A Simple Structure for Building a Robust Model

Xiao Tan, Jingbo Gao, Ruolin Li

A Simple Structure For Building A Robust Model*

Xiao Tan[1], Jingbo Gao[1], and Ruolin Li[1]

[1] Xidian Hangzhou Institute of Technology, Zhejiang Province, China
https://hz.xidian.edu.cn
[2] 1203550038@qq.com

Abstract. As deep learning applications, especially programs of computer vision, are increasingly deployed in our lives, we have to think more urgently about the security of these applications.One effective way to improve the security of deep learning models is to perform adversarial training, which allows the model to be compatible with samples that are deliberately created for use in attacking the model.Based on this, we propose a simple architecture to build a model with a certain degree of robustness, which improves the robustness of the trained network by adding an adversarial sample detection network for cooperative training.At the same time, we design a new data sampling strategy that incorporates multiple existing attacks, allowing the model to adapt to many different adversarial attacks with a single training.We conducted some experiments to test the effectiveness of this design based on Cifar10 dataset, and the results indicate that it has some degree of positive effect on the robustness of the model.Our code could be found at https://github.com/dowdyboy/simple_structure_for_robust_model.

Keywords: Robustness · Adversarial Training · Deep Learning

1 Introduction

Currently, applications using deep learning methods are gradually and widely used in our lives[1].In many scenarios, deep learning algorithms and models must be secure.However, models trained by general deep learning methods are often very sensitive to the input data.Small changes in the input data can lead to large deviations in the model output, which makes it possible to spoof the model by falsifying data that is indistinguishable to the human eye.Such artificially created samples used to deceive the model are called adversarial samples[2]. To reduce the negative impact of such problems, there are two mainstream approaches, one is adversarial sample detection[3] and the other is training robust models[4].

We carefully analyzed the existing methods[5] and found that the two solutions can be used not only in combination at the application level, but also in conjunction with each other at the training stage.Therefore, we propose a simple

* Supported by AutoDL.

structure(Fig. 1), which improves the efficiency of robustness training through the intermediate results of adversarial sample detection. We also designed different adversarial sample detection networks and conducted experiments, and found some interesting results(Table. 3).

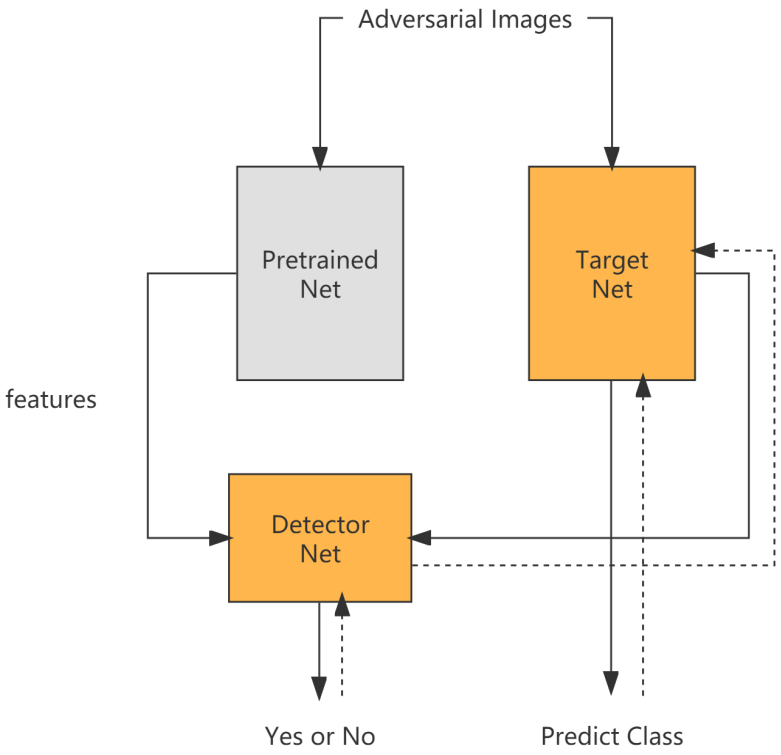

Fig. 1. Pretrained Net and Target Net have the same network structure. The input samples are entered into these two networks separately and the intermediate feature maps of the network outputs are obtained. The two feature maps compute the differences in Detector Net and provide feedback and fine-tuning to the upstream network.

We tested our designed structure using the Cifar10 dataset[6] and found that our scheme can improve the robustness of the model to some extent compared to the common adversarial training method[4].Also, because of the new data sampling strategy, the model is able to be more adaptable to more complex testing environments.

2 Related Work

Here, we will discuss those methods that are relevant to our schema and what they bring to our design.

2.1 Adversarial Attacks

Gaussian Noise The simplest adversarial attack is to add Gaussian noise to the image[7].These Gaussian noises, because they possess local randomness, can interfere with the pixel value distribution of the image.

FGSM FGSM was proposed by Goodfellow et al. and is a classical algorithm in the field of adversarial samples[8].The main reason for the vulnerability of neural networks to adversarial perturbations is their linear nature and, the linear behavior in high-dimensional spaces is sufficient to cause misclassification of samples.

BIM The BIM algorithm[10] uses an iterative approach to search for perturbations at individual pixel points, rather than modifying all pixel points at once as a whole.

DeepFool The DeepFool algorithm[11] adds less noise and takes less time to generate samples compared to the FGSM algorithm.It is based on the classification idea of hyperplane and can accurately calculate the perturbation value.

C&W The C&W algorithm[12] is an optimization-based attack algorithm.It sets a special loss function to measure the difference between the input and the output.

NST Neural Style Transfer is a way to perform style changes on images[13].Since the texture of the image is modified during the transformation of the image, it can obviously be applied to generate adversarial samples as well.

2.2 Adversarial Defensive

Currently popular adversarial defense methods include model distillation[14] and adversarial training[4].Model distillation uses a teacher model to guide the training of the student model, and the teacher model is a pre-trained model.When training is performed, the input data are first entered into the teacher model and the probability distribution of the output is obtained.

Adversarial training is a commonly used method to improve the robustness of models.It is to exploit the powerful expressive power of deep neural networks[15] to improve the robustness of the model by learning adversarial samples.

3 Methods

3.1 Structure Design

Adversarial training is an effective way to resist adversarial samples and improve model robustness.However, due to the large number of adversarial samples added to the training data, this leads to a longer convergence process of the training.Inspired by the design of the auxiliary head[17], our method also adopts a similar design as a way to improve the model convergence speed.

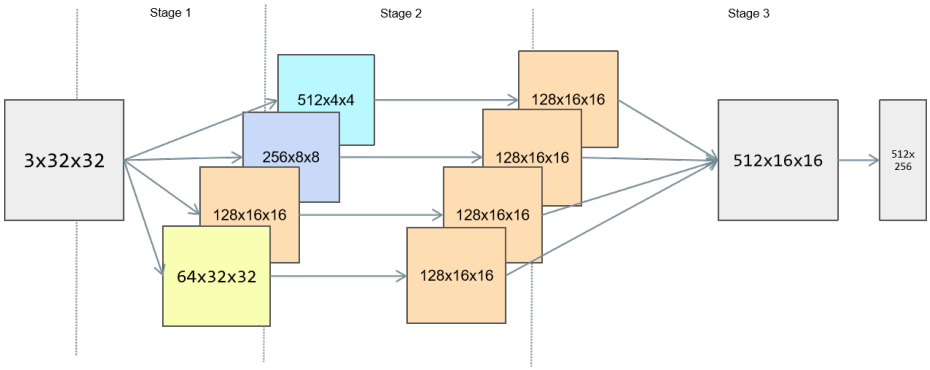

Fig. 2. The adversarial detection network receives the image input and goes through three stages of processing: in the first stage, the data is fed into the backbone network to obtain feature maps of different layers; in the second stage, data embedding is performed, and the feature maps are convolved or deconvoluted and outputted uniformly as feature maps of the same size and number of channels; in the third stage, all the feature maps are stitched and spread into the output that Transformer can receive.

We designed a simple shallow Transformer[18] structure as an auxiliary network to perform the adversarial sample detection task.However, unlike a normal network model, its input is not the original data, but a feature map of the output of an intermediate layer of a network that has been pre-trained on a clean dataset.And, inspired by Vision in Transformer[19], we abandon the use of fully connected layers as embedding layers and use convolutional structures as embedding layers instead(Fig. 2).Since our input needs to fuse feature maps from different levels of the pre-trained network as input, we introduce deconvolution in our embedding layer[20].After experiments, we found that fusing other levels of feature maps with a shallow feature map as the center can have better results.

Because the secondary and primary networks perform different classes of tasks, they cannot simply perform gradient returns and updates separately.For this case, our approach is to capture the output of a specific layer in the auxiliary network and then find the differential performance of the output features of the pre-trained network and the target network on the auxiliary network by using

the Smooth L1 function[9] as part of the loss function.The final loss function consists of the categorical cross-entropy error and the L1 error [equation 1].

$$loss(X,Y,L) = \sum_{i=1}^{N} \alpha \times CrossEntropyLoss(X_i, L_i) + \beta \times SmoothL1Loss(X_i, Y_i) \tag{1}$$

3.2 Sampling Strategy

Classical sampling strategies for adversarial training tend to use a fixed ratio of adversarial samples as input[4], however, this approach may in some cases make the network overfit to the selected adversarial sample data[16]. We designed a more flexible sampling strategy. It uses a dynamic probability with a bounded range to determine the type of adversarial attack to which the sample belongs.

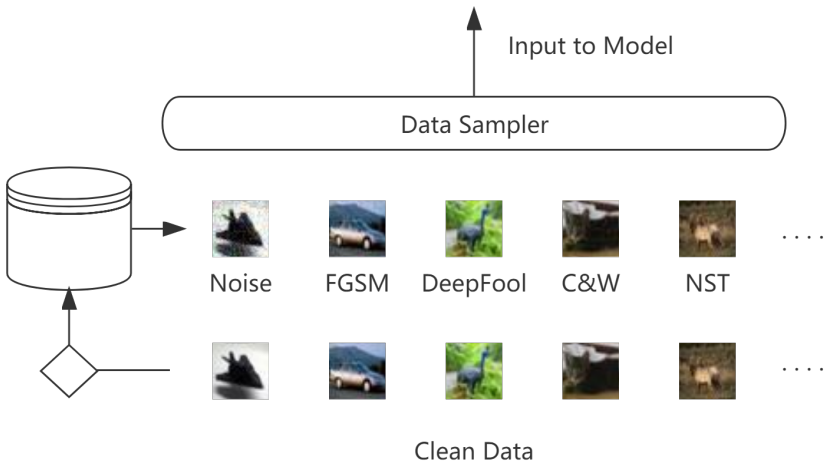

Fig. 3. The Clean dataset is programmed in an offline manner to generate adversarial samples of all categories and stored in a database. The sampler samples the database according to the configured dynamic probability range and feeds the selected samples to the model.

In addition, we take the approach of generating adversarial samples offline (Fig. 3).Adversarial samples of all attack types are generated for each sample in the training dataset and stored in the database before starting training.

4 Experiments

For our proposed model architecture, we conducted a series of experiments.All of our experiments were run on a dual-card RTX A4000 mainframe.The experimental results show that our method is able to make the network model more robust compared to ordinary adversarial training(Table. 2).

Table 1. Intensity of sample noise under different types of adversarial attacks.

Name:	Gaussian	FGSM	BIM	DeepFool	C&W	NST
Value:	0.1	0.005	0.001	0.1	0.005	0.1

We used the Cifar10 dataset[6] for our experiments.We randomly sampled 45,000 images from the original training set as our training set and the remaining 5,000 images as the test set.We generated datasets with adversarial sample ratios of 0.5, 0.75, and 0.25, the types of adversarial samples contain FGSM, DeepFool, etc., and they use the appropriate noise intensity(Table. 1), respectively.

Table 2. Comparison of the prediction results of the ordinary model, the adversarial training model and our method, in terms of accuracy.

Structure	Adv. Data Ratio	Accuracy
Baseline	0.5	0.6778
Adv Train	0.5	0.8464
Adv Train + Detector (ours)	0.5	**0.8688**

We chose ResNet34[21] as our experimental network, and we modified the network model so that it can output all intermediate feature maps.We first performed training on a clean dataset and general adversarial training, after which we conducted experiments using our method on the same dataset(Table. 1), and finally, we conducted some exploratory experiments for different parameter configurations of our method(Table. 3).In addition, experiments were conducted on a single adversarial sample test set in order to more fully validate the expressiveness of the model(Table. 4).The number of iterations in the training phase for all models is 150 epochs.

With the same adversarial sample ratio, our method has some improvement in accuracy over the general adversarial training(Table. 2).Also, our method is more robust to each different type of adversarial attack(Table. 4), which indicates that it does not have a significant bias on the performance improvement and is a more general method to improve the robustness of the model.

We also tried to configure different input types, adversarial sample ratios, and the number of self-attentive modules for the detection network(Table. 3).We find that embedding all level feature maps can improve the accuracy of the

Table 3. Comparison of results on adversarial sample detection and classification for different input types, adversarial sample ratios, and number of self-attentive layers.

Input Type	Adv. Data Ratio	SA Layers	Detector Accuracy	Classification Accuracy
Low	0.5	4	0.734	0.8658
Low	0.75	4	0.8332	0.842
Low	0.75	8	0.8302	0.845
Low	0.5	2	0.735	0.8694
Low	0.75	2	0.8364	0.8416
Full	0.5	4	0.7538	0.8674
Full	0.75	4	0.8458	0.8406
Full	0.75	8	0.8492	0.8446
Full	0.5	2	0.7506	0.8688
Full	0.75	2	0.8536	0.8462

detection network compared to embedding only low level feature maps, but has no significant effect on the classification accuracy of the images.

Table 4. Prediction results of different structures on different types of adversarial sample datasets.

Structure	Clean	Gaussian	FGSM	BIM	DeepFool	C&W	NST
Baseline	0.9012	0.6616	0.307	0.1748	0.1984	0.1074	0.6236
Adv Train	0.8737	0.8282	0.751	0.7906	0.867	0.7736	0.836
Adv Train + Detector (ours)	**0.8916**	**0.8542**	**0.7834**	**0.8296**	**0.8884**	**0.8066**	**0.8592**

5 Conclusion

As the security and robustness of deep learning models are increasingly emphasized, adversarial attacks may gradually become a necessary factor to be considered for training models.An effective way to deal with adversarial attacks is to train the model adversarially, which enhances the performance and robustness of the model in a hostile environment by strengthening its adaptation to adversarial samples.Our proposed method adds a small adversarial sample detection network to the adversarial training by which some additional features of the data are extracted and used to improve the efficiency and performance of the adversarial training.In addition, we design a new offline data sampling strategy that incorporates a variety of adversarial attack types and has stronger randomness, allowing the trained model to be more adaptable to complex environments while improving the training efficiency to a certain extent.

References

1. Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber and Saber Fallah. A Survey of Deep Learning Applications to Autonomous Vehicle Control. IEEE Transactions on Intelligent Transportation Systems, 2020.
2. Julia Lust, Alexandru Paul Condurache. A Survey on Assessing the Generalization Envelope of Deep Neural Networks: Predictive Uncertainty, Out-of-distribution and Adversarial Samples. arXiv:2008.09381, 2021.
3. Reuben Feinman, Ryan R. Curtin, Saurabh Shintre and Andrew B. Gardner. Detecting Adversarial Samples from Artifacts. ICML, 2017.
4. Zhuang Qian, Kaizhu Huang, Qiu-Feng Wang and Xu-Yao Zhang. A Survey of Robust Adversarial Training in Pattern Recognition: Fundamental, Theory, and Methodologies. arXiv:2203.14046, 2022.
5. Sebastian Scher, Andreas Trügler. Robustness of Machine Learning Models Beyond Adversarial Attacks. arXiv:2204.10046, 2022.
6. Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009.
7. Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka and Volkan Cevher. Adversarially Robust Optimization with Gaussian Processes. arXiv:1810.10775, 2018.
8. Ian J. Goodfellow, Jonathon Shlens and Christian Szegedy. Explaining and Harnessing Adversarial Examples. arXiv:1412.6572, 2014.
9. Hong Xuan and Robert Pless. Dissecting the impact of different loss functions with gradient surgery. arXiv:2201.11307, 2022.
10. Alexey Kurakin, Ian Goodfellow and Samy Bengio. Adversarial examples in the physical world. arXiv:1607.02533, 2016.
11. Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi and Pascal Frossard. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. CVPR, 2016.
12. N. Carlini and D. Wagner. Towards Evaluating the Robustness of Neural Networks. IEEE, 2017.
13. Yanghao Li, Naiyan Wang, Jiaying Liu and Xiaodi Hou. Demystifying Neural Style Transfer. IJCAI, 2017.
14. Bharat Bhusan Sau and Vineeth N. Balasubramanian. Deep Model Compression: Distilling Knowledge from Noisy Teachers. arXiv:1610.09650, 2016.
15. Or Sharir, Amnon Shashua and Giuseppe Carleo. Neural tensor contractions and the expressive power of deep neural quantum states. arXiv:2103.10293, 2021.
16. Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S. Dhillon and Cho-Jui Hsieh. The Limitations of Adversarial Training and the Blind-Spot Attack. ICLR, 2019.
17. Joel Ye, Dhruv Batra, Erik Wijmans and Abhishek Das. Auxiliary Tasks Speed Up Learning PointGoal Navigation. CoRL, 2020.
18. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762, 2017.
19. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR, 2021.
20. Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. arXiv:1603.07285, 2016.
21. Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. arXiv:1512.03385, 2015.