Alter the Structure of Adversarial Perturbations with Model-Agnostic Defense

Daoyang Shan(ds5471), Nan Su(ns3783), Yueqiu Sun(ys3202)

1 Introduction

Deep neural networks are vulnerable to adversarial perturbations and are easily fooled by adversarial samples. In our project, we present an evaluation of different defense schemes that alter the structure of adversarial perturbation and defend against adversarial attacks including Fast Gradient Sign Method(FGS), Iterative Fast Gradient Sign Method(I- FGSM) and L-BFGS. The defense schemes we investigate is model-agnostic. Even if the model we use to make the prediction and the defense schemes we implement is known to the attacker, it is difficult for the attacker to circumvent the defense due to the randomness of the method. The evaluation we obtained can be used as reference for deploying different defense strategy under different circumstances.

We first present the methodology for the adversarial attack and the defense schemes we evaluate. Then we perform multiple experiments to evaluate the defense schemes against the attack using three metrics. We then conclude the evaluation and further discuss the implications of the results.

2 Previous Work

It has been shown that machine learning models are often vulnerable to adversarial manipulation of their input intended to cause incorrect classification (Dalvi et al., 2004)[1]. In particular, neural networks and many other categories of machine learning models are highly vulnerable to attacks based on small modifications of the input to the model at test time (Biggio et al., 2013[2]; Szegedy et al., 2014[3]; Goodfellow et al., 2014[4]; Papernot et al., 2016b[5]).

Goodfellow et al. (2014)[6] gave an linear explanation of adversarial examples. Let $\tilde{x} = x + \eta$ be an adversarial example where x is the input and η is the perturbation with $||\eta||_\infty < \epsilon$. Consider the dot product between a weight vector w and $\tilde{x}$. When η is parallel to w, the change in activation caused by perturbation by η can be significant when there are a large amount of features, as it grows linearly with the dimension of the weight.

Adversarial examples pose potential security threats for practical machine learning applications. In particular, Szegedy et al. (2014)[3] showed that an adversarial example that was designed to be misclassified by a model M1 is often also misclassified by a model M2. This adversarial example transferability property indicates that it is possible to generate adversarial examples and perform a misclassification attack on a machine learning system without access to the underlying model. Papernot et al. (2016a)[7] and Papernot et al. (2016b)[5] demonstrated such attacks in realistic scenarios.

3 Adversarial Attack

3.1 Methods Generating Adversarial Examples

In this section, we illustrate three representative approaches for generating adversarial examples.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

L-BFGS Attack

Szegedy et al.[3] first introduced adversarial examples for deep neural networks in 2014. They generated adversarial examples using a L-BFGS method to solve the general targeted problem:

$$\min_{x'} c||\eta|| + J_\theta(x', y_{target})$$

$$s.t.\ x' \in [0, 1].$$

The objective is to minimize the distance between the image and the adversarial as well as the cross-entropy between the predictions for the adversarial and the target class. To find the constant c, L-BFGS Attack calculated approximate values of adversarial examples by line-searching $c > 0$.

Fast Gradient Sign Method(FGSM)

L-BFGS Attack uses an very expensive linear search method to find the optimal value, which is very time-consuming and impractical. Goodfellow et al. [6] proposed a faster method for generating adversarial examples called Fast Gradient Sign Method. This method only performs one step gradient update along the direction of the sign of gradient at each pixel. The perturbation can be expressed as:

$$\eta = \epsilon sign(\Delta_x J_\theta(x, y_{target}))$$

where ϵ is the magnitude of the perturbation. Thus the generated adversarial example x' is calculated as $x' = x - \eta$.

Iterative Fast Gradient Sign Method(I-FGSM)

I-FGSM [4] extends Fast Gradient Sign method by running a finer optimization (smaller change) for multiple iterations. In each iteration, it clips pixel values to avoid large change on each pixel:

$$Clip_{x,\epsilon}\{x'\} = \min\{255, x + \epsilon, \max\{0, x - \alpha\}\},$$

where $Clip_{x,\epsilon}\{x'\}$ is the clipping value in each iteration limited by ϵ. The adversarial examples are generated in multiple iterations:

$$x_0 = x$$

$$x_{n+1} = Clip_{x,\epsilon}\{x_n - \alpha sign(\Delta_x J(x_n, y_{LL}))\}$$

3.2 Experimental Results

In our experiment, FGSM has lower success rate than the other algorithms because it only performs one step gradient update along the direction of the sign of gradient. However it can significantly decreases the confidence of our original prediction. For example, when we apply FGSM attack to the original image which is predicted as apple with confidence 0.95(AlexNet pretrained on imageNet), the confidence drops to 0.605 after this single-step attack.

For L-BFGS attack, we used ResNet50 model which was pretrained on imagenet. In comparison to FGSM and I-FGSM, L-BFGS attack is extremely slow when generating adversarial image. The result is good though, we can hardly observe any changes. (See figure 1 below)

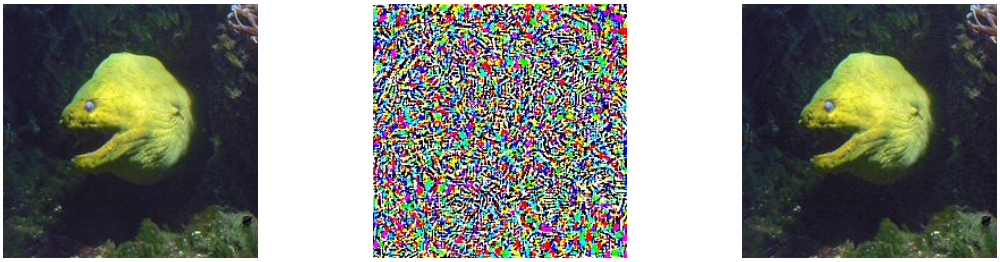
Original (predicted as cat)

Adversarial noise

Generated (predicted as rock python)

Figure 1: L-BFGS attack

I-FGSM illustrates the best performance among all methods. Given the same apple image, I-FGSM attack can misguide our pretrained AlexNet model to predict it as rock python in 5 iterations. When we change the pretrained model to resnet, it only needs 2 iterations to do the same thing. In addition, the generated image changes slightly. (See figure 2 below)



Original (predicted as eel) Adversarial noise Generated (predicted as bee)

Figure 2: I-FGSM attack

4 Adversarial Defense Methodology

As we've discussed before, adversarial attacks successfully fool model by slightly altering the structure of the original image. To defend a model against such attack, several attempts are tried by previous researchers. One widely held belief, as suggested by Guo et al.[8], is that some transformations on images may destroy the structures of noise, with which adversarial attacks fool models, without altering the image too much and therefore counter the effect of adversarial attack. Also, as Kolter et al. [9] presented in their work, we may propose to use more robust classifiers within our model to make sure that slight perturbations won't affect classification too much. Zheng et al.[10] propose stability training as a general technique that improves model output stability. In our project, we attempted several image transformation methods. Those transformations include:

- Gaussian Noise: a common statistical noise having a probability density function (PDF) equal to that of the Gaussian Distribution
- Speckle Noise: a granular noise that widely appear on images that originates from a phenomenon called Speckle, which is caused by the interference of waves that construct an image. Speckle noise usually appear like bright or dark dots on images.
- Gaussian Blur: a type of image-blurring filter that uses a Gaussian function (which also expresses the Gaussian Distribution in statistics) for calculating the transformation to apply to each pixel in the image.
- Random Crop: randomly crop a part of the original image and resize it back to the original size.

Figure 3: From left to right: original, Gaussian Noise, Gaussian Blur, Random Crop

5 Experimental results

We performed four experiments to evaluate the effectiveness of the defense schemes. We use a subset(1000 images) of the ImageNet image classification dataset. The model we use to produce adversarial samples is pretrained Alexnet. We apply the same input transformation on both the original image and the adversarial image, and then feed those two transformed images back to our model to produce the evalution metrics. We remove the samples that are classified wrong in the first place to achieve a fair evaluation for the defense schemes.

Goal and metrics

We aim to investigate how applying different combinations of input transformation with different parameter as defense schemes on input data influences the classification result. The result includes the confidence change for true label and target label(the label that the attack wants the classifier to predict) of both original and adversarial image. The evaluation of the defense schemes is based on three metrics:

1. M1: The gap between confidence of true class for untransformed sample and transformed sample. $(\frac{1}{n}\sum_{i=1}^{n} C_t^{y_i}(x_i) - C_u^{y_i}(x_i))$
2. M2: The increase of confidence of true class for adversarial samples. $(\frac{1}{n}\sum_{i=1}^{n} C_{at}^{y_i}(x_i) - C_{au}^{y_i}(x_i))$
3. M3 :The decrease of confidence of target class for adversarial samples. $(\frac{1}{n}\sum_{i=1}^{n} C_{at}^{y_t}(x_i) - C_{au}^{y_t}(x_i))$

where C_u^v denotes the confidence of class v for true&untransformed image , C_t^v denotes the confidence of class v for true&transformed image, C_{au}^v denotes the confidence of class v for adversarial&untransformed image,and C_{at}^v denotes the confidence of class v for adversarial&transformed image. y_t denotes the target class.

The first metric is not make sure the damage to the classifier's ability to classify normal image is controlled and minimized. The second and the third metric is used to assess the ability to defend against the attack, that is, to predict the true class regardless of the attack, and not realize attack's goal to predict the target class. It is worth noting that predicting the true label for adversarial samples don't mean that the other goal(M3) has been accomplished. Decreasing the confidence for target class is crucial in certain cases where, for example, you need to decide whether to fire the missile.

Results

First, we evaluate the defense schemes against I-FGSM. In our previous experiment, I-FGSM can effectively fool the classifier while remain undetected by the human eye.

Figure 4 shows the effectiveness of four proposed input transformation with different parameter to defend against I-FGSM. We can see that $|M1|$ decrease, $|M3|$ increase as sigma increase. And |M2| rises and falls as sigma increase.

Gaussian Blur is the most effective tool among four choices with regard to the metrics while Random Crop is not promising in defending adversarial attack. We try some combinations of input transformation as defense schemes to see if they are better than using only one transformation.

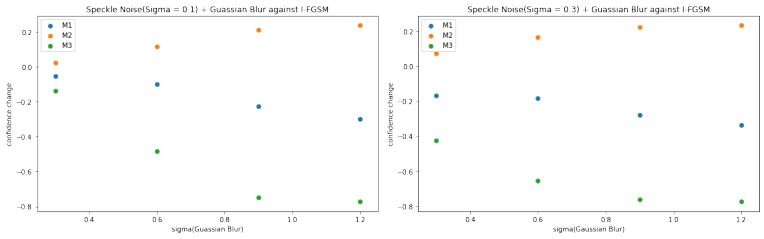
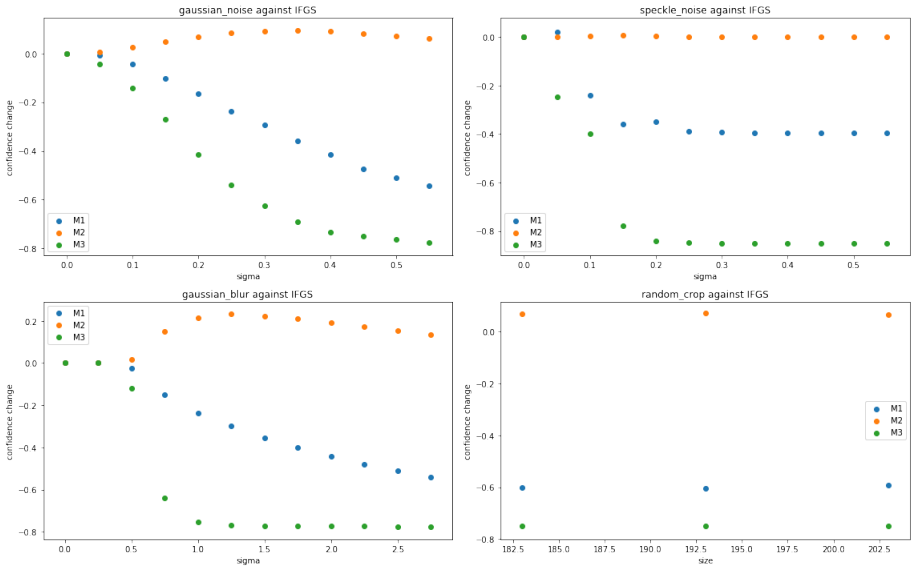


Figure 6: Evaluation for Speckle Noise + Gaussian Blur as defense schemes against I-FGSM



s

Figure 4: Evaluation for the four defense schemes against IBFGS

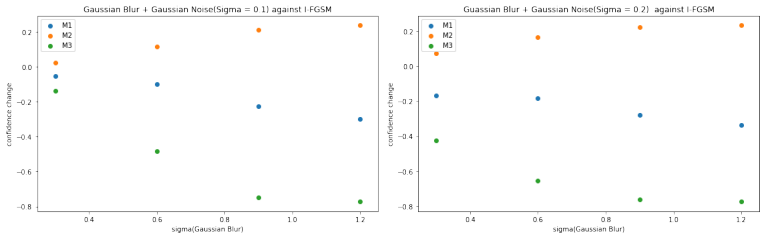


Figure 5: Evaluation for Gaussian Blur + Gaussian Noise as defense schemes against I-FGSM

Figure 5 and 6 shows that the combinations of input transformation as defense schemes perform better than the scheme that only use one input transformation with regard to the three metrics we use.

Figure 7 shows the evaluation of defense schemes against FGS. The relatively worse performance of the defense schemes can be explained by the ineffectiveness of FGS attack itself. In many cases, FGS don't have a large impact on the classification result. Therefore, it is reasonable that the input transformation don't change the confidence for true class and target class by a very large margin. Also, combination of input transformation is not needed because of the ineffectiveness of FGS attack.

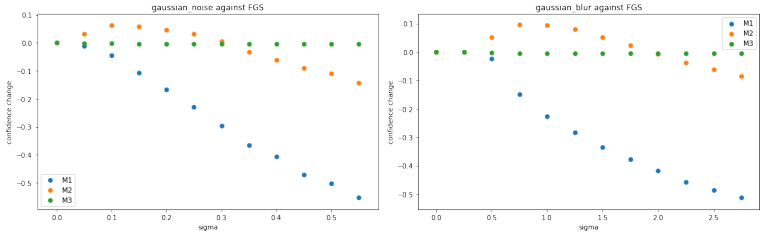


Figure 7: Evaluation for the defense schemes against FGS

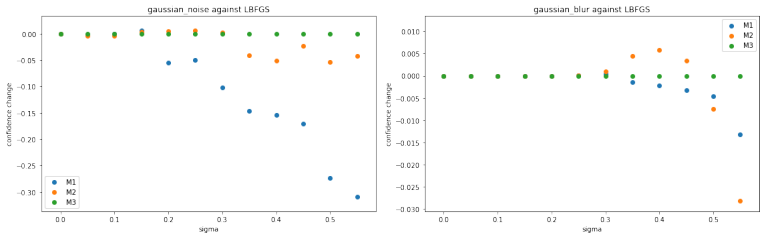


Figure 8: Evaluation for the defense schemes against L-BFGS

Figure 8 shows the evaluation of defense schemes against L-BFGS. We can see that the defense schemes don't really work on L-BFGS attack.

Table 1 collects the results for each metric of different defense schemes against I-FGSM attack. And table 2 collects the results for each metric of different defense schemes against FGS attack. The parameter of each defense scheme is chosen manually and reasonably.(most are inflection points of the metrics)

Table 1: Evaluation of the defense schemes against I-FGSM attack

	Gaussian Noise (sigma = 0.25)	Gaussian Blur (sigma = 1)	Speckle Noise (sigma = 0.1)	Random Crop (size = 203)	Gaussian Blur (sigma = 1.2) + Gaussian Noise (sigma = 0.1)	Speckle Noise (sigma = 0.1) + Gaussian Blur (sigma = 0.6)
M1	-0.23641397	-0.23594731	-0.242053	-0.59308326	-0.29859361	-0.11260712
M2	0.08504383	0.21500573	0.00398819	0.06539721	0.2378646	0.11711541
M3	-0.53920394	-0.7541101	-0.39943257	-0.7502144	-0.77034885	-0.48814908

Table 2: Evaluation of the defense schemes against FGS attack

	Gaussian Noise (sigma = 0.05)	Gaussian Blur (sigma = 0.5)
M1	-0.01029393	-0.02373083
M2	0.03224962	0.05157397
M2	-0.00103587	-0.00179084

6 Discussion

The paper evaluates the combination of four input transformation as defense strategy against adversarial attack. The results show that the defense scheme we propose is very effective in achieving M3(The decrease of confidence of target class for adversarial samples), but not so good at M2(the increase of confidence of true class for adversarial samples). The evaluation of the defense schemes above give us insights when implementing defense strategy against adversarial attack.

When countering I-GFSM attack, Gaussian Blur + Gaussian Noise is the most effective in situation where the decrease of confidence in target class(M3) is more valued. Speckle Noise + Gaussian Blur is most effective in situation where the tolerance for the gap between confidence of true class for untransformed sample and transformed sample(M1) is small.

When countering GFS attack, it is more reasonable to use the transformation with lower σ than the ones we use to counter I-GFSM attack. And the above defense schemes are not effective against L-BFGS attack.

The results of the evaluation also shows that the combination of input transformation works generally better than the scheme of using only one input transformation.

In future work, we intend to combine the input transformation strategy with other defense schemes including building more robust classifier and use stability training to improve model output stability.

References

[1] Nilesh N. Dalvi, Pedro M. Domingos, Mausam, Sumit K. Sanghai, and Deepak Verma. Adversarial classification. In *KDD*, 2004.

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Zelezný, editors, *ECML/PKDD (3)*, volume 8190 of *Lecture Notes in Computer Science*, pages 387–402. Springer, 2013.

[3] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[4] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[5] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016.

[6] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 12 2014.

[7] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016.

[8] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. *CoRR*, abs/1711.00117, 2017.

[9] J. Zico Kolter and Eric Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. *CoRR*, abs/1711.00851, 2017.

[10] Stephan Zheng, Yang Song, Thomas Leung, and Ian J. Goodfellow. Improving the robustness of deep neural networks via stability training. *CoRR*, abs/1604.04326, 2016.