Dauphine | PSL

Data Science Project A3 - IASD 2022-23
Training robust neural networks

Group ama_ADV: Adrien Golebiewski ,Alexis Hummel, Maximilien Wemaere

1 Introduction

For this last project of data science we have worked on a problem of attack and defense of neural networks (NN).

Deep neural networks have been shown to be sensible to adversarial attacks, slight modifications of normal inputs that are able to fool models. These modified inputs are called adversarial examples and the process of crafting them is called an adversarial attack.

Attack means trying to fool the NN by modifying its inputs. It consists in subtly modifying an original image such that the changes are almost undetectable to the human eye but will yet lead to mis-classification by a neural network. The most common constraint on those attacks are to impose an upper bound on how much perturbed examples can deviate from the original example - for instance in a $\ell\infty$ and $\ell 2$ ball.

The goal for the attack is to bring down the accuracy of the NN while attempting to modify the less its outputs. The defense have to reduce as much as possible the drop of accuracy.

2 Problem

The context of our work is well-defined : A basic CNN will be attacked by two types of white-box attacks: FGSM and PGD attack. Through our study, our objectives is to reduce the drop a accuracy cause by the attacks, by implementing differents defense strategies.

The toy-data set used is CIFAR10, which contains images which have to be classed into 10 different classes.

2.1 The Net

The net used is a basic CNN composed of two block of a convolutional layer, a pooling layer and a Relu activation then the channels are flatten for three linear layers.

Without attack, with 5 epochs of training the accuracy is 64%

2.2 Attacks

The attacks used are white-box attack, it means that they have access to the weights of the net to perturb the inputs of the net.

First attack is the Fast Gradient Sign Method (FGSM) which modify an image x by adding the sign of the gradient of the loss function.

$$x_{attacked} = x + \epsilon sign(\nabla_x l_f(x, y))$$

The ϵ, is a coefficient which convey the intensity of the attack. It have to be small enough to keep the images consistent.

A second attack is the Projected Gradient Based (PGM) which is pretty much the same method as FGSM, but the result is projected on a ball of center x and radius ϵ and we repeat several times

the operation for one image

$$x_0 = x$$
$$x_{t+1} = \Pi_{B(x_0,\epsilon)}(x_t + r sign(\nabla_x l_f(x, y))) \text{ (r is a stepsize)}$$

Best result are observed for PGM attack which are far better than FGSM (see 3). And moreover as observed on the 7 , PGM keep more consistency of the image than FGSM so it can be very useful for practical application.

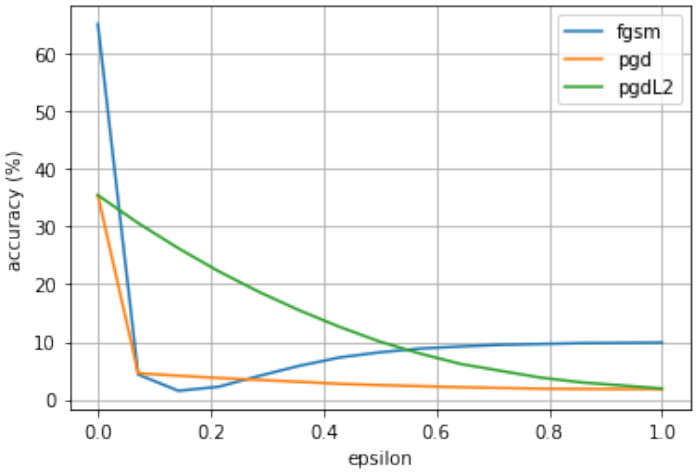
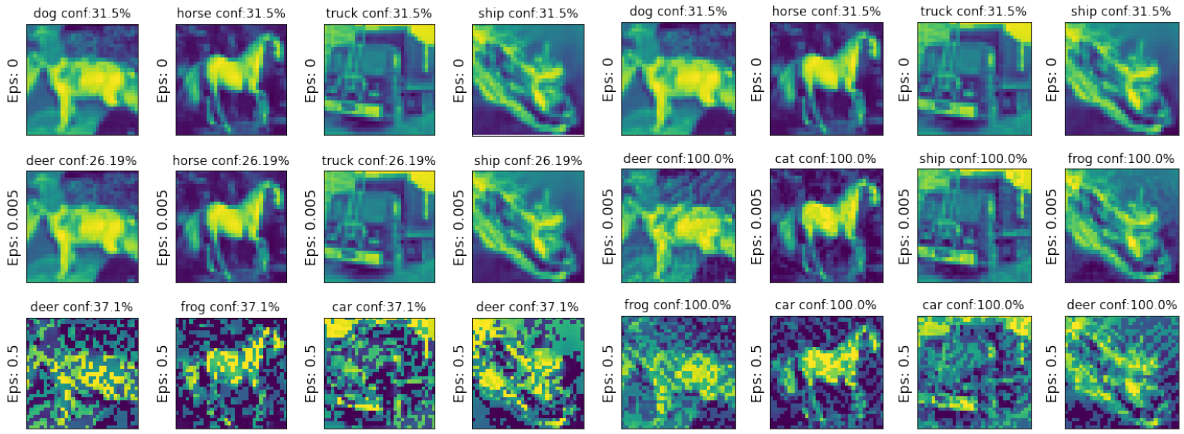


Figure 1: accuracy of the model depending on the ϵ of the attacks



(a) FGSM attack

(b) PGD attack

3 Classic defenses - Adversarial Training

A simple approach to defend against adversarial examples is to incorporate them during the training of our model.

3.1 Norm 2 and Norm infinity

Since adversarial training doesn't generalize well robustness, it could be interesting to compare the model performances under PGD-ℓ_2 attacks with ℓ_∞ training (or ℓ_∞-attack with ℓ_2 training).

Indeed, for large dimension images, $B_{\|\cdot\|_2}$ and $B_{\|\cdot\|_\infty}$ are mostly disjoints, given us different adversarial examples.

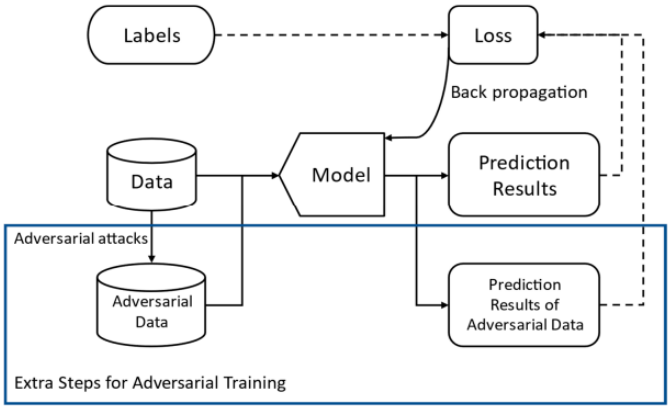


Figure 3: Adversarial training of a neural network model

3.2 Mixed Norm

Incorporating both ℓ_2 and ℓ_∞ during the training can improve our model robustness. We will see in the next session how Mixed Adversarial Training performs in comparison with ℓ_2 and ℓ_∞ training.

3.3 Comparative results and interpretation

Accuracy	AT-PGD$_{\ell_2}$	AT-PGD$_{\ell_\infty}$	AT-Mixed
ℓ_2-attack $\epsilon = 0.05$	27%	21%	45%
ℓ_{inf}-attack $\epsilon = 0.05$	9%	29%	28%
without attack $\epsilon = 0.05$	57%	49%	48%
ℓ_2-attack $\epsilon = 0.15$	44%	26%	41%
ℓ_{inf}-attack $\epsilon = 0.15$	11%	32%	29%
without attack $\epsilon = 0.15$	60%	52%	49%
ℓ_2-attack $\epsilon = 0.40$	35%	22%	35%
ℓ_{inf}-attack $\epsilon = 0.40$	16%	30%	25%
without attack $\epsilon = 0.40$	49%	49%	48%

Figure 4: Comparison of ℓ_∞, ℓ_2 and mixed adversarial training (5 epochs). All ϵ values showed for attack are the same for defense.

As excepted, AT-ℓ_∞ performs poorly under ℓ_2 in comparison with AT-ℓ_2 and vice versa. Using AT-Mixed provides us a way better accuracy under ℓ_∞-attack in comparison AT-ℓ_2 while limiting the impact on other accuracy (without attack and under ℓ_2-attack)

On the other hand, training with FGSM provides 23% of accuracy under FGSM-attack and 53% on clean data ($\epsilon = 0.15$). Besides, this results allowed us to conclude that the most efficient ϵ to train our model with is the one we could be attacked with.

4 Innovatives strategies and optimization

Secondly, we wanted to improve the robustness of our neural networks to attacks. To do so, we innovate by testing other strategies.

We explore three commmon ways of introducing randomness in the networks and improve generalization : noise injection at predict, NoiseNets and Denoise CNN.

4.1 Noise Injection

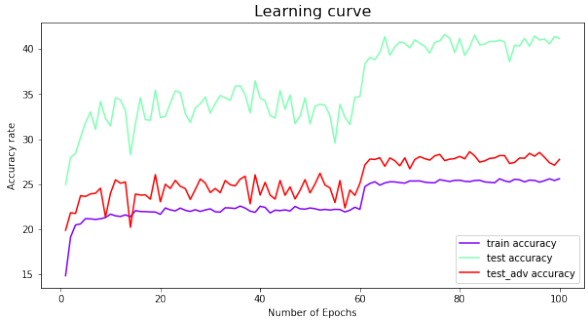
Noise injection is the injection of random noise at inference, taking x at input the model will predict fθ(x + η). As stated by Adnan Siraj Rakin and Zhezhi He - 2018, training the network with Gaussian noise is an effective technique to perform model regularization, thus improving model robustness against input variation. In this subpart, we propose to improve neural network robustness by adding noise in the input.

To isolate the effect of noise injection, we compared $\ell\infty$ norm PGD and FGSM adversarial training of BasicNets and two ResNets, introducing Gaussian Noise injection at prediction in one of them, and compare their results.

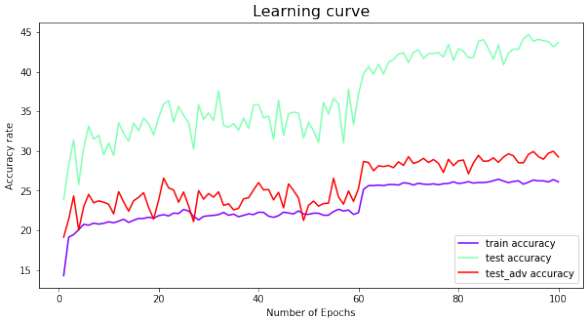
Below, the Adversarial test accuracies of a BasicNet and a wideResnet with and without noise injection at predict (in percentages) :

Accuracy	BasicNet	BasicNet with noise	ResNet with noise	WideResNet with noise
FGSM attack with l-inifnity	29.39%	37.26%	51.92%	68.7%
PGD attack with l-infinity	22%	29.26%	49.06%	60.81%

For the BasicNet, we observe a clear gain in test accuracies accross the board.



(a) Learning curves over 100 itrs of a BasicNet



(b) Learning curves over 100 itrs of a NoisyBasicNet

According to the table, the results for the ResNet are more surprising, as noise injection seem to increase adversarial robustness, with a much sharper increase. We globally notice, both for the Net with and without noise, a better score of Adversarial test accuracy with the FGSM attack.

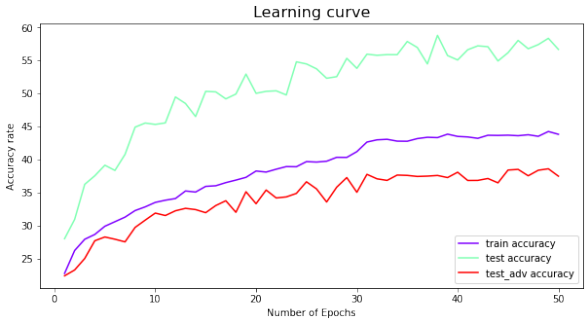
4.2 Network randomization

Among the popular defense mecanisms, we can find the randomization which has proven a pretty good efficiency in some context. The principles are simple, it consists in a noise injection both at inference and training time at selected layers. The procedure is therefore more complex than simple noise injection.

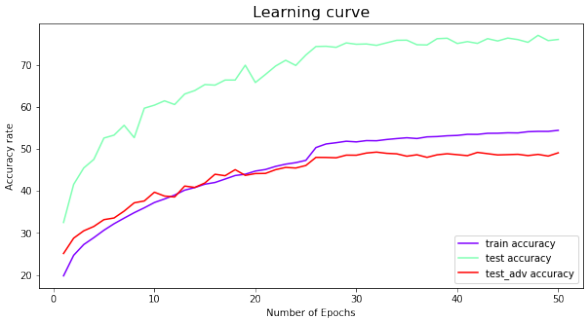
The common noises injected are drawn from the Exponential family such as Gaussian or Laplace distributions. For our experiments, we have chosen to use Gaussian noise, at first only during inference time, then during training. We got the following results for a basic classifier on CIFAR-10 with convolutional layers.

Accuracy	BasicNet	BasicNet with noise	ResNet with noise	WideResNet with noise
FGSM attack with l-inifnity	21.39%	29.26%	41.92%	62.20%
PGD attack with l-infinity	13%	24.25%	28.1%	37.49%

We tested this using different-sized ResNets (classical and wide Resnets in term of blocks number), one with noisy layers one without, following the same procedure as previously. We observe a significant increase in adversarial accuracy between this two implementations. It would be interesting to test different training schedules for a noisy version of the ResNet to fully exploit the architecture's potential.



(a) Learning curves over 50 itrs of a ResNet



(b) Learning curves over 50 itrs of a NoisyResNet

Due to lack of time and resources, we did not push our number of epochs very far, limiting them to 50. However, we observe a fairly different behavior of the Noise Net, with a slowly increasing plateau arising much earlier in training (around the 20th epochs contrary to the NoiseNet where the plateau appeared around the 40th epoch) and lasting longer, suggesting longer training time could have lead to better results.

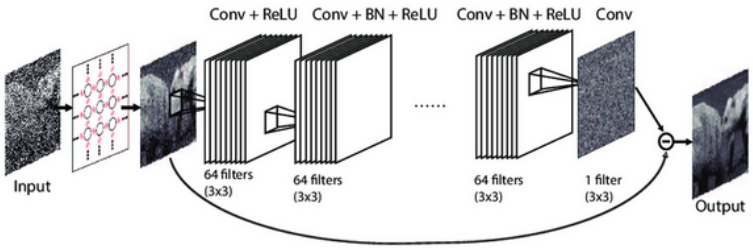
As for the Noising Injection, we notice globally, whether for the Nets with or without noise, a better Adversarial test accuracy with the FGSM attack

4.3 Denoiser CNN

We have also been working on a denoiser as innovative strategy of defense. It consist of a CNN used to denoise the input image of the model an trained aside with a dataset, containing as input, attacked images and as output, clean images. The random noise is added to each image by randomly picking pixel points from each image and randomly adding those pixel points to each image.

Rather than directly outputing the denoised image, the proposed DnCNN is designed to predict the residual image i.e., the difference between the noisy observation and the latent clean image.

The denoise of a convolutional is composed of layer+ReLU plus a series of blocks of convolutional+batch normalization + ReLU and a final convolutional layer. In other words, the proposed DnCNN implicitly removes the latent clean image with the operations in the hidden layers. For the tests, we set the number of hidden blocks to 10.



For these tests, we used 10 hidden blocks, and the attack was performed on the full model: the original CNN model plus the denoiser. The greater robustness of the Denoiser CNN results in greater accuracy compared with CNN without denoising. The best accuracy reach 50 percent with 15 epochs. As expected, under PGD attacks, it is very complicated to reinforce the network, but under FGSM accuracy tend to be better with this defense.

For further research, we could try to use a more complex architecture for our denoiser such as a U-net which is a quite good denoiser architecture.

Accuracy	No Denoiser	Denoiser 10 epochs	Denoiser 15 epochs
FGSM $\epsilon = 0.05$	9.96%	37.65%	41.92%
FGSM $\epsilon = 0.15$	1.72%	47.95%	50.6%
PGD $\epsilon = 0.001$	1.3%	24.25%	28.1%

Through these 3 methods of adversarial training by noising, we notice that more capacity drastically improve robustness and in our case, the accuracy metric. As part of our research, we have trained a few models over 50 iterations, like a 'deeper' VGG model to name but a few. Unsurprisingly, the example is striking with the VGG net : more capacity (about 10 million parameters) implies higher accuracy (as shown in the graph below) and more robust architecture.

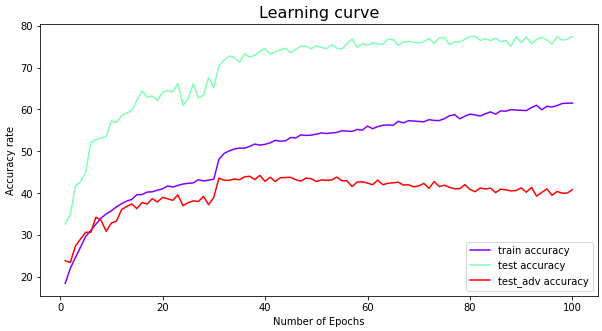


Figure 7: Learning curves over 100 itrs of a VGGNet

The intuition put forward by Madry et al. 2017 being that robust classification requires a much more complicated decision boundary, as it needs to handle the presence of possible adversarial examples.

5 Conclusion and perspectives

Through our work, we have found some ways to make networks robust to different attacks, notably through adversarial training and other more advanced randomization techniques. We have particular explored the trade off between robustness and accuracy, with our best model only reaching around 70 percent of accuracy for a fairly network in comparison to the dataset.

The two most promising defenses avenues for further exploration, given more time and resources, would be looking at :

• NASNets: an automated and theoretically informed network architecture optimization procedure (see Guo et al. n.d. and Liu and Jin 2021).

• Bayesian networks: as described in Panousis, Chatzis, and Theodoridis 2021 seem to be the most promising avenue to achieve greater performance with similar resources. It achieves interesting performances using a stochastic competition-based activation.

Finally, in this study, we have decided to focus only on White-Box gradient based attacks. It would also have been relevant to consider other attack approaches such as Black Box attacks. Current neural network-based classifiers are susceptible to adversarial examples even in the black-box setting, where the attacker only has query access to the model.

References

[1] Explaining and Harnessing Adversarial Examples [Anonymous authors] https://arxiv.org/abs/1412.6572

[2] Towards Deep Learning Models Resistant to Adversarial Attacks. https://arxiv.org/abs/1706.06083

[3] Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising https://doi.org/10.11092Ftip.2017.2662206

[4] Stochastic Local Winner-Takes-All Networks Enable Profound Adversarial Robustness. Konstantinos P. Panousis, Sotirios Chatzis, Sergios Theodoridis.

[5] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30.

[6] Advocating for Multiple Defense Strategies against Adversarial Examples. Alexandre Araujo, Laurent Meunier, Rafael Pinot, Benjamin Negrevergne. https://hal.archives-ouvertes.fr/hal-03118649/document