Dilated Denoising U-Net Network Improving the Adversarial Robustness

Xianshan Zhang and Fan Jia*
Faculty of Electronic Information Engineering, Beijing Jiaotong University, Beijing, China
Email: fjia@bjtu.edu.cn

Abstract. The adversarial attack generates adversarial samples by adding subtle perturbations to the image. Such perturbations are usually very small and undetectable, but the neural network will give completely different results from the real sample. Adversarial example completely fools the neural network. Therefore, it is very important to develop effective defending model. Convolutional neural networks have successfully achieved adversarial robustness by removing noise in adversarial samples. However, the convolutional neural network involves multiple layers, and the denoising model contains a large number of parameters. This paper proposes a dilated convolutional denoising U-Net network to remove adversarial sample noise. Compared with the previously proposed denoising U-Net, our model has fewer layers. Dilated convolution is used in the convolutional layer to expand the receptive field. Use zero padding to ensure that the output dimensions are consistent with the input dimensions. We conducted experiments on the ImageNet dataset. The experiments show that the extended of the receptive field can enhance the ability of the U-Net network to capture detailed image information, enhance denoising performance, and effectively improve the robustness at a lower computing cost.

Keywords. Adversarial examples; neural networks; image denoising; dilated convolution.

1. Introduction
Neural networks have been widely used in various fields. However, neural networks are vulnerable to adversarial attacks [1]. Adversarial attacks can effectively deceive the neural network by adding elaborate and small disturbances to the image. Moreover, the adversarial samples can be transferred between different models [1, 2]. Without knowing the weight and structure of the target model, a black box attack can be carried out on it [3]. This poses a serious threat to the application of neural networks.

Convolutional neural networks have been successfully used for image denoising [4-7]. The process is to train the network with clean images and noisy images, and also learn the mapping of noisy images to clean images. In Ref. [8], a high-level guided denoising U-Net (DUNET) is proposed to learn the noise mapping. Most denoising convolutional neural networks have many layers, and it contains a large number of parameters. The computational cost of training is very high. Experiments show that expanding the receptive field can improve the performance of convolutional neural networks. Increasing the size of the convolution kernel or stacking more convolutional layers can also expand the receptive field, but it inevitably increases the complexity of the model. The size of the convolution kernel is too large, which will reduce the model’s performance. Another way is to add pooling layers, but in the denoising task, the pooling operation will cause information loss and the denoising effect will be poor.

In this paper, we propose a dilated denoising U-Net network for adversarial examples denoising. Our model is inspired by the work of Ref. [8] and aims to learn the noise mapping between the adversarial
samples and the clean images. We can get denoised samples by using adversarial samples minus the noise map. More importantly, we use dilated convolution to expand the receptive field of convolutional layers. Increasing the receptive field allows the model to capture more image details and improve denoising performance. At the same time, we use the Smooth L1 loss function. Model can continue to converge in the later stage of training to achieve higher denoising precision. Experiments show that, in the case of fewer network layers, our model has similar ability to DUNET against adversarial sample denoising. Compared with DUNET, our model has higher computational efficiency.

2. Related Work
In this section, we introduce common methods of adversarial attack and defense.

2.1. Existing Methods for Adversarial Attack

2.1.1. Fast Gradient Sign Method. FGSM [9] is an attack method based on the $L_\infty$ norm. The main idea is to add disturbances in the direction where the model gradient changes the most, and then move the same step in each dimension. Although the step is very small, but the effects of each dimension are added together, which is enough to have a great influence on the discrimination results of the classifier. FGSM adds perturbation to the image by increasing the loss of the classifier. The resulting adversarial sample will cause some slight perturbation to all pixels of the original image.

2.1.2. Basic Iterative Method. Basic Iterative Method is proposed in Ref. [10]. FGSM only adds 1 step of disturbance along the direction of gradient increase, while BIM performs multi-step small disturbance along the direction of gradient increase by iteration. After each small step, the gradient direction is recalculated.

2.1.3. Jacobian-based Saliency Map Attack. JSMA [11] adds a limited number of pixels to the original image to construct an adversarial sample. JSMA modifies one pixel of the original sample at a time and monitors the effect on the output result. It obtains a saliency map by calculating the gradient of the output. A larger saliency map value will increase the probability of misclassification of the classifier. Once the saliency map is calculated, you can change the most effective pixels to deceive the network.

2.1.4. Carlini and Wagner Attack. Carlini and Wagner designed a function that has a smaller value in the adversarial sample, but a larger value in the normal sample so the adversarial sample can be found by minimizing this function. The C&W [12] attack limits the size of the $L_\infty$, $L_2$ and $L_0$ norms of noise while generating adversarial samples, making the perturbations undetectable.

2.1.5. DeepFool. DeepFool [13] iteratively calculates the minimum norm of image’s disturbance. This algorithm continuously adds disturbance to the samples, making the samples cross the decision boundary of the classifier. In each iteration, the image is disturbed by a small vector. When the perturbed image crosses the decision boundary and changes its label, the total perturbation is the final perturbation.

2.2. Existing Methods for Defense
Adversarial training is one of the most extensive methods against adversarial attacks. In Refs. [9, 14, 15], adversarial training is used as the first line of defense. This method adds adversarial samples generated by different attack methods to the training set for network training. Gradient Masking [16] trains a differentiable model and penalizes the degree of change in the output based on the change in input. The purpose of this method is that when the input changes slightly, the KL divergence between the predicted value and the label will not be significantly different. That is a small disturbance will not change the output of the model. Gu and Rigazio propose the deep contrastive network. The training process of DCN [17] uses a smooth loss similar to a compressed autoencoder. Its loss function contains a layer-by-layer compression penalty term. This penalty term approximately minimizes the variance of
the network output relative to the input disturbance. This allows the trained model to achieve "flatness" around the training data so that the output of each layer is not affected by input disturbances.

3. The Proposed Denoising Model

This section discusses the influence of receptive field on convolutional neural network, and then introduces dilated convolutional denoising U-Net network and compares it with DUNET.

3.1. Receptive Field

In the convolutional neural network, the calculation of an element on the feature map is affected by an area on the input image. This area is the receptive field of the element. Intuitively, the larger the receptive field, the richer the image features extracted by neurons. Generally, there are three ways to increase the receptive field. The most direct method is to increase the size of the convolution kernel, but it is not recommended. Large convolution kernels will increase the model parameters. Using several smaller convolution kernels instead of several larger convolution kernels can obtain the same receptive field, and the parameters amount are small [18]. Another way is to use pooling operation. Although it can increase the size of receptive field, it will lose the detail information of the image. Increasing the number of convolutional layers can expand the receptive field, but it will increase the amounts of parameters and computational burden. The advantage of dilated convolution is that without additional computational burden, it can capture more image detail information by increasing the receptive field [19, 20].

3.2. Dilated Denoising U-Net Network

Our model is based on the U-Net network. It is designed to learn the noise mapping. The left side of the convolutional layer is called the contraction path, and the right side is called the expansion path. The contraction path receives image input, obtains context information. It generates a set of feature maps with lower and lower resolutions [21]. The contraction path and the expansion path add horizontal connections to convolutional layers of the same resolution, which are used to transmit fine-scale information [8].

Our model is shown in figure 1. C is our basic block, which contains a 3x3 standard convolution layer, a batch normalization layer, and a linear rectification unit. DilatedC is to convert the standard convolution to a dilated convolution with 2-dilation in the basic block. The number after C indicates the number of basic blocks. F is the fuse unit. The contraction path contains four basic blocks. The first basic block is a standard convolution with a step size of 1x1, and the remaining basic blocks are DilatedC. Each DilatedC expansion rate is 2, and zero padding is set to 2. The expansion path contains three DilatedC basic blocks and a 1x1 convolution. The input of the expansion path is upsampled and superimposed with the output of the horizontal connection as the input of DilatedC. Finally, through a 1x1 regular convolution, the feature map is converted into a noise map d\(\hat{x}\). The denoised samples are obtained by subtracting the noise map from the adversarial samples.

\[
\hat{x} = x^* - d\hat{x}
\]

Because LGD [8] has better performance in noise suppression, we extract the logits of the clean samples and the denoised samples to calculate the loss function. The DUNET model uses the L1 loss function. In the later stage of model training, the absolute value of the derivative of the loss function to the predicted value is still 1, and the loss function will fluctuate around the stable value. It is difficult to continue convergence to achieve higher accuracy. The Smooth L1 loss function avoids such problems. The Smooth L1 loss function is as follows:

\[
L = \begin{cases} 
0.5 \left( f^{-1}(\hat{x}) - f^{-1}(x) \right)^2 & |y| < 1 \\
\left| f^{-1}(\hat{x}) - f^{-1}(x) \right| - 0.5 & other 
\end{cases}
\]

(2)
where $f_{-1}(\hat{x}) - f_{-1}(x)$ represents the difference between the output of the clean sample and the denoised sample in the target model at -1 level.

In Table 1, we compare the receptive field size of our model with the DUNET. Because the expansion path involves upsampling of bilinear interpolation, we only compare the receptive field size of the contraction path. The results show that when the down-sampling step of the DUNET is set to 2, the receptive field of the convolutional layer gradually increases by 2, 4, 8, 16, 32. The 14-layer convolutional receptive field size reaches 185. Our model's convolutional layer receptive field gradually increases by 2, 8, 16, 32. The 12-layer convolutional receptive field size reaches 205, and the minimum resolution of the contraction path output is 38 * 38. Our model has fewer layers (the entire u-net network is reduced by 5 layers), reducing the computational burden, while the receptive field is larger than the DUNET model.

**Table 1.** Receptive field size of our model and DUNET.

| Layer | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Method | Ours | 3  | 5  | 13 | 21 | 29 | 45 | 61 | 77 | 109 | 141 | 173 | 205 | -  | -  |
|        | DUNET | 3  | 5  | 9  | 13 | 17 | 25 | 33 | 41 | 57  | 73  | 89  | 121 | 153 | 185 |

4. Experiment

4.1. Dataset and Network Training
We follow the similar step in [8] to collect the dataset. We extracted 10k images from the ImageNet training set. The image size is cropped to 299*299*3, and five methods of FGSM, BIM, C&W, JSMA, and DeepFool are used to attack the pre-trained Inception V3 model. 50k samples are generated. We extract 4k pictures from the ImageNet training set and generate 20k adversarial samples as the validation set through the above attacks. We extracted 10k pictures from the ImageNet validation set and generated 50k adversarial samples as the test set through five attack methods.

Filter weight initialization is the key to network training. The most commonly used method is random initialization, but Yu in Ref. [19] found that random initialization was not effective for their dilated network when using a dilated network for intensive prediction tasks. In Ref. [22], He et al. proposed ‘MSRA’ initialization, which can converge the deep neural network. We use the ‘MSRA’ method to initialize the filter weights. In training, Adam [23] with momentum of 0.9 is used to optimize the denoiser.
4.2. Results Analysis

We compared the performance of our model with the DUNET (using LGD denoising method) on the test set. The results are shown in table 2. It can be seen that our model achieved similar results to DUNET under five kinds of attacks. Under BIM and JSMA attacks, our model performed even better than DUNET. Although under FGSM, C&W, DeepFool attacks, our model does not exceed DUNET, but our model contains fewer layers and requires less computational cost. In the U-Net structure, our model has 13,000 fewer parameters than the DUNET model. The average training time of our model is 28 hours under one type of adversarial samples, and the test time of one picture is 2.10s. DUNET is 37 hours and 2.47s respectively. DUNET needs more layers to compete with our model, which will bring more computational burden. The comparison shows that with fewer layers, our model is close to DUNET in performance and has higher computational efficiency.

We test the model’s transferability on different attack model. We use five methods to attack the Resnet network to generate adversarial samples. Comparing table 2 with table 3, we can see that the dilated convolutional denoising U-Net network achieves performance similar to it, when the attack model is IncV3. We also test the transferability of model on different classes. The training set contains 750 classes in ImageNet, and the test set contains the remaining 250 classes. As shown in table 4, the dilated convolutional denoising U-Net network still performs well on untrained category datasets.

Table 2. Accuracy rate under five attack methods.

| Method   | Clean | FGSM | BIM  | C&W  | JSMA | DeepFool |
|----------|-------|------|------|------|------|-----------|
| DUNET    | 81.6% | 76.3%| 71.4%| 75.1%| 68.7%| 79.2%     |
| Ours     | 80.9% | 75.9%| 72%  | 73.9%| 70%  | 77.3%     |

Table 3. Transferability analysis to different attack model.

| Method   | Clean | FGSM | BIM  | C&W  | JSMA | DeepFool |
|----------|-------|------|------|------|------|-----------|
| DUNET    | 81.6% | 75.7%| 71.1%| 75.2%| 67.7%| 79%       |
| Ours     | 80.8% | 74.9%| 70.3%| 74.9%| 70.2%| 77.5%     |

Table 4. Transferability analysis to different classes.

| Method   | Clean | FGSM | BIM  | C&W  | JSMA | DeepFool |
|----------|-------|------|------|------|------|-----------|
| DUNET    | 81.4% | 73.9%| 69.7%| 74.1%| 65.2%| 77.9%     |
| Ours     | 80.7% | 72.7%| 70%  | 73.6%| 66.8%| 76.2%     |

5. Conclusion

This paper proposes a dilated convolutional denoising U-Net network for adversarial sample denoising, which shows strong robustness under five kinds of adversarial attack. Dilated convolution can increase the receptive field more effectively than standard convolution and extract rich image detail information. In addition, the model uses the Smooth L1 loss function. When the difference between the denoised sample and the clean sample is small, the model can continue to converge and reach a higher denoising level. With fewer layers and less computational burden, our model is still robust, and it is still comparable to the state-of-the-art adversarial example denoising method.

References

[1] Christian S, Wojciech Z, Ilya S, Joan B, Dumitru E, Ian G and Rob F 2013 Intriguing properties of neural networks (arXiv preprint arXiv:1312.6199).

[2] Nicolas P, Patrick M and I Goodfellow 2016 Transferability in machine learning: from phenomena to black-box attacks using adversarial samples (arXiv preprint arXiv:1605.07277).
[3] Nicolas P, Patrick M, Goodfellow I, Somesh J, Berkay C Z and Ananthram S 2017 Practical black-box attacks against machine learning ACM Asia Conference on Computer and Communications Security pp 506-519.

[4] Viren J and Sebastian S 2009 Natural image denoising with convolutional networks Advances in Neural Information Processing Systems 769-776.

[5] Burger H C, Schuler C J and Stefan H 2012 Image denoising: Can plain neural networks compete with BM3D? Computer Vision and Pattern Recognition 2392-2399.

[6] Kai Z, Wangmeng Z, Yunjin C, Deyu M and Lei Z 2017 Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising IEEE Transactions on Image Processing.

[7] Stamatios L 2016 Non-local color image denoising with convolutional neural networks (arXiv preprint arXiv:1611.06757).

[8] Liao F, Liang M, Dong Y and Pang T 2018 Defense against adversarial attacks using high-level representation guided denoiser Computer Vision and Pattern Recognition.

[9] Goodfellow I, Shlens J and Szegedy C 2015 Explaining and harnessing adversarial examples (arXiv preprint arXiv:1412.6572).

[10] Kurakin A, Goodfellow I and Bengio S 2016 Adversarial examples in the physical world (arXiv preprint arXiv:1607.02533).

[11] Papernot N, McDaniel P, Jha S, et al. 2016 The limitations of deep learning in adversarial settings Proceedings of IEEE Symposium on Security and Privacy.

[12] Carlini N and Wagner D 2016 Towards evaluating the robustness of neural networks (arXiv preprint arXiv:1608.04644).

[13] Moosavi-Dezfooli S, Fawzi A and Frossard P 2016 Deepfool: A simple and accurate method to fool deep neural networks Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition pp 2574-2582.

[14] Tramer F, Kurakin A, Papernot N, Boneh D and McDaniel P 2017 Ensemble adversarial training: Attacks and defenses (arXiv preprint arXiv:1705.07204).

[15] Kurakin A, Goodfellow I and Bengio S 2017 Adversarial machine learning at scale (arXiv preprint arXiv:1611.01236).

[16] Ross A S and Doshi-Velez F 2017 Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients (arXiv preprint arXiv:1711.09404).

[17] Gu S and Rigazio L 2015 Towards deep neural network architectures robust to adversarial examples (arXiv preprint arXiv:1412.5068).

[18] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition Proceedings of the International Conference on Learning Representations.

[19] Fisher Y and Vladlen K 2015 Multi-scale context aggregation by dilated convolutions (arXiv preprint arXiv:1511.07122).

[20] Fisher Y, Vladlen K and Thomas F 2017 Dilated residual networks (arXiv preprint arXiv:1705.09914).

[21] Olaf R, Philipp F and T Brox 2015 U-net: Convolutional networks for biomedical image segmentation International Conference on Medical Image Computing and Computer-Assisted Intervention.

[22] Kaiming H, Xiangyu Z, Shaoqing R and Jian S 2015 Delving deep into rectifiers: Surpassing human-level performance on imagenet classification Proceedings of the IEEE International Conference on Computer Vision pp 1026-1034.

[23] Diederik K and Jimmy B 2014 Adam: A method for stochastic optimization (arXiv preprint arXiv:1412.6980).