Robust Recovery of Adversarial Samples

Abstract

Adversarial examples are semantically associated with one class, but modern Deep Learning architectures fail to see the semantics and associate them to another idea. As a result, these examples pose a profound risk to almost every Deep Learning architecture. Our proposed architecture is composed of a U-Net with the characteristics of Self Attention & Cross Attention. It can recover such examples effectively with more than 4x the magnitude of attacks that the state-of-the-art is capable of despite having lesser parameters than the VGG-13 model. Our study also encompasses the differences in the results between Noise and Image reconstruction of such examples.

1. Introduction

Adversarial examples are Machine Learning and Deep Learning model inputs that an attacker has purposefully designed to cause the model to make a mistake. They are similar to optical illusions for machines. A human is clearly able to understand the semantic meaning conveyed by an image-based adversarial example. Neural networks tend to create exponential classification boundaries, which are very sensitive to perturbation. These examples defeat the whole purpose of Deep Learning since even a tiny perturbation in an image can lead to misclassification rendering the entire infrastructure useless. A novel architecture is devised which is capable of recovering adversarial examples.

This misclassification of adversarial samples by Machine Learning and Deep Learning models in critical applications like self-driving cars where attackers may use stickers or paint to construct an adversarial stop sign, which the vehicle would misinterpret as a 'yield' or other signals, rendering the entire application at fault, it may even result in the loss of life. Attackers can use targeted attacks to make the model predict whatever they desire. Some of the most prevalent methods to generate adversarial examples are Fast Gradient Sign Method (FGSM) (1) and Random with FGSM (R-FGSM) (2). Adversarial examples are tough to fight against since creating a theoretical model of the adversarial example generation process is challenging.

This work aims at tackling the problem of recovering adversarial examples by generating negative noise and reconstructing the entire image. Previous works for recovering adversarial examples (4) only recover a limited range of attacks which excludes heavily attacked samples(max perturbation in the current state-of-the-art method is 16). Techniques which have been proven to be effective (4) are very computationally expensive. Our proposed architecture is capable of reversing a large range of adversarial attacks, including heavily attacked samples (Max perturbation of 128, 8 times more than ever achieved), which have lost their semantic meaning while being less computationally demanding than a VGG-13 model in terms of the number of parameters. Our proposed model can be easily added to existing application pipeline without changing the existing models.

Our contributions through this paper include:

- We propose a novel GANs architecture and training methodology to recover FGSM and R-FGSM adversarial samples irrespective of the magnitude of the attack, performing better than previous state-of-the-art on the CIFAR10 dataset (5).
- A novel training procedure with adversarial training followed by post-training, which uses hybrid losses.
- Our proposed generator has fewer parameters than a most basic classification model like VGG-13 (6).

2. Background

Adversarial samples are generally produced by adding some calculated noise which is unrecognisable by humans but is enough to fool a classifier like a deep convolutional network. Deep Neural Networks have an exponential number of decision boundaries, which are very close to each other, which makes even a slight perturbation in the right direction result in mis-classification. There are various types of adversarial attacks, which are: white-box attack where it is assumed that the attacker has complete information and access to the model, grey-box attack where attacker knows the model parameters of the target model but does not know that the target model has an active defensive mechanism,

. **AUTHORERR: Missing \icmlcorrespondingauthor.**

Preliminary work. Under review by the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning. Do not distribute.

and black-box attack where the attacker only knows the input and output of the model and does not have any other information. White Box attacks are the most devastating as the exact perturbation required can be found by the attacker.

2.1. Fast Gradient Sign Method

FGSM [6] attack is a type of white-box attack where the attack uses the gradient of the loss with respect to the input data, then adjusts the input data to maximize the loss.

$$adv_x = x + \epsilon * sign(\nabla_x J(\theta, x, y)) \quad (1)$$

In the above formula, adv_x represents adversarial image, 'x' is input image (Original), 'y' input label (Original), ϵ is the multiplier, θ is the parameter of the model and 'J' is the loss function. ∇_xJ(θ,x,y) is the gradient of loss with respect to 'x'.ϵ is the metric of the magnitude of the adversarial attack.

2.2. Random - Fast Gradient Sign Method

This variant of the Fast Gradient Sign Method attack is a more efficient attack compared to iterative method since this attack requires a single gradient computation. This attack applies a small random perturbation to before relinearizing the model's loss to escape the non-smooth vicinity of the data point. R+FGSM outperforms FGSM for the same perturbation norm.

$$x^{adv} = x' + (\epsilon - \alpha) * sign(\nabla_{x'} J(x', y_t rue)) \quad (2)$$

where

$$x' = x + \alpha * sign(N(0^d, I^d)) \quad (3)$$

3. Motivation

The current state-of-the-art(4) model uses U-Net architecture and has an extremely large number of parameters. It was trained to reverse adversarial ImageNet (7) images created using three different classification models and two attacks, FGSM and IFGSM, with an epsilon range of 1-16. Training to reverse ImageNet 300x300 size images requires immense computational resources. We did not have the computational budget to do so. Hence we focused on making the GAN (3) model much more efficient in terms of the number of parameters and reversing a much larger epsilon distribution on the CIFAR-10 dataset. The proposed GANs model has fewer parameters than a classification model and is trained to reverse CIFAR-10 adversarial images with an epsilon range between 1-128, attacked from 3 different models with FGSM and RFGSM attacks. Our method of generating adversarial samples equates to a total of 38.4 Million Training Images which is not practical to store, hence these training examples are constructed on the fly.

4. Dataset

Adversarial images are needed for both training and testing of the proposed method. To prepare the training dataset, We used 50k images from the CIFAR-10 training set (5000 images per class). Multiple adversarial attacking methods were used to distort these images and form a training set of adversarial images. Different attacking methods, including FGSM and RFGSM, are applied to the following models: Pre-trained Inception-V3, Resnet-18, VGG-13 individually. For each training sample, the perturbation level E is sampled uniformly from integers in [1, E_max], here E_max is the maximum possible value of epsilon for that experiment.

Dataset of adversarial samples are produced at the time of GANs training. Every step of the training, a batch of 96 images is split into 32, 32, and 32 for three attacking models. These 32 images are divided into 16-16 for two attacks FGSM and R-FGSM, for each model. Furthermore, every image in these 16 images is attacked with a different epsilon (range 1-16). Finally, making every image in a batch be attacked by a different model or different method or different epsilon, making the training super stable for the task.

Producing adversarial samples while training step gives the benefit of not storing a large amount of adversarial data and, second, making each training image in a batch more random. The permutation of different adversarial examples that could be generated is as follows:

50,000 Images * 3 Models * 2 Attacks * E_max

(This permutation represents the number of different training inputs that can be produced)

To prepare the validation set, We used 10k images from the CIFAR-10 validation set (1000 images per class). Then applied the same method as described above. For each testing sample, the perturbation level E is sampled uniformly from integers in [1, E_max]

For final testing, E_max test sets are constructed for every epsilon e in the range between 1 to E_max. They are obtained from the same clean 10K images from the CIFAR-10 validation set but using only the FGSM attack.

5. Architecture and Training Procedure

We used Attention U-Net as a base generator architecture because it automatically learns to focus on image structures of varying shapes and sizes and can be easily integrated into standard CNN architectures such as the U-Net model with minimal computational overhead while increasing the model prediction accuracy.

Added PixelShuffle rather than ConvTranspose, as Pix-

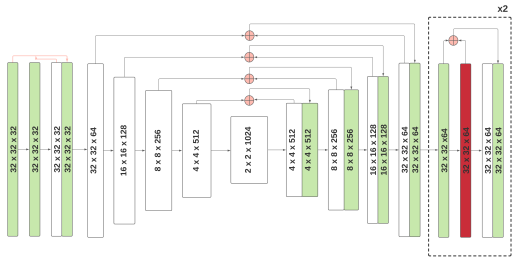

Figure 1. Generator Architecture (Red represents Self Attention Layer & Pink represents Cross Attention)

elShuffle is an operation used in super-resolution models to implement efficient sub-pixel convolution with a stride of 1/r. Specifically, it rearranges elements in a tensor shape (*, C x r^2, H, W) to a tensor shape (*, C, H x r, W x r). Outputs of the PixelShuffle layer are normalized using Batch Normalization and then passed through PReLU activation. Parametric Rectified Linear Unit (PReLU) is proposed to generalize the traditional rectified unit (ReLU). LeakyReLU was introduced to improve upon ReLU. It multiplies the negative input by a small value (like 0.01) and keeps the positive input the same. PReLU allows the network to learn that small value during training so that our activation function can adapt to the other parameters (like weights and biases). The network can learn the slope parameter using backpropagation at a negligible increase in the cost of training.

Generally, models are made deep with more hidden layers to increase their prediction accuracy, essentially giving more parameters for the model to learn. Usually, 3-4 layers of convolutional layers are used at every stage of U-Net architecture. We instead used a single convolutional layer with a kernel size of 3 but used self-attention layers at the decoding layers. Wang et al. proposed self-attention (8). The output of the Attention U-Net is passed through a self-attention layer followed by residual cross attention connection.

5.1. Training Procedure

The model is trained using adversarial procedure with three different loss functions: L1 loss (pixel-level loss), Adversarial loss from discriminator (MSELoss) and Classification loss from pre-trained CIFAR-10 Classifier (CrossEntropyLoss).
Even after Adversarial Training there are enough perturbations present in the generated image when compared to the original image, these perturbations may make the model yield the wrong prediction as this minimal perturbation is progressively amplified by every passing layer of the network known as error amplification effect. The top layers of deep neural networks cause misclassification due to small perturbation, training the model to minimize the difference of these top-level features can result in better predictions. This perturbation can be removed by post-training methodology, where the generator trained by adversarial procedure is guided by a loss function known as perceptual loss or feature matching loss.
Our experiments resulted in the conclusion that using two stages of training (Adversarial Training and Post Training) gives better results as compared to including feature loss at stage one (adversarial training, which made the training unstable), GANs training was much more stable and converged faster.

6. Image vs Noise Construction

Table 1. Performance comparison for 3 methods

Epsilon	Noise Construction		Image Construction		Image Construction-2	
	Inception	VGG	Inception	VGG	Inception	VGG
0	92.71%	92.26%	92.21%	92.25%	92.52%	92.92%
4	91.27%	88.88%	88.87%	88.72%	90.19%	89.69%
8	93.47%	88.5%	92.29%	89.04%	92.45%	89.66%
16	93.91%	88.3%	93.86%	89.02%	93.58%	89.69%
32	8.02%	26.98%	57.92%	72.57%	56.43%	74.18%

Our work focuses on the differences on the results by the procedure Image and Noise reconstruction using GANs. GANs can be trained to predict to reconstruct the original image with the input of an adversarial attacked image or they can be trained to anticipate the negative noise, which when added to the adversarial image, will result in an original image. Previous research works agree that noise construction produces a better result when compared to image reconstruction, including the authors of the current state-of-the-art model for ImageNet image recovery. On the contrary, our results show that image reconstruction can still be the preferred choice depending upon the purpose of model implementation. Understanding the trade-off between these approaches is now more important than before because this is the fundamental choice while dealing with Diffusion Models (**?**), which has shown to work better than GANs. We have demonstrated the advantages and disadvantages of both approaches in Table 1

The result of column Image Construction-2 represents the scenario of image construction with adversarial loss in post-training. The discriminator is not involved in post-training in the other two methods, hence only the feature loss as the training criterion.

From the results of Table1, Noise construction slightly gives higher accuracy but lacks robustness when dealing with an adversarial attack of higher E_max, which lies outside the training distribution. We hypothesize that constructing Noise works better than constructing images from perturbed images because of the difference in the scale of predicted pixel values. The range of the pixel values the generator has to predict while creating noise is much smaller [-E_max, E_max]. For image construction, the generator has to en-

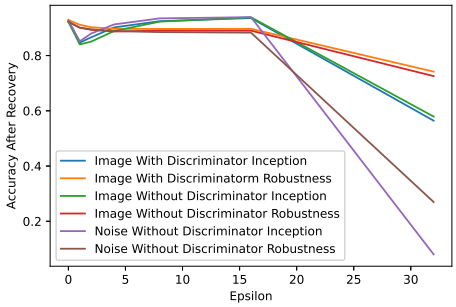

Figure 2. Performance Comparison of 3 methods

code all the relevant information and then reconstruct the image from the encoding. In this scenario, the generator has to predict the values for each pixel in the range of 0 - 255 ([-1,1] since the use of Tanh), which is a larger scale than that of constructing noise, as perturbation added is only between E[-16,16], the negative noise pixel value will also lie between this range.

This advantage of producing noise does not work well when trained to reverse data with higher perturbation. For example, epsilon = 128, but at the same time, these perturbed examples are not classified as adversarial samples but classified as rubbish class (1). In any case, we hypothesize that reversing images with high perturbation is much easier because training loss is much higher as the difference between perturbed data and original data is much higher, making training converge much faster.

One of the benefits of constructing the image is that the network becomes more robust for various epsilon outside the training distribution. After observing the recovery accuracy on epsilon 32, which is way out of the epsilon range used for training, the image construction method yields upto 75% reclassification accuracy. In contrast, the noise construction method yields upto 30%. This difference is due to how image reconstruction differs from noise construction. In the case of image reconstruction, the encoder can still extract partial semantic information about the image, which is consequently used to reconstruct it. While noise construction has not been trained to predict the scale of pixel values that will reverse the attack, which in this case is in between [-32,32], and the network was only trained to predict a range between [-16, 16].

7. Results

Our generator, after being trained for the range of 128 epsilons, reconstructed the attacked images. These images, when tested on Inception and for robustness tested on VGG-13, gave staggering results 2. Both the models were able to successfully classify the reconstructed images where in some cases, the model's results (25 Epsilon - 98.5%) were better than what the models performed on original images (93.1%). This proves that the generator is capable of reversing the images and enhancing the semantic meaning of the images, resulting in a performance boost in models. The results show us that the architecture can generalize and is robust.

Table 2. White Box Attacks

Epsilon	Inception-V3	VGG-13	Exposed VGG-13	Exposed Inception
5	90.32%	86.46%	43.18%	11.24%
10	96.32%	89.64%	38.89%	5.96%
20	98.13%	90.86%	24.86%	7.52%
25	98.5%	90.44%	18.34%	8.21%
35	98.4%	87.36%	12.65%	8.83%
50	96.83%	77.37%	10.69%	9.18%
65	91.85%	67.19%	10.40%	9.26%

Black-Box attacks were performed with the adversaries as DenseNet-121 and VGG-11, after the recovery they were reclassified by Inception-v3. In the table , Exposed 1 and Recovered 1 represent the attacking model as DenseNet-121 and similarly for VGG-11. All the accuracy represents the classification accuracies of Inception-v3.

Table 3. Black Box Attacks

Epsilon	Exposed 1	Recovered 1	Exposed 2	Recovered 2
5	70.22%	79.82%	64.84%	77.78%
10	53.88%	79.26%	46.85%	76.35%
20	27.51%	79.01%	24.96%	76.63%
25	20.61%	77.40%	19.39%	75.46%
35	14.76%	68.86%	14.69%	66.16%
50	11.03%	59.18%	12.04%	57.28%
65	9.67%	55.66%	10.78%	54.32%

8. Conclusion

A novel architecture and training procedure is devised where the adversarial images are recovered with enhancing the semantics of the original image making the models classify better than classification capacity for original images. This architecture is capable of recovering images on a large magnitude of attacks. We had limited computation available to us, so the architecture is yet to be validated on ImageNet.

References

[1] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[2] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., & McDaniel, P. (2017). Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204.

[3] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

[4] Li, M., Wang, J., Wu, Y., & Zhou, S. (2020). Detecting and Recovering Adversarial Examples: An Input Sensitivity Guided Method. arXiv preprint arXiv:2002.12527.

[5] Krizhevsky, A., Nair, V., & Hinton, G. (2014). The cifar-10 dataset. online: http://www. cs. toronto. edu/kriz/cifar. html, 55, 5.

[6] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[7] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[8] Yang, B., Wang, L., Wong, D., Chao, L. S., Tu, Z. (2019). Convolutional self-attention networks. arXiv preprint arXiv:1904.03107.