations. 2015
[12] Deep residual learning for image recognition. K. He, X. Zhang, S. Ren, et al. Computer Vision and Pattern Recognition. 2016
[13] Countering adversarial images using input transformations. GUO C, RANA M, CISSE M, et al. 6th International Conference on Learning Representations. 2018