Security and Privacy of Machine Learning - Homework 2

Wu-Jun Pei
National Taiwan University
b06902029@ntu.edu.tw

Abstract

Despite the mightiness of deep nerual networks, several studies have shown that they are vulnerable to adversarial examples,. In this homework, I built a

1 Introduction

In this homework, we're going to build a black-box defense on CIFAR-10 [1] dataset.

1.1 Threat Model

As suggested in class [2], we should state the considered threat model precisely. The threat model I consider is listed as below:

- The adversary has limited knowledge of the model, he only knows the model architecture, but is totally ignorant of the training process and the model weights.
- The adversary can perturb each pixel up to 8 (in 0-255 scale), which is given.

2 Methods

2.1 Baseline Method

2.2 Adversarial Training

Adversarial training [3]

2.2.1 Adversarial Training

Adversarial training may take a long time, and it's mostly contributed by *generating adversarial examples*.

2.2.2 Adversarial Training on Ensemble Models

Same as the previous subsection, adversarial training take a long time during the *generating adversarial examples* phase even in a single model setting. If we want to train an emsemble model of several models, the time would grow significantly. For example, it only takes **TBD**. To overcome the computational cost, I redesign the adversarial training process as:

2.3 Preprocessing-based Defenses

In my previous homework, I've already shown that some preprocessing-based defenses, such as vanilla JPEG Compression, are effective enough to eliminate the influence of adversarial perturbations. In this homework, I'm going to explore defenses that are more effective.

2.3.1 Baseline

Inspired by the preprocessing method TA used in the evaluation of previous homework, I setup the baseline method as:

- ColorJitter (brightness = 0.4, contrast = 0.4, saturation = 0.4, hue = 0.25)
- CenterCrop (size = 24)
- Pad (size = 4)

2.3.2 Vanilla JPEG Compression

We apply JPEG Compression on the entire image before feeding it to our model in order to reduce the adversarial noise.

2.3.3 SHIELD

Similar to "Vanilla JPEG Compression", we divide the image into several equal sized subimages. For each subimage, we apply different JPEG quality at random on it. And finally, we concatenate the compressed subimages back.

2.4 Evalution

To evaluate my work fairly, I unify the method to evaluate each model.

The adversarial examples are generated in the following way:

- PGD attack, constrained to l_2 norm $\epsilon = 8/256$
- The number of iterations are 8, 16, 32, 64 and 128.
- The proxy models are `nin`, `resnet20`, `sepreresnet56`, `densenet40-k12-bc`, and `diaresnet110`.

3 Experiments and Findings

3.1 Adversarial Training

3.1.1 Adversarial Training Epochs

Experiment Settings In this experiment, we want to examine if increasing adversarial training epochs improves the general adversarial accuracy. The model we used is `resnet20` since it's more lightweight. We regenerate new adversarial examples every epoch, and each adversarial example is generated with 8 iterations of PGD attack. We evaluate the model with adversarial sets with different attack strengthes.

Findings The experiment results are shown in figure 1. We can see that the model improves a lot in the first two epochs, and have no significant improvement on evaluation set afterwards. Also, we can see that training on adversarial examples generated with 8 iterations have a fair performance on all other evaluation sets generated with larger iterations. These findings suggest us that we can adversarially train our model with weaker adversarial set and with less training epochs, saving computational costs. Although it seems robust to the evaluation adversarial set, it's still vulnerable if the attacker knows the model weights. The accuracy on newly generated adversarial examples are about 15% on the last epoch.

3.2 Preprocessing-based Defenses

Experiment Settings To test the effectivness of those defenses, I used two pretrained `pytorchcv` models (without adversarial training), `resnet20` and `resnet1001`. The adversarial examples are generated as described in section 2.4 with 16 iterations of PGD.

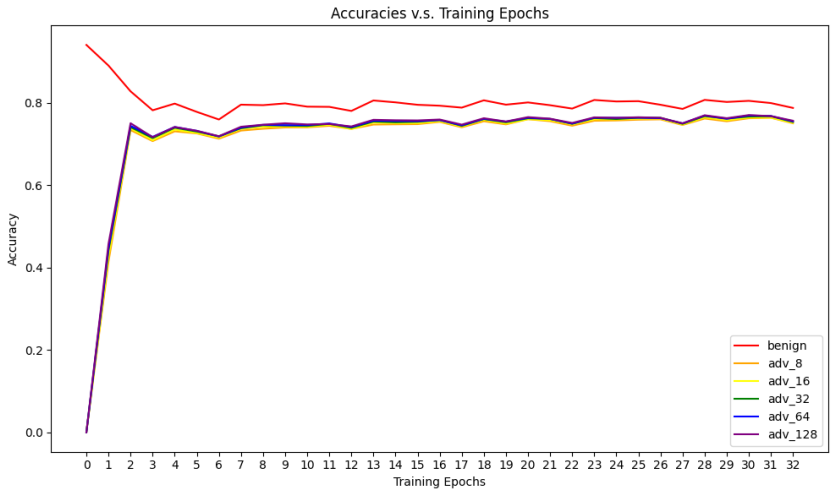

Figure 1: `resnet20`'s accuracies on different training epoch, evaluated on adversarial sets generated with different number of iterations

Table 1: Evaluation of different preprocessing based defense. The plus sign (+) indicates the benign set while the minus sign (-) indicates the adversarial set.

Defense Method	resnet20				resnet1001			
	Train +	Train -	Val. +	Val. -	Train +	Train -	Val. +	Val. -
None	0.9865	0.0000	0.9403	0.0001	1.0000	0.3962	0.9672	0.3420
Baseline	0.8622	0.1957	0.8181	0.1950	0.9403	0.4847	0.8792	0.4462
JPEG (quality = 60)	0.8211	0.6389	0.7924	0.6162	0.9257	0.8239	0.8671	0.7563
SHIELD (block size = 4)	0.7917	0.5902	0.7634	0.5688	0.8939	0.7719	0.8322	0.7068

Findings The experiment results are shown in table 3.2. We can see similar performance on the two models. The baseline method has little effect on adversarial examples, while both vanilla JPEG compression and SHIELD are more effective. To take a closer glimpse, we can see that vanilla JPEG compression method has better on both benign set and adversarial set. I guess that it may result from that the image size in CIFAR-10 is already small (32x32), and the smaller splitted images will not benefit from the JPEG compression.

4 Final Decision

The final model I use is

- An ensemble model consisting of ....
- Adversarial training for epochs, adversarial examples are regenerated for x iterations every y epochs.
- Equipped with a single preprocessing-based defense, JPEG-Compression with quality 60.

5 Conclusion

References

[1] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[2] S.-T. Chen, "Security and privacy of machine learning class on sep. 25th." "`https://www.csie.ntu.edu.tw/~stchen/teaching/spml20fall/`", 2020.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.