Homework 2
Security and Privacy of Machine Learning, Fall 2020

Chi-Pin Huang
Department of Computer Science and Information Engineering
National Taiwan University
b07501122@csie.ntu.edu.tw

Abstract

Our task is to build a robust model against black box adversarial attack. That is, the attacker would not know the detail of our defense method, model parameters and even model architecture we use. The only thing attacker know is the training set, which is CIFAR10. In this works, firstly we conduct the most used adversarial defense method: adversarial training(AT) to obtain the baseline of this task. We try then another AT technique and make comparison with the standard one. Secondly, for better resistance to the adversarial examples, we investigate pre-processing based methods, such as JPEG Compression and Defens-GAN, and finally evaluate the performance of each combination on three different testing set.

1 Introduction

The task is to classify CIFAR10 dataset against adversarial examples up to $\epsilon = 8$ in the Linf norm. And the adversarial attack is in the black-box setting.

Evaluation

Experiments below we use MI-FGSM with Linf constraint $\epsilon = 8/255$, iteration step $N = 150$, step size $\alpha = 8/255/100$, momentum $\mu = 1.0$ on three different targets to generate adversarial examples: (1) relative-weak model ensemble. (2) adversarial trained model. Then, we evaluate our defensive method(9 combinations in total) on these adversarial examples and the benign images, and record the table in Conclusion4.

2 Adversarial Training

Adversarial training is the most used defense approach. Madry et al. introduce the idea that if a model can "minimize" the loss under the most "powerful"(maximize) adversarial examples (up to some L2/Linf constraints), then this model is robust. They study this min-max objective function and use the PGD attack as the maximize step. Then we can just minimize the loss function to fulfill the objective.

2.1 Standard Adversarial Training (AT)

Given a dataset $S = \{(x_i, y_i\}_{i=1}^{n}, where\ x_i \in \mathcal{X}\ and\ y_i \in \mathcal{Y} = \{0, 1, ..., C-1\}$, the objective function of standard AT is a min-max optimization problem as

$$\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \Big\{ \max_{\delta, \|\delta\|_\infty \le \epsilon} \ell(h(x_i + \delta), y_i) \Big\} \tag{1}$$

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Algorithm 1: Adversarial training AT

INPUT: training set $\mathcal{D}$, epochs N, model parameters θ, learning rate η and PGD parameters(step K, perturbation bound ϵ, step size α)
OUTPUT: θ

```
for $i \leftarrow 1$ to $N$ do
    for minbatch $x_b, y_b \in \mathcal{D}$ do
        $x_b^{adv} \leftarrow PGD(x_b, y_b; K, \epsilon, \alpha)$;
        $\theta \leftarrow \theta - \eta \nabla_\theta \ell(x_b^{adv}, y_b; \theta)$
    end
end
```

2.2 Friendly Adversarial Training (FAT)

Zhang et al. come up with a different concept named Friendly Adversarial Training(FAT) that is a min-min optimization problem, instead of min-max in the standard AT. They explain that the maximize step in standard AT is too aggressive. The adversarial examples generated by original PGD can hurt the model decision boundary and cause a cross-over mixture problem. Hence, they study the min-min objective function:

$$\begin{aligned} \delta^* &= \arg\min_{\|\delta\| \le \epsilon} \ell(h(x+\delta), y) \\ &\qquad\qquad s.t.\ \ell(h(x+\delta), y) - \min_c \ell(h(x+\delta), y) \ge \rho \\ x^{adv} &= x + \delta^* \end{aligned} \tag{2}$$

$\rho > 0$ is the margin such that the adversarial examples would be misclassified with a certain amount of confidence.

To achieve so, the author introduce PGD-K-τ. That is, stop the PGD iteration after the input image has been misclassified for τ iteration(s). Through this approach, the generated adversarial examples can be more "friendly" to the classifier and the cross-over mixture problem is handled.

Algorithm 2: PGD-K-τ

INPUT: data x, label y, model h, loss function ℓ, maximum PGD step K, perturbation bound ϵ, step size α
OUTPUT: x^{adv}

```
x$^{adv} \leftarrow x$
while $K > 0$ do
    if $\arg\max_i h(x^{adv}) \neq y$ and $\tau \neq 0$ then
        break
    else if $\arg\max_i h(x^{adv}) \neq y$ then
        $\tau \leftarrow \tau - 1$
    end if
    x$^{adv} \leftarrow$ PGD_one_step($x^{adv}, y, h, \ell, \epsilon, \alpha$)
    K $\leftarrow K - 1$
end
```

2.3 Experiments

We do experiments on AT and FAT under the same setting. We choose SGD optimizer with $\eta = 0.1$(decay by 0.1 every 30 epochs), momentum $\mu = 0.9$, 90 epochs, $\epsilon = 8/255$, PGD(PGD-K-τ) iteration=10, $\alpha = 8/255/5$ and $\tau = 1$ for FAT. We perform adversarial training on the pytorchcv pretrained seresnet_56_cifar10 and evaluate on ordinary adversarial examples.

In Figure1, the FAT's training accuracy is significantly lower than AT's, which is caused by the "friendly" property. Futhermore, we can observe that FAT converges faster and better then AT in testing accuracy.

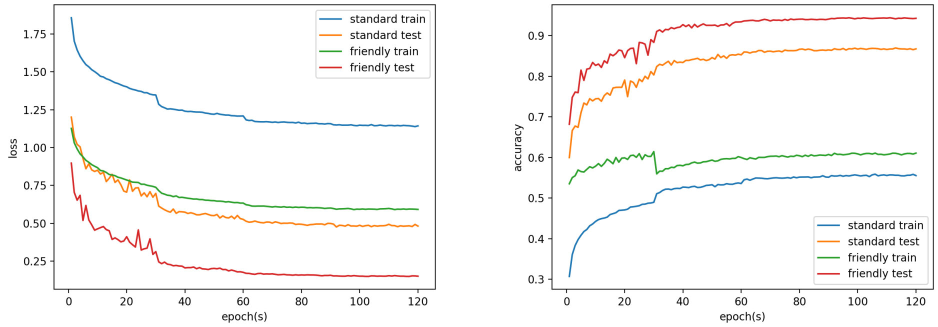

Figure 1: The training curve of AT and FAT(left: loss, right: accuracy), testing on ensemble targets adversarial examples

3 Input Transformation

Input transformation-based defense method is transforming the adversarial examples into benign data, that is like purifying the adversarial perturbation on the images. We implement two approachs, JPEG Compression and Defense-GAN.

3.1 JPEG Compression

JPEG compression is a common method of loosy compression images and can also be used in adversarial defense. JPEG Compression has a hyperparametr: quality. The lower quality means higher compress is performed. With lower quality the adversarial perturbation can be eliminated but the model classify the compressed image hardly. Hence, we would test on different quality later.

3.2 Defense-GAN

Generative Adversarial Network(GAN) is a class of machine learning frameworks designed by Ian Goodfellow. The architecture contains two neural networks, generator and discriminator. Generator learns the real data distribution and generate fake data, which would be classify by discriminator. During GAN training, these two networks compete with each other and become stronger together.

The key point is Generator learn the real data distribution. Defense-GAN takes advantage of this property to reconstruct the input adversarial examples into a benign one.

When an image is input, Defense-GAN algorithm starts from K random noise z_i, $i = 1$ to K, and feed them into the Generator to generate K fake images. Next, compute the reconstruction loss between fake images and the input itself, and perform gradient descent on the loss to update z_i. Repeat the procedure above, then the fake images would more and more like the input image. Finally, just pick the fake image with smallest reconstruction error, and feed it into the classifier.

4 Conclusion

We experiment on three models (normal train / AT / FAT) with two different input transformation method(JPEG Compression with quality(0 / 33 / 66) and Defense-GAN with initial num $K = 10$, and iterates 500 times to reconstruct the images). Then, we show the result on Table1.

We have some findings according to our experiment result:

1. FAT do better than AT. However, they all bring the accuracy down on benign images.
2. JPEG Compression and Defense work well on the normal train model, but often drag the adversarial train model down.
3. The attack on adversarial train model transfers hard to normal train model.
4. Defense-GAN is really time-consuming(hundred times slower than JPEG Compression).

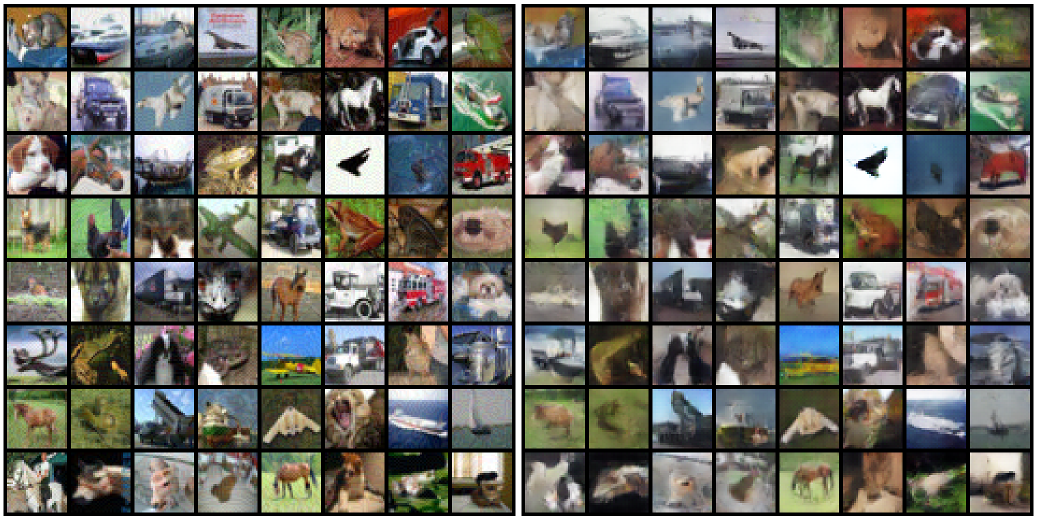

Figure 2: left: input images, right: reconstruct images.
Some images don't reconstruct well and cause the model accuracy decreases instead of increases.

Defense Methods **All models are Seresnet56**	Benign Images	Attack on weak model ensemble	Attack on adversarial trained model
Normal train	96.1%	1.7%	80.8%
Normal train + JPEG	21.3/71.2/84.8%	19.8/51.8/47.7%	19.5/49.8/62.1%
Normal train + DefGAN	53.9%	49.5%	41.5%
Standard Adv. train	81.7%	78.2%	61.3%
Standard Adv. train + JPEG	56.8/81.4/80.1%	53.4/73.8/74.5%	45.4/59.7/60.3%
Standard Adv. train + DefGAN	67.8%	61.4%	50.2%
Friendly Adv. train	86.2%	84.1%	64.7%
Friendly Adv. train + JPEG	54.2/85.1/86.2%	49.3/78.9/79.7%	42.9/62.1/63.3%
Friendly Adv. train + DefGAN	70.2%	62.1%	51.0%

Table 1: The performance of each combination of defense methods.
*JPEG Compression with different quality(0/33/66)

Finally, we choose the FAT model without input transformation.

References

Reference paper in this work.

[1] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu Boosting (2017) Towards Deep Learning Models Resistant to Adversarial Attacks `https://arxiv.org/abs/1706.060832`

[2] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, Mohan Kankanhalli (2020) Attacks Which Do Not Kill Training Make Adversarial Learning Stronger `https://arxiv.org/abs/2002.11242`

[3] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, Duen Horng Chau(2018) Shield: Fast, Practical Defense and Vaccination for Deep Learning using JPEG Compression `https://arxiv.org/abs/1802.06816`