Robustness and Curvature in Deep Learning

Vincent Bardusco
MVA
Télécom Paris
vincent.bardusco@telecom-paris.com

Matthieu Carreau
MVA
Télécom Paris
matthieu.carreau@telecom-paris.fr

ABSTRACT

This report summarizes some of the main contribution of the articles [2], [3] and [5], Fawzi, Moosavi-Dezfooli et al. on the robustness of image classifiers, and offers new ideas to complete them. These articles published between 2017 and 2019 study neural networks classifiers from a geometrical and topological point of view and propose several methods to increase the robustness in practice. In section 1 of this report, we introduce the problem of robustness in deep learning, then section 2 presents results from the articles and the experiments that we reproduced, and we eventually discuss limitations and possible extensions of these works in section 3.

1 CONTEXT

Deep neural networks have shown increasingly impressive results in a wide range of tasks, from prediction or image classification to answering questions and assisting with complex activities. As the capabilities of artificial intelligence grow, so do the responsibilities of these systems, with access to important data and decisions that can have a huge impact on people's lives, from credit scores to facial recognition for instance.

On the one hand, artificial intelligence systems have been presented with increasing performances. In scientific papers, they reach impressive scores on common evaluations, and they also give results with very high confidence. We could then assume that those systems become safer with time, being able to give precise results on the most difficult tasks, on which even humans would be challenged, revealing how useful they could be.

On the other hand, it seems that these advanced systems can indeed be fooled by very simple tricks. Adversarial perturbation attacks consist of taking samples on which a classifier gives a correct output and shifting it a bit, which often leads to a wrong classification. The most common example is to take an image classifier and add a small amount of noise to the input. For a human, the difference is so small that it is impossible to distinguish a distorted sample from a real one. On this new sample, however, the classifier fails completely, giving an answer that has nothing to do with the original one. Moreover, with a smarter solution, such as a well-designed gradient ascent, it is possible to modify an image so that the classifier gives any chosen output with a probability higher than 99%.

Obviously, this reveals a huge irregularity in decision functions. In critical systems, where the decisions made by the algorithm can have important consequences, robustness is crucial. If classifying a cat as a camera can be funny, giving drastically different results to two almost identical financial situations can be dramatic.

As a result, a lot of work has been pursued to improve AI robustness. The main idea is adversarial training: after usual training, the model is fine-tuned on adversarial examples so that it becomes more robust on such small perturbations. However, even after adversarial training it seems always possible to find new examples where the model fails. As a result, it is important to better understand how the model can become more robust in general, and not making it robust so specific perturbations after-hand.

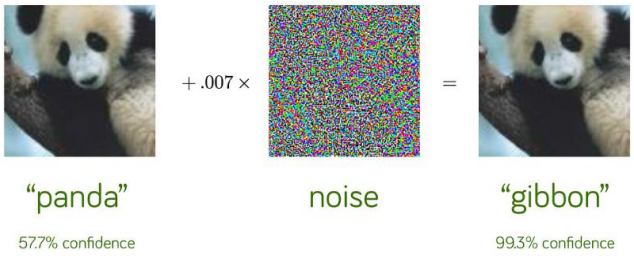


Figure 1: Example of noise-based adversarial attack on an image classifier

To do this, we will first study the structure of the decision functions, observing how adversarial training affects the geometry of these functions, and in particular how it regularizes the curvature. Then, we will interest ourselves in the topology of classification regions.

2 MAIN RESULTS OF THE ARTICLES

2.1 Robustness and Curvature

2.1.1 Link between robustness and regularity of decision region. When dealing with neural networks, it is extremely difficult to completely get a grasp on the structure of the decision function, and it is even harder to link the geometry of the decision space with relevant characteristics of the model.

In [5], authors offer the idea that the property of robustness, which means that two close samples will be classified similarly by the network (even though 'close' is yet to define as a lot of metrics can be used to comparate samples), is in fact linked with the curvature of the decision region. More precisely, they claim that if we visualize the decision border of a robust network, we will see that it is more flat and more regular than its of a non-robust network.

In this paper, 'robustness' is linked with resistance to perturbation-based adversarial attacks. A robust algorithm must then give the same output for a sample and its slightly-shifted version. To make comparisons, they used a usual ResNet classification network trained on CIFAR-10 or SVHN, and they used adversarial training to create a robust version of it. Indeed, networks trained against adversarial attacks with techniques such as Defense Distillation (presented in [6]) or with modified loss functions (shown for example in [4]) show great results against perturbation-based adversarial attacks, thus becoming arguably more robust.

While reproducing their process, we decided to expand a bit the experiments by using a dataset with more classes. Indeed, studying networks' robustness is especially important for huge models.

To reach a good trade-off between data complexity and computing power requirements, we settled on using the Tiny-Imagenet dataset, composed of images similar to those in Imagenet. This dataset contained 100000 images of 200 different classes, which is massive and complex enough for our experiments.

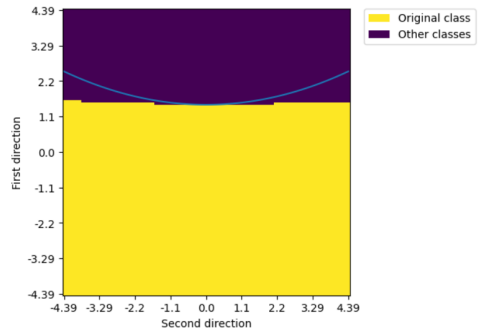
For implementation details, we trained from scratch a ResNet-512 architecture on this dataset using *AdamW* optimizer with a starting learning rate of $1e-02$ and an exponential decay learning rate scheduler of parameter $gamma = 0.7$ for 14 epochs reaching about 37% accuracy on the validation set. In all the experiments, we assumed not to focus too much on getting the highest accuracy possible, as we are more interested in studying robustness and geometry of the decision function, which does not require state-of-the-art classification performance.

We then copied our classification model and adversarial-trained this version using Defense Distillation as explained in [6]. Concretely, we began by building a dataset of adversarial examples. To do so, we took samples of the training set, and for each sample well classified by the initial model, we iteratively performed gradient ascent on the pixels of the sample until the initial second highest class probability become the first one, and the sample is thus misclassified while staying extremely close to the non-modified one. We built 50 adversarial samples per class using this method. Finaly, to create the adversarial-trained model, we fine-tuned our initial model on a set mixing adversarial samples and normal samples, until it reached about 30% accuracy on this adversarial set. For the record, the initial model reached about 10% accuracy on this adversarial set.

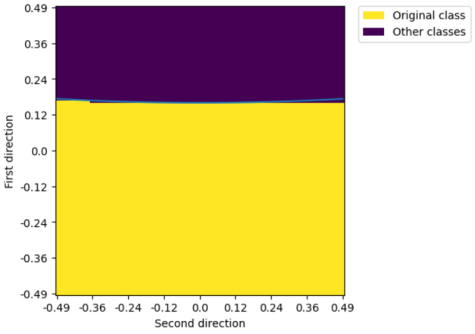
To visualize the regularity of the decision region, we used two different techniques. To begin with, we re-implemented the method presented in [5] to plot decision borders: the idea here is to build samples in a chosen 2D space to visualize the behaviour of the network in a certain region of the space of samples. To do so, we begin by choosing a sample (this sample will be fixed and always reused in all the experiments to be able to compare fairly the different models), and we will build an adversarial example just like we did earlier, using gradient ascent on the pixels. This adversarial sample will give us the direction to the decision border, which will be the first direction of our 2D space, and the second direction will be chosen randomly. Then, we fill the space by creating samples by shifting in the two chosen directions and computing for each sample the classification. For better curvature visualization, we also computed the curvature of the boundary decision (similarly to what is explained in [3]) and displayed the associated second order approximation.

As we can see in figure 2, the decision border is way more flat with adversarial-training than before, and it is especially highlighted by the fitted regression. This confirms both the expectations and results presented in [5].

As we also wanted to have a more general idea of the regularity of the decision function, we decided to visualize as well the decision region by taking two random directions in a wider area. Our idea with this setting was that a non-robust algorithm would have very small and concentrated regions for different classes in very close spaces (a bit like overfitted SVMs for instance), while robust networks will show wide and continuous regions.



(a) Initial model decision border



(b) Adversarial-trained model decision border

Figure 2: Decision borders near a lemon image for initial and adversarial-trained models

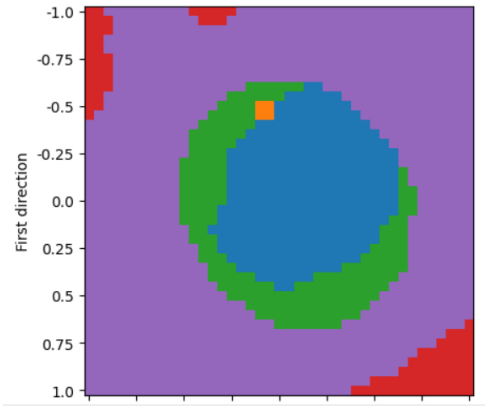
We can see in figure 3 that this is indeed well verified: on the same scale with the same central sample, the adversarial-trained model exhibits a way more regularized decision function with a wide central class, while the initial models shows lots of different classifications for very close samples.

2.1.2 CURE: imposing curvature regularization to gain robustness. Once we have noticed that robust networks exhibit more regularized decision functions and decision borders with low curvatures, it is logical to wonder if the opposite is true, which means if networks with regularized decision functions are also robust. Indeed, this would give birth to a new framework of robust networks where we could train them for robustness jointly with training them for performance, instead of trying to add robustness after training with adversarial training for instance, which often leads to decrease in performance.

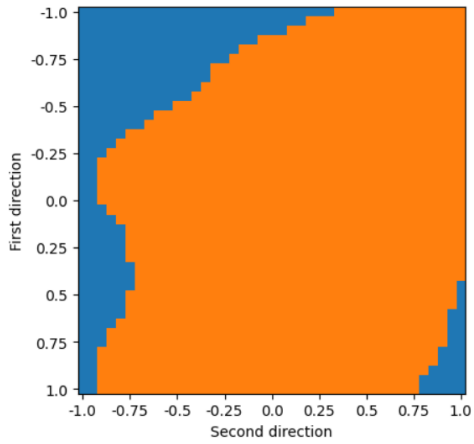
This is exactly the idea of the CURE method presented in [5]. Here, we add a regularization term directly in the loss so that the model is optimized both for giving right classifications and for outputing a low-curvature decision function. Concretely, we add to the usual cross-entropy loss the following term:

$$L_r = ||\nabla l(x + hz) - \nabla l(x)||^2$$

with l the cross-entropy loss and $z = \frac{\text{sign}\nabla l(x)}{||\text{sign}\nabla l(x)||}$.

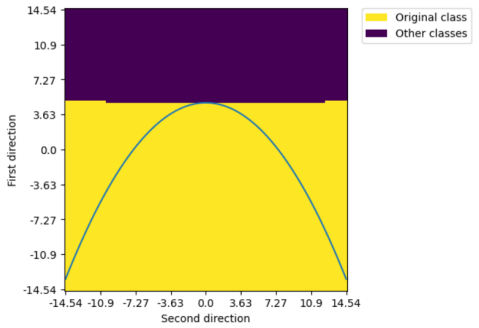


(a) Initial model decision regions

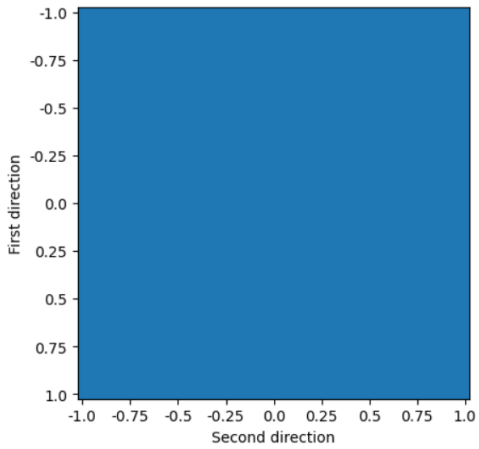


(b) Adversarial-trained model decision regions

Figure 3: Decision regions around a sample for initial and adversarial-trained models



(a) CURE model decision border



(b) CURE model decision regions

Figure 4: Results of experiments with CURE model

In practice, this new loss is way more costly than usual cross-entropy because it requires to do multiple forward pass but also multiple backward pass to compute all the terms. As a result, we could not train our CURE model from scratch on the 100000 samples in reasonable time due to computational costs, and we then decided to restrain the training to 10% of the dataset. With the same parameters as for the initial model, we trained the CURE version for 30 epochs, reaching 7% accuracy, which is way lower than before. We can think that on the one hand the model would require more time and compute to be trained properly, and on the other hand than restraining that much the dataset probably also impaired the performance.

The CURE model reached about 3% accuracy‘on the adversarial set, which is a lower decrease from the accuracy on the initial set than the initial model. However, we can argue that with such low performance values, it is difficult to draw convincing conclusions with these metrics.

To have a better idea of the real effects of this during-training regularization, we performed the same experiments as with the two other models.

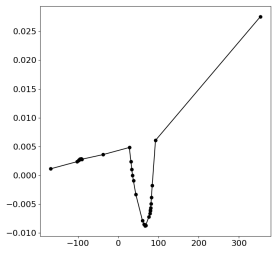
On figure 4, we can firstly notice that the decision border seems a bit more flat than with the initial model, but more importantly that the decision region is totally uniform around the chosen sample, highlighting a very high regularity, even better than with adversarial training.

We quantified the effects of the regularization by using 100 test images and computed for each of them a Monte-Carlo estimate of the norm of the Hessian of the loss, as well as the minimal norm of a perturbation needed to change their label. The results are summarized in 1, and we can see that the adversarial training improved slightly these metrics compared to the base model, but the CURE training outperforms by far the other ones with a significantly lower squared norm of the Hessian, and a larger distance to adversarial images.

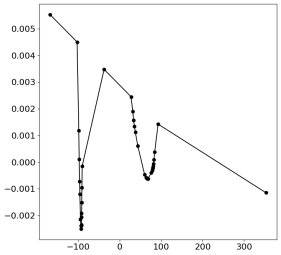
It shows that regularization can indeed be included during training by constraining the function’s curvature, and that it could become a safer way of reaching robustness than simple adversarial training. However, it seems to add a non-negligible computational cost to the training which has to be taken into account and which could be non-acceptable for bigger models with already important compute requirements.

	$\|\|H\|\|_F^2$	Perturbation size
Base model	5.5	0.67
Adversarial training	1.1	0.80
CURE training	6.4×10^{-4}	5.6

Table 1: Average values of the estimation of the squared norm of the Hessian and of the norm of the perturbation needed to change the predicted label for 100 test images



(a) Second vs. first principal components



(b) Third vs. first principal components

Figure 5: Low dimension visualizations of the path found between two natural images in the class *lemon*

2.2 Topology of classification regions

One of the main contributions of [3] is to propose an empirical validation of the statement that image classifiers learn path-connected classification regions. They present an algorithm to find a path between any two images that are similarly classified. Their empirical study focus on finding paths between an image from the validation dataset and three types of images: another image from the same set that has the same predicted label, an adversarial perturbation of a validation image that makes it classified similarly and a white noise image that is also perturbed to have the same label.

They find in all three cases that a continuous path between the two images exists and that it almost follows a straight line.

We reproduced some of these results by implementing the algorithm described in the article. We tested it on images that were in the same classification region and observed that in a lot of cases, a linear interpolation between two images was a valid path that stays in the decision region. When the path has to deviate from the straight line to stay in the region, we can visualize the path using a dimensionality reduction method, figure 5 present such a visualization of the projection of the path on the first principal components of the sequence of images, and figure 6 shows the ratio of the explained variance by the first components. We can see that the first component explains almost all the variance, which means that the path is almost straight which is coherent with what the results of the authors.

The conclusions of the authors is that the classification regions are path-connected. One limit of this reasoning is that the set of images on which the path-connectedness has been verified is composed only by natural images, adversarial perturbations of natural images and adversarial perturbations of white noise images. These three types of images do not necessarily cover all possible images,

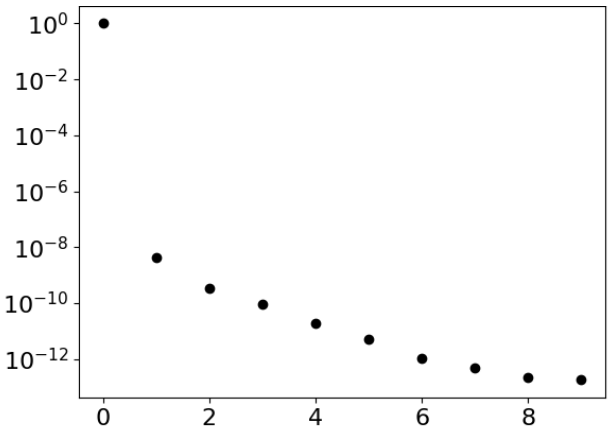


Figure 6: Ratio of explained variance by each of the first 10 principal components of the path

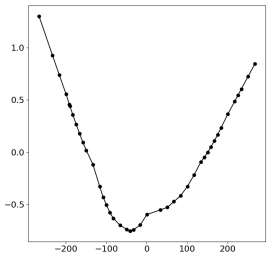
thus the path-connectedness is only verified empirically for these classes of images. Assessing the path-connectedness on the whole classification regions would be much more challenging, because the high dimensionality of the space of images makes it extremely difficult to describe exhaustively.

2.2.1 Complexity of the path for adversarial images. The paths found in the paper stay very close to straight lines, which is quite counter-intuitive. In all the scenarios tested in the paper, one of the two images that they try to connect is a *natural* image taken from from the validation set that has not been modified.

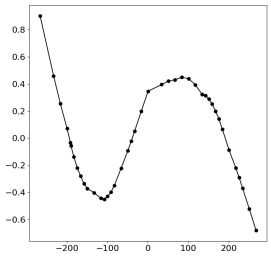
Therefore, we wanted to know if this still holds in more pathological cases. More precisely, in the case of a pair of images that are both perturbed to be predicted with the same new label, our goal was to quantify how much the path found by the algorithm deviates from a straight line. To do so, we tested the proposed algorithm on $N_{nat} = 18$ pairs of natural images *naturally* belonging to the same class and on $N_{adv} = 30$ pairs of adversarial images that were perturbed towards the same region (with a different label from both of the original ones).

For each path, represented by a list of images, we focus on two metrics which measures the complexity of the path. The first one is the number of points needed to represent it. The second one is the maximum of the distances between each of these images and the straight line between the two extremities, that we normalize by the length of this straight line.

For each type of pair of images, the mean and standard deviation of these metrics are reported in 2. These results show a significant increase in complexity for both metrics in the case of adversarial images, they need more points to be represented, and the mean deviation from the straight line is one order of magnitude above. However, this deviations remains relatively small and hard to detect by the human eye. More qualitatively, figures 7 and **??** illustrate an example of a path between two images from random classes that were adversely perturbed to be in the *lemon* region. We observe that the variance explained by components decreases slower in the case of adversarial images. Moreover, the deformation of the path



(a) Second vs. first principal components



(b) Third vs. first principal components

Figure 7: Low dimension visualizations of the path found between two adversarial images in the *lemon* region

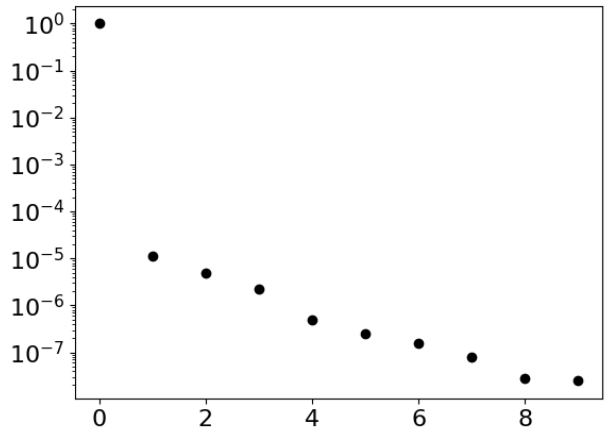


Figure 8: Ratio of explained variance by each of the first 10 principal components of the path

is more global, whereas for natural images, it only deviated at two precise locations from the straight line.

	Number of points	Normalized distance
Natural	11 ± 9	$2.7 \times 10^{-4} \pm 2.8 \times 10^{-4}$
Adversarial	70 ± 15	$3.0 \times 10^{-3} \pm 1.0 \times 10^{-3}$

Table 2: Mean and standard deviation of metrics for the paths found for the two types of image pairs

3 LIMITATIONS AND EXTENSIONS

The main limitation of all this techniques to improve robustness lies in the definition of said robustness. Indeed, we considered that a robust model is a model that gives the same output for close samples. And we assumed that such proximity was determined by a metric such as L_2 distance. However, these definitions have important flaws.

To begin with, it is always possible to find an adversarial sample with perturbation, even with adversarial training or regularization such as CURE. Indeed, it may simply take longer but gradient ascent will always give a new sample where the model fails. Obviously, we can argue that with a good enough training or regularization, this perturbed sample will be pretty far from the initial one, being very noisy for example. However, no threshold or quantified way has been given to determine at which point a sample is not 'close' anymore from the other and thus would not be qualified to tell a model is not robust because is misclassifies it. Qualitatively, we could say that a sample becomes 'far' when a human can easily distinguish one from the other, but once again this is not suitable to give a proper definition of robustness.

(a) Red pepper classified as pepper by all models

(b) Colour-changed pepper classified as orange by all models

Figure 9: Example of perceptual attack

More importantly, L_2 distance or similar metrics are not suited at all to compare images. Indeed, if we take an object and change its colour, for instance a red and blue umbrellas, they are extremely similar, and are in fact the same object, but a pixel-based distance between the two will be very high.

Actually, models do fail on such attacks that we can call 'perceptual attacks' where we change the colour or texture of objects. To highlight this, we used an image of red pepper, and changed the red colour to an orange colour (which is an alright colour for a pepper), as we can see in figure 9, and while all models well classified the red pepper as a pepper, the orange pepper is always classified as an orange, even by adversarial-trained and CURE models, which were shown as more robust. Other examples of perceptual attacks based on colour and textures are presented in [1].

All the methods described are relying only on pixel-based distance proximity, and thus do not consider such attacks at all, and models that fail in these cases are therefore classified as robust. However, these semantic attacks are much more likely to occur in real life than noise-based attacks, which produce unrealistic images. It is then arguably more important to build models that are robust to semantic perturbations, even if they still fail on common adversarial attacks, because they will be more robust to the real cases in which they are used, which is indeed the initial goal we are aiming for.

REFERENCES

[1] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A. Forsyth. 2019. Big but Imperceptible Adversarial Perturbations via Semantic Manipulation. *CoRR* abs/1904.06347 (2019). arXiv:1904.06347 http://arxiv.org/abs/1904.06347

[2] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2017. The Robustness of Deep Networks: A Geometrical Perspective. *IEEE Signal Processing Magazine* 34, 6 (2017), 50–62. https://doi.org/10.1109/MSP.2017.2740965

[3] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. 2018. Empirical Study of the Topology and Geometry of Deep Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3762–3770. https://doi.org/10.1109/CVPR.2018.00396

[4] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJzIBfZAb

[5] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. 2019. Robustness via Curvature Regularization, and Vice Versa. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9070–9078. https://doi.org/10.1109/CVPR.2019.00929

[6] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2015. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. *CoRR* abs/1511.04508 (2015). arXiv:1511.04508 http://arxiv.org/abs/1511.04508