Detection of Adversarial Examples for Adversarial Defense

Luping Liu* Hello Ketty

November 17, 2021

Abstract

Neural networks are at the heart of the current rise of artificial intelligence now. However, recent works show that they are vulnerable to adversarial attacks in the form of subtle perturbations to inputs that lead a model to predict incorrect outputs. Therefore, adversarial attacks pose a serious threat to the success of neural networks in practice. In this paper, to address this problem, we propose a simple strategy that addss a corrector before the main models. This corrector is a kind of denoising models and helps the main models to get clean inputs. According to our experiments, this structure can remove the attacks and even fool the attack strategies.

1 Introduction

Neural networks provide huge breakthroughs in solving problems in different fields, such as computer vision [1], natural language processing [2], robotics [3] and speech recognition [4]. However, recent works [5] show that they are vulnerable to adversarial attacks in the form of subtle perturbations to inputs that lead a model to predict incorrect outputs. Such perturbations are often too small to be perceptible for images, yet they completely fool the deep learning models. At the same time, neural networks enter into more security-critical fields, such as self-driving cars [6], surveillance [7] and malware detection [8]. Such a problem becomes more and more serious and poses a serious threat to the success of deep learning in practice.

*Luping Liu, Student ID: 22151314, School: Zhejiang University, Email: 3170105432@zju.edu.cn

In this paper, we try to solve this problem using a simple strategy. We add a corrector before the main models, which is a denoising model. This corrector detects the attack item from an adversarial attack and tries to restore inputs to corresponding clean inputs. As far as we know, there are many choices of denoising models [9–11]. Because of the impressive effects of DDPMs [11], in our experiments, we use a U-Net backbone similar to it.

To determine the effects of our strategy, we test it on the image classification task [12]. We use adversarial attack strategies [13–16] to attack our new models, a combination of a denoising model and a classification model. According to our experiments, our strategy can not only detect attacks, but also confuse the attack strategies.

2 Method

In this section, we show our strategy in detail. We first show the structure of our corrector and classification model and how we combine these two main parts together. After that, we introduce several adversarial attack strategies used in this paper.

2.1 Structure of Model

Classification Model Image classification task is one of the most classic neural network tasks. Many famous and pre-trained models can be used as the main models in our paper, such as VGG [17], ResNet [18] and Inception [19]. For our tasks, we choose ResNet as the main model in our tasks.

Denoising Model Image noise reduction is also an essential task. A very classic model is U-net [20],

which is designed for semantic segmentation. However, U-net has been successfully used in many different fields, such as noise reduction and image generation. In this paper, we choose U-net as our denoising model used before our ResNet model.
In our experiments, we first put the inputs into an U-net to get the noises in the inputs and denoising the inputs, namely, subtracting the noise from the inputs, to get clean inputs. Then, we put these new inputs into our main classification model, ResNet, to get the final classification results.

2.2 Adversarial Attack Strategy

There are many controllable choices of adversarial attack, and we select a representative strategy for testing.
Fast Gradient Sign Method Fast Gradient Sign Method (FGSM) gives a simple strategy to search the noise used to fool classification models. It add such a noise ϵ into the inputs I_c:

$$\mathrm{I}_\epsilon = \mathrm{I}_c + \delta \mathrm{sign}(\nabla \mathcal{J}(\theta, \mathrm{I}_c, l)) \tag{1}$$

Here, $\nabla \mathcal{J}$ computes the gradient of the cost function and sign denotes the sign function and δ is a small scalar value. Some work [14] further develops this strategy, which uses multi-step noising process instead of one-step noising process. It gets Basic Iterative Method (BIM), which satisfies:

$$\mathrm{I}_\epsilon^{i+1} = \mathrm{Clip}\{\mathrm{I}_\epsilon^i + \delta \mathrm{sign}(\nabla \mathcal{J}(\theta, \mathrm{I}_\epsilon^i, l)\} \tag{2}$$

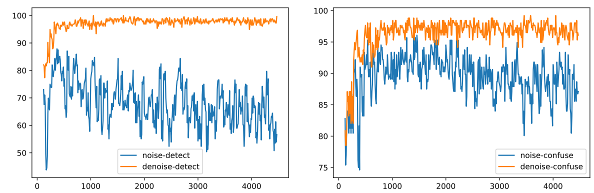
Here, Clip controls the new inputs I_ϵ^i in the input domain, usually, $[-1, 1]^n$.
In the experiments, we will test FGSM to show the ability of our adversarial defense strategy.

3 Experiment

In this section, we show the details and results of our experiments to show the effects of our strategy.

3.1 Detection

This test assumes that the FGSM only attacks the classification model. Then, we use our U-net to detection the noise generated by FGSM, and we find that our U-net successfully removes the noise and makes the prediction results pretty good.



3.2 Confusion

This test assumes that the FGSM attacks the combination of the classification model and the denoising model. Then, we use our U-net to detection the noise generated by FGSM, and we find that our U-net also removes the noise and makes the prediction results pretty good. This test also shows that adding a denoising model can make FGSM less valuable; the effect of FGSM is limited even we do not denoising the results of FGSM.

3.3 Image Result

Here, we put some visible results to show the effects of our strategy in Figure 1. We can find that the image was restored with great success.

Figure 1: Original inputs, dirty inputs and denoising inputs.

4 Conclusion

In this paper, we design a simple strategy to defend the adversarial attack. Our strategy can not only detect attacks but also confuse the attack strategies successfully. In the future, we may test our strategy on more adversarial attack strategies and datasets.

References

[1] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[2] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[3] Robin R Murphy. *Introduction to AI robotics*. MIT press, 2019.

[4] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech. *arXiv preprint arXiv:1905.09263*, 2019.

[5] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[6] Evan Ackerman. How drive. ai is mastering autonomous driving with deep learning. *IEEE Spectrum Magazine*, 1, 2017.

[7] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of big data*, 2(1):1–21, 2015.

[8] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[9] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[10] Yiyun Zhao, Zhuqing Jiang, Aidong Men, and Guodong Ju. Pyramid real image denoising network. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2019.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[12] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[14] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[15] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[16] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017.

[17] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on*

computer vision and pattern recognition, pages 2818–2826, 2016.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.