Dauphine | PSL

Robust Neural Networks

Professors:

Alexandre Verine
Benjamin Negrevergne

Group Name:

NeuralGuardians

Students:

Artur Dandolini Pescador
Caio Azevedo
Rafael Benatti

November
2023

Contents

1 Introduction

The main objective of this work is to better understand the robustness of a neural network. Firstly, we are going to implement different attacks methods to observe how the model is sensitive to these attacks. After, we are going to use the adversarial training technique to better improve the accuracy when determined attack is applied. Finally, some more generals defense methods will be implemented to protect the model from adversarial attacks.

2 Dataset

The dataset that was used for this project was the CIFAR-10 dataset. It consists of 6000 colour images divided into 10 classes. The size of each image of this dataset is 32 x 32 pixels.

3 Adversarial Attacks

Adversarial Attacks are techniques that attempt to cause the model to make mistakes. Usually, the attacks are carried out by manipulating the input data at test time.

3.1 FGSM Attack

The Fast Gradient Sign Method (FGSM) is an attack that create adversarial examples by using the gradients of neural networks to create some perturbations. The FGSM works by using the gradient of the loss with respect to the input data, then adjusting the input data to maximize the loss. It can be expressed as:

$$X_{adv} = X + \epsilon \cdot \text{sign}(\nabla_X L(\theta, X, y_{true}))$$

Where X is the original input, ϵ is a small constant (e.g. 0.001), $\nabla_X L(\theta, X, y_{true})$ is the gradient of the loss with respect to the input, θ represents the model parameters, and y_{true} is the true label for the input X.

3.1.1 FGSM results

Applying the FGSM attack, the accuracy dropped. The results obtained are descibred below.

- Normal training accuracy: 55%
- FGSM attack accuracy: 10%

It is possible to observer a significant drop in accuracy that indicates that the model is highly susceptible to adversarial attacks, which show the importance of implementing some kind of defensive strategy.

3.2 PGD Attack

Projected Gradient Descent (PGD) is an iterative method for creating adversarial examples. It is considered to be a more powerful attack compared to FGSM because it applies the perturbation multiple times.

3.2.1 PGD results

When the PGD attack is applied to the model, the accuracy dropped even more as it is possible to see below:

- Normal training accuracy: 55%
- PGD attack accuracy after 10 iterations: 9.2%
- PGD attack accuracy after 40 iterations: 6.3%

3.3 Carlini & Wagner Attack

The Carlini & Wagner Attack works similarly to the others explained adversarial attack methods. Here we try to minimize an objective function in the form:

$$\text{minimize} \quad \|\delta\|_p + c \cdot f(x + \delta)$$

The modified image will be $x + \delta$. We need to ensure that the modified result consists in a valid image. The original paper suggests 3 approaches to address this problem: Projected gradient descent, clipped gradient descent and change of variables. The last one is expected to outperform the other 2 approaches since it may work for complicated gradient steps, such as steps with momentum (Projected gradient descent may not work in this case) and because clipped gradient descent may get stuck in a flat spot where it has increased some component x_i to be substantially larger than the maximum allowed. In our implementation we use a change of variables as the one suggested in the paper where:

$$\delta_i = \frac{1}{2}\left(\tanh(w_i) + 1\right) - x_i$$

Finally, our objective function becomes:

$$\text{minimize} \quad \left\|\frac{1}{2}(\tanh(w) + 1) - x\right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(w) + 1)\right)$$

Here the function f is a function that penalizes that penalizes incorrect classification of images i.e. its value increases for misclassifications. As we are minimizing it, we will encourage the generation of adversarial examples. c is a weighting parameter for this term.

3.3.1 Carlini & Wagner results

The CW attack is able to drop the accuracy of the trained model, although how much the accuracy is affected depends on the choice of hyperparameters. The increase in c encourages more changes in the image. Resulting potentially in perceptible changes in the image.

Figure 1: CW attack with c=0.1

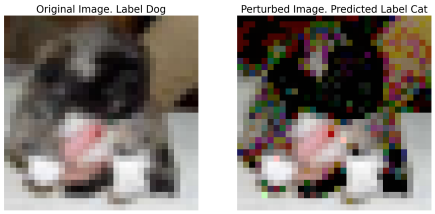

Figure 2: CW attack with c=100

This kind of change is also notable for changes in the number of steps and learning rates. Finally, we kept hyperparameters that yielded an accuracy of 32

4 Adversarial Training

Adversarial training is a defense mechanism against adversarial attacks such as the explained in the section above. It mainly involves training a model on a mixture of adversarial and clean examples. The objective of doing that is to build a more robust model by learning to classify correctly even when perturbations are present.

The table below compares the model's accuracy under normal conditions, under attack without defense, and under attack with adversarial training as a defense mechanism.

Adversarial Training	**Training Accuracy (with attack)**	**Adversarial Training Accuracy**
FGSM	10%	22.85%
PGD	6.3%	24.02%

Table 1: Accuracy comparison of models under adversarial training.

As it is possible to observe in the table 1, adversarial training can significantly reduce the effectiveness of adversarial attacks on the model's performance. Although there is a slight decrease in accuracy on clean examples.

5 Defenses

5.1 Random Self-Ensembling

Because adversarial training is not general enough, we search for other ways to make the network more robust to different kinds of attacks. One such proposal is Random Self-Ensembling [4], which tries to mitigate the effect of gradient-based attacks by adding noise layers before every convolution. Because of the strong noise variance suggested in the paper for these layers,

the final result tends to be unstable for the same input. This is mitigated by ensembling the output of the model many times during inference, and allows us to recover good accuracies.

Attack	With RSE	Without RSE
No attack	52%	55%
FGSM	10%	16%
PGD	6.8%	9.8%

Table 2: Model accuracy when applying and not applying RSE, for the basic kinds of attacks.

Table 2 shows the accuracy results when using RSE. Notice that although natural accuracy decreases, there is a considerable recovery in attack accuracy.

5.2 MixUp Defense

The Mixup Inference adversarial defense employs a stochastic interpolation procedure during the inference stage to effectively mitigate adversarial perturbations. The process involves computing K interpolations for each input, where K is a fixed hyper-parameter, typically set to 0.6 as suggested in the paper. For every input sample, the defense generates K interpolations by combining it with samples drawn from the dataset.

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_k \tag{1}$$

In the context of Mixup Inference, the results of normal training accuracy reveal a significant improvement, reaching 55%, whereas the accuracy without Mixup Inference is notably lower at 43%. This indicates that the introduction of interpolation during inference positively impacts the model's performance on normal data.

When evaluating the defense against adversarial attacks, particularly the Fast Gradient Sign Method (FGSM) attack, the results demonstrate a noteworthy enhancement. Without Mixup Inference, the accuracy in the face of FGSM attacks is 10%, but with Mixup Inference, the accuracy decreases slightly to 9.5%. This marginal decrease suggests that Mixup Inference contributes to maintaining a relatively higher accuracy under adversarial conditions.

Similarly, when subjected to the Projected Gradient Descent (PGD) attack, the defense mechanism showcases its effectiveness. The accuracy without Mixup Inference is 6.3%, whereas with Mixup Inference, the accuracy improves to 7.3%. This outcome underscores the utility of the Mixup Inference strategy in fortifying the model against adversarial attacks.

In essence, Mixup Inference serves as a robust adversarial defense by introducing stochastic interpolations during the inference stage. This technique not only bolsters the model's accuracy on normal data but also demonstrates a valuable impact on the model's resilience in the face of adversarial perturbations, as evidenced by the improved accuracy under FGSM and PGD attacks.

Finally we noted that as lambda decreases the model accuracy for natural images also decreases. Lambda can be seen as a parameter that controls how strong is the interpolation with other random images as it is possible to observe in the plot below (image 3)

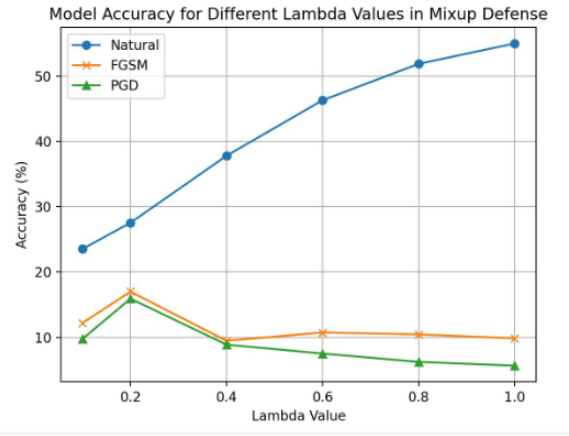

Figure 3: Accuracy for different lamda values

6 Conclusion

This project has provided significant conclusions into how important is to explore some kind of defense mechanisms to build a more robust neural networks. Our study started with implementing different adversarial attacks methods, such as FGSM, PGD, and the Carlini & Wagner Attack, which demonstrated the vulnerability of neural networks with some kind of perturbation. All three attacks implemented caused a drop in accuracy.

Adversarial training was the first method implement to reduce the effectiveness of these attacks. Although they worked well, we observed a decrease in accuracy on clean examples.

Furthermore, we explored two additional defense strategies: Random Self-Ensembling and Mixup-inference. Random Self-Ensembling, by introducing a noisy layer, showed really good results in mitigating gradient-based attacks. The second defense that was tried, Mixup inference, is based on a stochastic interpolation procedure that is done during inference, leading to an improvement in model accuracy.

In conclusion, the project contributed to the understanding of neural network robustness and how to safeguard these models against adversarial attacks.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017.

[2] Ting-Jui Chang, Yukun He, and Peng Li. Efficient two-step adversarial defense for deep neural networks, 2018.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[4] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble, 2018.

[5] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks, 2020.

[1] [2] [3] [4] [5]